IBM Software Group

IBM Tivoli Solutions

SAP High Availability with IBM Tivoli System Automation for Multiplatforms

Contents

Introduction

This paper describes an approach to creating a highly available SAP solution that covers all critial components. The IBM High Availability (HA) middleware solution (Tivoli System Automation for Multiplatforms) provides this level of high availability.

High availability is a term used to describe systems that are continously available that are up and running, performing the tasks they are dedicated to and are available to end users most of the time.

This implies:

- When failures occur, either caused by hardware or software, highly available systems must recover quickly.
- Even on peak loads and a loss of availablility, the systems most perform appropriately and process transactions within a reasonable amount of time.

In terms of service contracts with guaranteed availability levels and where the term recovery time objective is used to express the service level agreement (SLA), a highly available SAP installation will meet a recovery time objectives of a few minutes.

Tivoli System Automation for Multiplatforms is a high-availability cluster solution that provides several monitoring mechanisms to detect system failures and a set of rules to initiate the correct action without any user intervention. The set of rules is called a policy, this policy describes the relationships between applications or resources. This policy and the monitoring mechanisms provide System Automation for Multiplatforms with extensive up-to-date information about the system landscape so that it can restart the resource on the current node or move the whole application to another cluster node.

When the database of your SAP system is IBM DB2 for Linux, UNIX, and Windows the cluster manager System Automation for Multiplatforms is already included. DB2 for LUW itself can be protected using System Automation for Multiplatforms.

To protect the SAP Central Services, System Automation for Multiplatforms has to be deployed on the cluster nodes and it has to be configured to monitor the SAP Central Services. When a failure of an SAP Central Service is detected System Automation for Multiplatforms will autonomously choose the correct action to recover from the outage by restarting the SAP Central Service on the current node or another node. This paper will detail why companies need HA solutions for SAP, and introduce degrees of availability. Furthermore, it describes which components of an SAP system should be protected, and highlight the available IBM solutions for SAP high availability.

Significance of high availability for SAP applications

Availability of an application is defined as the amount of time the application is accessible to an end user and is measured as a percentage of availability over total time. Most enterprises running various SAP applications are very dependent on the availability of the data to be able to make critical business decisions. Customers from various industries like consumer product goods, manufacturing, transportation etc. are heavily invested in SAP applications and are increasingly dependent on them to be able to run their businesses. With the increased focus of businesses on globalization and expansion across borders, there is no suitable time when these critical applications can be pulled down for maintenance without a suitable redundant setup.

The Information Technology (IT) department must ensure availability of the SAP applications across a multitude of failure scenarios. Critical applications need to be designed to be deployed in a highly available environment to ensure availability in case of failures. With the increased adoption in virtualization technologies and cloud deployment models, the aspect of application high availability is taking more significance. The cost pressures and efficiencies driven by virtualization are forcing customers to explore those deployment alternatives, but those introduce new risks to the management framework and application availability. The need for high availability management and automation tools is more than ever driven by the complexities of deployment. A typical high availability tool would monitor the availability of

- The hardware elements like servers, network cards etc.
- Network components like routers, switches and load balancers
- Storage components like filesystems, mounted disks
- SAP Central Instance is part of the Netweaver stack
- Data stores like databases and other repositories for example DB2, Oracle, MySQL

The benefits of a high availability automation tool like Tivoli System Automation are two fold. First, it helps recovery from known or unknown failures within the recovery time objectives (RTO) set by the organization. Secondly, it helps enterprises to avoid costly mishaps due to unavailability of their critical systems like SAP applications that can lead to loss of revenue, drop in customer satisfaction and eventually loss in customers, or loss in image. Besides automation tools drive discipline in application management and greatly reduce operator errors due to the checks and bounds and recovery automation procedures. Managing complex applications like SAP and investigating root cause analysis during a failure situation can be challenging and hence Tivoli System Automation for Multiplatforms introduced a custom policy to manage SAP availability. It greatly helps the IT organization with automating failover as well enabling them to do root cause analysis in case of failures and drives efficiency by consolidating errors across the infrastructure.

Degrees of availability

The terms high availability, continuous operation, continuous availability and disaster recovery are generally used to express how available a system is. In the following sections, we define and discuss each of these terms.

High availability

High availability means being able to avoid unplanned outages by eliminating single points of failure. This is a reliability measure of hardware, operating system, and database manager software. Another measure of high availability is the ability to minimize the effect of an unplanned outage by masking the outage from the end users. This requires some sort of availability automation tool like Tivoli System Automation for Multiplatforms to manage application failover within the same server farm or backup systems to ensure that high availability measures are achieved.

Continuous operation

Continuous operation means being able to avoid planned outages or application maintenance driven down times. To provide continuous operation, there needs to be provided a mechanism to perform system checks as well as hardware and software maintenance like upgrading patch levels while the application remains available to the end users. This is accomplished by providing multiple servers and switching end users to an available server at times when one server is made unavailable. In this setup, automation software like Tivoli System Automation for Multiplatforms can drive automatic failover to an alternative setup, although for continuous operation high availability is not a requirement. Hence automation software like Tivoli System Automation Application Manager can also be used to toggle between multiple setups and maintain overall application availability.

Continuous availability

Continuous availability combines the characteristics of high availability and continuous operation to provide the ability to keep the SAP system running as close to 24x7x365 as possible. This is what most customers want to achieve.

Disaster Recovery

Critical applications that drive key business processes need to be enabled for disaster recovery protection. A typical disaster recovery enabled application reflects characteristics of continuos availability and in addition has some data replication capability put in place to ensure when the application is relocated to a different datacenter, the recovery point objective (RPO) requirements are met and no customer transactions are lost. Disaster Recovery is the most expensive setup and most enterprises are judicious in which selective applications are enabled for this capability since it requires investment in storage replication software in addition to automation and availability software like Tivoli System Automation suite of products.

Single point of failures of an SAP System

An SAP system consists of various servers and services and some of them are single point of failures. In this paper we focus on these several servers and services as described in Figure 1: Distributed SAP System. The following paragraphs will describe the individual components and the end-user impact of an outage of each component.

Database host

The DB2 for LUW database server is the persistent storage for the whole SAP system. How you use System Automation for Multiplatforms to protect the DB2 for LUWis described in detail in a separate guide.

This guide can be found as an attachment to SAP Note 960843, resp. IBM redbook 247636 "High Availability and Disaster Recovery Options for DB2 on Linux, UNIX, and Windows."

Outage impact

The whole SAP system is stopped. Once the database becomes available again, the SAP work processes reconnect and the users can continue their work. All transaction currently in progress are rolled back.

SAP Central Services (SCS Host)

As SAP Central Services we categorize all services of an SAP system that are essential for the SAP system to run properly and are single point of failures. This section will list all SAP Central Services including the impact if the services are not available.

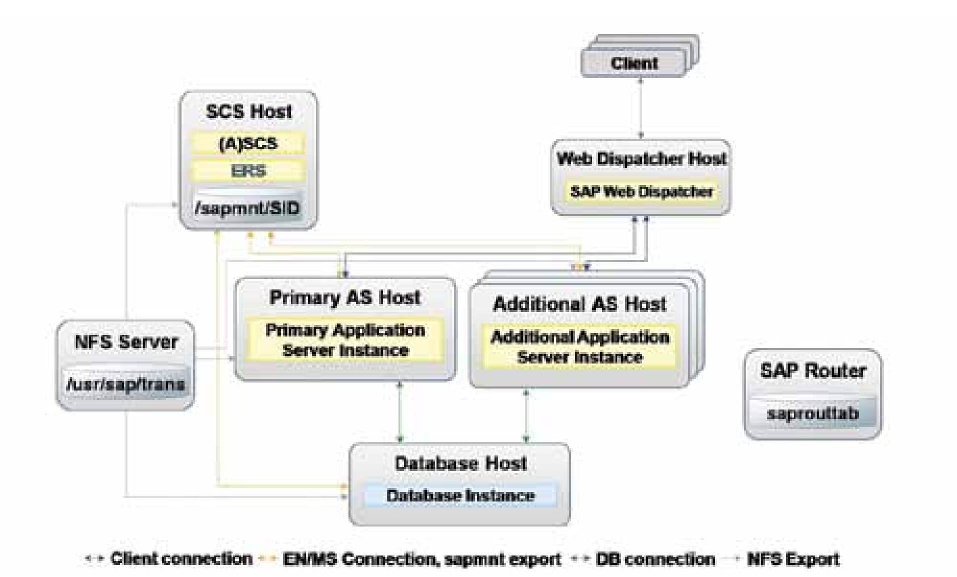

Figure 1: Distributed SAP System

Enqueue server and replicated enqueue server (ASCS and ERS)

An SAP system has its own locking mechanism on business transaction levels to synchronize database accesses. Due to this locking mechanism two transactions cannot update the same data in the database simultaneously.

The locks on objects are managed by the enqueue server. The SAP applications acquire and release locks through the enqueue server. The enqueue server itself stores all locks in the lock table. The locking mechanism is displayed in figure 2.

In highly available environments the enqueue server is installed as standalone enqueue server. On SAP Java systems the standalone enqueue server is executed in SAP Central Services. On SAP ABAP systems, depending on the SAP release, the enqueue server might be already installed as standalone enqueue server. If it is not already done, you have to install it according to the SAP documentation.

The advantage of a standalone enqueue server over the integrated enqueue server in the SAP Central Instance is that the standalone enqueue server can be monitored much better from the cluster management software and it can be restarted more quickly in case of failures.

The high availability enqueue server consists of the standalone enqueue server and an enqueue replication server. The replication enqueue server runs on another host and contains a replica of the lock table (replication table).

When the enqueue replication server is enabled, all lock entries from the enqueue server are replicated. Both lock tables in the enqueue server and enqueue replication server are kept in memory. In case of an outage of the enqueue server the cluster manager restarts the enqueue server on a node where the enqueue replication server is currently active. The enqueue server obtains the shared memory object of the enqueue replication server and terminates the enqueue replication server. Finally, it recovers the lock table from the replica. Once the origin hosts becomes online again the cluster manager will start a new enqueue replication server on that node.

Outage impact

When the enqueue server fails the SAP applications cannot acquire locks anymore. Therefore, the SAP system hangs because the applications waits for the lock or be terminated. With an enqueue replication server, the locks are not lost in case of an outage of the enqueue server, and recovery is much faster.

Message Server

The SAP message server runs as a separate process, mostly on the same host as the central instance (SCS host). If an (ABAP) SAP Central Services (SCS) instance is configured in the system, the message server is part of this instance.

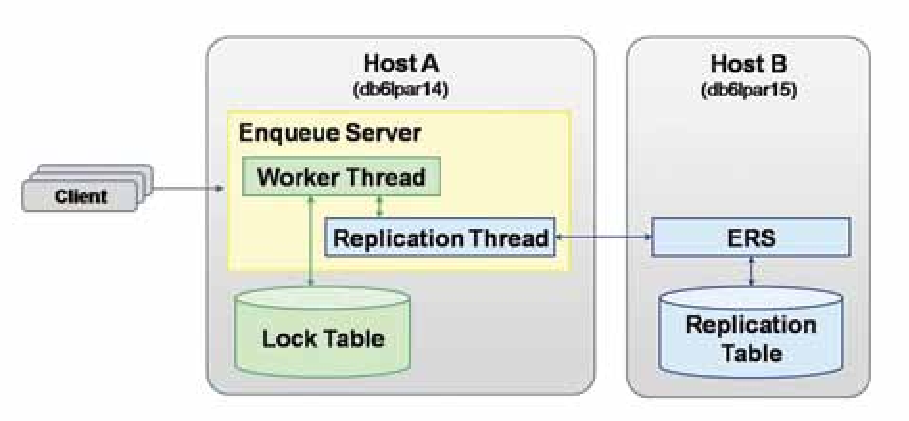

Figure 2: Standalone enqueue server and enqueue replication server

Only one message server can run in each SAP system. It performs the following functions in the SAP system:

- Central communication channel between the individual application servers (instances) of the system
- Load distribution of logons using SAP GUI and Remote Function Call (RFC) with logon groups
- Information point for the web dispatcher and the application servers (each application server of the system first logs on to the message server)

When an instance is started, the dispatcher process contacts the message server so that it can announce the services it provides. If the connection setup to the message server fails, an entry is made in the system log (syslog).

Outage impact

If the message server stops working, it must be restarted as quickly as possible to ensure that the system continues to operate smoothly. For example,requests cannot be executed for dialog, update and enqueue server.

High availability considerations

To create an abstraction layer for all SAP logon applications it is highly recommended that you create logon groups through the message server. An individual application server might be offline, whether this is planned or unplanned. When the users log on through logon groups they are automatically routed to an online application server.

Gateway

The SAP Gateway (not shown in Figure 1) carries out RFC services within the SAP world. These services are based on TCP/IP. The services enable SAP Systems and external programs to communicate with one another.

RFC services can be used either in the ABAP program or for the external programs using the interfaces. Each instance of an SAP System has a gateway. The gateway enables communication between work processes and external programs, as well as communication between work processes from different instances or SAP Systems.

Outage impact

Without the gateway process, no RFC communication and no execution of registered programs is possible.

Central NFS share (NFS server)

There are two central NFS shares which are essential for an SAP system:

- /sapmnt/<SID>: Required to share binaries and configuration data for the application servers.
- /usr/sap/trans: The transport directory has to be shared in a logical transport landscape so that the source SAP system can write the transport files to the share and the target systems can pick them up from this location.

Outage impact

If the /sapmnt/<SID> NFS share is not available, no SAP application server can be started. All active SAP application servers are not affected.

When the transport directory is not available the change and transport system cannot be used.

System log

SAP applications write the information needed for problem analysis to a central log (not shown in Figure 1).

Outage impact

Without the system log, no problem analysis can be performed.

SAP application servers (AS host)

The SAP application servers actually host the applications and serve the user requests. The application server layer is also the scale-out layer to ensure the performance for SAP applications.

Outage impact

All applications currently running on the application server will be terminated and the transactions are rolled back. The users will have to log on again.

High availability considerations

You must have more than one application server so that the users can log on again to another application server if one fails. The routing to an online application server is done through message server logon groups.

SAP web dispatcher

The SAP web dispatcher (not shown in Figure 1) lies between the Internet and your SAP system. It is the entry point for http or https requests into your system, which consists of one or more NetWeaver application servers. As a "software web switch," the SAP web dispatcher can reject or accept connections. When it accepts a connection, it balances the load to ensure an even distribution across the servers. The SAP web dispatcher therefore contributes to security and also balances the load in your SAP system.

You can use the SAP web dispatcher in double stack (ABAP and Java) systems and in pure Java systems, as well as in pure ABAP systems.

Outage impact

When the SAP Web Dispatcher is offline no http or https requests can be send to the SAP system by means of the virtual IP address. If not prohibited by the network topology or firewall, only direct requests to the SAP application servers are possible. Since the SAP web dispatcher handles http or https requests only, you only have to make this one highly available when you are running SAP applications based on http or https protocols such as Webdynpro or board support packages (BSPs).

SAProuter

SAProuter is an SAP program that acts as an intermediate station, or proxy, in a network connection between SAP systems, or between SAP systems and external networks. SAProuter controls the access to your network (application level gateway), and, as such, is a useful enhancement to an existing firewall system (port filter).

Outage impact

Connections using the SAProuter cannot be established.

Logical view of a highly available SAP system

In "Figure 3: High Available SAP System" all single point of failures are highlighted in red. These are the instances which should be the focus on in the following chapters.

There is also an option to protect the SAP application servers using System Automation for Multiplatforms (not described in this document). Because there will most likely be more than one SAP application server, and because it is certain that all application servers can continue their work without the primary instance, it is not absolutely necessary to put the application servers under control of System Automation for Multiplatforms.

IBM solutions for SAP high availability

Figure "IBM solutions for SAP high availability" shows the different IBM HA technologies that can be applied in order to eliminate the single points of failure described in the previous chapter. We will detail these technologies in later paragraphs.

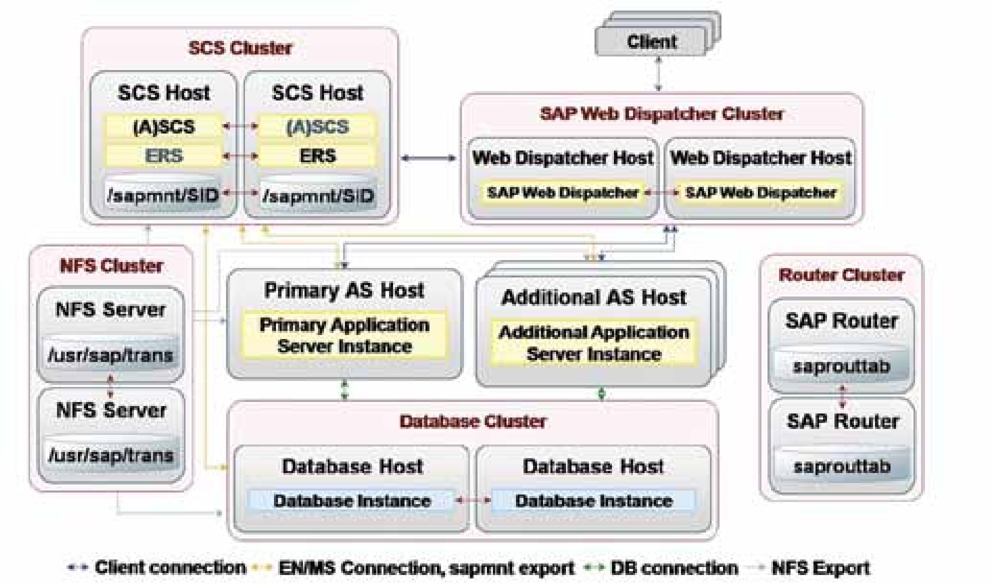

Figure 3: High Available SAP System.

IBM DB2 High Availability Disaster Recovery is a database log replication feature that provides a high availability solution for failures of a database node. The required takeover from such a failing node is controlled by System Automation for Multiplatforms.

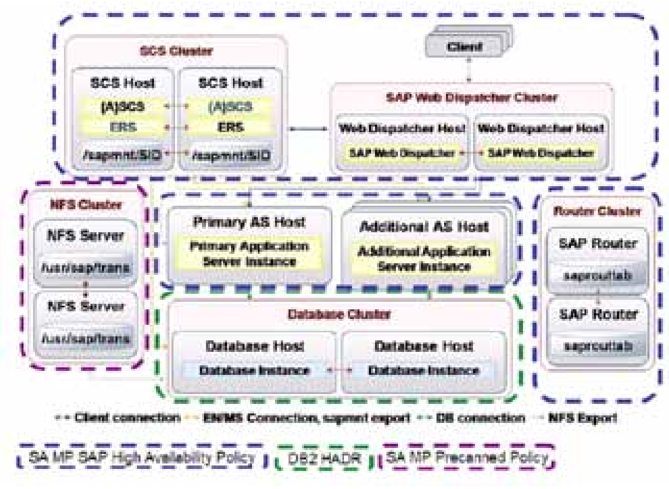

Figure 4: IBM solutions for SAP high availability.

The System Automation for Multiplatforms SAP High Availability policy uses System Automation for Multiplatforms to automate all SAP components. System Automation for Multiplatforms detects failed components and restarts or initiates a failover. The policy has been developed based on long and successful customer relationships. It also helps to reduce the operational complexity of an SAP environment and to avoid operator errors resulting from this complexity.

The System Automation for Multiplatforms Precanned policy for NFS is a System Automation for Multiplatforms policy that automates and keeps an NFS server highly available. It is based on best practices from the field and available free on the Tivoli Open Process Automation Library (For Linux systems, the link is “https://www-304.ibm.com/software/brandcatalog/ismlibrary/details?catalog.label=1TW10SA02”).

Now we turn to the HA technologies of SAP Central Services and the database and describe these technologies based on an exemplary SAP setup: SAP Customer Relationship Management based on NetWeaver 7.00. However, the described IBM technologies can also be used for all other SAP solutions based on NetWeaver[1].

There are two different cluster setup possibilities with SAP and DB2. A general SAP installation can be separated into a central and a distributed system installation and you can make the central services of SAP and /or the database high available. The following section will describe the two resulting cluster landscapes in more detail.

Cluster Setup: Central System Installation

A central system installation is an SAP system where the database, the central services and the application server are running on the same physical host If you want to make this central SAP system highly available you need another host where your services can run in case of a failure.

This means in cluster terminology that we have one node for the active SAP system and another node for the standby functionality of a high availability cluster system. “Figure: Setup on two Cluster Nodes” represents a basic two node cluster with highly available central services of SAP and a clustered database.

The central services of SAP are divided into two categories and each category has its own virtual hostname.

The SAP ABAP application server depends on the ASCS instance and the correlated *ABAP Enqueue Replication Server (AERS)*. The ASCS instance contains the message and the enqueue service for ABAP and the AERS replicates the enqueue service. Both instances are running on the virtual hostname *db6lparvascs*.

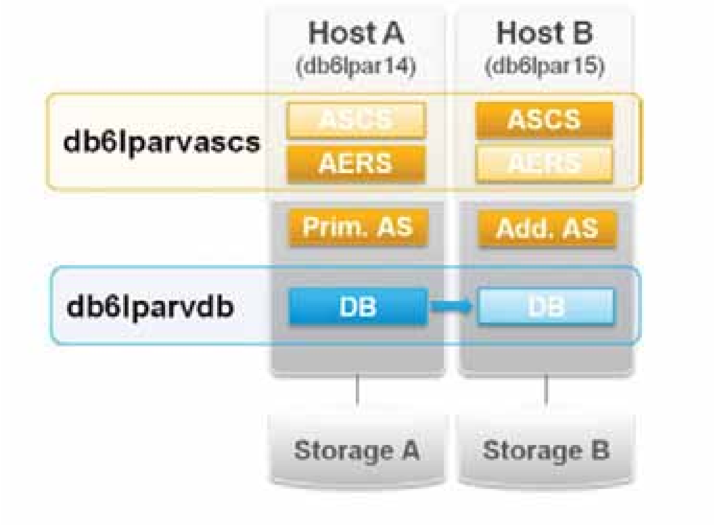

Figure 5: Setup on two cluster nodes.

The primary application server Prim.AS is the only instance which is not highly available in “Figure: Setup on two Cluster Nodes.” However, since there is an additional Application Server (Add.AS) users can relogon to this server in case the Prim.AS fails. All other single point of failure services are now separated in their own instances, which are each identified as the Central Service Instance for ABAP (ASCS) and Java (SCS). Furthermore, the highly available database gets its own virtual host name, so it can switch from the current active host *db6lpar14* to host *db6lpar15* without interfering with the SAP application server and his central services.

Cluster setup: distributed system installation (four nodes)

The distributed installation refers to an SAP system where the central services and the application server run on their own server and the application server connects to a dedicated database server which is generally useful for productive systems to improve performance.

For example, if you decide to make your distributed system high available, you will get a four node cluster, as you can see in “Figure: Setup on 4 Cluster Nodes” in contrast to the two node cluster from “Figure: Setup on two Cluster Nodes. You will use the same virtual hostnames and create the same central services. The only difference is our database, which runs in a dedicated cluster now.

The whole hardware of host db6lpar14 and db6lpar15 are only used for the standby and the primary database. The same is true for the SAP cluster running in its own cluster on host db6lpar16 and db6lpar17. This architecture is very powerful and enables you to set up a really high performance cluster. The challenge is not to set up such a cluster, as you will see in the next paragraphs. It is more a question of cost and hardware.

DB2 High Availability Disaster Recovery

The DB2 High Availability Disaster Recovery feature (DB2 HADR), is a database log replication feature that provides a high availability solution for various failure scenarios, like the failure of an individual system or even a whole site. High Availability Disaster Recovery continually replicates data changes from a primary source database to a target, or standby database. This protects against data loss. On a failure of the primary source database, the standby database becomes the new primary database.

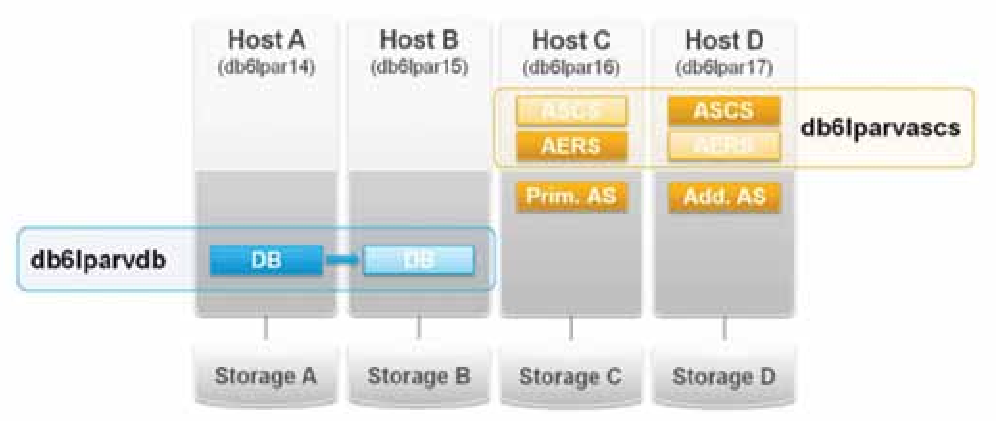

Figure 6: Setup on four cluster nodes..

The next paragraphs will describe DB2 High Availability Disaster Recovery and the System Automation for Multiplatforms High Availability policy based on the distributed system installation.

The DB2 HADR solution is very elegant for maintaining a "hot standby", but the actual takeover making the standby database the primary one, and potentially reintegrating the old primary database again requires manual intervention. This manual intervention consists of a specific HADR commands. Therefore, Tivoli System Automation for Multiplatforms can be used to provide an automatic takeover, that is for example detecting the outage of the primary system and automatically issuing the specific DB2 HADR commands.

System Automation for Multiplatforms is part of each DB2 installation, and the High Availability Disaster Recovery setup utility, db2haicu, can be used to configure HADR and a respective System Automation for Multiplatforms policy.

The setup of such a DB2 HADR is tightly integrated in the SAP installer sapinst.

The System Automation for Multiplatforms high availability policy

The System Automation for Multiplatforms high availability policy is a feature of System Automation for Multiplatforms and provides a HA solution for the SAP Central Services, the SAP Application Servers, the SAP router and the SAP web dispatcher. The HA policy feature comes with comprehensive documentation on SAP installation and configuration for high availability, a setup wizard to gather all required configuration parameters for setting up a policy, and the policy logic itself.

A System Automation for Multiplatforms policy consists of two major parts — a structural definition and scripts to start, stop and monitor the applications that are to be kept highly available. Such a structural definition can contain dependencies between application parts. For example, an application part must only be started when another part is already available, or grouping concepts in order to treat several application parts as one logical unit. Usually, customers have to write their own structural definitions and scripts. For the SAP high availability policy, the definitions and scripts to keep a SAP system highly available are part of the product. They result from long and successful customer relationships and are well tested and very stable. This section of the paper will focus on the policy part that covers the SAP Central Services. The SAP web dispatcher and the SAP router are described in the System Automation for Multiplatforms product documentation http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.samp.doc_3.2.1/HALICG21.pdf.

In order to get to a highly available SAP Central Services configuration, perform the following steps:

- Install SAP: Starting with SAP kernel version 7.10, the SAP installer sapinst contains the installation option “High-Availability System” and allows for a separate installation of the enqueue and the enqueue replication server. In our example, the installation needs to be performed on two nodes, host A and host B. Host B will initially host the enqueue server (ES), host A the enqueue replication server (ERS). The System Automation for Multiplatforms product documentation describes details of the required SAP installation steps.
- Install and configure System Automation for Multiplatforms: On host A and B, System Automation for Multiplatforms has to be installed and a cluster has to be configured. This usually requires two steps on each system and another two steps that have to be executed once the cluster has been established.
- Configure and activate the SAP HA policy: On either host A or host B, the System Automation for Multiplatforms SAP policy wizard has to be executed, and finally the SAP policy has to be activated. The policy wizard is a comprehensive command line tool that interactively asks for all required configuration parameters to keep an SAP system highly available. These include the SAP system ID, like AX6, instance IDs of the ES and ERS (such as ASCS00 and ERS01) and more.

Once these setup steps have been completed, System Automation for Multiplatforms takes over control and continuously monitors all components.

In case of a failure of host B — the node of the enqueue server System Automation for Multiplatforms quickly detects that one node of the cluster is gone, and the enqueue server is no longer avaiblable. System Automation for Multiplatforms therefore triggers a start of the enqueue server on host A.

The enqueue server obtains the shared memory object of the enqueue replication server and terminates the enqueue replication server. Finally, it recovers the lock table from the replica. If host B comes back online, System Automation for Multiplatforms then starts the enqueue replication server on host B.

For end users logged in to the application server on host A and any other than host B, this takeover process is completely transparent. If an end user had started a transaction using an application server during the enqueue server failure, this transaction would hang until the enqueue server is back online on the node of the enqueue replication server and the lock table has been taken over. This usually completes in less than 25 seconds.

Summary

With the SAP High Availability feature of System Automation for Multiplatforms each component of an SAP system can be made highly available. Together with detailed documentation that describes the recovery steps for each failure scenario, failure compensation time can be predicted. This allows proper planning for service level agreements.

The SAP High Availability feature of System Automation for Multiplatforms also allows to predictably plan project time lines when SAP is to be set up for high availability.

Basically the SAP High Availability feature gives back control to its users during the initial implementation phase as well as later when running an SAP cluster.

References:

SAP Help Portal
http://help.sap.com/

SAP Developer Network
http://www.sdn.sap.com/irj/sdn

IBM Tivoli System Automation for Multiplatforms Installation and Configuration Guide
http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/topic/com.ibm.samp.doc_3.2.1/HALICG21.pdf

De Silva, Raspudic, Kamath *"DB2 system topology and configuration for automated multi-site HA and DR"*,
IBM Toronto Lab, February 2010.

Authors

Subhayu Chatterjee is the product manager for IBM® Tivoli® System Automation for Multiplatforms and Application Manager

Hinnerk Gildhoff works as an software developer for IBM in the SAP DB2 development team.

Besides the installation of SAP systems on DB2, Hinnerk is focusing on distributed technologies such as LDAP and high availability scenarios in the SAP DB2 for LUW world. His experience covers various platforms, including Linux, UNIX and Windows as well as Microsoft Cluster Services, IBM Tivoli System Automation for Multiplatforms, Virtualization and others.

Markus Mueller is the lead developer for IBM Tivoli System Automation for Multiplatforms.

Isabell Schwertle works as a software engineer for IBM in the Tivoli System Automation development team. She is responsible for the SAP High Availability policy and topics concerning disaster recovery, virtualization and cloud.

Steffen Siegmund works as a senior developer in the IBM DB2 for Linux, UNIX, and Microsoft Windows development team at SAP. His area of coverage includes database monitoring based on SAP NetWeaver BW, integration with SAP Solution Manager and the setup of high available cluster solutions with IBM DB2® for Linux, UNIX, and Windows and IBM Tivoli System Automation for Multiplatforms.

IBM Global Services
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
August 2011
All Rights Reserved

Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

[1] Supplier relationship management, supply chain management, product lifecycle management, enterprise resource planning, but not SAP POS software which is based on IBM WebSphere® Application Server.

♲ Please Recycle