



IBM Systems - iSeries

Кластеры управления системами

Версия 5, выпуск 4





IBM Systems - iSeries

Кластеры управления системами

Версия 5, выпуск 4

Примечание

Перед началом работы с этой информацией и описанным в ней продуктом рекомендуется ознакомиться со сведениями, приведенными в разделе “Примечания”, на стр. 155.

Седьмое издание (февраль 2006 года)

Это издание относится к версии 5, выпуску 4, модификации 0 IBM i5/OS (код продукта 5722-SS1), а также ко всем последующим выпускам и модификациям, если в новых изданиях не будет указано обратное. Данная версия работает не на всех моделях систем с сокращенным набором команд (RISC) и не работает на моделях с полным набором команд (CISC).

© Copyright International Business Machines Corporation 1998, 2006. Все права защищены.

Содержание

Кластеры	1	Производительность кластера	113
Новое в выпуске V5R4	1	Завершение работы заданий кластера	114
PDF для печати	2	Подсистема контроля и управления ресурсами (RMC)	114
Кластеры - Введение	3	Пользовательские очереди и структура заданий	115
Достоинства кластеров	3	Управление пользовательскими профайлами на узлах кластера	117
Принципы работы кластера	3	Резервное копирование и восстановление кластеров.	117
Основные сведения о кластерах	4	Сохранение конфигурации кластера	118
Структура кластера	7	Примеры: Конфигурации кластеров	119
События кластера	18	Пример: Простой кластер из двух узлов.	119
Кластерные приложения	29	Пример: Кластер из четырех узлов	120
Проектирование кластеров	73	Пример: Кластер из двух узлов с перемещаемыми дисками	121
Способы настройки кластеров и работы с ними.	74	Пример: Управление равноправными ресурсами в домене управления кластером	122
Требования к работе кластеров	81	Пример: Независимые ASP с географической зеркальной защитой	124
Разработка структуры кластера	83	Устранение неполадок	125
Защита кластера	91	Проверка работоспособности кластера	125
Контрольная таблица настройки кластера	93	Сбор данных для восстановления кластера	126
Сервер INETD	96	Определение причины неполадки с помощью команды Dump Cluster Trace (DMPCLUTRC)	126
Настраиваемые параметры связи кластера	96	Определение причины неполадки с помощью макроса CLUSTERINFO	131
Справочная таблица по удалению кластера	98	Наиболее частые неполадки кластеров	137
Планирование домена управления кластером	98	Распад кластера	139
Настройка кластеров	99	Прочие неполадки	143
Создание кластера	100	Часто задаваемые вопросы об управлении кластером с помощью Навигатора iSeries	146
Управление кластерами	101	Сервисные центры по поддержке кластеров.	152
Добавление узла в кластер	102	Информация, связанная с кластерами	152
Запуск узла кластера	103	Приложение. Примечания	155
Завершение работы узла кластера	103	Программный интерфейс	157
Коррекция версии кластера	104	Товарные знаки.	157
Удалить кластер	105	Условия	157
Создание CRG	105		
Запуск CRG	106		
Изменение домена восстановления для CRG	106		
Принудительный перенос ресурсов	107		
Добавление узла в домен устройств	108		
Удаление узла из домена устройств	109		
Влияние системных событий на кластер	110		
Создание домена управления кластером.	110		
Добавление записей отслеживаемых ресурсов	110		
Мониторинг домена управления кластером	111		
Контроль состояния кластера	112		

Кластеры

Кластеры - это группы серверов iSeries, работающие в общей среде и обеспечивающие близкий к 100 процентам коэффициент готовности для приложений, устройств и данных.

Организация кластеров упрощает управление системами и позволяет прозрачно для пользователя масштабировать серверы по мере роста требований к производительности.

Используя эти примеры, вы автоматически соглашаетесь с условиями Лицензирование и отказ от гарантий на предоставляемый код.

Новое в выпуске V5R4

В этом разделе перечислены изменения, внесенные в данной версии.

Поддержка домена управления кластером

Домен управления кластером отслеживает и синхронизирует изменения выбранного узла кластера. С помощью домена управления кластером удобнее работать с атрибутами общих ресурсов кластера, например, переменными среды и пользовательскими профайлами. Дополнительные сведения о домене управления кластером приведены в разделах:

- “Домен управления кластером” на стр. 8
- “Планирование домена управления кластером” на стр. 98
- “Справочная таблица домена управления кластером” на стр. 99
- “Создание домена управления кластером” на стр. 110

Поддержка равноправной группы ресурсов кластера (CRG)

Во все интерфейсы CRG добавлена поддержка равноправной группы ресурсов кластера (CRG). *Равноправная группа ресурсов кластера (CRG)* - это перемещаемая CRG, в которой роли каждого узла домена восстановления в восстановлении ресурсов одинаковы. Дополнительная информация приведена в следующих разделах:

- Группа ресурсов кластера
- “Создание CRG” на стр. 105
- “Запуск CRG” на стр. 106

Расширения кластера



Для улучшения операций отключения, а также для устранения неполадок в кластерной среде добавлены некоторые расширения функций кластера. Сюда входит:

- Оптимизированный подход к завершению работы кластерных служб на узле при отключении всех активных подсистем, при отключении или завершении работы системы. Дополнительная информация приведена в разделе “Влияние системных событий на кластер” на стр. 110.
- Возможность настройки новой CRG приложений с активным устойчивым IP-адресом. Дополнительная информация приведена в разделе “Создание CRG приложения с активным устойчивым IP-адресом” на стр. 106.
- Возможность устранения неполадок, связанных с кластером, путем просмотра с активного узла сведений обо всем кластере и связанных с ним CRG. Дополнительная информация приведена в разделе “Сбор данных для восстановления кластера” на стр. 126.

- | • Добавлена новая информация о средствах отладки и результатах их работы. С помощью этих средств можно устранить неполадки кластера. Дополнительная информация приведена в следующих разделах:
 - | – “Определение причины неполадки с помощью команды Dump Cluster Trace (DMPCLUTRC)” на стр. 126
 - | – “Определение причины неполадки с помощью макроса CLUSTERINFO” на стр. 131

| Каким образом узнать о новых или измененных функциях

| Новая или измененная техническая информация обозначается следующим образом:

- | •  - изображение, обозначающее начало новой или измененной информации.
- | •  - изображение, обозначающее конец новой или измененной информации.




| Дополнительная информация о новом в этом выпуске приведена в разделе Информация для пользователей.

PDF для печати

В этом разделе приведена информация о том, как просмотреть или напечатать этот документ в формате PDF.

Для того чтобы просмотреть или загрузить этот документ в формате PDF, выберите ссылку Кластеры (около 938 кб).

Руководства по выполнению задач

- Clustering and IASPs for Higher Availability  (около 6,4 Мб). В этом руководстве содержится обзорная информация о кластерах и средствах организации переносимых дисков на серверах iSeries.
- iSeries Independent ASPs: A Guide to Moving Applications to IASPs  (около 3,4 Мб). В этом руководстве приведены пошаговые инструкции по организации независимых ASP на серверах iSeries.
- Roadmap to Availability on the iSeries 400  (около 626 кб). В этом руководстве приведены пошаговые инструкции по организации независимых ASP на серверах iSeries.

Сайты в Internet

- Кластеры и высокая готовность  (www.ibm.com/servers/eserver/series/ha)
Сайт фирмы IBM, посвященный кластерам и обеспечению высокой готовности

Сохранение файлов в формате PDF

Для того чтобы сохранить файл PDF на вашей рабочей станции, выполните следующие действия:

1. Правой кнопкой мыши щелкните на файл PDF Web-браузере (нажмите правой кнопкой на ссылке, приведенной выше).
2. Если вы работаете в Internet Explorer, выберите **Сохранить объект как...** В Netscape Communicator выберите **Сохранить ссылку как...**
3. Перейдите в каталог, в котором требуется сохранить файл PDF.
4. Нажмите кнопку **Сохранить**.

Загрузка продукта Adobe Acrobat Reader

Для просмотра и печати файлов в формате PDF необходима программа Adobe Acrobat Reader. Вы можете загрузить ее с Web-сайта компании Adobe по адресу <http://www.adobe.com/products/acrobat/readstep.html>

(www.adobe.com/products/acrobat/readstep.html) .

Кластеры - Введение

В этом разделе описаны принципы работы кластеров. Обсуждаются достоинства кластеров и способы их максимальной реализации.

Кластер iSeries - это группа систем или логических разделов, работающих как единое целое. Системы, входящие в кластер, называются узлами кластера. В кластер iSeries может входить до 128 узлов. Кластеры - это эффективное средство масштабирования систем iSeries, обеспечивающее близкий к 100 процентам коэффициент готовности для приложений и данных. Создав кластер, можно быть уверенным, что ваши системы и приложения будут работоспособны 24 часа в день и 7 дней в неделю. Организация кластеров упрощает управление системами и позволяет прозрачно для пользователя масштабировать серверы по мере роста требований к производительности.

Достоинства кластеров

Кластеры дают возможность обеспечить доступность ресурсов операционной системы постоянно и бесперебойно.

- | С помощью кластеров можно существенно уменьшить количество и продолжительность непредвиденных и запланированных простоев.

Главные достоинства кластеров заключаются в следующем:

Высочайшая готовность

Кластеры обеспечивают постоянную абсолютную готовность систем, данных и приложений.

Простота обслуживания

- | Управление группой систем становится таким же простым, как управление одной системой или
- | одной базой данных. Не требуется выполнять одни и те же операции на каждом сервере. Управлять
- | общими ресурсами удобно с помощью домена управления кластером.

Масштабируемость

Значительно упрощается масштабируемость серверов по мере роста требований к производительности.

Понятия, связанные с данным

“Автоматический перенос ресурсов” на стр. 18

Автоматический перенос ресурсов - это автоматический перенос точки доступа к ресурсу на один из резервных серверов в случае сбоя сервера, выполняющего роль главной точки доступа к ресурсу.

Задачи, связанные с данной

“Принудительный перенос ресурсов” на стр. 21

Принудительный перенос ресурсов выполняется в случае перемещения пользователем точки доступа к ресурсу на другой сервер.

Принципы работы кластера

- | Инфраструктура кластера, входящая в состав i5/OS, называется службой ресурсов кластера и служит для
 - | обеспечения устойчивости важных ресурсов. Это могут быть данные, приложения, устройства и другие
 - | общие для нескольких клиентов ресурсы.
-
- | При выходе системы из строя ее функции начинает выполнять какая-либо другая система кластера. Доступ к
 - | данным можно организовать на основе двух моделей: иерархической и равноправной. Дополнительные
 - | сведения о группах ресурсов кластера (CRG), создаваемых на основе этих моделей, приведены в разделе
 - | Группа ресурсов кластера.

Понятия, связанные с данным

“Автоматический перенос ресурсов” на стр. 18

Автоматический перенос ресурсов - это автоматический перенос точки доступа к ресурсу на один из резервных серверов в случае сбоя сервера, выполняющего роль главной точки доступа к ресурсу.

“Репликация” на стр. 26

Репликацией называется процесс непрерывного копирования объектов в реальном времени. Объекты копируются с одного узла кластера на заранее выбранные другие узлы.

“Устойчивые устройства” на стр. 15

Устойчивыми устройствами называются физические устройства, представленные объектами конфигурации (например, описания устройств), доступные на нескольких узлах кластера.

“Устойчивые данные” на стр. 15

Устойчивые данные - это данные, хранящиеся в нескольких экземплярах на разных узлах кластера.

“Возврат узла в кластер” на стр. 21

Возвратом называется ситуация, когда в кластер вновь входит узел, ранее входивший в кластер и вышедший из него.

“Сравнение логической репликации, коммутируемых дисков и зеркальной защиты, распределенной по сайтам” на стр. 87

В этом разделе приведен обзор различных способов обеспечения устойчивости данных, применяемых в кластерной среде для повышения готовности.

Задачи, связанные с данной

“Принудительный перенос ресурсов” на стр. 21

Принудительный перенос ресурсов выполняется в случае перемещения пользователем точки доступа к ресурсу на другой сервер.

Основные сведения о кластерах

Прежде чем приступить к созданию и настройке кластера, ознакомьтесь с основными сведениями, приведенными в этом разделе.

Два основных понятия, связанные с кластером, это: узлы кластера и группа ресурсов кластера. *Узел кластера* - это либо сервер iSeries, либо логический раздел, входящий в кластер. Первоначальный набор узлов задается при создании кластера. *Группа ресурсов кластера (CRG)* - это управляющий объект для набора устойчивых ресурсов. В CRG могут входить все узлы кластера или какая-то часть. Кластер iSeries поддерживает четыре типа CRG: приложения, данных, устройств и равноправную. У всех этих типов есть два общих элемента: домен восстановления и программа выхода.

Домен восстановления определяет роль каждого узла в CRG. При создании CRG соответствующие объекты создаются на всех узлах, входящих в домен восстановления данного CRG. Помимо этого, создается единый системный образ объекта CRG, доступный всем активным узлам домена восстановления CRG. Любые изменения, внесенные в CRG, автоматически становятся известны узлам из домена восстановления.

Программа выхода вызывается во время событий в CRG, связанных с кластером. В частности, это может быть перемещение точки доступа с одного узла на другой.

CRG можно создать по двум моделям: иерархической и равноправной. В иерархической модели узлы домена восстановления CRG распределяются следующим образом:

- *Главный узел* - узел кластера, являющийся главной точкой доступа к устойчивому ресурсу кластера.
- *Резервный узел* - это узел кластера, который станет главной точкой доступа в случае сбоя главного узла или принудительного переноса ресурсов.
- *Узел-копия* - узел кластера, хранящий копии ресурсов кластера. Этот узел не может служить точкой доступа к ресурсу.

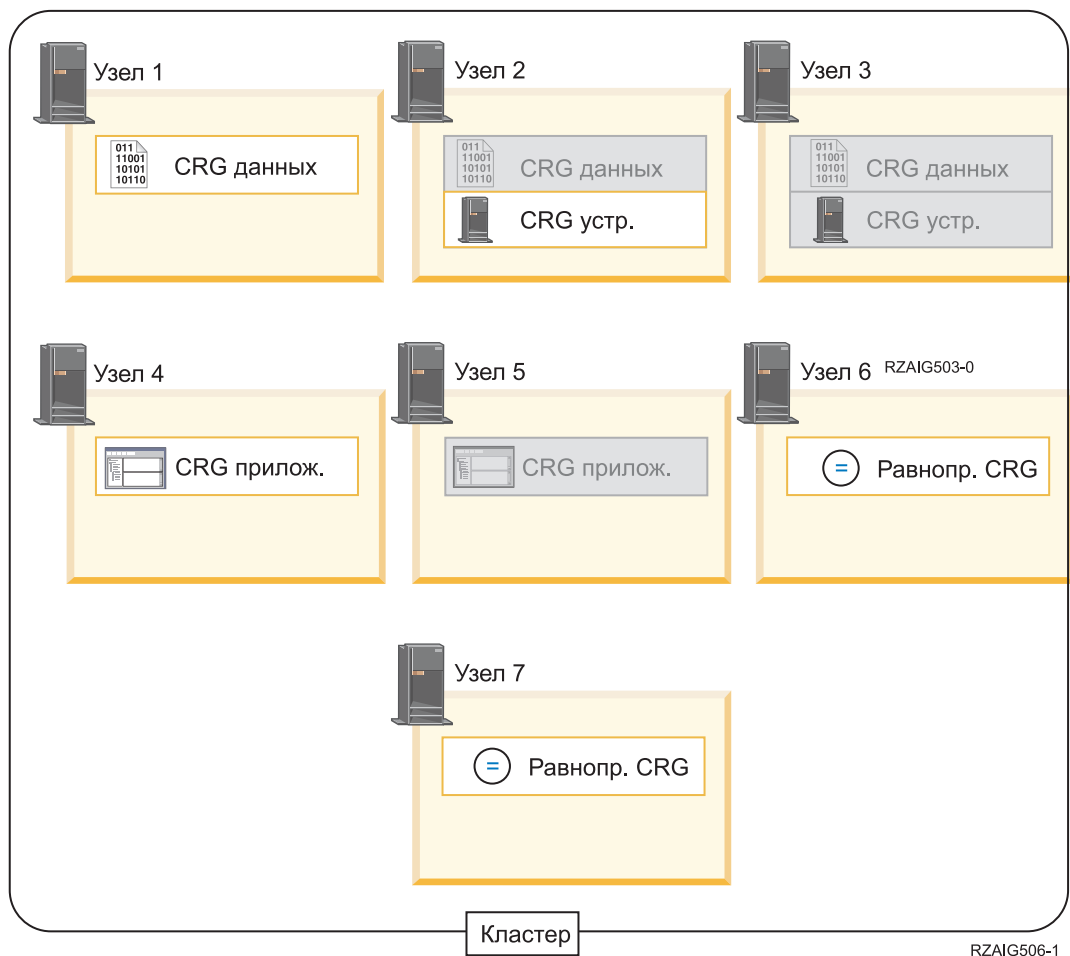
В равноправной модели отношения между узлами домена восстановления CRG равноправны. В равноправной модели узлы домена восстановления CRG распределяются следующим образом:

- *Равноправный узел* - узел кластера, который может стать активной точкой доступа к ресурсам кластера.
- *Узел-копия* - узел, на котором хранятся копии ресурсов кластера. В равноправных CRG узлы-копии представляют собой неактивные точки доступа к ресурсам кластера.

Участие узлов в процессе восстановления в равноправной CRG одинаково. Так как все узлы, в сущности, взаимозаменяемы, идея принудительного или автоматического переноса ресурсов в такой модели не имеет смысла. Отношения между узлами равноправны, поэтому в случае сбоя одного узла другие продолжают работу с того же места.

- | На основе равноправной CRG вы можете также создать домен управления кластером. Узлами этого домена
- | будут все равноправные узлы домена восстановления CRG. Узлов-копий в данном случае нет.

В приведенном ниже примере определено по одной CRG каждого типа:



RZAIG506-1

CRG данных

CRG данных есть на узлах 1, 2 и 3. Домен восстановления для CRG данных состоит из узла 1 (главный), узла 2 (первый резервный) и узла 3 (второй резервный). Узел 1 служит главной точкой доступа. Узел 2 - первый резервный узел. На узле 2 находится точная последняя текущая копия ресурса, создаваемая посредством логического копирования. В случае сбоя или переноса ресурса узел 2 становится главной точкой доступа.

CRG приложения

CRG приложений есть на узлах 4 и 5. Домен восстановления для этого CRG состоит из узлов 4 и 5. Главной точкой доступа является узел 4. В случае сбоя или переноса ресурса главной точкой доступа становится узел 5. Требуется устойчивый IP-адрес.

| Равноправная CRG

- | Равноправная CRG есть на узлах 6 и 7. Это значит, что домен восстановления для равноправной
- | CRG состоит из узлов 6 и 7. В данном примере узлы 6 и 7 могут быть либо равноправными, либо

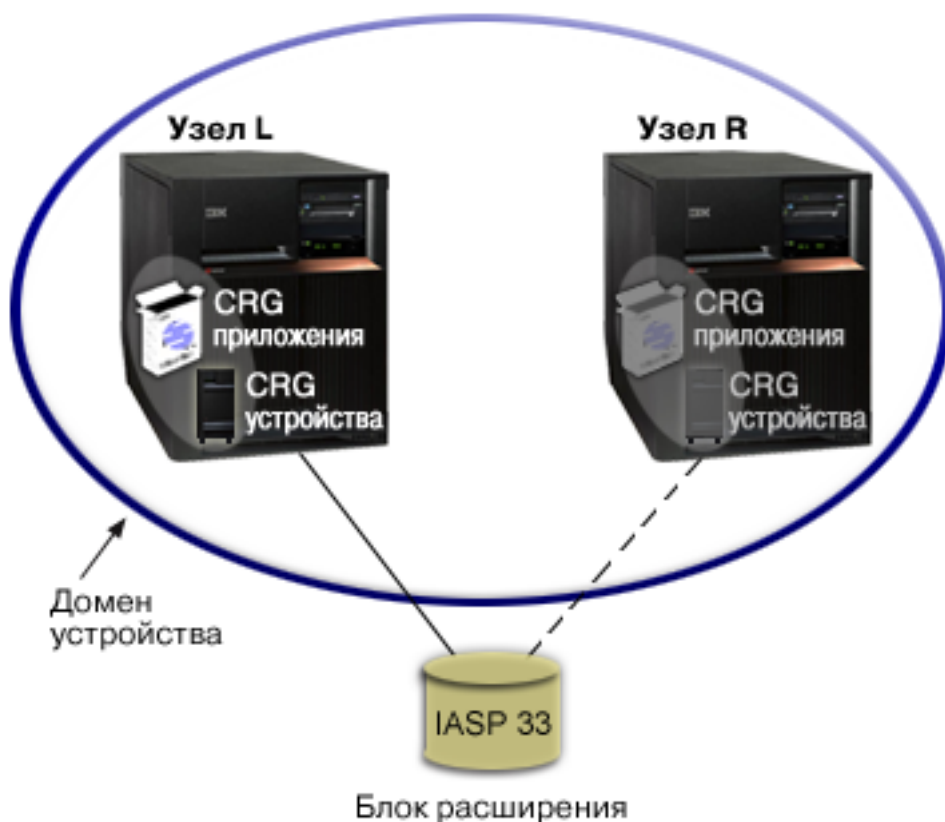
копиями. Если этот домен управления кластером представлен равноправной CRG, то изменения ресурсов, отслеживаемых этим доменом, будут синхронизироваться по домену из узлов 6 и 7 независимо от того, какой узел изменен.

CRG устройств

CRG устройств есть на узлах 2 и 3. Домен восстановления для этой CRG состоит из узла 2 (главная точка доступа) и узла 3. Это значит, что в настоящий момент главной точкой доступа для устройства, соответствующего CRG, служит узел 2. В случае сбоя или переноса ресурса главной точкой доступа становится узел 3.

Для создания CRG устройств требуется переносимый независимый ASP, подключенный к внешнему блоку расширения или к процессору ввода-вывода одного из логических разделов системы.

Узлы, входящие в домен восстановления CRG устройств, должны входить в один и тот же домен устройств. На следующем примере показан CRG устройств с доменом восстановления, состоящим из узлов L и R. Эти узлы входят в один домен устройств.



Понятия, связанные с данным

“Узел кластера” на стр. 7

Узлом кластера называется логический раздел или сервер iSeries, входящий в кластер.

“Группа ресурсов кластера (CRG)” на стр. 7

Группа ресурсов кластера (CRG) - это объект системы i5/OS, представляющий собой список ресурсов кластера, служащий для управления событиями в кластерной среде. Для группы ресурсов кластера должен быть задан домен восстановления и имя программы выхода CRG, вызывающейся при определенных событиях в кластере.

“Домен восстановления” на стр. 11

Домен восстановления - это множество узлов кластера, объединенных в CRG для выполнения определенной задачи - например, для обеспечения бесперебойного доступа к ресурсу или для синхронизации.

“Программы выхода CRG” на стр. 10

Программа выхода группы ресурсов кластера вызывается в случае возникновения событий CRG на уровне кластера.

Независимые пулы дисков

“Домены устройств” на стр. 16

Домен устройств - множество узлов кластера, совместно использующих ресурсы устройств. Узлы, входящие в домен устройств, совместно выполняют операции переноса точки доступа к набору устойчивых устройств.

Структура кластера

Кластер iSeries - это группа систем, работающих как единое целое. Приведенная здесь информация поможет разобраться с элементами и их взаимоотношениями.

Узел кластера

Узел кластера называется логический раздел или сервер iSeries, входящий в кластер.

Каждому узлу кластера присваивается 8-символьное имя и несколько IP-адресов, представляющих сервер iSeries. Узлам можно присваивать произвольные имена. Для простоты рекомендуется, чтобы имена узлов совпадали с соответствующими именами хостов или именами систем.

Связь между узлами кластера основана на протоколе TCP/IP. Набор узлов, входящих в кластер, называется списком членов кластера.

Группа ресурсов кластера (CRG)

Группа ресурсов кластера (CRG) - это объект системы i5/OS, представляющий собой список ресурсов кластера, служащий для управления событиями в кластерной среде. Для группы ресурсов кластера должен быть задан домен восстановления и имя программы выхода CRG, вызывающейся при определенных событиях в кластере.

Взаимоотношения узлов в кластере могут строиться на основе двух моделей: иерархической и равноправной. Эти модели можно применять как вместе, так и по отдельности.

Иерархическая модель

Узлы домена восстановления во всех CRG этой категории могут выполнять следующие роли: главный узел, резервный узел или узел-копия. Точками доступа к ресурсам кластера являются главный и резервные узлы. Однако активной точкой доступа в один момент времени может быть только один узел. Этот узел называется главным. Узлы-копии не могут быть точками доступа. Изменить это можно только преобразовав узел-копию в резервный узел. В иерархической модели предусмотрено три типа CRG: для реализации устойчивых данных, приложений и устройств. Устойчивость данных обеспечивается за счет хранения нескольких копий данных на разных узлах кластера, что позволяет безболезненно переносить точку доступа на любой резервный узел. Устойчивость приложения заключается в возможности его перезапуска на том же или другом узле кластера. Устойчивость устройства заключается в возможности перемещения точки доступа к устройству на другой узел кластера.

Для каждого CRG данных и приложений должна быть задана программа выхода. Программу выхода не обязательно задавать для CRG устройств.

В Навигаторе iSeries группа ресурсов кластера получает несколько иное значение.

- CRG устройств называется **переносимым устройством**.

- CRG приложений - это **переносимое приложение**.
- CRG данных называется **переносимой группой данных**.

Равноправная модель

- Узлы домена восстановления во всех CRG этой категории могут выполнять следующие роли: равноправный узел или узел-копия. Равноправные узлы могут быть точками доступа к группе ресурсов кластера. После запуска CRG все узлы, определенные как равноправные, будут точками доступа. Узлы-копии не могут быть точками доступа. Изменить это можно только преобразовав узел-копию в равноправный узел. В равноправной CRG на каждом узле содержатся одинаковые копии данных. Если узел выйдет из строя, сбойный элемент свяжется с остальными узлами кластера, и они продолжат работу с момента сбоя.

Домен управления кластером представляется равноправной CRG с доменом восстановления, все узлы которого равноправны.

Понятия, связанные с данным

“Домен восстановления” на стр. 11

- Домен восстановления* - это множество узлов кластера, объединенных в CRG для выполнения определенной задачи - например, для обеспечения бесперебойного доступа к ресурсу или для синхронизации.

“Программы выхода CRG” на стр. 10

Программа выхода группы ресурсов кластера вызывается в случае возникновения событий CRG на уровне кластера.

Управление группами ресурсов кластера

- При сбое узла выполняется автоматический перенос ресурсов. Порядок переноса CRG следующий: сначала переносятся все CRG устройств, затем все CRG данных, и напоследок CRG приложений. Для равноправных CRG порядок не важен, но каждый узел извещается о сбое.

- Для того чтобы определить, закончен ли перенос ресурсов, проще всего посмотреть на состояние CRG.

На время переноса данных приложения можно блокировать и вручную. Пока будет происходить перенос устойчивых данных, можно блокировать доступ к соответствующим CRG. Для блокирования применяются API Блокировать адрес EDRS (QxdaBlockEDRS) и the Проверить состояние блокировки EDRS (QxdaCheckEDRSBlock). При переносе ресурсов эти API позволяют блокировать и разблокировать доступ из программы выхода CRG.

Понятия, связанные с данным

“Автоматический перенос ресурсов” на стр. 18

Автоматический перенос ресурсов - это автоматический перенос точки доступа к ресурсу на один из резервных серверов в случае сбоя сервера, выполняющего роль главной точки доступа к ресурсу.

“Домен восстановления” на стр. 11

- Домен восстановления* - это множество узлов кластера, объединенных в CRG для выполнения определенной задачи - например, для обеспечения бесперебойного доступа к ресурсу или для синхронизации.

“Программы выхода CRG” на стр. 10

Программа выхода группы ресурсов кластера вызывается в случае возникновения событий CRG на уровне кластера.

Задачи, связанные с данной

“Принудительный перенос ресурсов” на стр. 21

Принудительный перенос ресурсов выполняется в случае перемещения пользователем точки доступа к ресурсу на другой сервер.

Домен управления кластером

- Домен управления кластером* служит для согласованного управления ресурсами узлов кластерной среды.

На каждом узле, служащем точкой доступа к устойчивым данным, приложениям и устройствам, необходимо определить ряд параметров конфигурации. Изменение какого-либо параметра на каком-либо узле должно распространяться на все остальные узлы кластера. Домен управления кластером предоставляет возможность идентификации ресурсов, которые должны быть согласованы на всех узлах домена. Кроме того, он отслеживает изменения этих ресурсов и синхронизирует их по активному домену. Домен управления кластером представляется равноправной CRG. При создании этого домена в системе создается равноправная CRG. Имя этой CRG соответствует имени домена управления кластером. Узлы, входящие в домен, определяются доменом восстановления равноправной CRG. Все узлы являются равноправными. Узлы копирования в домене управления кластером не разрешены. Узел в пределах кластера может быть определен только в одном домене управления. Дополнительная информация о задачах, связанных с доменом управления кластером, приведена в следующих разделах:

1. “Планирование домена управления кластером” на стр. 98
2. “Справочная таблица домена управления кластером” на стр. 99
3. “Создание домена управления кластером” на стр. 110
4. “Добавление записей отслеживаемых ресурсов” на стр. 110
5. “Запуск CRG” на стр. 106

Администрирование домена управления кластером осуществляется с помощью обычных функций CRG. Например, для добавления узла в домен управления кластером необходимо добавить узел в домен восстановления CRG и присвоить ему роль равноправного узла. Для запуска домена управления кластером нужно запустить равноправную CRG.

Путем запуска и завершения работы CRG можно управлять синхронизацией изменений. После завершения работы CRG изменения, внесенные в отслеживаемый ресурс на каком-либо узле домена, не будут распространяться на остальные узлы. Однако эти изменения распространятся когда CRG будет запущена снова. Когда CRG активна, изменения какого-либо отслеживаемого ресурса любого активного узла распространяются автоматически, поэтому ресурс остается согласованным по всему домену управления. Дополнительная информация приведена в разделе “Мониторинг домена управления кластером” на стр. 111.

Для добавления узла в домен управления кластером следует добавить этот узел в домен восстановления равноправной CRG. После этого на новом узле будут созданы все управляемые ресурсы, которые синхронизируются со всеми остальными узлами домена.

При удалении домена управления кластером все определенные в нем ресурсы удаляются с каждого узла домена. При этом сами ресурсы из системы не удаляются. Дополнительная информация приведена в разделе Отслеживаемые ресурсы.

Отслеживаемые ресурсы

Отслеживаемые ресурсы - это системные ресурсы, которыми можно управлять посредством домена управления кластером. В домене управления кластером этим ресурсам соответствуют *Записи отслеживаемых ресурсов (MRE)*.

Записям отслеживаемых ресурсов (MRE) соответствуют ресурсы, синхронизируемые посредством домена управления кластером. После добавления MRE в домен управления кластером изменения соответствующего ей ресурса на одном узле автоматически распространяется на все узлы активного домена. Для управления MRE служат три API рабочей среды:

- API Добавить запись отслеживаемого ресурса (QfpadAddMonitoredResourceEntry)
- API Удалить запись отслеживаемого ресурса (QfpadRmvMonitoredResourceEntry)
- API Показать сведения об отслеживаемом ресурсе (QfpadRtvMonitoredResourceInfo)

В домен управления кластером можно добавить записи для следующих ресурсов:

- Системные значения
- Пользовательские профайлы

- | • Описания заданий
- | • Класс
- | • Описание устройств независимых ASP
- | • Сетевые атрибуты
- | • Системные переменные среды
- | • Атрибуты TCP/IP

| В домен управления кластером можно добавить MRE только в том случае, если все узлы домена активны и входят в группу. Нельзя добавить MRE в распавшийся домен управления кластером. После добавления MRE изменения соответствующего ей ресурса будут автоматически распространяться на все активные узлы домена (если запущена CRG). Если CRG завершила работу, то все ожидающие изменения распространятся на активные узлы при следующем запуске CRG.

| С MRE связано глобальное состояние. Глобальное состояние для ресурса считается согласованным, если значения всех атрибутов ресурса, соответствующего MRE, совпадают для всех отслеживаемых атрибутов на всех узлах активного домена. Если попытка обновления ресурса на каком-либо узле доменом управления кластером оказывается неудачной, то глобальное состояние ресурса становится несогласованным. В такой ситуации администратору необходимо найти и устранить причину сбоя. При следующем обновлении домен управления кластером попытается синхронизировать ресурс. Скорее всего это произойдет в результате изменения ресурса после устранения причины ошибки обновления или при перезапуске CRG.

| Когда CRG домена управления кластером завершает работу, глобальное состояние всех MRE становится несогласованным. Причина следующая: во время завершения работы CRG могут измениться отслеживаемые ресурсы на различных узлах.

| Если ресурс, соответствующий MRE, является системным объектом, то для его удаления, переименования или перемещения в другую библиотеку следует сначала удалить MRE. В случае удаления, перемещения или переименования ресурса глобальное состояние MRE становится несогласованным, и последующие изменения этого ресурса не передаются в домен управления кластером.

| При завершении работы узла в домене управления кластером все MRE активного домена копируются на новый узел. На нем создаются все новые ресурсы, соответствующие этим MRE, и их атрибутам присваиваются те же значения, что и в активном домене.

| Если в домене управления кластером есть неактивные узлы, то любые изменения ресурсов этого домена распространятся эти узлы сразу же после возврата их в домен. Если домен управления кластером распался, то синхронизация изменений будет продолжаться, но уже среди активных узлов в каждой части. Когда узлы снова объединяются, домен управления кластером собирает все изменения от каждой части и распространяет их, синхронизируя ресурсы по всему домену. Если в разных частях кластера в один и тот же ресурс внесено несколько изменений, то после объединения частей все эти изменения обрабатываются доменом управления, правда, порядок их обработки непредсказуем.

| **Понятия, связанные с данным**

| “Домен управления кластером” на стр. 8

| *Домен управления кластером* служит для согласованного управления ресурсами узлов кластерной среды.

Программы выхода CRG

Программа выхода группы ресурсов кластера вызывается в случае возникновения событий CRG на уровне кластера.

| Программу выхода не обязательно применять для CRG устойчивых устройств, но она нужна для других типов CRG. Программа выхода группы ресурсов кластера, если она применяется, вызывается при возникновении событий на уровне кластера, в том числе при выполнении следующих операций:

- | • Узел неожиданно выходит из кластера.

- Узел выходит из кластера в результате выполнения API Завершить работу узла кластера (QcstEndClusterNode) или API Удалить запись узла кластера (QcstRemoveClusterNodeEntry).
- Кластер удален с помощью API Удалить кластер (QcstDeleteCluster).
- Узел активируется с помощью API Запустить узел кластера (QcstStartClusterNode).
- Восстановлена связь с узлом, состоящим из нескольких разделов.

Разработкой программ выхода занимаются фирмы, специализирующиеся на кластерном программном обеспечении, однако их можно создавать и самостоятельно.

Подробные сведения о программах выхода CRG, включая параметры, передаваемые для каждого кода действия, приведены в разделе Cluster Resource Group Exit Program документации по API кластеров.

Домен восстановления

- *Домен восстановления* - это множество узлов кластера, объединенных в CRG для выполнения определенной задачи - например, для обеспечения бесперебойного доступа к ресурсу или для синхронизации.

В домен входят узлы кластера, с которых доступен определенный ресурс. Множество узлов кластера, входящих в группу ресурсов кластера, образует главную точку доступа, резервные точки доступа, равноправные точки доступа и, при необходимости, узлы-копии.

Каждому узлу домена восстановления может соответствовать одна из следующих ролей:

Главный

Узел, выполняющий роль главной точки доступа к устойчивому ресурсу.

- В CRG данных главным будет узел, на котором хранится основная копия ресурса.
- В CRG приложений главным считается узел, на котором в данный момент работает приложение.
- В CRG устройств главным узлом будет текущий владелец ресурса устройства.

Примечание: В случае применения географической зеркальной защиты для узлов домена восстановления из CRG устройств необходимо указать место расположения IP-адреса портов данных. Дополнительные сведения приведены в разделе Место расположения и IP-адреса портов данных for details.

- В равноправной CRG главных узлов нет.

При сбое главного узла в CRG или при принудительном переносе ресурсов главная точка доступа CRG переносится на первый резервный узел.

Резервная точка доступа

Узел кластера, который станет точкой доступа в случае сбоя главного узла или принудительного перемещения ресурса.

- Для CRG данных это узел, на котором путем репликации поддерживается текущая копия данных.
- В равноправной CRG резервных точек доступа нет.

Копия Узел, на котором хранится копия ресурса, но который не может служить точкой доступа к ресурсу. Перенос ресурса на узел-копию не допускается. Если узел-копию нужно сделать главной точкой доступа, нужно сначала изменить его роль в кластере.

- В равноправных CRG узлы определяются как копии, представляющие собой неактивные точки доступа к ресурсам кластера.

Равноправный узел

Узел кластера, не входящий в иерархию. Этот узел может быть активной точкой доступа к ресурсам кластера. При запуске CRG активной точкой доступа могут быть все равноправные узлы.

- В равноправной CRG точка доступа полностью управляется приложением управления, а не системой. Равноправная роль поддерживается только равноправной CRG.

Иерархическая модель

При иерархической модели каждый узел в домене восстановления в каждый момент времени выполняет определенную роль. Эта роль называется *текущей ролью* узла в домене. Роль узла может изменяться вместе с состоянием кластера в результате запуска, завершения работы и сбоя узлов. Помимо этого, для каждого узла задана роль, которую он выполняет в нормальном состоянии кластера. Эта роль называется *предпочитаемой ролью* в домене восстановления. Предпочитаемая роль задается при создании CRG и не изменяется динамически. Предпочитаемая роль не зависит от текущего состояния кластера. Она изменяется только в результате изменения состава домена восстановления и при удалении узла из кластера. Предпочитаемые роли можно изменять вручную.

Домен восстановления в иерархической модели можно рассматривать следующим образом:

Таблица 1. Роли узлов в иерархической CRG

Узел	Текущая роль	Предпочитаемая роль
A	Резервный 1	Главный
B	Резервный 2	Резервный 1
C	Главный	Резервный 2
D	Копия	Копия

В данном примере иерархическая CRG состоит из узлов A, B, C и D. Узел C - текущий главный узел. Так как предпочитаемая роль для этого узла - второй резервный, то можно предположить, что текущая роль узла C стала результатом двукратного принудительного или автоматического переноса ресурсов. При первом переносе точки доступа роль главного узла перешла от узла A к узлу B, поскольку предпочитаемая роль узла B - первый резервный. При втором переносе ресурсов роль главного узла перешла к узлу C, поскольку это второй резервный узел. Предпочитаемая и текущая роли узла D совпадают: это узел-копия. При автоматическом или принудительном переносе ресурсов узел-копия не может считаться точкой доступа, если принудительно не сделать его основным или резервным.

Примечание: Роль узлов в домене восстановления можно изменять вручную. Данный пример иллюстрирует автоматическое изменение роли узлов домена восстановления без вмешательства пользователя.

Равноправная модель

При равноправной модели узел в группе ресурсов кластера может быть либо равноправным, либо копией.

Таблица 2. Роли узлов в равноправной CRG

Узел	Текущая роль	Предпочитаемая роль
A	Равноправный	Равноправный
B	Равноправный	Равноправный
C	Равноправный	Равноправный
D	Копия	Копия

В домене восстановления определены равноправные узлы A, B и C. Если узел A выходит из строя, он связывается со всеми узлами домена восстановления независимо от роли. Эти узлы продолжают операции с того места, на котором прервался узел A. Узел D содержит данные, но не возобновляет операции, поскольку он выполняет роль копии.

Сделать равноправными или копиями можно сколько угодно узлов. Равноправные узлы не упорядочиваются и могут стать активной точкой доступа к ресурсу кластера. Узлы-копии также не упорядочиваются, но не могут стать активной точкой доступа к ресурсам кластера, если не изменить их роль

на роль равноправного узла с помощью API Изменить группу ресурсов кластера (QcstChangeClusterResourceGroup).

Задачи, связанные с данной

“Изменение домена восстановления для CRG” на стр. 106

Вы можете произвольным образом менять состав домена восстановления, а также роли узлов в нем. Для группы ресурсов кластера устройств вы можете изменить место расположения и IP-адреса портов данных узла домена восстановления.

“Принудительный перенос ресурсов” на стр. 107

В результате принудительного переноса ресурсов текущий главный узел становится резервным узлом домена восстановления.

Версия кластера

Версией кластера называется идентификатор набора функций, которые может выполнять кластер.

Понятие версии необходимо для обеспечения полной совместимости систем кластера, поскольку на разных серверах могут быть установлены разные уровни программного обеспечения. Если в вашем кластере будут применяться системы с разными уровнями программного обеспечения, ознакомьтесь с разделом Кластеры смешанных версий.

Говоря о версиях кластера, нужно различать следующие понятия:

Потенциальная версия кластера

Максимальная из версий, поддерживаемая конкретным узлом. Данный узел при необходимости может работать в кластере указанной версии.

Текущая версия кластера

Версия, в которой выполняются все операции кластера на данный момент. Фактически это версия протокола связи, применяемого узлами кластера.

Номер потенциальной версии кластера увеличивается в каждом последующем выпуске операционной системы, в котором существенно расширен набор кластерных функций. Если текущая версия кластера ниже потенциальной версии, то новые функции не могут применяться, поскольку их могут поддерживать не все узлы кластера. Для применения новых функций в кластере нужно, чтобы все узлы поддерживали одну и ту же потенциальную версию, и чтобы потенциальная версия совпадала с текущей.

Когда узел входит в кластер, его потенциальная версия сравнивается с текущей версией кластера. Если потенциальная версия кластера для нового узла не соответствует с текущей версией кластера (N) и с последующей версией (N+1), то узлу запрещается вход в кластер. Текущая версия кластера первоначально задается первым узлом кластера согласно значению, указанному в команде или API создания кластера.

- | Например, предположим, что кластер должен состоять из двух узлов с версиями V5R3 и V5R4
- | соответственно. Тогда кластер можно создать одним из следующих способов:
- | • Создать кластер в системе V5R3, а затем добавить к нему узел V5R4.
- | • Создать кластер в системе V5R4, указав, что этот кластер должен поддерживать узлы предыдущих версий,
- | а затем добавить к нему узел V5R3.

В кластерах смешанных версий используются протоколы минимальной из используемых версий узлов. Эта версия задается при создании кластера. Она может быть равна либо потенциальной версии узла, создающего кластер, либо предыдущей по отношению к ней. Разница между максимальной и минимальной версиями всех узлов кластера не должна превышать 1.

После установки новой версии на всех узлах кластера можно повысить и версию кластера в целом, чтобы получить поддержку новых функций. Для этого нужно выполнить операцию коррекции версии кластера.

Внимание: Если номер новой версии операционной системы выше, чем номер текущей версии кластера,, то узел кластера при перезапуске выдаст ошибку. Исправить эту ситуацию можно путем удаления узла и повторного создания его с правильной версией.

- | **Внимание:** В случае применения в кластере переносимого независимого ASP на перенос между различными
- | версиями операционной системы накладываются ограничения. Необходимо перенести независимый ASP
- | предыдущего выпуска в систему с текущей версией i5/OS и активировать его. После того как он станет
- | доступным в текущем выпуске i5/OS, его внутреннее содержимое изменится и перестанет быть доступным в
- | предыдущей версии операционной системы.

Дополнительные сведения о версиях кластеров, в том числе информация об ограничениях для различных версий и таблица соответствия версий кластера выпускам i5/OS, приведены в документации по API кластера.

Понятия, связанные с данным

“Кластеры смешанных версий” на стр. 86

Процедура создания кластера, в который будут входить узлы с различными версиями кластера, требует некоторых дополнительных действий.

“Наиболее частые неполадки кластеров” на стр. 137

В этом разделе перечислены наиболее часто встречающиеся неполадки кластеров и обсуждаются способы их предотвращения и устранения их последствий.

Задачи, связанные с данной

“Создание кластера” на стр. 100

Процесс создания и настройки кластера начинается с настройки первого узла.

“Коррекция версии кластера” на стр. 104

Версия кластера - это версия протокола связи, применяемого узлами кластера для обмена данными.

Устойчивые ресурсы

Устойчивыми ресурсами называются системные ресурсы (данные, устройства, приложения), для которых в кластере обеспечена высокая готовность.

Для каждого устойчивого ресурса в кластере есть резервная точка доступа, которая задействуется в случае сбоя главной точки доступа.

Устойчивыми можно сделать следующие ресурсы системы:

1. Данные, копии которых хранятся на нескольких узлах.
2. Приложения, доступ к которым осуществляется по IP-адресу и которые можно перемещать с одних узлов на другие.
3. Аппаратные устройства, которые можно перемещать с одних узлов на другие.
- | 4. Равноправные ресурсы, поддерживаемые доменом управления кластером.

Схема взаимодействия узлов, связанных с устойчивыми ресурсами, хранится в объекте *группы ресурсов кластера (CRG)*. Группы ресурсов кластера хранятся в нескольких экземплярах на разных узлах кластера.

Понятия, связанные с данным

“Группа ресурсов кластера (CRG)” на стр. 7

- | *Группа ресурсов кластера (CRG)* - это объект системы i5/OS, представляющий собой список ресурсов
- | кластера, служащий для управления событиями в кластерной среде. Для группы ресурсов кластера
- | должен быть задан домен восстановления и имя программы выхода CRG, вызывающейся при
- | определенных событиях в кластере.

“Домен управления кластером” на стр. 8

Домен управления кластером служит для согласованного управления ресурсами узлов кластерной среды.

Устойчивые приложения:

Устойчивые приложения - это приложения, которые можно перезапустить на другом узле кластера без дополнительной настройки клиентов.

Требования, предъявляемые к устойчивым приложениям, описаны в разделе Обеспечение устойчивости приложений.

Для работы устойчивого приложения необходима возможность обнаружения сбоев связи по протоколу IP между клиентом и сервером. Приложение клиента должно получать информацию о том, что соединение IP будет временно недоступно, и в таких случаях должно повторять попытки доступа вместо того, чтобы завершать работу или начинать процедуру автоматического перемещения ресурсов. При принудительном переносе ресурсов приложения сервера должны получать информацию о том, что соединение IP более не доступно. Для этого в приложение сервера должно передаваться сообщение об ошибке. После получения сообщения об ошибке приложение сервера должно самостоятельно завершить работу.

Перенос IP-адресов - это базовая функция обеспечения высокой готовности сервера для приложений среды клиент-сервер. **Устойчивый IP-адрес приложения** - это плавающий адрес, назначенный приложению. Приложению назначается плавающий IP-адрес, на который перенаправляются данные, отправляемые на IP-адреса реальных серверов приложения. При сбое главного сервера приложения плавающий IP-адрес переназначается резервному серверу, при этом не требуется изменение конфигурации клиента.

Применение устойчивых IP-адресов тесно связано с понятием CRG приложений. CRG приложения - это группа ресурсов кластера, в которую входят ресурс устойчивого IP-адреса и домен восстановления. Домен восстановления - это список серверов, поддерживающих конкретное приложение. Если на одном из серверов происходит сбой, служба ресурсов кластера автоматически переносит точку доступа к приложению на другой сервер.

Понятия, связанные с данным

“Обеспечение устойчивости приложений” на стр. 30

Обеспечение устойчивости приложений.

“Домен восстановления” на стр. 11

Домен восстановления - это множество узлов кластера, объединенных в CRG для выполнения определенной задачи - например, для обеспечения бесперебойного доступа к ресурсу или для синхронизации.

Задачи, связанные с данной

“Кластерные приложения” на стр. 29

Устойчивость приложения - один из краеугольных камней кластерной среды. Если вы планируете разрабатывать и применять в кластере приложения с высокой готовностью, то следует помнить, что для таких приложений существуют свои требования к готовности.

Устойчивые данные:

Устойчивые данные - это данные, хранящиеся в нескольких экземплярах на разных узлах кластера.

Специальные механизмы репликации поддерживают копии данных на всех узлах домена восстановления. Резервные узлы в домене восстановления могут в любой момент принять на себя функции главной точки доступа к устойчивым данным. Узлы-копии также содержат копию данных, но не могут выполнять функцию главной точки доступа. Как правило, главный узел использует узел-копию как хранилище резервных копий или данных, предназначенных только для чтения.

Понятия, связанные с данным

“Репликация” на стр. 26

Репликацией называется процесс непрерывного копирования объектов в реальном времени. Объекты копируются с одного узла кластера на заранее выбранные другие узлы.

Устойчивые устройства:

Устойчивыми устройствами называются физические устройства, представленные объектами конфигурации (например, описания устройств), доступные на нескольких узлах кластера.

- | В случае сбоя точка доступа к ресурсу перемещается на первый резервный узел в домене восстановления
- | группы ресурсов кластера. Пример устойчивых устройств - независимые пулы дисков, или независимые ASP.
- | Они могут отключаться и подключаться, не затрагивая остальную систему. Кроме этого есть
- | географическая зеркальная защита, являющаяся частью функции зеркальной защиты, распределенной по
- | сайтам (XSM), которая, в свою очередь, входит в состав Компонента 41 (High Available Switchable Resources)
- | i5/OS. Географическая зеркальная защита - это функция, позволяющая в целях повышения готовности
- | хранить две одинаковых копии независимого пула дисков в двух местах. Копия, принадлежащая главному
- | узлу, называется рабочей копией, а копия, хранящаяся на резервном узле - зеркальной. Пользователи и
- | приложения работают с независимым пулом дисков на главном узле, т.е. с рабочей копией.

Группа ресурсов кластера (CRG) устойчивых устройств - это список устойчивых устройств. Каждое устройство соответствует независимому ASP. При сбое все устройства в списке автоматически переносятся на резервный узел. Помимо этого, все устройства группы можно совместно включать и отключать в рамках принудительного и автоматического переноса ресурсов. Существуют ограничения на физическую конфигурацию устойчивых устройств. Дополнительная информация о том, каким требованиям должны отвечать устойчивые ASP, приведены в разделе Независимые пулы дисков.

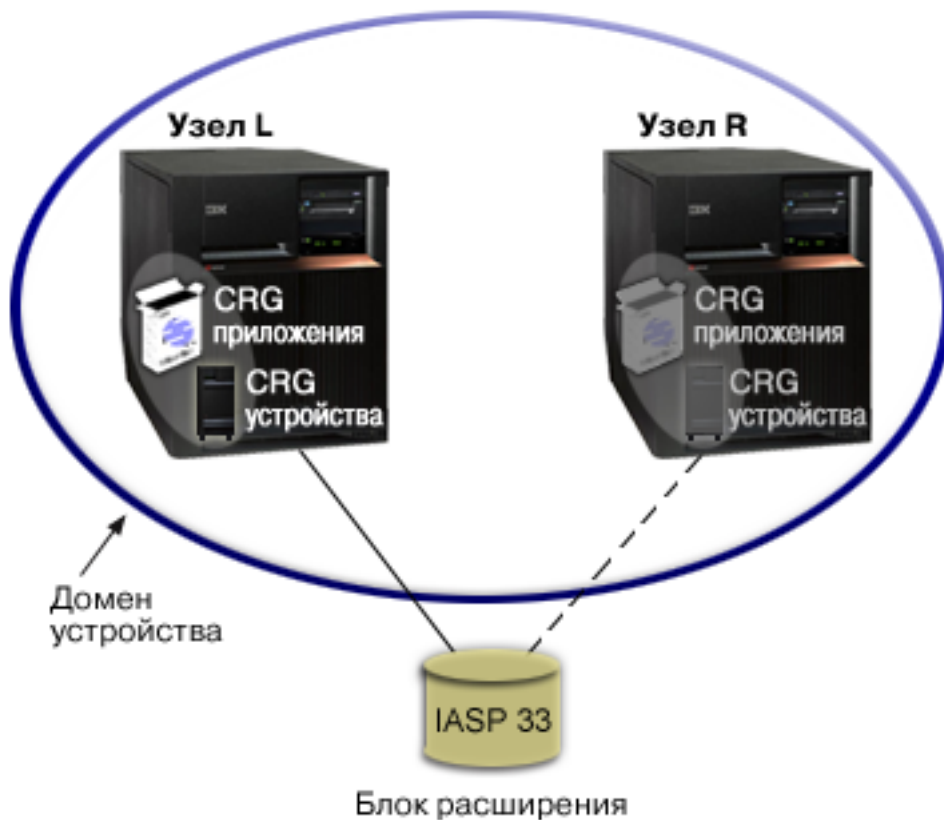
CRG устойчивых устройств во многом схожи с CRG других типов. Одно из отличий (список переносимых устройств) уже упоминалось выше. Другое отличие заключается в том, что для устройств CRG не обязательно применять программу выхода. Программы выхода требуются только в случае, если при переносе устройств необходимы особые действия по настройке среды или обработке данных. Дополнительные сведения об этом типе CRG приведены в описании the API Создать группу ресурсов кластера (QcstCreateClusterResourceGroup) .

Домены устройств

Домен устройств - множество узлов кластера, совместно использующих ресурсы устройств. Узлы, входящие в домен устройств, совместно выполняют операции переноса точки доступа к набору устойчивых устройств.

Для управления доменами устройств применяются специализированные интерфейсы, позволяющие изменять состав доменов.

Домены устройств осуществляют совместное управление глобальной информацией, необходимой для перемещения устойчивых устройств. Для того чтобы при перемещении устройств не было конфликтов, эта информация должна быть доступна всем узлам домена. Например, при работе с перемещаемыми ASP всем узлам домена должны быть известны идентификатор ASP, параметры дисков и их виртуальные адреса.



Каждый узел кластера может входить только в один домен устройств. Перед тем как добавить узел в домен восстановления CRG устройства, его нужно включить в домен устройств. Все узлы, входящие в домен восстановления CRG устройства, должны входить в один и тот же домен устройств.

Для работы с доменами устройств требуется наличие в системе Компонента 41 (i5/OS - HA Switchable Resources) с действующим ключом лицензии.

Понятия, связанные с данным

“Пример: Кластер из двух узлов с перемещаемыми дисками” на стр. 121

Для обеспечения устойчивости данных вместо репликации можно пользоваться перемещаемыми дисками. В кластере с перемещаемыми дисками данные хранятся в независимых ASP.

Задачи, связанные с данной

“Добавление узла в домен устройств” на стр. 108

Домен устройств - множество узлов кластера, совместно использующих ресурсы устройств.

“Удаление узла из домена устройств” на стр. 109

Домен устройств - множество узлов кластера, совместно использующих ресурсы устройств.

Компонент 41 (HA Switchable Resources):

Для работы с доменами устройств требуется наличие в системе Компонента 41 (i5/OS - HA Switchable Resources) с действующим ключом лицензии.

Этот компонент нужен для выполнения следующих операций в кластерной среде:

- Применение интерфейса управления кластером Навигатора iSeries.
- Применение переносимых независимых ASP.
- Распределенная по сайтам зеркальная защита между географически разнесенными системами

Задачи, связанные с данной

“Добавление узла в домен устройств” на стр. 108

Домен устройств - множество узлов кластера, совместно использующих ресурсы устройств.

“Удаление узла из домена устройств” на стр. 109

Домен устройств - множество узлов кластера, совместно использующих ресурсы устройств.

| **События кластера**

| В кластере есть множество событий, действий и служб.

| **Автоматический перенос ресурсов**

| *Автоматический перенос ресурсов* - это автоматический перенос точки доступа к ресурсу на один из резервных серверов в случае сбоя сервера, выполняющего роль главной точки доступа к ресурсу.

| Для сравнения, принудительный перенос ресурсов возникает в случае, когда пользователь вручную перемещает точку доступа к ресурсу на другой сервер. Процедуры автоматического и принудительного переноса ресурсов абсолютно идентичны, отличается только природа их возникновения.

| При автоматическом переносе точка доступа к ресурсу перемещается с узла, выполняющего функции главной точки доступа на данный момент, на узел, выполняющий функции первой резервной точки доступа. Информация о том, как выбираются узлы при переносе, приведена в разделе Домен восстановления.

| Если автоматический перенос выполняется для нескольких групп ресурсов кластера (CRG), то сначала обрабатываются CRG устройств (переносимые устройства), затем CRG данных (переносимые группы данных), и только после этого CRG приложений (переносимые приложения).

| Сведения об автоматическом переносе заносятся в очередь сообщений автоматического переноса. С помощью этой очереди можно управлять процедурой автоматического переноса CRG.

| **Понятия, связанные с данным**

| “Домен восстановления” на стр. 11

| *Домен восстановления* - это множество узлов кластера, объединенных в CRG для выполнения определенной задачи - например, для обеспечения бесперебойного доступа к ресурсу или для синхронизации.

| “Группа ресурсов кластера (CRG)” на стр. 7

| *Группа ресурсов кластера (CRG)* - это объект системы i5/OS, представляющий собой список ресурсов кластера, служащий для управления событиями в кластерной среде. Для группы ресурсов кластера должен быть задан домен восстановления и имя программы выхода CRG, вызывающейся при определенных событиях в кластере.

| “Очередь сообщений автоматического переноса ресурсов” на стр. 116

| Сведения об автоматическом переносе заносятся в очередь сообщений автоматического переноса.

| “Требования к аппаратному обеспечению в кластере” на стр. 81

| Функции узла кластера может выполнять любая система iSeries с операционной системой i5/OS версии V4R4M0 или выше.

| **Задачи, связанные с данной**

| “Принудительный перенос ресурсов” на стр. 21

| *Принудительный перенос ресурсов* выполняется в случае перемещения пользователем точки доступа к ресурсу на другой сервер.

| **Пример: Автоматический перенос ресурсов:**

| Обычно автоматический перенос ресурсов выполняется при выходе узла из строя, но этим список причин не ограничивается.

| Например, повреждение отдельного ресурса в CRG приведет к автоматическому переносу CRG, но при этом прочие CRG кластера останутся без изменения.

В следующей таблице различные типы сбоев распределены по основным категориям:

Сбой	Категория
Выход аппаратуры СЕС (например, процессора) из строя	2
Выход из строя сетевого адаптера, линии или маршрутизатора; выполнение команды ENDTCPICF для всех интерфейсов IP узла	4
Прекращение подачи питания СЕС	1
Машинный сбой программного обеспечения операционной системы	2
Выполнение команды ENDTCP(*IMMED или *CNTRLD с ограничением по времени)	1
Выполнение команды ENDSBS QSYSWRK(*IMMED или *CNTRLD)	1
Выполнение команды ENDSBS(*ALL, *IMMED или *CNTRLD)	1
Выполнение команды ENDSYS (*IMMED или *CNTRLD)	1
Выполнение команды PWRDWNSYS(*IMMED или *CNTRLD)	1
Нажатие кнопки IPL в момент, когда в системе запущена служба ресурсов кластера	1
Отмена задания QCSTCTL (*IMMED или *CNTRLD с ограничением по времени)	1
Отмена задания QCSTCRGM (*IMMED или *CNTRLD с ограничением по времени)	1
Отмена задания CRG(*IMMED или *CNTRLD с ограничением по времени)	3
Вызов API Завершить работу узла кластера	1
Вызов API Удалить запись узла кластера	1
Аварийное завершение задания CRG из-за программной ошибки	3
Выключение системы с помощью функции 8 или 3 панели управления	2
Отложенное выключение раздела с помощью функции 7	1
Сбой в программе, входящей в CRG приложений	3
<p>Категория:</p> <ol style="list-style-type: none"> 1. На узле прекращена работа службы ресурсов кластера, и узел рассматривается как вышедший из строя. Фактически сам узел может быть как работоспособным, так и нет (например, если прекращена подача питания). Для всех ресурсов кластера, для которых прекратила работу служба ресурсов кластера, будет выполнен автоматический перенос. 2. На узле прекращена работа службы ресурсов кластера, но узел по-прежнему виден как отдельный раздел кластера. Узел может быть как работоспособным, так и нет. 3. Произошел сбой отдельного CRG. Такая ситуация всегда квалифицируется как сбой. 4. Произошел сбой, но и узел, и служба ресурсов кластера по-прежнему работают, и узел виден как раздел кластера. 	

Действия, предпринимаемые при обнаружении сбоя, зависят от типа сбоя и состояния CRG. Во всех случаях вызывается программа выхода. Для процедуры автоматического переноса ресурсов может потребоваться список вышедших из строя узлов. Поэтому программа выхода, вызываемая для переноса, должна определить, какие узлы вышли из строя.

Если группа ресурсов кластера находится в состоянии *не работает*, то состояние отказавшего узла в домене восстановления CRG меняется на *не работает* или *распад*. Однако при этом не изменяются роли узлов и порядок резервных узлов в списке. Порядок резервных узлов в списке для неактивной группы ресурсов кластера можно изменить с помощью команды Запустить CRG (STRCRG) или API Запустить CRG (QestStartClusterResourceGroup). Однако API Запустить CRG нельзя применять в случае, если главный узел CRG не работает. В такой ситуации один из узлов следует сначала сделать главным с помощью команды Изменить группу ресурсов кластера (CHGCRG) command или API Изменить группу ресурсов кластера (QestChangeClusterResourceGroup), а затем снова вызвать API Запустить CRG.

Если CRG находится в состоянии *активен*, а узел, на котором произошел сбой, *не является* главной точкой доступа, то перенос заключается в изменении состояния домена восстановления CRG. Если узел, на котором возник сбой, был резервным узлом, то он перемещается в конец списка резервных узлов.

Если группа ресурсов кластера находится в состоянии *активна*, а сбойный узел является главным, то в соответствии с категорией сбоя выполняется одна из следующих процедур:

Сбой категории 1

Выполняется автоматическое перемещение ресурсов. Узел, на котором находилась главная точка доступа, переводится в состояние *не работает* для всех CRG и становится последним резервным узлом. Узел, который был первым резервным узлом, становится главным узлом. Сначала выполняется перемещение CRG устройств. Затем перемещаются все CRG данных. После этого перемещаются CRG приложений. Если при попытке переноса какого-либо CRG выяснится, что нет ни одного рабочего резервного узла, CRG будет переведен в состояние *под вопросом*.

Сбой категории 2

Выполняется автоматический перенос ресурсов, но главный узел остается без изменений. Все узлы в разделе кластера, у которых главный узел оказался в другом разделе, завершат работу с соответствующими CRG. В разделе, в котором находится главный узел, состояние таких узлов будет изменено на *распад*. Если главный узел CRG выйдет из строя, но служба ресурсов кластера ошибочно посчитает, что произошел распад кластера, то все данные и приложения, которые обслуживались этим узлом, станут недоступны, а автоматический перенос ресурсов не будет выполнен. В такой ситуации нужно либо объявить узел сбойным, либо восстановить работоспособность узла и вновь запустить на нем службу ресурсов кластера. Дополнительная информация приведена в разделе Перевод разделившихся узлов в разряд сбойных.

Сбой категории 3

Перенос выполняется только для тех CRG, которые затронуты сбоем. Хотя сбой может затронуть несколько CRG, поскольку CRG не зависят друг от друга, перенос выполняется индивидуально. Узел, на котором находилась главная точка доступа, переводится в состояние *не работает* для всех CRG и становится последним резервным узлом. Узел, который был первым резервным узлом, становится главным узлом. Если нет ни одного рабочего резервного узла, CRG переводится в состояние *под вопросом*.

Сбой категории 4

Эта категория схожа с категорией 2, однако для нее характерно то, что все узлы работоспособны, но нарушена связь между некоторыми из них. Кластер распадается на разделы, однако главные узлы по-прежнему выполняют свои функции. Такая ситуация потенциально может привести к возникновению различных неполадок. Например, если главный узел находится в одном разделе, а все резервные узлы и копии - в другом разделе, то будет прекращена репликация и таким образом потеряна защита от сбоя главного узла. В разделе, содержащем главный узел, в результате автоматического переключения ресурсов все узлы CRG домена восстановления, относящиеся к другим разделам, будут переведены в состояние *распад*. В разделе, не содержащем главный узел, состояние узлов CRG домена восстановления, относящихся к другому разделу, также будет установлено как *распад*.

Понятия, связанные с данным

“Распад кластера” на стр. 139

Большинство неполадок кластера устранить достаточно легко. Распад кластера - это одна из наиболее

серьезных неполадок. В этом разделе приведены рекомендации по предотвращению распада кластера и приведен пример процедуры восстановления распавшегося кластера.

Принудительный перенос ресурсов

Принудительный перенос ресурсов выполняется в случае перемещения пользователем точки доступа к ресурсу на другой сервер.

Принудительный перенос полезен при выполнении профилактического обслуживания систем, установке PTF и установке новых версий операционной системы. В отличие от принудительного переноса, автоматический перенос ресурсов выполняется автоматически в случае сбоя главного узла.

При принудительном переносе точка доступа к ресурсу перемещается с узла, выполняющего функции главной точки доступа на данный момент, на узел, выполняющий функции первой резервной точки доступа. Информация о том, как выбираются узлы при переносе, приведена в разделе Домен восстановления.

При выполнении принудительного переноса нескольких CRG следует уделить внимание выбору последовательности, в которой выполняется перенос. Например, если в вашей среде есть CRG приложения, зависящая от данных, связанных с CRG устройства, то перенос нужно выполнять в следующем порядке:

1. Остановить приложение на старом главном узле (для того чтобы прекратить изменение данных).
2. Переключить CRG устройства на новый главный узел.
3. Переключить CRG приложения на новый главный узел.
4. Запустить приложение на новом главном узле.

Понятия, связанные с данным

“Автоматический перенос ресурсов” на стр. 18

Автоматический перенос ресурсов - это автоматический перенос точки доступа к ресурсу на один из резервных серверов в случае сбоя сервера, выполняющего роль главной точки доступа к ресурсу.

“Домен восстановления” на стр. 11

Домен восстановления - это множество узлов кластера, объединенных в CRG для выполнения определенной задачи - например, для обеспечения бесперебойного доступа к ресурсу или для синхронизации.

Задачи, связанные с данной

“Принудительный перенос ресурсов” на стр. 107

В результате принудительного переноса ресурсов текущий главный узел становится резервным узлом домена восстановления.

Возврат узла в кластер

Возвратом называется ситуация, когда в кластер вновь входит узел, ранее входивший в кластер и вышедший из него.

Узел может возвращаться в кластер по разным причинам - например, при перезапуске служб кластера на узле. Ранее службы кластера на узле могли запускаться только с других узлов, входящих в кластер. Начиная с версии 3, узел может самостоятельно запустить службу ресурсов кластера и войти в кластер в случае, если ему удастся найти хотя бы один узел, уже входящий в кластер. Дополнительные сведения приведены в разделе Запуск узла кластера.

Предположим, кластер состоит из узлов А, В и С. На узле А возник сбой. Теперь в кластер входят узлы В и С. После того как узел А сможет продолжить работу, он сможет вернуться в кластер в случае, если на нем будут запущены службы кластера. Службы кластера могут быть запущены с любого узла кластера, в том числе с узла А. При выполнении операции возврата применяется объект CRG, то есть каждая группа ресурсов входит в кластер независимо от других.

Главная функция возврата заключается обновлении объекта CRG на всех активных узлах домена восстановления. На возвращающемся узле, как и на всех текущих активных узлах кластера, должна храниться текущая копия объекта CRG. Кроме того, на всех узлах должны храниться идентичные копии некоторых внутренних данных.

Если в домене управления кластером есть неактивные узлы, то любые изменения ресурсов этого домена распространятся эти узлы сразу же после возврата их в домен.

После сбоя узла данные CRG могут измениться в результате обращения к службам кластера на оставшихся узлах. Изменения могут быть вызваны вызовом API или сбоем еще одного узла. В простых кластерах на возвращающийся узел просто копируется CRG с одного из активных узлов кластера. Однако в некоторых случаях этого бывает недостаточно.

Задачи, связанные с данной

“Запуск узла кластера” на стр. 103

Операция запуска узла кластера заключается в запуске службы ресурсов кластера на узле. Начиная с версии 3, узел может самостоятельно запустить службу ресурсов кластера и войти в кластер в случае, если ему удастся найти хотя бы один узел, уже входящий в кластер.

“Перевод узлов после распада кластера в разряд сбойных” на стр. 141

В некоторых случаях выдается сообщение о разделе кластера, хотя в действительности просто узел вышел из строя. Это происходит в случаях, когда служба теряет связь с некоторыми узлами кластера и не может определить, вышли ли они из строя. Если это требуется, вы можете пометить узлы как сбойные вручную.

Пример: Возврат узла в кластер:

В этом разделе описывается возврат узла в кластер.

На следующей диаграмме показана процедура, выполняемая при возврате узла в кластер. Помимо действий, перечисленных на диаграмме, состояние узла в домене восстановления CRG изменяется с *не работает* на *работает*. На всех узлах домена восстановления CRG вызывается программа выхода с кодом действия Rejoin.

Таблица 3. Операция возврата

Операция возврата			
Возвращающийся узел		Узлы кластера	
Содержит копию CRG	Не содержит копию CRG	Содержат копию CRG	Не содержат копию CRG
(1)	(2)	(3)	(4)

В соответствии с диаграммой, возможны следующие ситуации:

1. 1 и 3
2. 1 и 4
3. 2 и 3
4. 2 и 4

Если на узле кластера есть копия CRG, то при возврате узла в кластер CRG копируется с активного узла на возвращающийся узел.

Вариант возврата 1

Копия объекта CRG с узла кластера передается на возвращающийся узел. В результате:

- Объект CRG обновляется на возвращающемся узле.
- Объект CRG может быть удален с возвращающегося узла. Это происходит в случае, если возвращающийся узел был удален из домена восстановления CRG за время своего отсутствия.

Вариант возврата 2

Копия объекта CRG с возвращающегося узла направляется на все узлы кластера. В результате:

- Если ни один узел кластера не входит в домен восстановления CRG, то ничего не происходит.
- На некоторых узлах кластера может быть создан объект CRG. Это можно проиллюстрировать на следующем примере:
 - Кластер состоит из узлов A, B, C и D.
 - Все узлы входят в домен восстановления CRG.
 - Пока узел A находился вне кластера, узел B был удален из домена восстановления CRG.
 - Узлы C и D выходят из строя.
 - В кластере остается только узел B, на котором нет копии CRG.
 - Узел A возвращается в кластер.
 - На узле A на этот момент находится устаревший CRG. На узле B нет CRG. CRG создается на узле B. Когда узлы C и D вернутся в кластер, на них будет обновлена информация о CRG, а узел B будет исключен из домена восстановления.

Вариант возврата 3

Копия объекта CRG с узла кластера передается на возвращающийся узел. В результате:

- Если возвращающийся узел не входит в домен восстановления CRG, ничего не происходит.
- Объект CRG может быть создан на возвращающемся узле. Это происходит в случае, если CRG был удален с узла за время его отсутствия в кластере.

Вариант возврата 4

На возвращающийся узел может быть передана внутренняя информация с какого-либо другого узла кластера, однако эта операция проходит незаметно для пользователя.

Объединение

Операция *объединения* аналогична операции возврата узла в кластер. Разница заключается в том, что объединение выполняется при восстановлении связи между частями кластера, образовавшимися в результате распада.

Предполагается, что при распаде кластера все узлы сохраняют работоспособность. Однако вследствие отказа линии связи нарушается связь между группами узлов. Впрочем, вполне возможно, что узел вышел из строя, но службе ресурсов кластера не удалось это обнаружить.

В первом случае части кластера, образовавшиеся после распада, автоматически объединяются после устранения причин распада. Это возможно благодаря тому, что части кластера после распада регулярно пытаются наладить связь с недоступными узлами, и как только это становится возможно, они вновь объединяются в кластер. Во втором случае автоматическое объединение невозможно, вместо этого нужно перезапустить службу ресурсов кластера на узле, вышедшем из строя, с другого узла кластера.

Понятия, связанные с данным

“Возврат узла в кластер” на стр. 21

Возвратом называется ситуация, когда в кластер вновь входит узел, ранее входивший в кластер и вышедший из него.

“Распад кластера” на стр. 139

Большинство неполадок кластера устранить достаточно легко. Распад кластера - это одна из наиболее серьезных неполадок. В этом разделе приведены рекомендации по предотвращению распада кластера и приведен пример процедуры восстановления распавшегося кластера.

Задачи, связанные с данной

“Запуск узла кластера” на стр. 103

Операция запуска узла кластера заключается в запуске службы ресурсов кластера на узле. Начиная с версии 3, узел может самостоятельно запустить службу ресурсов кластера и войти в кластер в случае, если ему удастся найти хотя бы один узел, уже входящий в кластер.

“Перевод узлов после распада кластера в разряд сбойных” на стр. 141

В некоторых случаях выдается сообщение о разделе кластера, хотя в действительности просто узел вышел из строя. Это происходит в случаях, когда служба теряет связь с некоторыми узлами кластера и не может определить, вышли ли они из строя. Если это требуется, вы можете пометить узлы как сбойные вручную.

Пример: Объединение частей кластера:

Процедура объединения частей кластера возможна для различных ситуаций.

Процедура объединения частей кластера зависит от того, какие части кластера объединяются:

Таблица 4. Объединение главной и дополнительной части

объединение	
главная часть	не главная часть

Таблица 5. Объединение главной и дополнительной части

объединение	
не главная часть	не главная часть

Понятие главной и не главных частей индивидуально для каждого CRG. Для каждой иерархической CRG главной будет та часть, в которую входит главный узел домена восстановления. Все остальные части кластера считаются дополнительными.

Для равноправной CRG главной будет та часть, в которую входят все узлы домена восстановления. Если узлы домена восстановления входят в несколько частей, то главной среди них не будет. Обе части будут дополнительными.

Если домен управления кластером распался на несколько частей, то каждая часть будет работать дальше как отдельная группа. Синхронизация изменений ресурсов будет выполняться в пределах каждой отдельной части. При объединении частей произойдет синхронизация изменений каждой части. В конечном итоге отслеживаемые ресурсы согласуются по всему домену. Нельзя добавлять или удалять MRE во время распада домена управления.

Таблица 6. Объединение главной и дополнительной части

процедура объединения			
главная часть		не главная часть	
содержит копию CRG	не содержит копию CRG	содержит копию CRG	не содержит копию CRG
(1)	(2)	(3)	(4)

В соответствии с диаграммой, возможны следующие ситуации:

- 1 и 3
- 1 и 4
- 2 и 3 (это сочетание невозможно, поскольку в главной части по определению должен быть активен главный узел домена восстановления, что возможно только при условии, если на нем есть копия CRG.)
- 2 и 4 (это сочетание невозможно, поскольку в главной части по определению должен быть активен главный узел домена восстановления, что возможно только при условии, если на нем есть копия CRG.)

Объединение главной части с дополнительными

Копия объекта CRG с главного узла направляется на все узлы домена восстановления из не главных частей кластера. В результате на узлах из не главных частей кластера могут быть выполнены следующие действия:

- Никаких действий не выполняется на узлах, не входящих в домен восстановления CRG.
- Если узел входит в домен восстановления, он получает копию CRG из главной части кластера.
- Если к моменту объединения узел, входивший в домен восстановления до распада, уже удален из домена восстановления, то с него удаляется объект CRG.
- Если на узле не существовал объект CRG, хотя узел входит в домен восстановления, на нем создается объект CRG. Поскольку этот узел не может быть главным в домене восстановления на данный момент, он получает копию CRG с главного узла.

Таблица 7. Объединение дополнительных частей

процедура объединения			
не главная часть		не главная часть	
содержит копию CRG	не содержит копию CRG	содержит копию CRG	не содержит копию CRG
(1)	(2)	(3)	(4)

В соответствии с диаграммой, возможны следующие ситуации:

1. 1 и 3
2. 1 и 4
3. 2 и 3
4. 2 и 4

Объединение дополнительных частей кластера, вариант 1

В иерархической группе сначала выбирается узел, на котором находится наиболее свежая копия CRG. Этот узел передает копию CRG всем узлам из домена восстановления, расположенным в другой части кластера. Если одинаковая (последняя из известных) копия CRG будет обнаружена на нескольких узлах, то приоритет будет отдан тому узлу, который стоит выше в списке домена восстановления.

При объединении двух дополнительных частей равноправной группы CRG с активным состоянием будет скопирована на все остальные узлы других частей. Если состояние равноправной CRG в обеих частях одинаково, то копироваться на все остальные узлы будет та часть, в которую входит первый узел из списка домена восстановления.

В результате на тех узлах, которым передавалась копия CRG, могут быть выполнены следующие действия (как для равноправной, так и для иерархической CRG):

- Никаких действий не выполняется на узлах, не входящих в домен восстановления CRG.
- Если узел входит в домен восстановления, то на нем обновляется CRG.
- Если согласно последнему состоянию CRG узел не входит в домен восстановления, хотя ранее входил в него, то объект CRG будет удален с узла.

Объединение дополнительных частей кластера, варианты 2 и 3

Поскольку в одной из частей кластера есть копия объекта CRG, она рассылается всем узлам другой части кластера. Объект CRG создается на тех узлах другой части кластера, которые входят в домен восстановления.

Объединение дополнительных частей кластера, вариант 4

Для проверки целостности информации о кластере выполняется обмен внутренними данными.

Главная часть кластера в дальнейшем может распасться на главную и не главную часть. При сбое главного узла домена восстановления служба ресурсов кластера (CRS) зафиксирует сбой узла. В этом случае главная часть кластера станет дополнительной. То же самое происходит при завершении работы главного узла с помощью API. Завершить работу узла кластера. Не главная часть может стать главной частью кластера в случае, если в ней появится главный узел домена CRG: он может либо вернуться в эту часть кластера, либо появиться в ней в результате объединения.

В ходе процедуры объединения на всех узлах домена восстановления CRG вызывается программа выхода, независимо от того, в какой части кластера находится узел. Программа выхода вызывается с тем же кодом действия, что применяется при возврате узла в кластер. В результате объединения частей кластера роли узлов не меняются, но состояние узлов в домене восстановления CRG изменяется с *распад* на *активен*. После объединения всех частей кластера флаг распада снимается, и как следствие отменяются все ограничения на применение API для работы с CRG.

Репликация

Репликацией называется процесс непрерывного копирования объектов в реальном времени. Объекты копируются с одного узла кластера на заранее выбранные другие узлы.

Репликация позволяет хранить несколько идентичных экземпляров объектов на разных узлах. Изменения, внесенные в объект на одном из узлов, автоматически реплицируются на другие узлы.

Понятия, связанные с данным

“Устойчивые данные” на стр. 15

Устойчивые данные - это данные, хранящиеся в нескольких экземплярах на разных узлах кластера.

“Планирование логической репликации” на стр. 89

Логической репликацией называется поддержка нескольких экземпляров данных. Данные копируются с главного узла домена восстановления на резервные узлы. Если главный узел выйдет из строя, точка доступа будет перенесена на один из резервных узлов.

Контроль пульса

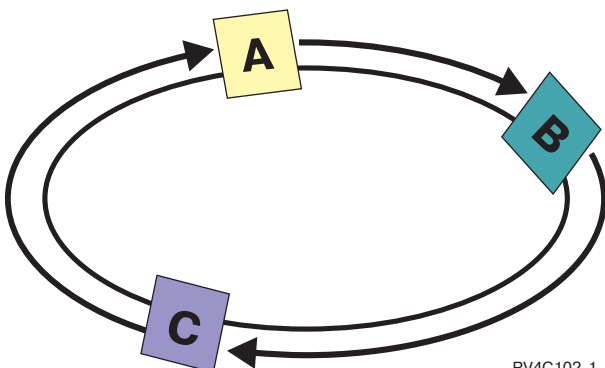
Контроль пульса - это функция службы ресурсов кластера, проверяющая работоспособность узлов путем отправки регулярных сигналов с каждого узла кластера всем остальным узлам кластера.

Если пульс от какого-либо узла прерывается, служба ресурсов кластера принимает необходимые меры.

Следующий пример иллюстрирует работу с пульсом:

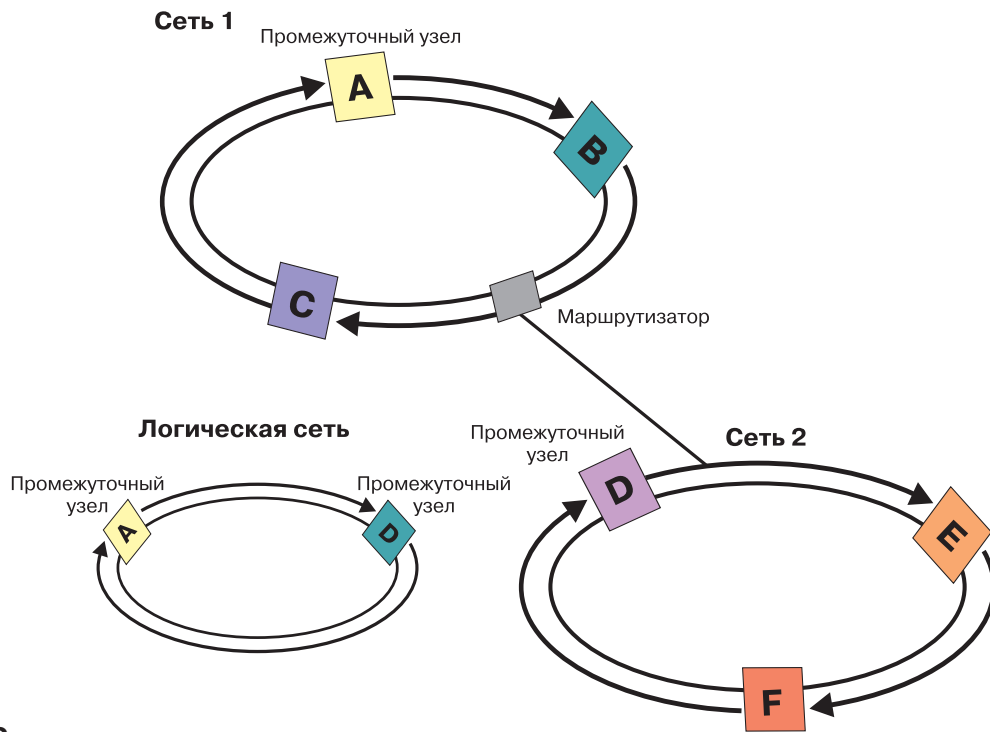
Пример 1

Сеть 1



RV4C102-1

По умолчанию каждый узел кластера раз в три секунды передает пульс соседу сверху. Например, если в сети 1 находятся узлы А, В и С, то узел А отправляет пульс узлу В, узел В отправляет пульс узлу С, а узел С - узлу А. Узел А будет ждать подтверждения пульса от узла В и собственно пульса от узла С. Таким образом сведения о пульсе идут двумя кольцами в противоположных направлениях. Каждый узел передает пульс независимо от того, получает ли он пульс от других узлов. Если, к примеру, узел С не передаст пульс четыре раза подряд (при стандартных параметрах), то будет считаться, что у этого узла пульс пропал.



Пример 2

Для иллюстрации работы маршрутизаторов добавим к предыдущему примеру еще одну сеть. Сеть 2 содержит узлы D, E и F. Между сетями 1 и 2 установлен маршрутизатор. Функции маршрутизатора может выполнять отдельный сервер iSeries или специализированный аппаратный маршрутизатор. В каждой локальной сети выделяется транзитный узел. Транзитным узлом становится узел с минимальным идентификатором в сети. В сети 1 транзитным будет узел А, в сети 2 - узел D. После этого создается логическая сеть, состоящая из узлов А и D. Теперь узлы в обеих сетях могут контролировать пульс друг друга и обмениваться сведениями о сбоях.

Понятия, связанные с данным

“Управление кластерами” на стр. 101

В этом разделе приведены инструкции по выполнению ряда операций над кластером.

“Производительность кластера” на стр. 113

Изменение кластера может повлиять на его производительность.

Задачи, связанные с данной

“Контроль состояния кластера” на стр. 112

Служба ресурсов кластера осуществляет базовый контроль состояния кластера и его компонентов с помощью функции надежных сообщений и контроля пульса.

Надежные сообщения

Функция *надежных сообщений* службы ресурсов кластера отвечает за то, чтобы на всех узлах кластера была актуальная информация о состоянии ресурсов кластера.

Реализация надежных сообщений основана на стандартных элементах - ограниченном времени ожидания подтверждения и ограниченном числе повторных попыток. Эти значения первоначально выбраны таким

образом, чтобы подходить для большинства стандартных задач. Однако их можно изменить с помощью интерфейса Изменить параметры служб ресурсов кластера. При превышении максимального числа попыток доставки сообщения принимается решение о том, что произошел сбой. В локальных сетях со стандартными значениями времени ожидания и числа попыток сбой обнаруживается в среднем за 45 секунд. В распределенных сетях на это требуется больше времени. Обычно это время колеблется от 4 минут до 15 секунд.

Понятия, связанные с данным

“Изменение параметров службы ресурсов кластера”

Параметры, применяемые службой ресурсов кластера по умолчанию, вполне удовлетворительны в большинстве рабочих сред. Однако их можно изменить.

“Управление кластерами” на стр. 101

В этом разделе приведены инструкции по выполнению ряда операций над кластером.

Задачи, связанные с данной

“Контроль состояния кластера” на стр. 112

Служба ресурсов кластера осуществляет базовый контроль состояния кластера и его компонентов с помощью функции надежных сообщений и контроля пульса.

Изменение параметров службы ресурсов кластера

Параметры, применяемые службой ресурсов кластера по умолчанию, вполне удовлетворительны в большинстве рабочих сред. Однако их можно изменить.

Параметры можно изменять следующими способами:

- Задать общий уровень производительности, наиболее соответствующий вашей среде
- Задать конкретные значения для каждого параметра для более тонкой настройки.

В первом случае у вас есть возможность выбрать один из трех предопределенных наборов значений параметров. Обычный набор, применяемый по умолчанию, подробно описан в разделе Контроль пульса.

Вторым способом изменения параметров следует пользоваться только после консультации со специалистом.

Подробные инструкции по применению обоих способов изменения параметров приведены в описании API Изменить службу ресурсов кластера (QcstChgClusterResourceServices).

Понятия, связанные с данным

“Контроль пульса” на стр. 26

Контроль пульса - это функция службы ресурсов кластера, проверяющая работоспособность узлов путем отправки регулярных сигналов с каждого узла кластера всем остальным узлам кластера.

Части кластера

Часть кластера - это множество активных узлов кластера, образовавшееся в результате сбоя связи. Серверы в отделившейся части по-прежнему поддерживают связь между собой.

Кластер распадается на части в случаях, когда связь между узлами прерывается, но при этом не удается подтвердить выход потерянных узлов из строя. Если служба ресурсов кластера обнаружит, что кластер распался на части, то будет ограничен набор действий, которые можно выполнять над узлами кластера. Это ограничение вводится для того, чтобы сохранить возможность объединения частей кластера после устранения причин его распада.

При распаде кластера некоторые операции CRG не выполняются. Дополнительные сведения об ограничениях операций для разных типов частей приведены в разделе API группы ресурсов кластера.

Если домен управления кластером распался, то синхронизация изменений будет продолжаться, но уже среди активных узлов в каждой части. Когда узлы снова объединяются, домен управления кластером собирает все изменения от каждой части и распространяет их, синхронизируя ресурсы по всему домену.

Понятия, связанные с данным

“Предотвращение распада кластера” на стр. 85

Возникновения наиболее часто встречающихся предпосылок к распаду кластера можно избежать путем создания избыточных средств связи между узлами кластера.

“Распад кластера” на стр. 139

Большинство неполадок кластера устранить достаточно легко. Распад кластера - это одна из наиболее серьезных неполадок. В этом разделе приведены рекомендации по предотвращению распада кластера и приведен пример процедуры восстановления распавшегося кластера.

“Требования к аппаратному обеспечению в кластере” на стр. 81

Функции узла кластера может выполнять любая система iSeries с операционной системой i5/OS версии V4R4M0 или выше.

Кластерные приложения

Устойчивость приложения - один из краеугольных камней кластерной среды. Если вы планируете разрабатывать и применять в кластере приложения с высокой готовностью, то следует помнить, что для таких приложений существуют свои требования к готовности.

Устойчивые приложения можно перезапускать на разных узлах кластера без изменения конфигурации клиентов. При сбое или переносе приложения на другой узел сохраняются его данные. Для пользователя переключение приложения с основного на резервный узел происходит практически незаметно. С точки зрения пользователя совершенно безразлично, где именно выполняется приложение.

Для обеспечения устойчивости приложений в кластере необходимо, чтобы все приложения соответствовали ряду требований. Часть этих требований нужна для возможности переключения приложений - основной операции, гарантирующей постоянную готовность приложений для пользователей. Дополнительные сведения об этих требованиях приведены в разделе Высокая готовность и кластеры. Эти требования можно выполнить различными способами:

1. Приобрести приложение с поддержкой кластеров

Продукты с поддержкой кластеров отвечают требованиям высокой готовности.

2. Разработать или адаптировать приложение с учетом требований высокой готовности

Независимые разработчики программного обеспечения могут самостоятельно адаптировать приложения для применения их в кластерной среде iSeries в качестве перемещаемых приложений.

После создания устойчивого приложения его работой должен управлять кластер.

Понятия, связанные с данным

“Устойчивые приложения” на стр. 14

Устойчивые приложения - это приложения, которые можно перезапустить на другом узле кластера без дополнительной настройки клиентов.

Архитектура i5/OS для приложений с поддержкой кластеров

В приложениях высокой готовности, сохраняющих устойчивость практически при любом сбое, предусматривается дополнительное значение конечного пользователя.

В системе i5/OS применяется архитектура устойчивых приложений, поддерживающая различные уровни приложений высокой готовности. На самом высоком уровне этого диапазона в приложения интегрируются функции с характеристиками высокой готовности, позволяющие автоматизировать среду высокой готовности и управляемые с помощью утилит управления кластерами.

Данные приложения обладают следующими характеристиками:

- Приложение может быть автоматически перенесено на резервный узел кластера в случае выхода главного узла из строя.
- Приложение создает устойчивую среду в соответствии с формальными требованиями к устойчивости и поддерживает автоматическую настройку и запуск посредством службы ресурсов кластера.

- Устойчивость приложения обеспечивается программой выхода CRG приложений, которая обрабатывает события кластера с помощью функций службы ресурсов кластера i5/OS.
- В приложении предусмотрена функция перезапуска, перенаправляющая пользователя в главное меню приложения или его аналог.

Приложения с более ограниченными возможностями по доступности и перезапуску, обладают следующими характеристиками:

- Устойчивость приложения повышается за счет максимально тщательной обработки событий кластера с помощью программы выхода.
- Приложение предоставляет расширенные возможности по перезапуску. Для приложений, ориентированных на сервер, это означает, что пользователь после перезапуска должен оказаться на границе последней выполненной транзакции управления фиксацией или на последней установленной контрольной точке. Для приложений, ориентированных на клиентские системы, это означает, что перезапуск приложения должен произойти полностью прозрачно для пользователя, при этом прерывание работы приложения будет минимальным.

Понятия, связанные с данным

Сайт iSeries, посвященный кластерам и обеспечению высокой готовности

Разработка кластерных приложений

Кластерные приложения, для которых гарантирована устойчивость при выходе системы из строя, иногда называют приложениями с высокой готовностью.

Различают разные уровни готовности:

1. Если в приложении возникает ошибка, приложение перезапускает себя на том же узле кластера и устраняет возможные причины ошибок (например, ошибки в управляющих данных). С точки зрения пользователя это будет выглядеть так, как будто приложение запущено в первый раз.
2. В приложении предусмотрена ограниченная поддержка возобновления работы с контрольных точек. С точки зрения пользователя это выглядит так, как будто приложение немного вернулось назад от точки сбоя.
3. При выходе системы из строя приложение перезапускается на резервном сервере. С точки зрения пользователя это будет выглядеть так, как будто приложение запущено в первый раз.
4. При выходе системы из строя приложение перезапускается на резервном сервере и выполняет ограниченные действия по обработке контрольных точек и возобновлению работы. С точки зрения пользователя это выглядит так, как будто приложение немного вернулось назад от точки сбоя.
5. При выходе системы из строя на резервный узел переносится не только приложение, но и его данные. С точки зрения пользователя это будет выглядеть так, как будто приложение запущено в первый раз.
6. При выходе системы из строя на резервный узел переносится не только приложение, но и его данные. Помимо этого, приложение ограничено поддерживает возобновления работы с контрольных точек. С точки зрения пользователя это выглядит так, как будто приложение немного вернулось назад от точки сбоя.

Примечание: В вариантах 1-4 ответственность за сохранность данных ложится на пользователя.

Обеспечение устойчивости приложений:

Обеспечение устойчивости приложений.

Устойчивым называется приложение, отвечающее следующим требованиям:

- Приложение можно перезапустить на данном или любом другом узле
- Приложение доступно клиенту по IP-адресу
- Если в приложении предусмотрены состояния, доступны сведения о состоянии
- Данные приложения доступны после восстановления приложения в случае сбоя

| Устойчивость приложения в кластерной среде в основном зависит от следующих главных составляющих:

Само приложение

Как сильно может повлиять сбой системы на работу приложения, и может ли приложение самостоятельно перезапуститься?

Эту задачу можно решить с помощью кластерных технологий.

Связанные данные

Может ли сбой системы привести к потере данных?

Сохранность данных в случае сбоя можно обеспечить с помощью продукта копирования для кластеров делового партнера IBM. Кроме того, можно хранить данные в переносимом пуле ASP (переносимом независимом ASP).

Управление возможностями и администрирование

Насколько просто создать среду, в которой будет обеспечена высокая готовность данных и приложения?

| Это можно сделать с помощью продуктов управления кластерами, работающих с кластерными API,
| от сторонних производителей.

Перезапуск кластерных приложений:

Для перезапуска приложения требуется, чтобы на момент начала автоматического или принудительного переноса приложению было известно его состояние.

Состав сведений о состоянии зависит от конкретного приложения, поэтому приложение самостоятельно устанавливает перечень того, что ему нужно. В отсутствие информации о состоянии приложение можно перезапустить с РС. Однако после этого придется вручную задать необходимое состояние приложения.

Сведения о состоянии приложения можно сохранять и передавать в резервную систему разными способами. При выборе подходящего способа следует учитывать особенности конкретного приложения.

- Приложение может передавать полные сведения о состоянии в запрашивающую систему клиента. При автоматическом или принудительном переносе ресурсов на новом сервере восстанавливается последнее известное состояние приложения. Для реализации этого метода можно воспользоваться API Разослать информацию или API группы Кластерная хэш-таблица.
- Приложение может осуществлять репликацию информации о состоянии (например, сведения о заданиях и прочих управляющих структурах) в режиме реального времени. При каждом изменении сведений приложение рассылает информацию об изменении в резервные системы.
- Приложение может хранить данные о состоянии в области данных программы выхода соответствующего CRG. Этот метод предполагает небольшой объем сохраняемой информации. Для реализации метода можно воспользоваться API Изменить группу ресурсов кластера (QcstChangeClusterResourceGroup).
- Приложение может хранить информацию о состоянии в объекте данных, для которого выполняется репликация.
- Приложение может хранить информацию о состоянии в объекте данных, содержащемся в переносимом независимом IASP и хранящем данные приложения.
- Приложение может хранить информацию о состоянии системы клиента.
- Приложение может не нуждаться в информации о состоянии.

Примечание: Правильное применение контрольных точек или их аналогов в приложении значительно сокращает потенциальный объем данных о состоянии, которые нужно хранить. Это связано с тем, что информацию о состоянии требуется сохранять только в ограниченном числе контрольных точек. Тогда при перезапуске приложения можно без труда выполнить откат на последнюю контрольную точку - подобно откату в службах фиксации баз данных.

Вызов программы выхода для группы ресурсов кластера:

Программа выхода группы ресурсов кластера вызывается при выполнении ряда операций в среде кластера.

Эта программа обеспечивает необходимую устойчивость для ресурсов кластера. Программу выхода не обязательно применять для CRG устойчивых устройств, но она нужна для других типов CRG. Программа выхода группы ресурсов кластера, если она применяется, вызывается при возникновении событий на уровне кластера, в том числе при выполнении следующих операций:

- Узел неожиданно выходит из кластера.
- Узел выходит из кластера в результате выполнения API Завершить работу узла кластера (QcstEndClusterNode) или API Удалить запись узла кластера (QcstRemoveClusterNodeEntry).
- Кластер удален с помощью API Удалить кластер (QcstDeleteCluster).
- Узел активируется с помощью API Запустить узел кластера (QcstStartClusterNode).
- Восстановлена связь с узлом, состоящим из нескольких разделов.

Данная программа выхода выполняет следующие действия:

- Выполняется в указанной группе активации или в группе активации вызывающего процесса (*CALLER).
- Игнорирует опцию перезапуска, если в программе выхода возникает непредвиденная исключительная ситуация, либо работа программы выхода отменяется.
- Предоставляет описатель для обеспечения возможности завершения работы программы выхода извне.

При выполнении API группы ресурсов кластера программа выхода запускается в отдельном задании под управлением пользовательского профайла, указанного в API Создать CRG (QcstCreateClusterResourceGroup). API самостоятельно создает отдельное задание при запуске программы выхода. Если программа выхода для CRG данных завершится с ошибкой, то она еще раз вызывается на всех активных узлах домена восстановления с кодом действия Undo. В результате отменяются результаты всех незавершенных операций и восстанавливается исходное состояние группы ресурсов кластера.

Если возникнет ошибка в программе выхода CRG приложения, и при этом CRG будет активна, то служба ресурсов кластера попытается перезапустить приложение. Программа выхода CRG будет вызвана с кодом действия Restart. Если не удастся перезапустить приложение за отведенное число попыток, то программа выхода CRG будет вызвана с кодом действия Failover. Счетчик попыток обнуляется при вызове программы выхода с кодом действия Start, то есть при запуске CRG, сбое или переходе на другой узел.

При запуске группы ресурсов кластера программа выхода CRG, вызванная на главном узле, возвращает управление службе ресурсов кластера только после завершения приложения или возникновения ошибки. Если во время работы CRG приложения службе ресурсов кластера потребуется сообщить программе выхода CRG приложения о каком-либо событии, то будет запущен еще один экземпляр программы выхода в отдельном задании. Если код действия отличен от Start и Restart, то программа выхода возвращает управление сразу.

При вызове в программу выхода CRG передаются параметры, идентифицирующие кластер, текущее и ожидаемое состояние ресурсов кластера.

Полное описание программы выхода группы ресурсов кластера, включая параметры, передаваемые для каждого кода действия, приведено в разделе Программа выхода группы ресурсов кластера документации по API кластера. В библиотеке QUSRTOOL предусмотрен пример программы выхода, который можно использовать в качестве шаблона для создания собственных программ. См. элемент TCSTAPPEXT файла QATTSYSC.

CRG приложения

CRG приложения предназначена для обеспечения устойчивости приложений.

Управление устойчивыми IP-адресами CRG приложений:

- | Управление устойчивыми IP-адресами CRG приложений с помощью службы ресурсов кластера. Кроме того,
- | предусмотрена возможность управления вручную.

Предусмотрено два способа управления IP-адресами CRG приложений. Простой способ (применяемый по умолчанию) заключается в том, что все управление берет на себя служба ресурсов кластера. В этом случае служба ресурсов кластера самостоятельно создает устойчивый IP-адрес для CRG приложений на всех узлах домена восстановления, включая узлы, которые добавляются в домен восстановления позднее. Этот способ предполагает, что устойчивый IP-адрес для CRG приложений не задан ни на одном узле домена восстановления.

Второй способ заключается в том, чтобы управлять устойчивыми IP-адресами вручную. В этом случае служба ресурсов кластера не принимает никакого участия в настройке устойчивых IP-адресов, и эта обязанность возлагается на пользователя. Устойчивый IP-адрес должен быть определен на всех узлах домена восстановления (за исключением узлов-копий) на момент включения CRG. Этот адрес также должен быть определен на всех узлах, добавляемых в домен восстановления CRG, на момент их добавления в домен.

Несколько подсетей

Теоретически применение устойчивого IP-адреса возможно даже в случае, если узлы домена восстановления распределены по нескольким подсетям. Дополнительная информация об этом приведена в разделе **Принудительный перенос приложений**.

Понятия, связанные с данным

“Пример: Действия при автоматическом переносе CRG приложений” на стр. 35

Обратитесь к примеру сценария автоматического переноса ресурсов.

“Создание CRG приложения с активным устойчивым IP-адресом” на стр. 106

При создании CRG приложения можно разрешить использование активного устойчивого IP-адреса. Однако это допустимо только в случае, если этот IP-адрес настроен пользователем.

Перенос приложений за пределы локальной подсети:

В общем случае предполагается, что все узлы кластера, входящие в домен восстановления конкретной CRG, находятся в одной подсети.

Для реализации виртуальных IP-адресов в пределах одной подсети применяются стандартные средства протокола ARP. Однако если это необходимо, домен восстановления можно разнести по разным подсетям.

Для этого необходимо, чтобы на всех узлах домена восстановления, во всех подсетях домена восстановления и на всех маршрутизаторах, установленных между ними, поддерживался усовершенствованный протокол маршрутизации RIP. Дополнительная информация приведена в разделе “Принудительный перенос приложений”.

Принудительный перенос приложений:

- | Служба ресурсов кластера позволяет при настройке CRG приложений вручную задавать IP-адреса резервных
- | узлов.

Процедура настройки параметров переноса вручную приведена ниже. **Данные действия нужно выполнить на всех узлах домена восстановления и на прочих узлах кластера, которые будут входить в домен восстановления для данного CRG приложения.**

1. Выберите устойчивый IP-адрес для CRG приложения.
 - Во избежание конфликтов этот адрес не должен отличаться от всех других IP-адресов, используемых узлами кластера и маршрутизаторами. Например, если будет выбран адрес 19.19.19.19, то убедитесь, что маршрутов к сетям 19.0.0.0 (19.19.0.0 и т.п.) нет в таблицах маршрутов.
 - Создайте интерфейс для резервного узла (например, 19.19.19.19) с описанием линии *VIRTUALIP, маской подсети 255.255.255.255 (маршрут к хосту), MTU 1500 (допустимы значения в диапазоне

576-16388) и параметром AUTOSTART(*NO). Этот адрес (например, 19.19.19.19) должен быть связан с линией *VIRTUALIP на момент выполнения следующего шага. Однако этот адрес не обязательно должен быть активным.

2. Свяжите устойчивый IP-адрес с одним или обоими IP-адресами, указанными для служб кластера при создании кластера или добавлении узла в кластер.
 - В нашем примере это означает, что адрес 19.19.19.19 нужно сделать связанным локальным интерфейсом для IP-адреса узла кластера на шине Ethernet для того, чтобы этот адрес мог применяться кластером. Эту операцию нужно выполнить на всех узлах кластера.

Примечание: В команде CFGTCP IP-адреса должны указываться полностью.

3. Создайте кластер и все необходимые CRG. Укажите для CRG приложений значение QcstUserCfgsTakeoverIpAddr в поле 'устойчивый IP-адрес'. Не запускайте пока CRG приложений.
4. Выберите в меню CFGTCP опцию 20 (Настроить приложения TCP/IP), затем опцию 2 (Настроить RouteD), затем 1 (Изменить атрибуты RouteD) и проверьте значение параметра Рассылать. Если оно отлично от *YES, укажите значение *YES и перезапустите демон ROUTED (RIP или RIP-2) на всех узлах кластера.
 - Если демон ROUTED работает, то опция 3 команды NETSTAT должна показать, что он подключен к локальному порту. На всех узлах домена восстановления CRG должен работать и рассылать информацию о маршрутах (параметр Рассылать = *YES) демон ROUTED.
5. Убедитесь в том, что все маршрутизаторы, установленные между узлами домена восстановления, получают и пересылают маршруты по протоколу RIP.
 - Этот режим далеко не всегда включен на маршрутизаторах. Производители используют разную терминологию, но так или иначе вам нужно включить режим пересылки данных о маршрутах к хостам и о динамических хостах по протоколу RIP.
 - Это относится как к интерфейсам маршрутизаторов, подключенным к серверам iSeries, так и к интерфейсам, связывающим маршрутизаторы.

Примечание: В такой ситуации не следует пользоваться сервером iSeries. Вместо этого лучше приобретите отдельный маршрутизатор (IBM или сторонних производителей), предназначенный именно для эти целей. Средства маршрутизации iSeries не поддерживают данную функцию.

6. Теперь можно вручную активизировать устойчивый IP-адрес на одном из узлов кластера и подождать около 5 минут для того, чтобы все узлы домена восстановления CRG и прочие заинтересованные хосты в локальной сети получили по протоколу RIP сведения о расположении этого хоста.
 - Убедитесь в том, что рассылка информации по протоколу RIP работает, и отключите адрес резервного узла.
 - Служба ресурсов кластера автоматически включит этот адрес при запуске CRG.
7. Запустите CRG приложений.
 - Служба ресурсов кластера задействует устойчивый адрес на главном узле домена восстановления, а протокол RIP начнет рассылать информацию о расположении главного узла по домену восстановления. Примерно через 5 минут всем узлам домена восстановления будет известно расположение главного хоста. Протокол RIP сам по себе никак не связан с функцией запуска CRG.

Важные примечания:

- Если процедура, приведенная выше, будет выполнена не для всех узлов домена восстановления CRG приложений, то попытка переноса ресурсов приведет к зависанию кластера.
- Хотя в данном примере не рассматривается возможность переноса данных на узлы копии, настоятельно рекомендуется разработать способ изменения роли узлов-копий на резервные узлы при необходимости. Тогда в критической ситуации узлы-копии смогут взять на себя роль главной точки доступа к ресурсам.

- Если применяется несколько виртуальных IP-адресов, то каждый из них должен быть связан с отдельной CRG приложений и отдельным неvirtуальным IP-адресом. Это может быть другой логический IP-адрес на том же физическом адаптере или на другом. Обязательно примите меры к предотвращению возникновения неоднозначности в таблицах маршрутов. Проще всего поступить так:
 - Добавьте маршрут *DFTRROUTE в таблицу маршрутизации для каждого виртуального IP-адреса.
 - Это можно сделать с помощью опции 2 команды CFGTSP.
 - Укажите все параметры, включая ближайший транзитный участок, и укажите в качестве предпочитаемой привязки IP-адрес локальной системы, связанный с данным виртуальным IP-адресом.

Пример: Действия при автоматическом переносе CRG приложений:

Обратитесь к примеру сценария автоматического переноса ресурсов.

При автоматическом переносе ресурсов CRG устойчивого приложения, сбой которого произошел из-за аварийного завершения задания или превышения числа попыток, происходит следующее:

- На всех активных узлах домена восстановления CRG вызывается программа выхода CRG с кодом действия failover. Служба ресурсов кластера готовится к автоматическому переносу точки доступа к приложению на первый резервный узел.
- Служба ресурсов кластера прерывает соединение IP с главным узлом CRG. Дополнительные сведения об устойчивых IP-адресах приведена в разделе Управление IP-адресами CRG приложения.
- Служба ресурсов кластера переносит IP-адрес на первый резервный узел, который становится главным узлом.
- Служба ресурсов кластера запускает задание, которое вызывает на новом главном узле программу выхода CRG с кодом действия Start. Программа выхода перезапускает приложение.

Данный пример иллюстрирует одну из процедур автоматического переноса ресурса. Могут применяться и другие процедуры.

Пример: Программа выхода приложения:

В этом примере содержится исходный код примера программы выхода группы ресурсов кластера приложений.

Данный пример исходного кода расположен в библиотеке QUSRTOOL.

Используя эти примеры, вы автоматически соглашаетесь с условиями раздела Лицензирование и отказ от гарантий на предоставляемый код.

```

/*****/
/*                                          */
/* Библиотека:  QUSRTOOL                    */
/* Файл:       QATTSYSC                     */
/* Элемент:    TCSTAPPEXT                   */
/* Тип:       ILE C                         */
/*                                          */
/* Описание:                                     */
/* Это пример программы выхода CRG приложений, которая вызывается для */
/* обработки различных событий кластера, а также для API кластера. В  */
/* этот пример необходимо добавить дополнительные процедуры, так      */
/* как они в значительной степени зависят от уникальных действий,    */
/* выполняемых для каждого конкретного приложения.                    */
/*                                          */
/* С помощью этого примера создается оболочка, содержащая основу     */
/* для создания программы выхода CRG. Комментарии, приведенные в этом */
/* примере, описывают те вопросы, на которые необходимо              */

```

```

/* обратить внимание при создании реальной программы выхода. */
/* */
/* В этом примере обрабатываются все коды действий, */
/* допустимые для CRG приложений. */
/* */
/* Включаемый файл tcstdtaara.h также входит в состав библиотеки QUSRT00L. */
/* См. элемент TCSTDТААRА в файле QАТTSYSС. */
/* */
/* Протокол изменений: */
/* Флаг Причина Версия Дата ИД польз. Описание */
/* ----- */
/* ... D98332 v5r1m0 000509 ROCH Начальное создание */
/* $A1 P9950070 v5r2m0 010710 ROCH Исправления области данных */
/* $A2 D99055 v5r2m0 010913 ROCH Добавлен код действия CancelFailover*/
/* $A3 D98854 v5r2m0 010913 ROCH Добавлен код действия VerificationPhase*/
/* $A4 P9A10488 v5r3m0 020524 ROCH Добавлен код примера ожидания данных*/
/* Код действия CRG при переносе */
/* */
/*****

```

```

/*-----*/
/* */
/* Заголовочные файлы */
/* */
/*-----*/
#include /* Полезно при отладке */
#include /* макрос offsetof */
#include /* системная функция */
#include /* строковые функции */
#include /* константы/структуры обработки исключительных ситуаций */
#include /* различные константы кластера */
#include /* структура информации CRG */
#include "qusrtool/qatssys/tcstdtaara" /* области данных QCSTHAAPPI/QCSTHAAPPO*/
#include /* API для извлечения содержимого из области данных */
#include /* API определения типа кода ошибки */
#include /* встроенная функция mitime */
#include /* встроенная функция waittime */

```

```

/*-----*/
/* */
/* Константы */
/* */
/*-----*/
#define UnknownRole -999
#define DependCrgDataArea "QCSTHAAPPO"
#define ApplCrgDataArea "QCSTHAAPPI"
#define Nulls 0x00000000000000000000

```

```

/*-----*/
/* */
/* Приведенные ниже константы используются в функции checkDependCrgDataArea().*/
/* Первая из них определяет время ожидания перед проверкой области */
/* данных. Вторая задает максимальное время ожидания готовности области данных*/
/* при выполнении функции Запустить CRG, перед отменой запуска приложения. */
/* Третья задает максимальное время ожидания для функций Выполнить */
/* принудительный перенос ресурсов или Выполнить автоматический перенос ресурсов.*/
/* */
/*-----*/
#define WaitSecondsIncrement 30
#define MaxStartCrgWaitSeconds 0
#define MaxWaitSeconds 900

```

```

/*-----*/
/* */

```

```

/* Так как данная программа выхода позволяет обрабатывать новые */
/* коды действий, измените значение, приведенное ниже, на */
/* наибольший код обрабатываемого действия. */
/* */
/*-----*/
#define MaxAc 21

/*-----*/
/* */
/* Если данные программы выхода в CRG обладают определенной структурой, */
/* включите заголовочный файл, в котором содержится определение этой */
/* структуры и укажите в следующей строке вместо char имя этой структуры. */
/* */
/*-----*/
#define EpData char

/*-----*/
/* */
/* Укажите в следующей строке имя библиотеки, в которой содержится */
/* приложение и где будут находиться области данных QCSTHAAPPO и */
/* QCSTHAAPPI. */
/* */
/*-----*/
#define ApplLib "QGPL"

/*-----*/
/* */
/* Прототипы внешних функций. */
/* */
/*-----*/
static int getMyRole(Qcst_EXTP0100_t *, int, int);
#pragma argopt(getMyRole)
static int doAction(int, int, int, Qcst_EXTP0100_t *, EpData *);
#pragma argopt(doAction)
static int createCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int startCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int restartCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int endCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int verifyPhase(int, int, Qcst_EXTP0100_t *, EpData *);
static int deleteCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int memberIsJoining(int, int, Qcst_EXTP0100_t *, EpData *);
static int memberIsLeaving(int, int, Qcst_EXTP0100_t *, EpData *);
static int switchPrimary(int, int, Qcst_EXTP0100_t *, EpData *);
static int addNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int rmvNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int chgCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int deleteCrgWithCmd(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoPriorAction(int, int, Qcst_EXTP0100_t *, EpData *);
static int endNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int chgNodeStatus(int, int, Qcst_EXTP0100_t *, EpData *);
static int cancelFailover(int, int, Qcst_EXTP0100_t *, EpData *);
static int newActionCode(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoCreateCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoStartCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoEndCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoMemberIsJoining(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoMemberIsLeaving(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoSwitchPrimary(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoAddNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoRmvNode(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoChgCrg(int, int, Qcst_EXTP0100_t *, EpData *);
static int undoCancelFailover(int, int, Qcst_EXTP0100_t *, EpData *);
static void bldDataAreaName(char *, char *, char *);
#pragma argopt(bldDataAreaName)
static int checkDependCrgDataArea(unsigned int);
#pragma argopt(checkDependCrgDataArea)

```

```

static void setApp1CrgDataArea(char);
#pragma argopt(setApp1CrgDataArea)
static void cancelHandler(_CNL_Hndlr_Parms_T *);
static void unexpectedExceptionHandler(_INTRPT_Hndlr_Parms_T *);
static void endApplication(unsigned int, int, int, Qcst_EXTP0100_t *, EpData *);
#pragma argopt(endApplication)

/*-----*/
/*
/* Процедуры отладки
/*
/*-----*/
static void printParms(int, int, int, Qcst_EXTP0100_t *, EpData *);
static void printActionCode(unsigned int);
static void printCrgStatus(int);
static void printRcvyDomain(char *,
                           unsigned int,
                           Qcst_Rcvy_Domain_Array1_t *);
static void printStr(char *, char *, unsigned int);

/*-----*/
/*
/* Определения типов
/*
/*-----*/

/*-----*/
/*
/* Эта структура определяет данные, передаваемые программам
/* обработки исключительных ситуаций и отмены заданий. Укажите в них
/* информацию, уникальную для вашего приложения.
/*
/*-----*/
typedef struct {
    int *retCode;           /* Указатель кода возврата
    EpData *epData;        /* Данные программы выхода от CRG
    Qcst_EXTP0100_t *crgData; /* Данные CRG
    unsigned int actionCode; /* Код действия
    int role;              /* Роль этого узла в домене восстановления
    int priorRole;        /* Предыдущая роль этого узла в домене восстановления*/
} volatile HandlerDataT;

/*-----*/
/*
/* Массив указателей на функции, обрабатывающие коды действий.
/* Если программа выхода обновляется для обработок кодов новых действий,
/* добавьте имена новых функций в этот массив указателей на функции.
/*
/*-----*/
static int (*fcn[MaxAc+1]) (int role,
                           int priorRole,
                           Qcst_EXTP0100_t *crgData,
                           EpData *epData) = {
    newActionCode, /* 0 - зарезервировано */
    createCrg,    /* 1 */
    startCrg,     /* 2 */
    restartCrg,  /* 3 */
    endCrg,       /* 4 */
    verifyPhase, /* 5 - зарезервировано */
    newActionCode, /* 6 - зарезервировано */
    deleteCrg,   /* 7 */
    memberIsJoining, /* 8 */
    memberIsLeaving, /* 9 */
    switchPrimary, /* 10 */
    addNode,     /* 11 */

```

```

rmvNode,          /* 12 */
chgCrg,           /* 13 */
deleteCrgWithCmd, /* 14 */
undoPriorAction, /* 15 */
endNode,          /* 16 */
newActionCode,   /* 17 - предназначено только для CRG устройств */
newActionCode,   /* 18 - предназначено только для CRG устройств */
newActionCode,   /* 19 - предназначено только для CRG устройств */
chgNodeStatus,   /* 20 */
cancelFailover   /* 21 */
};

/*-----*/
/*
/* Массив указателей на функции для обработки предыдущих кодов
/* действий, в случае вызова совместно с кодом действия Отменить.
/* Если программа выхода обновляется для обработки действия
/* Отмена с новыми кодами действий, добавьте имена новых
/* функций в этот массив указателей на функции.
/*
/*-----*/
static int (*undoFcn[MaxAc+1]) (int role,
                                int priorRole,
                                Qcst_EXTP0100_t *crgData,
                                EpData *epData) = {
newActionCode, /* 0 - зарезервировано */
undoCreateCrg, /* 1 */
undoStartCrg, /* 2 */
newActionCode, /* 3 */
undoEndCrg,    /* 4 */
newActionCode, /* 5 - для действия с эти кодом отмена не выполняется */
newActionCode, /* 6 - зарезервировано */
newActionCode, /* 7 */
undoMemberIsJoining, /* 8 */
undoMemberIsLeaving, /* 9 */
undoSwitchPrimary, /* 10 */
undoAddNode,       /* 11 */
undoRmvNode,       /* 12 */
undoChgCrg,        /* 13 */
newActionCode,     /* 14 */
newActionCode,     /* 15 */
newActionCode,     /* 16 */
newActionCode,     /* 17 - предназначено только для CRG устройств */
newActionCode,     /* 18 - предназначено только для CRG устройств */
newActionCode,     /* 19 - предназначено только для CRG устройств */
newActionCode,     /* 20 */
undoCancelFailover /* 21 */
};

/*-----*/
/*
/* Это точка входа для программы выхода.
/*
/*-----*/
void main(int argc, char *argv[]) {

    HandlerDataT hd1Data;

/*-----*/
/*
/* Преобразует все аргументы, переданные в массив argv, в данные
/* соответствующих типов.
/*
/*-----*/

```

```

/*-----*/
int *retCode      = (int *)argv[1];
unsigned int *actionCode = (unsigned int *)argv[2];
EpData *epData    = (EpData *)argv[3];
Qcst_EXTP0100_t *crgData = (Qcst_EXTP0100_t *)argv[4];
char *formatName  = (char *)argv[5];

/*-----*/
/*
/* Проверяет, соответствует ли ожидаемому формат передаваемых данных. */
/* Если нет, то вносится изменение, при этом данную программу выхода */
/* необходимо соответствующим образом обновить. Добавляет в приложение*/
/* функцию ведения протокола ошибок. */
/*
/*
/*-----*/
if (0 != memcmp(formatName, "EXTP0100", 8))
    abort();

/*-----*/
/*
/* Укажите данные, которые будут передаваться программам обработки */
/* исключительных ситуаций и отмены. */
/*
/*
/*-----*/
hdlData.retCode      = retCode;
hdlData.epData       = epData;
hdlData.crgData      = crgData;
hdlData.actionCode   = *actionCode;
hdlData.role         = UnknownRole;
hdlData.priorRole    = UnknownRole;
_VBDY(); /* принудительно записывать измененные переменные в основное расположение */

/*-----*/
/*
/* Применять программу обработки исключительных ситуаций для всех */
/* исключительных ситуаций. */
/*
/*
/*-----*/
#pragma exception_handler(unexpectedExceptionHandler, hdlData, \
                        _C1_ALL, _C2_ALL, _CTLA_INVOKE )

/*-----*/
/*
/* Применяется обработка отмены заданий для восстановления в случае отмены этого задания. */
/*
/*
/*-----*/
#pragma cancel_handler(cancelHandler, hdlData)

/*-----*/
/*
/* Извлечение текущей и предыдущей ролей узла, на котором выполняется */
/* программа выхода. Если API кластера или событие изменяет домен */
/* восстановления (роль узла или членство), то новое смещение домена */
/* восстановления передается в Offset_Rcvy_Domain_Array, а смещение */
/* домена восстановления до изменения передается в массив */
/* Offset_Prior_Rcvy_Domain_Array. Если домен восстановления не изменен,*/
/* то к нему можно обратиться только через Offset_Rcvy_Domain_Array. */
*/
*/

```

```

/*
/*
/*-----*/
hdlData.role = getMyRole(crgData,
                        crgData->Offset_Rcvy_Domain_Array,
                        crgData->Number_Nodes_Rcvy_Domain);
if (crgData->Offset_Prior_Rcvy_Domain_Array)
    hdlData.priorRole =
        getMyRole(crgData,
crgData->Offset_Prior_Rcvy_Domain_Array,
crgData->Number_Nodes_Prior_Rcvy_Domain);
else
    hdlData.priorRole = hdlData.role;
_VBDY(); /* принудительно записывать измененные переменные в основное расположение */

/*-----*/
/*
/* Для печати информации отладки разрешите следующую строку.
/*
/*-----*/
/*
printParms(*actionCode, hdlData.role, hdlData.priorRole, crgData,
epData);
*/

/*-----*/
/*
/* Позволяет не вносить изменения на основе кода действия. Код
/* возврата задается в соответствии с результатом функции doAction().
/*
/*-----*/
*retCode = doAction(*actionCode,
                    hdlData.role,
                    hdlData.priorRole,
                    crgData,
                    epData);

/*-----*/
/*
/* Задание программы выхода завершается, если в этой точке управление
/* возвращается операционной системе.
/*
/*-----*/
return;

#pragma disable_handler /* unexpectedExceptionHandler
#pragma disable_handler /* cancelHandler
} /* конец функции main()

/*****
/*
/* Получить роль этого узла из представления домена восстановления.
/*
/* Следующие API и события кластера передают программе выхода
/* обновленный и предыдущий домен восстановления:
/* QcstAddNodeToRcvyDomain
/* QcstChangeClusterNodeEntry

```

```

/* QcstChangeClusterResourceGroup */
/* QcstEndClusterNode (завершаемый узел не получает домен восстановления) */
/* QcstInitiateSwitchOver */
/* QcstRemoveClusterNodeEntry (удаленный узел не получает домен восстановления) */
/* QcstRemoveNodeFromRcvyDomain */
/* QcstStartClusterResourceGroup (если изменяется порядок неактивных */
/* резервных узлов) */
/* сбой, который привел к автоматическому переносу ресурсов */
/* узел, который возвращается в кластер */
/* объединение разделов кластера */
/*
/* Все остальные API передают только обновленный домен восстановления.
/*
/*****
static int getMyRole(Qcst_EXTP0100_t *crgData, int offset, int count) {
Qcst_Rcvy_Domain_Array1_t *nodeData;
    unsigned int iter = 0;

/*-----*/
/*
/* В некоторых случаях операционная система не может определить ИД узла */
/* и передает значение *NONE. Например, если в узле неактивны службы */
/* ресурсов кластера и применяется команда CL DLTCRG. */
/*
/*-----*/
    if (0 == memcmp(crgData->This_Nodes_ID, QcstNone,
sizeof(Qcst_Node_Id_t)))
        return UnknownRole;

/*-----*/
/*
/* Вычисление указателя на первый элемент массива домена восстановления. */
/*
/*-----*/
    nodeData = (Qcst_Rcvy_Domain_Array1_t *)((char *)crgData + offset);
/*-----*/
/*
/* Поиск узла в массиве домена восстановления. При добавлении узла с */
/* помощью API добавления узла в домен восстановления, он не будет найден*/
/* в предыдущем домене восстановления. */
/*
/*-----*/
    while ( 0 != memcmp(crgData->This_Nodes_ID,
nodeData->Node_ID,
sizeof(Qcst_Node_Id_t))
        &&
iter < count
    ) {
nodeData++;
iter++;
    }

    if (iter < count)
return nodeData->Node_Role;
else
return UnknownRole;
} /* конец функции getMyRole() */

/*****
/*
/* Вызов необходимой функции в соответствии с кодом действия кластера. */

```



```

/* Функция doAction() отделена от функции main() для упрощения */
/* примера. Дополнительная информация о действиях кластера приведена в */
/* первой части вызываемых функций. */
/* */
/* Каждый код действия выделен в отдельную функцию только для упрощения */
/* данного примера. В конкретной программе выхода некоторые коды функций */
/* могут соответствовать одной и той же функции, т.е. различные коды */
/* действий обрабатываются одной функцией. */
/* */
/*****/
static int doAction(int actionCode,
                  int role,
                  int priorRole,
                  Qcst_EXTP0100_t *crgData,
                  EpData *epData) {

/*-----*/
/* */
/* Выполняется вызов соответствующей функции для обработки */
/* кода действия, если он известен этой программе выхода. */
/* */
/*-----*/

if (actionCode <= MaxAc )
    return (*fcn[actionCode]) (role, priorRole, crgData, epData);
else

/*-----*/
/* */
/* В новом выпуске операционной системы компания IBM определила новый */
/* код действия, обработку которого эта программа выхода еще не */
/* поддерживает. Выполняется действие по умолчанию. */
/* */
/*-----*/
    return newActionCode(role, priorRole, crgData, epData);
} /* конец функции doAction() */

/*****/
/* */
/* Код действия = QcstCrgAcInitialize */
/* */
/* Вызван API QcstCreateClusterResourceGroup. Создается новый */
/* объект группы ресурсов кластера. */
/* */
/* Замечания: */
/* - Проверьте, что прикладная программа, а также все связанные */
/* объекты расположены в как основном, так и в резервном узле. Если */
/* данные объекты отсутствуют, отправьте сообщения об ошибке или */
/* предупреждающее сообщение или возвратите код возврата сбоя. */
/* - Проверьте, что все необходимые CRG данных и устройств */
/* существуют во всех узлах домена восстановления. */
/* - Выполните процедуру настройки, необходимую для */
/* запуска приложения на основном и резервном узлах. */
/* - Если данная CRG позволяет применять API */
/* QcstDistributeInformation, то можно создать пользовательскую */
/* очередь, необходимую для этого API. */
/* */
/*****/
static int createCrg(int role,
                   int doesNotApply,
                   Qcst_EXTP0100_t *crgData,
                   EpData *epData) {

```

```

    return QcstSuccessful;
} /* конец функции createCrg() */

/*****
/*
/* Код действия = QcstCrgAcStart
/*
/* Вызван API QcstStartClusterResourceGroup API. Запускается группа
/* ресурсов кластера.
/* Вызван API QcstInitiateSwitchOver и это второй код действия,
/* передаваемый программе выхода.
/* Произошло событие автоматического переноса
/* ресурсов и это второй код действия, передаваемый программе выхода.
/*
/* Проверка активности всех зависимых CRG выполняется с помощью
/* максимального времени ожидания. Если CRG запущена с помощью API
/* QcstStartClusterResourceGroup, то это время будет малым. В случае
/* принудительного или автоматического переноса ресурсов времени потребуется */
/* больше. При выполнении принудительного или автоматического переноса
/* ресурсов CRG данных и устройств становятся доступными не сразу, поэтому */
/* время ожидания увеличивается. Ожидание не будет длительным если используется API */
/* Запустить CRG, если зависимые CRG уже запущены, если возникли ошибки,
/* в результате которых CRG были запущены неупорядоченно, и т.д.
/*
/* Замечания:
/* - Если это основной узел, то запускается приложение. Эта программа
/* выхода должна вызвать приложение, которое будет выполняться в том же
/* задании или отслеживать все задания, запущенные этой программой
/* выхода, чтобы определить, когда завершается задание приложения.
/* Самый простой способ - вызов и выполнение приложения в этом
/* задании.
/* Службы ресурсов кластера не ожидают возврата этой программы
/* выхода до тех пор, пока не будет завершено выполнение приложения.
/* - При необходимости запустите связанные подсистемы, серверные задания
/* и т.д.
/* - Убедитесь, что необходимые CRG данных активны во всех узлах домена
/* восстановления.
/*
/*****
static int startCrg(int role,
                  int doesNotApply,
                  Qcst_EXTP0100_t *crgData,
                  EpData *epData) {

    unsigned int maxWaitTime;

    /* Запускает приложение, если этот узел является основным
    if (role == QcstPrimaryNodeRole) {

/*****
/*
/* Позволяет определить, готовы ли все CRG, от
/* которых зависит данная CRG приложения. Если в результате проверки
/* получен отрицательный результат, возвращается код действия Запуск.
/* Службы ресурсов кластера изменяют состояние CRG на Неактивен.
/*
/*
/*****
    if (crgData->Cluster_Resource_Group_Status == QcstCrgStartCrgPending)
maxWaitTime = MaxStartCrgWaitSeconds;
    else
        maxWaitTime = MaxWaitSeconds;
    if (QcstSuccessful != checkDependCrgDataArea(maxWaitTime))
        return QcstSuccessful;

```

```

/*-----*/
/*
/* Перед запуском приложения обновляет область данных для
/* отображения того, что приложение выполняется.
*/
*/

/*-----*/
    setApp1CrgDataArea(App1_Running);

/*-----*/
/*
/* Здесь укажите процедуры для вызова приложения. Предполагается, что
/* управление будет возвращено только в случае завершения приложения:
/* обычный возврат из программы выхода, отмена задания или возникновение
/* необрабатываемой исключительной ситуации. Общие способы отмены задания
/* приведены в описании функции cancelHandler().
*/
*/

/*-----*/

/*-----*/
/*
/* После завершения приложения обычным образом обновляет
/* область данных для отображения того, что приложение больше не
/* выполняется.
*/
*/

/*-----*/
    setApp1CrgDataArea(App1_Ended);
}
else

/*-----*/
/*
/* Для резервного узла и узла копии отмечает в области
/* данных, что приложение не выполняется.
*/
*/

/*-----*/
    setApp1CrgDataArea(App1_Ended);

    return QcstSuccessful;
} /* конец функции startCrg()
/*****
/*
/* Код действия = QcstCrgAcRestart
/*
/* Возникла ошибка в предыдущем вызове программы выхода, вследствие чего для
/* кода возврата указано значение QcstFailWithRestart. Или же возникла
/* исключительная ситуация, которой было разрешено отфильтровать стек вызовов.
/* В любом случае, максимальное количество разрешенных перезапусков программы
/* выхода еще не достигнуто.
*/
/*
/* Этот код действия передается только программам выхода CRG
/* приложений, которые были отменены с помощью кода действия Запустить.
/*
/*****
static int restartCrg(int role,
                    int doesNotApply,
                    Qcst_EXTP0100_t *crgData,
                    EpData *epData) {

```

```

/*-----*/
/*
/* Выполняет процедуры, необходимые для перезапуска
/* приложения после сбоя, затем вызывает функцию startCrg() для
/* выполнения функций запуска.
/*
/*-----*/

return startCrg(role, doesNotApply, crgData, epData);
} /* конец функции restartCrg() */

/*****/
/*
/* Код действия = QcstCrgAcEnd
/*
/* Код действия Завершить применяется в одном следующих случаев:
/* - Вызван API QcstEndClusterResourceGroup.
/* - Кластер был разбит на разделы. Этот узел расположен во
/* вспомогательном разделе. Код действия Завершить применяется
/* независимо от, была ли группа CRG активной или неактивной. Кроме
/* того, передаются зависимые данные кода действия
/* QcstPartitionFailure.
/* - Приложение завершено. Кроме того, передаются
/* зависимые данные кода действия QcstResourceEnd. Все узлы из домена
/* восстановления получат один и тот же код действия (включая основной).
/* - Задание CRG отменено. Программа выхода на данном узле будет
/* вызвана с кодом действия Завершить работу. В качестве кода действия
/* зависимых данных будет передано значение QcstMemberFailure.
/*
/*
/*
/* Замечания:
/* - Если CRG активна, то задание, управляющее приложением, отменяется
/* и после отмены программы выхода останавливается устойчивый IP-адрес.
/*
/* - Если задания подсистемы или серверные задания были запущены в
/* результате кода действия QcstCrgAcStart, их следует завершить
/* или объединить процедуры для завершения приложения в функции
/* cancelHandler(), так как она вызывается для всех API служб
/* ресурсов кластера, которые завершают приложение в текущем
/* основном узле.
/*
/*
/*****/
static int endCrg(int role,
                 int priorRole,
                 Qcst_EXTP0100_t *crgData,
                 EpData *epData) {

/*-----*/
/*
/* Завершает приложение, если оно выполняется в этом узле.
/*
/*-----*/

endApplication(QcstCrgAcRemoveNode, role, priorRole, crgData,
epData);

return QcstSuccessful;
} /* конец функции endCrg() */

```

```

/*****
/*
/* Код действия = QcstCrgAcVerificationPhase
/*
/* Код действия этапа проверки позволяет разрешить программе выхода
/* выполнить проверку перед дальнейшей обработкой
/* запрашиваемой функции, заданной с помощью зависимых данных кода
/* действия. Если программа выхода определяет, что запрашиваемую
/* функцию обработать невозможно, она должна вернуть
/* QcstFailWithoutRestart.
/*
/*
/* Примечание: Программа выхода не может быть вызвана с помощью кода
/* действия Отменить.
/*
/*
/*****
static int verifyPhase(int role,
                      int doesNotApply,
                      Qcst_EXTP0100_t *crgData,
                      EpData *epData) {

/*-----*/
/*
/* Выполняется проверка
/*
/*-----*/
    if (crgData->Action_Code_Dependent_Data == QcstDltCrg) {
        /* проверка */
        /* если условие не выполняется */
        /* вернуть QcstFailWithoutRestart */
    }

    return QcstSuccessful;
} /* конец функции verifyPhase() */

/*****
/*
/* Код действия = QcstCrgAcDelete
/*
/* Вызван API QcstDeleteClusterResourceGroup или QcstDeleteCluster.
/* Удаляется группа ресурсов кластера. При этом службы ресурсов кластера
/* активны.
/* В случае применения API QcstDeleteCluster передаются зависимые данные
/* кода действия QcstDltCluster.
/* Если при активной CRG применялся API QcstDeleteCluster, то после
/* обработки кода действия Удалить отменяется задание программы выхода,
/* по-прежнему активное для кода действия Запустить.
/*
/*
/* Замечания:
/* - Удалите прикладные программы и объекты из узлов, в
/* которых они больше не нужны. Например, из резервных узлов.
/* Проявите осторожность при удалении объектов приложений, так как
/* некоторые сценарии могут удалить CRG и оставить объекты
/* приложений во всех узлах.
/*
/*
/*****
static int deleteCrg(int role,
                    int doesNotApply,
                    Qcst_EXTP0100_t *crgData,
                    EpData *epData) {

    return QcstSuccessful;
} /* конец функции deleteCrg()
/*****

```

```

/* */
/* Код действия = QcstCrgAcReJoin */
/* */
/* Могут возникнуть следующие ситуации: */
/* 1. Неполадка, в результате которой кластер был разбит на */
/* разделы, исправлена и два раздела объединяются в единый кластер. */
/* Передаются зависимые данные кода действия QcstMerge. */
/* 2. В узле, в котором произошел сбой или который был */
/* остановлен, снова запущены службы ресурсов кластера и он */
/* добавляется в кластер. Передаются зависимые данные кода действия */
/* QcstJoin. */
/* 3. Перезапущено задание CRG в конкретном узле, которое было отменено */
/* или завершено. Передаются зависимые данные кода действия QcstJoin. */
/* */
/* */
/* Замечания: */
/* - Если приложение копирует информацию о состоянии приложения в другие */
/* узлы и если CRG активен, то при выполнении приложения эту информацию */
/* потребуется повторно синхронизировать в добавляемых узлах. */
/* - Проверьте отсутствующие объекты приложений в добавляемых узлах. */
/* - Убедитесь, что в добавляемых узлах присутствуют необходимые */
/* CRG данных. */
/* - Если CRG приложения активен, убедитесь, что активны необходимые CRG */
/* данных. */
/* */
/*****/
static int memberIsJoining(int role,
                           int priorRole,
                           Qcst_EXTP0100_t *crgData,
                           EpData *epData) {

/*-----*/
/*
/* Если данный узел не является основным, позволяет указать в
/* области данных, что приложение не выполняется.
*/
*/

/*-----*/
    if (role != QcstPrimaryNodeRole) {
        setApp1CrgDataArea(App1_Ended);
    }

/*-----*/
/*
/* Если в кластер добавляется один узел, то для выполнения доступен
/* определенный набор действий. В случае объединения узлов кластера,
/* который был разбит на разделы, может быть доступен другой
/* набор действий.
*/
*/

/*-----*/
    if (crgData->Action_Code_Dependent_Data == QcstJoin) {
        /* Действия для добавления узла в кластер */
    }
    else {
        /* Действия для объединения разделов */
    }

    return QcstSuccessful;
} /* конец функции memberIsJoining()

/*****/
/*
/* Код действия = QcstCrgAcFailover */

```

```

/* */
/* Для этой группы ресурсов кластера в конкретном узле */
/* или нескольких узлах произошел сбой служб ресурсов кластера или */
/* их работа была завершена. Код действия Автоматический перенос */
/* ресурсов передается независимо от того, был CRG активен или нет. */
/* Автоматический перенос ресурсов может произойти в следующих */
/* случаях: */
/* */
/* - оператор отменил задание CRG на узле. Передаются зависимые данные */
/* кода действия QcstMemberFailure. */
/* - Завершена работа служб ресурсов кластера (например, */
/* завершена подсистема QSYSWRK, пока CRS был активен). Передаются */
/* зависимые данные кода действия QcstNodeFailure. */
/* - Произошел сбой приложения для CRG приложений в основном */
/* узле, в результате которого это приложение не удалось перезапустить.*/
/* CRG активен. Передаются зависимые данные кода */
/* действия QcstAppFailure. */
/* - Произошел сбой узла (например, сбой питания). Передаются */
/* зависимые данные кода действия QcstNodeFailure. */
/* - Кластер распался вследствие сбоя связи, например, сбой линии связи */
/* или локальной сети. Код действия Автоматический перенос ресурсов */
/* передается узлам домена восстановления большинства разделов. */
/* Узлам меньшинства разделов передается код действия Завершить работу. */
/* Будут переданы зависимые данные кода действия QcstPartitionFailure. */
/* - Работа узла в домене восстановления CRG завершается с помощью */
/* API QcstEndClusterNode. Этот узел получает код действия Завершить */
/* узел. Все остальные узлы домена восстановления получают код действия*/
/* Автоматический перенос ресурсов. Для этого кода действия передаются*/
/* зависимые данные кода действия QcstEndNode. */
/* - Активный узел домена восстановления для активной CRG удаляется из */
/* кластера посредством API QcstRemoveClusterNodeEntry. Передаются */
/* зависимые данные кода действия QcstRemoveNode. Если для активной */
/* CRG удаляется неактивный узел, или если сама CRG неактивна, */
/* то передается код действия Удалить узел. */
/* */
/* Программа выхода вызывается независимо от того, активен CRG */
/* или нет. Если CRG неактивен, то для программы выхода могут */
/* отсутствовать доступные действия. */
/* */
/* Если CRG активен и оставшийся элемент был основным узлом, */
/* выполните функции, необходимые для автоматического */
/* переноса ресурсов и создания нового основного узла. */
/* */
/* Поле Action_Code_Dependent_Data позволяет определить, если */
/* - сбой произошел в результате неполадки, вследствие которой кластер */
/* был разбит на разделы (изменяются все CRG, для которых в домене */
/* восстановления существовали разделенные узлы) */
/* - произошел сбой узла или в узле завершилась работа служб ресурсов */
/* кластера (изменяются все CRG, для которых в домене восстановления */
/* существовал узел, работа которого была завершена или в ней */
/* произошел сбой) */
/* - изменена только одна CRG (например, в узле было отменено одно задание*/
/* CRG или же возникла ошибка в одном приложении) */
/* */
/* */
/* */
/* Замечания: */
/* - Подготовьте новый основной узел для запуска приложения. */
/* - Сейчас не следует запускать приложение. Если в момент */
/* сбоя CRG была активна, то снова будет вызвана программа выхода с */
/* кодом действия QcstCrgAcStart. */
/* - Если CRG приложения активен, убедитесь, что активны необходимые CRG */
/* данных. */
/* */
/*****/
static int memberIsLeaving(int role,
                           int priorRole,
                           Qcst_EXTP0100_t *crgData,

```

```

        EpData *epData) {

/*-----*/
/*
/* Если CRG активен, выполняется автоматический перенос
/* ресурсов. В противном случае действия не выполняются.
/*
/*-----*/
    if (crgData->Original_Cluster_Res_Grp_Stat == QcstCrgActive) {

/*-----*/
/*
/* CRG активен. Определяется, изменилась ли роль данного узла и
/* является ли он новым основным узлом.
/*
/*-----*/

        if (priorRole != role && role == QcstPrimaryNodeRole) {

/*-----*/
/*
/* Данный узел стал основным узлом. Выполняются действия по
/* автоматическому переносу ресурсов, так эта программа
/* выхода будет снова вызвана с кодом действия Запустить.
/*
/*-----*/

/*-----*/
/*
/* Указывает в области данных этого узла, что приложение не
/* выполняется.
/*
/*-----*/
        setApplCrgDataArea(Appl_Ended);

/*-----*/
/*
/* Если приложение не выполняет действий с кодом Запустить
/* и станет активным только после активации устойчивого IP-адреса,
/* то следующие строки необходимо раскомментировать. В них определя-
/* ется, все ли CRG, от которых зависит приложение, готовы.
/* Если проверка дала отрицательный результат, то возвращается
/* сбой кода действия.
/*
/*-----*/
/*
/* if (QcstSuccessful != checkDependCrgDataArea(MaxWaitSeconds))
/* return QcstFailWithOutRestart;
/*
/*
/*-----*/
        return QcstSuccessful;
    } /* конец функции memberIsLeaving()
}

/*****
/*

```



```

/* Код действия = QcstCrgAcSwitchover */
/* */
/* Вызван API QcstInitiateSwitchOver. Первый резервный узел из домена */
/* восстановления группы ресурсов кластера становится основным узлом, а */
/* текущий основной узел становится последним резервным узлом. */
/* */
/* Замечания: */
/* - Подготовьте новый основной узел для запуска приложения. */
/* - Запускать приложение в данный момент НЕ рекомендуется. Программа выхода */
/* будет вызвана с кодом действия QcstCrgAcStart. */
/* - До вызова программы выхода на текущем основном узле отменяется */
/* задание, управляющее приложением, и завершается устойчивый IP-адрес.*/
/* */
/* - Убедитесь, что перенесены и активны необходимые CRG */
/* данных или устройств. */
/* */
/*****/
static int switchPrimary(int role,
                        int priorRole,
                        Qcst_EXTP0100_t *crgData,
                        EpData *epData) {

/*-----*/
/* */
/* Проверяется, является ли текущий узел прежним основным узлом. */
/* */
/*-----*/

if (priorRole == QcstPrimaryNodeRole) {

/*-----*/
/* */
/* Перед переносом выполните необходимые действия по очистке прежнего */
/* основного узла. Не забудьте, что задание, управлявшее программой */
/* выхода, которая запустила приложение, уже отменено. */
/* */
/* Например, можно очистить все процессы, блокирующие базу данных. */
/* Для этого необходимо вызвать программу обработки отмены приложения. */
/*-----*/
}

/*-----*/
/* */
/* Текущий узел не является прежним основным. Проверяется, является ли */
/* он новым основным узлом. */
/* */
/*-----*/

else if (role == QcstPrimaryNodeRole) {

/*-----*/
/* */
/* Выполните действия над новым основным узлом, необходимые перед */
/* запуском приложения с кодом действия QcstCrgAcStart. */
/* */
/*-----*/

/*-----*/
/* */
/* Позволяет указать в области данных этого узла, что приложение не */
/* выполняется. */
/* */
/*-----*/
}
}

```

```

    setApp1CrgDataArea(App1_Ended);

/*-----*/
/*
/* Если приложение не выполняет действий с кодом Запустить */
/* и станет активным только после активации устойчивого IP-адреса, */
/* то следующие строки необходимо раскомментировать. В них определя-*/
/* ется, все ли CRG, от которых зависит приложение, готовы. */
/* Если проверка дала отрицательный результат, то возвращается */
/* сбой кода действия. */
*/
*/

/*-----*/
/*      if (QcstSuccessful != checkDependCrgDataArea(MaxWaitSeconds)) */
/*      return QcstFailWithOutRestart; */

}
else {

/*-----*/
/*
/* Данный узел является резервным узлом или узлом-копией. */
/* Если для этих узлов предусмотрены действия, то укажите их здесь. */
/* Если нет, то этот блок else можно удалить. */
*/
*/

/*-----*/

/*-----*/
/*
/* Позволяет указать в области данных этого узла, что приложение не */
/* выполняется. */
*/
*/
/*-----*/
    setApp1CrgDataArea(App1_Ended);
}

return QcstSuccessful;
} /* конец функции switchPrimary() */

/*****
/*
/* Код действия = QcstCrgAcAddNode */
/*
/* Вызван API QcstAddNodeToRcvyDomain. В домен восстановления группы
/* ресурсов кластера добавляется новый узел.
/*
/* Замечания:
/* - В домен восстановления добавляется новый узел. См. замечания,
/* приведенные для функции createCrg().
/* - Если данная CRG позволяет применять API
/* QcstDistributeInformation, то можно создать пользовательскую
/* очередь, необходимую для этого API.
/*
*****/
static int addNode(int role,
                  int priorRole,
                  Qcst_EXTP0100_t *crgData,
                  EpData *epData) {

/*-----*/

/*
*/

```

```

/* Определяется, является ли текущий узел добавляемым узлом. */
/*                                                                 */
/*-----*/

if (0 == memcmp(&crgData->This_Nodes_ID,
                &crgData->Changing_Node_ID,
                sizeof(Qcst_Node_Id_t)))
{
/*-----*/

/*                                                                 */
/* Задает состояние области данных нового узла.                  */
/*                                                                 */
/*-----*/
    setApp1CrgDataArea(App1_Ended);

/*-----*/

/*                                                                 */
/* Создает очередь, необходимую для API распространения информации. */
/*                                                                 */
/*-----*/

if (0 == memcmp(&crgData->DI_Queue_Name,
                Nulls,
                sizeof(crgData->DI_Queue_Name)))
{
}

return QcstSuccessful;
} /* конец функции addNode() */
/*****
/*                                                                 */
/* Код действия = QcstCrgAcRemoveNode                             */
/*                                                                 */
/* Вызван API QcstRemoveNodeFromRcvyDomain или                   */
/* QcstRemoveClusterNodeEntry. Узел удаляется из домена         */
/* восстановления группы ресурсов кластера или он полностью     */
/* удаляется из кластера.                                       */
/*                                                                 */
/* Код действия получается следующими элементами:               */
/* Для API QcstRemoveClusterNodeEntry:                           */
/* - Если удаленный узел активен, а CRG неактивна, то этот      */
/* код действия получают все узлы домена восстановления, включая */
/* удаляемый узел. Узлы, которые не удаляются, получают зависимые */
/* данные кода действия QcstNodeFailure.                         */
/* - Если удаленный узел активен и активна CRG, то удаляемый    */
/* узел получает код действия Удалить узел. Все остальные узлы  */
/* домена восстановления получают код действия                  */
/* Автоматический перенос ресурсов и зависимые данные кода действия */
/* QcstNodeFailure.                                             */
/* - Если удаляемый узел неактивен в кластере, все узлы домена  */
/* восстановления получают этот код действия.                   */
/* Для API QcstRemoveNodeFromRcvyDomain:                         */
/* - Все узлы получают код действия Удалить узел независимо     */
/* от того, активен CRG или нет. Кроме того передаются зависимые */
/* данные кода действия QcstRmvRcvyDmnNode.                     */
/*                                                                 */
/* Замечания:                                                    */
/* - Вы можете очистить удаленный узел. Для необходимо удалить  */
/* ненужные объекты.                                             */

```

```

/* - Если CRG активна и узел является основным, то после вызова */
/* программы выхода отменяется задание, управляющее приложением, и */
/* завершается устойчивый IP-адрес. */
/* - Если задания подсистемы или серверные задания были запущены в */
/* результате кода действия QcstCrgAcStart, их следует завершить */
/* или объединить процедуры для завершения приложения в функции */
/* cancelHandler(), так как она вызывается для всех API служб */
/* ресурсов кластера, которые завершают приложение в текущем */
/* основном узле. */
/*
/*****
static int rmvNode(int role,
                  int priorRole,
                  Qcst_EXTP0100_t *crgData,
                  EpData *epData) {

/*-----*/
/*
/* Определяется, является ли текущий узел удаляемым узлом.
/*
/*-----*/

if (0 == memcmp(&crgData->This_Nodes_ID,
               &crgData->Changing_Node_ID,
               sizeof(Qcst_Node_Id_t)))
{
/*-----*/
/*
/* Завершает приложение, если оно выполняется в этом узле.
/*
/*-----*/
    endApplication(QcstCrgAcRemoveNode, role, priorRole, crgData,
epData);
}
return QcstSuccessful;
} /* конец функции rmvNode() */

/*****
/*
/* Код действия = QcstCrgAcChange
/*
/* Вызван API QcstChangeClusterResourceGroup. Изменяется информация или
/* атрибуты, хранящиеся в объекте группы ресурсов кластера. Обратите
/* внимание, что программа выхода вызывается не для всех изменений
/* объекта CRG. Начиная с выпуска V5R1M0, программа выхода вызывается
/* только в следующих случаях
/* - изменяется текущий домен восстановления
/* - изменяется предпочитаемый домен восстановления
/*
/* Если вносятся указанные изменения и для программы выхода
/* задается значение *NONE, то программа выхода не вызывается.
/*
/* Замечания:
/* - Информацию и процессы этой группы ресурсов домена можно изменить
/* только в том случае, если вносятся изменения в домен восстановления.
/* Обратите внимание, что если активен CRG, то основной узел нельзя
/* изменить с помощью API QcstChangeClusterResourceGroup.
/*
/*****
static int chgCrg(int role,

```

```

        int priorRole,
        Qcst_EXTP0100_t *crgData,
        EpData *epData) {

    return QcstSuccessful;
} /* конец функции chgCrg() */

/*****
*/
/* Код действия = QcstCrgAcDeleteCommand */
/* */
/* Для удаления объекта группы ресурсов кластера вызвана команда CL */
/* Удалить группу ресурсов кластера (DLTCRG). При этом вызван */
/* API QcstDeleteCluster или API QcstRemoveClusterNodeEntry. */
/* В обоих случаях службы ресурсов кластера неактивны в узле, в котором, */
/* вызвана команда или API. Таким образом, действие этой функции не */
/* распространяется на весь кластер, а ограничено узлом, в котором вызвана */
/* команда CL или API. */
/* */
/* В случае применения API QcstDeleteCluster передаются зависимые данные */
/* кода действия QcstDltCluster. */
/* */
/* См. замечания, приведенные для функции deleteCrg() */
/* */
/*****
static int deleteCrgWithCmd(int role,
                           int doesNotApply,
                           Qcst_EXTP0100_t *crgData,
                           EpData *epData) {

    return QcstSuccessful;
} /* конец функции deleteCrgWithCmd() */

/*****
*/
/* Код действия = QcstCrgEndNode */
/* */
/* Вызван API QcstEndClusterNode или отменено задание CRG. */
/* */
/* Код действия QcstCrgEndNode передается программе выхода только в том */
/* узле, который завершается или в котором было отменено задание CRG. В узле, где отменено задание */
/* службы ресурсов кластера, передаются зависимые данные кода */
/* действия QcstMemberFailure. */
/* Завершение работы службы ресурсов кластера на этом узле или завершение */
/* задания CRG приведет к автоматическому переносу ресурсов во всех */
/* остальных узлах кластера. Этим узлам будут переданы зависимые данные */
/* кода действия QcstCrgAcFailover. В случае отмены задания будут переданы */
/* зависимые данные кода QcstMemberFailure, а в случае завершения работы */
/* узла - код QcstNodeFailure. */
/* */
/* */
/* Замечания: */
/* - Если CRG активна и узел является основным, то после вызова */
/* программы выхода отменяется задание, управляющее приложением, и */
/* завершается устойчивый IP-адрес. */
/* - Если задания подсистемы или серверные задания были запущены в */
/* результате кода действия QcstCrgAcStart, их следует завершить здесь.*/
/* */
/*****
static int endNode(int role,
                  int priorRole,
                  Qcst_EXTP0100_t *crgData,
                  EpData *epData) {

/-----*/

```

```

/* */
/* Завершает приложение, если оно выполняется в этом узле. */
/* */

/*-----*/
endApplication(QcstCrgEndNode, role, priorRole, crgData, epData);

return QcstSuccessful;
} /* конец функции endNode() */

/*****/
/* */
/* Код действия = QcstCrgAcChgNodeStatus */
/* */
/* Вызван API QcstChangeClusterNodeEntry. Данный API позволяет */
/* сообщить службам ресурсов кластера о том, что этот узел не был */
/* разбит на разделы, а в нем произошел сбой. */
/* */
/* Замечания: */
/* - Перед этим была вызвана программа выхода с кодом действия */
/* QcstCrgAcEnd при активной CRG, или с кодом действия */
/* QcstCrgAcFailover при неактивной CRG, поскольку служба ресурсов */
/* кластера предполагала, что кластер распался. Теперь пользователь */
/* сообщает службе ресурсов кластера о том, что на самом деле в узле */
/* произошел сбой. Программа выхода может выполнить действия только */
/* в том случае, если до получения подтверждения узла выполнялись */
/* какие-либо действия, которые теперь необходимо отменить. */
/* */
/*****/
static int chgNodeStatus(int role,
                        int priorRole,
                        Qcst_EXTP0100_t *crgData,
                        EpData *epData) {

return QcstSuccessful;
} /* конец функции chgNodeStatus() */

/*****/
/* */
/* Код действия = QcstCrgAcCancelFailover */
/* */
/* В основном узле произошел сбой или завершена работа служб */
/* ресурсов кластера этой группы ресурсов кластера. */
/* В очередь сообщений автоматического переноса */
/* ресурсов, указанную для этого CRG, отправлено сообщение, в */
/* результате которого восстановление после сбоя было отменено. */
/* При этом состоянии CRG будет изменено на неактивное, текущий */
/* основной узел останется основным. */
/* */
/* Замечания: */
/* - Основной узел больше не участвует в операциях кластера. Неполадка, */
/* которая привела к сбою основного узла должна быть устранена для */
/* того, чтобы снова запустить CRG. */
/* */
/*****/
static int cancelFailover(int role,
                          int priorRole,
                          Qcst_EXTP0100_t *crgData,
                          EpData *epData) {

return QcstSuccessful;
} /* конец функции cancelFailover() */

/*****/

```

```

/* */
/* Код действия = программа выхода его еще не знает */
/* */
/* Программа выхода получила новый код действия. Это может произойти */
/* установки нового выпуска i5/OS в случае вызова нового API кластера */
/* или появления нового события кластера. Программа выхода еще не была */
/* обновлена для поддержки нового кода действия. */
/* */
/* Для обработки новых кодов действий можно применять две */
/* различные стратегии. Выбор правильной стратегии зависит от */
/* действий, выполняемых этой программой выхода для приложения. */
/* */
/* Первая стратегия заключается в следующем: действия не выполняются и */
/* возвращается успешный код возврата. */
/* Такой подход позволяет полностью выполнить API кластера или событие. */
/* Функция будет выполнена успешно даже в том случае, если эта программа */
/* выхода не поддерживает новый код действия. При этом возникает риск */
/* невыполнения предусмотренного действия. В таких случаях рекомендуется */
/* заносить в протокол сообщения об ошибках, чтобы впоследствии можно */
/* было определить их причины и обновить программу выхода. */
/* */
/* Другой вариант - вернуть код ошибки, например, */
/* QcstFailWithRestart. Это означает, что новый API кластера */
/* нельзя будет применить до тех пор, пока программа выхода не получит */
/* новый код действия. В таких случаях также рекомендуется заносить */
/* в протокол сообщение об ошибке, чтобы потом определить ее причину. */
/* */
/* Выбрать наиболее подходящий подход может только разработчик */
/* программы выхода. */
/* */
/*****/
static int newActionCode(int role,
                        int doesNotApply,
                        Qcst_EXTP0100_t *crgData,
                        EpData *epData) {

/*-----*/
/* */
/* Добавьте процедуры для занесения в протокол сообщения об ошибке */
/* (очередь сообщений оператора, протокол задания, протокол ошибок */
/* приложения и т.д.). Это позволит вовремя обновлять программу */
/* выхода для обработки новых кодов действий. */
/* */
/* Обратите внимание, что по умолчанию применяется описанная */
/* выше стратегия, в результате которой действия не выполняются. */
/* */
/*-----*/

return QcstSuccessful;
} /* конец функции newActionCode() */

/*****/
/* */
/* Код действия = QcstCrgAcUndo */
/* */
/* Примечание: Программу выхода нельзя вызывать с кодом действия Отменить, */
/* если до этого выполнялось действие с одним из следующих кодов: */
/* QcstCrgAcChgNodeStatus */
/* QcstCrgAcDelete */
/* QcstCrgAcDeleteCommand */
/* QcstCrgEndNode */
/* QcstCrgAcRemoveNode (Если из кластера удаляется активный узел */
/* и применяется API Удалить узел кластера. */
/* Если из кластера удаляется неактивный узел, */

```

```

/*          то действие Отменить можно вызвать с API Удалить */
/*          узел из домена восстановления и Удалить узел */
/*          кластера. */
/* QcstCrgAcRestart */
/* QcstCrgAcUndo */
/*
/* В работе API, которые могут вызывать программы выхода, предусмотрены */
/* три этапа. */
/* 1. Выполняются процедуры, которые необходимо выполнить перед */
/* тем, как вызвать программу выхода. */
/* 2. Вызывается программы выхода. */
/* 3. Выполняются процедуры, которые необходимо выполнить после */
/* вызова программы выхода. */
/*
/* Ошибки, возникшие на этапах 2 и 3, учитываются при повторном */
/* вызове программы выхода с кодом действия Отменить. При таком подходе */
/* программа выхода может откатить все действия, выполненные при первом */
/* вызове API. Кроме того, API также откатит действия, выполненные */
/* при попытке возврата состояния кластера и объектов кластера, */
/* в котором они находились до вызова API. */
/*
/* Предполагается, что в ответ на указанный код действия возвращаются */
/* следующие коды возврата, так как от этого зависит, какое действие будет */
/* выполнено. */
/*
/* QcstCrgAcInitialize: QcstSuccessful; CRG не создается. */
/* QcstCrgAcStart:      QcstSuccessful; CRG не запускается */
/* QcstCrgAcEnd:        QcstFailWithOutRestart; CRG переходит в состояние*/
/*                      Под вопросом. Необходимо */
/*                      определить причину сбоя. */
/* QcstCrgAcReJoin:    QcstFailWithOutRestart; CRG переходит в состояние*/
/*                      Под вопросом. Необходимо */
/*                      определить причину сбоя. */
/* QcstCrgAcFailover:  QcstFailWithOutRestart; CRG переходит в состояние*/
/*                      Под вопросом. Необходимо */
/*                      определить причину сбоя. */
/* QcstCrgAcSwitchover: QcstFailWithOutRestart; CRG переходит в состояние*/
/*                      Под вопросом. Необходимо */
/*                      определить причину сбоя. */
/* QcstCrgAcAddNode:   QcstSuccessful;      Узел не добавляется */
/* QcstCrgAcRemoveNode: QcstFailWithOutRestart; CRG переходит в состояние*/
/*                      Под вопросом. Необходимо */
/*                      определить причину сбоя. */
/* QcstCrgAcChange:    QcstSuccessful;      Домен восстановления не */
/*                      изменяется. */
/*
/*****
static int undoPriorAction(int role,
                          int priorRole,
                          Qcst_EXTP0100_t *crgData,
                          EpData *epData) {

/*-----*/
/*
/* Предыдущий код действия определяет действия программы выхода в случаях*/
/* сбоя, отмены и получения безуспешного кода возврата. */
/*
/*-----*/
if (crgData->Prior_Action_Code <= MaxAc )
    return (*undoFcn[crgData-&lt;Prior_Action_Code])
        (role, priorRole, crgData,
epData);
else

/*-----*/

```



```

    /*
    /* В новом выпуске операционной системы компания IBM определила новый
    /* код действия, обработку которого эта программа выхода еще не
    /* поддерживает. Выполняется действие по умолчанию.
    /*
    /*-----*/
    return newActionCode(role, priorRole, crgData, epData);
} /* конец функции undoPriorAction()

/*****/
/*
/* Код действия = QcstCrgAcUndo
/*
/* Предыдущий код действия = QcstCrgAcInitialize
/*
/* Замечания:
/* CRG не создается. Объекты, которые могли быть созданы в узлах
/* домена восстановления, должны быть удалены, так как если они
/* будут существовать, то последующая попытка их создания может
/* быть не выполнена.
/*
/*****/
static int undoCreateCrg(int role,
                        int doesNotApply,
                        Qcst_EXTP0100_t *crgData,
                        EpData *epData) {

    return QcstSuccessful;
} /* конец функции undoCreateCrg()

/*****/
/*
/* Код действия = QcstCrgAcUndo
/*
/* Предыдущий код действия = QcstCrgAcStart
/*
/* Замечания:
/* Произошел сбой службы ресурсов кластеров при завершении API Запустить
/* CRG после вызова программы выхода с кодом действия Запустить.
/*
/* На основном узле будет отменено задание программы выхода, управляющее
/* приложением. Затем программа выхода будет вызвана с кодом действия
/* Отменить.
/*
/* Все остальные узлы домена восстановления будут вызваны с кодом
/* действия Отменить.
/*
/*****/
static int undoStartCrg(int role,
                       int doesNotApply,
                       Qcst_EXTP0100_t *crgData,
                       EpData *epData) {

    return QcstSuccessful;
} /* конец функции undoStartCrg()

/*****/
/*
/* Код действия = QcstCrgAcUndo
/*
/* Предыдущий код действия = QcstCrgAcEnd
/*
/* Замечания:

```

```

/* CRG не завершается. Если программа выхода выполнила действия по */
/* завершению работы приложения, то она может перезапустить приложение */
/* или принять решение о нецелесообразности перезапуска. Если приложение*/
/* не перезапускается, то код возврата должен быть равен QcstFailWithOutRestart, */
/* чтобы CRG можно было перевести в состояние Под вопросом. */
/* */
/*****/
static int undoEndCrg(int role,
                    int doesNotApply,
                    Qcst_EXTP0100_t *crgData,
                    EpData *epData) {

    return QcstFailWithOutRestart;
} /* конец функции undoEndCrg() */

/*****/
/* */
/* Код действия = QcstCrgAcUndo */
/* */
/* Предыдущий код действия = QcstCrgAcReJoin */
/* */
/* */
/* Замечания: */
/* Произошла ошибка, которая не позволит добавить компонент в эту группу */
/* CRG. Необходимо проверить все действия, выполненные при получении */
/* кода действия Добавить, и определить те из них, которые следует */
/* отменить, если этот элемент не является активным элементом группы CRG.*/
/* */
/*****/
static int undoMemberIsJoining(int role,
                              int doesNotApply,
                              Qcst_EXTP0100_t *crgData,
                              EpData *epData) {

    return QcstFailWithOutRestart;
} /* конец функции undoMemberIsJoining() */

/*****/
/* */
/* Код действия = QcstCrgAcUndo */
/* */
/* Предыдущий код действия = QcstCrgAcFailover */
/* */
/* */
/* Замечания: */
/* Это не означает, что сбой узла или компонента отменен. Также это не */
/* означает необратимость сбоя. Это означает ошибку программы выхода при */
/* обработке кода действия Автоматический перенос ресурсов. Также после */
/* вызова программы выхода могла быть запущена служба ресурсов кластера для*/
/* устранения неполадки. Если CRG была активной в процессе автоматического*/
/* переноса ресурсов, то не в этом участке. Устойчивый ресурс завершает */
/* работу и ожидает вмешательства пользователя. После устранения неполадки */
/* следует запустить CRG с помощью API Запустить CRG. */
/* */
/* */
/*****/
static int undoMemberIsLeaving(int role,
                              int doesNotApply,
                              Qcst_EXTP0100_t *crgData,
                              EpData *epData) {

    return QcstFailWithOutRestart;
} /* конец функции undoMemberIsLeaving() */

/*****/
/* */

```

```

/* Код действия = QcstCrgAcUndo */
/* */
/* Предыдущий код действия = QcstCrgAcSwitchover */
/* */
/* Замечания: */
/* Возникла ошибка в процессе перемещения точки доступа из */
/* исходного основного узла в новый основной узел. */
/* До перемещения точки доступа IP-адрес в исходном основном узле был */
/* завершен и снова запущен в этом же узле. Теперь служба ресурсов */
/* кластера пытается вернуть точку доступа обратно в исходный основной */
/* узел. Программа выхода приложения и устойчивый IP-адрес */
/* будут снова запущены в исходном основном узле. */
/* */
/* */
/*****/
static int undoSwitchPrimary(int role,
                             int doesNotApply,
                             Qcst_EXTP0100_t *crgData,
                             EpData *epData) {

    return QcstFailWithOutRestart;
} /* конец функции undoSwitchPrimary() */

/*****/
/* */
/* Код действия = QcstCrgAcUndo */
/* */
/* Предыдущий код действия = QcstCrgAcAddNode */
/* */
/* Замечания: */
/* Если объекты были созданы в новом узле, их следует удалить */
/* для того, чтобы впоследствии можно было выполнить команду */
/* Добавить узел в домен восстановления. */
/* */
/* */
/*****/
static int undoAddNode(int role,
                       int doesNotApply,
                       Qcst_EXTP0100_t *crgData,
                       EpData *epData) {

    return QcstSuccessful;
} /* конец функции undoAddNode() */

/*****/
/* */
/* Код действия = QcstCrgAcUndo */
/* */
/* Предыдущий код действия = QcstCrgAcRemoveNode */
/* */
/* Замечания: */
/* Узел все еще находится в домене восстановления. Если объекты были */
/* удалены из узла, их следует снова добавить в него. */
/* */
/* */
/*****/
static int undoRmvNode(int role,
                       int doesNotApply,
                       Qcst_EXTP0100_t *crgData,
                       EpData *epData) {

    return QcstFailWithOutRestart;
} /* конец функции undoRmvNode() */

/*****/
/* */

```

```

/* Код действия = QcstCrgAcUndo */
/* */
/* Предыдущий код действия = QcstCrgAcChange */
/* */
/* Замечания: */
/* Изменения, внесенные в CRG, будут отменены для того, чтобы */
/* восстановить состояние CRG и домена восстановления, в */
/* котором они находились до попытки изменения. Изменения, */
/* внесенные программой выхода, также должны быть отменены. */
/* */
/*****/
static int undoChgCrg(int role,
                    int doesNotApply,
                    Qcst_EXTP0100_t *crgData,
                    EpData *epData) {

    return QcstSuccessful;
} /* конец функции undoChgCrg() */

/*****/
/* */
/* Код действия = QcstCrgAcUndo */
/* */
/* Предыдущий код действия = QcstCrgAcCancelFailover */
/* */
/* Замечания: */
/* Это не означает, что сбой узла или компонента отменен. Также это не */
/* означает необратимость сбоя. Это означает ошибку службы ресурсов */
/* кластера при вызове программы выхода. */
/* CRG будет находиться в состоянии Под вопросом независимо от того, */
/* что вернет программа выхода. В данном случае требуется вмешательств */
/* пользователя. После устранения неполадки следует запустить CRG */
/* с помощью API Запустить CRG. */
/* */
/* */
/*****/
static int undoCancelFailover(int role,
                             int doesNotApply,
                             Qcst_EXTP0100_t *crgData,
                             EpData *epData) {

    return QcstSuccessful;
} /* конец функции undoCancelFailover() */

/*****/
/* */
/* Простая процедура, позволяющая из имени объекта и имени библиотеки, */
/* оканчивающихся символом NULL, создать 20-символьное полное имя, не */
/* оканчивающееся символом NULL. */
/* */
/*****/
static void bldDataAreaName(char *objName, char* libName, char *qualName) {

    memset(qualName, 0x40, 20);
    memcpy(qualName, objName, strlen(objName));
    qualName += 10;
    memcpy(qualName, libName, strlen(libName));
    return;
} /* конец функции bldDataAreaName() */

/*****/
/* */
/* В области данных проверяется готовность всех CRG, связанных с данным */
/* приложением. Если готовности нет, то программа через некоторое время */
/* повторяет проверку. */

```

```

/* Этот процесс продолжается до тех пор, пока не будут готовы все */
/* зависимые CRG или пока не истечет максимальное разрешенное время */
/* ожидания. Время ожидания можно изменить, если это целесообразно */
/* для конкретной ситуации. */
/* */
/* */
/*****/
static int checkDependCrgDataArea(unsigned int maxWaitTime) {

    Qus_EC_t errCode = { sizeof(Qus_EC_t), 0 };
    char dataAreaName[20];
    struct {
        Qwc_Rdtaa_Data_Returned_t stuff;
        char ready;
    } data;

/*-----*/
/*
/* Это общее время ожидания готовности зависимых CRG.
/* */
/* */
/*-----*/
    unsigned int timeWaited = 0;

/*-----*/
/*
/* Определение времени ожидания.
/* */
/* */
/*-----*/
    _MI_Time timeToWait;
    int hours = 0;
    int minutes = 0;
    int seconds = WaitSecondsIncrement;
    int hundreths = 0;
    short int options = _WAIT_NORMAL;
    mitime( &timeToWait, hours, minutes, seconds, hundreths );

/*-----*/
/*
/* Создается полное имя области данных.
/* */
/* */
/*-----*/
    bldDataAreaName(DependCrgDataArea, ApplLib, dataAreaName);

/*-----*/
/*
/* Из области данных загружается информация о готовности всех CRG. Если */
/* приложение может продолжить работу, эта область данных обновляется с */
/* помощью партнеров по разработке приложений высокой готовности. */
/* */
/* */
/*-----*/
    QWCRDTAA(&data,
            sizeof(data),
            dataAreaName,
            offsetof(Qcst_HAAPPO_t,Data_Status)+1, /* API необходим 1 источник */
            sizeof(data.ready),
            &errCode);

/*-----*/

```

```

/*
/* Если зависимые CRG не готовы, программа ожидает в течение заданного
/* времени, затем снова выполняет проверку.
*/
*/

/*-----*/
while (data.ready != Data_Available) {

/*-----*/
/*
/* Если в течение максимального допустимого времени ожидания
/* CRG не готовы, возвращается код ошибки. В таких случаях
/* рекомендуется заносить в протокол соответствующее сообщение об
/* ошибке для дальнейшего определения причины неполадки.
*/
*/
/*
/*-----*/
if (timeWaited >= maxWaitTime)
return QcstFailWithOutRestart;

/*-----*/
/*
/* Позволяет дождаться готовности CRG.
*/
*/
/*
/*-----*/
waittime(&timeToWait, options);
timeWaited += WaitSecondsIncrement;

/*-----*/
/*
/* Из области данных снова загружается информация о готовности CRG.
*/
*/
/*
/*-----*/
QWCRDTAA(&data,
sizeof(data),
dataAreaName,
offsetof(Qcst_HAAPPO_t,Data_Status)+1, /* API необходим 1 источник */
sizeof(data.ready),
&errCode);
}

return QcstSuccessful;
} /* конец функции checkDependCrgDataArea() */

/*****
/*
/* Обновляется область данных CRG приложения. При этом в ней отображается,
/* что приложение выполняется или не выполняется. Информация из этой
/* области данных применяется партнерами по разработке приложений высокой
/* готовности для координации операций принудительного переноса
/* ресурсов между CRG, зависимыми друг от друга.
*/
*/
/*****
static void setApp1CrgDataArea(char status) {

char cmd[54];
char cmdEnd[3] = {0x00, '}', 0x00};

/*-----*/
/*
*/

```

```

/* Создается командная строка CL с именем библиотеки области          */
/* данных, именем области данных, а также символом для записи в        */
/* область данных. Затем выполняется команда CL.                        */
/*                                                                       */

/*-----*/
memcpy(cmd, "CHGDTAARA DTAARA(", strlen("CHGDTAARA DTAARA")+1);
strcat(cmd, ApplLib);
strcat(cmd, "/");
strcat(cmd, ApplCrgDataArea);
strcat(cmd, " (425 1)) VALUE("); /* @A1C */
cmdEnd[0] = status;
strcat(cmd, cmdEnd);

system(cmd);

return;
} /* конец функции setApp1CrgDataArea() */

/*****
/*
/* Эта функция вызывается всякий раз когда программа получает исключительную*/
/* ситуацию, которая не отслеживается обработчиком исключительных ситуаций. Укажите */
/* здесь соответствующие процедуры очистки. */
/* При этом задается код возврата сбой и управление передается операционной*/
/* системе. Задание, управляющее этой программой выхода, завершается. */
/*
/* Если исключительная ситуация произошла до определения роли          */
/* данного узла, то при вызове этой функции переменная myData->role может */
/* содержать значение UnknownRole. Поэтому, перед принятием            */
/* решений на основе роли узла рекомендуется проверить, определено     */
/* ли соответствующее значение.                                         */
/*
*****/
static void unexpectedExceptionHandler(_INTRPT_Hndlr_Parms_T *exData) {
/*-----*/
/*
/* Создается указатель на структуру, в которой содержатся              */
даннные, переданные программе обработки исключительных ситуаций.
/*
/*
/*-----*/
HandlerDataT *myData = (HandlerDataT *)exData->Com_Area;

/*-----*/
/*
/* Выполните необходимые действия по очистке. Обработчику исключительных*/
/* ситуаций может потребоваться глобальная информация, описывающая     */
/* этапы, выполненные перед сбоем. На их основе подбираются действия по */
/* очистке. Эту информацию можно сохранить в структуре                  */
/* HandlerDataT или в другом месте, доступном для этой функции.        */
/*
/*
/*-----*/

/*-----*/
/*
/* Если приложение запущено на основном узле, оно завершается.         */
/* Это необходимо потому, что программа выхода будет вызвана снова     */
/* с кодом действия Перезапустить, при этом функция restartCrg() всегда */
/* должна работать однотипно. Кроме того, при завершении работы приложения*/
/* может сброситься условие, вызвавшее данную исключительную ситуацию. */
/* По возможности предупредите пользователей и попросите их закрыть   */

```

```

/* приложение, чтобы можно было правильно обработать исключительную ситуацию. */
/*                                                                 */

/*-----*/
endApplication(myData->actionCode,
               myData->role,
               myData->priorRole,
               myData->crgData,
               myData->epData);

/*-----*/
/*                                                                 */
/* Задается код возврата программы выхода.                       */
/*                                                                 */
/*-----*/
*myData->retCode = QcstFailWithRestart;

/*-----*/
/*                                                                 */
/* Исключительной ситуации разрешается фильтрация стека вызовов. */
/*                                                                 */
/*-----*/
return;
} /* конец функции unexpectedExceptionHandler()                  */

/*****
/*                                                                 */
/* Эта функция вызывается при отмене задания, управляющего программой */
/* выхода. Задание может быть отменено по следующим причинам        */
/* (список может быть неполным)-                                     */
/* - Отмена активного CRG приложения с помощью API. API           */
/*   Завершить CRG, Выполнить принудительный перенос ресурсов,     */
/*   Остановить узел кластера, Удалить узел кластера и Удалить     */
/*   кластер отменяют задание, переданное на выполнение в момент, */
/*   когда программа выхода уже была вызвана с кодом действия     */
/*   Запустить.                                                     */
/* - Отмена задания оператором с помощью одного из меню           */
/*   операционной системы. Например, Работа с активными заданиями. */
/* - Завершение работы подсистемы, в которой выполнялось это задание */
/* - Завершение работы всех подсистем                             */
/* - Выключение системы                                           */
/* - Машинный сбой операционной системы                           */
/*                                                                 */
/* Если отмена задания произошла до определения роли данного       */
/* узла, то при вызове этой функции переменная myData->role может  */
/* содержать значение UnknowRole. Поэтому, перед принятием решений на */
/* основе роли узла рекомендуется проверять, определено соответствующее */
/* значение.                                                       */
/*                                                                 */
*****/
static void cancelHandler(_CNL_Hndlr_Parms_T *cnlData) {

/*-----*/
/*                                                                 */
/* Создается указатель на структуру, в которой содержатся данные,   */
/* переданные программе обработки отмены.                            */
/*                                                                 */
/*-----*/
HandlerDataT *myData = (HandlerDataT *)cnlData->Com_Area;

```



```

/*-----*/
/*
/* Выполните необходимые действия по очистке. Обработчику отмены может
/* потребоваться глобальная информация, описывающая этапы, выполненные
/* перед отменой задания. На ее основе обработчик выясняет, полностью
/* или частично завершилась функция, и определяет необходимость очистки.
/* Эту информацию можно сохранить в структуре HandlerDataT
/* или в любом другом месте, доступном для
/* этой функции.
/*
/*-----*/

/*-----*/

/*
/* Задание отменяется. Если приложение запущено с помощью кода
/* действия Запустить или Перезапустить, то его работа завершается.
/* Задание завершается вследствие изменения основного узла в результате
/* применения API Принудительный перенос ресурсов, либо вследствие отмены
/* задания пользователем с помощью команды CL, системного меню и т.п.
/*
/*-----*/

endApplication(myData->actionCode,
               myData->role,
               myData->priorRole,
               myData->crgData,
               myData->epData);

/*-----*/
/*
/* Задается код возврата программы выхода.
/*
/*-----*/

*myData->retCode = QcstSuccessful;

/*-----*/
/*
/* Возврат к операционной системе для окончательного завершения
/* задания.
/*
/*-----*/

return;
} /* конец функции cancelHandler()

/*****
/*
/* Общая процедура, применяемая для завершения приложения с помощью
/* различных функций кода действия, программы обработки исключений и
/* программы обработки отмены заданий.
/*
/*-----*/
static void endApplication(unsigned int actionCode,
                          int role,
                          int priorRole,
                          Qcst_EXTP0100_t *crgData,
                          EpData *epData) {

```

```

if ( role == QcstPrimaryNodeRole
    &&
    crgData->Original_Cluster_Res_Grp_Stat == QcstCrgActive) {
/*-----*/
/*
/* Укажите здесь процедуры для завершения приложения. Кроме того,
/* вам может потребоваться указать процедуры, позволяющие
/* определить, если приложение все еще выполняется, так как эту
/* функцию можно вызвать только один раз для кода действия и второй
/* раз с помощью программы обработки отмены заданий (Например,
/* Завершить CRG).
*/
*/
/*-----*/

/*-----*/
/*
/* После завершения приложения обновите область данных для отображения
/* того, что приложение больше не выполняется.
*/
*/
/*-----*/
    setApp1CrgDataArea(App1_Ended);
}

return;
} /* конец функции endApplication() */

/*****
/*
/* Печать данных, переданных этой программе.
/*
/*
*****/
static void printParms(int actionCode,
                      int role,
                      int priorRole,
                      Qcst_EXTP0100_t *crgData,
                      EpData *epData) {

    unsigned int i;
    char *str;

    /* Печать кода действия */
    printf("%s", "Action_Code = ");
    printActionCode(actionCode);

    /* Печать зависимых данных кода действия */
    printf("%s", " Action_Code_Dependent_Data = ");
    switch (crgData->Action_Code_Dependent_Data) {
        case QcstNoDependentData: str = "QcstNoDependentData";
            break;
        case QcstMerge: str = "QcstMerge";
            break;
        case QcstJoin: str = "QcstJoin";
            break;
        case QcstPartitionFailure: str = "QcstPartitionFailure";
            break;
        case QcstNodeFailure: str = "QcstNodeFailure";
            break;
        case QcstMemberFailure: str = "QcstMemberFailure";
            break;
        case QcstEndNode: str = "QcstEndNode";
            break;
        case QcstRemoveNode: str = "QcstRemoveNode";

```

```

        case QcstApplFailure:      break;
                                  str = "QcstApplFailure";
        case QcstResourceEnd:     break;
                                  str = "QcstResourceEnd";
        case QcstDltCluster:      break;
                                  str = "QcstDltCluster";
        case QcstRmvRcvyDmnNode:  break;
                                  str = "QcstRmvRcvyDmnNode";
        case QcstDltCrg:          break;
                                  str = "QcstDltCrg";
        default: str = "неизвестные зависимые данные кода действия";
    }
    printf("%s \n", str);

    /* Печать предыдущего кода действия */
    printf("%s", " Prior_Action_Code = ");
    if (crgData->Prior_Action_Code)
        printActionCode(crgData->Prior_Action_Code);
    printf("\n");

    /* Печать имени кластера */
    printStr(" Cluster_Name = ",
            crgData->Cluster_Name, sizeof(Qcst_Cluster_Name_t));

    /* Печать имени CRG. */
    printStr(" Cluster_Resource_Group_Name = ",
            crgData->Cluster_Resource_Group_Name, sizeof(Qcst_Crg_Name_t));
    /* Печать типа CRG. */
    printf("%s \n", " Cluster_Resource_Group_Type = QcstCrgApplResiliency");
    /* Печать состояния CRG */
    printf("%s", " Cluster_Resource_Group_Status = ");
    printCrgStatus(crgData->Cluster_Resource_Group_Status);

    /* Печать исходного состояния CRG */
    printf("%s", " Original_Cluster_Res_Grp_Stat = ");
    printCrgStatus(crgData->Original_Cluster_Res_Grp_Stat);

    /* Печать имени очереди распределения информации */
    printStr(" DI_Queue_Name = ",
            crgData->DI_Queue_Name, sizeof(crgData->DI_Queue_Name));
    printStr(" DI_Queue_Library_Name = ",
            crgData->DI_Queue_Library_Name,
            sizeof(crgData->DI_Queue_Library_Name));

    /* Печать атрибутов CRG. */
    printf("%s", " Cluster_Resource_Group_Attr = ");
    if (crgData->Cluster_Resource_Group_Attr & QcstTcpConfigByUsr)
        printf("%s", "Принятый IP-адрес настраивается пользователем");
    printf("\n");

    /* Печать ИД этого узла. */
    printStr(" This_Nodes_ID = ",
            crgData->This_Nodes_ID, sizeof(Qcst_Node_Id_t));

    /* Печать роли этого узла. */
    printf("%s %d \n", " роль этого узла = ", role);

    /* Печать предыдущей роли этого узла. */
    printf("%s %d \n", " предыдущая роль этого узла = ", priorRole);

    /* Печать домена восстановления, из которого получен эта роль. */
    printf("%s", " Node_Role_Type = ");
    if (crgData->Node_Role_Type == QcstCurrentRcvyDmn)
        printf("%s \n", "QcstCurrentRcvyDmn");
    else

```

```

    printf("%s \n", "QcstPreferredRcvyDmn");

    /* Печать ID изменяемого узла (если он существует). */
    printStr(" Changing_Node_ID = ",
             crgData->Changing_Node_ID, sizeof(Qcst_Node_Id_t));

    /* Печать роли изменяемого узла (если он существует). */
    printf("%s", " Changing_Node_Role = ");
    if (crgData->Changing_Node_Role == -3)
        printf("%s \n", "*LIST");
    else if (crgData->Changing_Node_Role == -2)
        printf("%s \n", "не применяется");
    else
        printf("%d \n", crgData->Changing_Node_Role);

    /* Печать устойчивого IP-адреса. */
    printStr(" Takeover_IP_Address = ",
             crgData->Takeover_IP_Address, sizeof(Qcst_TakeOver_IP_Address_t));
    /* Печать имени задания. */
    printStr(" Job_Name = ", crgData->Job_Name, 10);

    /* Печать изменений CRG. */
    printf("%s \n", " Cluster_Resource_Group_Changes = ");
    if (crgData->Cluster_Resource_Group_Changes & QcstRcvyDomainChange)
        printf(" %s \n", "Изменен домен восстановления");
    if (crgData->Cluster_Resource_Group_Changes & QcstTakeOverIpAddrChange)
        printf(" %s \n", "Изменен принятый IP-адрес");

    /* Печать времени ожидания восстановления после сбоя. */
    printf("%s", "Failover_Wait_Time = ");
    if (crgData->Failover_Wait_Time == QcstFailoverWaitForever)
        printf("%d %s \n", crgData->Failover_Wait_Time, "Время ожидания не ограничено");
    else if (crgData->Failover_Wait_Time == QcstFailoverNoWait)
        printf("%d %s \n", crgData->Failover_Wait_Time, "Нулевое время ожидания");
    else
        printf("%d %s \n", crgData->Failover_Wait_Time, "минут");

    /* Печать действия по умолчанию для восстановления после сбоя. */
    printf("%s", "Failover_Default_Action = ");
    if (crgData->Failover_Default_Action == QcstFailoverProceed)
        printf("%d %s \n", crgData->Failover_Default_Action, "Продолжить");
    else
        printf("%d %s \n", crgData->Failover_Default_Action, "Отмена");
    /* Печать имени очереди сообщений восстановления после сбоя. */
    printStr(" Failover_Msg_Queue = ",
             crgData->Failover_Msg_Queue, sizeof(crgData->Failover_Msg_Queue));
    printStr(" Failover_Msg_Queue_Lib = ",
             crgData->Failover_Msg_Queue_Lib,
             sizeof(crgData->Failover_Msg_Queue_Lib));

    /* Печать версии кластера. */
    printf("%s %d \n",
           " Cluster_Version = ", crgData->Cluster_Version);

    /* Печать уровня модификации версии кластера. */
    printf("%s %d \n",
           " Cluster_Version_Mod_Level = ",
           crgData->Cluster_Version_Mod_Level);

    /* Печать запрашивающего пользовательского профайла. */
    printStr(" Req_User_Profile = ",
             crgData->Req_User_Profile, sizeof(crgData->Req_User_Profile));
    /* Печать длины данных структуры. */
    printf("%s %d \n",
           " Length_Info_Returned = ", crgData->Length_Info_Returned);
    /* Печать смещения массива домена восстановления. */
    printf("%s %d \n",

```

```

        " Offset_Rcvy_Domain_Array = ", crgData->Offset_Rcvy_Domain_Array);
/* Печать числа узлов массива домена восстановления. */
printf("%s %d \n",
        " Number_Nodes_Rcvy_Domain = ",
crgData->Number_Nodes_Rcvy_Domain);

/* Печать текущего/нового домена восстановления. */
printRcvyDomain(" Домен восстановления:",
        crgData->Number_Nodes_Rcvy_Domain,
        (Qcst_Rcvy_Domain_Array1_t *)
        ((char *)crgData + crgData->Offset_Rcvy_Domain_Array));
/* Печать смещения массива предыдущего домена восстановления. */
printf("%s %d \n",
        " Offset_Prior_Rcvy_Domain_Array = ",
crgData->Offset_Prior_Rcvy_Domain_Array);

/* Печать числа узлов массива предыдущего домена восстановления. */
printf("%s %d \n",
        " Number_Nodes_Prior_Rcvy_Domain = ",
crgData->Number_Nodes_Prior_Rcvy_Domain);

/* Печать предыдущего домена восстановления, если он был передан. */
if (crgData->Offset_Prior_Rcvy_Domain_Array) {
    printRcvyDomain(" Предыдущий домен восстановления:",
        crgData->Number_Nodes_Prior_Rcvy_Domain,
        (Qcst_Rcvy_Domain_Array1_t *)
        ((char *)crgData + crgData->Offset_Prior_Rcvy_Domain_Array));
}

return;
} /* конец функции printParms() */

/*****
/*
/* Печать строки кода действия.
/*
/*
*****/
static void printActionCode(unsigned int ac) {

char *code;
switch (ac) {
    case QcstCrgAcInitialize: code = "QcstCrgAcInitialize";
                                break;
    case QcstCrgAcStart:      code = "QcstCrgAcStart";
                                break;
    case QcstCrgAcRestart:   code = "QcstCrgAcRestart";
                                break;
    case QcstCrgAcEnd:       code = "QcstCrgAcEnd";
                                break;
    case QcstCrgAcDelete:    code = "QcstCrgAcDelete";
                                break;
    case QcstCrgAcReJoin:    code = "QcstCrgAcReJoin";
                                break;
    case QcstCrgAcFailover:  code = "QcstCrgAcFailover";
                                break;
    case QcstCrgAcSwitchover: code = "QcstCrgAcSwitchover";
                                break;
    case QcstCrgAcAddNode:   code = "QcstCrgAcAddNode";
                                break;
    case QcstCrgAcRemoveNode: code = "QcstCrgAcRemoveNode";
                                break;
    case QcstCrgAcChange:    code = "QcstCrgAcChange";
                                break;
    case QcstCrgAcDeleteCommand: code = "QcstCrgAcDeleteCommand";
                                break;
    case QcstCrgAcUndo:      code = "QcstCrgAcUndo";
}
}

```

```

        case QcstCrgEndNode:      break;
                                code = "QcstCrgEndNode";
        case QcstCrgAcAddDevEnt:  break;
                                code = "QcstCrgAcAddDevEnt";
        case QcstCrgAcRmvDevEnt:  break;
                                code = "QcstCrgAcRmvDevEnt";
        case QcstCrgAcChgDevEnt:  break;
                                code = "QcstCrgAcChgDevEnt";
        case QcstCrgAcChgNodeStatus: break;
                                code = "QcstCrgAcChgNodeStatus";
        case QcstCrgAcCancelFailover: break;
                                code = "QcstCrgAcCancelFailover";
        case QcstCrgAcVerificationPhase: code = "QcstCrgAcVerificationPhase";
break;
        default:                  code = "unknown action code";
                                break;
    }
    printf("%s", code);

    return;
} /* конец функции printActionCode() */

/*****
/*
/* Печать состояния CRG
/*
/*
*****/
static void printCrgStatus(int status) {

    char * str;
    switch (status) {
        case QcstCrgActive:      str = "QcstCrgActive";
                                break;
        case QcstCrgInactive:    str= "QcstCrgInactive";
                                break;
        case QcstCrgIndoubt:     str = "QcstCrgIndoubt";
                                break;
        case QcstCrgRestored:    str = "QcstCrgRestored";
                                break;
        case QcstCrgAddnodePending: str = "QcstCrgAddnodePending";
break;
        case QcstCrgDeletePending: str = "QcstCrgDeletePending";
                                break;
        case QcstCrgChangePending: str = "QcstCrgChangePending";
                                break;
        case QcstCrgEndCrgPending: str = "QcstCrgEndCrgPending";
                                break;
        case QcstCrgInitializePending: str = "QcstCrgInitializePending";
break;
        case QcstCrgRemovenodePending: str = "QcstCrgRemovenodePending";
break;
        case QcstCrgStartCrgPending: str = "QcstCrgStartCrgPending";
break;
        case QcstCrgSwitchOverPending: str = "QcstCrgSwitchOverPending";
break;
        case QcstCrgDeleteCmdPending: str = "QcstCrgDeleteCmdPending";
break;
        case QcstCrgAddDevEntPending: str = "QcstCrgAddDevEntPending";
break;
        case QcstCrgRmvDevEntPending: str = "QcstCrgRmvDevEntPending";
break;
        case QcstCrgChgDevEntPending: str = "QcstCrgChgDevEntPending";
break;
        case QcstCrgChgNodeStatusPending: str = "QcstCrgChgNodeStatusPending";
break;
        default: str = "неизвестное состояние CRG";
    }
}

```

```

    }
    printf("%s \n", str);

    return;
} /* конец функции printCrgStatus() */

/*****
*/
/* Печать домена восстановления. */
/*
*/
*****/
static void printRcvyDomain(char *str,
                           unsigned int count,
                           Qcst_Rcvy_Domain_Array1_t *rd) {

    unsigned int i;
    printf("\n %s \n", str);
    for (i=1; i<=count; i++) {
        printStr("    Node_ID = ", rd->Node_ID, sizeof(Qcst_Node_Id_t));
        printf("%s %d \n", "    Node_Role = ", rd->Node_Role);
        printf("%s", "    Membership_Status = ");
        switch (rd->Membership_Status) {
            case 0: str = "Активен";
                    break;
            case 1: str = "Неактивен";
                    break;
            case 2: str = "Раздел";
                    break;
            default: str = "неизвестное состояние узла";
        }
        printf("%s \n", str);
        rd++;
    }
    return;
} /* конец функции printRcvyDomain() */
/*****
*/
/* Объединяет строку, оканчивающуюся символом NULL со строкой, не
*/
/* оканчивающейся символом NULL, и распечатывает результат. */
/*
*/
*****/
static void printStr(char *s1, char *s2, unsigned int len) {

    char buffer[132];
    memset(buffer, 0x00, sizeof(buffer));
    memcpy(buffer, s1, strlen(s1));
    strcat(buffer, s2, len);
    printf("%s \n", buffer);
    return;
} /* конец функции printStr */

```

Проектирование кластеров

Этот раздел посвящен подготовке к созданию кластеров на серверах iSeries. Здесь указаны условия, которые должны быть выполнены для создания кластера. Вы также найдете советы по оптимальной настройке сети и повышению производительности.

В этом разделе перечислены требования, которые необходимо учитывать при работе с кластерами. Вы получите общее представление о кластерах, узнаете об условиях их применения и получите рекомендации по разработке схемы кластера.

Способы настройки кластеров и работы с ними

В службах ресурсов кластера предусмотрена основная инфраструктура для работы с кластерами. В ее состав входят методы, позволяющие применять функции служб ресурсов кластеров.

Служба ресурсов кластера i5/OS в iSeries предоставляет базовые функции по работе с кластерами. Служба ресурсов кластера состоит из компонентов, контролирующих топологию кластера, следящих за пульсом, а также предоставляющих возможность создания конфигурации кластера и групп ресурсов и работы с этими объектами. Кроме того, служба ресурсов кластера содержит функцию передачи надежных сообщений, которая позволяет следить за состоянием узлов кластера и синхронизировать информацию о состоянии ресурсов кластера на всех узлах.

Помимо службы ресурсов кластера, выполняющей базовые функции, предусмотрено несколько пользовательских интерфейсов для работы с кластером. У каждого из них свои достоинства и недостатки.

Важное замечание: Рекомендуется применять только одно из этих решений. Одновременное применение разных интерфейсов для работы с одним кластером может привести к конфликтам, неполадкам и непредсказуемым результатам выполнения рядовых операций. В справочной системе iSeries Information Center можно найти инструкции по работе с интерфейсом Навигатора iSeries, с командами CL и соответствующими API. Если вы планируете пользоваться продуктами партнеров фирмы IBM по разработке кластерных приложений, то ознакомьтесь с документацией к этим продуктам.

Навигатор iSeries - Управление кластером

Компания IBM разработала интерфейс управления кластером, предназначенный для Навигатора iSeries. Этот интерфейс поставляется в составе Компонента 41 (i5/OS - HA Switchable Resources).

| С помощью этого интерфейса можно создавать кластеры с переносимыми ASP и работать с ними. Кроме
| этого, этот интерфейс позволяет создавать и администрировать обычные кластеры, CRG, домены
| управления кластерами и ресурсы.

| **Важное замечание:** Интерфейс управления кластером Навигатора iSeries предоставляет доступ ко всем
| функциям службы ресурсов кластера. Хотя с помощью Навигатора iSeries можно
| выполнить большинство стандартных операций, следует помнить о том, что некоторые
| функции доступны только посредством команд CL, API, либо через специализированные
| продукты партнеров фирмы IBM по разработке кластерных приложений. Например,
| несмотря на то что в кластер iSeries могут входить до 128 узлов, с помощью Навигатора
| iSeries можно работать только с кластерами числом до 4 узлов. В интерфейсе управления
| кластером Навигатора iSeries предусмотрен мастер, упрощающий создание простого
| кластера из четырех узлов. Если в вашем кластере должно быть больше четырех узлов,
| рекомендуем вам воспользоваться командами и API для работы с кластером или
| продуктами партнеров фирмы IBM по разработке кластерных приложений.

| С помощью Навигатора iSeries можно выполнить ряд других операций над кластером. Например:

- Добавление узла в кластер
- Добавление переносимых устройств в кластер
- Добавление переносимых приложений в кластер
- Добавление переносимых данных в кластер
- Изменение роли узлов в домене восстановления
- Изменение описания кластера
- Удаление кластера
- Запуск кластера
- Завершение работы кластера
- Просмотр сообщений о работе кластера

- | • Создание домена управления кластером
 - | • Добавление записи отслеживаемого ресурса
- | Полный список задач, которые можно решить с помощью Навигатора iSeries, приведен в электронной справке по кластерам.

Примечание: Интерфейс управления кластером Навигатора iSeries не поддерживает репликацию логических объектов. Для этих целей необходимы продукты партнеров фирмы IBM по разработке кластерных приложений.

Понятия, связанные с данным

Навигатор iSeries

“Команды и API для работы с кластерами”

В состав службы ресурсов кластера i5/OS входит набор команд языка CL и соответствующих API, а также средств, которые могут применяться в прикладных программах iSeries.

“Партнеры IBM по разработке кластерных приложений и кластерные продукты” на стр. 81

Партнеры фирмы IBM по разработке кластерных продуктов предлагают линейку продуктов для управления кластерами, работы с ними, логической репликации данных и выполнения прочих подобных операций.

“Наиболее частые неполадки кластеров” на стр. 137

В этом разделе перечислены наиболее часто встречающиеся неполадки кластеров и обсуждаются способы их предотвращения и устранения их последствий.

Ссылки, связанные с данной

“Часто задаваемые вопросы об управлении кластером с помощью Навигатора iSeries” на стр. 146

Ответы на часто задаваемые вопросы по созданию кластеров и управлению ими с помощью Навигатора iSeries.

Команды и API для работы с кластерами

В состав службы ресурсов кластера i5/OS входит набор команд языка CL и соответствующих API, а также средств, которые могут применяться в прикладных программах iSeries.

При разработке приложений, предназначенных для управления работой кластера, можно пользоваться специальными командами CL и API. Эти команды и API предоставляют доступ к ресурсам кластеров в системе i5/OS.

QUSRTOOL

Для служб ресурсов кластера в библиотеке QUSRTOOL приведены примеры команд, связанных с API, для которых не предусмотрен командный интерфейс. В ряде случаев этими командами можно пользоваться, даже не внося в них изменения. Например, можно изменять параметры пульса и рассылки информации по кластеру. Дополнительные сведения о примерах команд приведены в элементе TCSTINFO файла QUSRTOOL/QATTINFO. В библиотеке QUSRTOOL также приведен пример программы выхода CRG. Его можно использовать как основу для создания собственных программ выхода. В элементе TCSTDTAEXT файла QATTSYSC приведен пример исходного текста программы создания областей данных QCSTHAAPPI и QCSTHAAPP0, а также файла QACSTOSDS (файл описания объектов).

Задачи, связанные с данной

“Добавление узла в кластер” на стр. 102

Добавить узел в кластер можно с помощью Навигатора iSeries или соответствующих команд.

Команды CL и API для работы с кластерами:

Для настройки и активизации кластеров, узлов кластеров и групп ресурсов кластера, а также для управления ими существует ряд API и команд CL.

В следующих таблицах приведены имена и краткие описания команд CL и API, предназначенных для работы с кластерами. Команды CL для работы с кластерами добавлены в операционную систему OS/400 в версии V5R2M0 и более поздние.

В первой таблице представлены команды и API, связанные с настройкой, активизацией и выполнением прочих операций над **кластером и его узлами**. Во второй таблице представлены команды и API для настройки и активизации **групп ресурсов кластера**, а также для управления ею. В третьей таблице приведены команды для работы с **доменом управления кластером**. Четвертая таблица содержит описания API интегрированной операционной системы, служащий для добавления и удаления записей отслеживаемых ресурсов в домене управления кластером.

Таблица 8. Команды CL и API для работы с кластером и узлами

Описание	Команда CL	Имя API
Добавить запись узла кластера Добавляет узел в список членов кластера. Присваивает IP-адреса, которые будут применяться для связи с кластером.	ADDCLUNODE	QcstAddClusterNodeEntry (Добавить запись узла кластера)
Добавить запись домена устройств Добавляет узел в список членов домена устройств. Данный узел можно будет применять в операциях по восстановлению доступа к устойчивым устройствам из указанного домена. Добавление первого узла в домен устройств равносильно созданию этого домена устройств.	ADDDEVDMNE	QcstAddDeviceDomainEntry (Добавить запись домена устройств)
Исправить версию кластера, Изменить версию кластера Корректирует версию кластера - например, для того, чтобы могли применяться новые функции.	CHGCLUVER	Исправить версию кластера (QcstAdjustClusterVersion)
Изменить запись узла кластера Изменяет запись узла кластера. Например, можно изменить IP-адреса, применяемые для связи с кластером.	CHGCLUNODE	Изменить запись узла кластера (QcstChangeClusterNodeEntry)
Изменить службу ресурсов кластера, Изменить конфигурацию кластера Корректирует параметры кластера, влияющие на производительность, для оптимального соответствия текущей среде связи кластера.	CHGCLUCFG	QcstChgClusterResourceServices (Изменить службу ресурсов кластера)
Создать кластер Создает кластер из одного или нескольких узлов.	CRTCLU	Создать кластер (QcstCreateCluster)
Удалить кластер Удаляет кластер. На всех активных узлах кластера прекращается работа службы ресурсов кластера, все узлы удаляются из кластера.	DLTCLU	Удалить кластер (QcstDeleteCluster)

Таблица 8. Команды CL и API для работы с кластером и узлами (продолжение)

Описание	Команда CL	Имя API
<p>Завершить работу узла кластера Завершает работу службы ресурсов кластера на заданных узлах кластера. Эти узлы становятся недоступны кластеру до тех пор, пока они не будут вновь запущены с помощью команды Запустить узел кластера.</p>	ENDCLUNOD	Завершить работу узла кластера (QcstEndClusterNode)
<p>Показать информацию о кластере, Показать данные кластера Показывает сведения о кластере. Например, с помощью этой операции можно получить полный список членов кластера.</p>	DSPCLUINF	Показать информацию о кластере (QcstListClusterInfo)
<p>Показать сведения о домене устройств, Показать информацию о кластере Показывает информацию о доменах устройств кластера. Например, с помощью этой операции можно получить список доменов устройств, определенных на данный момент.</p>	DSPCLUINF	QcstListDeviceDomainInfo (Показать сведения о домене устройств)
<p>Удалить запись узла кластера Удаляет узел из списка членов кластера. Узел удаляется из всех доменов восстановления, на нем прекращается выполнение любых кластерных операций, с него удаляются все объекты службы ресурсов кластера.</p>	RMVCLUNODE	Удалить запись узла кластера (QcstRemoveClusterNodeEntry)
<p>Удалить запись домена устройств Удаляет узел из домена устройств. Если этот узел был последним в домене устройств, то данная операция приводит к удалению этого домена устройств.</p>	RMVDEVDMNE	Удалить запись домена устройств (QcstRemoveDeviceDomainEntry)
<p>Загрузить информацию о кластере, Показать информацию о кластере Показывает сведения о кластере. С помощью этой команды, например, можно определить имя и версию кластера.</p>	DSPCLUINF	Загрузить информацию о кластере (QcstRetrieveClusterInfo)
<p>Показать информацию службы ресурсов кластера, Показать информацию о кластере Показывает параметры конфигурации и производительности службы ресурсов кластера.</p>	DSPCLUINF	Показать информацию службы ресурсов кластера (QcstRetrieveCRSInfo)
<p>Запустить узел кластера Запускает службу ресурсов кластера на узле, входящем в кластер, но не работающем в данный момент.</p>	STRCLUNOD	Запустить узел кластера (QcstStartClusterNode)

Таблица 8. Команды CL и API для работы с кластером и узлами (продолжение)

Описание	Команда CL	Имя API
Работа с кластерами Отображает узлы и объекты кластера и дает возможность выполнения операций над ними.	WRKCLU	нет

Таблица 9. Команды CL и API для работы с группами ресурсов кластера (CRG)

Описание	Команда CL	API
Добавить запись устройства в CRG Добавляет запись устройства в CRG. Устройство становится членом группы переносимых устройств.	ADDCRGDEVE	Добавить запись устройства в CRG (QcstAddClusterResourceGroupDevice)
Добавить узел в домен восстановления, Добавить запись узла в CRG Добавляет новый узел в домен восстановления существующей CRG.	ADDCRGNODE	Добавить узел в домен восстановления (QcstAddNodeToRcvyDomain)
Изменить группу ресурсов кластера Изменяет атрибуты группы ресурсов кластера. Например, можно изменить устойчивый IP-адрес для CRG приложения.	CHGCRG	QcstChangeClusterResourceGroup (Изменить группу ресурсов кластера)
Изменить запись устройства в CRG Изменить запись устройства в CRG. Например, можно изменить режим включения объекта конфигурации при переносе ресурсов.	CHGCRGDEVE	QcstChangeClusterResourceGroupDev (Изменить запись устройства в CRG)
Создать группу ресурсов кластера Создает объект CRG. Объект CRG, в частности, содержит определение домена восстановления (множества узлов, обеспечивающих устойчивость ресурсов данной CRG).	CRTCRG	Создать группу ресурсов кластера (QcstCreateClusterResourceGroup)
Удалить группу ресурсов кластера Удаляет группу ресурсов кластера (CRG) на локальном узле. Для удаления локальной группы ресурсов кластера требуется, чтобы служба ресурсов кластера была неактивной.	DLTCRG	нет
Удалить группу ресурсов кластера, Удалить CRG из кластера Удаляет CRG из кластера. Объект CRG будет удален из всех активных узлов в домене восстановления.	DLTCRGCLU	Удалить группу ресурсов кластера (QcstDeleteClusterResourceGroup)
Разослать информацию Рассылает информацию о CRG с данного узла на остальные узлы домена восстановления этой CRG.	нет	Разослать информацию (QcstDistributeInformation)

Таблица 9. Команды CL и API для работы с группами ресурсов кластера (CRG) (продолжение)

Описание	Команда CL	API
Завершить работу CRG Отключает функцию обеспечения устойчивости CRG. После выполнения данной операции CRG переходит в состояние 'отключена'.	ENDCRG	Завершить работу группы ресурсов кластера (QcstEndClusterResourceGroup)
Выполнить принудительный перенос ресурсов, Изменить главный узел группы ресурсов кластера Выполняет процедуру принудительного переноса ресурсов в CRG. Домен восстановления изменяется следующим образом: текущий главный узел становится последним резервным узлом, а текущий первый резервный узел - главным узлом.	CHGCRGPRI	Выполнить принудительный перенос ресурсов (QcstInitiateSwitchover)
Показать группу ресурсов кластера, Показать информацию о CRG Показывает сведения о группах ресурсов, определенных в кластере на данный момент.	DSPCRGINF	Показать сведения о CRG (QcstListClusterResourceGroups)
Показать сведения о CRG, Показать информацию о CRG Показывает содержимое объекта CRG. С помощью данной операции можно, например, просмотреть текущие роли узлов в домене восстановления CRG.	DSPCRGINF	Показать сведения о CRG (QcstListClusterResourceGroupInf)
Удалить запись устройства из CRG Удаляет запись устройства из CRG. Устройство перестанет быть перемещаемым.	RMVCRGDEVE	QcstRemoveClusterResourceGroupDev (Удалить запись устройства из CRG)
Удалить узел из домена восстановления, Удалить запись узла из CRG Удаляет узел из домена восстановления CRG. Узел перестанет принимать участие в обеспечении устойчивости CRG.	RMVCRGNODE	Удалить узел из домена восстановления (QcstRemoveNodeFromRcvyDomain)
Запустить CRG Включает средства обеспечения устойчивости для указанного CRG. Группа ресурсов кластера становится активной.	STRCRG	Запустить группу ресурсов кластера (QcstStartClusterResourceGroup)

Примечание: Для служб ресурсов кластера в библиотеке QUSRTOOL приведены примеры команд, иллюстрирующие применение приведенных выше команд CL и API. В ряде случаев этими командами можно пользоваться, даже не внося в них изменения. Например, они хорошо подходят для настройки простых кластеров для тестирования кластерных приложений. Дополнительные сведения о примерах команд приведены в элементе TCSTINFO файла QUSRTOOL/QATTINFO.

Таблица 10. Команды CL для домена управления

Описание	Команда CL	API
<p>Создать домен управления Создает равноправную CRG, соответствующую домену управления кластером. После создания домена управления кластером в него можно добавлять записи отслеживаемых ресурсов (MRE) для синхронизации изменений. Примечание: Для администрирования домена управления кластером можно применять команды CRG. См. раздел Табл. 9 на стр. 78.</p>	CRTADMDMN	Нет
<p>Удалить домен управления Удаляет равноправную CRG, соответствующую домену управления кластером. В процессе удаления из домена удаляются все MRE, и прекращается распространение изменений отслеживаемых ресурсов.</p>	DLTADMDMN	Нет

Таблица 11. Описания API, интегрированных в операционную систему. В дополнение к этим командам CL существуют программные интерфейсы (API), интегрированные в операционную систему. Эти API позволяют добавлять и удалять записи отслеживаемых ресурсов.

Описание	Команды CL ¹	API операционной системы
<p>Добавить запись отслеживаемого ресурса Добавляет запись для системного ресурса и его атрибутов.</p>	Нет	Добавить запись отслеживаемого ресурса (QfpadAddMonitoredResourceEntry)
<p>Удалить запись отслеживаемого ресурса Удаляет запись отслеживаемого ресурса (MRE) из каталога отслеживаемых ресурсов.</p>	Нет	Удалить запись отслеживаемого ресурса (QfpadRmvMonitoredResourceEntry)
<p>Показать сведения об отслеживаемых ресурсах Возвращает информацию об отслеживаемых ресурсах.</p>	Нет	Показать сведения об отслеживаемых ресурсах (QfpadRtvMonitoredResourceInfo)
<p>Примечание:</p> <p>1. Для этой функции эквиваленты команд CL не поддерживаются. В библиотеке QUSRTOOL предоставлен исходный код неподдерживаемых команд и вызова программы обработки. Узнать источник и CPP данной команды можно из элемента QFPADINFO файла QATTINFO.</p>		

Ссылки, связанные с данной

API кластеров

Партнеры IBM по разработке кластерных приложений и кластерные продукты

Партнеры фирмы IBM по разработке кластерных продуктов предлагают линейку продуктов для управления кластерами, работы с ними, логической репликации данных и выполнения прочих подобных операций.

Также партнеры фирмы IBM по разработке кластерных продуктов предлагают специализированные средства для репликации данных и управления кластерами. Если вам потребуется программное обеспечение, полностью совместимое с кластерами и обладающее встроенными функциями логической репликации данных и управления кластерами, обратитесь в торговое представительство фирмы IBM или ее деловых партнеров. Вам предоставят подробное описание всей линейки продуктов, предлагаемых партнерами IBM по разработке кластерных приложений.

Продукты для управления кластерами обладают следующими характеристиками:

- Предоставляют пользовательский интерфейс для создания и обслуживания кластера
- Предоставляют пользовательский интерфейс для создания и обслуживания групп ресурсов кластера
- С помощью API контролируют взаимосвязь между объектами кластера, и в том числе между CRG.
- Позволяют создавать CRG устройств, данных и приложений.

Продукты для репликации обладают следующими характеристиками:

- Создают управляющие структуры для отбора данных и объектов, нуждающихся в репликации.
- Создают CRG для важных данных и связывают их с управляющими структурами.
- Предоставляют программы выхода для CRG данных.

Задачи, связанные с данной

“Добавление узла в кластер” на стр. 102

Добавить узел в кластер можно с помощью Навигатора iSeries или соответствующих команд.

Требования к работе кластеров

В этом разделе перечислены требования, предъявляемые к аппаратуре, программному обеспечению и средствам связи.

Требования к кластерам зависят от того, какими возможностями кластеров вы планируете воспользоваться. Например, для логической репликации данных можно создать простой кластер из двух узлов. Или же вам потребуется кластер для работы с переносимыми дисками и независимыми ASP.

Понятия, связанные с данным

“Примеры: Конфигурации кластеров” на стр. 119

Примеры типичных конфигураций кластеров для разных производственных нужд.

Требования к аппаратному обеспечению в кластере

Функции узла кластера может выполнять любая система iSeries с операционной системой i5/OS версии V4R4M0 или выше.

Рекомендуется застраховаться на случай перебоев в электросети и установить внешний источник бесперебойного питания. В противном случае неожиданный перебой в питании узла кластера может привести к распаду кластера вместо автоматического переноса ресурсов.

Кластеры применяют функции групповой рассылки протокола IP. Разные физические среды передачи данных по-разному приспособлены к групповой рассылке. Информация о требованиях к аппаратному обеспечению для применения групповой рассылки приведена в руководстве TCP/IP Setup.

Если вы планируете применять независимые ASP в кластере, ознакомьтесь с разделом Требования к аппаратному обеспечению. Также можно воспользоваться такими средствами обеспечения целостности данных, как зеркальная защита и защита устройств с проверкой четности. Эти меры защиты в главной системе позволят избежать автоматического переноса ресурсов при выходе диска из строя. На случай, если автоматический перенос ресурсов все же произойдет (например, по другим причинам), эти же меры защиты могут пригодиться и в резервной системе. Дополнительная информация приведена в разделе Защита дисков.

Примечание: Подробное описание прочих требований к кластерам можно найти в разделе “Контрольная таблица настройки кластера” на стр. 93.

Понятия, связанные с данным

Источник бесперебойного питания

“Части кластера” на стр. 28

Часть кластера - это множество активных узлов кластера, образовавшееся в результате сбоя связи. Серверы в отделившейся части по-прежнему поддерживают связь между собой.

“Автоматический перенос ресурсов” на стр. 18

Автоматический перенос ресурсов - это автоматический перенос точки доступа к ресурсу на один из резервных серверов в случае сбоя сервера, выполняющего роль главной точки доступа к ресурсу.

Требования к программному обеспечению и лицензиям в кластере

Для создания кластера требуется специальное программное обеспечение и лицензии.

1. OS/400 версии V4R4M0 или выше с настроенным протоколом TCP/IP (TCP/IP Connectivity Utilities).
2. Программное обеспечение для настройки кластера и работы с ним. Для этого могут применяться следующие продукты:
 - Навигатор iSeries - Управление кластером
 - Специальный программный продукт партнеров фирмы IBM по разработке кластерных продуктов
 - Ваш собственный продукт, использующий команды или API для работы с кластером
3. Дополнительная информация приведена в разделе “Контрольная таблица настройки кластера” на стр. 93

Важное замечание: Для применения независимых дисковых ASP в качестве переносимых устройств должен быть выполнен ряд дополнительных требований. Дополнительная информация приведена в разделе Планирование применения независимых пулов дисков.

Понятия, связанные с данным

“Способы настройки кластеров и работы с ними” на стр. 74

В службах ресурсов кластера предусмотрена основная инфраструктура для работы с кластерами. В ее состав входят методы, позволяющие применять функции служб ресурсов кластеров.

“Версия кластера” на стр. 13

Версией кластера называется идентификатор набора функций, которые может выполнять кластер.

Требования к средствам связи в кластере

Кластер строится на любых средствах связи, поддерживающих протокол IP.

Служба ресурсов кластера работает на базе протокола TCP/IP. При этом поддерживается произвольное сочетание локальных сетей (LAN), глобальных сетей (WAN) и сетей OptiConnect (SAN). При выборе типа сети следует учитывать следующие факторы:

- Количество и объем транзакций
- Требования к быстродействию
- Физическое расстояние между узлами
- Цена

Если вы планируете создать резервные средства связи между узлами кластера, то при выборе типа сети для них следует руководствоваться теми же факторами. При проектировании кластера рекомендуется разместить часть резервных узлов в месте, физически изолированном от расположения основных узлов, на случай стихийных бедствий.

Для предотвращения возможных проблем с производительностью кластера при выборе типа сети следует учитывать объем информации, который планируется передавать между узлами. В настоящее время наиболее широкое распространение получили следующие типы сетей: Token-Ring, Ethernet, ATM, SPD OptiConnect, High-Speed Link (HSL) OptiConnect и Virtual OptiConnect (высокоскоростное внутреннее соединение между логическими разделами).

HSL OptiConnect - это технология, реализованная на базе программного обеспечения OptiConnect for i5/OS (Компонент 23 i5/OS - i5/OS OptiConnect). Она применяется для создания сетей с высокой готовностью. HSL OptiConnect - это высокоскоростная системная сеть топологии точка-точка, связывающая отдельные узлы кластера по технологии High Speed Link (HSL) Loop. В сетях HSL OptiConnect применяется стандартный кабель HSL и не требуется дополнительное аппаратное обеспечение.

Для создания переносимого программного обеспечения, или CRG устойчивых устройств, вам потребуется переносимый независимый ASP. В среде с логическими разделами независимый ASP - это набор дисков, подключенных к общей шине нескольких логических разделов или к процессору ввода-вывода, выделенному пулу ввода-вывода. В среде с несколькими системами независимый это набор переносимых блоков расширения, подключенных к тому же кольцу HSL, что и системы, входящие в соответствующий домен восстановления. В среде LPAR также могут применяться переносимые блоки расширения. Дополнительная информация по работе с переносимым аппаратным обеспечением и независимыми ASP приведена в разделе Планирование работы с независимыми ASP.

Примечание: Если в вашей среде применяются адаптеры 2810, на которых настроен **только** протокол TCP/IP, а протоколы SNA и IPX не применяются, то в системе версии V4R5M0 можно увеличить пропускную способность адаптеров путем указания опции Применять только для TCP(*YES) в описаниях соответствующих линий с помощью команды Работа с описаниями линии (WRKLIND). Эта опция автоматически применяется в версии V5R1M0 и последующих версиях.

Информация, связанная с данной

OptiConnect для i5/OS

Разработка структуры кластера

В этом разделе приведены рекомендации по разработке структуры кластера.

Для каждой рабочей среды можно разработать структуру кластера, которая будет оптимально отвечать ее требованиям. Поэтому перед созданием кластера следует тщательно распланировать его структуру.

Разработка структуры сети для кластера

Перед настройкой сети для кластера целесообразно выбрать подходящую структуру сети и выполнить ряд подготовительных действий по настройке - в частности это касается протокола TCP/IP.

Перед началом настройки кластера настоятельно рекомендуется ознакомиться со следующими разделами:

- Настройка IP-адресов
- Выбор атрибутов конфигурации TCP/IP
- Предотвращение распада кластера

Информация по настройке избыточных средств связи и по определению потребности в выделенной сети для кластера приведена в разделе Выделенная сеть для кластера.

Настройка IP-адресов:

Служба ресурсов кластера поддерживает *только* протокол IP, и поэтому все узлы кластера должны быть *доступны по протоколу IP*.

Это означает, что на всех узлах должны быть настроены интерфейсы IP. Узлам должны быть назначены IP-адреса - это можно сделать вручную, создав таблицы маршрутов TCP/IP, либо автоматически с помощью специализированных протоколов. Таблица маршрутов TCP/IP - это аналог карты, по которой ведется поиск узлов. Каждому узлу должен быть присвоен *уникальный* IP-адрес. Каждому узлу можно назначить до двух IP-адресов. Эти адреса не должны изменяться приложениями, не связанными с кластерами. При назначении IP-адресов запишите соответствие между адресами и линиями связи. Если по каким-либо причинам одна из линий будет более предпочтительна, чем другая, то предпочтительной линии следует назначить первый IP-адрес. Первый назначенный IP-адрес рассматривается функцией надежных сообщений и средствами контроля пульса как предпочитаемый. Все IP-адреса каждого узла должны иметь возможность связи со всеми остальными IP-адресами кластера. Связь между адресами проверяется так: между адресами в обоих направлениях должен проходить пакет ping и сообщение traceroute протокола UDP.

Примечание: Для работы кластера необходимо, чтобы был настроен и работал циклический интерфейс (127.0.0.1). Этот адрес применяется для отправки сообщений локальному узлу и в большинстве случаев активизируется автоматически. Однако если по каким-либо причинам его работа будет прекращена, перестанет работать служба сообщений кластера.

Понятия, связанные с данным

“Надежные сообщения” на стр. 27

Функция *надежных сообщений* службы ресурсов кластера отвечает за то, чтобы на всех узлах кластера была актуальная информация о состоянии ресурсов кластера.

“Контроль пульса” на стр. 26

Контроль пульса - это функция службы ресурсов кластера, проверяющая работоспособность узлов путем отправки регулярных сигналов с каждого узла кластера всем остальным узлам кластера.

Выбор атрибутов конфигурации TCP/IP:

Для работы службы ресурсов кластера нужно, чтобы ряду атрибутов TCP/IP были присвоены определенные значения.

Эти атрибуты должны быть заданы до начала создания кластера:

- Если вы планируете применять сервер iSeries в качестве маршрутизатора между разными сетями и на нем нет других протоколов маршрутизации, то необходимо присвоить опции пересылки IP-дейтаграмм значение *YES с помощью команды CHGTCPA (Изменить атрибуты TCP/IP).
- Запустите сервер INETD. Инструкции по запуску сервера INETD приведены в разделе Сервер INETD.
- Присвойте параметру UDP CHECKSUM значение *YES с помощью команды CHGTCPA (Изменить атрибуты TCP/IP).
- Если в вашей сети установлены мосты между сетями Token-Ring, присвойте опции пересылки MCAST значение *YES.
- Если узлы кластера связаны с помощью OptiConnect for i5/OS, запустите подсистему QSOC с помощью команды STRSBS(QSOC/QSOC).

Советы: Средства связи кластера:

При создании линий связи учтите следующие особенности.

- Пропускная способность линий связи должна быть достаточной для обычных потоков данных, а также для потоков данных кластера, включая его служебные потоки данных (например, пульс).
- Для повышения надежности не рекомендуется ограничиваться одной линией для связи нескольких узлов.
- Не следует перегружать линию связи, которая используется для контроля работоспособности узлов.

- По возможности следует устранить все узкие места - такие, в которых единичный сбой может привести к нарушению работы кластера. Например, не рекомендуется подключать несколько линий связи к одному адаптеру, одному процессору ввода-вывода или одному блоку расширения.
- При высокой нагрузке на основную сеть рекомендуется выделить отдельную сеть для репликации данных и контроля пульса.
- Если вы пользуетесь функцией групповой рассылки протокола IP, ознакомьтесь с ограничениями, накладываемыми этой функцией на физические среды передачи данных, в руководстве TCP/IP Configuration and Reference.
- Для передачи служебной информации между узлами кластера лучше всего подходит функция групповой рассылки протокола UDP. Если физическая среда передачи данных поддерживает возможность групповой рассылки, то в пределах одной подсети узлы кластера смогут обмениваться служебной информацией в режиме групповой рассылки, что существенно снизит нагрузку на сеть. За пределы локальной сети сообщения всегда передаются в режиме точка-точка протокола UDP. Средства связи кластера реализованы таким образом, чтобы не зависеть от маршрутизации при рассылке групповых сообщений.
- Групповые сообщения нестабильны по своей природе. Частота передачи сообщений зависит от количества узлов в сети и сложности структуры управления кластером и может достигать 40 пакетов в секунду и превышать это значение. Колебания частоты передачи сообщений могут отрицательно сказаться на устаревшем сетевом оборудовании. Например, применение агентов протокола SNMP, которым требуется анализ каждого группового пакета UDP, приводит к перегрузкам сети. Пропускной способности устаревшего сетевого оборудования может быть недостаточно для обработки таких потоков данных. При планировании создания кластера следует обязательно учесть дополнительную нагрузку в виде групповых пакетов UDP на вашу сеть, так как в противном случае кластер может снизить производительность сети в целом.

Понятия, связанные с данным

“Планирование логической репликации” на стр. 89

Логической репликацией называется поддержка нескольких экземпляров данных. Данные копируются с главного узла домена восстановления на резервные узлы. Если главный узел выйдет из строя, точка доступа будет перенесена на один из резервных узлов.

“Надежные сообщения” на стр. 27

Функция *надежных сообщений* службы ресурсов кластера отвечает за то, чтобы на всех узлах кластера была актуальная информация о состоянии ресурсов кластера.

Информация, связанная с данной

Настройка TCP/IP

Предотвращение распада кластера:

Возникновения наиболее часто встречающихся предпосылок к распаду кластера можно избежать путем создания избыточных средств связи между узлами кластера.

Средства связи считаются *избыточными*, если между любыми двумя узлами кластера есть по крайней мере две линии связи. При выходе любой линии из строя связь не будет прервана, и кластер не распадется. Однако следует иметь в виду, что если обе линии связи подключены к одному адаптеру, то узким местом становится адаптер.

- | Еще стоит помнить, что не всегда можно избежать распада кластера. Кластер может распасться при
- | перебоях в питании или аппаратном сбое системы.

Понятия, связанные с данным

“Части кластера” на стр. 28

Часть кластера - это множество активных узлов кластера, образовавшееся в результате сбоя связи. Серверы в отделившейся части по-прежнему поддерживают связь между собой.

“Советы: Средства связи кластера” на стр. 84

При создании линий связи учтите следующие особенности.

“Распад кластера” на стр. 139

Большинство неполадок кластера устранить достаточно легко. Распад кластера - это одна из наиболее серьезных неполадок. В этом разделе приведены рекомендации по предотвращению распада кластера и приведен пример процедуры восстановления распавшегося кластера.

Организация выделенной сети для кластера:

В обычной среде кластер создает незначительную нагрузку на сеть. Тем не менее, настоятельно рекомендуется создать избыточные средства связи для каждого узла кластера.

Если к каждому узлу будут подключены две линии связи, то одну из них можно выделить в исключительное использование кластера, а вторую отдать под прочие сетевые задачи и рассматривать как резервную линию на случай выхода выделенной линии из строя.

Понятия, связанные с данным

“Предотвращение распада кластера” на стр. 85

Возникновения наиболее часто встречающихся предпосылок к распаду кластера можно избежать путем создания избыточных средств связи между узлами кластера.

Кластеры смешанных версий

Процедура создания кластера, в который будут входить узлы с различными версиями кластера, требует некоторых дополнительных действий.

По умолчанию текущая версия кластера становится равной потенциальной версии первого узла. Это приемлемо, если создание кластера начинается с узла с минимальной версией. Однако если вам потребуется начать создание кластера с узла с максимальной версией, то в дальнейшем вы не сможете добавлять в кластер узлы предыдущей версии. Это ограничение можно обойти, если при создании кластера явно указать, что версия кластера должна быть предыдущей по отношению к потенциальной версии первого узла.

Рассмотрим это на примере кластера, состоящего из двух узлов. В кластер входят следующие узлы:

ИД узла	Выпуск	Потенциальная версия
Узел А	V5R3	4
Узел В	V5R4	5

Если начать создание кластера с узла В, то нужно явно указать, что в данный кластер будут входить узлы разных версий. Для этого нужно задать целевую версию кластера 2.

Понятия, связанные с данным

“Версия кластера” на стр. 13

Версией кластера называется идентификатор набора функций, которые может выполнять кластер.

Выбор серверов для кластера

При выборе серверов для кластера в первую очередь следует учитывать способность системы полноценно выполнять функции резервных узлов для данных и приложений на случай переноса ресурсов.

Для этого нужно ответить на следующие вопросы:

- На каких серверах хранятся важные данные и приложения?
- Какие серверы будут резервными для серверов из предыдущего списка?

Ответив на эти вопросы, вы получите список серверов, которые нужно включить в кластер.

Сравнение иерархической и равноправной моделей

Несмотря на то, что и иерархическая, и равноправная CRG обеспечивают устойчивость ресурсов кластера, между ними есть отличия, которые необходимо знать.

| CRG в конкретной можно определить с помощью двух моделей. И для иерархической, и для равноправной модели задаются роли. В иерархической модели роли упорядочиваются. Есть резервные узлы, призванные в случае сбоя обеспечивать доступ к ресурсам на основном узле. В равноправной модели все узлы выполняют одинаковые роли и могут обеспечивать доступ к ресурсу. Упорядочивание здесь не поддерживается.

| **Иерархическая модель**

| В иерархической модели каждому узлу отводится соответствующая роль: главный, резервный или копия. Определение ролей и управление ими осуществляется в домене восстановления. Если один узел настроен как главная точка доступа к ресурсу, то остальные узлы будут обеспечивать резервирование на случай сбоя основного.

| **Равноправная модель**

| В равноправной модели CRG домен восстановления не упорядочивается. В равноправной модели узлы могут быть либо равноправными, либо копиями. Все узлы домена восстановления равноправны и обеспечивают одинаковый доступ к ресурсу.

| **Выбор приложений для кластера**

| Далеко не все приложения могут воспользоваться преимуществами кластеров.

| Прежде всего, приложение должно обладать свойством устойчивости - только тогда для него будет возможен автоматический или принудительный перенос. Устойчивые приложения можно перемещать на резервные узлы без перенастройки клиентов. Для того чтобы приложение было устойчивым, оно должно удовлетворять ряду требований. Дополнительная информация об устойчивых приложениях приведена в разделе Кластерные приложения.

| **Планирование обеспечения устойчивости данных**

| Устойчивыми называются данные, постоянно доступные конечным пользователям и приложениям. Для обеспечения устойчивости применяется логическое копирование и переносимые независимые пулы дисков.

| **Выбор данных, которые должны быть устойчивыми:**

| Определите, какие данные должны быть устойчивыми.

| Эта процедура схожа с отбором данных для резервного копирования. Вам нужно определить, какие данные наиболее важны для вашего предприятия.

| Например, если ваша организация занимается бизнесом в Internet, то наибольшую важность могут представлять следующие данные:

- | • Заказы, полученные сегодня
- | • Склад
- | • Информация о клиентах

| Вряд ли стоит обеспечивать устойчивость данных, которые изменяются крайне редко и не участвуют в повседневной работе.

| **Понятия, связанные с данным**

| Планирование стратегии резервного копирования и восстановления

| **Сравнение логической репликации, коммутируемых дисков и зеркальной защиты, распределенной по сайтам:**

| В этом разделе приведен обзор различных способов обеспечения устойчивости данных, применяемых в кластерной среде для повышения готовности.

| Устойчивость данных обеспечивает доступность данных для приложений и пользователей даже в случае сбоя
| системы, в которой эти данные содержались первоначально. Удачно выбрать комплекс технологий для
| обеспечения устойчивости данных, отвечающих всем необходимым требованиям, довольно сложно. Для
| того чтобы повысить готовность в среде с несколькими системами, необходимо ориентироваться в
| множестве решений, предназначенных для обеспечения устойчивости данных. Вы можете выбрать одно
| решение или сочетать несколько.

Описания различных технологий приведены в разделе Data Resilience Solutions for IBM i5/OS High Availability Clusters. Раздел "Data Resilience Solutions for IBM i5/OS High Availability Clusters" содержит подробное сравнение атрибутов всех технологий.

Логическая репликация

Логическая репликация - это автоматическое создание и поддержка нескольких идентичных копий объекта на разных узлах кластера.

Если для объекта (приложения или данных) включен режим репликации, то он копируется с одного из узлов кластера на другие узлы. При этом экземпляры объекта на всех серверах в домене восстановления полностью идентичны. При изменении объекта на одном из узлов автоматически корректируются все остальные экземпляры. В случае принудительного или автоматического переноса ресурса не требуется никакой синхронизации, и резервный узел может без дополнительной подготовки начать выполнение функций главной точки доступа. Список резервных серверов задается в определении домена восстановления. При сбое главного сервера домена восстановления точка доступа перемещается на первый резервный узел.

Репликацию можно организовать либо собственной программой, либо средствами продуктов Cluster Middleware Business Partner. Дополнительные сведения приведены в разделе Планирование логической репликации.

Переносимые диски

Переносимыми называются ресурсы (например, данные или приложения), расположенные в блоке расширения или процессоре ввода-вывода (IOP) на общей шине или в пуле ввода-вывода логического раздела, которые одновременно доступны нескольким узлам кластера. Например, в случае сбоя главного сервера данные, хранящиеся на независимом ASP, по-прежнему останутся доступны через резервный узел, на который будет перемещена точка доступа.

Для организации перемещаемых ресурсов применяются независимые пулы дисков (ASP). Дополнительная информация приведена в разделе Планирование применения независимых пулов дисков.

Зеркальная защита, распределенная по сайтам

Зеркальная защита, распределенная по сайтам совместно с географической зеркальной защитой позволяют создавать зеркальные копии данных на дисках в офисах, расположенных на значительном расстоянии от данного. Эта технология позволяет расширить область действия группы ресурсов кластера (CRG) за пределы физического соединения компонентов. С помощью географической зеркальной защиты вы можете копировать изменения, внесенные в рабочую копию независимого пула дисков, в зеркальную копию этого пула дисков. После записи данных в рабочую копию независимого пула дисков операционная система создает зеркальную копию этой информации во второй копии независимого пула дисков с помощью другой системы. Этот процесс позволяет сохранять несколько одинаковых копий данных.

С помощью CRG в случае принудительного или автоматического переноса ресурса не требуется никакой синхронизации, и резервный узел может без дополнительной подготовки начать выполнение функций главной точки доступа. Список резервных серверов задается в определении домена восстановления. Резервные узлы могут быть расположены как в одном помещении с основным, так и в разных. При сбое главного сервера домена восстановления точка доступа перемещается на первый резервный узел, в котором

будет расположена рабочая копия независимого пула дисков. Таким образом обеспечивается защита от сбоев, связанных с переносимыми ресурсами.

Таблица 12. Сравнительная характеристика технологий обеспечения устойчивости данных для кластерной среды. Знание особенностей разных технологий обеспечения устойчивости данных незаменимо для выбора оптимального решения для вашей среды.

Фактор	Репликация	Переносимые диски	Зеркальная защита, распределенная по сайтам
Гибкость	Десятки серверов	2 сервера	4 сервера
Критический элемент	Нет	Дисковая подсистема	Нет
Стоимость	<ul style="list-style-type: none"> Дополнительная дисковая память. Программное обеспечение деловых партнеров для репликации. Создать копию диска 	<ul style="list-style-type: none"> Блок расширения с поддержкой переноса ресурсов. Опция 41 	<ul style="list-style-type: none"> Дополнительный диск для зеркальной копии независимого пула дисков Дополнительный блок расширения ввода-вывода Опция 41
Производительность	Расход ресурсов на репликацию	Не влияет	Дополнительная нагрузка на функции географического расширения
Расходы в режиме реального времени	Ведение журналов для объектов	Обслуживание объектов в независимом ASP	Обслуживание объектов в независимом ASP
Территориальное распределение ресурсов	Ограничено только соображениями производительности	Ограниченное - блоки расширения должны быть подключены к серверам с помощью кабеля HSL OptiConnect (максимальная длина - 250 метров)	Ограничено особенностями производительности (Система не накладывает ограничений. Однако ограничения могут быть связаны с временем ответа и пропускной способностью выбранных линий связи.)
Защита от стихийных бедствий	Да	Нет	Да
Резервное копирование в реальном времени	Да	Нет	Нет
Настройка	<ul style="list-style-type: none"> Среда репликации. Выбор объектов для репликации. 	<ul style="list-style-type: none"> Среда независимых ASP. Запись данных в независимые ASP. 	<ul style="list-style-type: none"> Среда независимых ASP (включая установку географической зеркальной защиты) Запись данных в независимые ASP.

Понятия, связанные с данным

“Планирование логической репликации”

Логической репликацией называется поддержка нескольких экземпляров данных. Данные копируются с главного узла домена восстановления на резервные узлы. Если главный узел выйдет из строя, точка доступа будет перенесена на один из резервных узлов.

“Планирование работы с переносимыми независимыми ASP и распределенной по сайтам зеркальной защиты (XSM)” на стр. 90

Хранится только одна копия данных, но она размещена на переносимом аппаратном обеспечении - либо во внешнем блоке, либо в IOP логического раздела.

Планирование логической репликации:

Логической репликацией называется поддержка нескольких экземпляров данных. Данные копируются с главного узла домена восстановления на резервные узлы. Если главный узел выйдет из строя, точка доступа будет перенесена на один из резервных узлов.

Репликацией называется процесс непрерывного копирования объектов в реальном времени. Объекты копируются с одного узла кластера на заранее выбранные другие узлы. Репликация позволяет хранить несколько идентичных экземпляров объектов на разных узлах. Изменения, внесенные в объект на одном из узлов, автоматически реплицируются на другие узлы.

Необходимо выбрать программное обеспечение для логической репликации. Для репликации данных можно воспользоваться следующими программными средствами:

- **Продукты партнеров фирмы IBM**

Партнеры фирмы IBM по разработке кластерных продуктов предлагают специализированное программное обеспечение для репликации данных в кластере. Дополнительная информация приведена в разделе “Партнеры IBM по разработке кластерных приложений и кластерные продукты” на стр. 81.

- **Пользовательские программы репликации**

Стандартные средства управления журналами, поставляемые фирмой IBM, позволяют сохранить данные об операциях, выполненных над объектами системы. Эти данные могут успешно использоваться для логической репликации. Информация по работе с журналами приведена в разделе Управление журналамиSeries.

- **Понятия, связанные с данным**

- Управление журналами

Выбор систем для логической репликации:

При выборе систем для логической репликации необходимо учесть некоторые важные факторы.

Эти факторы следующие:

- Производительность
- Объем памяти
- Важность данных
- Защита на случай стихийных бедствий

На случай автоматического переноса ресурсов всегда полезно знать, какие данные и приложения расположены на главном и резервном узлах. Наиболее важные данные следует расположить на узле, который позволит обеспечить максимальную производительность в случае автоматического переноса ресурсов. На этом узле должно быть достаточно свободной памяти. Если главная система прекратит работу из-за недостатка памяти, после переноса ресурсов на резервном узле также может оказаться недостаточно памяти. На случай полного уничтожения вашего офиса стихийными бедствиями (наводнения, ураганы и тому подобное) рекомендуется расположить резервную систему в физически удаленном месте.

Планирование работы с переносимыми независимыми ASP и распределенной по сайтам зеркальной защиты (XSM):

Хранится только одна копия данных, но она размещена на переносимом аппаратном обеспечении - либо во внешнем блоке, либо в IOP логического раздела.

В случае сбоя главного узла точка доступа к ресурсу переносится на один из резервных узлов. Кроме того, независимые ASP можно применять в среде зеркальной защиты, распределенной по сайтам (XSM). Такой подход позволяет для обеспечения лучшей готовности и защиты создать и поддерживать зеркальную копию независимого ASP в системе, расположенной в удаленном офисе.

Применение распределенной по сайтам зеркальной защиты (XSM) или независимых ASP в качестве переносимых ресурсов кластера требует тщательного планирования.

Понятия, связанные с данным

Планирование работы с независимыми пулами дисков

Защита кластера

В этом разделе приведены рекомендации по применению средств защиты в кластере.

Подготовка узла к добавлению в кластер

Для того чтобы узел можно было добавить в кластер, нужно задать сетевой атрибут ALWADDCLU (Разрешить добавление в кластер).

Для этого нужно выполнить команду Изменить сетевые атрибуты (CHGNETA) на сервере, который будет узлом кластера. Команда Изменить сетевые атрибуты (CHGNETA) служит для изменения сетевых атрибутов системы. Атрибут ALWADDCLU указывает, разрешено ли удаленным серверам включать данный сервер в свои кластеры.

Примечание: Для изменения значения атрибута ALWADDCLU нужны права доступа *IOSYSCFG.

Возможны следующие значения:

*SAME

Значение не изменяется. Система поставляется с заранее установленным значением этого атрибута: *NONE.

*NONE

Другие системы не могут сделать данный сервер узлом кластера.

*ANY Любая другая система может сделать данный сервер узлом кластера.

*RQSAUT

Другая система может сделать данный узел узлом кластера при условии, что будет успешно выполнена идентификация этой системы.

Когда система получает запрос на добавление в кластер, она проверяет значение атрибута ALWADDCLU и выполняет одно из следующих действий: отклоняет запрос, разрешает добавить себя в кластер или запрашивает разрешение с помощью цифровых сертификатов X.509. **Цифровой сертификат** - это электронный аналог удостоверения личности, подлинность которого можно проверить. В случае, если установлен режим проверки, в локальной и удаленной системах должны быть установлены следующие продукты:

- i5/OS Компонент 34 (Диспетчер цифровых сертификатов)
- Cryptographic Access Provider

Если указано значение *RQSAUT, то должен быть настроен список уполномоченных сертификатных компаний сервера защиты кластера i5/OS. Идентификатор сервера защиты кластера - QIBM_QCST_CLUSTER_SECURITY. В этот список нужно включить сертификатные компании, выдавшие сертификаты узлам, которым вы хотите разрешить добавление данной системы в кластер.

Понятия, связанные с данным

Управление цифровыми сертификатами

“Наиболее частые неполадки кластеров” на стр. 137

В этом разделе перечислены наиболее часто встречающиеся неполадки кластеров и обсуждаются способы их предотвращения и устранения их последствий.

Ссылки, связанные с данной

Команда Изменить сетевые атрибуты (CHGNETA)

Рассылка информации по кластеру

- | Объяснение необходимости защиты рассылаемой по кластеру информации.

С помощью API Разослать информацию (QcstDistributeInformation) узлы домена восстановления могут обмениваться сообщениями. Эта возможность часто применяется в программах выхода. Применяя этот API, следует помнить, что данные передаются без шифрования. Поэтому данный API можно применять для рассылки конфиденциальной информации только в случае, если сеть защищена в целом.

Для рассылки временных данных и обмена ими можно пользоваться API хэш-таблиц кластера. Эти API хранят данные во временной памяти. Как только узел выходит из хэш-таблицы кластера, данные пропадают. Указанные API можно вызывать только с узлов, входящих в домен хэш-таблиц кластера. Кроме того, на узле должны быть запущены службы ресурсов кластера.

Прочая информация, передаваемая при обмене сообщениями в кластере, также не шифруется. Это относится и к низкоуровневым сообщениям. Как следствие, если в программах выхода применяются стандартные средства обмена сообщениями, то следует помнить о том, что по сети передаются незашифрованные данные.

Ссылки, связанные с данной

API Разослать информацию (QcstDistributeInformation)

API кластерных хэш-таблиц

Управление пользовательскими профайлами на узлах кластера

| Управлять пользовательскими профайлами на всех узлах кластера можно двумя способами.

| Первый способ заключается в создании домена управления кластером, отслеживающего общие ресурсы кластера. Кроме пользовательских профайлов, домен управления кластером позволяет отслеживать еще несколько типов общих ресурсов. Дополнительная информация приведена в разделе Отслеживаемые ресурсы. Если домен управления кластером активен, то при обновлении пользовательских профайлов изменения автоматически распространяются на остальные узлы. В противном случае изменения распространятся сразу после активизации домена управления кластером.

| **Примечание:** Если требуется сделать общими пользовательские профайлы, синхронизирующие пароли в пределах кластера, то установите значение системного параметра Сохранять идентификационные данные на сервере (QRETSVRSEC) равным 1.

| Второй способ работы с пользовательскими профайлами - посредством Централизованного управления в Навигаторе iSeries. Эта возможность позволяет выполнять операции над несколькими системами и группами систем. С помощью Централизованного управления можно выполнять стандартные задачи оператора одновременно на нескольких узлах кластера. В частности, с помощью Централизованного управления удобно синхронизировать пользовательские профайлы на нескольких системах. Администратор может создать специальную команду синхронизации и выполнять ее всякий раз при создании нового профайла на одном из узлов.

Понятия, связанные с данным

“Пользовательские очереди и структура заданий” на стр. 115

При управлении кластером необходимо разбираться в структуре заданий и принципе работы пользовательских очередей.

“Домен управления кластером” на стр. 8

Домен управления кластером служит для согласованного управления ресурсами узлов кластерной среды.

Замечания по применению кластеров с брандмауэрами

| При создании кластеров в среде с брандмауэрами необходимо учитывать ряд ограничений и требований.

| Если вы применяете кластеры вместе с брандмауэрами, то для каждого узла рекомендуется разрешить обмен исходящими и входящими сообщениями с другими узлами кластера. Такое “окно” в брандмауэре должно быть для каждого адреса кластера на каждом узле, чтобы он мог обмениваться данными со всеми остальными узлами кластера. По сети могут проходить различные типы пакетов IP. В кластерах используются пакеты ping (тип ICMP), UDP и TCP. При настройке брандмауэра пользователь может

фильтровать поток данных по типам. Для обеспечения нормальной работы кластеров с брандмауэрами следует разрешить пакеты ICMP, UDP и TCP. Для исходящего потока можно использовать любой порт, а для входящего - порты 5550 и 5551.

Контрольная таблица настройки кластера

Перед настройкой кластера заполните эту таблицу, чтобы проверить, насколько подготовлена ваша рабочая среда.

Таблица 13. Таблица конфигурации TCP/IP для кластеров

Требования к TCP/IP	
—	Запустить протокол TCP/IP на всех узлах, которые будут включены в кластер (команда Запустить TCP/IP (STRTCP)).
—	Настроить циклический адрес TCP (127.0.0.1) и перевести его в <i>Активное</i> состояние. Состояние можно проверить с помощью команды Работа с состоянием сети TCP/IP (WRKTCPSTS) на каждом узле кластера.
—	Проверить, что IP-адрес кластера находится в состоянии <i>Активен</i> , выполнив команду Работа с состоянием сети TCP/IP (WRKTCPSTS) на нужном узле.
—	Проверить, что сервер INETD активен на всех узлах кластера либо с помощью команды STRTCPSVR *INETD, либо в Навигаторе iSeries с помощью следующих действий: <ol style="list-style-type: none"> 1. В Навигаторе iSeries откройте категорию Сеть. 2. Откройте Серверы. 3. Откройте TCP/IP. 4. Щелкните правой кнопкой на сервере INETD и выберите Запустить. <p>О его наличии можно судить по наличию задания QTOGINTD (пользователь QTCP) в списке активных заданий узла.</p>
—	Проверить, что пользовательскому профайлу для INETD, указанному в файле /QIBM/ProdData/OS400/INETD/inetd.config, предоставлены только минимально необходимые права доступа. В противном случае возникнет ошибка при выполнении команды Запустить узел кластера (STRCLUNOD). По умолчанию в качестве пользовательского профайла для INETD указывается QUSER.
—	Проверить возможность связи и обмена дейтаграммами UDP для всех IP-адресов кластера. Это можно сделать с помощью команды PING с локальным IP-адресом и команды TRACEROUTE с сообщением UDP.
—	Проверить, что порты 5550 и 5551 не применяются другими приложениями. Эти порты зарезервированы для служб кластера IBM. Список занятых портов можно получить с помощью команды Работа с состоянием сети TCP/IP (WRKTCPSTS). Порт 5550 будет открыт в режиме прослушивания для служб кластера с момента запуска сервера INETD.

Если в кластере будут применяться перемещаемые устройства, то для них должны быть выполнены следующие требования:

Таблица 14. Таблица конфигурации устойчивых устройств для кластеров

Требования к устойчивым устройствам	
—	Проверьте, что на всех узлах кластера, входящих в домен устройств, установлен Компонент 41 (HA Switchable Resources) и для него существует действующий ключ лицензии. Обратите внимание, что этот компонент также требуется для работы интерфейса Управление кластером Навигатора iSeries .
—	Для работы с функциями управления дисками Навигатора iSeries следует настроить сервер сервисных средств (STS), доступ к DST и пользовательские профайлы. Дополнительная информация приведена в разделе Настройка связи.

Таблица 14. Таблица конфигурации устойчивых устройств для кластеров (продолжение)

Требования к устойчивым устройствам	
—	<p>Если вы переключаете устойчивые устройства между логическими разделами системы и управляете логическими разделами без помощи НМС, включите поддержку Virtual OptiConnect в разделах. Управление этим продуктом осуществляется при входе в DST. Дополнительная информация приведена в разделе Virtual OptiConnect.</p> <p>Если для управления логическими разделами вы применяете Консоль аппаратного обеспечения, то измените свойства профайла раздела на вкладке OptiConnect и включите поддержку Virtual OptiConnect в каждом разделе в перемещаемой конфигурации. Для вступления изменения в силу активируйте профайл раздела.</p>
—	<p>Если блок расширения подключен к шине HSL OptiConnect и используется двумя системами, и в одной из этих систем есть логические разделы, то во всех логических разделах должна быть включена поддержка HSL OptiConnect. Если вы управляете логическими разделами без помощи НМС, то управление этим продуктом осуществляется при входе в DST.</p> <p>Если для управления логическими разделами вы применяете Консоль аппаратного обеспечения, то измените свойства профайла раздела на вкладке OptiConnect и включите поддержку HSL OptiConnect для каждого раздела в перемещаемой конфигурации. Для вступления изменения в силу активируйте профайл раздела.</p>
—	<p>Если вы переключаете устойчивые устройства между логическими разделами и управляете логическими разделами без помощи НМС, то вы должны настроить либо общую шину для разделов, либо пул ввода-вывода. В одном разделе шина должна быть настроена в режиме "владелец общей шины", а во всех остальных разделах, которые будут участвовать в переключении устройств, - в режиме "пользователь общей шины".</p> <p>Если вы применяете для управления логическими разделами Консоль аппаратного обеспечения, то настройте пул ввода-вывода, содержащий процессор ввода-вывода, адаптер ввода-вывода и все подключенные ресурсы. Этот пул дает возможность переключения независимого пула дисков между разделами. Этот пул ввода-вывода должен быть доступен каждому разделу. Дополнительная информация приведена в разделе Переносимость аппаратного обеспечения. Дополнительные сведения по требованиям к физическому планированию переносимых устройств приведены в разделе Требования к физическому планированию.</p>
—	<p>После подключения блока расширения к шине HSL между двумя системами блок расширения должен быть настроен как переносимый. Дополнительная информация приведена в разделе Переносимость аппаратного обеспечения.</p>
—	<p>После подключения блока расширения к шине HSL нужно перезапустить все серверы, подключенные к этой шине.</p>
—	<p>Значение MTU для всех транзитных участков сети должно быть больше, чем размер фрагмента сообщений кластера. Значение MTU для IP-адреса кластера можно узнать с помощью команды Работа с состоянием сети TCP/IP (WRKTCPSST) на соответствующем узле. Помимо этого, следует проверить значение MTU на всех транзитных участках линии связи. После запуска кластера будет проще уменьшить размер фрагмента сообщения, чем повсеместно увеличить значение MTU. Дополнительная информация о размере фрагмента сообщения приведена в разделе Настраиваемые параметры связи кластера. Текущие параметры производительности можно посмотреть посредством API Показать информацию службы ресурсов кластера (QcstRetrieveCRSInfo) и изменить с помощью API Изменить службу ресурсов кластера (QcstChgClusterResourceServices).</p>

Таблица 15. Таблица конфигурации защиты для кластеров

Требования к защите	
—	<p>Если ошибка возникает при попытке запустить удаленный узел, следует проверить значение атрибута ALWADDCLU (Разрешить добавление в кластер) на удаленном узле. Этому атрибуту должно быть присвоено значение *ANY или *RQSAUT, в зависимости от вашей среды. Для применения значения *RQSAUT на узле должны быть установлены i5/OS Компонент 34 (Диспетчер цифровых сертификатов) и продукт Cryptographic Access Provided. Дополнительная информация о сетевом атрибуте ALWADDCLU приведена в разделе Подготовка узла к добавлению в кластер.</p>
—	<p>Поддержка состояния пользовательского профайла для INETD, указанного в /QIBM/ProdData/OS400/INETD/inetd.config. Специальные права доступа *SECADM и *ALLOBJ не требуются. По умолчанию в качестве пользовательского профайла для INETD указывается QUSER.</p>

Таблица 15. Таблица конфигурации защиты для кластеров (продолжение)

Требования к защите	
—	Убедитесь, что на всех узлах кластера, с которых вызываются кластерные API, существует специальный пользовательский профайл для вызова этого API, и этому профайлу должны быть предоставлены права доступа *IOSYSCFG.
—	Убедитесь, что на всех узлах, входящих в домены восстановления CRG, существует специальный пользовательский профайл для выполнения программ выхода.

Таблица 16. Таблица конфигурации заданий для кластеров

Рекомендации по работе с заданиями	
—	Кластерные API запускают задания для выполнения ряда операций. Эти задания выполняются либо под управлением пользовательского профайла программы выхода, указанной при создании CRG, либо под управлением пользовательского профайла, вызвавшего данный API (только при включении CRG устойчивых устройств). Для подсистемы, с которой связана очередь заданий указанного пользовательского профайла, должно быть снято ограничение на число заданий (*NOMAX), поступающих из данной очереди.
—	Задания направляются в очередь, указанную в описании задания. В свою очередь, описание задания берется из пользовательского профайла, связанного с CRG. По умолчанию задания направляются в очередь QWATCH. Поскольку эта очередь интенсивно используется другими заданиями, ее применение может замедлить выполнение программ выхода. Рекомендуется создать специальную очередь заданий для выполнения кластерных программ выхода.
—	Задания программ выхода используют пул оперативной памяти и атрибуты выполнения, указанные в описании задания. По умолчанию задания будут выполняться в пуле, в котором выполняется большое число прочих заданий с приоритетом 50. Это может привести к снижению производительности заданий программ выхода. Подсистема, вызывающая задания программ выхода, должна помещать их в пул, который не используется другими заданиями. Кроме того, рекомендуется назначить заданиям программ выхода приоритет не ниже 15, чтобы они могли выполняться быстрее большинства других заданий.
—	Системное значение QMLTTHDACN должно быть равным 1 или 2.

Для создания кластеров и работы с ними можно пользоваться разными интерфейсами. Один из таких интерфейсов - Управление кластером Навигатора iSeries. Для применения Навигатор iSeries должны быть выполнены следующие условия:

Таблица 17. Таблица конфигурации Навигатора iSeries для кластеров

Особенности управления кластером с помощью Навигатора iSeries	
—	На всех узлах кластера, входящих в домен устройств, должны быть установлены Компонент 41 (i5/OS - HA Switchable Resources) и соответствующий ключ лицензии.
—	Убедитесь, что запущены все серверы хоста. Это можно сделать с помощью команды STRHOSTSVR (Запустить сервер хоста): STRHOSTSVR SERVER(*ALL)
—	Убедитесь, что запущен сервер централизованного управления. Для этого служит команда STRTCPSVR (Запустить сервер TCP/IP): STRTCPSVR SERVER(*MGTC)

Понятия, связанные с данным

“Навигатор iSeries - Управление кластером” на стр. 74

Компания IBM разработала интерфейс управления кластером, предназначенный для Навигатора iSeries. Этот интерфейс поставляется в составе Компонента 41 (i5/OS - HA Switchable Resources).

“Сервер INETD” на стр. 96

Для выполнения любых операций на узле кластера должен быть запущен сервер INETD.

“Настраиваемые параметры связи кластера” на стр. 96

С помощью API QcstChgClusterResourceServices (Изменить службу ресурсов кластера) можно настраивать ряд параметров службы ресурсов кластера, влияющих на производительность, для достижения максимальной эффективности в конкретной среде. Этот API можно применять только для работы с кластерами версии 2 и выше.

Ссылки, связанные с данной

“Справочная таблица домена управления кластером” на стр. 99

В ней собраны все предварительные требования, которые необходимо выполнить перед созданием домена управления кластером.

Сервер INETD

Для выполнения любых операций на узле кластера должен быть запущен сервер INETD.

Рекомендуется, чтобы этот сервер всегда был запущен на всех узлах кластера.

Работа с Навигатором iSeries

Для выполнения этой операции должен быть установлен и зарегистрирован компонент 41 (i5/OS - HA Switchable Resources).

Для запуска сервера INETD выполните следующие действия:

1. В Навигаторе iSeries откройте категорию **Сеть**.
2. Откройте **Серверы**.
3. Откройте **TCP/IP**.
4. Щелкните правой кнопкой на сервере **INETD** и выберите **Запустить**.

С помощью команд CL и API

Сервер INETD можно запустить с помощью команды STRTCPSVR (Запустить сервер TCP/IP). Укажите для типа сервера значение *INETD. После запуска сервера INETD в списке активных заданий должно появиться задание QTOGINTD (пользователь QTCP).

Понятия, связанные с данным

“Наиболее частые неполадки кластеров” на стр. 137

В этом разделе перечислены наиболее часто встречающиеся неполадки кластеров и обсуждаются способы их предотвращения и устранения их последствий.

Ссылки, связанные с данной

Команда Запустить сервер TCP/IP (STRTCPSVR)

Настраиваемые параметры связи кластера

С помощью API QcstChgClusterResourceServices (Изменить службу ресурсов кластера) можно настраивать ряд параметров службы ресурсов кластера, влияющих на производительность, для достижения максимальной эффективности в конкретной среде. Этот API можно применять только для работы с кластерами версии 2 и выше.

Команда Изменить конфигурацию кластера (CHGCLUCFG) позволяет изменить только базовые параметры производительности, тогда как API QcstChgClusterResourceServices дает гораздо больше возможностей.

С помощью API QcstChgClusterResourceServices и команды CHGCLUCFG можно изменить параметры кластера, связанные с производительностью. На базовом уровне можно выбрать один из предопределенных наборов параметров. Если нужен расширенный уровень настройки, то с помощью того же API можно изменять значения параметров по отдельности. В этом случае рекомендуется консультироваться со специалистами фирмы IBM. Неудачное сочетание значений отдельных параметров может резко снизить производительность кластера.

Когда следует менять параметры, влияющие на производительность кластера?

В команде CHGCLUCFG и API QcstChgClusterResourceServices предусмотрены простые средства изменения параметров производительности, не требующие глубокого понимания принципов работы кластера. На базовом уровне в основном изменения затрагивают чувствительность пульса и время ожидания для отправки сообщений. Предусмотрены следующие значения:

1 (Большое время ожидания/редкий пульс)

2 (Значения по умолчанию)

Для частоты пульса и времени ожидания применяются стандартные значения. Эта опция удобна для восстановления стандартной конфигурации параметров производительности кластера.

3 (Малое время ожидания/частый пульс)

Пульс становится более частым, время ожидания для сообщений сокращается. В этом режиме кластеру потребуется меньше времени на обнаружение сетевых неполадок.

В следующей таблице приведены экспериментальные результаты применения различных значений.

	1 (Низкая чувствительность)			2 (Значения по умолчанию)			3 (Высокая чувствительность)		
	Выявление неполадок пульса	Анализ	Всего	Выявление неполадок пульса	Анализ	Всего	Выявление неполадок пульса	Анализ	Всего
Одна подсеть	00:24	01:02	01:26	00:12	00:30	00:42	00:04	00:14	00:18
Несколько подсетей	00:24	08:30	08:54	00:12	04:14	04:26	00:04	02:02	02:06

| **Примечание:** Время указывается в формате минуты:секунды.

При выборе частоты пульса и времени ожидания следует учитывать особенности нагрузки на сеть и физической среды передачи данных. В высокоскоростной и надежной среде передачи данных (например, когда все системы кластера подключены к одной шине OptiConnect) имеет смысл применять более чувствительную среду в целях ускорения автоматического переноса ресурсов. В такой ситуации лучше выбрать опцию 3. Если же в вашей среде применяется сильно нагруженная шина Ethernet пропускной способностью 10 Мбит/с, и применение стандартных значений приводит к периодическому переносу ресурсов из-за тайм-аутов, вызванных недостаточной пропускной способностью сети, то будет целесообразно выбрать опцию 1 для того, чтобы фильтровать временные увеличения времени отклика в моменты пиковой нагрузки на сеть.

С помощью API Изменить службу ресурсов кластера можно изменять отдельные параметры производительности. Например, предположим, что все узлы кластера подключены к шине OptiConnect. Можно значительно увеличить производительность системы передачи сообщений, увеличив размер фрагмента сообщения до 32500 байт (в соответствии со значением MTU Opticonnect, существенно превышающим стандартное значение в 1464 байта). За счет этого будет снижена фрагментация сообщений и сокращены издержки на их сборку-разборку. Разумеется, выигрыш от такого изменения будет зависеть от характера приложений, применяемых в кластере. Прочие параметры описаны в документации по API и могут применяться для регулировки системы передачи сообщений и скорости обнаружения распада кластера.

Понятия, связанные с данным

“Настройка производительности кластера” на стр. 113

Для обеспечения максимальной эффективности служб кластера в различных рабочих средах предусмотрена возможность изменения параметров, влияющих на производительность кластера.

Справочная таблица по удалению кластера

Если требуется удалить кластер или CRG, то для полного удаления необходимо последовательно удалить все компоненты кластера.

Таблица 18. Справочная таблица по удалению независимых пулов дисков для кластеров

Требования к независимому пулу дисков	
—	Если требуется удалить часть группы независимого пула дисков или, например, последний независимый пул дисков в перемещаемом устройстве, то сначала необходимо завершить работу CRG. Для этого можно применить команду Завершить работу CRG (ENDCRG).
—	Если требуется удалить независимый ASP, входящий в кластер, то настоятельно рекомендуется сначала удалить объект конфигурации этого пула из переносимого устройства (CRG устройств). Для удаления объекта конфигурации пула дисков из переносимого устройства выполните следующие действия: Для удаления пула дисков из переносимого устройства выполните следующие действия: <ol style="list-style-type: none">1. В Навигаторе iSeries откройте Централизованное управление → Кластеры.2. Откройте <i>кластер, содержащий переносимое устройство</i> → Переносимые устройства.3. Щелкните на имени переносимого устройства.4. В правой панели Навигатора iSeries щелкните правой кнопкой мыши на пуле дисков и выберите Удалить. Для этой же цели можно воспользоваться командой Удалить запись устройства CRG (RMVCRGDEVE) .
—	После удаления объекта конфигурации независимого ASP из переносимого устройства кластера можно удалить независимый пул дисков .
—	Удалите описание устройства для независимого пула дисков. Для этого выполните следующие действия: <ol style="list-style-type: none">1. В командной строке введите WRKDEV D DEV D(*ASP) и нажмите Enter.2. Прокрутите текст до нужного описания устройства независимого ASP.3. Выберите опцию 4 (Удалить) рядом с именем описания устройства и нажмите Enter.

Таблица 19. Справочная таблица по удалению группы ресурсов кластера

Требования к группе ресурсов кластера	
—	Для удаления группы ресурсов кластера можно воспользоваться любым из следующих действий: <ol style="list-style-type: none">1. Если узел не входит в кластер, то введите в командной строке команду DLTCRG CRG(CRGNAME). CRGNAME - это имя удаляемой CRG. Нажмите Enter.2. Если узел входит в кластер, то введите в командной строке команду DLTCRGCLU CLUSTER(CLUSTERNAME) CRG(CRGNAME). CLUSTERNAME - имя кластера. CRGNAME - это имя удаляемой CRG. Нажмите Enter.

Планирование домена управления кластером

Планирование домена управления кластером помогает оптимизировать управление общими ресурсами.

При создании домена управления автоматически создается соответствующая равноправная CRG. Управлять доменом можно с помощью API, команд CL и Навигатора iSeries.

Администратор кластера может создать домен управления кластером и добавить в него отслеживаемые ресурсы, общие для узлов. Кластер i5/OS поддерживает список системных ресурсов, которые можно сделать общими в пределах домена управления. Ресурсам в списке соответствуют *записи отслеживаемых ресурсов (MRE)*. Полный список отслеживаемых системных ресурсов приведен в разделе Отслеживаемые ресурсы.

При разработке домена управления кластером необходимо учитывать следующие факторы:

Какие ресурсы будут общими?

Продумайте, какие системные ресурсы требуется сделать общими. Для каждого ресурса можно выбрать определенные атрибуты. Для правильной работы некоторых приложений, использующих

несколько узлов, могут потребоваться специальные переменные среды. Более того, для работы с данными от нескольких узлов могут потребоваться определенные пользовательские профайлы. Также перед настройкой общих ресурсов необходимо проверить соответствие приложений и данных требованиям операционной системы.

Какие узлы будут входить в домен управления кластером?

Определите, какими узлами кластера должен управлять домен управления кластером. Одни и те же узлы не могут входить в несколько доменов. Например, предположим кластер из четырех узлов (Узел А, Узел В, Узел С и Узел D). Узлы А и В относятся к одному домену управления кластером, а узлы С и D - к другому. Однако узлы В и С нельзя добавить в третий домен.

Каким будет соглашение об именах для домена управления кластером?

В зависимости от размера и сложности кластерной среды может потребоваться установить стандартное соглашение об именах для равноправных CRG и доменов управления кластерами. Так как равноправная CRG представляет домен управления кластером, то, возможно, потребуется различать ее и CRG, отслеживающую ресурсы кластера. Например, равноправные CRG для доменов управления кластерами могут называться *ADMDMN1*, *ADMDMN2*, и так далее, а другие равноправные CRG - *PEERI* и так далее. Также для определения, используется ли какая-либо CRG в качестве домена управления кластером, можно воспользоваться API Показать информацию о группе ресурсов кластера (`QcstListClusterResourceGroupIn`).

Справочная таблица домена управления кластером

В ней собраны все предварительные требования, которые необходимо выполнить перед созданием домена управления кластером.

Таблица 20. Справочная таблица домена управления кластером

Требования к домену управления кластером	
—	Проверьте, настроен ли кластер. Дополнительные сведения приведены в разделе Справочная таблица по конфигурации кластера.
—	Если требуется отслеживать пользовательские профайлы, синхронизирующие пароли в пределах кластера, то установите значение системного параметра Сохранять идентификационные данные на сервере (QRETSVRSEC) равным 1.
—	Для того чтобы в домен управления кластером можно было добавить ресурсы, все узлы этого домена должны быть активны, должны входить в группу и не должны быть распавшимися.

Настройка кластеров

Инструкции по настройке кластеров.

IBM совместно со своими партнерами направляет все свои усилия на разработку универсального удобного и простого графического пользовательского интерфейса для управления кластерами. Служба ресурсов кластера i5/OS состоит из компонентов, контролирующих топологию кластера, следящих за пульсом, а также предоставляющих возможность создания конфигурации кластера и групп ресурсов и работы с этими объектами. Кроме того, служба ресурсов кластера содержит функцию передачи надежных сообщений, которая позволяет следить за состоянием узлов кластера и синхронизировать информацию о состоянии ресурсов кластера на всех узлах. В состав службы ресурсов кластера входит набор команд языка CL и соответствующих API и средств, которые могут применяться в прикладных программах iSeries. Кроме того, для работы с функциями службы ресурсов кластера можно воспользоваться интерфейсом управления кластером Навигатора iSeries, а также специализированными продуктами партнеров фирмы IBM по разработке кластерных приложений.

Начало работы

Перед началом настройки кластера выполните следующие действия:

1. Выберите интерфейс для работы с кластером.

Информация о различных способах настройки кластера и работы с ним приведена в разделе “Способы настройки кластеров и работы с ними” на стр. 74.

2. **Убедитесь в том, что выполнены требования к программному и аппаратному обеспечению и к средствам связи.**

Ознакомьтесь с требованиями, приведенными в разделе Планирование применения кластеров.

3. **Настройте среду для работы кластера.**

При настройке сверяйтесь с таблицей “Контрольная таблица настройки кластера” на стр. 93.

4. **Настройте кластер.**

Понятия, связанные с данным

“Сервисные центры по поддержке кластеров” на стр. 152

В этом разделе приведена контактная информация сервисного центра IBM.

Создание кластера

Процесс создания и настройки кластера начинается с настройки первого узла.

Для настройки остальных узлов кластера этот узел должен быть доступен администратору. Полный список требований для создания кластера приведен в разделе “Контрольная таблица настройки кластера” на стр. 93.

Для применения расширенных устройств должен быть выполнен ряд дополнительных условий. В этом случае следует уделить особое внимание предотвращению возможных конфликтов. Инструкции по настройке переносимых устройств приведены в разделе Создание переносимого независимого ASP.

Работа с Навигатором iSeries

Для выполнения этой операции должен быть установлен и зарегистрирован Компонент 41 (HA Switchable Resources).

В разделе Навигатор iSeries - Управление кластером приведено описание мастера, упрощающего создание простого кластера, состоящего из одного или двух узлов. После создания такого кластера к нему можно добавлять дополнительные узлы. Навигатор iSeries поддерживает кластеры, в которые может входить до четырех узлов. Мастер предложит вам выбрать серверы для кластера и создать группы ресурсов. Система, с которой создается простой кластер, должна стать одним из узлов кластера.

Проще всего создать кластер с помощью мастера создания кластера продукта Навигатор iSeries:

1. Запустите Навигатор iSeries и откройте **Централизованное управление**.
2. Щелкните правой кнопкой на значке **Кластеры** и выберите пункт **Создать кластер**.
3. Следуйте инструкциям мастера.

После создания кластера нужно выполнить следующие действия:

1. Добавить в кластер все необходимые узлы. С помощью Навигатора iSeries можно включить в кластер до 4 узлов.
2. Добавить нужные узлы в домены восстановления (если применяется переносимое аппаратное обеспечение или независимые ASP).
3. Создать и запустить переносимые ресурсы (переносимые устройства, приложения и данные).

В электронной справке Навигатора iSeries приведены пошаговые инструкции по выполнению этих задач.

С помощью команд CL и API

Кластер можно также создать с помощью команд CL и API.

1. **Создайте кластер.**

Команда Создать кластер (CRTCLU)
API Создать кластер (QcstCreateCluster)

2. **Добавьте необходимые узлы в кластер с первого активного узла.**

Команда Добавить запись узла кластера (ADDCLUNODE)
API Добавить запись узла кластера (QcstAddClusterNodeEntry)

3. **Запустите узел кластера.**

Команда Запустить узел кластера (STRCLUNOD)
API Запустить узел кластера (QcstStartClusterNode)

4. **Определите домены устройств.** Если вы планируете применять переносимые устройства, сначала нужно создать домен устройств.

Команда Добавить запись домена устройств (ADDDEVDMNE)
API Добавить запись домена устройств (QcstAddDeviceDomainEntry)

5. **Создайте группы ресурсов кластера (CRG).**

Команда Создать группу ресурсов кластера (CRTCRG)
API Создать группу ресурсов кластера (QcstCreateClusterResourceGroup)

6. **Запустите группы ресурсов кластера (CRG).**

Команда Запустить группу ресурсов кластера (STRCRG)
API Запустить группу ресурсов кластера (QcstStartClusterResourceGroup)

Управление кластерами

В этом разделе приведены инструкции по выполнению ряда операций над кластером.

Если вы еще не выбрали интерфейс, который будет применяться для работы с кластером, сначала ознакомьтесь с разделом Способы управления кластером.

После настройки кластера в нем чаще всего выполняются следующие операции:

Операции над кластером

- Добавление узла в кластер
- Удаление узла из кластера
- Запуск узла кластера
- Завершение работы узла кластера
- Коррекция версии кластера
- Удалить кластер
- Изменение узла кластера

Операции над группами ресурсов кластера

- Создание новой CRG
- Удаление CRG
- Запуск CRG
- Добавление узла в группу ресурсов кластера
- Удаление узла из группы ресурсов кластера
- Завершение работы CRG
- Изменение домена восстановления для CRG
- Принудительный перенос ресурсов
- Добавление узла в домен устройств
- Удаление узла из домена устройств

В этом разделе также приведены инструкции по сохранению конфигурации кластера. Рекомендуем также ознакомиться со структурой заданий службы ресурсов кластера и узнать о применении пользовательских очередей в API. Узнайте о том, как правильно завершать работу заданий кластера и контролировать состояние кластера. Вам наверняка пригодятся знания о функции передачи надежных сообщений и о том, как можно контролировать пульс кластера.

Операции над доменом управления кластером

- Создание домена управления кластером
- Добавление отслеживаемых ресурсов
- Удаление домена управления кластером

Понятия, связанные с данным

“Надежные сообщения” на стр. 27

Функция *надежных сообщений* службы ресурсов кластера отвечает за то, чтобы на всех узлах кластера была актуальная информация о состоянии ресурсов кластера.

“Контроль пульса” на стр. 26

Контроль пульса - это функция службы ресурсов кластера, проверяющая работоспособность узлов путем отправки регулярных сигналов с каждого узла кластера всем остальным узлам кластера.

Добавление узла в кластер

Добавить узел в кластер можно с помощью Навигатора iSeries или соответствующих команд.

Работа с Навигатором iSeries

Для выполнения этой операции должен быть установлен компонент 41 (HA Switchable Resources).

С помощью Навигатора iSeries можно создавать простые кластеры числом до четырех узлов. Если в кластере уже есть четыре узла, то опция **Добавить узел** отключается. Для создания кластеров числом более 4 узлов нужно пользоваться командами и API для работы с кластерами или соответствующими промежуточными продуктами партнеров фирмы IBM - в этом случае можно создавать кластеры числом до 128 узлов.

Для добавления узла в кластер выполните следующие действия:

1. В Навигаторе iSeries откройте Централизованное управление.
2. Откройте **Кластеры**.
3. Откройте кластер, в который требуется добавить узел.
4. Щелкните правой кнопкой на значке **Узлы** и выберите опцию **Добавить узел...**

Работа с командами кластера и с API

Для добавления узла в кластер можно воспользоваться следующими средствами:

- Команда **Добавить запись узла кластера (ADDCLUNODE)**
- API **Добавить запись узла кластера (QcstAddClusterNodeEntry)**

Понятия, связанные с данным

“Команды и API для работы с кластерами” на стр. 75

В состав службы ресурсов кластера i5/OS входит набор команд языка CL и соответствующих API, а также средств, которые могут применяться в прикладных программах iSeries.

“Партнеры IBM по разработке кластерных приложений и кластерные продукты” на стр. 81

Партнеры фирмы IBM по разработке кластерных продуктов предлагают линейку продуктов для управления кластерами, работы с ними, логической репликации данных и выполнения прочих подобных операций.

Запуск узла кластера

Операция запуска узла кластера заключается в запуске службы ресурсов кластера на узле. Начиная с версии 3, узел может самостоятельно запустить службу ресурсов кластера и войти в кластер в случае, если ему удастся найти хотя бы один узел, уже входящий в кластер.

Работа с Навигатором iSeries

Для выполнения этой операции должен быть установлен компонент 41 (HA Switchable Resources).

В результате запуска службы ресурсов кластера состояние узла должно измениться на *Работает*.

Для запуска узла выполните следующие действия:

1. Запустите Навигатор iSeries и откройте **Централизованное управление**.
2. Откройте **Кластеры**.
3. Откройте кластер, содержащий запускаемый узел.
4. Щелкните на значке **Узлы**.
5. Щелкните правой кнопкой мыши на нужном узле и выберите **Кластер → Запустить**.

Кластер

Работа с командами и API командной строки

С помощью команд CL и API также можно запустить узел кластера. В результате запуска службы ресурсов кластера состояние узла должно измениться на *Работает*.

- Команда Запустить узел кластера (STRCLUNOD)
- API Запустить узел кластера (QcstStartClusterNode)

Задачи, связанные с данной

“Завершение работы заданий кластера” на стр. 114

Никогда не пытайтесь напрямую завершить задание кластера.

“Устранение последствий сбоев в заданиях кластера” на стр. 143

Аварийное завершение задания службы ресурсов кластера, как правило, свидетельствует о какой-то внешней неполадке.

| Завершение работы узла кластера

| Запуск и завершение работы узла кластера сводятся к запуску и завершению работы службы ресурсов кластера на узле.

| Работа с Навигатором iSeries

| Для выполнения этой операции должен быть установлен компонент 41 (HA Switchable Resources).

| В результате завершения работы службы ресурсов кластера состояние узла должно измениться на *Остановлен*.

| Для завершения работы узла выполните следующие действия:

- | 1. Запустите Навигатор iSeries и откройте **Централизованное управление**.
- | 2. Откройте **Кластеры**.
- | 3. Откройте кластер с узлом, работу которого требуется завершить.
- | 4. Щелкните на значке **Узлы**.
- | 5. Щелкните правой кнопкой мыши на нужном узле и выберите **Кластер → Остановить**.

| Работа с командами и API командной строки

| С помощью команд CL и API также можно завершить работу узла. В результате завершения работы службы ресурсов кластера состояние узла должно измениться на *Не работает*.

- | • Команда Завершить работу узла кластера (ENDCLUNOD)
- | • API Завершить работу узла кластера (QcstEndClusterNode)

| **Задачи, связанные с данной**

| “Завершение работы заданий кластера” на стр. 114

| Никогда не пытайтесь напрямую завершить задание кластера.

| “Устранение последствий сбоев в заданиях кластера” на стр. 143

| Аварийное завершение задания службы ресурсов кластера, как правило, свидетельствует о какой-то внешней неполадке.

Коррекция версии кластера

Версия кластера - это версия протокола связи, применяемого узлами кластера для обмена данными.

Понятие версии необходимо для обеспечения полной совместимости серверов кластера, поскольку на разных серверах могут быть установлены разные уровни программного обеспечения.

Для коррекции версии кластера нужно, чтобы новая (последняя, потенциальная) версия кластера была установлена на всех узлах. После коррекции версии кластера все узлы будут применять последнюю версию протокола связи. Номер версии можно увеличивать только на единицу. Номер версии кластера нельзя уменьшить. Если вам требуется уменьшить версию кластера, нужно удалить кластер и создать его вновь, задав нужную версию. Текущая версия кластера первоначально задается первым узлом кластера. Остальные узлы можно будет добавлять в кластер только в случае, если они поддерживают эту версию.

Работа с Навигатором iSeries

Для выполнения этой операции должен быть установлен компонент 41 (HA Switchable Resources).

Для коррекции версии кластера выполните следующие действия:

1. Запустите Навигатор iSeries и откройте **Централизованное управление**.
2. Откройте **Кластеры**.
3. Щелкните правой кнопкой на кластере и выберите опцию **Свойства**.
4. Укажите нужное значение в поле Версия кластера.

Работа с командами кластера и с API

Для коррекции версии кластера можно воспользоваться следующими средствами:

- Команда Изменить версию кластера (CHGCLUVER)
- API Скорректировать версию кластера (QcstAdjustClusterVersion)

Понятия, связанные с данным

“Версия кластера” на стр. 13

Версией кластера называется идентификатор набора функций, которые может выполнять кластер.

“Наиболее частые неполадки кластеров” на стр. 137

В этом разделе перечислены наиболее часто встречающиеся неполадки кластеров и обсуждаются способы их предотвращения и устранения их последствий.

Задачи, связанные с данной

“Удалить кластер” на стр. 105

При удалении кластера во всех активных узлах кластера прекращается работа службы ресурсов кластера, все узлы удаляются из кластера.

Удалить кластер

При удалении кластера во всех активных узлах кластера прекращается работа службы ресурсов кластера, все узлы удаляются из кластера.

- | **Важное замечание:** Если в кластере есть независимые ASP, то перед удалением кластера с помощью команды Удалить запись домена устройств (RMVDEVDMNE) следует удалить все узлы из домена устройств.

Работа с Навигатором iSeries

Для выполнения этой операции должен быть установлен компонент 41 (HA Switchable Resources).

Для удаления кластера выполните следующие действия:

1. Запустите Навигатор iSeries и откройте **Централизованное управление**.
2. Откройте **Кластеры**.
3. Щелкните правой кнопкой мыши на кластере, который необходимо удалить, и выберите **Удалить**.

Работа с командами и API командной строки

С помощью команд CL и API также можно удалить кластер.

- | • Команда Удалить кластер (DLTCLU)
- | • API Удалить кластер (QcstDeleteCluster)

Задачи, связанные с данной

“Коррекция версии кластера” на стр. 104

Версия кластера - это версия протокола связи, применяемого узлами кластера для обмена данными.

Создание CRG

- | Группы ресурсов кластера бывают нескольких типов: CRG приложения, данных, устройств и равноправная CRG.

| Для создания CRG в кластере выполните следующие действия:

- | 1. В Навигаторе iSeries откройте **Централизованное управление** → **Кластеры**.
- | 2. Откройте кластер, в котором вы будете создавать CRG.
 - | a. Если требуется создать CRG устройств, щелкните правой кнопкой мыши на пункте **Переносимое аппаратное обеспечение** и выберите **Создать группу**. Примечание: Опция **Создать группу** доступна только если запущены все узлы домена восстановления. Дополнительные сведения приведены в разделе **Запуск узла кластера**.
 - | b. Если требуется создать CRG приложений, щелкните правой кнопкой мыши на пункте **Переносимое программное обеспечение** и выберите **Добавить продукт**.
 - | c. Если требуется создать CRG данных, щелкните правой кнопкой мыши на пункте **Переносимые данные** и выберите **Создать группу**.
 - | d. Если требуется создать равноправную CRG, щелкните правой кнопкой мыши на пункте **Равноправные ресурсы** и выберите **Создать равноправную CRG**.

Работа с командами и API командной строки

| Для создания CRG можно воспользоваться следующими командами и API:

- | • Команда Создать CRG (CRTCRG)
- | • API QcstCreateClusterResourceGroup (Создать группу ресурсов кластера)

Создание CRG приложения с активным устойчивым IP-адресом

При создании CRG приложения можно разрешить использование активного устойчивого IP-адреса. Однако это допустимо только в случае, если этот IP-адрес настроен пользователем.

Раньше можно было создать CRG приложения с активным устойчивым IP-адресом, если этот адрес был настроен пользователем. Но если этот адрес уже был активен, то CRG приложения не запускалась. Теперь же при создании CRG приложения появилась возможность указывать активный устойчивый IP-адрес. Запуск CRG приложения с поддержкой активных устойчивых IP-адресов будет успешным.

Для создания CRG приложения с поддержкой активного устойчивого IP-адреса выполните следующие действия:

1. В командной строке введите:

```
CRTCRG CLUSTER(MYCLUSTER) CRG(MYCRG) CRGTYPE(*APP) EXITPGM(QDEVELOP/EXITPGM)
USRPRF(USER) RCDYDMN((NODE1 *PRIMARY) (NODE2 *BACKUP)) TKVINTNETA('10.1.2.1') CFGINTNETA(*USR *YES)
```

Параметр **TKVINTNETA** задает устойчивый IP-адрес, **CFGINTNETA** говорит о том, что этот IP-адрес будет настроен пользователем и активирован при запуске CRG.

После создания CRG приложения с поддержкой активного устойчивого IP-адреса можно запустить CRG.

Запуск CRG

Вы можете запустить CRG нескольких типов: CRG приложений, данных, устройств и равноправную.

Для запуска CRG выполните следующие действия:

1. В Навигаторе iSeries откройте **Централизованное управление** → **Кластеры**.

2. Откройте кластер, в котором требуется запустить CRG.

a. Если необходимо запустить CRG устройств, выберите **Переносимое аппаратное обеспечение**, щелкните правой кнопкой мыши на нужном устройстве и выберите **Запустить**.

b. Если необходимо запустить CRG приложений, выберите **Переносимое программное обеспечение**, щелкните правой кнопкой мыши на нужном программном продукте и выберите **Запустить**.

c. Если необходимо запустить CRG данных, выберите **Переносимые данные**, щелкните правой кнопкой мыши на нужной группе переносимых данных и выберите **Запустить**.

d. Если необходимо запустить CRG данных, выберите **Равноправные ресурсы**, щелкните правой кнопкой мыши на нужной равноправной CRG в появившемся списке и выберите **Запустить**.

Работа с командами и API командной строки

Для запуска CRG можно воспользоваться следующими командами и API:

- Команда Запустить CRG (STRCRG)
- API Запустить CRG (QcstStartClusterResourceGroup)

Изменение домена восстановления для CRG

Вы можете произвольным образом менять состав домена восстановления, а также роли узлов в нем. Для группы ресурсов кластера устройств вы можете изменить место расположения и IP-адреса портов данных узла домена восстановления.

Работа с Навигатором iSeries

Для выполнения этой операции должен быть установлен компонент 41 (HA Switchable Resources).

Для изменения роли узлов в домене восстановления CRG (независимо от типа CRG) или для изменения состава домена восстановления выполните следующие действия:

1. Запустите Навигатор iSeries и откройте **Централизованное управление**.

2. Откройте **Кластеры**.
3. Откройте кластер с переносимым аппаратным обеспечением, приложениями или данными, для которого требуется изменить домен восстановления.
4. Откройте список CRG нужного типа (переносимое аппаратное обеспечение, приложения или данные).
5. Щелкните правой кнопкой на нужном ресурсе и выберите опцию **Свойства**.
6. Перейдите на страницу **Домен восстановления**.

Если вам нужны инструкции по изменению роли узлов или состава домена восстановления, нажмите кнопку Справка.

Работа с командами и API командной строки

Роль узлов и состав домена восстановления можно изменять с помощью следующих команд CL и API:

- Команда Добавить запись узла группы ресурсов кластера (ADDCRGNODE)
- API Добавить узел в домен восстановления (QcstAddNodeToRcvyDomain)
- Команда Изменить группу ресурсов кластера (CHGCRG)
- API QcstChangeClusterResourceGroup (Изменить группу ресурсов кластера)
- Команда Удалить запись узла группы ресурсов кластера (RMVCRGNODE)
- API Удалить узел из домена восстановления (QcstRemoveNodeFromRcvyDomain)

Понятия, связанные с данным

“Домен восстановления” на стр. 11

Домен восстановления - это множество узлов кластера, объединенных в CRG для выполнения определенной задачи - например, для обеспечения бесперебойного доступа к ресурсу или для синхронизации.

Принудительный перенос ресурсов

В результате принудительного переноса ресурсов текущий главный узел становится резервным узлом домена восстановления.

Роли узлов изменяются следующим образом:

- Текущий главный узел становится последним резервным узлом.
- Текущий первый резервный узел становится главным узлом.
- Остальные резервные узлы сдвигаются в списке на одну позицию вверх.

Принудительный перенос можно выполнять только для иерархических CRG, находящихся в состоянии 'активна'.

Примечание: При принудительном переносе CRG устройств по соображениям производительности рекомендуется синхронизировать имена пользовательских профайлов, UID и GID.

Работа с Навигатором iSeries

Для выполнения этой операции должен быть установлен компонент 41 (HA Switchable Resources).

Принудительный перенос можно выполнить для ресурсов, находящихся в состоянии **Работает**.

Для принудительного переноса ресурса выполните следующие действия:

1. Запустите Навигатор iSeries и откройте **Централизованное управление**.
2. Откройте **Кластеры**.
3. Откройте нужный кластер.

4. Щелкните на значке **Переносимое аппаратное обеспечение, Переносимое программное обеспечение** или **Переносимые данные**.
5. Щелкните правой кнопкой на ресурсе и выберите **Перенести**.

С помощью API для работы с кластерами

Для принудительного переноса ресурсов можно воспользоваться следующими командами и API:

- Команда Изменить главный узел CRG (CHGCRGPRI)
- API Выполнить принудительный перенос ресурсов (QcstInitiateSwitchOver)

Понятия, связанные с данным

“Домен восстановления” на стр. 11

Домен восстановления - это множество узлов кластера, объединенных в CRG для выполнения определенной задачи - например, для обеспечения бесперебойного доступа к ресурсу или для синхронизации.

Задачи, связанные с данной

“Принудительный перенос ресурсов” на стр. 21

Принудительный перенос ресурсов выполняется в случае перемещения пользователем точки доступа к ресурсу на другой сервер.

Синхронизировать имя пользовательского профайла, UID и GID

Добавление узла в домен устройств

Домен устройств - множество узлов кластера, совместно использующих ресурсы устройств.

Перед тем как добавить узел в домен восстановления CRG устройства, его нужно включить в домен устройств. Все узлы, входящие в домен восстановления CRG устройства, должны входить в один и тот же домен устройств. Каждый узел кластера может входить только в один домен устройств.

Для работы с доменами устройств на всех узлах кластера, которые будут входить в домен, должны быть установлены компонент 41 - HA Switchable Resources и ключи лицензий.

Работа с Навигатором iSeries

Для добавления узла в домен устройств в Навигаторе iSeries выполните следующие действия:

1. Запустите Навигатор iSeries и откройте **Централизованное управление**.
2. Откройте **Кластеры**.
3. Откройте кластер, содержащий добавляемый узел.
4. Щелкните на значке **Узлы**.
5. Щелкните правой кнопкой мыши на добавляемом узле и выберите **Свойства**.
6. В поле **Домен устройств** на странице **Кластеры** укажите имя домена устройств.

Работа с командами и API командной строки

Для добавления узла в домен устройств можно воспользоваться следующими средствами:

- Команда Добавить запись домена устройств (ADDDEVDMNE)
- API Добавить запись домена устройств (QcstAddDeviceDomainEntry)

Понятия, связанные с данным

“Домены устройств” на стр. 16

Домен устройств - множество узлов кластера, совместно использующих ресурсы устройств. Узлы, входящие в домен устройств, совместно выполняют операции переноса точки доступа к набору устойчивых устройств.

Задачи, связанные с данной

“Удаление узла из домена устройств”

Домен устройств - множество узлов кластера, совместно использующих ресурсы устройств.

Удаление узла из домена устройств

Домен устройств - множество узлов кластера, совместно использующих ресурсы устройств.

Важное замечание:

К удалению узлов из домена устройств следует относиться с предельным вниманием. Если по ошибке из домена устройств будет удален узел, выполняющий функции главной точки доступа, то домен устройств останется без главной точки доступа. Это означает, что все устройства домена станут недоступны.

После того как узел будет удален из домена устройств, его нельзя будет добавить обратно, если в этом домене устройств есть хотя бы один узел кластера. Для того чтобы добавить узел обратно в домен устройств, нужно выполнить следующие операции:

1. Удалить независимые ASP, принадлежащие узлу.
2. Перезапустить узел (выполнить IPL).
3. Добавить узел в домен устройств. См. раздел Добавление узла в домен устройств.
4. Вновь создать ASP, удаленный на шаге 1.

Работа с Навигатором iSeries

Для выполнения этой операции должен быть установлен компонент 41 (HA Switchable Resources).

Для удаления узла из домена устройств в Навигаторе iSeries выполните следующие действия:

1. Запустите Навигатор iSeries и откройте **Централизованное управление**.
2. Откройте **Кластеры**.
3. Откройте кластер, в который входит удаляемый узел.
4. Щелкните на значке **Узлы**.
5. Щелкните правой кнопкой на нужном узле и выберите **Свойства**.
6. На странице Кластеры удалите отметку из поля **Домен устройств**.

Работа с командами и API командной строки

Для удаления узла из домена устройств можно воспользоваться следующими средствами:

- Команда Удалить запись домена устройств (RMVDEVDMNE)
- API Удалить запись домена устройств (QcstRemoveDeviceDomainEntry)

Понятия, связанные с данным

“Домены устройств” на стр. 16

Домен устройств - множество узлов кластера, совместно использующих ресурсы устройств. Узлы, входящие в домен устройств, совместно выполняют операции переноса точки доступа к набору устойчивых устройств.

Задачи, связанные с данной

“Добавление узла в домен устройств” на стр. 108

Домен устройств - множество узлов кластера, совместно использующих ресурсы устройств.

Добавление накопителя или дискового пула

Влияние системных событий на кластер

Некоторые системные завершающие команды, как то: Выключить систему (PWRDWNSYS), Завершить работу системы (ENDSYS) и Завершить работу подсистемы (ENDSBS) могут внезапно завершить работу кластера и вызвать его распад.

В версии V5R4 команды PWRDWNSYS, ENDSYS и ENDSBS расширены. Если на узле работает служба ресурсов кластера, то при выполнении на нем этих команд вызывается API Завершить работу узла кластера (QcstEndClusterNode).

Если вам необходимо выполнить эти команды, то рекомендуется воспользоваться опцией OPTION(*CNTRLD) и указать в параметре DELAY необходимое время задержки. В противном случае API Завершить работу узла кластера может не успеть вызваться до того, как управление будет передано в системную функцию завершения.

Примечание: Если пользователь указывает OPTION(*IMMED), то API QcstEndClusterNode дается около 30 секунд на выполнение, после чего система завершает работу. В итоге вместо завершения узла кластера может начаться автоматический перенос ресурсов.

Создание домена управления кластером

Домен управления кластером можно создать с помощью Навигатора iSeries или команды Создать домен управления кластером (CRTADMDMN).

Для создания домена управления кластером и его администрирования пользователь должен иметь права доступа к создаваемой CRG, командам CRG и пользовательскому профайлу QCLUSTER.

Работа с Навигатором iSeries

Для создания домена управления кластером выполните следующие действия:

1. В Навигаторе iSeries откройте **Централизованное управление → Кластеры**.
2. Откройте кластер, для которого требуется создать домен управления.
3. Щелкните правой кнопкой мыши на пункте **Равноправные ресурсы** и выберите **Создать домен управления**.

Работа с командами и API командной строки

Для создания домена управления кластером можно воспользоваться следующими командами и API:

- Команда Создать домен управления кластером (CRTADMDMN)
- API для создания домена управления кластером не поддерживается.

Понятия, связанные с данным

“Команды и API для работы с кластерами” на стр. 75

В состав службы ресурсов кластера i5/OS входит набор команд языка CL и соответствующих API, а также средств, которые могут применяться в прикладных программах iSeries.

Добавление записей отслеживаемых ресурсов

В домен управления кластером можно добавить записи отслеживаемых ресурсов, общих для всех узлов.

Для добавления записи отслеживаемого ресурса выполните следующие действия:

1. В Навигаторе iSeries откройте **Централизованное управление → Кластеры**.
2. Откройте кластер, в который требуется добавить запись отслеживаемого ресурса.
3. Откройте **Равноправные ресурсы**. Появится список всех равноправных ресурсов кластера.
4. Откройте домен управления кластером, в который требуется добавить запись отслеживаемого ресурса.

5. Щелкните правой кнопкой мыши на типе отслеживаемого ресурса и выберите **Добавить запись отслеживаемого ресурса**.
6. Выберите атрибуты записи, которые требуется отслеживать, и нажмите **ОК**.

Примечание: При добавлении пользовательских профайлов, использующих в качестве записей отслеживаемого ресурса синхронизацию пароля, установите значение параметра Сохранять идентификационные данные на сервере (QRETSVRSEC) равным 1.

Работа с командами и API командной строки

Для добавления отслеживаемых ресурсов можно применять следующие команды и API:

- Для этой функции эквиваленты команд CL не поддерживаются. В библиотеке QUSRTOOL предоставлен исходный код неподдерживаемых команд и вызова программы обработки. Узнать источник и CPP данной команды можно из элемента QFPADINFO файла QATTINFO.
- API Добавить запись отслеживаемого ресурса (QfpadAddMonitoredResourceEntry)

Мониторинг домена управления кластером

После создания домена управления кластером и добавления записей об отслеживаемых ресурсах администратору кластера необходимо какое-то время понаблюдать за работой этого домена, чтобы убедиться, что ресурсы в нем остаются согласованными.

Если общее состояние противоречит состоянию отслеживаемого ресурса, то администратор должен принять необходимые меры для определения и устранения причины ситуации, а потом повторно синхронизировать ресурс.

Если несогласованность возникла вследствие сбоя обновления на одном или нескольких узлах, то определить причину сбоя можно по данным, хранящимся для MRE. В журнал сбойного узла заносится сообщение, содержащее запись MRE сбойного ресурса. На других узлах будут сообщения, свидетельствующие о неполадке, и перечень узлов, на которых не выполнено обновление.

Как только причина несогласованности будет определена, ресурс можно синхронизировать снова. Это можно сделать либо путем обновления "виноватого" узла, либо путем перезапуска домена управления.

Если ресурс был удален, переименован или перемещен на другой узел домена, общее состояние отслеживаемого ресурса всегда будет противоречивым. В такой ситуации необходимо удалить запись MRE, поскольку домену управления кластером больше не удастся синхронизировать этот ресурс.

Работа с Навигатором iSeries

Для наблюдения за доменом управления кластером выполните следующие действия:

1. В Навигаторе iSeries откройте **Централизованное управление → Кластеры**.
2. Откройте кластер, с которым связан домен управления.
3. Откройте **Равноправные ресурсы** и щелкните правой кнопкой мыши на **Создать домен управления**. Затем выберите **Свойства**. Состояние ресурса в активном домене управления кластером может принимать следующие значения:

Согласован

Значения всех атрибутов ресурса, отслеживаемых системой, совпадают для всех активных узлов домена управления кластером.

Несогласован

Значения всех атрибутов ресурса, отслеживаемых системой, различны для всех активных узлов домена управления кластером. Возможно, домен управления неактивен.

Ожидание

Значения отслеживаемых атрибутов находятся в процессе синхронизации по домену управления кластером.

Добавлен

Запись ресурса добавлена в каталог отслеживаемых ресурсов в домене управления, но еще не синхронизирована.

Работа с командами и API командной строки

Для наблюдения за доменом управления кластером можно воспользоваться следующими командами и API:

- Для этой функции эквиваленты команд CL не поддерживаются. В библиотеке QUSRTOOL предоставлен исходный код неподдерживаемых команд и вызова программы обработки. Узнать источник и CPP данной команды можно из элемента QFPADINFO файла QATTINFO.
- API Показать сведения об отслеживаемых ресурсах (QfpadRtvMonitoredResourceInfo)

Контроль состояния кластера

Служба ресурсов кластера осуществляет базовый контроль состояния кластера и его компонентов с помощью функции надежных сообщений и контроля пульса.

Помимо этого, состояние компонентов кластера можно контролировать вручную.

Работа с Навигатором iSeries

Для выполнения этой операции должен быть установлен компонент 41 (HA Switchable Resources).

Для контроля состояния кластера в Навигаторе iSeries выполните следующие действия:

1. В Навигаторе iSeries откройте Централизованное управление.
2. Откройте **Кластеры**.
3. Откройте нужные ресурсы (кластер, узлы или отдельные ресурсы) в Навигаторе iSeries - их состояние будет указано в столбце Состояние. Описание значений в этом столбце можно найти в электронной справке. Кроме того, можно щелкнуть правой кнопкой мыши на значке любого компонента кластера и выбрать пункт **Свойства**.

Работа с командами и API командной строки

Следующие команды и API могут применяться для контроля состояния кластера:

Информация о кластере

Следующие команды и API позволяют получить различную информацию о кластере: списки узлов, IP-адреса, состояние узлов и т.п.

- Команда Показать информацию о кластере (DSPCLUINF)
- API Показать информацию о кластере (QcstListClusterInfo)
- API Показать сведения о домене устройств (QcstListDeviceDomainInfo)
- API Показать информацию службы ресурсов кластера (QcstRetrieveCRSInfo)
- API Загрузить информацию о кластере (QcstRetrieveClusterInfo)

Информация о группах ресурсов кластера

Следующие команды и API показывают сведения о группах ресурсов кластера - имена главных узлов CRG и т.п.

- Команда Показать информацию о CRG (DSPCRGINF)
- API Показать группы ресурсов кластера (QcstListClusterResourceGroups)
- API Показать информацию о CRG (QcstListClusterResourceGroupInf)

Понятия, связанные с данным

“Надежные сообщения” на стр. 27

Функция *надежных сообщений* службы ресурсов кластера отвечает за то, чтобы на всех узлах кластера была актуальная информация о состоянии ресурсов кластера.

“Контроль пульса” на стр. 26

Контроль пульса - это функция службы ресурсов кластера, проверяющая работоспособность узлов путем отправки регулярных сигналов с каждого узла кластера всем остальным узлам кластера.

Производительность кластера

Изменение кластера может повлиять на его производительность.

Для работы кластера ресурсы системы расходуются на контроль пульса, управление CRG и узлами кластера, а также обмен сообщениями между CRG и узлами кластера. После первоначальной настройки кластера снижение его производительности может произойти только в результате его изменения.

В обычных условиях кластер расходует несущественный объем ресурсов.

Понятия, связанные с данным

“Контроль пульса” на стр. 26

Контроль пульса - это функция службы ресурсов кластера, проверяющая работоспособность узлов путем отправки регулярных сигналов с каждого узла кластера всем остальным узлам кластера.

“Наиболее частые неполадки кластеров” на стр. 137

В этом разделе перечислены наиболее часто встречающиеся неполадки кластеров и обсуждаются способы их предотвращения и устранения их последствий.

Распределение нагрузки на сеть в кластере

Постарайтесь равномерно распределить нагрузку по линиям связи, соединяющим узлы кластера.

Чем более равномерного распределения вы добьетесь, тем выше в целом будет производительность и устойчивость сети.

Нагрузка на процессоры резервных узлов:

Используйте резервные системы по максимуму, но помните, что в случае переноса ресурсов нагрузка на эти системы может резко возрасти.

Если в вашей среде работают чрезвычайно важные приложения, для которых периодически выполняется процедура переноса, то следует позаботиться о запасе мощности процессоров на резервных системах.

Настройка производительности кластера

Для обеспечения максимальной эффективности служб кластера в различных рабочих средах предусмотрена возможность изменения параметров, влияющих на производительность кластера.

В большинстве случаев можно пользоваться значениями, предусмотренными по умолчанию. Если же значения по умолчанию не обеспечивают хорошую производительность в вашей среде, попробуйте изменить их. Предусмотрено два уровня настройки.

Базовая настройка

Базовая настройка позволяет выбрать один из трех предопределенных наборов значений для времени ожидания сообщения и частоты пульса. Если выбран набор 2, то для частоты пульса и времени ожидания применяются стандартные значения. Если выбран набор 1, то пульс становится более редким, время ожидания для сообщений повышается. В этом режиме кластеру потребуется больше времени на обнаружение сетевых неполадок. Если выбран набор 3, то пульс становится более частым, время ожидания

для сообщений сокращается. В этом режиме кластеру потребуется меньше времени на обнаружение сетевых неполадок.

Расширенная настройка

В режиме расширенной настройки значения параметров изменяются независимо друг от друга, и каждому параметру можно присвоить любое значение из допустимого диапазона. Это позволяет добиться максимального соответствия параметров производительности условиям конкретной среды. Прежде, чем приступить к расширенной настройке, рекомендуется проконсультироваться со специалистом из сервисного центра фирмы IBM. Неудачный выбор значений отдельных параметров может привести к резкому снижению производительности.

Понятия, связанные с данным

“Настраиваемые параметры связи кластера” на стр. 96

С помощью API QcstChgClusterResourceServices (Изменить службу ресурсов кластера) можно настраивать ряд параметров службы ресурсов кластера, влияющих на производительность, для достижения максимальной эффективности в конкретной среде. Этот API можно применять только для работы с кластерами версии 2 и выше.

Ссылки, связанные с данной

API Изменить службу ресурсов кластера (QcstChgClusterResourceServices)

Завершение работы заданий кластера

Никогда не пытайтесь напрямую завершить задание кластера.

Если потребуется прекратить все операции в среде кластера, нужно выполнить следующие действия:

1. Завершить работу узла кластера.
2. Устранить неполадку.
3. Запустить узел кластера.

Задачи, связанные с данной

“Завершение работы узла кластера” на стр. 103

Запуск и завершение работы узла кластера сводятся к запуску и завершению работы службы ресурсов кластера на узле.

“Запуск узла кластера” на стр. 103

Операция запуска узла кластера заключается в запуске службы ресурсов кластера на узле. Начиная с версии 3, узел может самостоятельно запустить службу ресурсов кластера и войти в кластер в случае, если ему удастся найти хотя бы один узел, уже входящий в кластер.

Подсистема контроля и управления ресурсами (RMC)

Подсистема контроля и управления ресурсами (RMC) - это обобщенная структура для управления ресурсами (например, физическими или логическими элементами системы) и наблюдения за ними.

RMC служит в качестве средства связи для передачи служебных событий в Консоль аппаратного обеспечения (HMC). В случае неактивности подсистемы RMC служебные события не будут передаваться в HMC. Службы, связанные с RMC, описываются в следующем списке:

Демон CAS

Назначение: Сервер идентификации для RMC.

Имя задания: QCSTCTCASD

Демон RMC

Назначение: Отслеживает ресурсы, обмениваясь данными с Администратором ресурсов.

Имя задания: QCSTCTRMCD

Демон SRC

Назначение: Отслеживает состояние других заданий RMC; перезапускает неожиданно завершившееся задание.

Имя задания: QCSTSRCD

Администраторы ресурсов (RM)

Администратор ресурсов (RM) - это задание, управляющее действительными физическими или логическими элементами, а также предоставляющее интерфейс для обмена данными между ними и RMC. Несмотря на то, что в RMC организованы основные абстракции (классы ресурсов, ресурсы, атрибуты для представления физических или логических элементов), на самом деле эта подсистема не представляет ресурсы. Администратор ресурсов преобразует фактические ресурсы в абстракции RMC. В следующем списке приведены администраторы ресурсов, поддерживаемые RMC:

RM протокола контроля

Назначение: Предоставляет средства для записи информации о работе системы.

Имя задания: QYUSALRMD

RM CSMAgent

Назначение: Предоставляет классы ресурсов для представления Сервера управления (HMC).

Имя задания: QYUSCMCRMD

RM хостов

Назначение: Предоставляет классы ресурсов для представления конкретной системы.

Имя задания: QCSTCTHRMD

RM служб

Назначение: Управляет информацией о неполадках и подготавливает ее к передаче в HMC.

Имя задания: QSVRMSERMD

Запуск и завершение работы RMC

Все задания RMC, в том числе и задания RM, относятся к подсистеме QSYSWRK и автоматически запускаются при запуске подсистемы. Для запуска необходимо, чтобы в системе была настроена поддержка TCP/IP. Включенный TCP/IP необходим для демона RMC. При отключении TCP/IP демон RMC завершает работу. Как только TCP/IP восстанавливается, демон RMC автоматически перезапускается демоном SRC. Никаких действий пользователя при этом не требуется. Если необходимо запустить RMC вручную, то выполните следующую команду:

```
SBMJOB CMD(CALL PGM(QSYS/QCSTCTSRCD)) JOBD(QSYS/QCSTSRCD) PRTDEV(*JOB) OUTQ(*JOB)
USER(*JOB) PRTTXT(*JOB) RTGDTA(RUNPTY50)
```

Если необходимо вручную завершить работу RMC, то завершите задание QCSTSRCD с помощью команды ENDJOB. Эта команда завершает все задания RMC. Если этого не происходит, то завершите вручную каждое задание из вышеприведенного списка.

Пользовательские очереди и структура заданий

При управлении кластером необходимо разбираться в структуре заданий и принципе работы пользовательских очередей.

Структура заданий службы ресурсов кластера

Служба ресурсов кластера состоит из нескольких многонитевых заданий. После запуска кластера на сервере под управлением профайла QSYS выполняются перечисленные ниже задания в подсистеме QSYSWRK. Эти задания используют описание QDFTSVR с единственным изменением: для них включен режим ведения протокола.

- Управление кластером выполняет задание QCSTCTL.
- CRG управляется заданием QCSTCRGM.
- Для каждого CRG выполняется отдельное задание. Имя этого задания совпадает с именем группы ресурсов. Это относится и к домену управления кластером.
- Для запуска и переноса устойчивых устройств, входящих в CRG устройств, запускаются дополнительные задания.

Важнейшие задания, обеспечивающие работу кластера, - это QCSTCTL и QCSTCRGM. Они необходимы для того, чтобы система могла выполнять функции узла кластера.

Большинство API, предназначенных для работы с CRG, выполняются в отдельных заданиях под управлением профайла, указанного при создании CRG. Программы выхода, связанные с CRG, выполняются в этом задании. По умолчанию задания ставятся в очередь QBATCH. Эта очередь используется для системных пакетных заданий и в ней установлен низкий приоритет для программ выхода. Для повышения эффективности API следует создать отдельный пользовательский профайл, описание задания и очередь для выполнения заданий, связанных с CRG. Этот пользовательский профайл следует указывать для всех CRG. Такой подход должен быть реализован на всех узлах, входящих в домен восстановления CRG.

- Для перезапуска аварийно завершенного задания группы ресурсов кластера, а также для запуска службы ресурсов кластера на узле можно воспользоваться командой Изменить восстановление кластера (CHGCLURCY).

Применение пользовательских очередей кластерными API

- API, в которых предусмотрен параметр Информация о результатах, выполняются в асинхронном режиме и направляют результаты в пользовательскую очередь. Эта очередь должна существовать на момент вызова API. Для создания пользовательских очередей применяется API Создать пользовательскую очередь (QUSCRTUQ). Очередь должна быть создана с ключом. Ключ пользовательской очереди должен быть описан в формате записи пользовательской очереди. В API при вызове передается имя очереди. Примеры применения пользовательских очередей кластерными API приведены в документации по кластерным API.

Если применяется API Разослать информацию (QcstDistributeInformation), то информация, которой обмениваются узлы, помещается в пользовательскую очередь, указанную при создании CRG. На момент вызова API Разослать информацию очередь с таким именем должна уже существовать на всех активных узлах домена восстановления. Дополнительная информация о требованиях к очереди для рассылки информации приведена в описании API Создать группу ресурсов кластера(QcstCreateClusterResourceGroup).

Сведения об автоматическом переносе заносятся в очередь сообщений автоматического переноса.

Понятия, связанные с данным

“Управление пользовательскими профайлами на узлах кластера” на стр. 92

Управлять пользовательскими профайлами на всех узлах кластера можно двумя способами.

“Проверка работоспособности кластера” на стр. 125

Устранение неполадок следует начать с диагностики.

Задачи, связанные с данной

“Устранение последствий сбоев в заданиях кластера” на стр. 143

Аварийное завершение задания службы ресурсов кластера, как правило, свидетельствует о какой-то внешней неполадке.

Очередь сообщений автоматического переноса ресурсов

Сведения об автоматическом переносе заносятся в очередь сообщений автоматического переноса.

Наблюдение за этой очередью может упростить работу администратора. Например, администратор может отменить намеченный автоматический перенос ресурсов, если он нежелателен в настоящий момент.

Очередь сообщений автоматического переноса ресурса создается вместе с группой ресурсов кластера с помощью API Создать кластер (QcstCreateCluster). Эту очередь можно изменять вместе с CRG с помощью соответствующих команд CL и API. Очередь сообщений автоматического переноса ресурсов недоступна через интерфейс Управления кластером Навигатора iSeries.

Дополнительные сведения об очереди сообщений автоматического переноса ресурсов приведены в документации по API групп ресурсов кластера. Инструкции по работе с этой очередью сообщений можно найти в разделах справки, указанных ниже.

Команды CL

- Команда CRTCRG (Создать группу ресурсов кластера)
- Команда CHGCRG (Изменить группу ресурсов кластера)

API

- API Создать группу ресурсов кластера (QcstCreateClusterResourceGroup)
- API Изменить группу ресурсов кластера (QcstChangeClusterResourceGroup)

Управление пользовательскими профайлами на узлах кластера

Управлять пользовательскими профайлами на всех узлах кластера можно двумя способами.

Первый способ заключается в создании домена управления кластером, отслеживающего общие ресурсы кластера. Кроме пользовательских профайлов, домен управления кластером позволяет отслеживать еще несколько типов общих ресурсов. Дополнительная информация приведена в разделе Отслеживаемые ресурсы. Если домен управления кластером активен, то при обновлении пользовательских профайлов изменения автоматически распространяются на остальные узлы. В противном случае изменения распространятся сразу после активизации домена управления кластером.

Примечание: Если требуется сделать общими пользовательские профайлы, синхронизирующие пароли в пределах кластера, то установите значение системного параметра Сохранять идентификационные данные на сервере (QRETSVRSEC) равным 1.

Второй способ работы с пользовательскими профайлами - посредством Централизованного управления в Навигаторе iSeries. Эта возможность позволяет выполнять операции над несколькими системами и группами систем. С помощью Централизованного управления можно выполнять стандартные задачи оператора одновременно на нескольких узлах кластера. В частности, с помощью Централизованного управления удобно синхронизировать пользовательские профайлы на нескольких системах. Администратор может создать специальную команду синхронизации и выполнять ее всякий раз при создании нового профайла на одном из узлов.

Резервное копирование и восстановление кластеров

Несмотря на то, что кластеры обладают высокой устойчивостью, ни в коем случае не следует пренебрегать резервным копированием.

Если кластер используется, в частности, для упрощения резервного копирования (на время сохранения данных с одной из систем работает другая, и наоборот), то настоятельно рекомендуется включить в кластер не менее трех систем. В этом случае при выходе системы из строя всегда будет наготове резервная система, даже если третья система в этот момент занята резервным копированием.

Сохранение и восстановление CRG

Группы ресурсов кластера можно сохранять как во время работы кластера, так и тогда, когда кластер выключен. Установлены следующие ограничения:

- Если кластер включен, то нельзя восстанавливать CRG, которые определены в кластере.

- Если узел не входит в кластер, на нем нельзя восстановить CRG.

Группу ресурсов кластера можно восстановить в случае, если кластер включен, группа ресурсов кластера не определена в кластере, узел входит в домен восстановления CRG и имя кластера, указанное в определении CRG, совпадает с именем текущего кластера. Кроме того, группу ресурсов кластера можно восстановить, если кластер настроен, но служба ресурсов кластера не запущена на данном узле, а узел при этом входит в домен восстановления CRG.

Подготовка к критическим ситуациям

Серьезные сбои могут привести к необходимости повторной настройки кластера. На случай возникновения такой ситуации рекомендуется сохранить информацию о конфигурации кластера и держать под рукой распечатку этой информации.

1. После каждого изменения конфигурации кластера сохраняйте ее с помощью команды Сохранить конфигурацию (SAVCFG) или Сохранить систему (SAVSYS). Инструкции по применению команд SAVCFG и SAVSYS приведены в разделе Сохранение информации о конфигурации.
2. После каждого изменения конфигурации кластера создавайте бумажную копию параметров конфигурации. Распечатать конфигурацию кластера можно с помощью команды Показать информацию о кластере (DSPCLUINF) command. Храните эту копию вместе с магнитными лентами, с которых в случае аварии будет восстанавливаться кластер.

Понятия, связанные с данным

“Восстановление кластера из резервной копии” на стр. 146

При нормальной работе восстановление с резервной копии обычно не требуется.

Сохранение данных конфигурации

“Восстановление кластера после полного выхода из строя” на стр. 145

Информация из этого раздела, а также руководство Резервное копирование и восстановление помогут вам восстановить систему после полного выхода из строя в результате непредвиденного отключения питания.

“Сохранение конфигурации кластера”

Для сохранения объектов группы ресурсов кластера можно воспользоваться командами.

“Восстановление кластера после стихийных бедствий” на стр. 145

Стихийные бедствия могут привести к уничтожению всех узлов кластера, и тогда вам придется полностью восстанавливать кластер.

Задачи, связанные с данной

Планирование стратегии резервного копирования и восстановления

Печать системной информации

Сохранение конфигурации кластера

Для сохранения объектов группы ресурсов кластера можно воспользоваться командами.

С помощью команды SAVSYS (Сохранить систему) можно полностью сохранить систему, а не только конфигурацию кластера. С помощью команды SAVCFG (Сохранить конфигурацию) можно сохранить конфигурацию всей системы.

```
SAVOBJ(QUSRSYS/*ALL) OBJTYPE (*CRG)
```

Примечание: Объекты CRG можно сохранять только для текущей версии операционной системы.

Задачи, связанные с данной

“Резервное копирование и восстановление кластеров” на стр. 117

Несмотря на то, что кластеры обладают высокой устойчивостью, ни в коем случае не следует пренебрегать резервным копированием.

Ссылки, связанные с данной

Команда SAVSYS (Сохранить систему)

Команда SAVCFG (Сохранить конфигурацию)

Примеры: Конфигурации кластеров

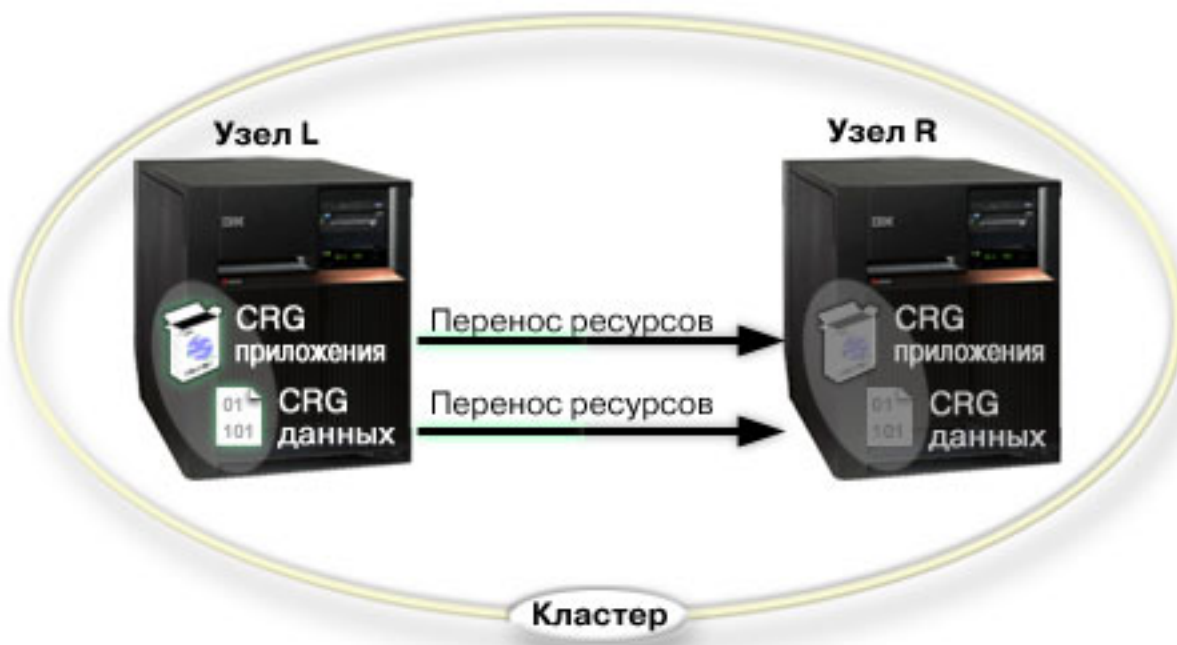
Примеры типичных конфигураций кластеров для разных производственных нужд.

Пример: Простой кластер из двух узлов

Пример конфигурации простого кластера из двух узлов.

Данный пример иллюстрирует следующие элементы:

- Одноуровневая репликация и перемещение ресурсов
- Двухуровневая среда
- Совместное перемещение приложений и данных
- Резервные узлы не получают нагрузки и задействуются только при сбое
- Бесперебойная работа в равноправной CRG



В данном примере узел L выполняет функции главной точки доступа к двум CRG: CRG приложений и CRG данных. На нем же находится и равноправная CRG, обеспечивающая бесперебойную работу обоих узлов. На узле L для CRG приложений регулярно вызываются две программы выхода. Эти программы выхода могут выполняться одновременно, поскольку первая программа выхода запускается при активации CRG с помощью API Запустить CRG и работает все время, пока работает CRG приложений. При завершении работы CRG приложений (API Завершить работу CRG) вызывается вторая программа выхода. Узел R выполняет функции резервной точки доступа для всех ресурсов кластера. Для данных и приложений, связанных с соответствующими CRG, выполняется репликация с узла L на узел R. Если узел L выйдет из строя или его потребуется отключить для профилактического обслуживания, то в результате переноса ресурсов узел R станет главной точкой доступа. Узлу R будет назначен IP-адрес, по которому выполняется обращение к CRG приложений.

Примечание: Пока узел L будет отключен, готовность ресурсов будет снижена, так как для узла R не будет резервного узла на случай сбоя. После возврата в кластер узел L станет резервной точкой

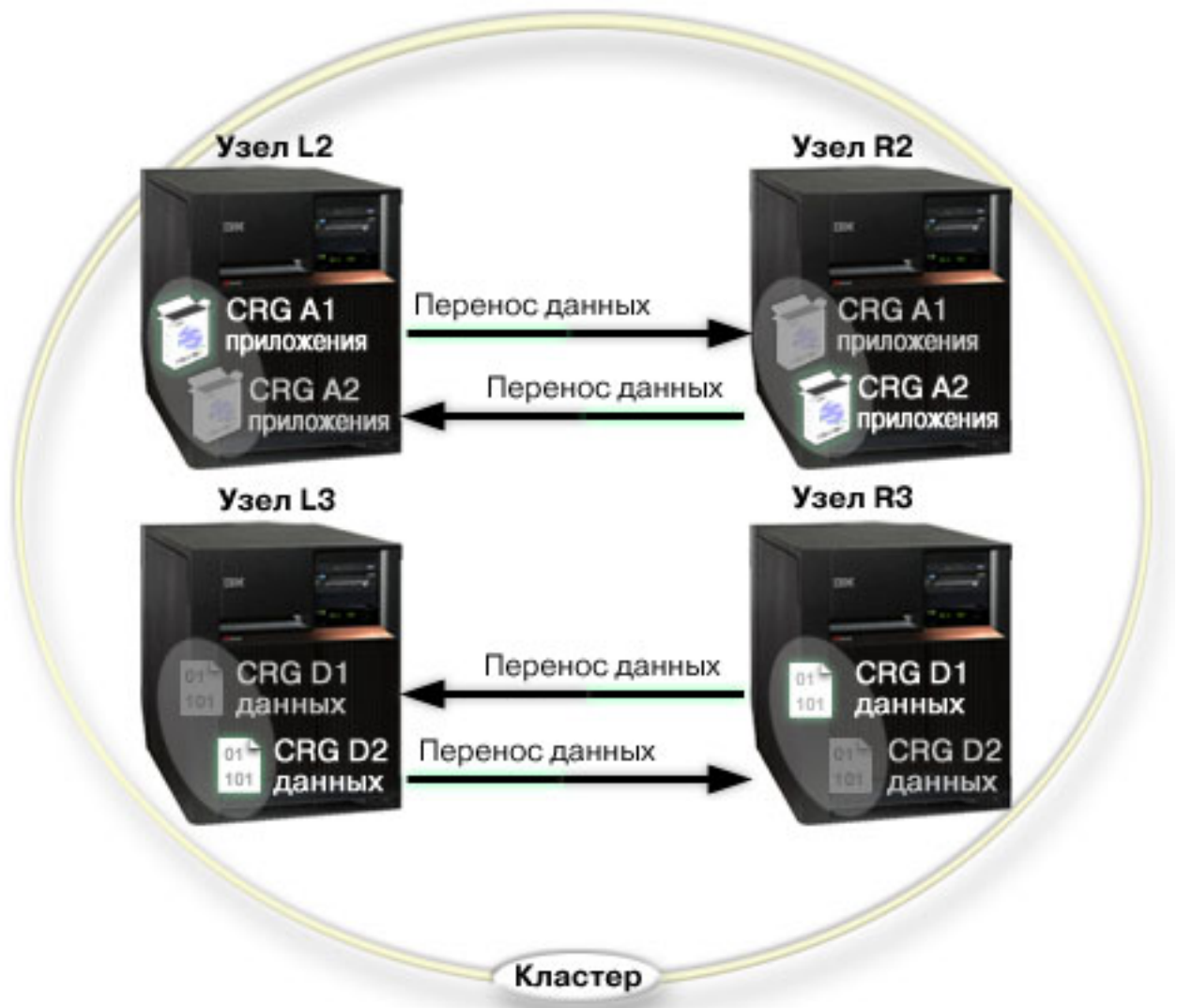
доступа для обоих CRG. С этого момента репликация будет выполняться с узла R на узел L. Если по каким-либо причинам потребуется, чтобы узел L вновь стал главной точкой доступа, то нужно будет выполнить принудительный перенос ресурсов на узел L.

Пример: Кластер из четырех узлов

В этом примере рассматривается создание более сложного кластера - из четырех узлов.

Данный пример иллюстрирует следующие элементы:

- Двухуровневая репликация и перенос ресурсов
- Трехуровневая среда
- Независимое перемещение приложений и данных
- Равномерное распределение нагрузки между главными и резервными узлами



Этот пример демонстрирует гибкость кластеров iSeries. В кластере определено два CRG приложений (A1 и A2) и два CRG данных (D1 и D2). Данные в CRG D1 необходимы для работы приложений из CRG A1. Данные из CRG D2 необходимы для работы приложений из CRG A2. В данном кластере применяется трехуровневая среда: приложения находятся на втором уровне (узлы L2 и R2), а данные - на третьем уровне (узлы L3 и R3).

Группа ресурсов кластера (CRG)	Главный	Резервный
CRG приложений A1	L2	R2
CRG приложений A2	R2	L2
CRG данных D1	R3	L3
CRG данных D2	L3	R3

Такая конфигурация при равномерном распределении нагрузки обеспечивает устойчивость приложений и данных. Все узлы кластера получают примерно равную нагрузку. Каждый из них выступает в роли резервной точки доступа. И приложения, и данные будут доступны в любой момент времени. Отказ любого узла не повлияет на готовность приложений и данных. Более того, даже одновременный отказ одного узла на уровне приложений и другого узла на уровне данных не повлияет на готовность.

Примечание: Данный кластер обладает следующим недостатком: прекращение работы узла приводит к прекращению репликации соответствующих ресурсов. Для устранения этого недостатка нужно организовать несколько резервных точек доступа для каждого ресурса.

Пример: Кластер из двух узлов с перемещаемыми дисками

Для обеспечения устойчивости данных вместо репликации можно пользоваться перемещаемыми дисками. В кластере с перемещаемыми дисками данные хранятся в независимых ASP.

Данный пример иллюстрирует следующие элементы:

- Один независимый ASP с простаивающим резервным сервером. Независимый ASP входит в группу перемещаемых дисков.
- Двухуровневая среда
- Совместное перемещение приложений и данных
- Резервный сервер выполняет нагрузку, не связанную с данными основного приложения
- Репликация не применяется, в кластере хранится только один экземпляр данных



В этом примере узлы L и R входят в один домен устройств. Узел L выполняет функции главной точки доступа к двум CRG: CRG приложений и CRG устройств. Узел R - первый резервный узел для обоих CRG. Данные, связанные с CRG устройств, хранятся в независимом ASP, расположенном во внешнем блоке расширения. Информация, необходимая для работы приложений из CRG приложений, либо хранится в том же блоке расширения, либо для нее выполняется репликация из узла L в узел R. Если узел L выйдет из строя или будет отключен для профилактического обслуживания, узел R станет главной точкой доступа к обоим CRG. Узлу R будет назначен IP-адрес, по которому выполняется обращение к CRG приложений. Узел R также станет владельцем перемещаемого ресурса, определенного в CRG устройств.

Примечание: Пока узел L будет отключен, готовность ресурсов будет снижена, так как для узла R не будет резервного узла на случай сбоя. После возврата в кластер узел L станет резервной точкой доступа для обоих CRG. Если потребуется, чтобы этот узел вновь стал главной точкой доступа, то нужно будет выполнить принудительный перенос ресурсов.

Понятия, связанные с данным

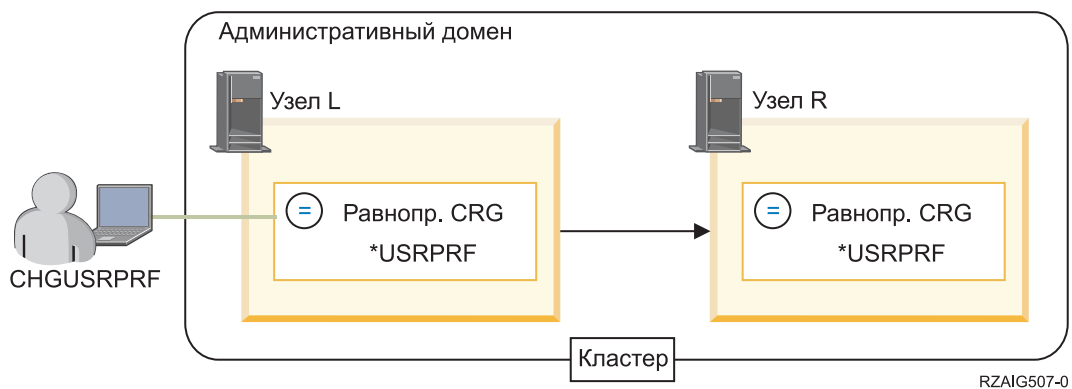
Настройка независимых пулов дисков

Пример: Управление равноправными ресурсами в домене управления кластером

В этом разделе приведен пример конфигурации домена управления кластером для отслеживания ресурсов кластера.

Данный пример иллюстрирует следующие элементы:

- Кластер из двух узлов
- Домен управления кластером, состоящим из двух узлов
- Запись отслеживаемого ресурса (MRE) для пользовательского профайла, синхронизируемая по домену.



В этом примере администратору необходимо, чтобы пользовательский профайл все время был согласованным по всему кластеру. Для отслеживания и синхронизации изменений профайла создается домен управления кластером. Домен управления кластером представляет собой равноправную CRG, состоящую из узлов L и R. В домен добавляется запись отслеживаемого ресурса для пользовательского профайла. В данном примере все атрибуты профайла уже заданы перед добавлением MRE. Следовательно, после запуска CRG изменение одного атрибута на узле L или R автоматически распространяется на активные узлы домена.

Дальше описана процедура настройки:

1. Создается кластер с узлами L и R.
2. На узлах L и R создается домен управления кластером
3. Добавляется MRE для пользовательского профайла
4. Запускается равноправная CRG, представляющая домен управления кластером
5. На узле L или R изменяется пользовательский профайл. Домен управления кластером автоматически изменяет профайл на втором узле. Если это изменение успешно, то глобальное состояние отслеживаемого ресурса будет согласованным.

Пример: Независимые ASP с географической зеркальной защитой

В примере показан вариант настройки географической зеркальной защиты. Узел А и Узел В находятся в Нью-Йорке. Узлы С и D - в Бостоне. Все четыре узла входят в один домен восстановления. Рабочую копию можно переносить между узлами А и В. Зеркальную копию можно переносить между узлами С и D. Так как все узлы находятся в одном домене восстановления, исходная система в Нью-Йорке и целевая система в Бостоне могут меняться ролями. Таким образом, рабочая копия может храниться и в Бостоне.



Роли узлов в домене восстановления следующие:

Узел	Роль
Узел А	Главный
Узел В	Резервный 1
Узел С	Резервный 2
Узел D	Резервный 3

В случае стихийного бедствия в Нью-Йорке узел С в Бостоне становится главным узлом, а его зеркальная копия - рабочей. Узел С становится для географической зеркальной защиты исходной системой, при этом защита приостанавливается, поскольку целевой системы больше нет из-за стихийного бедствия в Нью-Йорке. Когда в Нью-Йорке ликвидируют последствия стихийного бедствия и восстановят систему, Узел А становится резервным, а хранящаяся на нем рабочая копия - зеркальной.

Устранение неполадок

Процедуры устранения типичных неполадок, возникающих при работе с кластерами.

Кластеры, как и любые другие аппаратно-программные решения, подвержены неполадкам. В этом разделе приведена информация о том, какие неполадки могут возникать при работе с кластерами.

Проверка работоспособности кластера

Устранение неполадок следует начать с диагностики.

Кластеры, как и любые другие аппаратно-программные решения, подвержены неполадкам. Если у вас возникло впечатление, что кластер работает неправильно, то сначала следует проверить его работоспособность.

- **Определите, работает ли в системе служба ресурсов кластера.**

Для того чтобы определить, работает ли кластер, нужно проверить наличие заданий QCSTCTL и QCSTCRGM в подсистеме QSYSWRK. Если эти задания активны, то служба ресурсов кластера работает. Проверить наличие заданий можно с помощью функции управления заданиями Навигатора*raiSeries* (Посмотреть задания в подсистеме) или с помощью команды WRKACTJOB (Работа с активными заданиями) *command*. Кроме того, можно посмотреть текущее состояние кластера с помощью команды DSPCLUINF (Показать информацию о кластере).

- Помимо этих, кластер может использовать и другие задания. Например, Структура задания службы ресурсов кластера предоставляет информацию о формате заданий службы ресурсов кластера.

- **Попытайтесь найти сообщения об ошибках.**

- Просмотрите, нет ли в очереди QSYSOPR сообщений-вопросов, ожидающих ответа.

- Просмотрите, нет ли в очереди QSYSOPR сообщений об ошибках, связанных с кластером. Идентификаторы большинства таких сообщений лежат в диапазоне CPFBB00-CPFBBFF.

- Проверьте, нет ли сообщений о неполадках кластера в протоколе хронологии (команда CL DSPLOG). Идентификаторы большинства таких сообщений лежат в диапазоне CPFBB00-CPFBBFF.

- **Проверьте протоколы заданий кластера на наличие сообщений о серьезных ошибках.**

Обычно для этих заданий установлен уровень (4 0 *SECLVL) ведения протокола, а на этом уровне в протокол заносятся все сообщения об ошибках. Проверьте, правильно ли задан уровень ведения протокола для этих заданий кластера и заданий программ выхода. Даже если кластер не запущен, вы можете просмотреть буферные файлы заданий кластера и заданий программ выхода.

- **Если кластер не отвечает на запросы, проанализируйте стеки вызовов заданий кластера.**

Посмотрите, есть ли в очередях программы вида DEQW (ожидание освобождения очереди). Если такие программы есть, просмотрите все нити этих программ и выясните, есть ли у каких-либо из них *getSpecialMsg* в стеке вызова.

- **Проверьте содержимое протоколов на наличие записей вертикального лицензионного внутреннего кода (VLIC).**

Это должны быть записи с главным кодом 4800.

- **С помощью команды NETSTAT определите состояние среды связи.**

Команда NETSTAT позволяет получить сведения о состоянии маршрутов TCP/IP, интерфейсов, соединений TCP и портов UDP.

- С помощью опции 1 команды NETSTAT (Работа с состоянием интерфейсов TCP/IP) убедитесь в том, что IP-адреса, выделенные кластеру, находятся в состоянии 'Активен'. Убедитесь, что активен циклический адрес (127.0.0.1).

- С помощью опции 3 команды NETSTAT (Работа с состоянием соединений TCP/IP) просмотрите номера портов (F14). Локальный порт 5550 должен находиться в состоянии 'Прослушивание'. Если в списке активных заданий есть задание QTOGINTD (пользователь QTCP), то этот порт должен был быть открыт командой STRTCPSVR *INETD. Если на узле запущена служба ресурсов кластера, то

локальный порт 5551 должен быть открыт и находиться в состоянии '*UDP'. Если служба ресурсов кластера не запущена, порт 5551 должен быть закрыт (если он будет открыт, то вам не удастся запустить на этом узле службу ресурсов кластера).

- Воспользуйтесь командой ping. Если вы пытаетесь запустить узел кластера, с которым нет связи, вы получите внутреннюю ошибку кластера (CPFBB46).

• С помощью макрокоманды CLUSTERINFO можно просмотреть структуру кластера в том виде, в котором она известна службе ресурсов макроса. Помимо всего прочего, будут показаны список узлов кластера, состав доменов восстановления CRG и IP-адреса, выделенные кластеру.

Несогласованность или ошибки в этой информации могут дать ответ на вопрос о том, почему кластер работает не так, как планировалось. Сведения о расшифровке результатов макрокоманды CLUSTERINFO приведены в разделе “Определение причины неполадки с помощью макроса CLUSTERINFO” на стр. 131.

Понятия, связанные с данным

“Пользовательские очереди и структура заданий” на стр. 115

При управлении кластером необходимо разбираться в структуре заданий и принципе работы пользовательских очередей.

Задачи, связанные с данной

Просмотр заданий в подсистеме

Ссылки, связанные с данной

WRKACTJOB (Работа с активными заданиями)

Команда DSPCLUINF (Показать сведения о кластере)

Сбор данных для восстановления кластера

Для сбора полной информации о кластере можно воспользоваться командой Работа с кластером (WRKCLU). Эта информация может помочь при устранении неполадок.

Для просмотра узлов и объектов кластера, а также для работы с ними служит команда Работа с кластером (WRKCLU). При вызове этой команды показывается окно Работа с кластером. Кроме просмотра узлов и сведений о кластере эту команду можно применить для сбора данных о кластере

Для сбора сведений для восстановления выполните следующие действия:

1. В командной строке введите WRKCLU OPTION(OPTION). С помощью приведенных ниже опций можно указать требуемый тип информации о состоянии.

*SELECT

Отображается меню Работа с кластером.

*NODE

Отображается окно Сведения о кластере, представляющее собой список узлов кластера.

*CFG

Отображаются все параметры конфигурации кластера. Эта же опция позволяет получить подробную информацию о группе ресурсов кластера.

*CRG

Отображается список групп ресурсов конкретного кластера.

*SERVICE

Сбор отладочной информации и данных трассировки для всех заданий служб ресурсов кластера. Эта информация сохраняется в файле. Каждое задание представлено отдельным элементом. Эту опцию рекомендуется применять только по указанию сотрудника службы поддержки. Эта опция вызывает окно с приглашением Создать дамп трассировки кластера (DMPCLUTRC).

Определение причины неполадки с помощью команды Dump Cluster Trace (DMPCLUTRC)

С помощью команды Dump Cluster Trace (DMPCLUTRC) можно выявить и устранить неполадки кластера.

С помощью команды Dump Cluster Trace (DMPCLUTRC) можно определить, завершилось ли задание кластера, или какое задание выполняется в данный момент. Эта команда создает дампы кластера и записывает его вместе с отладочной информацией и данными трассировки кластера в файл. Дампы информации создаются локально на одном или нескольких узлах кластера. Этой командой можно воспользоваться для создания дампа одного или всех заданий службы ресурсов кластера (CRS). Каждому записанному заданию CRS в файле соответствует свой элемент. Эти элементы названы по именам заданий CRS. Для того чтобы генерировался вывод команды, поддержка кластеров должна быть включена. Вывод будет генерироваться только для узлов с активными заданиями CRS. Часть информации для дампа берется из пользовательской трассировки, еще часть - из объектов кластера. Объем информации определяется уровнем дампа. Уровни бывают следующими: базовая информация, сведения об ошибках, информационные данные и подробная информация. Уровень дампа обуславливает количество информации, которая будет заноситься в файл. В большинстве случаев сотрудник сервисного представительства фирмы IBM сообщает оптимальный для вас уровень дампа, но обычно для большинства неполадок достаточно уровня LEVEL(*ERROR). Если вы не знаете, какой уровень подойдет для вашей среды, обратитесь к сотруднику сервисного представительства фирмы IBM.

Интерпретация результатов трассировки

С помощью результатов трассировки можно разобраться в том, что делает кластер, например, определить задание, которого ожидает протокол. Вывод, который генерирует пользовательская трассировка, содержит разделительные линии из одинаковых символов (=). Эти линии разделяют результаты вызовов команды DMPCLUTRC. В одном файле может отражаться несколько вызовов DMPCLUTRC. Самая свежая информация отображена в последних дампах стека. В некоторых случаях задание CRG может содержать две группы. Каждой группе в файле соответствует отдельный раздел.

Ниже приведен пример результата работы команды Dump Cluster Trace для кластера MYCLUSTER с двумя узлами (SYSTEM1 и SYSTEM2). В кластере есть одна группа ресурсов (CRG) с именем MYCRG. Оба узла входят в домен восстановления MYCRG. Пользователь вызвал команду STRCRG CL, которая работает долго. Пользователь другой рабочей станции вводит в командной строке команду DMPCLUTRC CLUSTER(MYCLUSTER) CRG(*ALL) LEVEL(*ERROR) FILE(MYFILE).

В этом примере вывод команды DMPCLUTRC сохраняется в элементе MYCRG файла MYFILE. Для удобства интерпретации этот элемент разбит на два раздела MYCRG. Можно заметить, что в разделах подсвечены числа в круглых скобках. Эти числа идентифицируют описываемую информацию. Такие подробности помогают устранить неполадки кластера.

Примечание: Вертикальные многоточия обозначают, что часть информации была вырезана из вывода.

Раздел 1 результатов DMPCLUTRC

Дамп пользовательской трассировки для задания 073586/QSYS/MYCRG. Размер: 300K, Перенесен 0 раз.

```

--- 08/22/2005 16:43:32 ---
(1a) 00000006:658536 Main thread handle 2
(1b) 00000008:748016 Work thread 1 handle 13
(1b) 00000007:754576 Work thread 2 handle 11
--- 08/22/2005 16:46:04 ---
00000008:269608 CSTDAMBR 1115: WaitForMsg 4 1005 CPFBB3C
--- 08/22/2005 16:48:17 ---
00000006:925112
(1c) DMPCLUTRC Node SYSTEM1 Group MYCRG
=====

```

В первом разделе содержатся номера нитей и ссылок на задания кластера. Заданию кластера может соответствовать несколько нитей. В данном примере это одна главная нить (1a), в которую приходит вся информация, и две рабочие нити (1b). Также в этом разделе содержится информация о системе и о задании кластера, к которым относится трассировка (1c).

Раздел 2 результатов DMPCLUTRC

```
00000006:925168 Stack Dump For Target Thread: Handle 2 (0x00000002)
00000006:925192 Stack:
(2aa) Main Thread Stack MYCRG
00000006:925256 Stack: Library / Program Module Stmt Procedure
00000006:933432 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 0 : _CXX_PEP__Fv
00000006:933488 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 46 : main
00000006:933536 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 65 : completeStartup_FP8CstDAMbr
00000006:933584 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 26 : mainQueueProcessLoop_FP8Cs
DAMbr
00000006:933616 Stack: QSYS / QCSTCMN CSTDAMBR 57 : processQueueMsgs__8CstDAMbrF
Q2_8CstDAMbr13CstQueueIndex
00000006:933664 Stack: QSYS / QCSTCMN CSTDAMBR 53 : processMsg__8CstDAMbrFP6CstM
sg
00000006:933712 Stack: QSYS / QCSTCMN CSTDAMBR 17 : callFnPtr__8CstDAMbrFPQ2_8Cs
tDAMbr19MsgFunctionPtrEntryP6
00000006:933744 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 94 : crgDump_FP6CstMsgP8CstDAMbr
00000006:933792 Stack: QSYS / QCSTCMN CSTACK 95 : CstAckQueryMsg
00000006:933832 Stack: QSYS / QP0ZCPA QP0ZDBG 3 : Qp0zDumpTargetStack
00000006:933864 Stack: QSYS / QP0ZSCPA QP0ZSDBG 12 : Qp0zSUDumpTargetStack
00000006:934016 Stack: Exception In Stack Dump Code
00000006:934040 Stack: thread is likely terminated or no longer running the same code as the
captured stack
00000006:934080
(2a) Work Thread Index 1 Group MYCRG Last or current values
(2e) 00000006:934112 Request handle 8E3E1002 EE3218A1 824F0004 AC000456
(2c) 00000006:934136 SPI name QcstStartClusterResourceGroup
00000006:934160 (2g) POF 10, Completed ack round 1 (2i)
00000006:934176 (2o) In waitForJobEnd QDFTJOB MYCLUSTER 073590
00000006:934216 Node Ack Status POF (2bb) Nack Msg Id
00000006:934240 (2n) SYSTEM1 (2cc) Ready
00000006:934272 SYSTEM2 Ack 10 (2k)
00000006:934296 Messages
00000006:934320 Stack Dump For Target Thread: Handle 13 (0x0000000d)
00000006:934344 Stack: Work Thread 1 Stack MYCRG
00000006:934792 Stack: Library / Program Module Stmt Procedure
00000006:934840 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 9 : workThreadRoutine_FPv
00000006:934888 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 28 : workQueueProcessLoop_FP8Cst
DAMbr
00000006:941688 Stack: QSYS / QCSTCMN CSTDAMBR 57 : processQueueMsgs__8CstDAMbrF
Q2_8CstDAMbr13CstQueueIndex
00000006:941696 Stack: QSYS / QCSTCMN CSTDAMBR 33 : processMsg__8CstDAMbrFP6CstM
sg
00000006:941712 Stack: QSYS / QCSTCMN CSTDAMBR 17 : callFnPtr__8CstDAMbrFPQ2_8Cs
tDAMbr19MsgFunctionPtrEntryP6
00000006:941728 Stack: QSYS / QCSTCMN CSTACK 3 : CstStripOffHeaderMsgPart
00000006:941736 Stack: QSYS / QCSTCMN CSTDAMBR 53 : processMsg__8CstDAMbrFP6CstM
sg
00000006:941752 Stack: QSYS / QCSTCMN CSTDAMBR 17 : callFnPtr__8CstDAMbrFPQ2_8Cs
tDAMbr19MsgFunctionPtrEntryP6
00000006:970888 Stack: QSYS / QCSTCRGS2 CSTCRGSS 39 : startCrg
00000006:970912 Stack: QSYS / QCSTCRGS2 CSTCRGSS 344 : doMessageProcessing_FP6CstM
sgP8CstDAMbr
00000006:970928 Stack: QSYS / QCSTCRGS2 CSTCRGSS 57 : doExitPgmPhase_FP6CstMsgP8C
stDAMbr
00000006:981984 Stack: QSYS / QCSTCMN CSTDAMBR 52 : waitForJobEnd__8CstDAMbrFPA2
6_ci
00000006:982000 Stack: QSYS / QCSTCMN CSTDAMBR 73 : waitForSpecialMsg__8CstDAMbr
FP17CstSpecialMsgListPA8_ciT3
00000006:982016 Stack: QSYS / QC2UTIL1 QC2MI3 1 : (2dd) deq
00000006:982136 Stack: Exception In Stack Dump Code
00000006:982136 Stack: thread is likely terminated or no longer running the same code as the
captured stack
00000006:982160
(2b) Work Thread Index 2 Group MYCRG Last or current values
(2f) 00000006:982176 Request handle D9C3C8C3 E2E3F5F2 0003 0000
(2cc) 00000006:982176 SPI name
```

```

|         00000006:982184 (2h) POF 0, (2d)Completed ack (2j)round 0
|         00000006:982184   In getNextWorkMsg
|         00000006:982208   Node      Ack Status    POF      Nack Msg Id
| (21) 00000006:982208   SYSTEM1  Ready
| (21) 00000006:982232   SYSTEM2  Ready
|         00000006:982248   Messages
|         00000006:982256 Stack Dump For Target Thread: Handle 11 (0x0000000b)
|         00000006:982256 Stack: Work Thread 2 Stack MYCRG
|         00000006:982344 Stack: Library / Program Module Stmt Procedure
|         00000006:982360 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 9 : workThreadRoutine_FPv
|         00000006:982376 Stack: QSYS / QCSTCRGJOB CSTCRGJOB 28 : workQueueProcessLoop_FP8Cst
|         DAMbr
|         00000006:982392 Stack: QSYS / QCSTCMN CSTDAMBR 51 : processQueueMsgs__8CstDAMbrF
|         Q2_8CstDAMbr13CstQueueIndex
| (2m) 00000006:982400 Stack: QSYS / QCSTCMN CSTDAMBR 105 : getNextWorkMsg__8CstDAMbrFv
|         00000006:982416 Stack: QSYS / QC2UTIL1 QC2MI3 1 : deq
|         00000006:982480 Stack: Exception In Stack Dump Code
|         00000006:982480 Stack: thread is likely terminated or no longer running the same code as the
| captured stack

```

Во втором разделе содержатся стеки вызовов для каждой нити, входящей в задание кластера. Главная нить большей частью показывает только что выполненную команду DMPCLUTRC (2aa). В рабочих нитях (2a и 2b) содержатся данные трассировки, по которым можно выяснить, что происходило с заданием кластера. В этом разделе содержится подробная информация о стеке вызовов, в частности: имя SPI (2c), подтверждение приема (ACK)(2d), описатель запроса для связанных API (2e) или последний описатель запроса (2f), точка сбоя (POF) (2g и 2h), текущий комплект подтверждений (ACK) (2i и 2j) и подтвержденные узлы (ACK) (2k и 2l).

В данном контексте термин *точка сбоя (POF)* обозначает внутреннее значение, характеризующее место в текущем протоколе. При этом не важно, возникла в этом месте неполадка или нет. Пометка *Ack* обозначает, что узел успешно выполнил эту часть протокола и ждет подтверждения (ACK) или отказа в подтверждении (Nack) от остальных узлов. Пометка *Nack* означает, что узлу не удалось выполнить эту часть протокола и он ждет ответа от остальных узлов. ИД сообщения для Nack приводится в следующем столбце (2bb). Это то же самое сообщение, которое отправлено в очередь RIQ отправителя. Если во время обработки протокола узел выйдет из строя, то ему будет соответствовать состояние Fail и возможно (в зависимости от протокола и узла), состояние Nack. Состояние Ack или Inactive означает, что узел не участвует в протоколе. Состояние Ready означает, что узел еще не ответил. Если нить находится в состоянии getNextWorkMsg (2m), то это говорит о том, что она ожидает задания.

Названия процедур по стеку вызовов читаются снизу вверх. В файле данного примера содержится deq (2dd) с состояниями waitForSpecialMsg, waitForJobEnd и doExitPgmPhase. Это означает, что протокол ждет завершения работы программы выхода. По значению Состояние Ack (2k) можно определить, от какого узла протокол ждет ответа. В этом примере это узел SYSTEM1 (2n). Задание, завершения которого ждет система, можно определить по полному имени задания (2e). Как только задание определено, остается только исправить его и устранить причину задержки. Эти причины могут быть следующими: задание еще находится в очереди заданий, для выполнения задания требуется время, или же задание ждет объект, а тот заблокирован.

В данном примере протокол ожидает завершения программы выхода. Определить, ожидает ли протокол завершения программы выхода или других заданий проще всего по первому разделу: посмотрите строку waitForJobEnd (2o). В этой строке находится имя задания, завершения которого ожидает протокол. Вам не придется просматривать все стеки.

Раздел 3 результатов DMPCLUTRC

```

| 5722SS1 V5R4M0 060210 AS/400 DUMP 073586/QSYS/MYCRG 08/22/05 16:48:18 PAGE 1
| DMPYSOBJ PARAMETERS
| (3a)OBJ- MYCRGAIX CONTEXT- QTEMP
| TYPE- *ALL SUBTYPE--ALL
| OBJECT TYPE- INDEX *CRGM
| NAME- MYCRGAIX TYPE- 0E SUBTYPE- A5

```

```

| LIBRARY- QTEMP 006B8A19B00C9E807000 TYPE- 04 SUBTYPE- C1
| CREATION- 08/22/05 16:43:32 SIZE- 0000007000
| ATTRIBUTES- 0000 ADDRESS- C7FE286F04 000000
| .
| .
| .POINTERS-
| NONE
| OIR DATA-
| NONE
| END OF DUMP
|
| * * * * * Конец распечатки * * * * *

```

В третьем разделе показаны внутренние объекты, хранящие информацию о заданиях кластера. В данном примере это внутренний индекс MYCRGAIX (3a). Третий раздел более понятен, чем второй.

Раздел 4 результатов DMPCLUTRC

```

| 5722SS1 V5R4M0 060210 AS/400 DUMP 073586/QSYS/USER 08/22/05 16:48:18 PAGE 1
| DMPYSOBY PARAMETERS
| (4a) OBJ- MYCRGTQ CONTEXT- QTEMP
| TYPE- 0A SUBTYPE-EF
| OBJECT TYPE- QUEUE *QTQ
| NAME- MYCRGTQ TYPE- 0A SUBTYPE- EF
| LIBRARY- QTEMP 006B8A19B00C9E807000 TYPE- 04 SUBTYPE- C1
| CREATION- 08/22/05 16:43:32 SIZE- 000002C000
| ATTRIBUTES- 0000 ADDRESS- CC6765CAA2 000000
| QUEUE ATTRIBUTES-
| .
| .
| .POINTERS-
| 000010 SPP 1A EF MYCRG QSYS 073586 00002160 0000
| 000020 SPP 1A EF MYCRG QSYS 073586 00001540 8000
| 000030 SPP 1A EF MYCRG QSYS 073586 000016E0 0000
| 000040 SPP 1A EF MYCRG QSYS 073586 00001690 0000
| 000050 SPP 1A EF MYCRG QSYS 073586 000016A0 0000
| 000070 SPP 1A EF MYCRG QSYS 073586 00002160 0000
| OIR DATA-
| NONE
| END OF DUMP
|
| * * * * * Конец распечатки * * * * *

```

В четвертом разделе показана очередь трассировки (4a). В примере она называется MYCRGTQ. В ней содержится информация о том, какие задания выполнялись в кластере и как каждое задание отвечало на запрос.

Примечание: Мы не будем здесь подробно описывать каждое сообщение.

Раздел 5 результатов DMPCLUTRC

```

| 5722SS1 V5R4M0 060210 AS/400 DUMP 073586/QSYS/MYCRG 08/22/05 16:48:18 PAGE 1
| DMPYSOBY PARAMETERS
| (5a) OBJ- MYCRG CONTEXT- QUSRSYS
| TYPE- 19 SUBTYPE-2C
| OBJECT TYPE- SPACE *CRG
| NAME- MYCRG TYPE- 19 SUBTYPE- 2C
| LIBRARY- QUSRSYS TYPE- 04 SUBTYPE- 01
| CREATION- 08/17/05 07:16:40 SIZE- 0000002000
| OWNER- MYCLUSTER TYPE- 08 SUBTYPE- 01
| ATTRIBUTES- 0800 ADDRESS- 1EC687A1F3 000000
| SPACE ATTRIBUTES-
| .
| .

```


END OF DUMP

* * * * * Конец распечатки * * * * *

В пятом разделе содержится информация об объекте CRG (5а).

Определение причины неполадки с помощью макроса CLUSTERINFO

Макрос CLUSTERINFO отображает информацию об узлах, группах CRG и IP-адресах активного кластера. Эта информация хранится в службе ресурсов кластера.

Макрос CLUSTERINFO создает моментальную копию информации о текущем кластере. Команда создает описание кластера локального узла, проходя по всем кластерным объектам. CLUSTERINFO предоставляет "черный ящик" для различных кластерных объектов, который может помочь в определении источника неполадки в кластере. Для получения доступа к макросу CLUSTERINFO выполните следующие действия:

1. В командной строке введите STRSST.
2. Войдите в систему с пользовательским профайлом сервисных средств.
3. В меню системного инструментария выберите опцию 1 (Запустить сервисное средство).
4. Выберите опцию 4 (Показать/Изменить/Создать дамп).
5. Выберите опцию 2 (Вывести дамп на принтер).
6. Выберите опцию 2 (Данные лицензионного внутреннего кода (LIC)).
7. Выберите опцию 14 (Расширенный анализ).
8. Введите 1 перед опцией макроса CLUSTERINFO. Нажмите Enter.

Когда появится макрос CLUSTERINFO, укажите опцию -H для отображения справки по всем опциям этого макроса. Опции макроса CLUSTERINFO описаны в следующей диаграмме:

Таблица 21. Опции макроса CLUSTERINFO

Опция	Описание
-H	Показать справку по опциям
-A	Показать всю информацию
-FR	Показать записи "черного ящика"
-HB	Показать информацию пульса
-PERF	Показать счетчики производительности
-Q	Показать состояние очереди исходящих сообщений
-G	Не подавлять отображение всех групп широковещания CC
-T	Показать параметры настройки
-M	Показать матрицы для всех DA
-DP	Показать информацию о dataPort

Интерпретация результатов работы макроса CLUSTERINFO

В данном примере указана опция -A, поэтому в дампе присутствуют все поля. Наиболее полезными для отладки являются "черные ящики". Обратите внимание, что эти "черные ящики" очищаются после закрытия или удаления узла. Для определения причин неполадок макрос CLUSTERINFO следует запускать до закрытия или удаления узла кластера. В некоторых случаях есть возможность записывать черные ящики в протокол vlog после удаления или закрытия узла. В черные ящики записываются различные события, воздействующие на структуру и производительность кластера. Подробная расшифровка данных черного ящика выходит за рамки данной публикации.

l **Примечание:** Вертикальные многоточия обозначают, что часть информации была вырезана из вывода.

Раздел 1 результатов макроса CLUSTERINFO

```
DISPLAY/ALTER/DUMP CLUSTERINFO -NEW2 08/23/05 13:36:37 PAGE 1
Running macro: CLUSTERINFO -A
Use -H for command information
Cluster Name : MYCLUSTER
Local Node Name: SYSTEM1
CC/CTS Version : 5
Macro Timestamp: 08/23/05 13:36:37.079
```

В разделе 1 содержится общая информация о кластере, в частности, имя кластера, версия, а также системное время создания отчета. В данном примере имя кластера - MYCLUSTER, а имя локального узла - SYSTEM1.

Раздел 2 результатов макроса CLUSTERINFO

```
Адреса объектов кластера
CstcClusterServices Address: DBF08681C9161580
Cluster Address : FC5B04B0D4001000
Cluster Task Address : B00010000E932000
Cluster Task Q Address : DBF08681C9169A00
Clue Group Services Address: CDAB6D0339001000
CC Services Address : FC5B04B0D4008000
```

В разделе 2 приводятся указатели на расположение ключевых объектов кластера.

Раздел 3 результатов макроса CLUSTERINFO

```
Статистика сообщений
Number of non-fragmented messages: 250
Number of fragmented messages : 1
Number of fragments : 7
Number of acks : 148
```

В разделе 3 содержится статистика сообщений кластера, например, количество фрагментов и подтверждений (ack).

Раздел 4 результатов макроса CLUSTERINFO

```
Схема узлов
Node ID : SYSTEM1
GenesisSubnetId : 9.5.251.0
CCNode * : FC5B04B0D4007000
CCSrvNode *: FC5B04B0D404F000
Adapter 1 : 9.5.251.46 Primary
Status : 0x01 Reachable
Line Type : 0x09 Ethernet
Node ID : SYSTEM2
GenesisSubnetId : 9.5.251.0
CCNode * : FC5B04B0D4060000
CCSrvNode *: FC5B04B0D4061000
Adapter 1 : 9.5.251.47 Primary
Status : 0x01 Reachable
Line Type : 0x09 Ethernet
```

В разделе 4 представлены в виде схемы узлов все узлы кластера, активные на данный момент. В данном примере два активных узла: SYSTEM1 и SYSTEM2. T

Раздел 5 результатов макроса CLUSTERINFO

```
Схема подсети
Subnet ID: 127.0.0.0
CCSubnet * : FC5B04B0D4006000
CCSrvSubnet* : FC5B04B0D400C000
Retries : 0
```

```
Msg Timeouts          : 0
Bad Msg Counter       : 0
Failed Default Address: 0
Subnet ID: 226.5.5.5
CCSubnet *           : FC5B04B0D405B100
CCSrvSubnet*         : FC5B04B0D405C000
Retries              : 0
Msg Timeouts          : 0
Bad Msg Counter       : 0
Failed Default Address: 0
```

Раздел 5 представляет собой список всех объектов подсети кластера.

Раздел 6 результатов макроса CLUSTERINFO

```
Схема групп
Group ID: 0x0000000000000001
Name : CTS
CCGroup * : FC5B04B0D405FF00
CCSrvGroup *: FC5B04B0D4064B00
Member Nodes
SYSTEM1
SYSTEM2
Group ID: 0x0000000000000002
Name : CTS
CCGroup * : FC5B04B0D4055100
CCSrvGroup *: FC5B04B0D4055200
Member Nodes
SYSTEM1
SYSTEM2
.
.
.
```

В разделе 6 перечислены все группы кластера. Группы называются группами распределенных операций. Эти группы служат для связи между группами на каждом активном узле кластера. Большинство групп работают с лицензионным внутренним кодом (LIC). Об этом свидетельствуют имена групп CTS и BADA. Кроме того, есть группы для CCTL (задание QCSTCTL в операционной системе), CRGM (задание QCSTCRGM в операционной системе) и для каждого задания группы ресурсов кластера (CRG). Группам для заданий CRG имена не присваиваются. Все группы состоят из узлов. Узлы-члены группы - это узлы, обменивающиеся данными в пределах этой группы.

Раздел 7 результатов макроса CLUSTERINFO

```
Схема разделов
Схема разделов пуста
```

Раздел 7 представляет собой список всех узлов в разделах SLIC.

Примечание: Его принцип отличается от разделившихся узлов XPF.

Раздел 8 результатов макроса CLUSTERINFO

```
Список клиентов CTS
Список клиентов CTS пуст
```

В разделе 8 содержится список всех зарегистрированных клиентов кластера, например, портов данных.

Раздел 9 результатов макроса CLUSTERINFO

```
Flight Recorder       : CSTCSVFR
Flight Recorder Address: DBF08681C9161620
```

В разделе 9 содержится черный ящик служб кластера (CSTCSVFR), находящихся в системе во время начальной загрузки (IPL).

Раздел 10 результатов макроса CLUSTERINFO

Статистика сообщений

```
Number of non-fragmented messages: 250
Number of fragmented messages   : 1
Number of fragments             : 7
Number of acks                  : 148
Time Stamp: 08/18/05 14:00:15.329
Trace Point: 0x0010 CstcClusterServicesTracePtCreatedFlightRecorder
C3D9C5C1E3C5C6D9 <CREATEFR>
Time Stamp: 08/22/05 16:43:28.912
Trace Point: 0x0020 CstcClusterServicesTracePtCreatedClusterObject
D4D6D9C5E8404040 4040C5F8D3770500 <MYCLUSTER E8L...>
1000 <.. >
Time Stamp: 08/23/05 13:33:40.935
Trace Point: 0x0030 CstcClusterServicesTracePtDeletedClusterObject
D4D6D9C5E8404040 404040E2E3 <MYCLUSTER ST >
Time Stamp: 08/23/05 13:33:41.204
Trace Point: 0x0030 CstcClusterServicesTracePtDeletedClusterObject
C3D4D7E3 <CMPT >
Time Stamp: 08/23/05 13:33:55.122
Trace Point: 0x0020 CstcClusterServicesTracePtCreatedClusterObject
D4D6D9C5E8404040 4040FC5B04B0D400 <MYCLUSTER ....M.>
1000 <.. >
```

В разделе 10 содержится черный ящик для CSTCCCFR. Этот черный ящик остается в системе только пока на данном узле работает поддержка кластера.

Раздел 11 результатов макроса CLUSTERINFO

Черный ящик : CSTCCCLR
Flight Recorder Address: FC5B04B0D4001E80

```
-----
Time Stamp: 08/23/05 13:33:54.944
Trace Point: 0x1010 CstcClusterTracePtCreatedSubnetObject
7F000000FC5B04B0 D4006000 <.....M.-. >
Time Stamp: 08/23/05 13:33:55.062
Trace Point: 0x1000 CstcClusterTracePtCreatedNodeObject
C3E2E3D9D9C3C8C3 E2E3F5F2FC5B04B0 <CSTRSYSTEM1....>
D4007000 <M... >
Time Stamp: 08/23/05 13:33:55.122
Trace Point: 0x1020 CstcClusterTracePtCreatedMCGroupObject
0000000000000001 00000000D9C3C8C3 <.....RCHC>
E2E3F5F2 <ST52 >
.
.
.
```

В разделе 11 содержится черный ящик связи кластера (CSTECLFR). Этот черный ящик остается в системе только пока на данном узле работает поддержка кластера.

Раздел 12 результатов макроса CLUSTERINFO

Черный ящик : CSTCCCFR
Flight Recorder Address: FC5B04B0D4006380

```
-----
Time Stamp: 08/23/05 13:33:55.080
Trace Point: 0x3000 CstcCCScamTracePtScamOpen
FC5B04B0D400E480 0000000000000000 <....M.U.....>
Time Stamp: 08/23/05 13:33:55.097
Trace Point: 0x3010 CstcCCScamTracePtScamBind
FC5B04B0D400E480 0000000000000000 <....M.U.....>
```

```

Time Stamp: 08/23/05 13:33:55.100
Trace Point: 0x3000 CstcCCScamTracePtScamOpen
FC5B04B0D400E480 0000000000000000 <....M.U.....>
D6E4E3 <OUT >
Time Stamp: 08/23/05 13:33:55.100
Trace Point: 0x3010 CstcCCScamTracePtScamBind
FC5B04B0D400E480 0000000000000000 <....M.U.....>
.
.
.

```

В разделе 12 содержится ключевой черный ящик (CSTCCCFR), который хранится в системе только пока на данном узле работает поддержка кластера.

Раздел 13 результатов макроса CLUSTERINFO

```

Системное время: 08/23/05 13:33:55.201
C3A2A385C7E27A7A C3A2A385C7E24082 <CsteGS::CsteGS b>
85878995A2 <egins >
Time Stamp: 08/23/05 13:33:55.201
C3A2A385C4C14083 9695A2A399A483A3 <CsteDA construct>
85847A40C2C1C4C1 404040404040 <ed: BADA >
Time Stamp: 08/23/05 13:33:55.201
C3A2A385C7E27A7A C3A2A385C7E24081 <CsteGS::CsteGS a>
8484408281848140 A39640C4C16D9389 <dd bada to DA_li>
A2A3 <st >
.
.
.

```

Раздел 13 отображает содержимое очередей отправки и очередей активных сообщений. Если этот раздел долго не очищается, это может говорить о неполадках в кластере.

Раздел 14 результатов макроса CLUSTERINFO

```

Черный ящик : CSTECLF2
Flight Recorder Address: CDAB6D0339001300
-----
Time Stamp: 08/23/05 13:33:55.201
C3A2A385C4C17A7A C3A2A385C4C16B40 <CsteDA::CsteDA, >
83998581A385D4C3 C79996A49740C9C4 <createMCGroup ID>
407E40F0A7F1F5 < = 0x15 >
Time Stamp: 08/23/05 13:33:55.209
C3A2A385E2C3D985 977A7A84859389A5 <CsteSCRep::deliv>
85994094A287E3A8 97857EF0A7F140A2 <er msgType=0x1 s>
A482E3A897857EF0 A7F240C4C17EC2C1 <ubType=0x2 DA=BA>
C4C1404040404040 <DA >
Time Stamp: 08/23/05 13:33:55.209
C3A2A385C4C17A7A A58985A66B409985 <CsteDA::view, re>
9496A585D4C3C799 96A497D485948285 <moveMCGroupMembe>
99A240C9C4407E40 F0A7F1F540969384 <rs ID = 0x15 old>
6D959684856D9389 A2A340A289A98540 <_node_list size >
7E40F0A7F0 <= 0x0 >
.
.
.

```

В разделе 14 содержится информация черного ящика.

Раздел 15 результатов макроса CLUSTERINFO

```

Очереди сообщений
Send Queues:
Send Queue: 00 Messages: 00 MessageQueue*: FC5B04B0D400BF80
Send Queue: 01 Messages: 00 MessageQueue*: FC5B04B0D400DF80
Send Queue: 02 Messages: 00 MessageQueue*: FC5B04B0D400E600

```

```

Send Queue: 03 Messages: 00 MessageQueue*: FC5B04B0D400E680
Send Queue: 04 Messages: 00 MessageQueue*: FC5B04B0D400E700
Send Queue: 05 Messages: 00 MessageQueue*: FC5B04B0D400E780
Send Queue: 06 Messages: 00 MessageQueue*: FC5B04B0D400E800
Send Queue: 07 Messages: 00 MessageQueue*: FC5B04B0D400E880
Send Queue: 08 Messages: 00 MessageQueue*: FC5B04B0D400E900
Send Queue: 09 Messages: 00 MessageQueue*: FC5B04B0D400E980
Send Queue: 10 Messages: 00 MessageQueue*: FC5B04B0D400EA00
Send Queue: 11 Messages: 00 MessageQueue*: FC5B04B0D400EA80
Send Queue: 12 Messages: 00 MessageQueue*: FC5B04B0D400EB00
Send Queue: 13 Messages: 00 MessageQueue*: FC5B04B0D400EB80
Send Queue: 14 Messages: 00 MessageQueue*: FC5B04B0D400EC00
Send Queue: 15 Messages: 00 MessageQueue*: FC5B04B0D400EC80
Send Queue: 16 Messages: 00 MessageQueue*: FC5B04B0D400ED00
Send Queue: 17 Messages: 00 MessageQueue*: FC5B04B0D400ED80
Send Queue: 18 Messages: 00 MessageQueue*: FC5B04B0D400EE00
Send Queue: 19 Messages: 00 MessageQueue*: FC5B04B0D400EE80
Active Message Queues:
Active Message Queue: 00 Messages: 00 MessageQueue*: FC5B04B0D4008570
Active Message Queue: 01 Messages: 00 MessageQueue*: FC5B04B0D4008640
Active Message Queue: 02 Messages: 00 MessageQueue*: FC5B04B0D4008710
Active Message Queue: 03 Messages: 00 MessageQueue*: FC5B04B0D40087E0
Active Message Queue: 04 Messages: 00 MessageQueue*: FC5B04B0D40088B0
Active Message Queue: 05 Messages: 00 MessageQueue*: FC5B04B0D4008980
Active Message Queue: 06 Messages: 00 MessageQueue*: FC5B04B0D4008A50
Active Message Queue: 07 Messages: 00 MessageQueue*: FC5B04B0D4008B20
Active Message Queue: 08 Messages: 00 MessageQueue*: FC5B04B0D4008BF0
Active Message Queue: 09 Messages: 00 MessageQueue*: FC5B04B0D4008CC0
Active Message Queue: 10 Messages: 00 MessageQueue*: FC5B04B0D4008D90
Active Message Queue: 11 Messages: 00 MessageQueue*: FC5B04B0D4008E60
Active Message Queue: 12 Messages: 00 MessageQueue*: FC5B04B0D4008F30
Active Message Queue: 13 Messages: 00 MessageQueue*: FC5B04B0D4009000
Active Message Queue: 14 Messages: 00 MessageQueue*: FC5B04B0D40090D0
Active Message Queue: 15 Messages: 00 MessageQueue*: FC5B04B0D40091A0
Active Message Queue: 16 Messages: 00 MessageQueue*: FC5B04B0D4009270
Active Message Queue: 17 Messages: 00 MessageQueue*: FC5B04B0D4009340
Active Message Queue: 18 Messages: 00 MessageQueue*: FC5B04B0D4009410
Active Message Queue: 19 Messages: 00 MessageQueue*: FC5B04B0D40094E0

```

Tuning Values

```

cstcRcvSendTimerRatio      : 2 Default: 2
cstcMcastRelayTimerRatio   : 8 Default: 8
cstcMcastRelayHBTTimerRatio : 4 Default: 4
cstcHeartbeatBaseTimer     : 12288000000 Default: 12288000000
cstcHeartbeatBasePrecision : 4096000000 Default: 4096000000
cstcRetryPrecision         : 4096000000 Default: 4096000000
cstcRetryTimerVal          : 4096000000 Default: 4096000000
cstcCDATBaseTimer          : 491520000000 Default: 491520000000
cstcCDATBasePrecision      : 40960000000 Default: 40960000000
cstcRecoveryBaseTimer      : 3686400000000 Default: 3686400000000
cstcRecoveryBasePrecision  : 491520000000 Default: 491520000000
cstcMaxRetryTime           : 32768000000 Default: 32768000000
cstcCCFragmentSize         : 1464 Default: 1464
cstcCCSendQOverflow        : 1024 Default: 1024
cstcBadMsgCtrThreshold     : 3 Default: 3
cstcUnreachableHBAckThreshold: 1 Default: 1
cstcReachableHBAckThreshold : 3 Default: 3
cstcUnreachableHBThreshold : 4 Default: 4
cstcReachableHBThreshold   : 4 Default: 4
cstcMaxHBThreshold         : 16 Default: 16
cstcDisableMsgTimer        : 0 Default: 0
cstcRepeatAckThreshold     : 10 Default: 10
DISPLAY/ALTER/DUMP CLUSTERINFO -NEW2 08/23/05 13:36:37 PAGE 87
cstcDelayedAckTimer        : 409600000 Default: 409600000
cstcDelayedAckPrecision    : 40960000 Default: 40960000
cstcCCSendWindow           : 2 Default: 2

```

```
cstcCCEnableMcast      : 1 Default: 1
cstcCCPerfClass        : 2 Default: 2
```

***** КОНЕЦ ДАМПА *****

Раздел 15 результатов макроса CLUSTERINFO Настраиваемые значения - это параметры конфигурации и параметрам производительности кластера, описанные в разделе API Показать информацию службы ресурсов кластера (QcstRetrieveCRSInfo) API. В разделе 14 содержатся текущие значения этих полей и их значения по умолчанию.

Наиболее частые неполадки кластеров

В этом разделе перечислены наиболее часто встречающиеся неполадки кластеров и обсуждаются способы их предотвращения и устранения их последствий.

В этом разделе перечислены наиболее частые незначительные и легко устранимые неполадки, возникающие при работе с кластерами.

Не удается запустить или перезапустить узел кластера.

Эта неполадка обычно вызвана неправильной настройкой сети. Проверьте, правильно ли заданы сетевые атрибуты: циклический адрес, параметры INETD, атрибут ALWADDCLU и IP-адреса, выделенные кластеру.

- Если ошибка возникает при попытке запустить удаленный узел, следует проверить значение атрибута ALWADDCLU на удаленном узле. Этому атрибуту должно быть присвоено значение *ANY или *RQSAUT, в зависимости от вашей среды.
- IP-адреса, выделенные кластеру на локальном и удаленном узлах, должны находиться в состоянии *Активен*.
- Циклический адрес (127.0.0.1) на локальном и удаленном узлах должен находиться в состоянии 'активен'.
- Локальный и удаленные узлы должны отвечать на запросы PING по IP-адресам, выделенным кластеру.
- На удаленном узле должен быть запущен сервер INETD. Во время работы сервера INETD порт 5550 на удаленном узле должен находиться в состоянии *Прослушивание*. Инструкции по запуску сервера INETD приведены в разделе Сервер INETD.
- До запуска узла на нем должен быть открыт порт 5551.

Удалось создать только несколько разрозненных кластеров, каждый из которых состоит из одного узла.

Такая ситуация чаще всего бывает вызвана проблемами со связью между узлами. Проверьте параметры сети.

Программа выхода работает очень медленно.

Чаще всего эта неполадка бывает вызвана неправильными параметрами в описании задания, используемого программой выхода. Например, может быть указано слишком малое значение параметра MAXACT, из-за чего в каждый момент времени может выполняться слишком мало экземпляров программы выхода. Этому параметру рекомендуется присвоить значение *NOMAX.

В целом наблюдается низкая производительность.

Эта неполадка может быть вызвана целым рядом причин.

- Скорее всего, недостаточна пропускная способность линии связи.
- Еще одна часто встречающаяся причина - несовместимость среды связи и параметров передачи сообщений. Текущие параметры производительности можно посмотреть с помощью API Показать информацию службы ресурсов кластера (QcstRetrieveCRSInfo) и изменить с помощью API Изменить службу ресурсов кластера (QcstChgClusterResourceServices). Параметры производительности кластера,

предусмотренные по умолчанию, могут оказаться неэффективными при работе с устаревшими сетевыми адаптерами. В качестве *устаревших* рассматриваются модели 2617, 2618, 2619, 2626 и 2665. Для таких адаптеров следует установить *Класс производительности Обычный*.

- Еще одна возможная причина - неправильная настройка групповой рассылки IP. Если главные адреса кластера (адреса, указанные для узлов при создании кластера или первоначальном добавлении узлов в кластер) находятся в одной локальной сети, в кластере будет применяться функция групповой рассылки пакетов IP. С помощью команды NETSTAT проверьте, определена ли для главных адресов кластера группа рассылки 226.5.5.5. Для этого выберите опцию 14 *Показать группу рассылки*. Если группа рассылки не существует, проверьте, присвоено ли параметру производительности кластера *Применять групповую рассылку* значение ДА. Это можно сделать с помощью API QcstRetrieveCRSInfo (Показать информацию службы ресурсов кластера).
- Если все узлы кластера находятся в одной локальной сети или поддерживают значения MTU, превышающие 1464 байта, то можно значительно повысить производительность за счет увеличения максимального размера сообщений кластера (свыше 1536 Кб) - для этого нужно увеличить значение параметра производительности кластера *Размер фрагмента сообщения* до максимального возможного значения MTU.

Недоступны новые функции, добавленные в текущей версии.

Если при попытке выполнения новой функции появляется сообщение CPFBB70, это означает, что в конфигурации все еще указана предыдущая версия кластера. Нужно установить новую версию кластера на всех узлах, а затем изменить номер версии в конфигурации с помощью интерфейса *Скорректировать версию кластера*. Дополнительная информация приведена в разделе *Коррекция версии кластера*.

Не удалось добавить узел в домен устройств или обратиться к интерфейсу управления кластером Навигатора iSeries.

Для применения интерфейса управления кластером Навигатора iSeries и для работы с переносимыми устройствами нужно установить в системе i5/OS компонент 41, - HA Switchable Resources. Для установки этого компонента требуется ключ лицензии.

PTF для кластера установлен, но не работает.

1 Проверьте, выполнены ли после применения PTF следующие задачи:

1. Завершение работы кластера
2. Выход из системы и повторный вход
До тех пор, пока группа активации не будет уничтожена, в ней будет работать старая версия программы. Весь код кластера (включая API) работает в группе активации, применяемой по умолчанию.
3. Запуск кластера
Большинство PTF кластера требуют для активации завершения и перезапуска узла.

В протоколе программы выхода появляются сообщения CEE0200.

Это сообщение об ошибке выдается процедурой Q_LE_1eBdyPeilog из модуля QLEPM. Все программы, вызываемые программой выхода, должны выполняться в группе активации *CALLER или в явно указанной группе активации. Для устранения этой неполадки нужно скорректировать программу выхода или программу, при вызове которой возникает ошибка.

В протоколе службы ресурсов кластера появляются сообщения CPD000D и CPF0001.

Убедитесь, что системное значение QMLTTHDACN равно 1 или 2.

Кластер не подает признаков жизни.

Проверьте состояние программ выхода CRG. Это можно сделать с помощью команды WRKACTJOB (Работа с активными заданиями): для программ выхода кластера в столбце Функция указано значение PGM-QCSTCRGEXT.

Понятия, связанные с данным

“Подготовка узла к добавлению в кластер” на стр. 91

Для того чтобы узел можно было добавить в кластер, нужно задать сетевой атрибут ALWADDCLU (Разрешить добавление в кластер).

“Производительность кластера” на стр. 113

Изменение кластера может повлиять на его производительность.

“Версия кластера” на стр. 13

Версией кластера называется идентификатор набора функций, которые может выполнять кластер.

“Навигатор iSeries - Управление кластером” на стр. 74

Компания IBM разработала интерфейс управления кластером, предназначенный для Навигатора iSeries. Этот интерфейс поставляется в составе Компонента 41 (i5/OS - HA Switchable Resources).

Задачи, связанные с данной

“Коррекция версии кластера” на стр. 104

Версия кластера - это версия протокола связи, применяемого узлами кластера для обмена данными.

Распад кластера

Большинство неполадок кластера устранить достаточно легко. Распад кластера - это одна из наиболее серьезных неполадок. В этом разделе приведены рекомендации по предотвращению распада кластера и приведен пример процедуры восстановления распавшегося кластера.

Кластер распадается на части в случаях, когда связь между узлами прерывается, но при этом не удается подтвердить выход потерянных узлов из строя. Не следует путать распад кластера с разбиением на логические разделы в среде LPAR.

Если в протоколе хронологии (QHST) или протоколе задания QCSTCTL появится сообщение CPFBB20, оно сигнализирует о распаде кластера и необходимости предпринять действия по устранению последствий распада. Ниже приведен пример распада кластера, состоявшего из четырех узлов: А, В, С и D. В данном примере предполагается, что разорвана связь между узлами В и С, что привело к распаду кластера на две части. Перед распадом кластера существовали четыре CRG (тип CRG не имеет значения): CRG А, CRG В, CRG С и CRG D. В данном примере показаны изменения, которые произошли в доменах восстановления этих CRG в результате распада кластера.

Таблица 22. Пример домена восстановления распавшегося кластера

Узел А	Узел В	х	Узел С	Узел D
CRG А (первый резервный)	CRG А (главный)			
	CRG В (главный)		CRG В (первый резервный)	
	CRG С (главный)		CRG С (первый резервный)	CRG С (второй резервный)
CRG D (второй резервный)	CRG D (главный)		CRG D (первый резервный)	
Часть 1			Часть 2	

Распад кластера может произойти от того, что размер MTU на одном из транзитных участков меньше, чем размер фрагмента сообщения кластера. Значение MTU для IP-адреса кластера можно узнать с помощью команды Работа с состоянием сети TCP/IP (WRKTCPSTS) на соответствующем узле. Помимо этого, следует проверить значение MTU на всех транзитных участках линии связи. Если MTU меньше, чем размер фрагмента сообщения, нужно либо увеличить MTU, либо уменьшить размер фрагмента сообщения. Текущие

параметры производительности можно посмотреть с помощью API Показать информацию службы ресурсов кластера (QcstRetrieveCRSInfo) и изменить с помощью API Изменить службу ресурсов кластера (QcstChgClusterResourceServices).

После устранения причины распада кластер самостоятельно определит, что линия связи вновь работает, и выдаст сообщение CPFBB21 в протокол хронологии (QHST) или в протокол задания QCSTCTL. Это сообщение информирует оператора о том, что распавшийся кластер вновь объединен. Учтите, что на объединение кластера после устранения причин распада может потребоваться несколько минут.

Понятия, связанные с данным

“Части кластера” на стр. 28

Часть кластера - это множество активных узлов кластера, образовавшееся в результате сбоя связи. Серверы в отделившейся части по-прежнему поддерживают связь между собой.

“Предотвращение распада кластера” на стр. 85

Возникновения наиболее часто встречающихся предпосылок к распаду кластера можно избежать путем создания избыточных средств связи между узлами кластера.

“Объединение” на стр. 23

Операция *объединения* аналогична операции возврата узла в кластер. Разница заключается в том, что объединение выполняется при восстановлении связи между частями кластера, образовавшимися в результате распада.

“Пример: Автоматический перенос ресурсов” на стр. 18

Обычно автоматический перенос ресурсов выполняется при выходе узла из строя, но этим список причин не ограничивается.

Определение роли частей кластера после распада

Для того чтобы выполнить правильные действия по устранению последствий распада кластера, нужно определить, какая из частей кластера после распада стала главной, а какие - вторичными. Как только это будет сделано, каждой части кластера будет соответствовать главная или вторичная роль по отношению к группам ресурсов кластера.

В иерархической модели главным будет считаться часть, содержащая текущий главный узел. Все остальные части кластера считаются дополнительными. Для каждой группы ресурсов кластера может быть своя главная часть.

В равноправной модели действуют следующие правила:

- Главной будет считаться та часть, в которую входят все узлы домена восстановления.
- Если узлы домена восстановления входят в несколько частей, то главной среди них не будет. Обе части будут дополнительными.
- Если в какой-либо части кластера нет равноправных узлов и в ней активна группа ресурсов кластера, то работа CRG в этой части будет завершена.
- В дополнительной части разрешены операционные изменения (с учетом их ограничений).
- Изменения конфигурации в дополнительной части не допускаются.

В результате распада устанавливаются следующие ограничения на API для работы с CRG:

Таблица 23. Ограничения на API для работы с CRG, связанные с распадом кластера

API	Разрешено в главной части	Разрешено во вторичных частях
Добавить узел в домен восстановления	X	
Добавить запись устройства в CRG		
Изменить группу ресурсов кластера	X	
Изменить запись устройства в CRG	X	X
Создать группу ресурсов кластера.		

Таблица 23. Ограничения на API для работы с CRG, связанные с распадом кластера (продолжение)

API	Разрешено в главной части	Разрешено во вторичных частях
Удалить группу ресурсов кластера	X	X
Разослать информацию	X	X
Завершить работу CRG ¹	X	
Выполнить принудительный перенос ресурсов	X	
Показать список CRG	X	X
Показать сведения о CRG	X	X
Удалить узел из домена восстановления	X	
Удалить запись устройства из CRG	X	
Запустить CRG ¹	X	
Примечание:		
1. Разрешено для всех частей для CRG, но влияет только на ту часть, в которой запущен API.		

Эти ограничения установлены для того, чтобы сохранить возможность синхронизации CRG после восстановления кластера. По мере того, как узлы будут возвращаться в кластер, CRG с главных узлов будут обновляться на резервных узлах.

- | При объединении двух дополнительных частей в равноправной модели копироваться на все остальные узлы
- | будет та часть, в которой была активная CRG. Если состояние CRG в обеих частях одинаково, то
- | копироваться будет та часть, в которой был первый узел из списка домена восстановления CRG. Версия CRG
- | в этой части будет копироваться на узлы всех остальных частей.

Как только будет обнаружен распад, во всех частях кластера запрещается применение API Добавить запись узла кластера, Скорректировать версию кластера и Создать кластер. API Добавить запись в домен устройств можно применять только в том случае, если все узлы домена восстановления находятся в одной части кластера. Все API управления кластером можно выполнять без ограничений. Однако действие этих API распространяется только на ту часть кластера, из которой они вызваны.

Перевод узлов после распада кластера в разряд сбойных

В некоторых случаях выдается сообщение о разделе кластера, хотя в действительности просто узел вышел из строя. Это происходит в случаях, когда служба теряет связь с некоторыми узлами кластера и не может определить, вышли ли они из строя. Если это требуется, вы можете пометить узлы как сбойные вручную.

Внимание: Если вы пометите узел как сбойный, это упростит восстановление распавшегося кластера. Однако не следует помечать узлы как сбойные в ситуации, когда действительно произошел распад кластера и все узлы по-прежнему работают. В противном случае может возникнуть ситуация, когда несколько узлов в разных частях кластера попытаются выполнять функции главного узла для одной и той же CRG. Это может легко привести к повреждению или рассинхронизации файлов и данных, поскольку все узлы, считающие себя главными узлами CRG, будут независимо друг от друга вносить изменения в собственные копии файлов. Кроме того, если в разных частях кластера находятся узлы, считающие себя главными по отношению к одной и той же CRG, то объединить эти части в кластер невозможно.

Перевод узла в разряд сбойных может повлечь за собой перераспределение ролей в доменах восстановления CRG, в которых участвовал этот узел. Узел, переведенный в разряд сбойных, станет последним резервным узлом. Если несколько узлов вышли из строя одновременно, то итоговый порядок узлов в домене восстановления будет зависеть от того, в каком порядке эти узлы переводились в разряд сбойных. Если сбойный узел был главным узлом домена восстановления CRG, то первый активный узел в списке резервных станет главным узлом домена восстановления.

Понятия, связанные с данным

“Объединение” на стр. 23

Операция *объединения* аналогична операции возврата узла в кластер. Разница заключается в том, что объединение выполняется при восстановлении связи между частями кластера, образовавшимися в результате распада.

“Возврат узла в кластер” на стр. 21

Возвратом называется ситуация, когда в кластер вновь входит узел, ранее входивший в кластер и вышедший из него.

Задачи, связанные с данной

“Советы: распад кластера”

При распаде кластера воспользуйтесь приведенными советами.

Ссылки, связанные с данной

Команда CHGCLUNODE

API Изменить запись узла кластера (QcstChangeClusterNodeEntry)

Команда STRCLUNOD

API Запустить узел кластера (QcstStartClusterNode)

Работа с Навигатором iSeries:

Для выполнения этой операции должен быть установлен и зарегистрирован компонент 41 (i5/OS - HA Switchable Resources).

Если служба ресурсов кластера потеряет связь с узлом и не сможет определить, работает ли этот узел, то в списке узлов Навигатора iSeries ему будет присвоено состояние **Нет связи**. В некоторых случаях в такой ситуации следует поменять состояние узла с **Нет связи** на **Сбой**. После этого вы сможете перезапустить узел.

Для того чтобы изменить состояние узла с **Нет связи** на **Сбой**, выполните следующие действия:

1. Запустите Навигатор iSeries и откройте **Централизованное управление**.
2. Откройте **Кластеры**.
3. Откройте кластер с узлом, состояние которого требуется изменить.
4. Щелкните на значке **Узлы**.
5. Щелкните правой кнопкой мыши на узле, состояние которого необходимо изменить, и выберите **Cluster** → **Изменить состояние**.

и выберите опцию ClusterChange Status

Для перезапуска узла выполните следующие действия:

1. Щелкните правой кнопкой на нужном узле и выберите **Кластер** → **Запустить**.

С помощью команд CL и API:

Для того чтобы изменить состояние узла с **Нет связи** на **Сбой**, выполните следующие действия:

1. Измените состояние узла с помощью команды CHGCLUNODE или API Изменить запись узла кластера (QcstChangeClusterNodeEntry). Эту операцию нужно сделать для всех узлов, которые действительно вышли из строя.
2. С помощью команды STRCLUNOD или API Запустить узел кластера (QcstStartClusterNode) запустите узел, чтобы вернуть его в кластер.

Советы: распад кластера

При распаде кластера воспользуйтесь приведенными советами.

1. Ограничения на выполнение операций после распада кластера вводятся для того, чтобы сохранить возможность объединения частей обратно в кластер. Если не соблюдать эти ограничения, то могут быть выполнены операции, после которых восстановление кластера станет чрезвычайно трудоемким.

2. Уничтожение узлов в главной части кластера затрудняет работу с узлами в остальных частях кластера. Обычно потеря узлов главной части кластера происходит из-за физического повреждения помещения, в котором находилась главная часть кластера. Воспользуйтесь примером, приведенным в разделе Устранение ошибок, возникших в результате распада кластера, исходя из предположения, что повреждена часть 1. В этом случае главные узлы для CRG B, C и D будут находиться в части 2. Тогда проще всего восстановить кластер путем перевода узлов A и B в разряд сбойных. Дополнительная информация об этом приведена в разделе Перевод узлов после распада кластера в разряд сбойных. Кластер можно восстановить вручную. Для этого нужно выполнить следующие действия:
 - a. Удалите узлы A и B из кластера для части 2. Теперь часть 2 будет составлять весь кластер.
 - b. Настройте в новом кластере среду логической репликации. IE. Воспользуйтесь командой или API Запустить группу ресурсов кластера.

Поскольку из определения кластера в части 2 были удалены узлы, объединение частей 1 и 2 теперь невозможно. Для того чтобы устранить несоответствия в определениях кластера, вызовите API Удалить кластер (QcstDeleteCluster) на всех узлах части 1. Затем добавьте узлы из части 1 в кластер и восстановите все определения CRG, домены восстановления и среду логической репликации. Это требует больших усилий и предельного внимания. Настоятельно не рекомендуется выполнять эту процедуру в случаях, когда можно обойтись менее разрушительными средствами.

3. Процедура запуска узла кластера зависит от состояния запускаемого узла:

Если узел вышел из строя, либо его работа была завершена с помощью операции Завершить работу узла:

- a. На узле, добавляемом в кластер, будет запущена служба ресурсов кластера.
- b. С одного из активных узлов кластера на этот узел будет скопировано определение кластера.
- c. На добавляемый узел будут скопированы определения CRG, в домены восстановления которых входил этот узел. Эти определения будут взяты с других активных узлов кластера. Никакие CRG с добавляемого узла не передаются на другие узлы.

Если узел вышел из кластера в результате распада:

- a. Определение кластера на запускаемом узле сравнивается с определением кластера на одном из активных узлов. Если определения одинаковые, то дальше выполняется обычная процедура объединения кластера. Если определения разные, процедура объединения будет прервана и потребуются вмешательство пользователя.
- b. Если операция объединения будет продолжена, то добавляемый узел станет активным узлом кластера.
- c. Для всех CRG, в домены восстановления которых входит новый активный узел, будет выполнена синхронизация на всех узлах кластера. В ходе синхронизации CRG могут копироваться как на новый активный узел, так и с этого узла на другие.

Задачи, связанные с данной

“Перевод узлов после распада кластера в разряд сбойных” на стр. 141

В некоторых случаях выдается сообщение о разделе кластера, хотя в действительности просто узел вышел из строя. Это происходит в случаях, когда служба теряет связь с некоторыми узлами кластера и не может определить, вышли ли они из строя. Если это требуется, вы можете пометить узлы как сбойные вручную.

Ссылки, связанные с данной

API Удалить кластер (QcstDeleteCluster)

Прочие неполадки

В этом разделе обсуждаются процедуры восстановления после прочих неполадок в работе кластера.

Устранение последствий сбоев в заданиях кластера

Аварийное завершение задания службы ресурсов кластера, как правило, свидетельствует о какой-то внешней неполадке.

Просмотрите протокол сбойного задания и найдите сообщения, в которых указана причина сбоя. Устраните причины возникновения ошибки.

- | Для перезапуска аварийно завершеного задания группы ресурсов кластера, а также для перезапуска
- | поддержки кластера на узле можно воспользоваться командойИзменить восстановление кластера
- | (CHGCLURCY).
- | 1. CHGCLURCY CLUSTER(EXAMPLE)CRG(CRG1)NODE(NODE1)ACTION(*STRCRGJOB) - эта команда вызовет отправку
- | задания группы ресурсов кластера CRG1 на узле NODE1. Для того чтобы на узле NODE1 можно было
- | запустить группу ресурсов кластера, на нем должна быть активна служба ресурсов кластера.
- | 2. Запустите службу ресурсов кластера на узле.

Если вы пользуетесь для работы с кластером специализированным продуктом, обратитесь к документации по этому продукту.

Понятия, связанные с данным

“Пользовательские очереди и структура заданий” на стр. 115

При управлении кластером необходимо разбираться в структуре заданий и принципе работы пользовательских очередей.

Задачи, связанные с данной

“Завершение работы узла кластера” на стр. 103

Запуск и завершение работы узла кластера сводятся к запуску и завершению работы службы ресурсов кластера на узле.

“Запуск узла кластера” на стр. 103

Операция запуска узла кластера заключается в запуске службы ресурсов кластера на узле. Начиная с версии 3, узел может самостоятельно запустить службу ресурсов кластера и войти в кластер в случае, если ему удастся найти хотя бы один узел, уже входящий в кластер.

Восстановление поврежденного объекта кластера

Повреждение объектов кластера как таковых маловероятно, но иногда получают повреждения объекты службы ресурсов кластера.

Если система работает в составе кластера, она попытается восстановить поврежденные объекты с другого активного узла. При этом будут выполнены следующие действия:

Если поврежден внутренний объект кластера:

1. Работа узла, на котором поврежден объект, будет завершена.
2. Если в кластере есть еще хотя бы один активный узел, данный узел автоматически перезапустится и вернется в кластер. В процессе возврата узла в кластер поврежденный объект будет исправлен.

Если повреждена CRG:

1. Узел, на котором расположен поврежденный CRG, аварийно завершит все операции, выполняемые в данный момент над этим CRG. После этого система попытается автоматически восстановить поврежденный CRG с другого активного узла кластера.
2. Если в домене восстановления CRG есть еще хотя бы один активный узел, то CRG будет успешно восстановлен. В противном случае задание CRG будет завершено.

Если системе не удастся найти ни одного другого активного узла кластера, то процедуру исправления нужно будет выполнить вручную.

Если поврежден внутренний объект кластера:

Будет выдано сообщение о внутренней ошибке кластера (CPFBB46, CPFBB47 или CPFBB48).

1. Завершите работу службы ресурсов кластера на узле, на котором поврежден объект.
2. Запустите службу ресурсов кластера на этом узле. Данную операцию нужно выполнить с другого активного узла кластера.
3. Если после выполнения шагов 1 и 2 неполадка не будет устранена, удалите узел с поврежденным объектом из кластера.

4. Вновь добавьте этот узел в кластер и в нужные домены восстановления CRG.

Если повреждена CRG:

Будет выдано сообщение о том, что поврежден объект (CPF9804).

1. Завершите работу службы ресурсов кластера на узле, на котором поврежден объект.
2. Удалите CRG с помощью команды DLTCRG.
3. Если в кластере больше нет активных узлов из домена восстановления данного CRG, восстановите CRG из резервной копии.
4. Запустите службу ресурсов кластера на узле, на котором был поврежден объект. Эту операцию можно выполнить с любого активного узла.
5. После запуска службы ресурсов кластера все CRG будут автоматически синхронизированы. Вам может потребоваться вновь создать CRG, если в его домене восстановления нет ни одного узла, кроме данного.

Восстановление кластера после полного выхода из строя

Информация из этого раздела, а также руководство Резервное копирование и восстановление помогут вам восстановить систему после полного выхода из строя в результате непредвиденного отключения питания.

Пример 1: Восстановление данных в исходной системе

1. Для предотвращения противоречий в информации о домене устройств LIC и i5/OS рекомендуется установить LIC с помощью опции 3 (Установить LIC и восстановить конфигурацию).

Примечание: Для выполнения этой операции в системе должны быть установлены те же диски, что и на момент сохранения системы. Это требование не распространяется только на загрузочный диск. Кроме того, восстанавливаться должен тот же выпуск операционной системы, который был установлен.

2. После установки Лицензионного внутреннего кода перейдите к процедуре *Восстановление конфигурации дисков* в главе 5 книги *Резервное копирование и восстановление*. Эта процедура поможет вам обойтись без повторной настройки ASP.
3. После восстановления системы нужно запустить службу ресурсов кластера на локальном узле с другого активного узла. В результате этой операции на узел будет загружена вся информация о кластере.

Пример 2: Восстановление данных в другой системе

После восстановления системы и проверки протокола задания на отсутствие ошибок нужно выполнить следующие действия по настройке домена устройств кластера.

1. С узла, на котором восстановлена система, нужно удалить службу ресурсов кластера.
2. С любого активного узла кластера выполните следующие действия:
 - a. Удалите восстановленный узел из кластера.
 - b. Добавьте восстановленный узел в кластер.
 - c. Добавьте восстановленный узел в домен устройств.
 - d. Создайте CRG или добавьте узел в домен восстановления.

Задачи, связанные с данной

“Резервное копирование и восстановление кластеров” на стр. 117

Несмотря на то, что кластеры обладают высокой устойчивостью, ни в коем случае не следует пренебрегать резервным копированием.

Информация, связанная с данной

Резервное копирование и восстановление

Восстановление кластера после стихийных бедствий

Стихийные бедствия могут привести к уничтожению всех узлов кластера, и тогда вам придется полностью восстанавливать кластер.

На случай возникновения такой ситуации рекомендуется сохранить информацию о конфигурации кластера и держать под рукой распечатку этой информации.

Задачи, связанные с данной

“Резервное копирование и восстановление кластеров” на стр. 117

Несмотря на то, что кластеры обладают высокой устойчивостью, ни в коем случае не следует пренебрегать резервным копированием.

Восстановление кластера из резервной копии

При нормальной работе восстановление с резервной копии обычно не требуется.

Оно необходимо только в случае аварии и потери данных на всех узлах кластера. Если нечто подобное произойдет, то следуйте обычным процедурам восстановления данных из резервной копии.

Задачи, связанные с данной

“Резервное копирование и восстановление кластеров” на стр. 117

Несмотря на то, что кластеры обладают высокой устойчивостью, ни в коем случае не следует пренебрегать резервным копированием.

Информация, связанная с данной

Резервное копирование и восстановление

Часто задаваемые вопросы об управлении кластером с помощью Навигатора iSeries

Ответы на часто задаваемые вопросы по созданию кластеров и управлению ими с помощью Навигатора iSeries.

Фирма IBM разработала графический интерфейс для создания кластеров и управления ими в Навигаторе iSeries. Этот интерфейс поставляется в составе Компонента 41 (HA Switchable Resources). Дополнительные сведения об этом интерфейсе приведены в разделе Навигатор iSeries - Управление кластером.

Ниже приведен список часто задаваемых вопросов по работе с интерфейсом управления кластером Навигатора iSeries.

Общие

1. Где можно найти контрольную таблицу по процедуре создания кластера?

Навигатор iSeries - Управление кластером

1. Где находится функция Кластеры в интерфейсе Навигатора iSeries?
2. Как создать кластер?
3. Как связаны папка Кластеры и группа систем в приложении Централизованное управление?
4. Я уже создал кластер из нескольких систем iSeries. Как сделать, чтобы им можно было управлять с помощью Навигатора iSeries?
5. Ни один из узлов кластера не запущен. Какой узел запускать первым?
6. Имеет ли значение порядок запуска узлов?
7. Что означает столбец “Текущий главный узел” в папках перемещаемого программного и аппаратного обеспечения?
8. Как найти CRG устройств в Навигаторе iSeries?
9. Как найти CRG приложений в Навигаторе iSeries?
10. Как найти CRG данных в Навигаторе iSeries?
11. Можно ли определить состояние CRG устройств без перехода в папку Перемещаемое аппаратное обеспечение? Как это сделать?

Связь

1. Какой IP-адрес применяется функцией Кластеры Навигатора iSeries для связи с узлами кластера? Что именно применяется: IP-адрес или имя узла?

Защита

1. Почему большинство меню папки Кластеры Навигатора iSeries недоступны?
2. Применяются ли в функции Кластеры Навигатора iSeries значения приложения Администрирование приложений?
3. Почему функция Кластеры Навигатора iSeries показывает меню входа в систему для узлов кластера?

Неполадки

1. Почему в приложении Централизованное управление нет папки Кластеры?
2. Почему существующий кластер не показан в папке Кластеры?
3. Почему в папке Кластеры показаны устаревшие сведения о состоянии?
4. Почему не произошел автоматический перенос устройств или приложений?
5. Получено сообщение о повреждении объекта. Какие действия следует предпринять?
6. Для просмотра списка IP-адресов узлов применяется кнопка "Обзор" мастера. Почему в окне обзора показаны не все IP-адреса?
7. Почему большинство меню папки Кластеры Навигатора iSeries недоступны?
8. Мастер "Создать кластер" показал окно "Создать кластер - Не найдено переносимое программное обеспечение". О чем это говорит?
9. Один из узлов кластера находится в состоянии "Нет связи". Что делать?

Общие

Где можно найти контрольную таблицу по процедуре создания кластера?

Да. Воспользуйтесь таблицей Контрольная таблица конфигурации кластера.

К списку вопросов

Управление кластером с помощью Навигатора iSeries: Где находится функция Кластеры в интерфейсе iSeries?

Интерфейс управления кластером Навигатора iSeries поставляется в пакете программного обеспечения IBM iSeries Access. Приложение Кластеры находится в папке Централизованное управление Навигатора iSeries. Дополнительные сведения об этом интерфейсе приведены в разделе Навигатор iSeries - Управление кластером.

К списку вопросов

Как создать кластер?

Проще всего создать кластер с помощью мастера создания кластера продукта Навигатор iSeries:

1. Запустите Навигатор iSeries и откройте **Централизованное управление**.
2. Щелкните правой кнопкой на значке **Кластеры** и выберите пункт **Создать кластер**.
3. Следуйте инструкциям мастера.

Подробные инструкции по созданию и настройке кластеров приведены в разделе Настройка кластера.

К списку вопросов

Как связаны папка Кластеры и группа систем в приложении Централизованное управление?

При создании кластера с помощью интерфейса управления кластером Навигатора iSeries для сервера Централизованного управления автоматически создается группа систем. Группе систем присваивается то же имя, что указано для кластера, а набор конечных систем совпадает с множеством узлов кластера. Группе систем назначается особый тип объекта для того, чтобы интерфейс управления кластером Навигатора iSeries мог определить, что эта группа систем соответствует кластеру.

Важное замечание: Приложение Централизованное управление оперирует группами систем. Если в Навигаторе iSeries будет изменена система централизованного управления, то у новой системы централизованного управления не будет определений особых групп систем, и поэтому кластеры, созданные ранее, не будут показаны в папке Кластеры.

К списку вопросов

Я уже создал кластер из нескольких систем iSeries. Как сделать, чтобы им можно было управлять с помощью Навигатора iSeries?

Для того чтобы Навигатор iSeries узнал о существовании вашего кластера, выполните следующие действия:

1. Запустите Навигатор iSeries и откройте **Централизованное управление**.
2. Щелкните правой кнопкой на значке **Кластеры** и выберите пункт **Добавить существующий кластер**.
3. В окне **Добавить существующий кластер** укажите любой сервер кластера.
4. Нажмите кнопку ОК.

К списку вопросов

Ни один из узлов кластера не запущен. Какой узел следует запускать первым?

Первым следует запускать тот узел, который последним оставался в состоянии "Запущен". Предположим, что кластер состоял из узлов А и В. Оба узла не запущены в данный момент. Узел В завершил работу позднее, чем узел А. В такой ситуации первым следует запустить узел В, поскольку в нем хранится наиболее актуальная информация о кластере.

К списку вопросов

Имеет ли значение порядок запуска узлов?

Да. Это связано с тем, что наиболее актуальная информация о состоянии кластера находится на том узле, который последним завершил работу. Если первым будет запущен другой узел, то с некоторой вероятностью на нем будет храниться устаревшая информация о кластере. Эта устаревшая информация может быть разослана на другие узлы кластера при их запуске. Предположим, что кластер состоит из узлов А и В, причем узел В завершил работу позднее узла А, и поэтому на нем хранится более актуальная информация о кластере. Предположим, что на узле А находится устаревшая информация о кластере, так как этот узел раньше завершил работу, и первым будет запущен узел А. В этом случае кластер В при запуске вернется в кластер, в котором уже будет активен узел А. Узел А перешлет устаревшую информацию о кластере на узел В, и в результате на обоих узлах будет храниться устаревшая информация о кластере. Поэтому в такой ситуации следует первым запускать узел В. Устаревшая информация о кластере может повлиять на конфигурацию групп перемещаемого аппаратного обеспечения. Если при запуске групп перемещаемых устройств возникнут ошибки, вызванные неправильным выбором главного и резервного узлов, то вам потребуется изменить роли узлов в домене восстановления для того, чтобы они соответствовали текущей принадлежности групп устройств.

К списку вопросов

Что значит столбец "Текущий главный узел" в папках перемещаемого программного обеспечения, перемещаемого аппаратного обеспечения и перемещаемых данных?

В столбце "Текущий главный узел" указан узел, выполняющий в данный момент роль главной точки доступа к ресурсу. В терминологии API это означает, что у данный узел играет роль главного узла в домене восстановления.

К списку вопросов

Как найти CRG устройств в Навигаторе iSeries?

CRG устройств называются группами перемещаемого аппаратного обеспечения и показаны в подпапке **Перемещаемое аппаратное обеспечение** папки Кластеры.

К списку вопросов

Как найти CRG приложений в Навигаторе iSeries?

CRG приложений называются группами перемещаемых приложений и показаны в подпапке **Перемещаемое программное обеспечение** папки Кластеры.

К списку вопросов

Как найти CRG данных в Навигаторе iSeries?

CRG данных называются группами перемещаемых данных и показаны в подпапке **Перемещаемые данные** папки Кластеры.

К списку вопросов

Можно ли определить состояние CRG устройств без перехода в папку Перемещаемое аппаратное обеспечение?

Вместо перехода в папку Перемещаемое аппаратное обеспечение можно открыть новое окно, щелкнув правой кнопкой на папке **Перемещаемое аппаратное обеспечение** и выбрав пункт **Открыть**. Появится отдельное окно с информацией о состоянии CRG устройств. Точно так же можно просмотреть состояние **перемещаемого программного обеспечения и перемещаемых данных**.

К списку вопросов

Связь: Какой IP-адрес применяется функцией Кластеры Навигатора iSeries для связи с узлами кластера? Что именно применяется: IP-адрес или имя узла?

В папке Кластеры показан столбец "Сервер", в котором приведены сведения о кластерах. Имя сервера также показано в окне свойств кластера. Серверы, показанные в столбце "Сервер", - это узлы, применяемые Навигатором iSeries для связи с кластером. Обратите внимание, что этот сервер имеет значение только для Навигатора iSeries; для узлов кластера значение, показанное в этом столбце, не имеет никакого значения. Сервер, применяемый интерфейсом управления кластером Навигатора iSeries, никак не связан с текущим сервером централизованного управления.

Если узел, применяемый Навигатором iSeries для связи с кластером, прекратит работу, вместо него можно выбрать другой узел.

Для того чтобы изменить сервер, применяемый Навигатором iSeries для связи с кластером, выполните следующие действия:

1. Запустите Навигатор iSeries и откройте **Централизованное управление**.
2. Откройте **Кластеры**.
3. Щелкните правой кнопкой на кластере и выберите опцию **Изменить сервер**.

К списку вопросов

Безопасность: Почему большинство меню папки Кластеры Навигатора iSeries недоступны?

Набор доступных операций зависит от состояния кластера в целом и его отдельных объектов. Например, нельзя остановить узел, который уже не работает. Нельзя добавить узел в кластер, в который уже входит максимальное возможное количество узлов (4). Причины, по которым недоступны отдельные пункты меню, приведены в электронной справке по конкретным задачам.

Для выполнения ряда операций требуются особые права доступа. В случае работы с Навигатором iSeries права доступа *SECOFR дают полный доступ к операциям кластера и управлению им. При выполнении операций интерфейса управления кластером Навигатора iSeries действуют текущие права на управление приложениями, установленные для текущей системы Централизованного управления.

Дополнительные сведения приведены в разделе Администрирование приложений.

К списку вопросов

Применяются ли в функциях Навигатора iSeries для работы с кластерами значения администрирования приложений?

Да. При выполнении операций управления кластером с помощью Навигатора iSeries действуют текущие права на управление приложениями, установленные для текущей системы Централизованного управления.

В Навигаторе iSeries предусмотрено два класса доступа: **Операции над кластером** и **Администрирование кластера**

С правами на **Операции над кластером** можно выполнять следующие действия:

- Просматривать состояние кластера
- Запускать и останавливать узлы
- Запускать и остановить переносимое аппаратное и программное обеспечение
- Выполнять принудительный перенос программного и аппаратного обеспечения

С правами на **Управление кластером** можно выполнять следующие действия:

- Создавать и удалять кластеры
- Добавлять и удалять узлы
- Добавлять и удалять перемещаемое программное и аппаратное обеспечение, а также пулы дисков
- Изменять свойства перемещаемого программного и аппаратного обеспечения

К списку вопросов

Почему функция Кластеры Навигатора iSeries показывает меню входа в систему для узлов кластера?

В некоторых случаях Навигатор iSeries пытается связаться со всеми узлами кластера. Это зависит от состояния кластера. Если навигатору iSeries требуется установить связь с узлом, то сначала проверяется наличие открытых соединений в кэше входа в систему Навигатора iSeries. Если в кэше нет рабочего соединения, то пользователю будет предложено вновь войти в систему. Если вы закроете окно входа в систему, то Навигатор iSeries предоставит пользователю возможность управления кластером. Однако для выполнения ряда операций Навигатору iSeries необходима возможность связи с узлами.

К списку вопросов

Устранение неполадок: Почему в приложении Централизованное управление нет папки Кластеры?

Возможно, продукт iSeries Access для Windows установлен не полностью. Была выполнена базовая установка или настраиваемая установка с неправильными параметрами. Инструкции по установке приведены в разделе iSeries Access.

К списку вопросов

Почему существующий кластер не показан в папке Кластеры?

Это означает, что для данного кластера не создана группа систем централизованного управления. Эта группа создается при создании кластера с помощью интерфейса управления кластером Навигатора iSeries и при добавлении кластера в папку Кластеры с помощью функции "Добавить существующий кластер..." Для просмотра списка групп систем откройте значок **Группы систем** в приложении Централизованное управление. Группы систем, соответствующие кластерам, будут показаны как "иные", однако это не означает, что все группы, помеченные как "иные", соответствуют кластерам.

К списку вопросов

Почему в папке Кластеры показаны устаревшие сведения о состоянии?

Навигатор iSeries получает сведения о кластере от узлов кластера. Эти сведения отображаются в окне Навигатора iSeries в виде моментальной копии. При этом по умолчанию Навигатор не выполняет автоматическую регулярную проверку состояния. Для того чтобы получить текущие сведения о кластере, лучше всего воспользоваться функцией Обновить. Откройте в Навигаторе iSeries меню **Вид** и выберите **Обновить**. Кроме того, можно настроить в Навигаторе iSeries автоматическое обновление сведений о состоянии.

К списку вопросов

Почему не произошел автоматический перенос устройств, программ или данных?

Скорее всего, на кластере не был запущен соответствующий перемещаемый ресурс (CRG). Другими словами, перемещаемый ресурс не находился в состоянии "Запущен" перед выполнением автоматического переноса ресурсов. Для выполнения автоматического переноса ресурс должен быть запущен.

К списку вопросов

Получено сообщение о повреждении объекта. Какие действия следует предпринять?

Скорее всего, вы получили сообщение вида CPF811C Повреждена пользовательская очередь QUGCLUSRQ из библиотеки QCLUMGT

Вариант 1: удалить и восстановить поврежденный объект. Это возможно только в случае, если ранее объект сохранялся.

Вариант 2: удалить поврежденный объект. Например, если повреждена очередь QUGCLUSRQ из библиотеки QCLUMGT, удалите эту очередь. Затем добавьте существующий кластер в Навигатор iSeries. При добавлении кластера функции GUI проверят наличие объектов и создадут их, если они еще не существуют. Дополнительная информация о добавлении кластера приведена в разделе Как сделать, чтобы можно было управлять кластером с помощью Навигатора iSeries? .

К списку вопросов

Для просмотра списка IP-адресов узлов применяется кнопка "Обзор" мастера. Почему в окне обзора показаны не все IP-адреса?

Данный список приводится лишь для ориентира. Могут использоваться и другие адреса. Любому сетевому интерфейсу кластера можно назначить любой адрес. При этом помните, что если Навигатор iSeries не сможет связаться с кластером через указанный основной IP-адрес, то возникнет ошибка. С помощью этого IP-адреса Навигатор iSeries подключается к узлам кластера.

К списку вопросов

Мастер "Создать кластер" показал окно "Создать кластер - Не найдено переносимое программное обеспечение". О чем это говорит?

Это сообщение не является признаком ошибки. Оно обозначает, что интерфейсу Навигатора iSeries не удалось найти перемещаемое программное обеспечение, которое можно было бы автоматически установить с помощью мастера. Навигатор iSeries может автоматически устанавливать только переносимое программное обеспечение, удовлетворяющее требованиям архитектуры i5/OS для приложений с поддержкой кластеров. Более того, Навигатор iSeries поддерживает не все приложения, соответствующие этой архитектуре.

К списку вопросов

Один из узлов кластера находится в состоянии "Нет связи". Что делать?

Если связь между узлами кластера будет прервана и не удастся подтвердить выход потерянных узлов из строя, кластер разделится на несколько частей. Дополнительная информация приведена в разделе Ошибки, связанные с разделением кластера.

В некоторых случаях выдается сообщение о разделе кластера, хотя в действительности просто узел вышел из строя. Это происходит в случаях, когда служба теряет связь с некоторыми узлами кластера и не может определить, вышли ли они из строя. Если это требуется, вы можете пометить узлы как сбойные вручную. Дополнительные сведения приведены в разделе Перевод разделившихся узлов в разряд сбойных.

К списку вопросов

Сервисные центры по поддержке кластеров

В этом разделе приведена контактная информация сервисного центра IBM.

Если вы не уверены, целесообразно ли применение кластеров в вашей среде, или у вас возникнут неполадки при работе с кластерами, обратитесь в фирму IBM:

- Для получения консультаций и советов по работе с кластерами обратитесь в отдел IBM Continuous Availability Center центра iSeries Technology Center по следующему адресу: rchelst@us.ibm.com.
- По прочим вопросам обращайтесь в организацию, в которой вы приобрели кластерное программное обеспечение, либо в фирму IBM по телефону 1-800-IBM-4YOU (1-800-426-4968).

Задачи, связанные с данной



“Настройка кластеров” на стр. 99




Инструкции по настройке кластеров.

| Информация, связанная с кластерами

| Поиск информации, связанной с кластерами.

| Руководства по выполнению задач

- | • Data Resilience Solutions for IBM i5/OS High Availability Clusters  .
- | • Clustering and IASPs for Higher Availability 

- | • High Availability on the AS/400 System: A System Manager's Guide 
- | • IBM eServer iSeries Independent ASPs: A Guide to Moving Applications to IASPs 
- | • The System Administrator's Companion to AS/400 Availability and Recovery 

| Сайты в Internet

- | • Кластеры и высокая готовность  (www.ibm.com/servers/eserver/series/ha)
| Сайт фирмы IBM, посвященный кластерам и обеспечению высокой готовности

| Сохранение файлов в формате PDF

| Для того чтобы сохранить файл PDF на вашей рабочей станции, выполните следующие действия:

- | 1. Правой кнопкой мыши щелкните на файл PDF Web-браузере (нажмите правой кнопкой на ссылке, приведенной выше).
- | 2. Если вы работаете в Internet Explorer, выберите **Сохранить объект как...** В Netscape Communicator выберите **Сохранить ссылку как...**
- | 3. Перейдите в каталог, в котором требуется сохранить файл PDF.
- | 4. Нажмите кнопку **Сохранить**.

| Загрузка продукта Adobe Acrobat Reader

| Для просмотра и печати файлов в формате PDF необходима программа Adobe Acrobat Reader. Вы можете загрузить ее с Web-сайта компании Adobe по адресу <http://www.adobe.com/products/acrobat/readstep.html>

| (www.adobe.com/products/acrobat/readstep.html)  .

Лицензия и отказ от гарантий на предоставляемый код

Компания IBM предоставляет вам неисключительную лицензию на использование примеров исходного кода программ, с помощью которых вы можете создавать аналогичные функции в соответствии с вашими конкретными потребностями.

| ЗА ИСКЛЮЧЕНИЕМ УСТАНОВЛЕННЫХ ЗАКОНОМ ГАРАНТИЙ, ОТКАЗ ОТ КОТОРЫХ
| НЕВОЗМОЖЕН, ФИРМА ИВМ И РАЗРАБОТЧИКИ И ПОСТАВЩИКИ ЕЕ ПРОГРАММ НЕ ДАЮТ
| НИКАКИХ ГАРАНТИЙ И ОБЯЗАТЕЛЬСТВ, НИ ЯВНЫХ, НИ ПОДРАЗУМЕВАЕМЫХ, ВКЛЮЧАЯ,
| НО НЕ ОГРАНИЧИВАЯСЬ ЭТИМ, ПОДРАЗУМЕВАЕМЫЕ ГАРАНТИИ И ОБЯЗАТЕЛЬСТВА
| ОТНОСИТЕЛЬНО КОММЕРЧЕСКОЙ ЦЕННОСТИ, ПРИГОДНОСТИ ДЛЯ КАКОЙ-ЛИБО
| КОНКРЕТНОЙ ЦЕЛИ И СОБЛЮДЕНИЯ АВТОРСКИХ ПРАВ, ПО ОТНОШЕНИЮ К ПРОГРАММАМ
| И ТЕХНИЧЕСКОЙ ПОДДЕРЖКЕ, ЕСЛИ ТАКОВЫЕ ПРЕДОСТАВЛЯЮТСЯ.

| НИ ПРИ КАКИХ ОБСТОЯТЕЛЬСТВАХ ФИРМА ИВМ И РАЗРАБОТЧИКИ И ПОСТАВЩИКИ ЕЕ
| ПРОГРАММ НЕ НЕСУТ ОТВЕТСТВЕННОСТЬ НИ ЗА КАКОЕ ИЗ СЛЕДУЮЩИХ СОБЫТИЙ, ДАЖЕ
| ЕСЛИ ОНИ БЫЛИ ЗАРАНЕЕ ИНФОРМИРОВАНЫ О ВОЗМОЖНОСТИ НАСТУПЛЕНИЯ ЭТИХ
| СОБЫТИЙ:

- | 1. ПОТЕРЯ ИЛИ ПОВРЕЖДЕНИЕ ДАННЫХ
- | 2. СПЕЦИАЛЬНЫЕ, СЛУЧАЙНЫЕ, ПРЯМЫЕ ИЛИ КОСВЕННЫЕ УБЫТКИ, ЛИБО ЛЮБЫЕ
| ВЗАИМОСВЯЗАННЫЕ УБЫТКИ; ИЛИ
- | 3. НЕПОЛУЧЕННЫЕ ПРИБЫЛЬ, ВЫГОДА, ДОХОД, ПРЕСТИЖ ИЛИ ПРЕДПОЛАГАЕМАЯ
| ЭКОНОМИЯ СРЕДСТВ.

| В ЗАКОНОДАТЕЛЬСТВАХ НЕКОТОРЫХ СТРАН НЕ ДОПУСКАЕТСЯ ОТКАЗ ИЛИ ОГРАНИЧЕНИЕ
| ОТВЕТСТВЕННОСТИ ЗА ПРЯМЫЕ, СЛУЧАЙНЫЕ ИЛИ ВЗАИМОСВЯЗАННЫЕ УБЫТКИ,

| ПОЭТОМУ НЕКОТОРЫЕ ИЛИ ВСЕ УКАЗАННЫЕ ВЫШЕ ОГРАНИЧЕНИЯ И ОГОВОРКИ МОГУТ
| НЕ ИМЕТЬ СИЛЫ В ВАШЕМ СЛУЧАЕ.

Приложение. Примечания

Настоящая документация была разработана для продуктов и услуг, предлагаемых на территории США.

В других странах компания IBM может не предлагать продукты, услуги и функции, описанные в этом документе. За дополнительной информацией о доступных продуктах и услугах обратитесь в местное представительство компании IBM. Ссылки на продукты, программы и услуги компании IBM не означают, что применять можно только эти программы, продукты или услуги. Вместо них можно использовать продукты, программы и услуги, не являющиеся интеллектуальной собственностью компании IBM. Однако в этом случае ответственность за проверку работы этих продуктов, программ и услуг возлагается на пользователя.

В этом документе описаны объекты, на которые компания IBM может обладать патентами или заявками на патенты. Предоставление вам настоящего документа не означает предоставления каких-либо лицензий на эти патенты. Запросы на приобретение лицензий можно отправлять по следующему адресу:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Запросы на лицензии, связанные с информацией DBCS, следует направлять в отдел интеллектуальной собственности в местном представительстве IBM или в письменном виде по следующему адресу:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

Следующий абзац не относится к Великобритании, а также к другим странам, в которых это заявление противоречит местному законодательству: ФИРМА INTERNATIONAL BUSINESS MACHINES CORPORATION ПРЕДОСТАВЛЯЕТ НАСТОЯЩУЮ ПУБЛИКАЦИЮ НА УСЛОВИЯХ “КАК ЕСТЬ”, БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ЭТИМ, НЕЯВНЫЕ Г ГАРАНТИИ СОБЛЮДЕНИЯ ПРАВ, КОММЕРЧЕСКОЙ ЦЕННОСТИ И ПРИГОДНОСТИ ДЛЯ КАКОЙ-ЛИБО ЦЕЛИ. В некоторых странах запрещается отказ от каких-либо явных и подразумеваемых гарантий при заключении определенных договоров, поэтому данное заявление может не действовать в вашем случае.

В данной публикации могут встретиться технические неточности и типографские опечатки. В информацию периодически вносятся изменения, которые будут учтены во всех последующих изданиях настоящей публикации. Компания IBM может вносить изменения в продукты и программы, описанные в этой публикации в любое время без предварительного уведомления.

Все встречающиеся в данной документации ссылки на Web-сайты других компаний предоставлены исключительно для удобства пользователей и не являются рекламой этих Web-сайтов. Материалы, приведенные на этих Web-сайтах, не являются частью материалов данного продукта IBM. Ответственность за использование этих Web-сайтов лежит на пользователе.

IBM может использовать и распространять любую предоставленную вами информацию на свое усмотрение без каких-либо обязательств перед вами.

Для получения информации об этой программе для обеспечения: (i) обмена информацией между независимо созданными программами и другими программами (включая данную) и (ii) взаимного использования информации, полученной в ходе обмена, пользователи данной программы могут обращаться по адресу:

IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Такая информация может предоставляться на определенных условиях, включая, в некоторых случаях, уплату вознаграждения.

- | Описанная в этой информации лицензионная программа и все связанные с ней лицензионные материалы
- | предоставляются IBM в соответствии с условиями Соглашения с заказчиком IBM, Международного
- | соглашения о лицензии на программу IBM, Лицензионного соглашения на машинный код IBM или любого
- | другого эквивалентного соглашения.

Все приведенные показатели производительности были получены в управляемой среде. В связи с этим результаты, полученные в реальной среде, могут существенно отличаться от приведенных. Некоторые измерения могли быть выполнены в системах, находящихся на этапе разработки, поэтому результаты измерений, полученные в серийных системах, могут отличаться от приведенных. Более того, некоторые значения могли быть получены в результате экстраполяции. Реальные результаты могут отличаться от указанных. Пользователи, работающие с этим документом, должны удостовериться, что используемые ими данные применимы в имеющейся среде.

Информация о продуктах других изготовителей получена от поставщиков этих продуктов, из их официальных сообщений и других общедоступных источников. Эти продукты не тестируются компанией IBM, поэтому IBM не может подтвердить достоверность уровня производительности, совместимости и других свойств таких продуктов. Запросы на получение дополнительной информации об этих продуктах должны направляться их поставщикам.

Все заявления, касающиеся намерений и планов IBM, могут изменяться и отзываться без предварительного уведомления, и отражают только текущие цели и задачи.

Все приведенные цены являются предполагаемыми текущими розничными ценами IBM и могут быть изменены без предварительного уведомления. Цены дилеров могут отличаться от этих цен.

Эта информация предназначена только для целей планирования. Информация, приведенная в этом документе, может быть изменена к моменту, когда продукт станет доступным.

Эта информация содержит примеры данных и отчетов, применяемых в ежедневных бизнес-операциях. Для наиболее полного их описания в примерах приведены имена сотрудников, компаний, торговых марок и продуктов. Все эти имена являются вымышленными и любые совпадения с именами и адресами реальных компаний являются случайностью.

Лицензия на авторские права:

Эта информация содержит примеры прикладных программ на исходном языке, которые позволяют объяснить технику программирования для различных платформ. Вы можете копировать, изменять и распространять эти программы бесплатно для разработки, применения, сбыта и распространения прикладных программ, соответствующих интерфейсу прикладного программирования платформы, для которой они предназначены. Они не проверялись для работы во всех условиях. Поэтому компания IBM, не может дать гарантию, как явную, так и подразумеваемую, надежности, удобства обслуживания и стабильной работы этих программ.

Каждая копия или часть этих программ или работ, созданных на их основе, должна содержать следующее примечание об авторских правах:

© (название вашей фирмы) (год). Данная программа содержит исходный код из примеров программ, предоставленных компанией IBM. © Copyright IBM Corp. _укажите здесь год или несколько лет_. All rights reserved.

В случае просмотра этой информации с помощью компьютера некоторые фотографии и цветные иллюстрации могут не появиться.

Программный интерфейс

В документах настоящей публикации Кластеры описываются программные интерфейсы для разработки программ в IBM i5/OS.

Товарные знаки

Ниже перечислены товарные знаки International Business Machines Corporation в Соединенных Штатах и/или других странах:

- | 400
- | i5/OS
- | IBM
- | iSeries
- | OS/400
- | Руководства по выполнению задач

- | Intel, Intel Inside (эмблемы), MMX и Pentium являются товарными знаками Intel Corporation в США и/или других странах.

Microsoft, Windows, Windows NT и логотип Windows являются торговыми марками Microsoft Corporation в США и других странах.

Java и все торговые марки Java являются торговыми марками Sun Microsystems Inc. в США и других странах.

- | Linux является товарным знаком Линуса Торвальдса (Linus Torvalds) в Соединенных Штатах и/или других странах.

UNIX является торговой маркой Open Group в США и других странах.

Другие названия фирм, продуктов и услуг могут быть товарными или сервисными знаками других фирм.

Условия

Разрешение на использование публикаций предоставляется на следующих условиях.

Использование в личных целях: Разрешается воспроизведение этих публикаций для личного, некоммерческого использования при условии сохранения в ней всех заявлений об авторских правах. Запрещается распространение, демонстрация и использование этой публикации в качестве основы для последующих произведений, полностью или частично, без явного согласия на то фирмы IBM.

Использование в коммерческих целях: Разрешается воспроизведение, распространение и демонстрация этих публикаций исключительно в пределах организации при условии сохранения в них всех заявлений об авторских правах. Запрещается использование этих публикаций в качестве основы для последующих произведений, а также воспроизведение, распространение и демонстрация этих публикаций, полностью или частично, за пределами предприятия без явного согласия на то фирмы IBM.

За исключением явно оговоренных в данном разрешении случаев, на публикацию и любые содержащиеся в ней данные, программное обеспечение и другие объекты интеллектуальной собственности не предоставляются никакие разрешения, лицензии и права, ни явные, ни подразумеваемые.

Фирма IBM оставляет за собой право в любой момент по своему усмотрению аннулировать предоставленные настоящим разрешением права, если сочтет, что использование этой информации наносит ущерб ее интересам или что указанные инструкции не соблюдаются должным образом.

Загружать, экспортировать и реэкспортировать эту информацию разрешается только при условии полного соблюдения всех надлежащих законов, правил и предписаний, включая все действующие в Соединенных Штатах Америки законы и законодательные акты об экспорте.

ФИРМА IBM НЕ ДАЕТ НИКАКИХ ГАРАНТИЙ ОТНОСИТЕЛЬНО СОДЕРЖИМОГО ЭТИХ ПУБЛИКАЦИЙ. ПУБЛИКАЦИИ ПРЕДОСТАВЛЯЮТСЯ "КАК ЕСТЬ", БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, КАК ЯВНЫХ, ТАК И ПОДРАЗУМЕВАЕМЫХ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ЭТИМ, ПОДРАЗУМЕВАЕМЫЕ ГАРАНТИИ КОММЕРЧЕСКОЙ ЦЕННОСТИ, СОБЛЮДЕНИЯ АВТОРСКИХ ПРАВ И ПРИГОДНОСТИ ДЛЯ КАКОЙ-ЛИБО КОНКРЕТНОЙ ЦЕЛИ.



Напечатано в Дании