Architecting a high availability SAP NetWeaver infrastructure

Strategies for ensuring a successful, cost-effective implementation

by Matt Kangas

Matt Kangas
Product Manager,
SAP NetWeaver,
SAP Labs, LLC

Matt Kangas joined SAP Labs in 2003, where he has worked as an SAP NetWeaver Product Manager specializing in systems topics. His areas of coverage include software lifecycle management in both ABAP and Java, IT landscape and architecture, installations and upgrades, system management, high availability, platforms, and the Internet Transaction Server. Prior to working for SAP Labs, he was a systems consultant for SAP America for more than five years. Matt has a B.A. in economics from the University of California at Berkeley. You may reach him at matt.kangas@sap.com.

In today's business world, automation and computing have become key differentiators that can increase process efficiency and productivity. This is especially the case with the rise of the Internet, wireless networking, radio frequency identification (RFID), and Web services. With increased automation levels comes the increased need for reliable availability of the systems that support these business processes, like SAP NetWeaver.

Providing high availability (HA) for an enterprise service-oriented architecture (enterprise SOA) like SAP NetWeaver is a challenge, however — typical setups include multiple integrated SAP systems that are required around the clock for continuous, one-step business scenarios. This article will help you assess the SAP NetWeaver architecture, its configuration, the procedures involved, and the implication of these elements on systems availability; it offers recommendations for formulating an HA strategy for your own SAP NetWeaver-based systems.

In the first part of the article, I present a summary of HA basics to help project planners and managers understand the strategic importance and role of HA within the SAP system landscape. In the second part, I discuss the technical details of an HA setup with SAP NetWeaver, including topics such as architectural single points of failure (SPOFs) and ways to isolate and protect such failure points. Armed with the understanding of HA provided by this article, both system architects and administrators will be able to implement their own HA setups with confidence.

Note!

This article applies to SAP NetWeaver '04 and SAP NetWeaver 2004s, and applications based upon these releases, such as the mySAP Business Suite.

Availability description	Availability in %*	Downtime per year
Conventional	99.0	3.7 days
Highly reliable	99.9	8.8 hours
Highly available	99.99	52.6 minutes
Fault resilient	99.999	5.3 minutes
Fault tolerant	99.9999	32 seconds
Disaster tolerant	99.99999	3 seconds
* Percentage of system uptime during a given time period		

Figure 1 System availability measurements (source: Harvard Research Group)

The fundamentals of HA

In the next sections, I briefly take you through fundamental HA concepts, including what constitutes HA and the causes of system downtime, the tradeoffs involved in increasing availability, and who is responsible for creating and maintaining an HA system. A good understanding of the considerations involved in each of these areas will provide you with a solid foundation for devising your own HA strategy.

What is HA?

HA addresses the critical business need to avoid unplanned downtime. An unplanned downtime takes place if an application crashes unexpectedly — i.e., if a “failure” occurs. In many cases, the failure is caused by a hardware fault. Other frequent reasons are a crash of the host operating system or human error, for example. There are also situations that require a restart or temporary downtime of the application itself, such as updates, security fixes, patches, or tests. These situations are referred to as planned downtime. Since planned downtime by definition is expected, most HA efforts are centered on avoiding unplanned downtime, which is more risky — for example, an unplanned downtime of an SAP system during working hours would probably have more of an impact on business operations than a planned downtime on Sunday. However, reducing planned downtime should be a part of any complete HA strategy.

In Service Level Agreements (SLAs) created by hosting providers and computing centers for their customers and users,[1] system availability is usually measured in percentage of uptime per year, as shown in **Figure 1**. In general, these SLAs define the term “high availability” as a system availability rate of at least 99.99% (52.6 minutes of downtime per year).

> *Note!*
>
> **Figure 1** also helps to illustrate the difference between perception and reality, sometimes called “the myth of the nines.” Of course “Fault tolerant” and “Disaster tolerant” availability would be better than “Fault resilient”; so would 100% availability. But the reality is that not only are those availability levels not currently achievable — or achievable only at an extremely unreasonable cost — what could you even accomplish in any planned downtime at those availability levels? Can you perform maintenance on a system in 32 seconds of downtime per year, or 0.6 seconds of down-time per week? Such levels could be possible if a system were completely static, but the dynamic nature of business requires system changes, which in turn requires some amount of downtime, restart, and so on.

[1] For more on SLAs, see the *SAP Professional Journal* article, “Defining SAP Service Level Agreements: An IT Manager’s Survival Guide” (November/December 2000).

Note!

Even though the numbers in **Figure 1** may look pretty impressive, they are valid only for unplanned downtime. The calculations do not include any planned downtime estimations for offline backups, upgrades, updates, security fixes to applications and operating systems, and so on. By their very nature, the business anticipates these activities and can take actions to mitigate their impact; HA measurements, on the other hand, address events that have an unknown and unplanned occurrence. As previously mentioned, in a real-world scenario you will need to consider planned downtime as well.

A "Fault resilient" availability of 99.999% (also known as "the five nines"), is currently accepted in the industry as the best possible availability.

To achieve a scalable and flexible system landscape, professional application servers are usually installed in a software cluster, which means that there are various instances of the same service, available on different physical machines. These physical instances are usually reached via a load balancer that dispatches requests from the clients. In addition to dispatching the workload, this concept also provides a redundant infrastructure that enables resistance to most hardware and software faults.

Despite this redundancy, in many system landscapes there are still services — e.g., the load balancer, central database, and so on — that are unique within the cluster environment and vital for the cluster operation. If one of these services crashes, the whole cluster might not work anymore. These central, non-redundant services are referred to as potential "SPOFs. Therefore, an SPOF is defined as "a hardware or software service that, if failing, will cause the entire system to fail, leading to unplanned downtime."

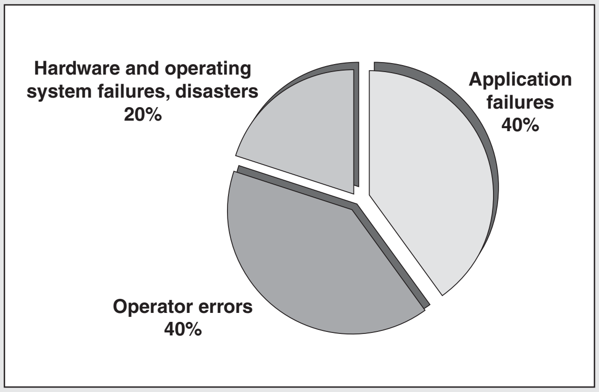


Figure 2 Causes of unplanned downtime (source: Gartner Group)

From a technical infrastructure point of view, architectural and technical SPOFs must be identified and secured in an appropriate manner to help ensure system availability. It is often advisable to use additional hardware and fault-tolerant software so the system can continue operating, uninterrupted, if any single component fails. For example, disk drives could be a hardware SPOF; disk mirroring reduces the likelihood of failure and can effectively eliminate disk drives as an SPOF.

Figure 2 shows the major causes of unplanned downtime. Hardware failures, operating system failures, environment failures, and disaster effects can be avoided by eliminating the SPOFs within systems and implementing disaster recovery scenarios to minimize the impact of possible disasters on the system infrastructure.[2] "Human errors" (e.g., errors caused by faulty configuration, bad change control, etc., as opposed to errors caused by a failed hardware component, earthquake, etc.) make up 80% of downtime causes, though,[3] and these cannot be solved through redundancy or techniques involving switching to alternate resources. Human errors need to be addressed

[2] For more on disaster recovery, see the *SAP Professional Journal* articles, "Is Your R/3 System Recovery Plan a Disaster? A Three-Step Approach for Designing Recovery and Availability Plans" (September/October 2001) and "The 15 Most Overlooked Items in Planning for High Availability and Disaster Recovery" (July/August 2002).

[3] See www.gartner.com/webletter/ibmglobal/edition2/article5/article5.html.

through ease of system management and with improved change and problem management processes. A good source of more information on useful system management software is available at http://service.sap.com/ha.[4]

As mentioned earlier, unplanned downtime poses more of a risk to your technical infrastructure than planned downtime due to its unpredictability, but that doesn't mean that you can ignore the effects of planned downtime. In the end, HA is measured from the perspective of the *end user*. If a system is running, but a user cannot access the system, then the system cannot be considered available. In an Internet sales model, for example, the lack of availability can be critical, since the end user may go to a competitor to complete the transaction. For this reason, it is important to also reduce planned downtimes for tasks such as system and/or infrastructure maintenance, patching, upgrading, and so on through strategies that involve scalable components that enable rolling maintenance, efficient upgrade and patching processes, and proven software lifecycle management engines such as the SAP Transport Management System (TMS) for ABAP and the Change Management Service (CMS) for Java. I go into more detail on some of the features provided by SAP for minimizing planned downtime later in the article.

Understanding the end user perspective of availability will also help with setting and meeting appropriate SLA requirements, as well as with making informed decisions when weighing the costs of increasing availability, which I discuss next.

When do the costs of increasing availability outweigh the benefits?

The main ingredient for successfully increasing availability in any kind of technical system is to provide redundancy. If one component or subsystem fails, at least one other component or subsystem must be available to take over for the failed component. The process of switching from a failed component or subsystem to its redundant replacement is called "failover."[5]

When designing a highly available system, however, you must consider the tradeoff between the costs of increasing your system availability and the costs of system downtime (see **Figure 3**). As you can see on the left side of the figure, the costs of downtime are not linear with respect to the duration of the downtime: with longer downtimes, the increase in costs is closer to exponential. For example, when supply chain processes are stuck for longer than three hours, the entire production process may become stuck, which then produces even higher downtime costs. Conversely, as shown on the right side of the figure, the measures taken to increase system availability — such as redundant components, a disaster recovery site, first-class system management tools, a skilled IT staff, system capacity planning, and proactive services — cost progressively more money to implement.

Because of this HA cost tradeoff, a business case must be made for the availability level of a system. Businesses need to determine their realistic business needs for availability. Measure the cost of system downtime and then balance this availability level with the cost required to provide it. For example, development, sandbox, training, and production systems will all have different availability needs; a reporting system can have more system downtime than an operational system. There may well be cases where 99% availability (outage of less than two hours per week) is good enough.

Who is responsible for ensuring system availability?

Creating highly available SAP NetWeaver systems is a joint responsibility of SAP, the platform/solution partner, and the customer.

SAP's responsibility is to provide an HA-capable integration and application platform (i.e., SAP

[4] From the Media Library, navigate to Documentation → HA Documentation, select the document "BC SAP High Availability NW 2004s," and go to the section "System Automation Software for the SAP Environment" that begins on page 217.

[5] In this article, I differentiate between "failover" (the process of switching to an alternate system in the case of an unplanned outage caused by some failing system element) and "switchover" (the process of switching to an alternate system for a reason other than failure).

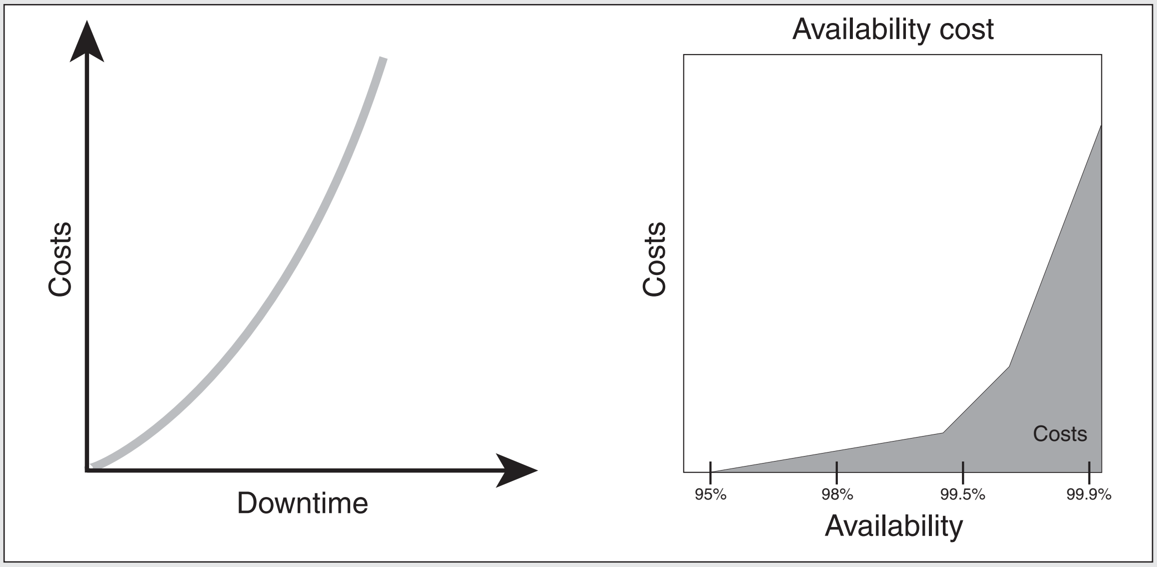


Figure 3 Availability costs vs. system downtime costs (source: Gartner Group)

NetWeaver) and application scenarios that run on top of SAP NetWeaver. SAP systems must be able to utilize the HA-capable computing infrastructure you have in place.[6] SAP also works with partners regarding platform-specific procedures, and in the case of Microsoft Cluster Server (MSCS), provides specific HA procedures and documentation.

HA-capable computing infrastructure elements (hardware, operating system, database system, file system, etc.) are provided by SAP platform/solution partners, who must also provide their platform-specific HA procedures. This includes the HA configuration and switchover ability, and support of the implementation at the customer site.

The customer must define the suitable and required HA levels for their systems. It must also provide the proper IT management concepts and guidelines, and ensure the appropriate operating procedures and training of their IT staff.

Now that you understand the fundamental principles and considerations involved with HA systems, let's take a look at the possible points of failure that can exist in your systems and the different methods you can use to minimize downtime in your system environment, both unplanned and planned, and to improve availability.

Keys to minimizing system downtime and improving availability

Understanding the preventative options available to you for protecting common failure points, avoiding unplanned downtime, and reducing planned downtime will help set you up for success when assessing and planning your own HA implementation. In the next sections, I take you through these options and point out common strategies, recommendations, and best practices for a successful HA implementation.

Protecting common failure points

A system landscape consists of different components and infrastructures, each of which must be checked for potential SPOFs and protected with appropriate

[6] The computing infrastructure, cluster software, etc., are provided by third-party partners.

Cluster types

In the technology world, the word "cluster" has different meanings. There are at least three kinds of clusters:

- **Hardware:** A hardware cluster provides HA by using redundant hardware for every piece of equipment that may fail. At the same time, additional hardware options are needed to control running units and to detect if such hardware has failed. Hardware clustering is the only way to protect (hardware) against SPOFs.
- **Software:** In this scenario, software runs in a distributed environment, mainly to provide scalability to the overall system. At the same time, this means of deployment implies a certain kind of HA, since failing parts may be replaced through the distributed nature of the system. Unfortunately, this technique can become quite complicated, due to system requirements for maintaining the integrity of distributed data.
- **Database:** Database vendors are using the same principles to provide their products with HA. One way is to run a database in a clustered environment. This means that the database is running on more than one machine and in case of failure a switchover will occur. There are different technologies available to achieve this, such as the Microsoft Cluster Server (MSCS) and Oracle RAC, for example.

measures (which may rely on partner technology; more on this later when I discuss switchover solutions). Common points within the system landscape requiring protection and ways to protect them include the following:

- Network — redundant network components and topology; redundant provider links; protected network services (DNS, Mail, LDAP, etc.)
- Storage — RAID technologies (protecting disk availability through redundancy); SAN storage networks with some HA features (split mirrors, synchronous write)
- Server hardware — redundant components (power supplies, buses, coolers, boards); hot-pluggable components; ECC memory; server domains; manageability (remote management, automatic restart, etc.)
- Server operating system (hostnames, IP addresses, file services, applications, name servers, Windows domain controllers) — cluster (several computers coupled to work together as one computer, with shared resources varying from none to all); when one computer fails another takes over its resources and provides them transparently to the outside world (see the sidebar above for more on the different types of clusters)
- Database (vendor-specific) — cluster, replication, shadow database, multi-runtime, etc.

Avoiding unplanned downtime

As stated previously, SAP shoulders the responsibility for leveraging a computing infrastructure that is HA-capable to protect against unplanned downtime. SAP NetWeaver achieves this using four strategies:

- You can install SAP NetWeaver into an HA environment that includes switchover software[7] or clustered resources.

[7] Software that automatically starts a failed service on a different physical host machine. Switchover software reduces unplanned downtime by enabling rapid resumption of a failed service on a substitute host. SAP system services that can be susceptible to failure because they cannot be configured on multiple hosts (such as the DBMS or enqueue and message services) can benefit from the extra resilience of switchover software. The substitute host must be sufficiently powerful to support the additional workload following switchover.

- You can operate SAP NetWeaver in an HA environment — that is, all SPOFs can be protected (either by SAP functionality and/or partner features) by redundant components and/or switchover techniques.
- Software maintenance tools can operate in an HA environment such as a switchover or cluster.
- You can operate SAP scenarios in redundant disaster recovery sites.

Reducing planned downtime

SAP also provides features that help minimize planned downtime, including:

- Kernel upgrades — Enable one-by-one "rolling kernel upgrades" between (SAP-defined) compatible kernels on application servers to eliminate full system shutdown[8]
- Profile parameter changes — Make changes to SAP configuration profile parameters online without restart (for a list of parameters that can be changed online, see SAP Note 102428)
- Operation mode changes — Dynamically change the work process type to adjust to different work profiles (day vs. night, dialog vs. batch)
- Object imports or transports — For standard SAP software maintenance activities (e.g., transports), provide predictability and planning (for system outage) through optimization of software maintenance tools, such as ABAP Transport Management System (TMS) or Java Change Management Service (CMS)
- Applying support packages — Perform a shadow import of support packages; that is, perform a parallel import of inactive new repository objects into the database, then activate them for the runtime environment through a "switch" procedure. The advantage to this technique is that you can import a large portion of the objects of a support package (reports) into the running system without changing the system's runtime behavior, thereby reducing the subsequent downtime for the remaining import. Especially in connection with extensive or several support packages, this reduces the system downtime significantly (see SAP Note 361735).
- Release upgrades — Reduce downtime for release upgrades through a "system switch" upgrade, where many of the upgrade activities (such as imports and activation) are performed in a shadow system in parallel to the running production system, reducing downtime of the production system.[9] To further reduce downtime, SAP also offers the Customer-Based Upgrade (CBU), where customer-specific post-upgrade activities, such as modification adjustments, add-on installations, etc., are performed on a copy of the production system. A custom upgrade package that includes these changes is then created on this system and used for the production system upgrade. The overall user downtime for the upgrade of the production system is thereby reduced, since these activities no longer need to be performed post-upgrade.
- End of daylight savings time — Stretch time in the "double hour" (kernel release 6.40 and higher, see SAP Note 7417)
- Database reorganization — Perform table/index defragmentation, etc., to improve performance[10]
- Offline backup — Enable offline backup, with or without split-mirror technology. Performing a backup on a split mirror has the advantage of less downtime because the system may be started up on the single disk array while the offline backup is performed on the mirror, but the disadvantage is that the system is not mirrored during the backup (unless it is a triple mirror, that is!).
- Online backup — All SAP products allow consistent online backup

[8] This capability is currently in testing and has not yet been released.

[9] For more on the system switch upgrade method, see the article, "A Basis Administrator's Step-by-Step Guide to Preparing for an Upgrade to SAP R/3 Enterprise" (*SAP Professional Journal*, July/August 2004).

[10] For more on database reorganizations, see the article "Boost SAP R/3 Performance by Reorganizing Your Oracle Database: A Proven Reorganization Strategy" (*SAP Professional Journal*, July/August 2005).

With the keys to minimizing system downtime in hand, we're ready to take a test drive through implementing an HA landscape in preparation for devising the HA strategy best suited to your own environment.

Implementing an HA landscape

To implement an HA landscape with SAP NetWeaver, it is best to take a stepwise approach to ensure that you choose the most appropriate HA setup and achieve an optimal balance between cost and availability. The sidebar on the next page provides an overview of recommended design patterns to keep in mind during the HA implementation process. Over the remainder of this article, I walk you through the following five HA implementation steps:

1. Understand the architecture.
2. Find the SPOFs that you need to isolate.
3. Understand ways to isolate the SPOFs.
4. Choose a setup that secures the SPOFs.
5. Implement the system landscape.

> ***Note!***
>
> Before implementing an HA strategy at the software level with SAP NetWeaver, you must have a HA hardware infrastructure already in place. In my experience, I have found that some customers think that setting up their SAP software in a highly available manner is sufficient — this is not the case! You must consider your whole environment, including switches, routers, name servers or Windows domain controllers, and so on. In reality, if your hardware fails, it doesn't matter how well you set up your software, so be sure to collect a list of potential infrastructure hotspots that you should protect at all costs and address them as needed.

1. Understand the architecture

The first, and most important, step in implementing an HA landscape is to understand the SAP NetWeaver system architecture, so you can choose the best setup to meet your particular business needs.

There are three kinds of installation options you can choose among for SAP NetWeaver systems:

- SAP NetWeaver AS ABAP — SAP NetWeaver Application Server[11] with only the ABAP runtime installed

> ***Note!***
>
> SAP NetWeaver AS ABAP systems are based on proven technology from the ABAP-based SAP R/3 world with which most customers are familiar.

- SAP NetWeaver AS Java — SAP NetWeaver Application Server with only the Java runtime installed
- SAP NetWeaver AS ABAP+Java — SAP NetWeaver Application Server with both the ABAP and Java runtimes installed

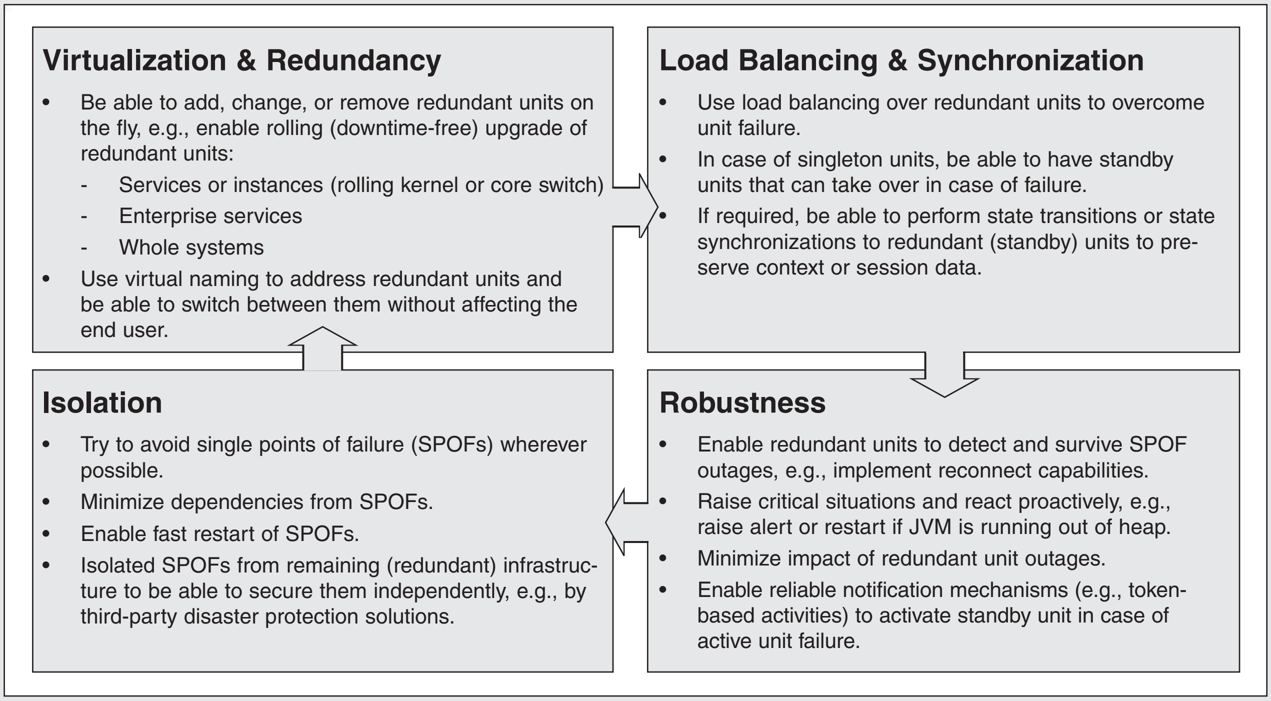
The SAP NetWeaver usage type drives the installation type. For example, SAP NetWeaver Portal requires Java, so an SAP NetWeaver AS Java or SAP NetWeaver AS ABAP+Java system must be installed. Process Integration (PI) with SAP Exchange Infrastructure (SAP NetWeaver XI) requires an SAP NetWeaver AS ABAP+Java system, and so on.

HA setups for SAP NetWeaver AS ABAP are well-known and established from the SAP R/3 world. The latter two installation options are new to HA, so let's take a closer look at these.

[11] Known as SAP Web Application Server (SAP Web AS) prior to SAP NetWeaver 2004s.

Recommended HA design patterns

The following diagram shows design patterns that are common to all successful HA solutions — both hardware- and software-based:



Virtualization & Redundancy

- Be able to add, change, or remove redundant units on the fly, e.g., enable rolling (downtime-free) upgrade of redundant units:
  - Services or instances (rolling kernel or core switch)
  - Enterprise services
  - Whole systems
- Use virtual naming to address redundant units and be able to switch between them without affecting the end user.

Load Balancing & Synchronization

- Use load balancing over redundant units to overcome unit failure.
- In case of singleton units, be able to have standby units that can take over in case of failure.
- If required, be able to perform state transitions or state synchronizations to redundant (standby) units to preserve context or session data.

Isolation

- Try to avoid single points of failure (SPOFs) wherever possible.
- Minimize dependencies from SPOFs.
- Enable fast restart of SPOFs.
- Isolated SPOFs from remaining (redundant) infrastructure to be able to secure them independently, e.g., by third-party disaster protection solutions.

Robustness

- Enable redundant units to detect and survive SPOF outages, e.g., implement reconnect capabilities.
- Raise critical situations and react proactively, e.g., raise alert or restart if JVM is running out of heap.
- Minimize impact of redundant unit outages.
- Enable reliable notification mechanisms (e.g., token-based activities) to activate standby unit in case of active unit failure.

Each of the components of an SAP NetWeaver system can be classified according to these four design patterns. Some examples include:

- **Virtualization & Redundancy:** Application servers are redundant units, because more than one application server (dialog instance) can be installed on separate machines. This type of configuration eliminates application servers as SPOFs, and at least some users will be able to survive a crash of a dialog instance.
- **Load Balancing & Synchronization:** The system's message server can distribute and load balance user requests to the active (redundant) application servers.
- **Robustness:** ABAP-based systems have a "db reconnect" feature, so that if a work process loses its connection to the database (for example, through a crash or network traffic problem), it will attempt to reconnect to the database without aborting. This way, the user session is not lost and the transaction does not have to be rolled back if the work process can reconnect during the (configured) reconnect time period.
- **Isolation:** Since the database and the central services (the enqueue server and the message server) of the system architecturally appear only once within the landscape, they are SPOFs and must be isolated.

SAP NetWeaver AS Java

The SAP NetWeaver AS Java architecture contains many features that customers may not be familiar with (see **Figure 4**). The proven approach from the ABAP stack has been transferred to the Java stack in SAP NetWeaver '04. The Java Central Instance (Java CI) contains the Java dispatcher, which, similar to the ABAP dispatcher, receives client requests and forwards them to the appropriate server processes. It is the Java server process that actually processes the requests and holds the session data. Also included in the Java CI is Internet Graphics Service (IGS) for rendering graphics and Software Deployment Manager (SDM) for managing an SAP Java development landscape.

The SAP Central Services (SCS) Instance contains the Java enqueue (ENQ) server and message (MSG) server, which are also similar to their ABAP counterparts. The Java enqueue server manages the logical locks in the system and ensures server synchronization. The Java message server is the central service for internal cluster communication (such as event notifications, broadcasts, or an exchange of cache content) and provides state information to the SAP Web Dispatcher, which processes and routes Web requests (via HTTP/HTTPS) from external clients/applications to application servers in the SAP system, and balances the load between application server instances.

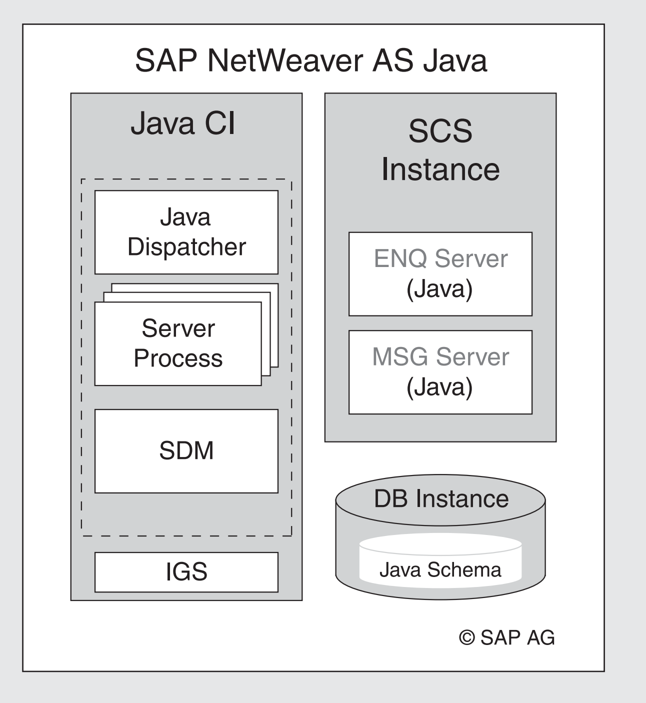


Figure 4 SAP NetWeaver AS Java architecture

The database (DB) instance contains a single Java schema.

SAP NetWeaver AS ABAP+Java

This installation option combines both the ABAP and Java stacks in a single SAP instance — the Add-In Central Instance, as shown in **Figure 5**. The Add-In Central Instance consists of the ABAP Central Instance (ABAP CI) containing the ABAP dispatcher, work processes, gateway, enqueue server, and message server; the Java Central Instance (Java CI) containing the Java dispatcher, server processes, and SDM; and the IGS. In this setup, an Internet Communication Manager (ICM) service manages communications between the application server and external clients/applications — it receives requests from the "outside world" (HTTP, SMTP, etc.) and forwards them to the appropriate stack for processing.

As in the SAP NetWeaver AS Java scenario, the SCS Instance contains the Java enqueue (ENQ) server and message (MSG) server. So, in an ABAP+Java installation, there is an enqueue and message server for each stack.

The database (DB) instance contains two separate schemas — one for the ABAP stack and one for the Java stack.

Once you understand the technical features of the SAP NetWeaver architecture, you next need to identify the failure points that could compromise the availability of your system.

2. Find the SPOFs that you need to isolate

While each SAP NetWeaver installation type has its own specific SPOFs, there are three main points that, if they fail, cause the entire system to fail, leading to unplanned downtime (see **Figure 6** on page 42):

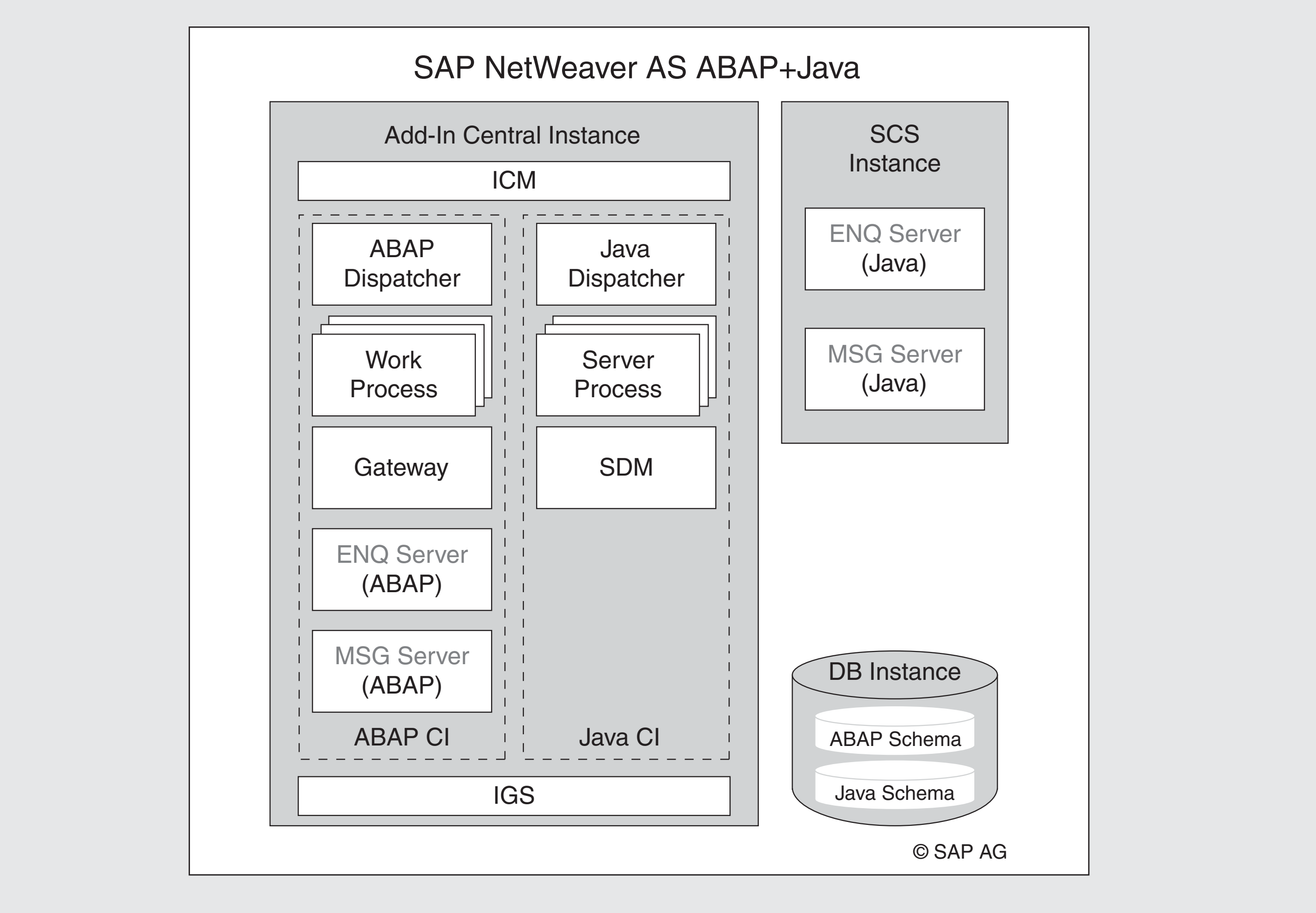


Figure 5 SAP NetWeaver AS ABAP+Java architecture

- In all three installation types (SAP NetWeaver AS ABAP, AS Java, and AS ABAP+Java), there are the central services that, architecturally, appear only once in a system. These are the enqueue and message servers that for Java reside in the SCS Instance, and for ABAP reside in the ABAP CI. In an ABAP+Java installation there is an enqueue server and a message server for the ABAP stack that resides in the Add-In Central Instance as well as an enqueue server and a message server for the Java stack that resides in the SCS Instance; these must be secured. (Note that the SDM appears only once in the architecture, but it is not considered to be an SPOF because it is not a runtime-critical component.) I demonstrate some possibilities for isolating the central services to protect your system from downtime in step 3, with a particular focus on the enqueue server since the consequences of its failure can be severe.

- The central database instance is an SPOF that must be secured. In step 3, I show you a useful way to secure the central database instance.

- Load balancers (such as SAP Web Dispatcher) and other Web infrastructure components (such as reverse proxies) are SPOFs from an end-user point of view.

  - For the SAP Web Dispatcher, two may be run in a redundant setup, the second able to take over if the first fails. For additional information, go to the SAP NetWeaver 2004s online help at http://help.sap.com, search for "Web Dispatcher," and navigate to Architecture and

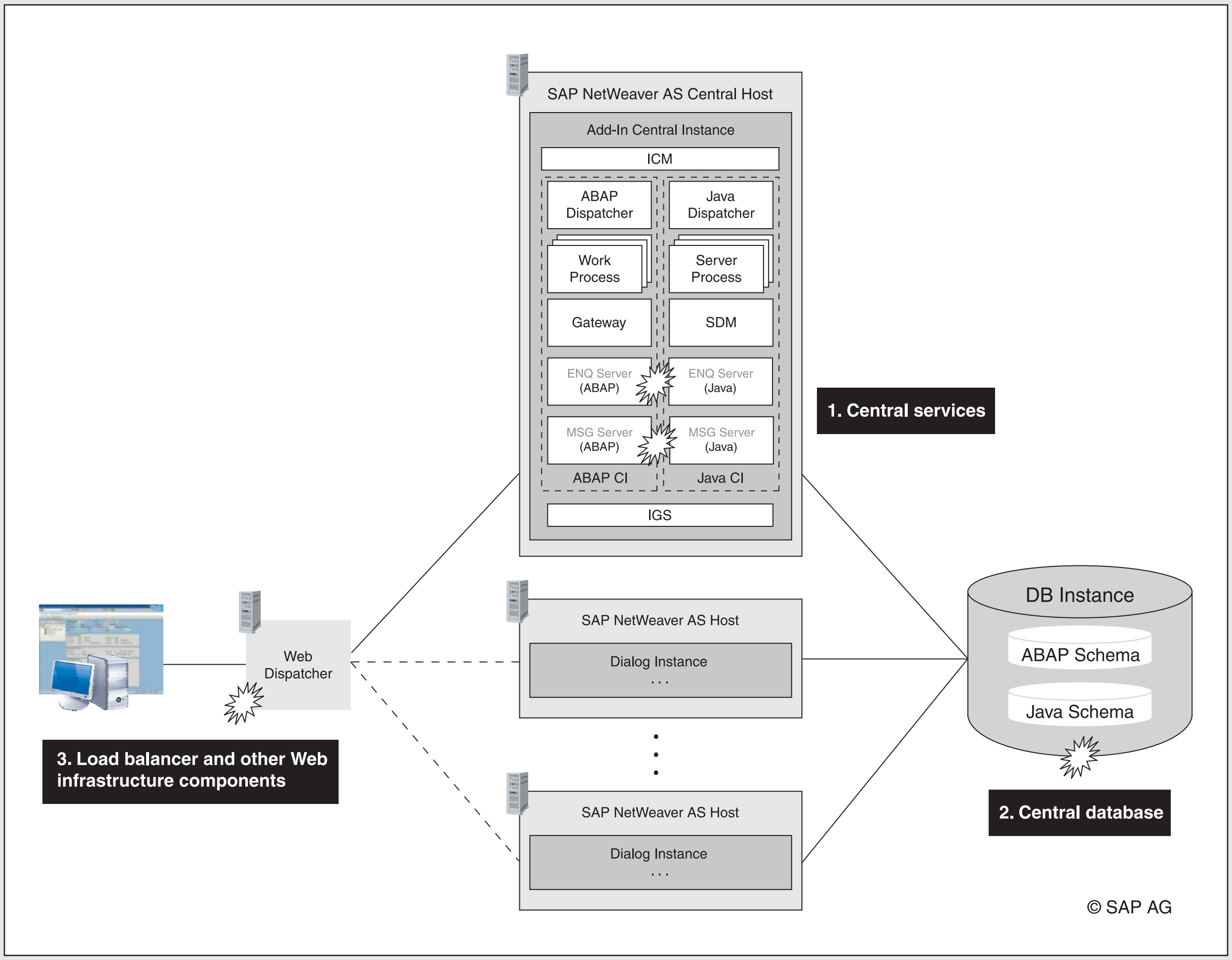


Figure 6 SAP NetWeaver AS architectural SPOFs

Functions of the SAP Web Dispatcher → High Availability of the SAP Web Dispatcher.

- Reverse proxies are a partner technology that you can secure as described in the server hardware and operating system bullet items in the section "Protecting common failure points" earlier in the article. Contact the individual vendor for additional information.

Because configuring SAP Web Dispatcher is straightforward, and reverse proxies are vendor specific, I do not go into more detail on these in step 3.

Note!

In addition to the three identified architectural SPOFs, the central file share (/sapmnt/...) also represents an SPOF from a technical (installation) point of view and needs to be secured at the file system level (for some recommendations, see the bullet item on storage in the section "Protecting common failure points" earlier in the article).

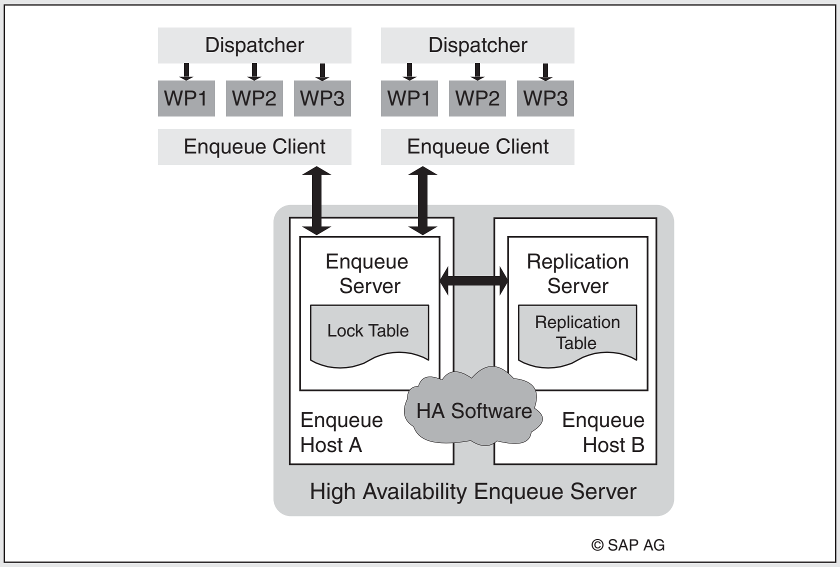


Figure 7 Using the standalone enqueue server with replication

3. Identify ways to isolate the SPOFs

There are different possibilities for securing the central services and central database instance SPOFs identified in step 2 within the SAP NetWeaver architecture. In the next sections, I first show you how to use a standalone enqueue server with replication to avoid session rollbacks, and then I show you how to use a switchover solution to provide redundancy for the central services and the central database instance.

Use a standalone enqueue server with replication to avoid session rollbacks

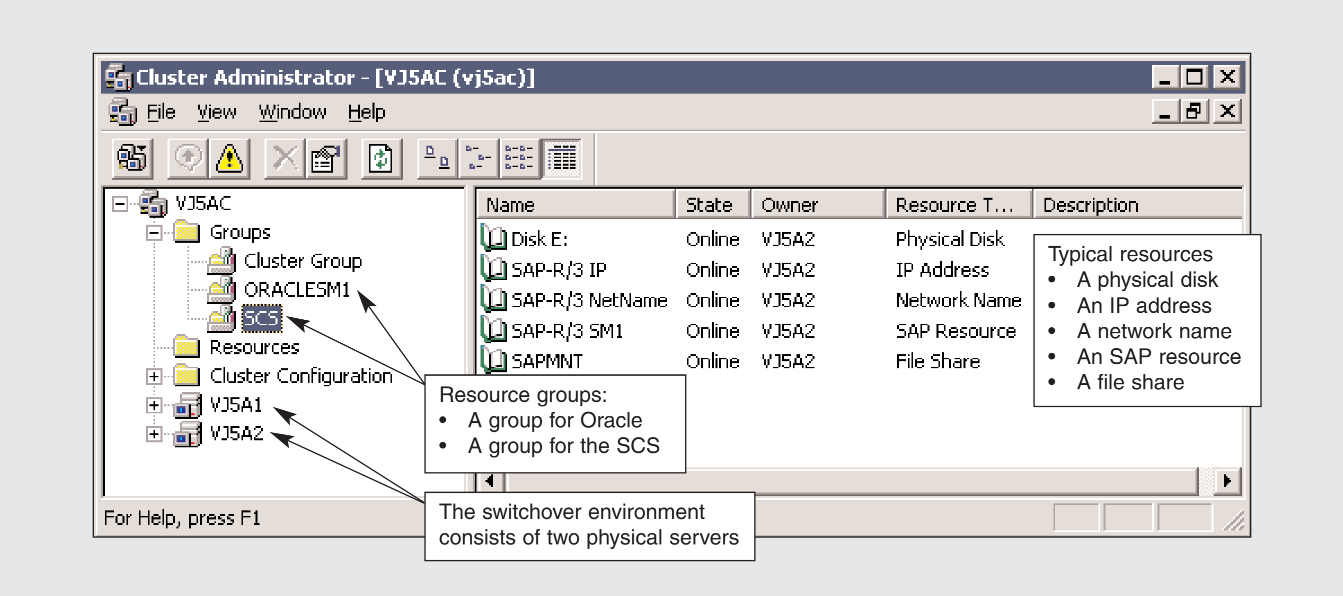
The enqueue server is critical to any SAP system because it manages all of the SAP locks. Although similar in concept in both ABAP and Java, the failure of the enqueue has different consequences within each stack.

For the ABAP stack, the enqueue table contains session-bound locks on data objects. When the enqueue service is lost, the enqueue table also is lost, which causes an automatic rollback of any concerned sessions. However, only these sessions (work processes) are rolled back; you do not need to restart the software cluster.

For the Java stack, the consequences are much more severe. The enqueue table contains locks on data objects *and* infrastructure locks. In addition, these locks are system bound, not session bound. When the enqueue service is lost in the Java stack, the enqueue table is lost, causing a session rollback, just as in ABAP. But, because there are also infrastructure locks in the lock table, a restart of all J2EE instances is necessary (and enforced as of SAP NetWeaver ’04 SPS15)!

To achieve state preservation and thus provide an HA strategy for the enqueue service, SAP provides a standalone enqueue server with enqueue replication. Enqueue replication prevents session rollback due to enqueue server failure and, for the Java stack, must be implemented to ensure an HA strategy.

The HA enqueue server consists of the standalone enqueue server and a replicated enqueue server (see **Figure 7**). The replicated enqueue server runs on another host and contains a replica of the lock table (replication table). The standalone enqueue server is no longer integrated into an SAP application server (central instance) and is instead provided by SAP as an independent program. The main advantage of using the standalone enqueue server is that you can replicate



Name	State	Owner	Resource T...	Description
Disk E:	Online	VJ5A2	Physical Disk	
SAP-R/3 IP	Online	VJ5A2	IP Address	
SAP-R/3 NetName	Online	VJ5A2	Network Name	
SAP-R/3 SM1	Online	VJ5A2	SAP Resource	
SAPMNT	Online	VJ5A2	File Share	

Typical resources
- A physical disk
- An IP address
- A network name
- An SAP resource
- A file share

Resource groups:
- A group for Oracle
- A group for the SCS

The switchover environment consists of two physical servers

Figure 8 Typical example of a switchover environment (MSCS)

the lock table on another host, which means that if the standalone enqueue server fails, you can very quickly start a new standalone enqueue server (on the alternative host) that will continue working seamlessly with the current status of the lock table. All clients and the replication server are connected to the standalone enqueue server.

In the case of SAP NetWeaver AS ABAP, each work process has a separate connection to the standalone enqueue server. With SAP NetWeaver AS Java, each server process in the J2EE cluster has a separate connection to the standalone enqueue server.

In step 4, when I take you through possible HA setups, I show an example configuration that incorporates a replicated enqueue server.

Using a "switchover solution" to provide redundancy

To fully safeguard the central services (enqueue server and message server) and the central database, all HA setups for SAP NetWeaver '04 must include at least one "switchover solution." Switchover solutions provide the necessary redundancy for these SPOFs because services can be automatically switched from a failed host to a standby host in the event of failure, allowing SAP system operation to continue. Outside of the switchover solution, it is just a basic setup of SAP NetWeaver '04. It doesn't matter how the installation is distributed and how many dialog instances are installed! In each HA setup, third-party software is involved to execute the switchover.

HA solutions are heavily platform-dependent and rely on third-party switchover solutions such as:

- Microsoft Cluster Server (MSCS)
- HP Service Guard
- SUN Cluster
- Veritas Cluster Server
- Oracle Failsafe, Oracle RAC
- IBM HACMP
- SteelEye LifeKeeper
- EMC Autostart

The vendors of these solutions may also offer specific consulting and support. Contact the individual vendor for additional information.

In general, switchover products are capable of monitoring and controlling different system resources such as host machines, network adapters, and so on. In the event of failure, the service offered by the resource is automatically taken over by a standby resource. To perform and administer these capabilities, the software is configured with all of the elements of the environment. For example, MSCS typically creates the following setup (see **Figure 8**):[12]

- Switchover environment
    - Some number of physical servers combined together as a (hardware) cluster. These servers each have their own physical hostname but present themselves to the other resources under a single virtual hostname, so the other resources are not "aware" of the physical host they are using.
    - The servers within the switchover environment have the ability to share the clustered resources.
- Cluster resources
    - Services offered by the cluster to the outside world that can be failed over to an alternate resource.
    - Resources can be combined into resource groups (see below).
    - Typical resource types
        - IP addresses (e.g., virtual IP addresses)
        - Network names (e.g., virtual hostname)
        - Processes (e.g., msgserver processes)
        - UNC or NFS shares (e.g., \\<vSCShost>\sapmnt\...)
        - Others
- Resource group[13]
    - Collection of resources bundled together. Actions are taken on a resource group as a whole, so the resources within the group should somehow belong together. For example, the central services and central database instance should be in different resource groups because they are logically different resources and the database does not need to be taken offline when the central services are taken offline.
- Actions on a cluster resource or resource group
    - Bring resource or resource group online.
    - Switch resource group.
    - Take resource or resource group offline.

4. Choose a setup that secures the SPOFs

Once you have a clear understanding of the architecture and technology associated with HA and how the potential SPOFs should be addressed, you can identify solid choices for HA setups. See the sidebar on the next page for key criteria and questions to ask when evaluating the possibilities.

SAP NetWeaver '04 provides for the following potential HA setups:

- DB only in a switchover group
- CI, SCS, and DB in one switchover group
- CI and SCS in one switchover group, DB in another
- DB and SCS in one switchover group
- DB and SCS each in its own switchover group

Each of the possible setups has pros and cons. For example, fewer switchover groups may be simpler to configure and administer, but may not meet the evaluation criteria. As described earlier, one of the key ingredients for an HA setup and implementation is failover time. In other words, it is recommended that you keep the switchover groups as lightweight as possible so that when failures occur, the (designed) redundant components can start up quickly and provide the least amount of interruption to the end user. Manageability is a concern, too: one does not want the switchover groups so granular that there are (unnecessarily) too many of them to administer. For

[12] Because this example is an MSCS example, only MSCS terms are used here. For other vendor solutions, resources and group terms differ. Nevertheless, the technology is basically the same.

[13] A "resource group" is known as a "switchover group" in generic HA terminology.

Key HA evaluation criteria and questions

When evaluating possible HA setups, consider the following criteria and questions:

- Degree of HA functionality (remaining SPOFs)
  - Are all SPOFs secured and thus eliminated?
  - How many of them remain?
- Implementation effort
  - How long does it take to implement the HA solution?
- Failover detection
  - How does the switchover software detect that it needs to switch over?
- Failover time
  - How long does it actually take to bring all resources online after a switchover?
- Number of necessary machines
  - How many servers (machines) are necessary to implement the HA setup?
- Architectural sustainability
  - Is the chosen HA setup future-proof?

example, although it is possible to have the CI, SCS, and DB in a single switchover group, this does not fully meet the evaluation criteria. In this case, the DB, which usually takes a long time to start up, would be part of a switchover and restart when the message server, which can start up on another machine very quickly, fails.

For these reasons, SAP provides the recommendations listed in **Figure 9** for the HA setups possible for SAP NetWeaver '04. Each HA setup also should be extended by the replicated enqueue server discussed previously.

Recommendations for SAP NetWeaver 2004s

With SAP NetWeaver 2004s, the setup of an ABAP-SCS (ASCS) is possible and recommended. In this case, the CI itself will no longer be an SPOF, nor will it be so "central," as it will then have the characteristics of any other application server (dialog instance).

The five different HA setup possibilities for SAP NetWeaver '04 in **Figure 9** are, in principle, also possible for SAP NetWeaver 2004s. From a technical perspective, SAP recommends the following settings:

- Separate the SCS instances (ABAP and Java) from the CI
- Only SPOFs should be protected by the switchover cluster (to achieve fast switchover times, as well as for simplicity and handling reasons)
- Keep the switchover groups as simple and lightweight as possible (reduce complexity)
- Avoid dependencies between different protected SPOFs as much as possible (try to avoid a common switchover group for different SPOFs)

SAP provides the recommendations in **Figure 10** for the HA setups possible with SAP NetWeaver

HA setup type	SAP NetWeaver AS Java	SAP NetWeaver AS ABAP+Java	SAP NetWeaver AS ABAP*
1. DB only in switchover group	Possible	Possible	Possible
2. CI, SCS, and DB in one switchover group	Not recommended	Possible	Possible
3. CI and SCS in one switchover group, DB in another	Possible	Recommended	Recommended
4. DB and SCS in one switchover group	Not recommended	Not applicable	Standalone instance for MSG/ENQ server
5. DB and SCS, each in its own switchover group	Recommended	Not applicable	Standalone instance for MSG/ENQ server

* In an ABAP-only installation, there is no SCS instance.

Figure 9 HA setup recommendations for SAP NetWeaver ’04

HA setup type*	SAP NetWeaver AS Java	SAP NetWeaver AS ABAP+Java**	SAP NetWeaver AS ABAP**
1. DB only in switchover group	Possible	Possible	Possible
2. CI, ASCS/JSCS, and DB in one switchover group	Possible	Possible	Possible
3. CI and ASCS/JSCS in one switchover group, DB in another	Possible	Possible	Possible
4. DB and ASCS/JSCS in one switchover group	Possible	Possible	Possible
5. DB and ASCS/JSCS, each in its own switchover group	Recommended	Recommended	Recommended

* Separation of SCS for ABAP and Java is a prerequisite.

** In an ABAP-only or ABAP+Java installation, it is possible to separate an ASCS instance.

Figure 10 HA setup recommendations as of SAP NetWeaver 2004s

2004s. As you can see, SAP gives a clear recommendation for HA setup type 5 (again, each HA setup should be extended by the replicated enqueue server discussed previously).

Figure 11 on the next page shows how the recommended HA setup type 5 summarized in **Figure 10** is extended with the replicated enqueue server described earlier in step 3. As you can see, this example configuration illustrates the previously discussed technical HA recommendations:

- The ABAP and Java SCS are separate from the CI.
- Only SPOFs are within the switchover cluster.
- The switchover groups are simple.
- There are no dependencies between different protected SPOFs.
- The replicated enqueue server is implemented to provide SAP lock table protection.
- The least number of servers are used to reduce cost.

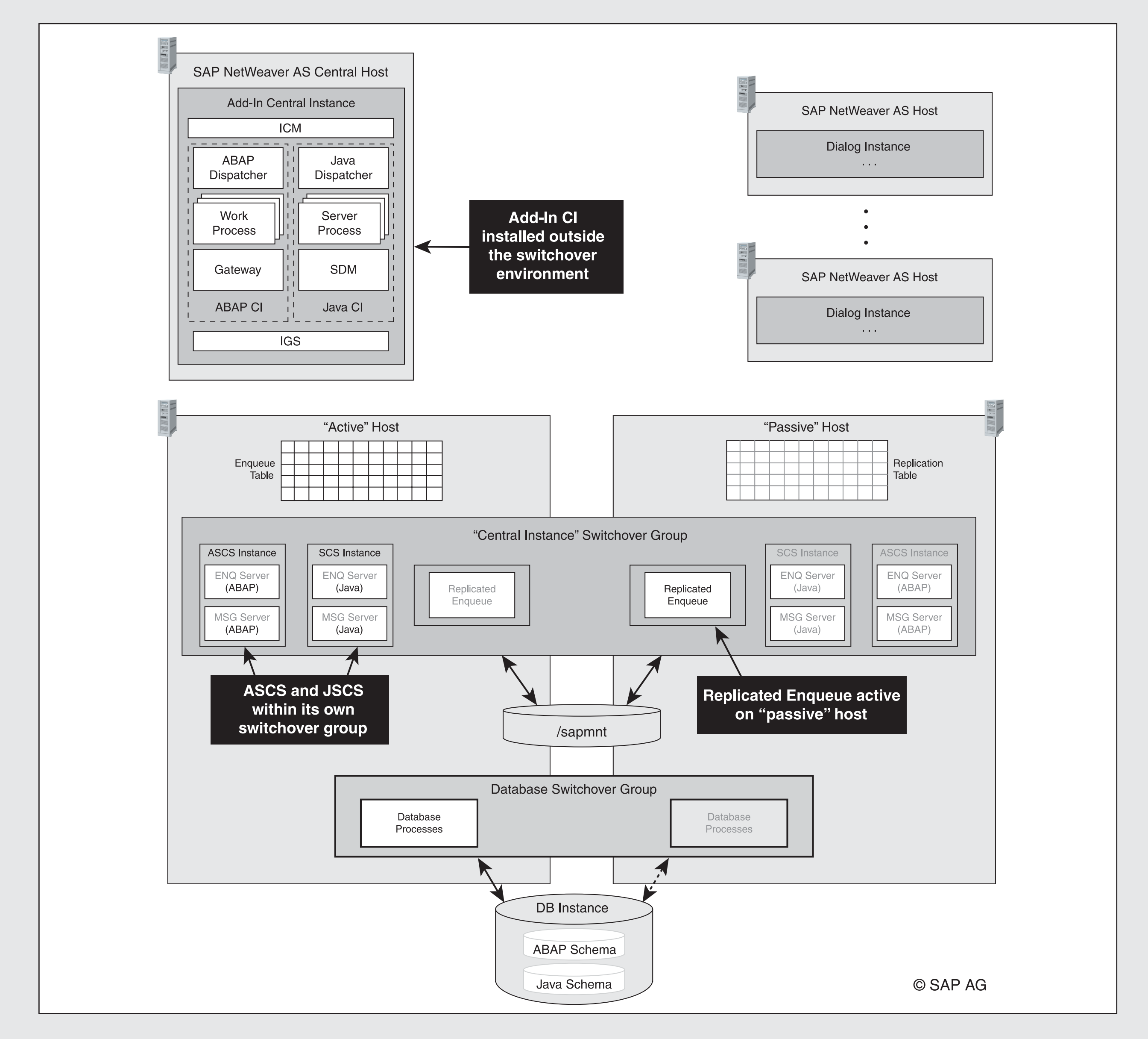


Figure 11 HA setup with the replicated enqueue server

5. Implement your system landscape

Once you have chosen an HA landscape, it is time to perform the implementation! Fortunately, the SAP installation tool sapinst supports HA installations "out-of-the-box," especially through support of installation on virtual hostnames. As a summary, keep the following advice in mind when performing an HA landscape configuration:

- Isolate SPOFs as much as possible.
  - Minimize the impact of a failure and streamline time and effort needed for replacement.
- Provision for a virtual hostname is mandatory.
  - A component that runs inside the switchover environment needs to run on a virtual host because the physical host may automatically change (on purpose).

- Automatic reconnection to all software components is mandatory.
  - In case of a switchover, all dependent software components need to reconnect automatically.
  - Reconnection parameters must be configured according to the expected duration of a failover, which depends on operating system, database, switchover software, HA setup, etc.
- Enable local loading of executables and binaries.
  - If the executable switches over, the local machine is in serious trouble because its binaries cannot be reloaded (TCP connections are broken) — SAP provides the tool SAPCPE to synchronize binaries and executables automatically.

You can now use these five steps on your own systems to design and implement a successful and cost-effective HA infrastructure with SAP NetWeaver. As a final review, let's look at the lessons learned in each step:

1. **Understand the architecture.** A technical understanding of the system architecture is a critical first step toward understanding the failure points within the architecture. There are three different SAP NetWeaver installation types — SAP NetWeaver AS ABAP, AS Java, and AS ABAP+Java — and the business scenario being implemented will drive the necessary installation type to support it.

2. **Find the SPOFs that you need to isolate.** Each SAP NetWeaver installation type has its own specific SPOFs, but there are general areas you should always check, including the central services (the message and enqueue servers), the central database, and any load balancers or other Web infrastructure.

3. **Understand ways to isolate the SPOFs.** Different SPOFs are isolated in different ways. I showed you how to address two key SPOFs — the central services and the central database instance. The standalone enqueue server should be implemented to avoid session rollbacks, and switchover solutions should be implemented to provide redundancy for the central services (the enqueue and message servers) and the central database instance.

4. **Choose a setup that secures the SPOFs.** There are different potential HA setups that you can implement to secure the SPOFs. A setup should not only secure the SPOFs, but also provide fast switchover times, be easy to administer, and be cost effective.

5. **Implement the system landscape.** With your homework complete, you are well prepared to go forward with a successful implementation. Fortunately SAP tools support HA installations right "out of the box."

Conclusion

With the adoption of SAP NetWeaver and the enterprise SOA, more and more business processes rely on IT. Mission-critical business functions such as sales and order entry, continuous manufacturing, and even the ability of consumers to make purchases all depend on the availability of IT services, making those services increasingly mission-critical themselves. All of the players in a business process — internal, external partners, and customers — demand a minimal amount of unplanned downtime of systems and services. Depending on the type of business and service, an outage of one hour can cost millions of dollars. What really counts is the availability of mission-critical services *from an end-user point of view*, regardless of which system or combination of systems is needed to provide this availability.

The consequences of IT failing to meet the demands of business are increasingly costly. Therefore, an important goal for SAP is not only to provide high availability of systems such as SAP NetWeaver and mySAP ERP, but also to facilitate the high- and near-continuous availability of cross-system business processes such as order management, production management, asset management, and more.

This article has examined the implementation of a HA infrastructure from both a strategic and implementation view. Strategically, there is a tradeoff that needs to be balanced between the cost of availability and the cost of downtime. Availability is implemented through redundancy, but at an associated cost — from redundant hardware components to the cost of failover techniques. The higher the degree of system and process availability, the more challenging and expensive that availability is to achieve. A realistic business need for availability must be established balanced against the cost of providing this availability level.

Implementing a high-availability infrastructure involves finding the SPOFs within the architecture — a hardware or software service that, if it fails, causes the entire system to fail, leading to unplanned downtime — isolating them within the architecture and securing them with some sort of redundancy. With SAP NetWeaver, it is recommended that you create simple switchover groups for the software services that are SPOFs and do not have mutual dependencies. In this way, the user impact of failure can be minimized and the time and effort for replacement minimized. The links and SAP Notes in the resource listing available at www.SAPpro.com will aid in the technical implementation of the setup.

Continuous availability of business operations has become increasingly important for many customers. SAP has already strengthened focus and investments in this area over the last few years, and will continue to increase attention and efforts in the years to come.