Paper 358-2011

Considerations for Implementing a Highly Available or Disaster Recovery Environment

Diane Hatcher, SAS Institute Inc., Cary, NC, USA
Jochen Kirsten, SAS Institute Inc., Heidelberg, Germany

ABSTRACT

It is a fairly common practice to clone your SAS®9 metadata-based environment to create a backup for disaster recovery purposes. We discuss what you should consider when designing your backup strategy. The strategy could also have implications on how you should configure your production environment.

AVAILABILITY AND DOWNTIME

“*High availability is a system design protocol and associated implementation that ensures a certain degree of operational continuity during a given measurement period.*”[1]

Translated to a business analytics environment, the previous statement means:

- Assure SAS applications and services are available
- Support business continuity as a goal

When speaking of High Availability it can be interesting to have a look at how the terms availability and downtime actually are defined. The following table, taken from Wikipedia, displays some of the common levels of availability, which is a relatively abstract measure, and what they mean in terms of downtime over a year, month, and week. The latter are basically what an IT manager is interested in when purchasing and operating a hardware component.

Availability %	Downtime Per Year	Downtime Per Month[2]	Downtime Per Week
90%	36.5 days	72 hours	16.8 hours
95%	18.25 days	36 hours	8.4 hours
98%	7.30 days	14.4 hours	3.36 hours
99%	3.65 days	7.20 hours	1.68 hours
99.5%	1.83 days	3.60 hours	50.4 minutes
99.8%	17.52 hours	86.23 minutes	20.16 minutes
99.9% ("three nines")	8.76 hours	43.2 minutes	10.1 minutes
99.95%	4.38 hours	21.56 minutes	5.04 minutes
99.99% ("four nines")	52.6 minutes	4.32 minutes	1.01 minutes
99.999% ("five nines")	5.26 minutes	25.9 seconds	6.05 seconds
99.9999% ("six nines")	31.5 seconds	2.59 seconds	0.605 seconds

Table 1: Some Common Availability Levels and Associated Downtimes

AVAILABILITY OF A SERIAL SYSTEM

A serial system consists of a series of hardware components that are connected to each other in a sequential way. As a result, if one of the components fails, the entire system becomes unavailable.

[1] “High availability,” Wikipedia, available at http://en.wikipedia.org/wiki/High_availability (accessed on February 2, 2011).

[2] For monthly calculations, a 30-day month is used.

Following figure assumes a system consisting of a computer and a Digital Subscriber Line (DSL)-modem. They are connected serially. If any one of these two components would become unavailable the Internet becomes unavailable to the user of the computer.

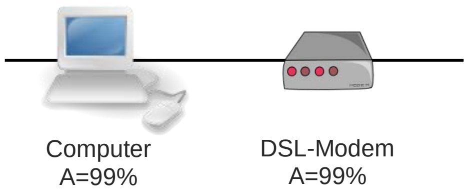


Figure 1: Serial System with Two Components

Assuming the availability of the computer and the modem would be at 99% for each, which translates to a downtime of 3.65 days over a year, the overall availability of the whole system is computed using the following equation.

$$\begin{aligned} \text{Availability serial system } (A_S) &= A(\text{Computer}) * A(\text{Modem}) \\ &= 0.99\ [3.65\ \text{days/year}] * 0.99\ [3.65\ \text{days/year}] \\ &= 98\%\ [7.30\ \text{days/year}] \end{aligned}$$

Equation 1: Calculation of Serial System Availability

As a result the availability of a serial system, consisting of two components with 99% availability each, drops down to 98%. This does not sound dramatic at first. However, when looking at the downtime of the whole system compared to its components, this is doubling from 3.65 days/year to 7.30 days/year.

AVAILABILITY OF A PARALLEL SYSTEM

A parallel system consists of components, which are connected in a parallel way, backing each other up in case there is a failure with one component. Often these components are identical.

Following assumes a system consisting of two server computers. They are connected in parallel. If one would become unavailable the other could still be used.

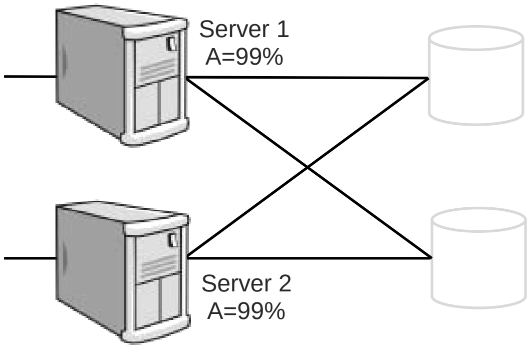


Figure 2: Parallel System with Two Components

Assuming the availability of the servers[3] would be at 99% for each, which translates to a downtime of 3.65 days over a year, the overall availability of the whole system is computed using the following equation.

$$\begin{aligned} \text{Availability parallel system } (A_P) &= 1 - (1 - A(\text{Server 1}))^2 \\ &= 1 - (1 - 0.99\ [3.65\ \text{days/year}])^2 \\ &= 99.99\%\ [52\ \text{minutes/year}] \end{aligned}$$

Equation 2: Calculation of Parallel System Availability

As a result the availability of a parallel system, consisting of two components with 99% availability each, rises to 99.99%. In other words, the downtime of the whole system compared to its components is dropping from 3.65 days to 52 minutes over a whole year.

As for a SAS Grid Computing solution, as discussed in this paper, it is difficult to quantify the overall availability as this depends on the number and availability of the parts it consists of. The availability of hardware components can be inquired from vendor making these parts.

HIGH AVAILABILITY OPTIONS IN SAS 9.2

Some years ago SAS started testing the compatibility of the vital server components, such as the SAS Metadata Server, the SAS OLAP Server, the object spawner, the SAS/CONNECT® Server, and the SAS Scalable Performance Data Server® with third-party high availability solutions from vendors like HP, IBM, Sun, and Microsoft. Most, if not all these solutions, feature a setup where an active server, receiving the entire application load, is backed up by a passive machine, which stands idle waiting for processes to be failed over from the other machine, once this becomes unavailable for some reason. This is called an active-passive setup, introducing redundancy, as only one available machine will be used at a time.

A few years later, when the SAS Grid Manager became available, these solutions were complemented with a high availability scenario featuring an active-active setup using grid architecture. This scenario minimizes hardware redundancy since all machines can be used at any time. In case one of the machines fails, its load will be distributed across the available machines.

HIGHLY AVAILABLE CONFIGURATIONS

The following two types of high availability setups are mainly used today:

- Active-Passive
- Active-Active

Active-Passive Setup

This configuration type consists of at least two computers, forming a cluster. If one computer goes down, the other will take over. The active system is also called primary system, the passive one is called backup or stand-by system.

Often a software component (cluster manager) is used to automate the failover process between the systems. While this setup features redundancy (the passive node typically sits idle) it will not suffer from performance degradation after a failover has occurred, since the available resources will be identical before and after.

The following third-party solutions have been tested to work with SAS:

- HP ServiceGuard
- IBM High Availability Cluster Multi-Processing (HACMP)
- IBM Tivoli Enterprise Console
- Microsoft Cluster Server

[3] For simplicity the disk subsystem is not considered.

- Sun Clusters

The following SAS software components have been tested to failover properly with some of these third-party solutions:

- SAS Metadata Server
- SAS OLAP Server
- SAS/CONNECT Server
- SAS Object Spawner
- SAS Scalable Performance Data Server

White papers and other materials are available for many of them. Contact your SAS account team if you are interested in finding out more about support for specific third-party solutions.

It is common to see disaster recovery systems configured in an active/passive scenario. Typically, the backup, or redundant, system is located in another geographically different environment. It is started up only in the case of catastrophic failure of the primary system.

Active-Active Setup

This type of configuration typically consists of multiple computers, which are all active at the same time. Typically all these systems are configured identically running the same software components. If one system fails, its load will be balanced to the other systems. While there is no redundancy implied in this setup, the performance of the cluster might suffer in this case.

LEVERAGING SAS GRID COMPUTING TECHNOLOGY

Grid technologies elevate the basic concepts of load balancing to a much more robust architecture. You can use grid technologies to better leverage existing resources, while automatically controlling the allocation of resources based on demand.

SAS Grid Manager is an add-on solution that provides grid middleware deployed specifically to support your SAS business analytics environment. SAS Grid Manager integrates SAS technology with components from Platform Computing, a leading vendor of grid enablement solutions, to provide load balancing, policy enforcement efficient resource allocation, and prioritization running in a shared environment. This architecture allows you to decouple SAS applications from the execution infrastructure to provide the ability to transparently grow or contract hardware resources as needed and provide fault tolerance.

The following picture shows a typical SAS grid architecture.

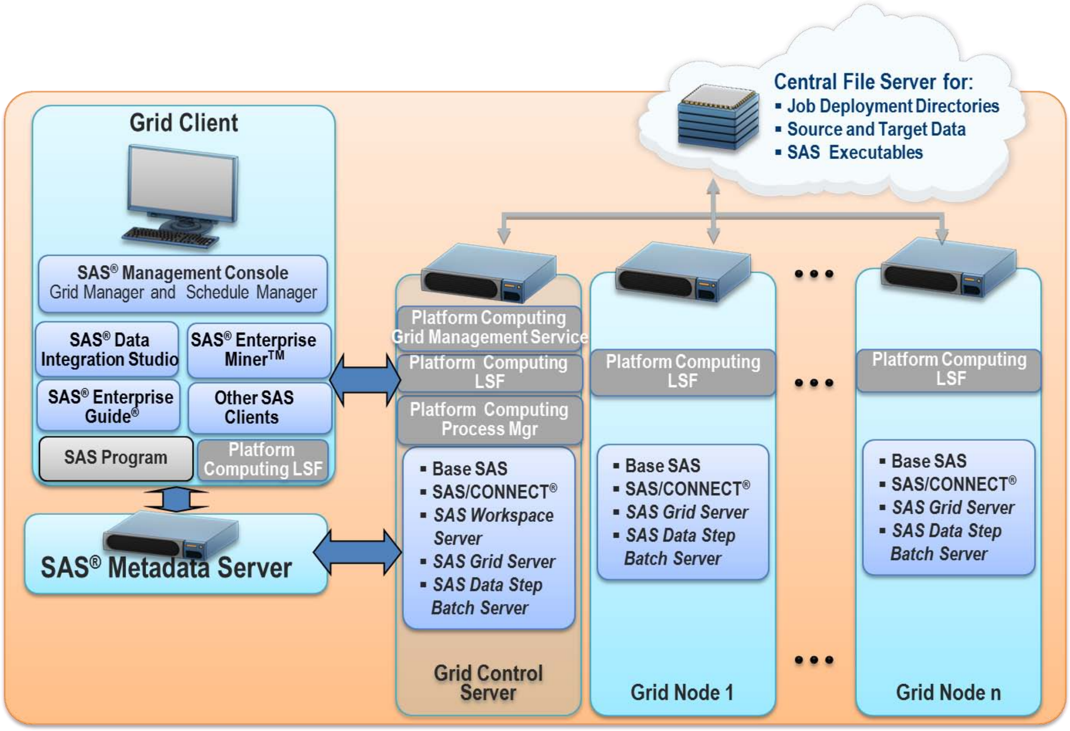


Figure 3: Sample SAS Grid Architecture

The SAS grid architecture includes the following major components:

- SAS Metadata Server – contains the definitions of the grid resources
- Grid Control Server – distributes requests to the grid nodes
- Grid Nodes – receives requests from the Grid Control Server to execute the SAS workload
- Central File Server – a shared file repository across all grid machines to contain job deployment directories, data sources, and log files

HIGH AVAILABILITY ON GRID RESOURCES

When discussing high availability of a SAS grid the following levels of resources should be considered:

- The hardware that makes up the grid infrastructure
- Vital software components
    - Platform Load Sharing Facility (LSF) services operating the grid
    - SAS services providing functionality to clients in the grid
- SAS jobs submitted to the grid

These levels are discussed in the following sections.

BENEFITS OF USING HIGH AVAILABILITY WITH PLATFORM SUITE FOR SAS

If any vital component in the grid architecture would fail, the entire grid would become unavailable to the clients. This applies to the SAS Metadata Server and the grid-specific services running on the grid control server machine. Likewise, if for example a SAS OLAP Server executing on one of the grid nodes would become unavailable, access to the OLAP cubes from the clients would fail, too. The grid can, however, be configured to fail any of these vital services over to another node, hence making it highly available.

Platform Enterprise Grid Orchestrator (EGO) is part of the Platform Suite for SAS Version 4.1 and is provided as part of SAS Grid Manager for 9.2. EGO is a collection of cluster orchestration software components that, among other things, provide high availability to critical services. Besides other functionality, EGO can ensure high availability of services running on the grid through built-in disaster recovery scenarios.

The SAS grid solution provides high availability support across the SAS infrastructure. The compute tier is handled as part of the load balancing support across the grid nodes, as you would expect. However, the SAS grid solution, when using EGO, provides more robust failover support of the SAS Metadata Server than other availability solutions.

CANDIDATES FOR GRID HIGH AVAILABILITY

The services that can be failed over can be grouped into the following three main categories:

1. SAS services
   - SAS Metadata Server
   - SAS Object Spawner
   - SAS OLAP Server
   - SAS/SHARE® Server
2. Grid operational services
   - Platform Process Manager
   - Platform Grid Management Service
3. Third-party services and other applications
   - Web application tier components
   - Any other critical services including third-party services (JBoss)

MAINTAINING HIGH AVAILABILITY

In order to guarantee high availability of services in the grid, the following two main components are required:

1. A solution to monitor the health of the applications and to automatically restart the applications on available hosts when failures are detected. When using the SAS Grid Manager, this functionality is achieved with LSF EGO. EGO runs on all nodes participating in the high availability failover system. When detecting a problem with a critical service on one node, EGO will restart it on another node ensuring client requests can continue to be served.
2. A solution to enable clients to access the applications without the clients knowing the physical locations of the applications. A solution needs to be provided that resolves the Internet Protocol (IP) name to a different IP address, when the service is failed over to another node by LSF EGO. This makes the redirection transparent to the client.

REQUESTING A SERVICE BY AN APPLICATION

Transparent redirection to other IP addresses can basically be achieved in the following two ways:

1. Using the Corporate Domain Name System (DNS)
   This method dynamically reconfigures the DNS, associating the name to the new IP address after a failover of a service has happened. While this does not require additional hardware, the whole corporate network could be affected, as the network clients' DNS caching needs to be disabled for the changed relation to be picked up immediately. This might slow down network traffic and conflict with corporate policies.

2. Using a load-balancing switch
   This method uses an additional piece of hardware, leaving the corporate DNS untouched. The IP switch will be configured to sense the health of the affected service and carries a list of failover nodes. All network traffic is routed through this switch. When the service becomes unavailable the switch will automatically use an alternate IP address to resolve requests to the service now running on another node. This is entirely transparent to the clients. These switches are available from certain vendors including Cisco, BIG-IP, Barracuda, and others.

Using a Load-Balancing Switch

Figures 4 and 5, below, show how client request redirection would work for a client requesting a connection to the SAS Metadata Server. The same concepts apply for other SAS servers, as well.

Service Request Redirection in Normal State

The following schematic shows the steps taken when an application requests a service and a load balancing switch is used. In the example below it is assumed that a SAS client sends a metadata request to the SAS Metadata Server.

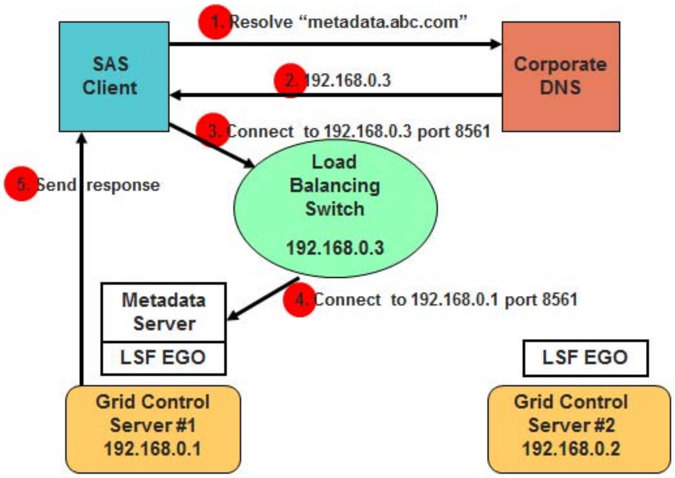

SAS Client
1. Resolve "metadata.abc.com"
Corporate DNS
2. 192.168.0.3
3. Connect to 192.168.0.3 port 8561
5. Send response
Load Balancing Switch
192.168.0.3
Metadata Server
LSF EGO
4. Connect to 192.168.0.1 port 8561
LSF EGO
Grid Control Server #1 192.168.0.1
Grid Control Server #2 192.168.0.2

Figure 4: SAS Metadata Server Request in Normal State

1. The SAS client asks the corporate DNS server for the IP address of the metadata server's host named *metadata.abc.com.*
2. The DNS server responds by sending the IP address, which is the address of the load balancing switch *192.168.0.3.*
3. With that information the client makes a connection to this address at port 8561, unaware of the fact this is the switch, not the actual metadata server machine.
4. The switch, which was configured appropriately, now redirects the request to the actual metadata server machine *192.168.0.1* at port *8561.* This is the grid control server #1.
5. After the connection to the metadata server was made, the requested metadata is sent to the SAS client.

Service Request Redirection in Failover State

The following diagram shows the steps that are taken when the SAS client sends a request to the SAS Metadata Server, after the server was failed over to another machine by LSF EGO, since its original host became unavailable. The change in machines for the SAS Metadata Server is unknown to the client.

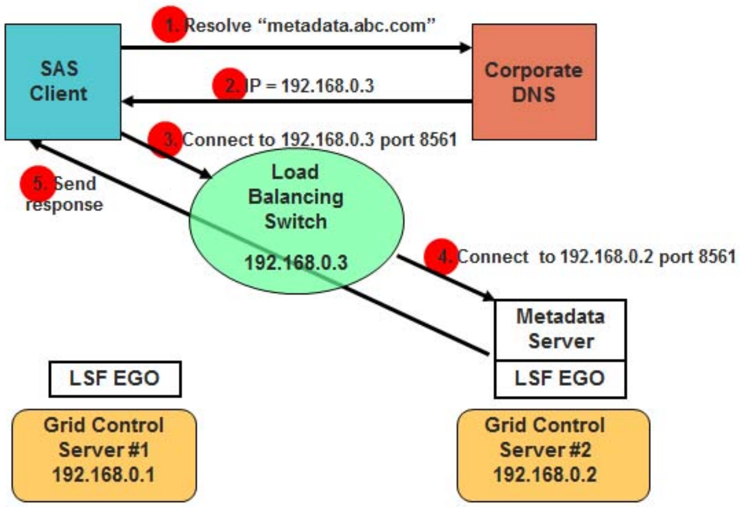

1. Resolve "metadata.abc.com"
SAS Client
Corporate DNS
2. IP = 192.168.0.3
3. Connect to 192.168.0.3 port 8561
Load Balancing Switch
192.168.0.3
5. Send response
4. Connect to 192.168.0.2 port 8561
Metadata Server
LSF EGO
LSF EGO
Grid Control Server #1 192.168.0.1
Grid Control Server #2 192.168.0.2

Figure 5: SAS Metadata Server Request in Failover State

1. The SAS client asks the corporate DNS server for the IP address of the metadata server's host named *metadata.abc.com.*
2. The DNS server responds by sending the IP address, which is the address of the load-balancing switch *192.168.0.3.*
3. With that information the client makes a connection to this address at port 8561, unaware of the fact this is the switch, not the actual metadata server machine.
4. The switch, knowing there is a problem with grid control server #1, now redirects the request to the failover metadata server machine *192.168.0.2* at port *8561.* This is the grid control server #2.
5. After the connection to the metadata server was made, the requested metadata is sent to the SAS client.

HIGH AVAILABILITY OF SAS JOBS

Another aspect of availability is whether SAS jobs can be guaranteed to execute, hence making their results available. Jobs might fail because of the following reasons:

- Unexpected reboot
- System failure
- Long running jobs did not complete due to exceeding quotas

Historically, these jobs have to be rerun again from the very beginning, even when they executed successfully up to a certain point at first. The following sections explain how these jobs can be enabled for re-execution starting at the step where they failed during the last submission, ensuring running these jobs in minimal time. This process can be automated using the LSF re-queue capability.

SAS Command Line Grid Submission Utility (SASGSUB)

Prior to SAS 9.2M2, when submitting a SAS job to the grid, it was required to have Base SAS® and SAS/CONNECT installed on the client node. Moreover the client session needed to remain active to collect the results of the job.

As a stand-alone utility SASGSUB, first shipping with SAS 9.2M2, can be used to submit SAS jobs to the grid without the need of having SAS installed on that client node. There is no need to remain connected to the job submitted (submit and forget). The utility can be used to view the status and output of submitted jobs as well as to kill jobs if they were submitted mistakenly. The shared file system used by all grid nodes must be accessible from the node where SASGSUB runs. The utility uses the SAS Metadata Server for centralized control of grid resources and can enable SAS jobs to be executed in checkpoint/restart mode.

SAS Checkpoint/Restart Capability

Used together, checkpoint mode and restart mode enable batch programs that terminate before completing to be resubmitted, resuming execution with the DATA or PROC step that was executing when the failure occurred. DATA and PROC steps that already completed will not be re-executed unless this is desired.

When checkpoint mode is enabled, SAS records information about DATA and PROC steps in a checkpoint library. When a batch program terminates prematurely, it can be resubmitted in restart mode to complete execution. In restart mode, global statements and macros are re-executed and SAS reads the data in the checkpoint library to determine which steps completed. Program execution resumes with the step that was executing when the failure occurred.

The checkpoint-restart data contains only information about the DATA and PROC steps that completed and the step that did not complete. It does not contain information about macro variables, macro definitions, SAS data sets, or any other information that might have been processed in the step that did not complete.

LSF Re-Queuing Capability

Automatic job rerun occurs when the execution host becomes unavailable while a job is running. When a job is rerun or restarted, it is returned to the queue from which it was dispatched with the same options as the original job. The priority of the job is set sufficiently high to ensure that the job gets dispatched before other jobs in the queue. The job uses the same job ID number. It is executed when a suitable host is available.

A similar feature to job rerun is job re-queue. Automatic job rerun occurs when a job finishes and has a specific exit code. This exit code can be used by a SAS program to indicate whether it should be rerun from scratch or from the last checkpoint.

The figure below shows an excerpt of an LSF queue definition. The queue's name is *SAS_RERUN* and it has been enabled to rerun SAS jobs that failed exiting with any return codes, except *0* and *1*, using the *RERUNNABLE = YES* and *REQUEUE_EXIT_VALUES = all ~0 ~1* options.

```
Begin Queue
QUEUE_NAME    = sas_rerun
PRIORITY      = 40
NICE          = 10
RERUNNABLE    = YES
REQUEUE_EXIT_VALUES = all ~0 ~1
DESCRIPTION   = Jobs submitted to this queue will be requeued automatically and also
rerunnable.
End Queue
```

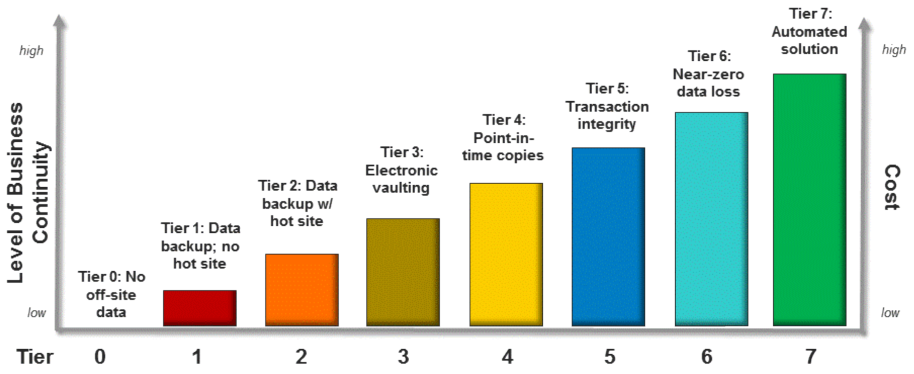
Figure 6: Enabling an LSF Queue for Re-Queuing Jobs

By combining job rerun and re-queue, you can automatically restart SAS jobs from last checkpoint on another suitable host when the original execution host failed. You can also automatically restart a failed job from the last checkpoint automatically on a suitable host should the exit code indicates that the job should be rerun.

CONSIDERATIONS FOR DISASTER RECOVERY

As one of the main component of business continuity planning, disaster recovery for your software systems introduces additional considerations that could impact how SAS is deployed for your organization. The discussion above on high availability focuses on active/active failover solutions. On the other hand, the disaster recovery plan is often an active/passive configuration, where the backup system is started up only when there is a catastrophic failure to the primary system. It is also common to find the backup system in a geographically separate location from the primary system.

In an active/passive configuration, backing up your primary system is mandatory for being able to recover from failure. Backups include the data and user-created content, as well as software deployment and configuration files. The level and frequency of backups can range from limited, manual backups to real-time mirroring.



Source: Wikipedia,
http://en.wikipedia.org/wiki/Seven_tiers_of_disaster_recovery

Figure 7: Disaster Recovery Tiers and Cost

- **Tier 0: No off-site data – Possibly no recovery**

  Businesses with a Tier 0 business continuity solution have no business continuity plan. There is no saved information, no documentation, no backup hardware, and no contingency plan. The time necessary to recover in this instance is unpredictable. In fact, it might not be possible to recover at all.

- **Tier 1: Data backup with no hot site**

  Businesses that use Tier 1 continuity solutions back up their data and send these backups to an off-site storage facility. The method of transporting these backups is often referred to as the "Pick-up Truck Access Method" (PTAM). Depending on how often backups are created and shipped, these organizations must be prepared to accept several days to weeks of data loss, but their backups are secure off-site. However, this tier lacks the systems on which to restore data.

- **Tier 2: Data backup with a hot site**

  Businesses using Tier 2 business continuity solutions make regular backups on tape. This is combined with an off-site facility and infrastructure (known as a hot site) in which to restore systems from those tapes in the event of a disaster. This solution will still result in the need to recreate data based on several hours or days, but the recovery time is more predictable.

- **Tier 3: Electronic vaulting**

  Tier 3 solutions build on the components of Tier 2. In addition, some mission critical data is electronically vaulted. This electronically vaulted data is typically more current than that which is shipped via PTAM. As a result there is less data recreation or loss after a disaster occurs.

- **Tier 4: Point-in-time copies**

  Tier 4 solutions are used by businesses that require both greater data currency and faster recovery than users of lower tiers. Rather than relying largely on shipping tape, as is common on the lower tiers, Tier 4 solutions begin to incorporate more disk based solutions. Several hours of data loss are still possible, but it is easier to make such point-in-time (PiT) copies with greater frequency than tape backups even when electronically vaulted.

- **Tier 5: Transaction integrity**

  Tier 5 solutions are used by businesses with a requirement for consistency of data between the production and recovery data centers. There is little to no data loss in such solutions, however, the presence of this functionality is entirely dependent on the application in use.

- **Tier 6: Zero or near-Zero data loss**

  Tier 6 business continuity solutions maintain the highest levels of data currency. They are used by businesses with little or no tolerance for data loss and who need to restore data to applications rapidly.

These solutions have no dependence on the applications or applications staffs to provide data consistency. Tier 6 solutions often require some form of disk mirroring. There are various synchronous and asynchronous solutions available from the mainframe storage vendors. Each solution is somewhat different, offering different capabilities, and providing different Recovery Point and Recovery Time objectives. Often some form of automated tape solution is also required. However, this can vary somewhat depending on the amount and type of data residing on tape.

- **Tier 7: Highly automated, business integrated solution**

  Tier 7 solutions include all the major components being used for a Tier 6 solution with the additional integration of automation. This allows a Tier 7 solution to ensure consistency of data above that which is granted by Tier 6 solutions. Also, recovery of the applications is automated, allowing for restoration of systems and applications much faster and more reliably than would be possible through manual business continuity procedures.

CONFIGURATION OF BACKUP SYSTEM FOR DISASTER RECOVERY

Ideally, the disaster recovery system should be configured as completely identical to the production system. This means having the same type of hardware, the same number of physical systems, and the same software configuration. However, it is not always realistic to duplicate the complete environment upfront, particularly if it is sitting idle until a disaster strikes the production system.

If you have a business continuity plan that includes a disaster recovery strategy, it pays to incorporate these requirements into the planning for the configuration of the initial SAS production environment. It is much easier to configure the initial environment upfront with DNS aliases or other references, than deciding to do that after the fact. If you decide to still use physical system references, you can manage configuration files to make it easier to start the backup system.

When planning your disaster recovery system, the following considerations are key:

- For the backup environment, you should try to mimic the production environment topology as much as possible. For example, if you have separate machines for the metadata server, compute servers, and mid-tiers, then have a metadata server, single compute server and single mid-tier on their own systems.
  - The best scenario would be to have dedicated hardware to use as the backup system. You would back up the entire production systems (such as using cloning software), which can then be restored onto the backup machines in their entirety. For Windows systems, for example, you must also backup the registry.
  - If you are not able to dedicate hardware as the backup system, you should install and configure SAS environment on the machines, but do not have the SAS servers actively running. You should have a documented process on how to start up the SAS servers if the need arises.

- If you choose to use actual machine names, rather than DNS aliases or network switches as described above, it is still possible to configure a disaster recovery backup system, but it will take more time to get the backup running.
  - Machine names will need to be modified manually. There are several locations where this must be done, and this should be documented as part of the disaster recovery plan for your SAS environment.
    1. Update the metadata server configuration files first.
    2. Update metadata references and application server configuration files.
    3. Update web application configuration files.
  - If the backup system is located in a different network domain, it might be necessary to take additional steps to make the new servers visible to users on the network.

- It is critical to back up SAS content on a regular basis and stored in a place that is away from the primary system location and accessible to the backup system. Any content that is dynamic in nature (log files, data, metadata, analytical models, reports, and other user-created content) should be stored separately from more static SAS files, such as configuration files.
  - Mirrored directory structures would make restoring a new system much easier, with minimal loss of work in progress.
  - Essential backup processes need to include metadata repositories, the SAS Content Server, and the database supporting SAS Shared Services. The timing of the backups should coincide with the backup of physical data sources to ensure that all components stay in synch.

  - Backups of static files, such as configuration files can be taken less frequently with less risk.

CONCLUSION

High availability refers to the ability to consistently keep your information systems operational and accessible to the business. There are many solutions for keeping your systems highly available and minimize downtime. In general, all solutions encompass the following concepts:

- Regular backups of the primary environment
- Monitoring the server processes and resources
- Recovering from failure when detected
- Reconnecting to client applications

There are a number of options for configuring your SAS business analytics environment to be highly available. Some techniques are embedded within the SAS technologies themselves, or you can leverage SAS Grid Manager or third-party offerings to provide more robust capabilities.

REFERENCES

- Hatcher, Diane. (in press). “SAS Architecture for Business Analytics – Configuring for High Availability.” White paper. Cary, NC: SAS Institute Inc.
- Holzworth, Steve, and Clarke Thacher. 2006. “Achieving High Availability for the SAS®9 Metadata Server.” *Proceedings of the SAS Global 2006 Conference.* Cary, NC: SAS Institute Inc. Available at www2.sas.com/proceedings/sugi31/002-31.pdf.
- Hunt, Arthur, et al. 2008. “Backup and Disaster Recovery: When Disaster Strikes, What Will You Do? What Will You Do?” *Proceedings of the SAS Global 2008 Conference.* Cary, NC: SAS Institute Inc. Available at support.sas.com/resources/papers/sgf2008/recovery.pdf.
- Wong, Daniel. 2009. “Achieving High Availability in a SAS® Grid Environment.” *Proceedings of the SAS Global 2009 Conference.*Markham, Canada: Platform Computing, Inc. Available at support.sas.com/resources/papers/proceedings09/001-2009.pdf.
- “Seven tiers of disaster recovery.” Wikipedia. Available at http://en.wikipedia.org/wiki/Seven_tiers_of_disaster_recovery. Accessed on February 2, 2011.
- “High availability.” Wikipedia. Available at http://en.wikipedia.org/wiki/High_availability. Accessed on February 2, 2011.

ACKNOWLEDGMENTS

Special thanks to Cheryl Doninger, head of SAS Grid Manager Research and development, and her team for providing technical and other information used in this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Diane Hatcher
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
E-mail: Diane.Hatcher@sas.com

Jochen Kirsten
SAS Institute GMBH
In der Neckarhelle 162
Heidelberg 69118 Germany
E-mail: Jochen.Kirsten@sas.com