January 2006

DB2 Information Management Software

IBM® DB2®
Integrated Cluster Environment V2

Technical backgrounder and setup guide
Version 2.1.1

Boris Bialek,
Belal Tassi

IBM Toronto Lab

1. Introduction

Since its inception, the DB2 Integrated Cluster Environment (DB2 ICE) has set the standard for simple, efficient, and price-sensitive Linux® cluster deployment.

The introduction of IBM DB2 ICE V2 brings simple database clusters to a new level and again reduces the implementation time for each customer. Its introduction delivers a new, never-seen-before level of integration for the core operating system components during deployment, operation, and upgrade phases that is the envy of the database industry.

This paper can be broken down into two parts. The first part - consisting of sections 2 and 3 - describes the operation and functional usage of the various parts of a DB2 ICE V2 cluster and the design behind them. The second part - consisting of sections 4, 5 and 6 - is a detailed implementation guide to assist you in rolling out a similar solution stack if you want to do so.

Note that although this documentation is detailed enough to allow others to implement equivalent solutions, it is meant for reference purposes only - as a simple introduction to a fast and functional method of deploying a DB2 ICE V2 cluster. It is not meant to be a replacement for the extensive IBM documentation, Linux documentation, and HOW-TO guides that are available for each subject.

Lastly, it must also be noted that the tools chain found in a DB2 ICE V2 deployment will continue to evolve over time to functionally enhance the solution. What will not change, however, is the primary focus on the simplicity of the solution - and its direct applicability to an end user environment without any problematic prerequisites.

2. Design Points

Why IBM DB2 ICE V2?

The situation in the marketplace today clearly indicates that a number of customers are looking for a very simple – you may want to call it frugal – implementation of large-scale Linux clusters that does not introduce any unneeded applications into their environment. This imperative makes it necessary to look into utilizing existing pieces, and to design a solution stack fitting to the following very clearly defined goals:

- Rapid deployment of the DB2 ICE cluster with minimal or even no Linux knowledge
- Removal of any “slipping points”, prerequisites, and configuration steps including service configuration or simple security settings on the system
- No additional new tools required besides those shipped in the base Linux distributions and the DB2 delivered content
- A simple “dashboard” view of the state of the Linux cluster
- No user-specific configuration needed beyond the creation of the database objects themselves

There are already a number of solutions available on the market ranging from those that provide simple system configuration to highly sophisticated provisioning and management environments – such as Tivoli® Orchestrator and Provisioner - that would address some of the goals above. However, we are finding that for this set of customers, the added complexity this introduces into the setup is a significant barrier at this time. Furthermore, DB2 ICE V2 differs from the traditional IBM approach in that it is not a new development but rather an offering that utilizes the available “board equipment” of Linux and existing IBM tools.

DB2 ICE V2 focuses on a very specific hardware set prescribed by the DB2 Linux team. It allows IBM factories, business partners, and potential end customers to rapidly deploy DB2 Linux clusters in their environments. Its “building blocks” approach makes it very easy to customize and expand their functionality, and is geared specifically to the minimum customer needs that differentiate it clearly from more embracing and more complex environments such as Tivoli Intelligent Orchestrator® (TIO) or Tivoli Provisioning Manager® **(**TPM). As an example, existing out-of-the-box cluster management kits like Cluster Systems Management (CSM) and XCAT are geared more towards the HPC cluster setup, which is a stateless system environment with a single management node and, therefore, is not currently utilized in the current DB2 ICE offering.

Architecture philosophy

DB2 ICE has always utilized a building blocks approach. So it is natural that the DB2 ICE V2 configuration continues this approach that easily allows for future migration, addition, and removal of components based on the current needs of the market. This allows for a quick reaction to changes in requirements without changing the overall concept. The overall architecture can be seen in Figure 1 below.

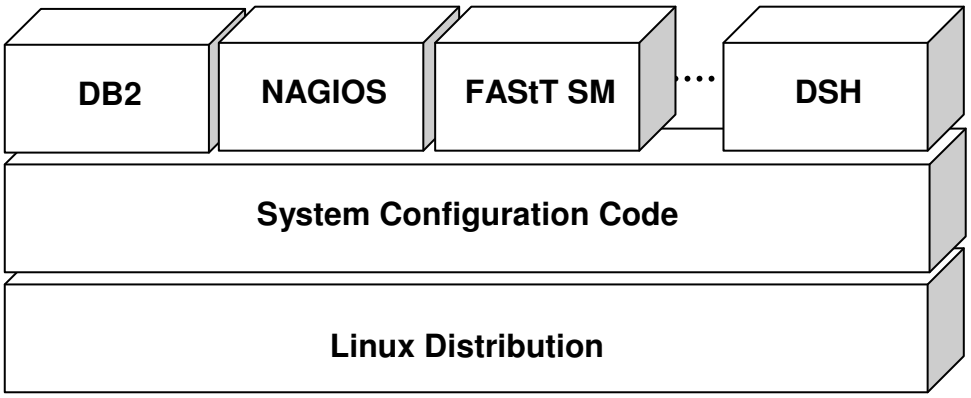


Figure 1: DB2 ICE V2 Software Architecture

The fundamental base of the software architecture is the Linux distribution. DB2 ICE currently supports the two most popular enterprise Linux distributions today: namely, Novell SUSE Linux, and Red Hat Linux[1]. All distributions bundle their specific configuration utilities to deliver a simple base environment - but with an **unconfigured** application space. History shows that often - cynics might say almost every single time - a specific set of Linux drivers or modules are in need of updating to allow full, efficient, and secure operation of DB2. This stage has historically been the stumbling ground for many deployments - and for this reason the decision was made to use fully tested preconfigured and setup images instead of using the "install from scratch and update afterwards" approach. This means the prepared image from the DB2 ICE team can be deployed directly to the next customer environment with an assurance that it will be configured properly. Deltas are addressed through the given rpm package lists for further updates. **At the time of writing, a SUSE version of the setup image is available, with a Red Hat version currently in development.**

On top of the Linux distribution setup sits a set of IBM system configuration code that automatically customizes each specific node for their function in the cluster – performing things such as the configuration of hostnames, TCP/IP addresses and necessary gateway functions. This system configuration layer can then also be used any time afterwards to change the setup if necessary during operation, e.g., when moving a cluster between data centers.

Last are a set of building blocks – each with a specific function - that will vary as time progresses and the needs of our customers change. At this time, this includes blocks for Nagios system management (which is part of the Linux core distributions), IBM FAStT Storage Manager, and of course DB2. The simple distributed shell "dsh" is also included in this layer, while a building block to configure the complete XCAT tool set is currently under development.

The complete ICE cluster is provided as a single image that contains the complete DB2 ICE software stack - all necessary services and software components described above. The specific building blocks are then configured, enabled, or disabled accordingly during deployment of each node in the cluster.

All setup and configuration is performed through one simple configuration file (xcluster.cfg) to keep the overhead as simple as possible. While there are many reasons to introduce more advanced functionality, such as auto-detection of nodes and graphical utilities to manage the nodes, the decision was made to keep this functional "red line" for DB2 ICE V2 and leave more sophisticated tools for TIO/TPM at this time.

IBM DB2 ICE Configurations

IBM DB2 ICE V2 is focusing on offering very practical and scalable cluster configurations utilizing the most common server types on the market. Its current configurations are based on the multi-year experience of the original DB2 ICE. The two server types currently used are:

- 2U servers with enough disk space for internal database disk storage in the server itself – and making any external storage unnecessary
- 1U pizza boxes with two sockets (and up to four cores, at the time of writing) and significant I/O capabilities

Let's look in more detail at the two possible DB2 ICE V2 configurations:

[1] At the time of writing, the Ubuntu distribution - representing the Debian space – has just successfully passed DB2 validation as well.

Configuration #1

Configuration #1 is the most simple and robust implementation with an unbeaten price point for exemplary performance. The design point for this solution was simplicity and scalability. In contrast to the “classical” database solution, in this configuration the database storage is integrated into the actual server and does not utilize external storage at all. The IBM xSeries® x346 server provides outstanding performance with six internal hard drives that are connected via an IBM ServeRAID™ adapter. While the server can handle up to 2 CPU, this configuration is using only 1 CPU to keep the balance between disks and processing performance in check. The sizing per node is between 100 GB and 125 GB raw data.

At first glance, the availability characteristics of configuration #1 looks limited, since a single node outage is not covered. But a more detailed look will show that the redundancy of all key components – especially the redundancy of network, power supplies, and RAID disks – provides solid availability for most enterprise production environments. Furthermore, its effective price point allows the acquisition of a secondary cluster specifically for backup purposes at a very low price point (only a single CPU needs to be acquired).

Configuration #2

This configuration is the “classic” high-end enterprise configuration. Based on the IBM x346 server and the IBM e326 server with external DS4300 Fibre Channel (FC) storage systems, this configuration allows a much higher data density than configuration #1. Furthermore, all components are structured in a redundant array so that a failure is unlikely, and continuous operation is provided even in the failure of a full server node. While the single core IBM x346 server is capable of 250 GB raw data, the IBM e326 is sized with 400 GB raw data per node (or 100 GB per core). The selection of the latter system may introduce the need for an additional storage expansion cabinet.

Generally for configurations with more than 8 nodes, InfiniBand high-speed interconnects are recommended, and for a cluster with 16 or more nodes, they are “required”. This is to ensure the optimal throughput and communication between the nodes. This applies to all the DB2 ICE configurations.

Node types

DB2 ICE allows you to define each machine in your Linux cluster as a specific type of node. This in turn defines how your machine will then be configured during deployment. Here is a short overview of the various node types:

The **system node** has the purpose of managing the overall cluster and to be available as the system management, installation, and configuration arm of the cluster. It can be the failover node for the first administration server as well. The system node features the DHCP daemon needed for the installation of additional nodes, the Nagios front end, the IPMI front end and the parallel shell “dsh” needed to configure the cluster in the case of major reconfigurations and changes. (Note: Some people may compare it to the management node for HPC but the system does not perform scheduling nor does it hold the management node information for DB2 that resides on the first administration node).

The **administration node** is specifically added to the cluster as a DB2 coordinator node – that is the node that users will connect to. The first administration node is the system catalog node that contains the exported DB2 home directory. Otherwise, an administration node is nothing more than a data node without data. It is recommended to have the administration node be equipped with 2 CPU and more memory in the case of larger user numbers connecting to the system.

Data nodes are the workhorses inside the DB2 cluster. They each operate one subset of the overall database cluster and contain no additional functions. Users normally should not connect to those nodes, and they need no external connectivity whatsoever – in many cases, for simplicity in management and security, it is best not to connect those systems to the existing external network.

The **load node –** if present - is responsible for processing all of the extract, transformation, and load (ETL) work for the cluster. It is a recommended best practice to isolate ETL on a node in the cluster to be used exclusively for ETL work, to ease the burden of workload management especially when an ETL tool such as Data Stage, Informatica, or Ab Initio is being used. It is configured similarly to a non-first administration node and a backup node.

The last node type is a **backup node**. In contrast to the "classical" method of attaching all nodes to a Fibre Channel based backup, we recommend the backup to be executed via the high speed interconnect to a smaller number of nodes that have low price SAT storage attached. This form of using the DB2 backup allows a very efficient high-speed backup of multiple TB while keeping the backup window itself small. After backup completion, this node can then dump the data to secondary storage devices. This asynchronous backup to a backup server and then later dumping to tape can be used as well for ETL operations as long as realistic expectations are there for the actual read/write performance of the SAT storage. Naturally, the system can be equipped with either a FC subsystem or SCSI attached concatenated disks (JBODs), but this would be a significant higher price.

Operating Systems

DB2 ICE V2 currently recommends either Novel SUSE Enterprise Server 9 or Red Hat Enterprise Server 4 in their base configurations. In addition, other Linux distributions – most recently the popular Debian based Ubuntu distribution – are passing DB2 validation on a regular basis. Please check the DB2 Linux Validation website[2] as only the listed Linux distributions have been tested to ensure they will work with DB2 at this time.

With the introduction of DB2 UDB V8.2.3, and this introduction of 64-bit DB2 Query Patroller, any major reason for 32-bit operating system usage has disappeared – and in most cases will hinder the exploitation of full system performance. Hence, the base DB2 ICE configurations always utilize 64-bit machines and software versions (where possible) as well as the Linux 2.6 kernel and upwards, as provided by the Linux distributions.

[2] http://www.ibm.com/software/data/db2/linux/validate/

Onboard Linux tools

All Linux distributions come with a broad variety of tools that when configured allow for an appliance-like behaviour for the database cluster. In DB2 ICE V2, we utilize Nagios for the first time as a basic management utility that allows the monitoring of the system resources and system status. Nagios 1.2 is the default version delivered with the Novell and RedHat distributions. While newer versions of Nagios are available, the decision was made to stay with the versions currently supported by the base distributions instead of going the route of self-maintenance.

In addition, the DB2 ICE V2 solution includes a number of simple tools that enhance the functionality of the overall solution, e.g., the ServeRAID manager, db2top, dsh, Firefox, FAStT Storage Manager, etc.

3. Implementation Flow

In this section, the rollout of a DB2 ICE cluster is described in a small number of simple steps that follow the physical assembly of the hardware[3].

Before we start, a quick hardware check should be performed. While the cluster 1350 is thoroughly validated in the factory, it is still recommended to quickly check over the various BIOS levels applied because sometimes even between assembly and delivery critical fixes may be released. During this quick check, all the RAID volumes can be created if needed and a short power up test can occur.

The actual installation of each node follows:

1.) The first component needed in the DB2 ICE V2 cluster is an installation server. This server can either be a properly configured system node[4] or a machine external to the cluster such as a laptop. If the system node will be used as an installation server, it must first be created itself through an external installation server. So in either case you need to have an external installation server - at least for the first node created.

    A native Linux notebook or even a VMware partition is enough to fulfill this role. If no external system node is available, the core kit is available from the IBM DB2 ICE team as a VMware image residing in a single drive that can be attached to any existing VMware Workstation 5 Linux system.

2.) The second step is the gathering and validation of the existing hardware and network layouts. To do so, the primary MAC addresses of the Ethernet network adapter are located (it is highly recommended to have this information written on the front of a large cluster anyway), and each node is defined as an administration, data, load, or backup node, depending on its intended function. The list is entered into the xcluster.cfg file with the selected parameters for the network and function inside the cluster.

[3] The IBM Linux cluster 1350 comes completely preassembled, so there is actually no physical assembly required in this case beyond connection to power and the corporate network.
[4] At the time of writing, configuration of a system node to act as a DB2 ICE installation server must be performed manually.

3.) Now the nodes can be booted. They connect to your installation server (either the newly created system node or an external machine) and start their own installation. For logical flow, it is a good idea to install the first administration server, holding the management node and NFS share information for the database *first*, and wait until it is mostly completed before starting the other nodes.

4.) During the installation, the operating system, DB2 and all higher-level building blocks are installed and all provided information is configured, and the setup is completed. A check through the Nagios console on the system node (available from any Web browser) should show all the correctly installed nodes. In the case of wrong TCP/IP addresses - or, in fact, errors of any kind - each node can be installed as many times as possible until the setup is satisfactory.

5.) As a last step, any necessary installation checks are executed and the setup is complete!

Looking at steps 3 and 4 above in more detail, during deployment each node is going through the following installation procedure[5]:

i. The node is booting cold into network boot (depending on the BIOS of the system, this happens automatically or needs to be initiated manually *once* usually by pressing F12 during bootup).

ii. The node picks up a DHCP network boot address from the system installation server together with the direction for the network boot to load the file “pxelinux.0” via TFTP.

iii. The PXE boot strap initiates the download from the PXE configuration file, which itself is just a pointer to the Linux kernel, INITRD file and the appended boot parameter.

iv. The Linux kernel and the INITRD file is transferred via TFTP to the machine and the normal boot strap begins the installation based on the given installation parameters without any user interaction.

v. As part of the installation, the primary hard drive is configured with an emergency Linux operating system and the actual space and directories for the production image.

vi. After the core installation of the rescue system, the production system is *copied* to the given drive structure. In addition, the xcluster.cfg file is transferred from the system node to each machine for temporary configuration (this – as well as the installation log - can be found in the /root/.db2ice directory).

vii. TCP/IP address, hostname, and all network parameters are set and the specific packages and services are enabled – all based on the information provided in the xcluster.cfg file.

[5] Interested parties in the specific processes and their function should read the usual Linux operating system documentation.

4. Implementation Details

Building the default image

The ultimate target of our deployment is the creation of an up-and-running DB2 database for the customer, and the creation of a production image is the first step towards this goal. The default image provided by the IBM DB2 ICE team has been designed with the following criteria:

- Ensure the broadest possible support for boot devices in the kernel including the normal Adaptec SCSI controller used by IBM, the IPS driver for the ServeRAID adapter, and support for BusLogic and QLogic drivers.
- All needed packages are on one single image that is used by all node types to simplify the setup and keep the distributed setup kits small.
- All other components and drivers are updated to the latest given **stable** and **tested** level as recommended by the DB2 Linux team.

At the time of writing, a SUSE version of the setup image is available, with a Red Hat version currently in development.

The disk layout for each system is identical (even if it has external storage components available) and makes the setup logically sound. Note, : in the table below, we use the term /dev/sda generically to refer to all device types (i.e., for example in the IBM BladeCenter® Serial ATA disks are used with /dev/hda identifier)[6].

Disk/Partition	Size (MB)	Mount point	Purpose
All nodes			
/dev/sda1	100	/boot	Kernel and boot loader partition
/dev/sda2	5000	/ (RESCUE)	Small rescue system if everything breaks
/dev/sda3	20000	/ (PROD)	Production OS core
/dev/sda5	16000	Swap	Swap space 2x default memory
/dev/sda6	10000	/var	Independent /var to preserve the OS stability
/dev/sda7	10000	/tmp	Independent /tmp to preserve the OS stability
/dev/sda8	Rest of sda	/db2data	DB2 home directory on this machine
Data node	**with FC**		
/dev/sdb1	360000	/db2data1	Six spindles RAID5 plus one hot spare
/dev/sdc1	360000	/db2data2	
/dev/sdd1	360000	/db2data3	
/dev/sde1	360000	/db2data4	
Backup node	**with SATA**		
/dev/sdb1	3000000	/backup1	Always 3TB building blocks in DS4100

[6] As well the references for the backup node and fibre channel configuration are for reference only at this time because they need to be manually configured depending on storage used at the customer site. The partition number and rough sizes should be kept.

The setup will be identical for either RedHat or SUSE with respect to the disk layout and the additional “non-default” packages added. The actual build of the image is a full backup of a “golden” system from the root of the system with the simple command:

```
tar -cpzvf image001.tgz / --exclude=proc/kcore
```

This single golden image file is the base for all further setups, regardless of the node type. At the time of writing, the file contained the following additional components added to a default SUSE Enterprise Linux install:

Package	Source/Owner	Current Level
IBM DB2 UDB ESE V8.2.2	IBM	DB2 V8.1.3.96 (Fixpack 10)
ServeRAID Manager	IBM	7.12.02
ServeRAID agent	IBM	RaidMan-7.10-18
IB driver	Voltaire	ibhost-v3.4.5_12
IB driver	Mellonox	1.8.0
Distributed Shell (dsh)	IBM CSM	csm.dsh-1.4.1.3-116
Apache	Apache Foundation	apache-1.3.29-71.15
Nagios	http://www.nagios.org	Nagios-1.2-73.1
FAStT Storage Manager	IBM	9.1 V15
IBM SDK for Java™	IBM	IBM SDK for Java 2-1.4.2-1.0
SanSurfer Pro 2.0.30 build 58	QLogic	2.0.30 build 58
AutoYast 2 (SUSE ONLY)	Novell	autoyast2-2.9.53-0.2
Q Logic Drivers	QLogic	qla2xxx 8.01.01 qla2300 8.01.01 qla2xxx_conf 8.01.01
Open IPMI	http://ipmitool.sourceforge.net/	OpenIPMI-1.3.11-0.2
IPMI Scripts	IBM	1.0

The following changes were also made to the image itself:

File	Change
~db2inst1/.ssh/id_rsa ~root/.ssh/id_rsa /sbin/.ssh/id_rsa	Generate ssh keys for these two users and the system daemon
/usr/RaidMan/RaidAgnt.pps	Ensure this line exists: agent.enable.security=false
/etc/sysctl.conf	Ensure these lines exist: vm.swappiness = 0 vm.dirty_ratio = 10 vm.dirty_background_ratio = 5
Nagios configuration files	Many changes were made to support DB2 UDB out of the box.
/etc/sysconfig/kernel	Ensure this line exists: INITRD_MODULES = “mptbase mptscsih ips qla2xxx_conf qla2xxx qla2300”

/etc/modprobe.conf.local	Change the line "options qla2xxx ql2xfailover=0 configrequired=0" to: "options qla2xxx ql2xfailover=1 ConfigRequired=0"

All the services described below can be configured for permanent operation with the command:

```
chkconfig <service name> on
```

This command will start the services at the next reboot automatically. The following command will turn off the service permanently:

```
chkconfig <service name> off
```

Configuration of the Installation server

To prepare the installation server to deploy the DB2 ICE V2 image, you need to configure three services below. Note that if you are using the VMWare image provided by the IBM DB2 ICE team , the script "**reset.sh"** found in the /tftpboot directory will conveniently restart all three necessary services for you.

1.) DHCP server: The DHCP daemon provides the initial boot addresses during the installation cycle.The provisioning of the DHCP addresses is important, especially if the cluster is installed with its own private network. The DCHP address also comes with the critical information for the BOOTP/PXE process-related information (where to get the pxelinux.0 file). The service can be provided from an outside resource. The configuration for the DHCPD server is located in the file /etc/dhcpd.conf – a copy is in the Appendix of this paper. The service uses the range of 192.168.254.1 until 250 for the setup process. The system node should have the default boot address 192.168.254.254. Of course, those addresses can be easily changed in the setup scripts. The command

   ```
   /etc/init.d/dhcpd restart
   ```

   ensures the service is started and functional, and

   ```
   /etc/init.d/dhdpd stop
   ```

   will ensure it is stopped. All DHCP addresses get logged in the file /var/log/messages for debugging purposes. The DHCPD daemon is not needed for normal cluster operation and can be switched off at this point.

   Lastly, ensure that no other DHCP servers are running in your environment because this will conflict with the installation server[7].

[7] This is particularly easy to miss when running the installation server inside Vmware, which runs its own DHCP server available to each virtual machine. If you are installing to a virtual machine, be sure to turn this off.

2.) TFTP server: The TFTP server is the most rudimentary file server provider. It is needed for the transfer of the boot image. Per definition, this server can be created as a standalone service or as part of the XINETD service, which is needed for the RSH operation of DB2 UDB in a cluster anyway. The TFTP server resides in the home directory /tftpboot, which is also the default share for the NDS server-based installation. The critical files in the directory are:

   a. pxelinux.0 (pxelinux boot file)
   b. pxelinux.cfg/default (pxelinux configuration file)
   c. *.profile files

   The TFTP daemon is automatically started when the xinet daemon is started with

   ```
   /etc/init.d/xinetd restart
   ```

   and stopped with

   ```
   /etc/init.d/xinetd stop
   ```

   The function can easily be tested on the command line by connecting to it with

   ```
   tftp localhost
   ```

   which should deliver a connection to the /tftpboot directory.

3.) NFS server: The NFS server is used to provide access to the software on the installation server and does this by sharing two important directories: 1) the /tftpboot directory itself and 2) the operating system sub-directory chosen. Together they contain the production OS image, the actual installation software as well as the DB2 ICE configuration scripts. These shares are only needed during installation and upgrades, and can be disabled afterwards. As well, the example export is pretty broad and can be restricted to the needed machines and subnets using the usual /etc/exports settings (see Linux manual for details).

   The NFS server is started with

   ```
   /etc/init.d/nfslock start; /etc/init.d/nfsserver start
   ```

   and terminated with

   ```
   /etc/init.d/nfsserver stop;/etc/init.d/nfslock stop
   ```

   For convenience, it is recommended to set the NFS server for automatic startup during boot time. The NFS server on the system node needs to be enabled for the following directories at a minimum:

   - /tftpboot
   - /tftpboot/suse/sles9x86_64 or /tftpboot/redhat/rhel4

   The latter directory depends on your operating system choice - either SUSE or RedHat. (Note: In our example in the Appendix, we also have a third share, which is used for a legacy 32-bit installation as well.)

Configuration of the System Node

The two key functions of the system node are:

1) To monitor and perform necessary upgrades to the cluster
2) To be used as an installation server to deploy further nodes.

In order to use the system node to monitor the cluster, you need the Nagios system management service. Nagios is a simple system monitoring tool allowing “phoning home” in the case of any hardware-related errors on the cluster. This service is automatically configured and started on part of the system node during deployment. More about Nagios is contained in a later chapter.

To use the system node as an installation server, just follow all steps mentioned above. Note, however, that in many customer environments certain services should not run on permanent base and are disabled for normal operation (such as the DHCP server).

Configuration of the Administration Nodes

Administration nodes are identical to the other physical nodes participating in the DB2 cluster except that they have no data to manage and instead are used to manage user connectivity, staging of data, and potentially load/unload operations. They can be implemented as nodes with additional storage space for staging as well as a server without those.

The first administration node is important since it holds the DB2 home directory, which is consequently shared across all database nodes. Therefore, the first administration node needs to have an active NFS server distributing the home directory. The installation of the NFS server is already part of the default image so only the NFS lock and NFS server daemons need to be enabled as well as exporting the DB2 home directory. (Both are done automatically during the DB2 ICE deployment.)

Note that although it contains no data, the file system layout will be identical to all the other nodes, as mentioned above. No other configuration is necessary.

Configuration of Data Nodes

Data nodes are the workhorses of the cluster and do not need any node-specific configuration. They mount the DB2 home directory and operate the actual database partitions with their data volumes. They get completely preconfigured by the system setup.

Configuration of Load and Backup Nodes

Load nodes are responsible for processing all of the extract, transformation, and load (ETL) work for the cluster and do not need any node-specific configuration.

Backup nodes in the terminology of DB2 ICE V2 are nodes with a large amount of available SATA disks. Normally this is a dual CPU server with a QLA2344 and then attached DS4100 storage - but other options will work as well. The general goal is to have large amounts of cheap (but slow) storage available to write the backup to those servers, where a normal tape process can then be initiated. Optical storage is also an option. Backup nodes usually provide access to their storage by sharing their volumes to the database server via NFS (these shares are usually named starting with ” /backup1” etc.). An additional way to use the backup node is its implementation as an IBM Virtual Tape Library. The exact options and details for the backup node are under investigation for a later release.

5. System configuration

Overview of the xcluster.cfg file

The focal point of DB2 ICE V2 configuration during deployment is the xcluster.cfg file found in the /tftpboot directory. It is the only file that needs any adaptation for usage on a customer site - assuming that a prepared installation server is available (usually obtained from the IBM DB2 ICE team as a VMware virtual machine). This important file contains a list all the machines that will make up the ICE cluster – each uniquely identified by the MAC address of its first Ethernet adapter (eth0).

In addition to identifying the machines in the cluster, it allows you to specify the system configuration for each machine in four major areas:

A) Node Type

Indicate whether the machine will be used as a System, Administration, Data, Load, or Backup node. This in turn will enable and disable the necessary settings – for example, the required network services - on each type of node during deployment.

B) Network Information

Indicate the requested network settings for each node in the cluster. This means the hostname, network adapter type, TCP/IP addresses, net masks, and default gateway will be identified for one or two networks on the cluster. The DB2 ICE configuration scripts will automatically configure these network settings on each node during deployment.

While the network can be configured in any way chosen by the deployment team, it is normally recommended to configure the cluster as a standalone cluster with two networks – one an Ethernet management network (e.g., 192.168.x.y), and one dedicated high-speed network for DB2 Fast Communication Manager (e.g., 172.16.x.y). The connection to the corporate user network is established – usually only on selected machines in the cluster - through additional available network ports.

See subsection “Network Configuration” below for additional details about supported network configurations.

C) DB2 UDB Configuration

Indicate the mandatory DB2 UDB settings that DB2 ICE needs for deployment. Currently, this consists of only the user names and new passwords for your three default DB2 UDB users (instance owner, DAS owner, and fenced user ID).

Note, the user names must match the user names that exist in the installation image you are using. If you are using the default image provided by the IBM DB2 ICE team then these are set to the DB2 UDB installation defaults, namely: db2inst1, dasusr1, and db2fenc1.

Using this information, the DB2 ICE deployment scripts will automatically export and mount the instance home directory, update the required configuration files for DB2

DPF to function (db2nodes.cfg, .rhosts), and start/stop the proper services for DB2 DPF to function.

D) Nagios Configuration

If you choose to utilize Nagios – the open source host, service and network monitoring program - in the DB2 ICE cluster you must indicate the mandatory Nagios settings needed to deploy it. Currently, this consists of only the Nagios administrator user name, password, and a contact e-mail address. Using this information, the DB2 ICE deployment scripts will automatically configure Nagios to monitor all machines in your clusters on the system node (if present in the xcluster.cfg file, of course – if no system node is present, no configuration will occur). The Nagios administration console can then be run on any Web browser on this System node.

Linux Distribution Support

At the time of writing, the system configuration layer was complete and available for a SUSE-based Linux system - with support for Red Hat systems currently under development.

Details of the xcluster.cfg file

The xcluster.cfg file specifications are documented below. Note that the text file format has been designed to be simple and flexible – and is structured to be extendable for future needs. Details below are provided mainly for reference purposes since it is recommended to edit a pre-existing xcluster.cfg file instead of creating one from scratch. A default xcluster.cfg file can be found in the Appendix.

The xcluster.cfg file is composed of three sections, each delimited by the following labels:

[KEYWORDS]

[NETWORK]

[NAGIOS]

The sections [KEYWORDS] and [NETWORK] are required.

The [NAGIOS] section is optional - Nagios will be configured for use on the System node if, and only if, this section is present in the xcluster.cfg file.

The order of the sections is important and should always follow the order above. The file itself is terminated by an empty section entitled: [END]

Section: [KEYWORDS]

The [KEYWORDS] section contains a list of keywords and their associated values that are used for DB2 UDB configuration as well as general network configuration for all nodes in the cluster.

Currently valid keywords include:

KEYWORK	EXAMPLE VALUE	REQ?	DESCRIPTION
NODE_PREFIX	XCLUSTER	No	A prefix that can be appended to each hostname in the cluster. It is used as a typing shortcut when all nodes in the cluster will be given an identical prefix (recommended).
NETWORK_GATEWAY	192.168.1.1	Yes	This address will be configured as the default gateway on each node in the cluster.
DB2_PRIMARY_NETWORK	eth0	Yes	This identifies the network – identified by this network adapter name - that will be used as the primary network for DB2. This is the network that will be identified in the db2nodes.cfg file as well as the one that will be used by rsh/db2_all/nfs/db2start etc. If only one network adapter is provided in the [NETWORK] section it must match this adapter. If two networks are provided, one of them must match this adapter.
TIMEZONE	US/Central	Yes	This is the time zone that will be configured on each node. The value must be provided in the standard Linux format as found in the “clock” file.
FCM_SECONDARY_NETWORK	ipoib0	No	This identifies the network that is used as the network for DB2 Fast Communication Manager (FCM). Usually this is a high-speed interconnect network. This keyword should only be specified if two networks have been specified in the [NETWORK] section.
ROOT_PASSWORD	root9man	Yes	The password that will be set for the root user on each node. Note: This will automatically be masked out of the xcluster.cfg file after installation on all nodes except for system node.
INST_USERNAME	db2inst1	Yes	The username of the DB2 instance owner. This must match the username found in the installation image.
INST_PASSWORD	db2admin01	Yes	The password that will be set for the DB2 instance owner. Note: This will automatically be masked out of the xcluster.cfg file after install on

			all nodes except for system node.
DAS_USERNAME	dasusr1	Yes	The username of the DB2 Administration Server (DAS) owner. This must match the username found in the installation image.
DAS_PASSWORD	db2das01	Yes	The password that will be set for the DB2 Administration Server (DAS) owner. Note: This will automatically be masked out of the xcluster.cfg file after installation on all nodes except for system node.
FENCED_USERNAME	db2fenc1	Yes	The username of the DB2 fenced user. This must match the username found in the installation image.
FENCED_PASSWORD	db2admin01	Yes	The password that will be set for the DB2 fenced user. This will automatically be masked out of the xcluster.cfg file after installation on all nodes except for system node.

Section: [NETWORK]

The [NETWORK] section contains a list of all machines in the cluster. This is a very critical part of the xcluster.cfg file. **If you attempt to deploy a DB2 ICE V2 image on a machine that is not listed – or not correctly listed - in this section of the xcluster.cfg file (i.e., its eth0 MAC address does not exactly match one found in the file) no configuration will occur**. In most cases, this means you should correctly add this machine to the xcluster.cfg file and re-run the installation.

Each line in the [NETWORK] section consists of six space-separated columns. The last column, which represents a second network, is the only optional column in the line. Note that the last column can optionally be separated by a comma.

Some examples of valid lines include:

-1 SYS 00:0C:29:E8:D4:CA SYSHOST eth0:192.168.99.200/24

0 ADM 00:0C:29:E8:D4:C9 <NODE_PREFIX>ADM eth0:192.168.99.200/24 eth1:192.168.12.10/24

2 DATA 00:0C:29:E8:D4:CB <NODE_PREFIX>DATA eth0:192.168.99.20/24,ipoib0:172.16.12.1/23

Below is an explanation of valid values for each column in the line:

COL.	CONTAINS	VALID VALUES	DESCRIPTION
1	DB2 Partition Number	Integer Number	DB2 Database Partition number for all nodes except for the system node (i.e.

			nodes of type “SYS”). For SYS node you should set this to be a negative number.
2	Node Type	SYS ADM DATA LOAD BACK	Indicates the node type of this machine in the DB2 cluster and hence how it will be configured during deployment. Note that for the ADM nodes, the first node of this type read in from the file will be considered the FIRST ADMINISTRATION NODE on the system and will be configured as such (i.e., it contains the instance home directory). The remaining ADM nodes will only contain coordinator functionality.
3	Identifying MAC address	A valid MAC address (colon delimited)	Mac Address of first Ethernet adapter (eth0) on the machine. This uniquely identifies the machine in the cluster.
4	Hostname	A string - optionally containing the value “<NODE_PREFIX>”	The hostname that will be assigned to this node. Note that if you include the value “<NODE_PREFIX>” anywhere in this string, it will be substituted with the value of the NODE_PREFIX keyword found in the [KEYWORK] section. If two networks are provided for this node, the second hostname will be generated by adding a suffix to this hostname, depending on the second adapter type : • eth1 -> <HOSTNAME>_E1 • iboip -> <HOSTANAME>_IB
5	Network Adapter 1 Configuration	adapter:IPADDRESS/XX Adapter can be either: • ethX • ipoibX where X is an integer number starting at 0. Also note that the IP address is in CIDR notation (netmask is included in the address).	The adapter for the first (required) network as well as the IPADDRESS and NETMASK that will be configured for this adapter. Note that in most cases this adapter name will correlate with the adapter name specified in the DB2_PRIMARY _NETWORK keyword in the [KEYWORDS] section.
6	Network Adapter 2 Configuration (OPTIONAL)	adapter:IPADDRESS/XX Adapter can be either: • ipoibX • ethX	The adapter for the second (optional) network as well as the IPADDRESS and NETMASK that will be configured for this adapter.

		where X is an integer number starting at 0. Also note that the IP address is in CIDR notation (netmask is included in the address).	Note that in most cases this adapter name will correlate with the adapter name specified in the FCM_SECONDARY_NETWORK keyword in the [KEYWORDS] section.

Section: [NAGIOS]

The [NAGIOS] section contains a list of keywords and their associated values that are required during Nagios configuration on the system node. If this section is not present – specifically, if the section header "[NAGIOS]" is not present in the file - Nagios will not be configured.

The network information that is provided in the [NETWORK] section of this file contains a majority of the information needed to configure Nagios on the cluster; however, there are a few Nagios specific keywords that are required in this section.

Currently the required keywords include:

KEYWORK	EXAMPLE VALUE	DESCRIPTION
NAGIOS_ADMIN_USERNAME	nagiosadmin	The username of the primary Nagios administrator. Note that in addition to this administrator, the DB2 instance owner will also be added as a Nagios administrator.
NAGIOS_ADMIN_PASSWORD	nagios01adm	The password that will be set for the Nagios administrator. This will automatically be masked out of the xcluster.cfg file after installation on all nodes except for the system node.
NAGIOS_ADMIN_EMAIL	tassi@server.com	The contact e-mail address for all Nagios administrators.

Network Configuration

One of the most important aspects of the xcluster.cfg file is the network information provided. It is critical that this information be specified correctly; otherwise, network conductivity - and everything in your cluster built on top of it (DB2, Nagios etc.) - will not function.

In particular, note the following points:

- Two adapter types are supported: ethX and ipoibX.
- No more then two networks can be specified for each machine (in column 5 and 6).

- The first network is mandatory; having a second network is optional.
- One of these networks must be an Ethernet network (eth0).
- The keyword DB2_PRIMARY_NETWORK must be set to match one of these 2 networks.
- The keyword FCM_SECONDARY_NETWORK is optional, and will only be used if it matches one of the networks specified in the [NETWORK] section.

Following these rules, that means that there are three possible ways to configure your networking in the xcluster.cfg file:

1.) One Ethernet network

```
DB2_PRIMARY_NETWORK   = eth0

0   ADM 00:0C:29:E8:D4:C9   HOSTNAME1   eth0:192.168.99.2/24
0   SYS 00:0C:29:E8:D4:CA   HOSTNAME2   eth0:192.168.99.3/24
```

2.) Two Ethernet networks

```
DB2_PRIMARY_NETWORK   = eth0 or eth1
FCM_SECONDARY_NETWORK = eth1 or N/A

0   ADM 00:0C:29:E8:D4:C9   HOSTNAME1   eth0:192.168.99.2/24, eth1:192.168.98.2/24
0   SYS 00:0C:29:E8:D4:CA   HOSTNAME2   eth0:192.168.99.3/24, eth1:192.168.98.3/24
```

3.) One Ethernet network and one InfiniBand network

```
DB2_PRIMARY_NETWORK   = eth0 or ipoib0
FCM_SECONDARY_NETWORK = ipoib0 or N/A

0   ADM 00:0C:29:E8:D4:C9   HOSTNAME1 eth0:192.168.99.2/24, ipoib0:192.168.98.2/24
0   SYS  00:0C:29:E8:D4:CA  HOSTNAME2 eth0:192.168.99.3/24, ipoib0:192.168.98.3/24
```

Details about what gets configured during deployment

The following list shows all elements that will automatically be configured as part of the DB2 ICE V2 deployment per node type:

ALL NODES

- Set IP address for network 1
- Set IP address for network 2 (optional)
- Set hostname for network 1
- Set hostname for network 2 (optional)
- Set default gateway
- Set up the /etc/hosts file
- Set up the machine timezone

- Set up root & DB2 user passwords
- Set up .rhosts file for root
- Set required environments variables for root (.bashrc)
- Set up dsh configuration for root
- Set up ssh no password logon between machines

SYSTEM NODES

- Set proper services for SYSTEM NODE
- Set up the .rhosts file in instance owner
- Nagios Configuration (optional). This involves all the steps that are documented in Section " 6. NAGIOS" below.

FIRST ADMINISTRATOR NODE (First node of type ADM in the file)

- Set up the db2nodes.cfg file in instance home
- Set up the .rhosts file in instance owner
- Set up the exports file to export instance home
- Set up the DB2 registry values for admin tools (default.reg)
- Set proper services for FIRST ADMINISTRATOR NODE
- Mask out passwords from xcluster.cfg file

NON-FIRST ADMINISTRATOR NODES

- Mount the instance directory in the fstab file
- Set proper services for NON-FIRST ADMINISTRATOR NODE
- Mask out passwords from xcluster.cfg file

DATA NODES

- Mount the instance directory in the fstab file
- Set proper services for DATA NODE
- Mask out passwords from xcluster.cfg file

LOAD and BACKUP NODES

- Set proper services for LOAD and BACKUP NODE
- Mask out passwords from xcluster.cfg file

6. NAGIOS

About Nagios

Nagios is an open source and GPL-licensed system and network monitoring application. This host and service management tool allows you to monitor your systems and services that you specify remotely, and will alert you when service or host problems occur. The problems are fixed before your end users and management realize they exist.

Originally designed for the Linux operating system, it can also be used on any flavor of the UNIX® operating system. You can download the software from www.nagios.org or simply use one of the Linux distributions that already have Nagios as an installable option such as SUSE Linux Enterprise Server Edition V9.

For customization, Nagios provides a simple plug-in design that allows you to develop your own service checks by using such tools as C++, Perl, Python, etc. Service checks are parallelized, and contact notifications can be sent via e-mail, pager, SMS, or any user-defined method that you develop through a plug-in. System Administrators can monitor the hosts and services and manage Nagios through an optional Web interface.

In the DB2 ICE cluster, the Nagios monitoring host is found on the System node. Nagios also has the ability to implement redundant monitoring hosts.

Linux Distribution Support

At the time of writing, the auto configuration of Nagios was complete and available for a SUSE-based Linux system - with support for Red Hat systems currently under development. **Hence all information that is detailed in this section is valid for a SUSE-based Linux system only.**

DB2 ICE Nagios Environment

SUSE Linux Enterprise Server provides the Nagios packages so that you do not need to download it from the Nagios Web site. The DB2 ICE Cluster image comes with all the Nagios components pre-installed, as listed in the following table:

Nagios components	Description
Nagios	Nagios Network Monitor
nagios-nsca	Nagios Service Check Acceptor
nagios-plugins	Nagios plugins
nagios-plugins-extras	Nagios plugins that depend on additional package
nagios-www	Nagios network monitor

Nagios' directory structure and file location consists of five directories. Note that for SUSE Linux Enterprise Server Edition V9, these directories are in different locations from those of versions you may have downloaded from the Nagios Web site. (www.nagios.org).

A brief description of what each directory contains is given in the following table:

Sub-Directory	Contents
/usr/init.d/ and /usr/sbin	Nagios core program/executables
/etc/nagios	Main, resource, object, and CGI configuration files should be put here
/usr/lib/nagios/plugins	Nagios plugins
/usr/lib/nagios/cgi	Location of the CGIs
/usr/share/nagios	HTML files (for Web interface and online documentation)
/var/log/nagios/	Empty directory for the log file, status file, retention file, etc.
/var/log/nagios/archives	Empty directory for the archived logs
/var/log/rw	Empty directory for the external command file

With this version of DB2 ICE, Nagios comes preconfigured for immediate use once the system node of the cluster is up and running.

Nagios comes with many rich monitoring services such as monitoring of network services (SMTP, POP3, HTTP, NNTP, ICMP, SNMP, PING, etc.) and host resources (processor load, disk usage, system logs).

In addition, in the DB2 ICE preconfigured version of Nagios, we have provided services to monitor host availability, processor load, disk usage for /db2data, /, and /var, the number of users logged into each machine and the number of total processes.

All monitoring of the remote servers is done through ssh. For Nagios to use ssh in a SUSE environment, this requires that the user's root, wwwrun, and daemon have ssh access to all other nodes in the cluster. To ensure that ssh works, you can run the following command on each host:

```
ssh <hostname>
```

Note that the following files must exist with the correct entries:

- /etc/Nagios/.ssh/id_rsa
- /var/lib/wwwrun/.ssh/id_rsa
- /root/.ssh/id_rsa
- /sbin/.ssh/known_hosts
- /root/.ssh/known_hosts

With the DB2 ICE deployment, this has already been done for you, so there is no need to configure this, only ensure that they are not removed.

The Optional but Handy Web Interface:

Nagios provides a Web interface that allows you to monitor and manage your cluster remotely. For the Web interface to work, we need the Apache Web server running; you can check this by running the command:

```
chkconfig apache
```

To access the Web interface, we have configured it to use Web Authentication. You log into the interface as a Nagios administrator. During a DB2 ICE installation, two administrators were created, depending on the values specified for the following keywords in the xcluster.cfg file:

NAGIOS_ADMIN_USERNAME (usually: nagiosadmin)
INST_USERNAME (usually: db2inst1nagiosadmin)

Their passwords are set during installation to the values specified for these keywords in the xcluster.cfg file :

NAGIOS_ADMIN_PASSWORD
INST_PASSWORD

If you want to change the passwords, you can do so by running the following command:

```
htpasswd -c /etc/nagios/htpasswd.users <username>
```

Any time you make any modifications to the Apache Web server, you will need to restart Apache. You can do so with the following command:

```
/etc/init.d/apache restart
```

You can now access the Web interface through the URL:

http://*your-nagios-box-hostname*/nagios

At this point, you should be prompted to authenticate yourself (default is nagiosadmin, password: db2ice*)*

Configuring Nagios:

For Nagios services to work, the following configuration files have been preconfigured through the deployment of the DB2 ICE cluster.

All of these files are found under /etc/nagios:

i. Nagios.cfg:

Nagios.cfg is the main configuration file where you specify all other configuration files to be used. For more information on how to configure this, you can go to http://nagios.sourceforge.net/docs/2_0/configmain.html

ii. Resources.cfg:

The Resource configuration file is used to specify $USERn$ macro definitions. These, $USERn$ macros are useful for storing usernames, passwords, and items commonly used in command definitions (like directory paths). The CGIs will not attempt to read the resource file, so to protect sensitive information, you can restrict access to these via permissions (600 or 660). Multiple resource files can be used by defining them in the main configuration file nagios.cfg.

iii. Object configuration files:

The Object configuration files contain all the object definitions that Nagios should use for monitoring. Object configuration files contain definitions for hosts, host groups, contacts, contact groups, services, commands, etc. You can separate your configuration information into several files and specify multiple **cfg_file=** statements in the main configuration file nagios.cfg to have each of them processed.
The following files have been configured already.

- contacts.cfg
- contactgroups.cfg
- checkcommands.cfg
- command.cfg
- escalations.cfg
- hostgroups.cfg
- hosts.cfg
- services.cfg

iv. cgi.cfg:

The CGI configuration file **cgi.cfg** allows you to configure which users have access to what monitoring services and management abilities from the Web interface.

Note that all of these files have been preconfigured through the DB2 ICE deployment. There is no need to modify any of these unless you want to customize Nagios yourself. If you choose to do so, you can check to see if your configuration files are correct by running the following command:

```
/etc/init.d/nagios -v /etc/nagios/nagios.cfg
```

Any errors with Nagios can be found in /usr/local/nagios/var/nagios.log. Should you make any changes to the Nagios scripts, be sure to restart Nagios.

Starting and Stopping Nagios

With the DB2 ICE cluster, the Nagios service is started once the System node has been started. To stop, start, or restart the Nagios service, you can run the following command:

```
/etc/init.d/nagios [start | stop | restart]
```

7. Frequently Asked Questions

- **I want to deploy a DB2 ICE V2 cluster and already have the installation server VMWare virtual machine provided by the IBM DB2 ICE team. How do I get started?**

  A quick summary:

  1. Ensure the target machine you are installing to can see (i.e., ping) your installation server. For the VMWare installation server, that means ensuring you are using the “bridged” networking option.

  2. To turn on the proper services on your installation server, run the script /tftpboot/reset.sh.

3. Update the file /tftpboot/xcluster.cfg with your intended values. Double-check to ensure your machine's eth0 mac addresses have been properly entered.

4. Reboot the target machine and get to the boot selection menu by pressing F12. Choose the network boot option.

That is all. You can now watch the installation take place.

For more information, see sections 3, 4 and 5 of this paper.

- **How do I change the TCP/IP addresses used by the installation server?**

  To change the IP address of the installation server itself, just search and replace the current IP address in the following files (default is 192.168.254.254):

  a. /tftpboot/pxelinux.cfg/default
  b. /tftpboot/bcu_x364.profile64

  To change the IP address subnet for those that are temporarily handed out during installation time to each node, just update the /etc/dhcp.conf file on the installation server.

  To change the final IP address of the nodes being installed, just update the /tftpboot/xcluster.cfg file with the new values for each node in the cluster.

- **What hardware have you tested on so far?**

  The DB2 ICE configurations are constantly moving forward as updated software and new hardware is available in the IBM product line. The installation image has purposefully been generically configured to be useful on a large number of IBM xSeries servers.

  Nonetheless, it is highly recommended to stick to the tested versions below.

  - Server: x346 (both 32-bit and 64-bit)
  - Server: HS20 (32-bit)
  - Server: e326 (64-bit) with QLA controllers and DS4300 storage,
  - ServeRAID (Adaptec),
  - AIC7xxx and AIC 79xx controllers.
  - VMware images (32-bit) with Buslogic controller setting

- **Can I install more than one node simultaneously using the same installation server?**

  Absolutely – in fact this is the recommended approach to speed up deployment. Normally, we have four to six nodes installing simultaneously at any one time - but of course this depends on the network bandwidth and the hardware of your installation server. As well, it is recommended to initiate the first installation until the point of operating system installation before starting the remaining nodes. This will ensure that everything is set up correctly in your setup before starting all the other machines.

- **Where can I find the xcluster.cfg file that was used to install this specific node?**

  This file can be found for each node at:

  /root/.db2ice/xcluster.cfg

Note that all passwords used during deployment will be masked out of the file for all nodes except for the system node.

- **What log files are generated during a DB2 ICE node deployment? Where can I find them?**

  In addition to the regular information that is logged to the /var/log/messages file on the system node, two important log files are generated on each node during an installation:

  1. xcluster.pl.log (generated by DB2 ICE configuration scripts)
  2. y2log (generated by autoyast)

  The log files can be found on the rescue partition of each system. To access this partition and view the log, you can issue the following commands on each node after deployment:

```
> mount /dev/sda2 /mnt
> less /mnt/var/adm/autoinstall/logs/xcluster.pl.log
> less /mnt/var/log/YaST2/y2log
```

- **I made a mistake in my xcluster.cfg file and received an error during/after deployment. What should I do now?**

  In almost all cases, it is best to “fix” your xcluster.cfg file on your installation server and re-issue the installation on the target node (i.e., reboot the target machine and network boot again). Generally, it takes only 10 to 15 minutes to deploy a new node from an installation server so restarting from scratch is normally quite painless.

- **I see a “fail safe” option in the boot menu. When was this installed and what is it used for?**

  This is a Linux rescue system that gets automatically installed on each node on partition /dev/sda2. It allows you to access and fix your system in the event of problems on your primary Linux partition.

  As well, during deployment it was used as a base for the IBM system configuration code to run.

  Note that if you are ever making any changes to your system that would involve remaking the init RAM disk (e.g., installing a new Qlogic driver), it is a good idea to make a copy of your current one using the following steps:

  1) cd /boot
  2) Copy “initrd” to “initrd-backup”
  3) Edit /boot/grub/menu.lst

     - Add the new entry for the backup ram disk
     - Copy the three lines for the basic Linux boot and edit the line "initrd (hd0,0)/initrd" by changing "initrd" to “initrd-backup”

- **I notice you are using SUSE Autoyast to install the initial Operating System. What about using the Red Hat kickstart?**

  The RedHat version of the DB2 ICE Deployment kit using kickstart is currently under development.

8. Appendix

This section contains all the complete setup scripts needed during the setup and operation of the cluster.

/etc/dhcpd.conf

```
allow bootp;
allow booting;
allow unknown-clients;
boot-unknown-clients on;
next-server 192.168.254.254;
filename "pxelinux.0";
log-facility local7;
option option-150 code 150 = string;
option option-150 "(nd)/menu.lst";
ddns-update-style none;
subnet 192.168.254.0 netmask 255.255.255.0 {
       range 192.168.254.1 192.168.254.250;
       option domain-name "mysite.com";
              option domain-name-servers 192.168.4.5, 192.168.4.6;
              option routers 192.168.254.254;
              default-lease-time 5400;
}
```

/etc/xinetd.d/tftpd

```
# default: off
# description: tftp service is provided primarily
# for booting or when a router needs an upgrade.
# Most sites run this only on machines acting as
# "boot servers".
service tftp
{
              socket_type     = dgram
              protocol        = udp
              wait            = yes
              user            = root
              server          = /usr/sbin/in.tftpd
              server_args     = -s -v /tftpboot
              disable         = no
}
```

/etc/exports

```
/tftpboot/suse/sles9i386   *(rw,sync,no_root_squash)
/tftpboot/suse/sles9x86_64 *(rw,sync,no_root_squash)
/tftpboot                  *(rw,sync,no_root_squash)
```

/tftpboot/pxelinux.cfg

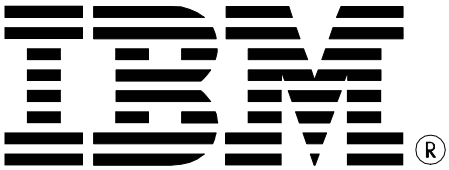
```
DEFAULT netboot

LABEL netboot
kernel /suse/sles9x86_64/boot/loader/linux
append initrd=/suse/sles9x86_64/boot/loader/initrd ramdisk_size=65536
splash=verbose showopts textmode=1 instmode=nfs netconfig-dhcp netdevice=eth0
install=nfs://192.168.254.254/tftpboot/suse/sles9x86_64
autoyast=tftp://192.168.254.254/bcu_x346.profile load_ramdisk=1
       IPAPPEND 1
```

/tftpboot/xcluster.cfg (SAMPLE1)

```
# DO NOT DELETE THE SECTION HEADERS!
#
# VALID REQUIRED SECTIONS INCLUDE:      [KEYWORDS] [NETWORK]
# VALID NON-REQUIRED SECTIONS INCLUDE:  [NAGIOS]

# Two networks with Nagios installed

[KEYWORDS]

NODE_PREFIX           = TEST
NETWORK_GATEWAY       = 192.168.99.1
TIMEZONE              = US/Central

DB2_PRIMARY_NETWORK   = eth0
FCM_SECONDARY_NETWORK = ipoib0

ROOT_PASSWORD     = root9man
DAS_USERNAME      = dasusr1
DAS_PASSWORD      = db2admin
INST_USERNAME     = db2inst1
INST_PASSWORD     = db2admin
FENCED_USERNAME   = db2fenc1
FENCED_PASSWORD   = db2admin

[NETWORK]

#
# NODE# NODETYPE    MAC               HOSTNAME                IPADDRESS (up to 2)
#
-1  SYS     00:0C:29:AB:52:2C  <NODE_PREFIX>SYS  eth0:192.168.99.254/24, ipoib0:192.168.98.254/24
0  ADM      00:0C:29:E8:D4:C9  <NODE_PREFIX>ADM  eth0:192.168.99.200/24, ipoib0:192.168.12.10/24
1  LOAD     00:0C:29:AB:52:2E  <NODE_PREFIX>001  eth0:192.168.99.50/24,  ipoib0:192.168.12.11/12
2  DATA     00:0C:29:AB:52:2F  <NODE_PREFIX>002  eth0:192.168.99.51/24,  ipoib0:192.168.12.12/12
3  DATA     00:0C:29:AB:52:2G  <NODE_PREFIX>003  eth0:192.168.99.52/24,  ipoib0:192.168.12.13/12

[NAGIOS]

NAGIOS_ADMIN_USERNAME     = nagiosadmin
NAGIOS_ADMIN_PASSWORD     = root9
NAGIOS_ADMIN_EMAIL        = btassi@system.com

[END]
```

/tftpboot/xcluster.cfg (SAMPLE 2)

```
# DO NOT DELETE THE SECTION HEADERS!
#
# VALID REQUIRED SECTIONS INCLUDE:      [KEYWORDS] [NETWORK]
# VALID NON-REQUIRED SECTIONS INCLUDE:  [NAGIOS]

# Two Ethernet networks configured but none used for FCM - Nagios not configured

[KEYWORDS]

NODE_PREFIX           = TEST
NETWORK_GATEWAY       = 192.168.99.3
TIMEZONE              = US/Eastern

DB2_PRIMARY_NETWORK   = eth0
FCM_SECONDARY_NETWORK = N/A

ROOT_PASSWORD     = root9man
DAS_USERNAME      = dasusr1
DAS_PASSWORD      = db2admin
INST_USERNAME     = db2inst1
INST_PASSWORD     = db2admin
FENCED_USERNAME   = db2fenc1
FENCED_PASSWORD   = db2admin

[NETWORK]

#
# NODE# NODETYPE    MAC           HOSTNAME   IPADDRESS (up to 2)
#
-1  SYS       00:0C:29:E8:D4:CA  XCLUSTER0  eth0:192.168.99.10/24  eth1:192.168.12.50/23
0  ADM        00:0C:29:E8:D4:C9  XCLUSTER1  eth0:192.168.99.11/24  eth1:192.168.12.51/23
1  DATA       00:0C:29:AB:52:2F  XCLUSTER2  eth0:192.168.99.12/24  eth1:192.168.12.52/23
2  DATA       00:0C:29:AB:52:2B  XCLUSTER3  eth0:192.168.99.13/24  eth1:192.168.12.53/23
3  BACK       00:0C:29:AB:52:2C  XCLUSTER3  eth0:192.168.99.14/24  eth1:192.168.12.54/23


[END]
```



IBM Canada
8200 Warden Avenue
Markham, ON
L6G 1C7
Canada

Printed in United States of America
01/06