# HP Asset Manager

# Asset Manager® 5.x High Availability Guidelines

Ensuring That Your Asset Manager 5.x Installation Supports Your Critical Business Needs



hp invent

# Legal Notices

# Introduction

With globalization underway, today's organizations span around the planet and several time zones. One part of the complex challenge of maintaining an up to date overview of the assets managed in every country is to provide a 24x7 service.

This white paper gives hints and directions to find out precisely which high availability solution is best suited to your needs. Asset Manager can be implemented on a wide array of solutions designed for high availability, from low cost simple solutions suitable for a small business to costly fault tolerant environments.

# Definitions

- **HA**: High availability is a system design protocol and associated implementation that ensures a certain absolute degree of operational continuity during a given measurement period.
- **Availability**: Availability refers to the ability of the user community to access the system, whether to submit new work, update or alter existing work, or collect the results of previous work. If a user cannot access the system, it is said to be *unavailable*.
- **Downtime**: Generally, the term *downtime* is used to refer to periods when a system is unavailable.

  A non-leap year has 8760 hours. The following formula calculates the number of hours per year of downtime you can afford:

  N = 8760 x (1- Y)

  Where N represents the number of hours per hour of allowed downtime, and Y represents the desired availability divided by 100.

  For instance, if you desire 99.5% availability, Y will be 0.995. N will then be 43.8 hours / years.
- **RAID**: A redundant array of independent (or inexpensive) disks (or drives) is a family of data storage schemes based on disk redundancy. You will find a detailed discussion of the various available types at this URL: http://en.wikipedia.org/wiki/RAID
- **Computer Clusters**: Several types of computer clusters exist. We will focus in this paper on the High Availability clusters (Also referred to as Failover cluster) and the load balancing clusters. They are two distinct functionalities, but some products do combine both concepts.

  *Failover clusters* are basically a group of two or more computers using a similar installation package. Each computer is called a Node. All the nodes are connected via a high performance sub network.

  At any given time, only one node is active. If for some reason, the active node goes offline, the next node in the row will start up to replace the failed node. The active node may be suffering from a failure or taken off for maintenance…

  Several commercial and open source "out of the box" software exist to smooth out the node swapping operation and eliminate any downtime provided the operation was willingly triggered. If on the contrary a real failure situation occurred, there is usually a short service blackout and the transactions which were underway might be lost.

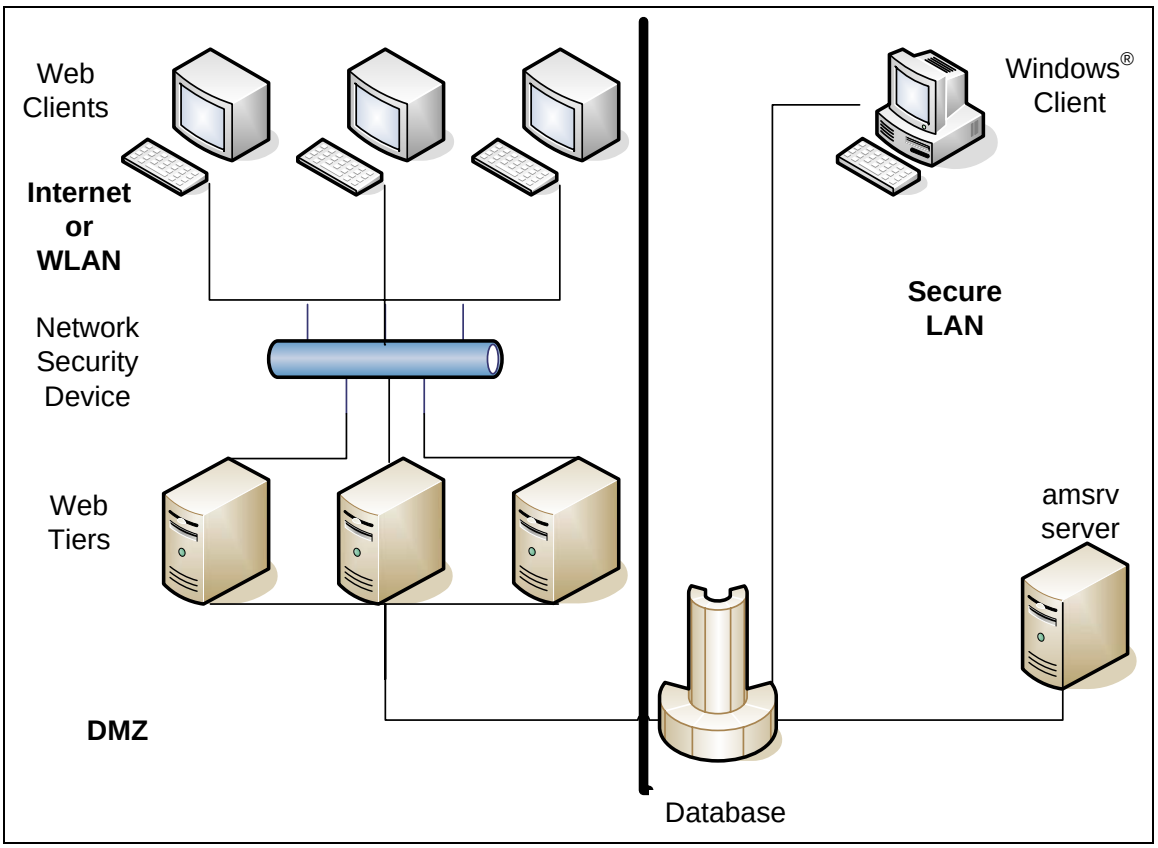
  *Load-balancing clusters* are a group of nodes which are all activated. The group is often called a "server farm."

Each one of those nodes handles a fraction of the total requests that are forwarded by the load balancer. This piece of hardware is usually located before the server farm on the network. The load balancer can be a dedicated device such as an "Alteon® switch" or a dedicated server with the adequate software.

If a node fails, you can instruct the load balancer to remove it from the balancing scheme, therefore keeping the service available, although the users might notice a performance discrepancy since fewer servers process the same number of requests.

Load-balancing clusters primarily address performance issues, but can add additional flexibility to a high availability cluster.

## Overview — Logical components of the platform




The platform can be divided into several layers of services:

- The database
- The amsrv (Asset Manager Automated Process Manager) server
- The Windows® client
- The Web Tiers servers
- The Web clients

You can think of High Availability as a ladder. Adding flexibility, availability and redundancy will take your architecture higher toward the ideal platform. Nevertheless each step has a material cost that you will have to pay to secure it. This functional breakdown supposes that you have already made sure that your network is sufficiently sized and secured for production applications. Let's review each service's importance and function.

## The database

The database is the most crucial single point of the architecture. In High Availability terms, this cannot be a single server. If it becomes completely unavailable, the whole platform will be down and no services will be offered to the users. (Multiple applications may be hosted on the same database server)

We will explore several database means and features to maintain your data online even in case of severe disaster. Going up a level of this ladder is rather expensive. The highest levels should be implemented only if Asset Manager is mission-critical to your organization.

The database is the repository of the data. All other components of the architecture link to it in a direct or indirect way. If it is down, no other part of the service will be working. There is no offline mode whatsoever.

## The amsrv server

This is the second most important part of the system. This node will be in charge of triggering automated tasks like workflows and messaging. When amsrv is down, you might pass several automatic data processes, database performances upkeep and such things.

This might seriously hinder the user experience on the front end machine, and you could also get behind your jobs schedule, making processed data available later than usual.

Keeping this application up and running is important but less complex than other organs of this platform.

## The Windows® clients

The Windows client interacts with the Asset Manager database. It is designed for advanced users, who are mainly administrators and expert functional users. The clients' main cause of failure is poor customization (such as poorly written wizards or workflows) or environment issues (such as a corrupted registry, ODBC connection problems, network issues, and so forth). They can usually be used in two modes, local or distant.

In local mode, the users are responsible for turning it on or off since it is installed on their local workstation. In a production environment, local mode's main edge is to remove this layer from the global availability issues.

In distant mode, the clients are gathered in a server farm served by an application like Citrix®, allowing users to take over a Windows machine by using a remote computer that simply has a Web browser. In this case, some level of redundancy might be appreciated since a crashed server will prevent several people from working. This adds a degree of complexity, but since all the clients are gathered at a given place, hot fixes and version coherency is easier to manage.

## The Web Tier servers

The Web Tier contains the elements that will generate the pages that are sent to the Web clients. These servers will be probably the most exposed to the users. They will allow someone without any specific components installed on the PC to interact with Asset Manager.

You have a myriad of options to consider from the architectural point of view. Functionally speaking you can distinguish two activities:

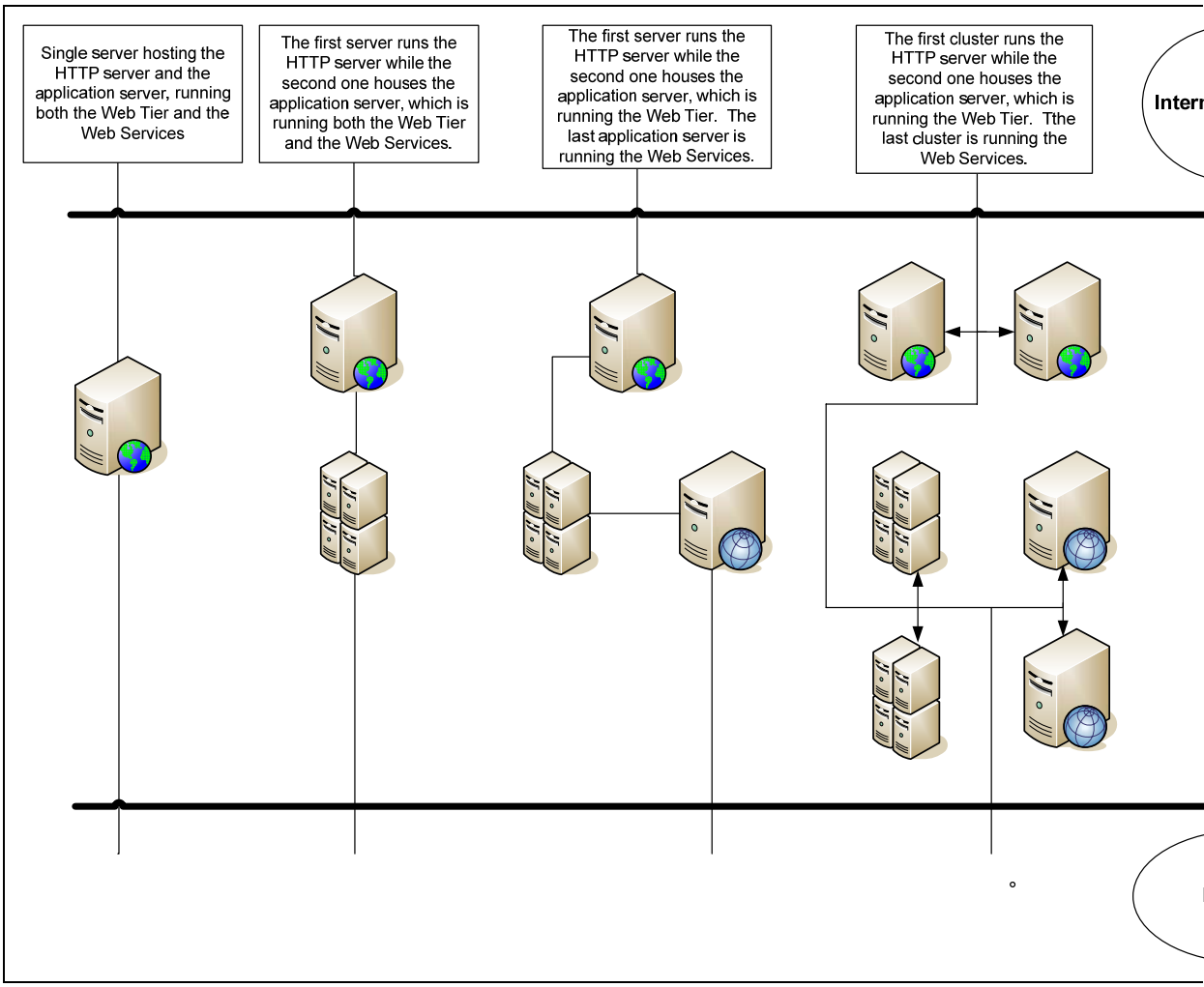
- Compiling the HTML pages combining components stored in the database and pages husks contained locally (This is the application server's job.)
- Broadcasting those generated pages using the http protocol the pages to the end users (This is the HTTP server's job.)

You can either group these functions on a single box or divide them between several boxes.

In the case of Asset Manager, the application server must in fact run two applications:

- Web Services: In charge of connecting to the database and serving the WSDL.
- Web Tier: In charge of collecting the user requests and generating the pages required using the Web Services' components.

Depending on your needs and the axis of growth of your database, for performance purposes, you might consider breaking down the two applications between two application servers, as depicted below:



These servers will need both a load balancing scheme to assure a good quality of service to end users and a failover scheme to guarantee that the service is available to all of the people who need to interact with Asset Manager.

### The Web clients

This is the user's computer. It must be able to connect to the Web Tier servers using the TCP/IP protocol on the port you have decided to use for the service. There does not have to be any specific software installed on them other than Internet Explorer®. They are not part of any High Availability or HP scheme.

## Overview — Securing database availability

As discussed above, this is the single most important node of your architecture.

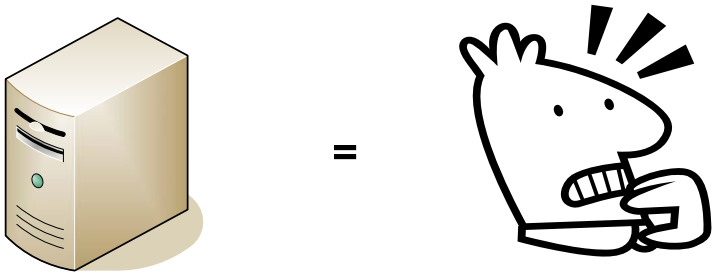
At the moment, several relational database management system (RDBMS) products on the market offer features supporting heavy service data availability. Currently the only RDBMS providing High Availability supported by Asset Manager is the Oracle® 10g database.

We will discuss your options from the minimal to the optimal. Keep in mind that there is a huge financial gap between each level in this ladder.

### Level 0: No redundancy, no backups

This should absolutely never happen in a production environment. If your database suffers any kind of physical damage, you would lose all your data with no chance of getting your data back.

Your organization would have to restart the asset management project as if all the previous work had never existed and face a severe frustration from the users of the application.



***Always back up your data!***

### Level 1: No redundancy, "hard" backups implemented

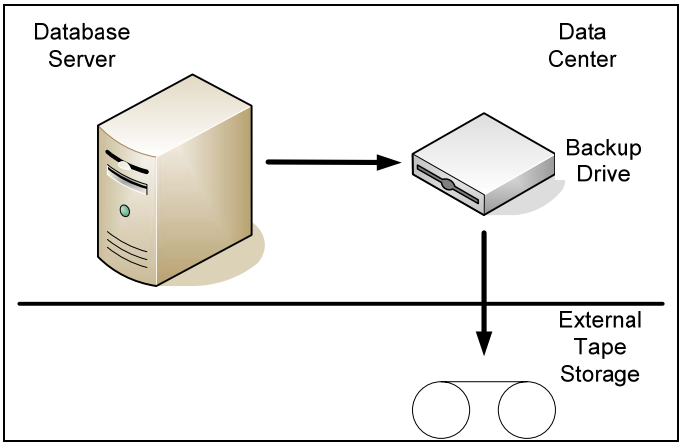
This level requires you to back up your database to a safe external media like tape, using cold integral backup, differential backup, and so forth.

If you are in this position and your backups are regularly verified in a backup recovery test campaign, you could recover and reopen the application to users after a service blackout that could reasonably take between one and two business days.

Note that a successfully implemented backup scheme means external storage of the backup media. If you do not hire an external company to handle this aspect of the backup policy or otherwise arrange for off-site storage of your backup media, your efforts are equal to naught. For instance, a fire might

break out in your facilities. In such a case, your external, off-site backup will give you the opportunity to restart the project quickly.

**Warning**: All data entered between the last backup and a system failure or a catastrophe will be lost for good. This may be manageable, but frustrating to users since they will have to perform transactions a second time.

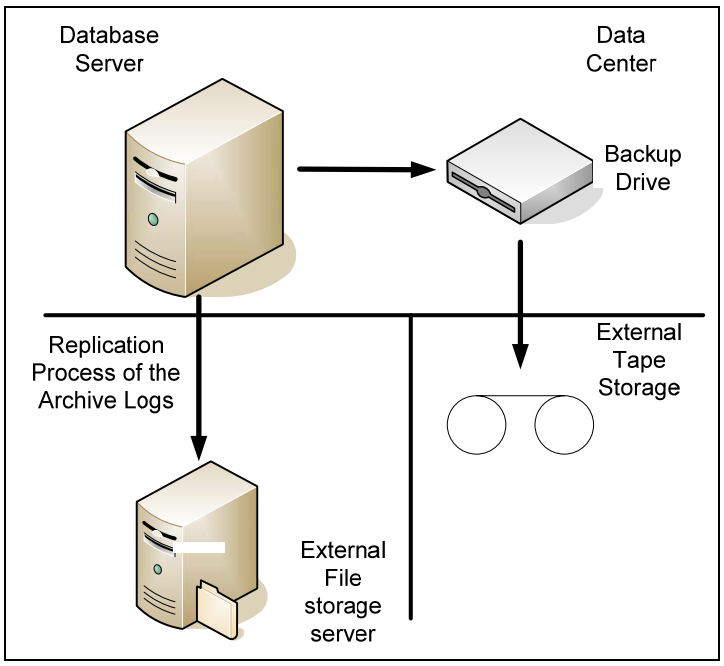



## Level 2: No redundancy, "soft" and "hard" backup implemented

All of what is described above is already implemented, but you also have started to use a database feature like archive logs. The goal of the archive logs is to allow a disaster-stuck data center to re-open the database in the same state as it was when it crashed. The archive logs simply contain the details of the update operations that users performed on the database since it was last backed up.

Once the backup media has been used to restore the database, you just have to feed it the archive logs (the differential backup) in order to recover the data that was not in the last backup. Users can resume working in the same environment they were in when the service went down.

The recovery time would be roughly the same as in Level 1. In terms of investment, you should consider saving the archive logs to a safe network location like an external FTP that you would update every 30 minutes with the latest files; or even better, use a replication process like the rsync open-source utility.




## Level 3: Failover clustering

Failover clustering is a clever, macroscopic way to start guaranteeing high availability of the platform. Your database is broken down into several sub-parts. You can distinguish the database engine (the binaries of your database application) and the data files which contain the actual data.

Experience shows that most unexpected downtime occurs because the database engine stopped without the order to do so. A failover cluster prevents this mode of failure.

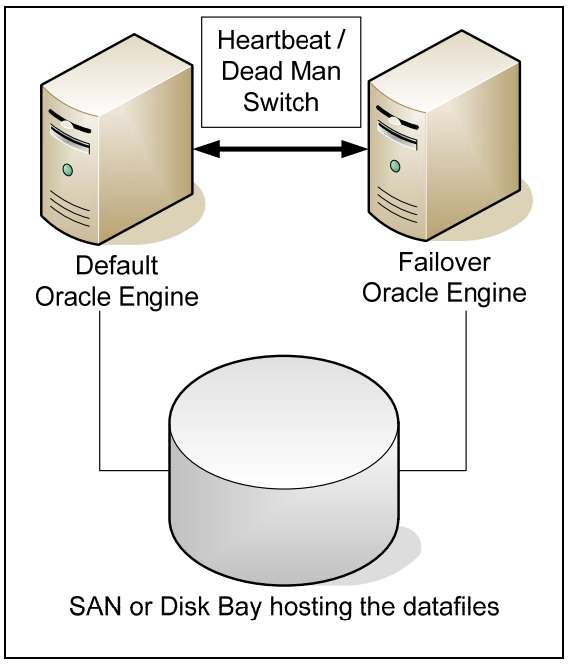
In a simple clustering scheme, there are two identical engines on two separate machines. At all times both physical machines must be running. Meanwhile, the first database engine is on, and the second one is off. A heartbeat process, also called a dead man switch, links both servers. The second server knows the first one is active and serving the data files when it receives a signal on short intervals. If the first server stops emitting the signal, the second server assumes that it is down (frozen, hanging, shut down, and so forth).

At this moment, the database engine on the first server is stopped and the one on the second server is started. The cluster as a whole will also redirect the traffic to the second node so that clients requesting data from the cluster do not have to be reconfigured.

The users will note a short blackout, but as soon as the second engine is properly started, they can resume their activities. Commercial clusters come in several flavors, from pure hardware solutions, to pure software solutions, to composite solutions.

The data files must be stored on a separate structure like a storage area network (SAN) or a file server using RAID. A SAN can be an expensive solution, but it offers high I/O performance and is easily scalable. A file server is certainly cheaper, but will offer lower I/O performance and fewer upgrade options.

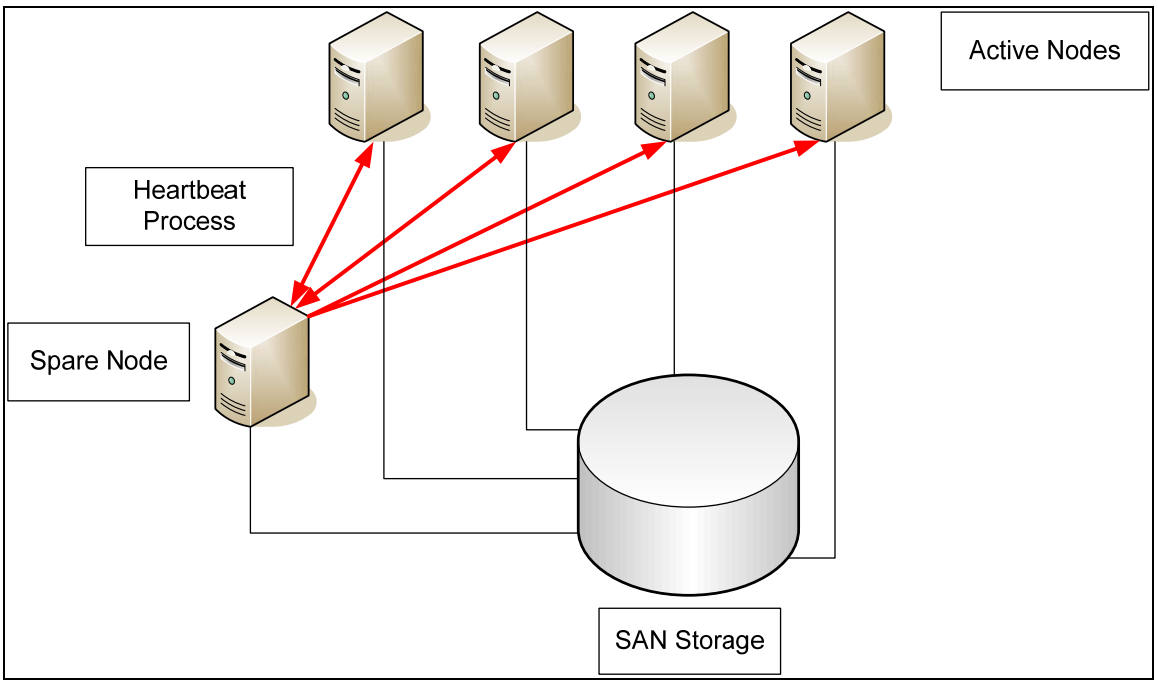
To explore some relevant commercial examples of HP solutions, click here.




## Level 4: High performance clustering

This level of clustering adds better performance to the previous, Level 3 scheme. The storage part remains the same; the difference lies in the database engines. Instead of having just 2 of them (a main node and a backup node), you can have *N*+1 nodes. In this server farm *N* nodes serve the data files, splitting the work load and therefore providing better global quality of service. The last node remains offline and turns on only if another node is taken offline (either for maintenance or because of a failure).

The Oracle RAC (Real Application Cluster) solution supports up to 64 nodes.

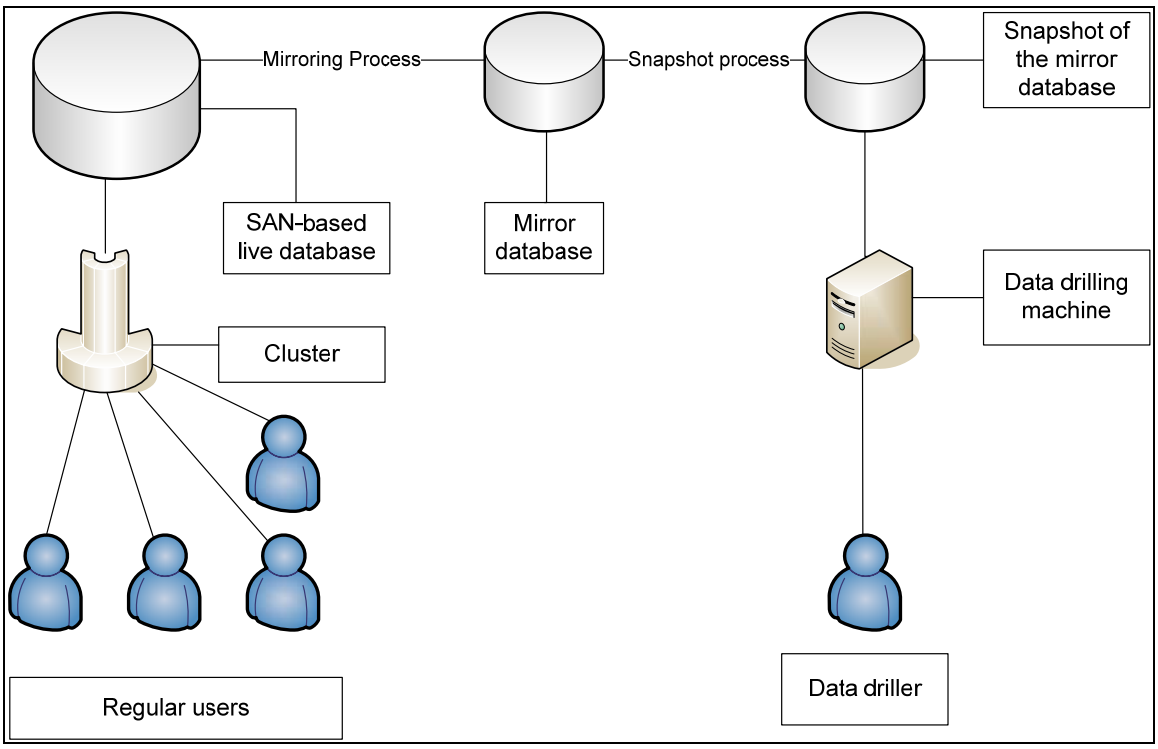



## Level 5: Local mirroring

Level 4 gave you mainly additional performance, and raised the global availability of the platform a bit. Nevertheless, when performing a cold backup of the data files, the service would be turned off. Those backup operations occur inevitably if you use the steps described above in order to guarantee a service restart in case of disaster and to safeguard the database coherency. There is a way around this "scheduled" maintenance.

A High Availability database product nowadays must provide a set of mirroring and snapshot features. They can be used to:

- Provide a quick solution to a SAN unavailability issue.
- Perform a cold backup operation without actually stopping the online database.
- Isolate a data drilling application's accesses from mainstream users' accesses, enhancing their experience by preventing loss of performance.




The database mirroring process generates a full image of itself on another hardware platform (which should be reasonably similar in terms of RAID and inner redundancy). This mirror can be used as the reference for the integral backup of the data or the differential backup. Therefore, the need to stop the live database disappears.

What happens if the live database breaks when the mirror database is offline for a cold backup? No data loss. Since your archive logs are part of a different scheme, you can replace the dead Database with the mirror which has been turned off for a while because of the backup, and then update it with the archive logs which were stored on another remote system.

Note that the mirror database must not be opened to any user. Its sole missions are to be the backup source and the live Database replacement in case of failure.

At this point, you can enjoy this opportunity to add a snapshot process on the mirrored database. This generates an open second instance of the live database separate from the mirror and the live Database which regular users won't be able to reach.

This snapshot can be opened to other users like auditors using Business Intelligence applications or reports generating software. Those particular jobs are only relying on massive Database reading traditionally undermining live database performances.

The snapshot can be hosted on cheaper hardware, but it should absolutely never be considered as a potential recovery referential.

## Level 6: Disaster recovery-ready data centers

This is the penultimate level of high availability. This should be considered only if your Asset Manager installation is "business critical."

After an objective review of your organization's mission, you identify Asset Center as a vital, business-critical component. Without it, your organization would be working in a manner that is unsatisfying to your customers and would lose money on each transaction.

In this case, what would happen if your live instance were destroyed? In addition to natural disasters, organizations are now vulnerable to terrorism, to accidents such as plane crashes and explosions, and to war.

At this stage you have mainly two options: maintain a dormant spare data center or have a secondary live data center.

The dormant data center is often hired from some IT service provider. They could host your backup as well as dormant machines that could be initialized with your latest backups when you instruct them to do so. This option could take a few days - or weeks - to go online.

The live secondary data center is basically a copy of your live data center that is always in synchronization with it. The secondary data center can come online instantly. Infrastructure is essentially duplicated, and a private high bandwidth network between the two data centers must be maintained.

### Level 7: Geographical Cluster

The previous system, Level 6, offers a very high level of reliability. The ultimate step in this High Availability ladder is offered as a service by several housing and hosting companies.

The nodes of the clusters are spread between several (at least 3) data centers. In such a case, the infrastructure cost goes up a lot. In addition to additional servers being added to the failover and load balancing schemes, the network connecting the various facilities must be of the highest quality; otherwise the platform's overall performances will quickly go down.

## Securing server side availability

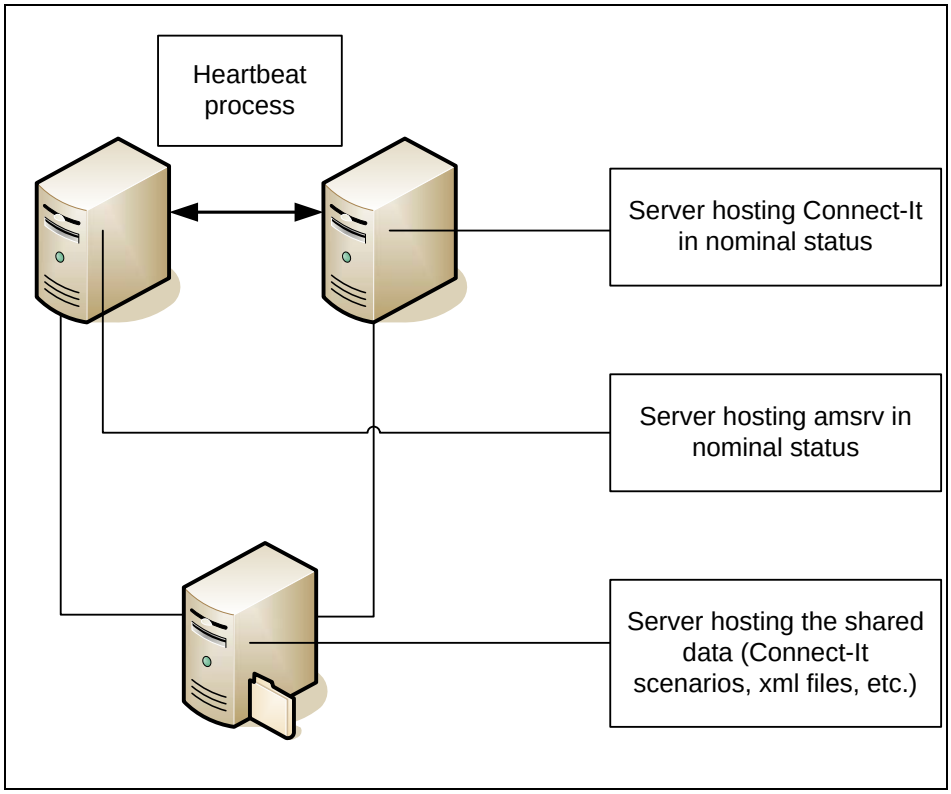
As discussed above, the Asset Manager Automated Process Manager does not handle client connections. It manages the automated processes of the database, and background workflows and processes. The silent tasks it performs have an evident added value. Generally, Asset Manager Server should be on the same network area as the database, for performance issues.

Authorization code verification occurs when Asset Manager Automated Process Manager regularly sends a signal to the database server to indicate that it is functioning. If the database server does not receive a signal from Asset Manager Automated Process Manager for over one hour, a message is displayed when a user connects to the database under Asset Manager. This message indicates that Asset Manager Automated Process Manager has not been launched on this database for over one hour, and that without this process, monitoring functions will be interrupted. If the database server does not receive a signal from Asset Manager Automated Process Manager for over a week, it is no longer possible to connect to the database.

The various tasks performed by Asset Manager Automated Process Manager can be split between several instances. In such a case, the server performing the authorization is the one that needs to be declared in your production license. The MAC addresses of all of its network interface cards (NICs) must be declared in your license file to keep them functioning if one of them fails.

In this example we will illustrate how to use a cluster in a cost-effective way. Instead of having a server permanently on standby, both will be busy handling different tasks. If one server breaks down, the other (functioning) server will handle both applications. During the restoration of the damaged server, the two applications might work in a slightly deprecated mode, but the service provided will not stop.

For this configuration the best recommendation would be to have the amsrv service hosted on a classical 2 nodes cluster along with another application like Connect-It.




In nominal status each server runs an application:

- Server 1 runs Asset Manager Automated Process Manager and houses an inactive Connect-It instance.
- Server 2 runs Connect-It and houses an inactive Asset Manager Automated Process Manager.

In deprecated mode, we have two possible situations:

- Server 1 is offline; Server 2 runs both Asset Manager Automated Process Manager and Connect-It.
- Server 2 is offline; Server 1 runs both Connect-It and Asset Manager Automated Process Manager.

You can switch between nominal status and deprecated mode either manually (if you decide to stop a server to perform a maintenance operation, for example); or automatically when the heartbeat process stops due to a system failure.

A third server will be used for network storage space. Its job is to store the configuration files of Asset Manager and Connect-It. Server 1 and Server 2 are both connected to Server 3. Therefore, both Asset Manager instances and both Connect-It instances will have to fetch their configurations at the same place, guaranteeing that there is no discrepancy between instances.

# Securing front Web server availability

The Web server is the second most important node to secure, right after the database. Most users will perceive Asset Manager as the combination of the database and the Web server. For performance and availability issues, it is highly advisable to use a solution that incorporates load balancing and several physical nodes.

You will have to choose between two options: a software solution which is cheaper since you get those features with the license of your application server; and a hardware solution which is more expensive but gives more control over your traffic options.

## Software solutions

You can use several software options to handle load balancing and high availability, with the WebSphere® application server or Tomcat. Refer to those products' documentation.

Please make sure that those application clusters do not use session replication. Asset Manager 5.x uses large sessions to store the wealth of information it needs and does not support session forwarding between different servers. The server that started the session must manage it until the end.

If you are using WebSphere 6, refer to this IBM® Redbook® for more information about soft clustering: http://www.redbooks.ibm.com/redbooks/pdfs/sg246688.pdf

If you are using WebSphere 5, please refer to this article: http://www-128.ibm.com/developerworks/library/i-wasldbal/

If you are using Tomcat, you can refer to this article which contains some hints on how to use a customized Apache to distribute the load among several Tomcat servers: http://www.javaworld.com/javaworld/jw-12-2004/jw-1220-tomcat.html?page=1

## Hardware solutions

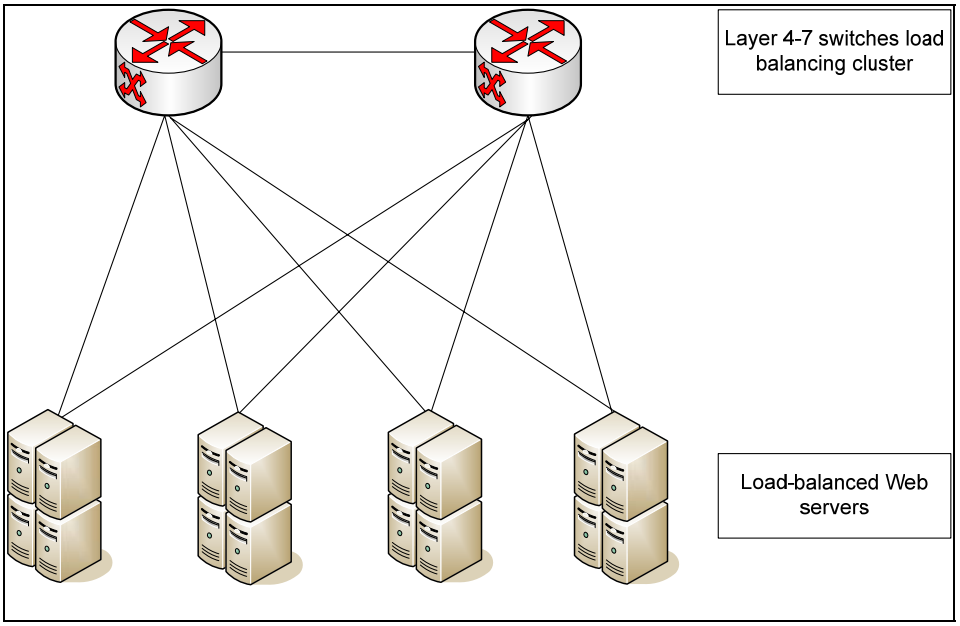
The best approach is to break down your Web server functional area into two sub-layers: the balancers and the server farm.

The server farm is made up of several unitary servers. Each will handle a fraction of the traffic from the initial connection until a logout (or timeout).

The load balancing layer should be made of several Layer 4-7 switches. They are products that will intelligently handle the connections to the Web servers. Depending on the manufacturer, you can have several advanced features like heartbeat processes monitoring the Web servers and automatically removing from the load balancing scheme a down server; automatic failover to another switch connected to the first one; load balancing between several switches of the same type; and filtering of malicious packets (such as denial of service attacks and SYN attacks).

Two companies selling such solutions are Nortel® Networks® and its Alteon line of products, and Citrix and its Netscaler® technology. Many others brands are available in the marketplace. Those products are famous for adding such functionalities as SSL tunneling that can help you insert firewalls between the load balancing layer and the server farm to add security to your Web platform.

Each Web server is configured as a standalone machine connecting to the database platform. Sessions are intelligently handled by the switches in order to guarantee transaction coherency.



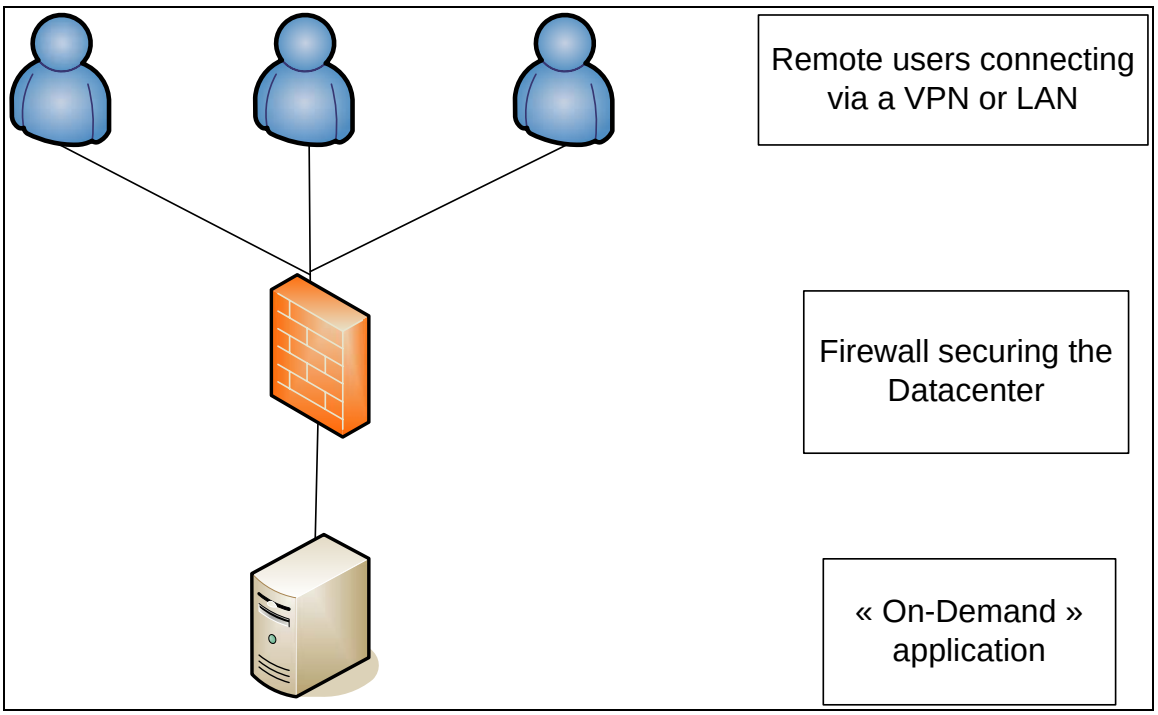

## Securing Windows client availability

Basically, the Windows client has advanced technical features that are not available in the Web client, such as workflow and wizard design, security administration, and so forth. Those features are reserved for only a few users with specific roles and a good overview of their functional domains. You can assume that the typical cause of a crash for this component is a buggy customized script or workflow, or problems with the client operating system, such as a problem with the registry.

After crashing, you only have to launch the Windows client again. Data can not be damaged in such an event. You would simply have to double check the last actions you performed. Two architecture options can be conceptualized

The first one pushes the charge of administering the local installation and management of the Windows client to the advanced users. This is an inexpensive option, but users might have a hard time keeping their own personal versions up to date. Moreover, roaming users can be prevented from working if different branches of the company use different security policies.

If your IT administrators are wary of leaving control of this component to the end users, or want to maximize security by using firewalls on as many components as possible, there is a reasonable alternative.




On this schema, end users connect via a Web browser to a machine serving the Asset Manager Windows client using an "on demand" application like Citrix. Such commercial applications allow a remote user to access all the functionalities using a single port, which eases the deployment of firewalls. Giving full access from the safe "on demand" application to the database or to another network's area is no problem since you can trust this machine. IT administrators can also perfectly master the environment and deploy hot fixes or upgrades to clients with immediate effects for all users.

## Conclusion

Having a satisfactory level of availability can be very costly. Before setting your goal to 0.5 % or 0.1% of downtime, you should determine what degree of Asset Manager availability is important for your organization. Once you have determined where you want to go, it is a matter of adding appliances and learning the necessary skills to administer them.

Even if in the early stages your availability scheme is stretching your maintenance options thin, you can always add additional layers gradually. Asset Manager has a robust architecture that can be easily implemented in a variety of environments.

## References

You will find below additional information on actual hardware that supports clustering:

- The Windows 2003 cluster using HP hardware:

  http://www.windowsservercatalog.com/results.aspx?&bCatID=1291&cpID=897&ocID=0&OR=3

- The Nonstop® range of products from HP:

  http://h20223.www2.hp.com/nonstopcomputing/cache/76385-0-0-0-121.aspx

- The Oracle RAC homepage :

  http://www.oracle.com/technology/products/database/clustering/index.html

- The Nortel Networks® Alteon range of products :

  http://products.nortel.com/go/product_content.jsp?segId=0&parId=0&prod_id=25080

- Citrix Netscaler:

  http://www.citrix.com/English/ps2/products/product.asp?contentID=21679

# For more information

Please visit the HP Software support Web site at:

http://www.hp.com/managementsoftware/support

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valuable support customer, you can benefit by being able to:

- Search for knowledge documents of interest
- Submit and track progress on support cases
- Submit enhancement requests online
- Download software patches
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

**Note:** Most of the support areas require that you register as an HP Passport user and sign in. Many also require an active support contract.

To find more information about support access levels, go to the following URL:

http://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to the following URL:

http://www.managementsoftware.hp.com/passport-registration.html

### Limited responsibility clause

Asset Manager is integrated with several third-party applications. Examples: Database engines, Web servers, single sign-on software, load-balancing and clustering hardware and software solutions, reporting software such as Crystal Reports, etc.

Support for these applications is limited to their interface with Asset Manager. Support does not cover installation problems, setup and customization problems nor malfunctioning of the third-party application.

White papers contain examples of implementations that may work in your environment with or without customization. There is no guarantee that this will be the case. It could also be that some of the solutions covered by white papers appear as standard features in a future release of the software. When this is the case, there is no guarantee that you will be able to upgrade the solution you implemented based on the white paper to the equivalent standard feature.