CREATING HIGHLY AVAILABLE SAS® ENTERPRISE INTELLIGENCE PLATFORM ENVIRONMENTS

White Paper
May 2008

Table of Contents

Chapter 1

Why High Availability Matters for the SAS® Enterprise Intelligence Platform

From a business point of view, high availability for the SAS® Enterprise Intelligence Platform means that application services and data are nearly always accessible. Because today's application services are increasingly interlinked, failure of any single component in an application environment can have a cascading effect throughout the enterprise that can pose significant business risk and financial loss.

In the SAS Enterprise Intelligence Platform, there are interdependencies between the various SAS applications servers and services. When even one service is unavailable, it can dramatically affect all of the other services. If this happens, people are forced to make business decisions without supporting data, which can lead to disastrous results.

In order to appreciate the importance of high availability in the SAS Enterprise Intelligence Platform, it is helpful to first understand the underlying architecture, how the various SAS servers interact with each other, and how potential failures impact the operation of the entire platform. The primary components of the SAS Enterprise Intelligence Platform include the SAS® Metadata Server, SAS® OLAP Server, object spawner, workspace server, and SAS Stored Process Server. Optionally, components like SAS/CONNECT® and SAS/SHARE® may be incorporated to provide additional functionality. In addition, SAS client applications and the data sources that SAS processes should also be included in any high availability strategy.

SAS® Metadata Server

The SAS Metadata Server is the single most important component of the SAS Enterprise Intelligence Platform. It handles authentication (confirming that users are who they say they are) and authorization (determining what users can view and do within the system). It also contains information on all of the other components that make up a particular instance of the SAS Enterprise Intelligence Platform. For example, the SAS Metadata Server contains information that details:

- The existence of other SAS components such as SAS OLAP Server, workspace servers, etc.
- The actual commands that are used to start some of these servers
- The location of configuration files

Given its critical role within the SAS Enterprise Intelligence Platform, maximizing the availability of the SAS Metadata Server should be a key objective of administrators.

Object spawner, workspace server, and SAS® Stored Process Server

These three components are the workhorses of the SAS Enterprise Intelligence Platform. These components are responsible for the bulk of the data processing and analysis required to respond to end-user requests and to execute SAS code (programming logic written in the SAS language). The object spawner acts as a kind of traffic cop or dispatcher — as requests come in from clients, it connects the client to a workspace server or SAS Stored Process Server process that can process the request.

In the standard configuration, a number of SAS Stored Process Server processes are up and running and waiting for work. When a user asks for something that requires a stored process to execute, the client interacts with the object spawner and is told how to connect to of one of these SAS Stored Process Server processes. The client then interacts with this process (passing parameters in and getting output back). At the end of the interaction, the process remains running and waits for another request to do work. The workspace server can also be configured to run in this *pooled* configuration —however, it does not have to be.

The workspace server's standard configuration is unpooled. In this configuration, a process is only started (or spawned) when there is work to be performed. For example, when a user asks for a specific report, the client interacts with the object spawner and requests that a workspace server session/process be started. The object spawner starts the session and passes information back to the client so that it can interact directly with the newly spawned workspace server session. After the workspace server session has finished its work, it is terminated.

The metadata stored in the SAS Metadata Server provides the object spawner with information about the workspace server and SAS Stored Process Server, such as: how many workspace and/or SAS Stored Process Server sessions can be started, which users have access to which server, and the commands needed to launch these servers.

Failure of the object spawner would make both stored process and workspace server processes unavailable to the various clients.

SAS® OLAP Server

The SAS OLAP Server is the component that handles requests for access to cube data. It takes requests from the reporting clients, finds the appropriate data in the specified cube, and returns the data to the clients. If it were to fail, the reporting clients would not be able to access any cube-based data.

SAS/CONNECT®

This component allows a SAS session on one system to perform processing on a separate system. This can include executing SAS code or accessing data.

SAS/SHARE®

SAS/SHARE allows multiple users to edit the same SAS data source. It can also be used to provide access to data sources on additional SAS servers outside of the core SAS Enterprise Intelligence Platform deployment.

Data sources

The SAS Enterprise Intelligence Platform exists to allow users to report on, explore, and analyze data. Without access to the data, there is nothing to work against. This data can come from many formats: relational tables, online analytical processing (OLAP) cubes, SAS structured data sets, text files, spreadsheets, etc. Relational tables refer to a basic relational database management system (RDBMS) data structure. OLAP cubes are a specialized data source that are designed to provide fast access to data at many different levels of aggregation. For instance, a cube built on US Census information might contain data at the national, state, county, and census tract level. These four different levels of information make up a geographical *dimension*. Another dimension might be home ownership (homeowner, renter) or time. By summarizing the low level data along and across these various dimensions ahead of time to populate a cube, reporting client users can explore the data very easily and quickly. Both SAS cubes and SAS data sets (the proprietary SAS relational tables) are file structures on disk as opposed to running processes. SAS can also access relational tables stored in a RDBMS such as Oracle, IBM DB2, etc.

SAS® clients

The end-users who are trying to run reports or analyze data interact with the SAS Enterprise Intelligence Platform through various SAS reporting clients. There are two types of clients: thick clients (software installed on an end-user's PC) and thin clients (software accessed through a Web browser). SAS Enterprise Guide™ software is the primary thick client for business intelligence (BI). It is a sophisticated interface that allows programmers to fully exploit the power and breadth of SAS 4GL language programming. For non-programmers, an extensive set of dialogs and wizards and a drag-and-drop interface allow them to analyze their data without needing to learn a programming language. More sophisticated users might want to leverage their knowledge of the SAS language to perform more advanced analytics and data processing. The analytic interface from SAS Enterprise Guide software is a workspace server, which is started on behalf of the client through the object spawner. The workspace server is a persistent server that is active for the lifetime of the SAS Enterprise Guide software session.

The Web-based clients for BI include SAS® Web Report Studio, SAS® Web OLAP Viewer for Java, and the SAS® Information Delivery Portal. All of these are Web applications that use server-side Java™ technology and require a Web application server such as Tomcat, WebLogic, or WebSphere. These application servers handle all of the Java platform processing needed to render the interfaces and interact with the other components of the SAS Enterprise Intelligence Platform.

SAS Web Report Studio is the primary reporting Web client. It is designed to allow the experienced and novice SAS user to access the analytic capabilities of the SAS Enterprise Business Intelligence Platform architecture. SAS Web Report Studio can work with both relational and OLAP data sources. Requests for relational data and any necessary processing are handled by interacting with a workspace server session. For SAS Web Report Studio, the workflow is:

- The user requests a report
- The Web application server interacts with the object spawner to get access to a workspace server session
- The object spawner then passes the user's request on to the workspace server session
- The workspace server performs the processing and passes the results back to the Web application server
- The Web application server renders the results appropriately and sends it back to the user's browser

SAS Web Report Studio also allows users to execute stored processes, which are executed on the SAS Stored Process Server in a similar process flow.

SAS Web OLAP Viewer for Java is a specialized client designed to allow users to explore OLAP data. It interacts with SAS Metadata Server and SAS OLAP Server.

The SAS Information Delivery Portal allows users to create custom portals made up of the BI content they need. The content that can be surfaced includes links to reports created in SAS Web Report Studio, links to cube explorations created in SAS Web OLAP Viewer for Java, stored processes, Web sites, text, etc. Depending on the type of content being surfaced, it might need to interact with workspace server, SAS Stored Process Server or SAS OLAP Server.

Providing high availability protection to the Web-based components/clients is generally implemented using standard Web server protection strategies such as supporting multiple Web server nodes. SAS Web server applications cooperate with normal IT high availability processes enabled for the Web application server.

Figure 1 shows the components of the SAS Enterprise Intelligence Platform and how they interact with each other.

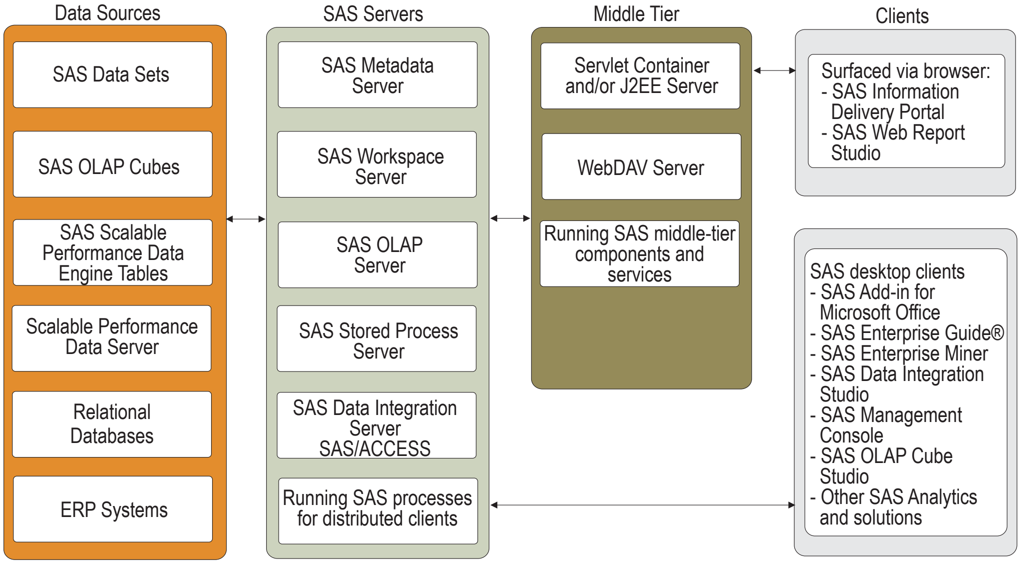
Data Sources
SAS Data Sets
SAS OLAP Cubes
SAS Scalable Performance Data Engine Tables
Scalable Performance Data Server
Relational Databases
ERP Systems
SAS Servers
SAS Metadata Server
SAS Workspace Server
SAS OLAP Server
SAS Stored Process Server
SAS Data Integration Server SAS/ACCESS
Running SAS processes for distributed clients
Middle Tier
Servlet Container and/or J2EE Server
WebDAV Server
Running SAS middle-tier components and services
Clients
Surfaced via browser:
- SAS Information Delivery Portal
- SAS Web Report Studio
SAS desktop clients
- SAS Add-in for Microsoft Office
- SAS Enterprise Guide®
- SAS Enterprise Miner
- SAS Data Integration Studio
- SAS Management Console
- SAS OLAP Cube Studio
- Other SAS Analytics and solutions

Figure 1. SAS Enterprise Intelligence Platform

Chapter 2

Creating a Highly Available SAS® Enterprise Intelligence Platform Environment

Sun's Datacenter of the Future for SAS Enterprise Intelligence Platform employs a layered approach that can help keep SAS application services operating continuously, regardless of the cause of failure. Each layer of availability provides the tools to shield SAS application services from failures in hardware components, software, hardware and software partitions (Dynamic System Domains and Solaris™ Containers), nodes, buildings, and geographic sites.

- Layer 1 — Reliable systems for SAS components. This layer configures systems with redundant components.
- Layer 2 — Access to data and SAS components. All systems should be configured with multiple network cards, ports, and paths. Systems that require direct access to SAS data, such as the system where the SAS OLAP Server runs, should have multiple host bus adapter (HBA) cards, ports, and multiple paths to storage.
- Layer 3— Self healing. This layer provides the ability to monitor, predict failures, and fence-out failed hardware components without rebooting the system or disrupting SAS processes or components.
- Layer 4— Automatically restart SAS components. This layer includes the ability to catalog processes on which a SAS process or components relies, monitor the services, and automatically restart parts of the service or the entire service in dependency order.
- Layer 5— Protecting against unrecoverable failures. The layer provides the ability to fail over a single SAS service or every service running on a physical system to another node in a cluster in the event of an unrecoverable error on the primary system.
- Layer 6 — Campus, metro, and geographic failover. This layer provides the ability to fail over to another room, campus, city, or country in the event of a catastrophic disaster.

All of the layers may be applied to all of the systems and SAS components in order to create a highly available SAS Enterprise Intelligence Platform environment. The only exceptions are the Web clients and Web application servers, which due to their ability to be implemented across multiple systems, do not necessarily need to function within a cluster for high availability, unless layer 6 is implemented.

Layer 1 — Reliable systems for SAS® components

The first layer in creating highly available SAS Enterprise Intelligence Platform environments implements reliable systems through redundancy where economically possible in the physical systems that support SAS. This is the easiest way to increase availability of the underlying hardware that SAS components run on.

To increase reliability, it is important to configure systems with redundant components. Sun implicitly understands the need for 24x7 access to SAS Enterprise Intelligence Platform applications and data, which is why Sun™ servers and storage feature redundant and often hot-swappable components such as processors, power supplies, fans, etc. Where possible, systems should be configured with multiple CPUs, Ethernet ports, HBA ports, power supplies, and fans.

Some SAS applications can be deployed on multiple systems, such as applications in the middle tier, or applications that can be deployed on a grid of smaller systems with fewer redundant components, such as Sun's x64 servers or servers with CoolThreads™ technology. One of the benefits of grid computing and the middle tier is inherent system redundancy through multiple nodes and storage. As the number of servers increases, the risk and impact of a failed node decreases because there are more servers to take over the load of the failed node.

Figure 2 shows a typical configuration with redundant Web servers, a large system to support most of the server tier components, and a separate system for the SAS Metadata Server.

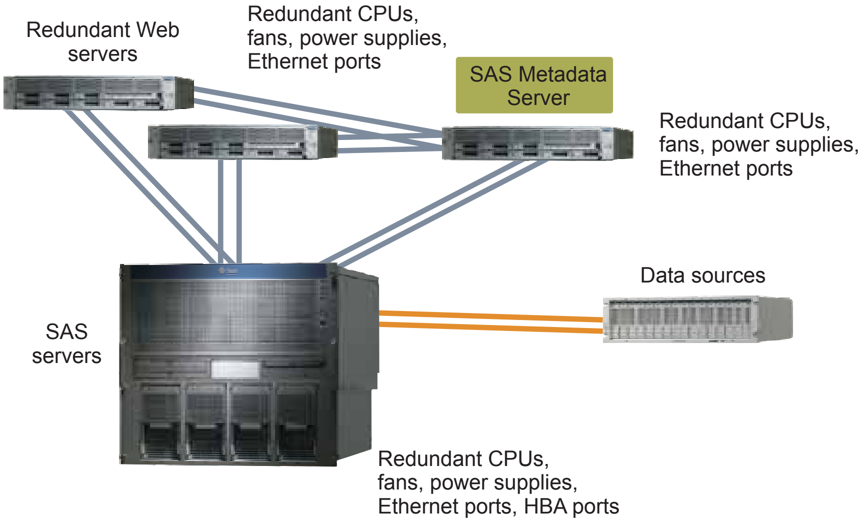
Redundant Web servers
Redundant CPUs, fans, power supplies, Ethernet ports
SAS Metadata Server
Redundant CPUs, fans, power supplies, Ethernet ports
Data sources
SAS servers
Redundant CPUs, fans, power supplies, Ethernet ports, HBA ports

Figure 2. Layer 1 — Typical configuration with redundant components

For large environments, the Sun SPARC® Enterprise M-Series systems provide the highest levels of reliability in the Sun system product line with redundant and hot swappable disk drives, power supply units, fan units, and external I/O expansion units (optional). Sun SPARC Enterprise M8000 and M9000 servers also offer redundant and hot swappable CPU memory board units, service processors, crossbar units, and optional dual power feeds, as well as degradable crossbar switches and bus routes. And, state-of-the-art fault isolation identifies faults at the chip level rather than the component level, enabling the systems to take only the errant ASIC off-line, improving system resilience.

Layer 2 — Access to data and SAS® components

I/O and networking throughput are important factors in the performance of SAS Enterprise Intelligence Platform environments. Implementing higher levels of availability also requires redundant host connectivity to storage and networking devices. The Solaris Fibre Channel and Storage Multipathing software, incorporated into the Solaris 10 Operating System (Solaris OS), manages failed storage paths while maintaining host I/O connectivity through available secondary paths. IP Multipathing (IPMP) enables IP fail-over and IP link aggregation, helping to manage network workloads and failures.

Multiple paths to network ports

As described in Chapter One, the SAS distributed architecture, including the SAS Metadata Server, relies on the networking infrastructure for communication in order to initiate processes, gather data, and return results. Therefore, it is imperative to ensure that there are redundant network ports and paths. In addition, systems need to be configured with adequate bandwidth to support the user base. Real-time users often perceive performance degradations as service outages. Therefore, redundant and resilient networking paths are an important factor in availability.

IPMP provides high availability of network connections by detecting a network adapter failure and automatically switching (failing over) its network access to an alternate network adapter. It can also detect repair or replacement of a previously failed network and automatically switch back (fail back) network access from an alternate network adapter. On Sun systems that support hot-swappable network cards, the card can be replaced without interrupting system operations and then IPMP can automatically start using the port or path again.

To increase bandwidth, IPMP supports outbound load spreading where outbound network packets are spread across multiple network adapters, without affecting the ordering of packets, to achieve higher throughput.

A network adapter with multiple interfaces could become a single point of failure if the entire adapter fails. For maximum availability, configure systems with multiple adapter cards where possible.

Multiple paths to storage

Implementing higher levels of reliability and availability for SAS data requires redundant host connectivity or HBA ports to the storage devices that contain SAS relational tables and OLAP cubes. The Solaris FC and Storage Multipathing software, which is integrated in the Solaris 10 OS, manages the failure of multiple storage paths while maintaining host I/O connectivity through available secondary paths.

The Solaris FC and Storage Multipathing software dynamically manages the paths to any storage devices the software supports. Adding or removing paths to a device is performed automatically when a path is brought online or removed from a service. This allows systems configured with the Solaris FC and Storage Multipathing software to begin with a single path to a device and add more host controllers, increasing bandwidth and availability, without changing device names or modifying applications. For Sun storage, there are no configuration files to manage or databases to keep current. To provide high availability in SAS Enterprise Intelligence Platform environments, there should be a minimum of two HBA connections from the systems that require access to data to the storage devices.

In addition to providing simple failover support, the Solaris FC and Storage Multipathing software can use any active paths to a storage device to send and receive I/O. With I/O routed through multiple host connections, bandwidth to SAS data can be increased by adding host controllers. The Solaris FC and Storage Multipathing software uses a round-robin load-balancing algorithm, which routes individual I/O requests to active host controllers in a series, one after the other.

Finally, the Solaris FC and Storage Multipathing software further increases availability by automatically recognizing devices and any modifications to device configurations. When new storage devices are added to the system they are available to the system without requiring a reboot or a manual change to information in configuration files. This increases the availability of the system while adding or upgrading storage.

As with network interface cards, an HBA with multiple ports can be a single point of failure if the entire card fails. To provide higher availability, configure systems with multiple HBA cards where possible. Figure 3 illustrates layer 2.

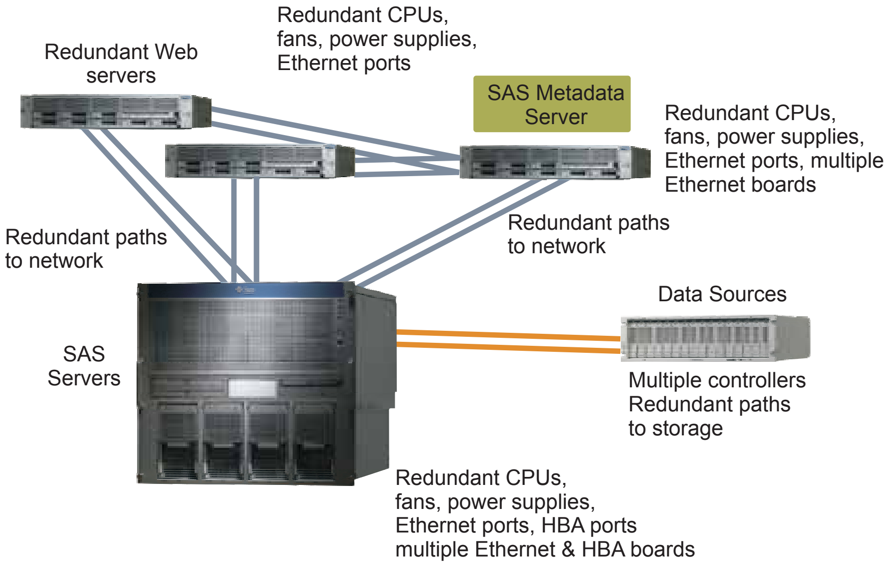

Figure 3. Layer 2 — access to data and SAS components

Layer 3 — Self healing

Configuring systems with multiple components provides a certain level of availability, but in typical environments, if a hardware component fails, the system often crashes and must reboot. The failed component either causes the system to stay down, or the system can reboot around the failed component. In some cases, it can take hours or days to diagnose the problem and replace the faulty component. Once the system is back up, all of the applications and services must be restarted, usually with manual intervention. To overcome the downtime caused by unexpected hardware failures, a system needs the ability to self-heal around failed components, without rebooting. The Solaris 10 OS includes an architecture for building and deploying systems and services designed for Predictive Self Healing. Self-healing technology enables Sun systems and services to maximize availability for the SAS Enterprise Intelligence Platform environment in the face of software and hardware faults.

Solaris Predictive Self Healing enables the system to heal itself. An innovative capability, it automatically diagnoses, isolates, and recovers from many hardware and application faults in just a few seconds, instead of the many days it can take IT staff to diagnose and correct a problem. As a result, SAS Enterprise Intelligence Platform services and essential system services can continue uninterrupted in the event of software failures, and/or hardware component failures, and even software misconfiguration problems.

Solaris Predictive Self Healing monitors CPU, memory, and I/O bus components in the system. The Solaris Fault Manager within Solaris Predictive Self Healing is responsible for replacing traditional error messages intended for IT staff with binary telemetry events that are then dispatched to the appropriate diagnosis engines. The diagnosis engine is responsible for identifying the underlying hardware faults or software defects that are producing the error symptoms. After processing sufficient telemetry to reach a conclusion, a diagnosis engine produces another event called a fault event. The fault event is then broadcast to all agents that are interested in the specific fault event. An agent is a software component that initiates self-healing activities such as administrator messaging, isolation or deactivation of faulty components, and guided repair.

The fault manager daemon starts at boot time and loads all of the diagnosis engines and agents available on the system. The Solaris Fault Manager also provides interfaces for IT operators and service personnel to observe fault management activity.

When appropriate, a self-healing system might direct an IT operator to a knowledge article to learn more about a problem impact or repair procedure. The knowledge article corresponding to a self-healing message can be accessed by taking the Sun Message Identifier (SUNW-MSG-ID) and appending it to the link *http://www.sun.com/msg/* in a Web browser. However, in the meantime, other agents participating in the self-healing system might have already off-lined an affected component and taken other action to keep the system and services available.

Layer 4 — automatic restart of SAS® components

The next layer in creating a highly available SAS Enterprise Intelligence Platform environment helps ensure that the individual SAS processes discussed above, as well as system services, remain running with minimal assistance from IT operators. The Solaris Service Management Facility can provide this functionality by restarting SAS services in dependency order when necessary.

The Service Management Facility provides an infrastructure that augments the traditional UNIX® start-up scripts and configuration files that IT operators must modify to start system and SAS application services. Service Management Facility makes it easier to manage system and application services by delivering new and improved ways to control services. The fundamental unit of administration in the Service Management Facility framework is the service instance. An instance is a specific configuration of a service. Multiple instances of the same version can run on a single Solaris system. For example, a Web server is a service, and a specific Web server daemon that is configured to listen on port 80 is an instance.

Service Manager Facility provides a mechanism for defining the various components of an application service and starts application services in dependency order. It also enables failed services to restart automatically in dependency order, regardless of whether they are accidentally terminated by an IT operator, if they are aborted as the result of a software programming error, or if they are interrupted by an underlying hardware problem.

Each software service has an advertised state. Should a failure occur, the system automatically diagnoses the failure and locates/pinpoints the source of the failure. Failing services are automatically restarted whenever possible, reducing the need for manual intervention. Hardware faults that affect software services, as well as software failures, cause the affected services to be restarted automatically, along with any services that declared a need to be restarted when the directly impacted services are restarted. Should manual intervention be required, IT operators can quickly identify the root cause of the service's failure and significantly reduce the time-to-repair and recover the service.

For example, the object spawner depends on the SAS Metadata Server to provide information about workspace server and SAS Stored Process Server processes. If the SAS Metadata Server is not running, then it does not make sense to restart the object spawner until the SAS Metadata Server is successfully restarted.

SAS, in conjunction with the Sun℠ Competency Center for SAS Solutions, has created example Service Management Facility scripts to monitor the SAS Metadata Server, object spawner, SAS OLAP Server, SAS/SHARE, and SAS/CONNECT. These scripts are designed to automatically restart these SAS processes on the same node if they should fail. SAS client applications will be able to reconnect to a restarted SAS server application. In some instances, the failure and restart of a server might be not be observed by a user. In other cases, a client application might need to reconnect to the failed server to resume processing.

Service Management Facility provides the following functions:

- Automatically restarts failed services in dependency order, whether they fail as the result of IT operator error, software bug, or an uncorrectable hardware error.
- Makes it easy to backup, restore, and undo changes to services by taking automatic snapshots of service configurations.
- Simplifies debugging and diagnosis of problems by providing an explanation for why a service may have failed.
- Allows for services to be enabled and disabled automatically.
- Enhances the ability of IT operators to securely delegate tasks to non-root users, including the ability to modify properties and enable, disable, or restart services on the system.

- Boots faster on large systems by starting services in parallel according to the dependencies of the services. Similarly, shutdown processing time is reduced due to disabling services in parallel based on service dependencies.

Layer 5 — protecting against unrecoverable failures

In addition to the previously described layers and fail-safe mechanisms catastrophic problems might cause situations where a service cannot be restarted, or where a Solaris Container, domain (hardware partition), or the entire system fails. In this case, in order to provide access to the SAS Enterprise Intelligence Platform, it is necessary to fail the services on the container, domain, or system over to another system. Solaris Cluster software can help protect against rare but still possible system failures by providing a cluster of nodes where services can fail over to other nodes.

Solaris Cluster Overview

Solaris Cluster is integrated with the Solaris 10 Operating System kernel to deliver support for unique features such as Solaris Containers, Solaris Predictive Self Healing, and Solaris ZFS. This kernel integration results in more reliable and faster failover of mission-critical applications. In addition, a new dual-partition software update feature simplifies the upgrade process — any component of the software stack, along with Solaris Cluster, can be upgraded in one step, including the Solaris OS, file systems, and volume managers. This reduces the ever present risk of human error during complex cluster upgrades.

Solaris Cluster can also help minimize planned downtime in SAS Enterprise Intelligence Platform environments. For example, if the system that runs the primary instance of the SAS Metadata Servers needs to be taken off-line for maintenance, the SAS Metadata Server can be halted and then restarted on the failover node. When the maintenance is complete, the SAS Metadata Server is halted on the failover node and restarted on the primary node, with minimum disruption to users and the other SAS components that access the SAS Metadata Server.

Solaris Cluster integrates tightly with Solaris Predictive Self Healing and allows Solaris Service Management Facility controlled applications to be automatically integrated within the Solaris Cluster (if Solaris Service Management Facility is in use). Local service-level management continues to be operated by Solaris Service Management Facility, while whole resource level cluster-wide failure handling operations (node and storage) are carried out by Solaris Cluster software. Within a cluster, failures are handled in a cascading fashion, first by trying to restart on the same node, then failing over within the nodes of the cluster.

For example, if the SAS Metadata Server fails, Solaris Service Management Facility will try to restart the service as stipulated by the IT operator in the controlling script. If the service cannot be restarted, Solaris Service Management Facility passes responsibility to Solaris Cluster, which can then try to restart the service in another container on the same system. If that fails, Solaris Cluster then fails the service over to a secondary node and restarts the SAS Metadata Server.

Note that systems in a grid or in the middle tier are afforded availability through redundancy in that they are part of a group of replicated servers that can assume the load of a failed node. However, they are still vulnerable to more catastrophic failures that could disable an entire datacenter, as described below in Campus and Metro Failover.

Node failover

If an application service cannot be restarted on its host containers, domain, or node, Solaris Cluster software fails the service over to its target node. This target could be a Solaris Container on another server, which could be automatically configured to provide adequate resources for the failed-over application service. The ability to failover to containers on other nodes helps reduce the cost of implementing clustering for high availability.

For example, if the SAS Metadata Server is running on its own small system, such as a Sun SPARC Enterprise T5220 server, one way to implement HA is to have a duplicate system standing by as the failover server. However this is a expensive use of resources. A more cost-effective and ecological solution is to use the duplicate system to provide Web client and Web application services. With Solaris Containers, the Web client and Web application server can run in their own containers and another container can be configured as the SAS Metadata Server secondary node. Solaris Resource Manager assigns system resources to each container. If the SAS Metadata Server should need to failover, the Solaris Resource Manager can be configured to automatically provide the SAS Metadata Server with the appropriate resources. Although the performance of the Web tier might be slightly degraded, the SAS Metadata Server is available to allow the SAS Enterprise Intelligence Platform to function properly, at a reduced or degraded performance level.

Figure 4 illustrates a typical configuration that is now protected against unrecoverable failures.

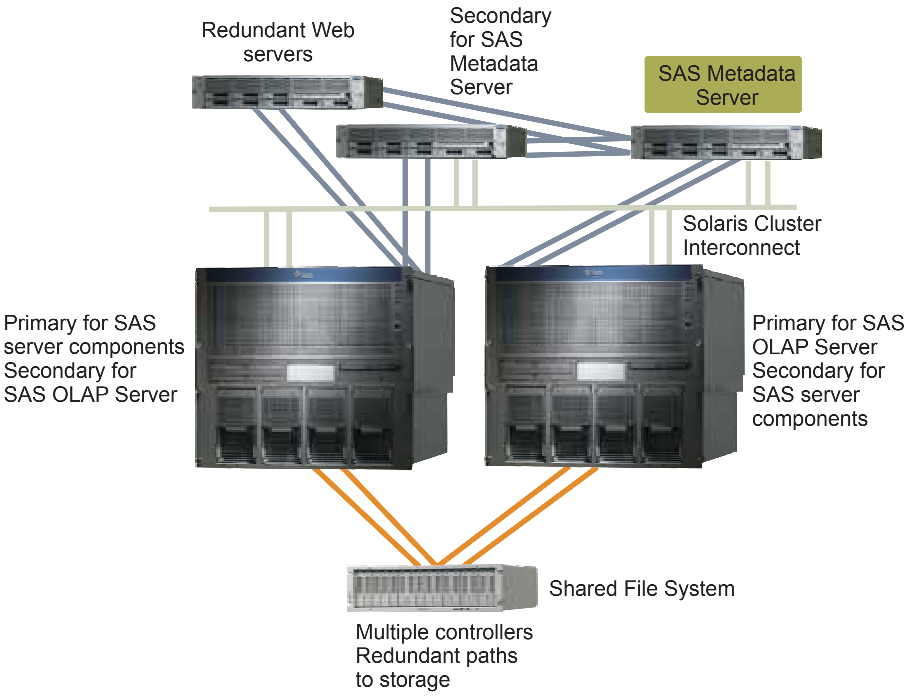
Redundant Web servers
Secondary for SAS Metadata Server
SAS Metadata Server
Solaris Cluster Interconnect
Primary for SAS server components Secondary for SAS OLAP Server
Primary for SAS OLAP Server Secondary for SAS server components
Shared File System
Multiple controllers Redundant paths to storage

Figure 4. Layer 5 — protecting against unrecoverable failures with clustering

Solaris Cluster Concepts

Sun and SAS have created and tested scripts to help organizations configure Solaris Cluster software for SAS Enterprise Intelligence Platform environments. The procedures to implement the scripts are detailed in the companion guide to this paper, *SAS Enterprise Intelligence Platform High Availability Implementation Guide*. In order to understand the procedures, it is important to know the key concepts of the Solaris Cluster software.

The Solaris Cluster software makes all components on the *path* between users and data highly available, including network interfaces, the applications themselves, the file system, and the multihost devices.

A cluster is a collection of two or more loosely connected systems, or nodes, that work together as a single, continuously available system to provide applications, system resources, and data to users. Each node in a cluster is a fully functional standalone system that runs its own processes. No single failure — hardware, software, or network — can cause a cluster to fail.

Cluster nodes are generally attached to one or more multihost devices. Disks that can be connected to more than one node at a time are multihost devices. In the Solaris Cluster environment, multihost storage makes disks highly available. For example, it is a good idea to ensure that the storage devices that store SAS relational tables and

OLAP cubes are multihost devices. That way, if the OLAP Server fails over to another node, it can still access the OLAP cube data.

Solaris Cluster software requires multihost storage for two-node clusters to establish a quorum. Greater than two-node clusters do not require quorum devices.

All nodes in the cluster are grouped under a common name (the cluster name), which is used for accessing and managing the cluster. Public network adapters attach nodes to the public networks, providing client access to the cluster.

Cluster members communicate with the other nodes in the cluster through one or more physically independent networks. This set of physically independent networks is referred to as the cluster interconnect.

Every node in the cluster is aware when another node joins or leaves the cluster. Additionally, every node in the cluster is aware of the resources that are running locally as well as the resources that are running on the other cluster nodes.

Nodes in the same cluster should have similar processing, memory, and I/O capability to enable failover to occur without significant degradation in performance. Because of the possibility of failover, every node must have enough excess capacity to support the workload of all nodes for which they are a backup or secondary.

Global devices

The Solaris Cluster software uses global devices to provide cluster-wide, highly available access to any multi-hosted device in a cluster, from any node, without regard to where the device is physically attached. In general, if a node fails while providing access to a global device, the Solaris Cluster software automatically discovers another path to the device. The Solaris Cluster software then redirects the access to that path. The cluster automatically assigns unique IDs to each disk in the cluster. This assignment enables consistent access to each device from any node in the cluster.

Cluster file systems

A cluster file system is mounted on all cluster members. With a cluster file system, all file access locations are transparent. A process can open a file that is located anywhere in the system. Processes on all nodes can use the same path name to locate a file. Applications running on multiple cluster nodes can access files, for both read and write, in the same way they would access the files from the same server. File locks are recovered immediately from nodes that leave the cluster and from applications that fail while holding locks. Thus, continuous access to data is ensured, even when failures occur. Applications are not affected by failures if a path to disks is still operational.

Using the cluster file system for SAS binaries has advantages and disadvantages. Installing binaries on a cluster file system allows a single install to work on all cluster nodes. The disadvantages become evident during upgrades or patches, where the changes must be applied directly to the single copy. This requires downtime of the SAS component while the changes are applied. Further downtime can occur if a change creates problems or needs to be backed out.

The other option, using different storage locations for the binaries for each node, minimizes downtime associated with this maintenance strategy. This option allows IT operators to upgrade the one node that is off-line, switch the service to the upgraded node, and then upgrade the remaining node. However, this approach has the disadvantage that while upgrading, only one node is functional and failover to a second node is not available. For sites that cannot tolerate any downtime, a cluster consisting of three nodes provides the highest level of protection. While any one node is upgraded, another node is always available to serve as the secondary node if failover becomes necessary. Although this approach of using separate locations for the binaries for each node does introduce a certain amount of complexity, the redundant binaries do offer a measure of protection against disk failure. In the end, either approach can be acceptable and organizations need to decide which one makes the most sense for the environment.

Quorum and quorum devices

Because cluster nodes share data and resources, a cluster must never split into separate partitions that are active at the same time because multiple active partitions might cause data corruption. The Cluster Membership Monitor (CMM) and quorum algorithm guarantee that at most one instance of the same cluster is operational at any time, even if the cluster interconnect is partitioned. Solaris Cluster assigns each node one vote and mandates a majority of votes for an operational cluster. A partition with the majority of votes gains quorum and is allowed to operate. This majority vote mechanism prevents split brain and amnesia when more than two nodes are configured in a cluster. However, counting node votes alone is not sufficient when more than two nodes are configured in a cluster. In a two-node cluster, a majority is two. If such a two-node cluster becomes partitioned, an external vote is needed for either partition to gain quorum. This external vote is provided by a quorum device. A quorum device can be a small partition on a shared storage device. For more information on quorums and other Solaris Cluster topics, see the *Sun Cluster Concepts Guide for Solaris OS* at *http://docs.sun.com/app/docs/doc/819-2969?l=en&a=load*

Data services

A data service is an application, such as the SAS Metadata Server, that has been configured to run on a cluster rather than on a single server. A data service consists of an application, Solaris Cluster configuration files, and Solaris Cluster management methods that start, stop, monitor, and take corrective measures. Each Solaris Cluster data service supplies a fault monitor that periodically probes the data service to determine its health. A fault monitor verifies that the application daemon or daemons are running and that clients are being served. Based on the information that these probes return, predefined actions such as restarting daemons or causing a failover can be initiated. The Solaris Cluster, by default, attempts to restart the application on the original node a number of times. The actual number of restart attempts is configurable. If Solaris Cluster cannot successfully restart the application on the original node, it fails over the application to another node. If Solaris Cluster needs to failover the application repeatedly within a short period of time — the actual time is configurable — Solaris Cluster puts the application in failed state and does not attempt to restart it. The behavior of the cluster can be tuned through a number of parameters.

Resource types and groups

A resource type is a bundle of all the information needed by Solaris Cluster to manage the data service. Some resource types are predefined in the Solaris Cluster. The HAStoragePlus resource type, for example, is used for providing high availability storage. Once a resource type is created, one or more instances of the data service can be created on any or all of the cluster nodes.

A resource is an instance of a resource type. For example, the IP address on which a data service listens is a resource of type *LogicalHostname*. When more than one IP address is needed on the same server, multiple resources of the type *LogicalHostname* can be created. Network resources are either *SUNW.LogicalHostname* or *SUNW.SharedAddress* resource types. These two resource types are preregistered by the Solaris Cluster software.

A resource group is composed of one or more resources that are dependent on each other. The resources in a resource group failover together, as a unit. When one of the resources needs to failover to another node, the entire resource group is failed over. For example, a resource group for the *SASMetaData* data service contains a resource and a logical hostname resource. If one of these resources fails, the whole group fails. A given failover resource group and its resources can only be online on one node at any time. During failover, resource groups and the resources within them are taken offline, or shutdown, on one node and are then brought online on another node.

Solaris Cluster provides a wizard to help create resource types, resource groups, and resources. The wizard creates a Solaris package that holds all the information and scripts needed for deployment. This package needs to be deployed on all cluster nodes.

One critical element of data services is network connectivity. A resource that has the *LogicalHostname* resource type is one of the failover resources and part of a failover resource group. This resource allows Solaris Cluster to bind an IP address to one node at a time in a cluster. If there is a failure that forces the entire resource group to be switched to another node, the IP address associated with the LogicalHostname follows. This allows the application to continue to be accessible at the same IP address regardless of the node on which it is running.

Solaris cluster data services for SAS

Sun and SAS have created scripts to help organizations configure failover data services for the SAS Metadata Server, SAS OLAP Server, object spawner, SAS/CONNECT, and SAS/SHARE. The scripts provide start, stop, and monitor functionality. The companion guide to this paper, *SAS Enterprise Intelligence Platform High Availability Implementation Guide*, provides details on how to use the scripts to create SAS data services, as well as how to use the scripts to provide automatic restart within the Solaris Service Management Facility. The scripts are available from the Sun and SAS Datacenter of the Future Knowledge Center: *http://www.sas.com/partners/directory/sun/knowledge.html*

Layer 6 — Campus, metro, and geographic failover

The final layer should be implemented by organizations that rely so heavily on the SAS Enterprise Intelligence Platform that any downtime would be detrimental to the business. Layer 6 protects against catastrophic failures such as the loss of a datacenter, campus facility, major metropolitan power loss, etc.

Campus failover

Campus clustering enables components, such as nodes and shared storage, to be located up to 10 kilometers apart. In the event of a localized disaster such as a flood, fire, or building power outage, the surviving nodes can support the service for a failed node.

Metro failover

For greater availability across an increased distance, cluster nodes can be separated by up to 400 kilometers using Solaris Cluster software and dense wave division multiplexing (DWDM) technology to provide application service continuity in the event of a catastrophic failure. However, this is a single cluster solution supported across a limited distance that can still be affected by a single disaster. Various methods for replicating data to distance-separated nodes can be employed including tape backup/restore, remote mirror, synchronous or asynchronous host-based data replication, or synchronous storage-based replication. Sun offers solutions and products to implement all of these capabilities. Enterprises can take advantage of industry-leading data replication software from Sun, Hitachi, and EMC to manage their Solaris Cluster environments across multiple data centers.

Geographic failover

Sun Cluster Geographic Edition software enables a multi-site disaster recovery solution designed to manage the availability of application services and data across geographically dispersed Solaris Clusters. In the event of a disaster in which the primary Solaris Cluster becomes unavailable, Sun Cluster Geographic Edition software enables IT operators to start up the business services with replicated data on the secondary Solaris Cluster.

Geographic clustering employs multiple clusters of systems separated by long distances, a duplicated application configuration, and redundant storage infrastructure to replicate data between these clusters. Because the data is replicated between clusters, application services can be migrated to a geographically separated secondary cluster in the event of a disaster or planned maintenance.

Data is continuously copied or replicated from the primary cluster to the secondary cluster. Sun Cluster Geographic Edition software currently supports Sun StorageTek™ Availability Suite software, Sun StorageTek 9900 TrueCopy software, and EMC Symmetrix Remote Data Facility software for data replication. Sun Cluster Geographic Edition software also supports protection groups that do not require data replication, offering flexibility for environments that handle data replication differently.

Chapter 3
Conclusion

The SAS Enterprise Intelligence Platform can provide organizations with the power to enable their entire enterprise to better understand their data in order to compete using sophisticated analytics. If a component in the SAS Enterprise Intelligence Platform is unavailable, even for a short period of time, it can impact an organization's ability to leverage its information resources, with dramatic results. Implementing as many layers of high availability as economically possible for systems that run SAS components can help create a highly available SAS Enterprise Intelligence Platform environment, where the right data is available — all the time.

Sun realizes that architecting an end-to-end highly available infrastructure can be a challenge and believes the people who intimately understand the technology are the best people to determine specific needs and design the right solution. Toward that end, the consultants in Sun Client Solutions provide a business-focused, architecture driven approach that uniquely satisfies high availability deployment challenges.

Next steps

Sun^{SM} Continuity and Recovery Services can help enterprises reduce risk and deployment times of HA and disaster recovery solutions. Sun experts start by reviewing business continuity and disaster recovery procedures, including the people, processes, systems, network, data, and facilities that are critical for business continuation. Following a thorough analysis, a detailed solution and process plan can be designed to recover operations in the event of unplanned interruptions.

References

Sun Microsystems posts product information in the form of data sheets, specifications, and white papers on its Web site at *www.sun.com*.

The scripts mentioned in this paper, as well as the *SAS Enterprise Intelligence Platform High Availability Implementation Guide*, are available from the Sun and SAS Datacenter of the Future Knowledge Center: *http://www.sas.com/partners/directory/sun/knowledge.html*

For information on Sun's Datacenter of the Future for SAS Enterprise Intelligence Platform visit *www.sun.com/sas*.

To contact the Sun Solution Center for SAS Competency, select the Contact tab from the following Web site: *www.sun.com/third-party/global/sas/sas-cc.jps.*

For more information on the SAS Enterprise Intelligence Platform see: *www.sas.com*.

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA **Phone** 1-650-960-1300 or 1-800-555-9SUN (9786) **Web** sun.com