

High Availability for SAS[®]9 Metadata Server

White Paper
November 2005

Table of Contents

Introduction to Sun Cluster.....	3
High Availability for SAS Metadata Server using Sun Cluster.....	3
Creating a SAS Metadata Server Data Service for Sun Cluster software.....	4
Demonstrating High Availability of SAS Metadata Server.....	5
Conclusion.....	5
Appendix A: OMRconfig file.....	6
Appendix B: pingMetadata script.....	7
Appendix C: Solaris 10 Service Manager for Single-node cluster.....	7
References.....	8

Introduction to Sun Cluster

Sun Cluster software is the industry's premier availability platform for improving the predictability and resilience of mission-critical applications. Sun Cluster software is designed to deliver highly available service levels by providing continuous network, data, and application availability, making recovery from failure transparent to clients. Sun Cluster software automatically restarts a failed resource — on the same node if it is healthy or on a backup node if it is not — without manual intervention. This enables more predictable application service levels while simultaneously reducing risk and costly outages.

The Sun Cluster software extends the Solaris Operating System into a cluster operating system. Key fault detection and recovery functions of the Sun Cluster software are integrated with the operating system, leading to faster failure detection and recovery. A cluster is two or more systems, or nodes, that work together as a single, continuously available system. Each node on a cluster is a fully functional standalone system that runs its own processes. The nodes of a cluster communicate with one another to form what looks like (to a network client) a single system. Failures that would normally bring down a single server system are automatically remediated by the cluster. No single failure—hardware, software, or network—can cause a cluster to fail.

High Availability for SAS Metadata Server using Sun Cluster

Time to intelligence is critical for any business to make quick and accurate, business decisions. The SAS Business Intelligence Server provides an enterprise view of business intelligence. It is critical that this enterprise application and its data are highly available. The foremost component of the SAS®9 architecture which requires high availability is the SAS Metadata Server. The SAS Metadata Server is an “in-memory” server, which means all queried and updated metadata is held in memory by the server. All modifications to these metadata memory pages are committed to the filesystem storage before completing. If a failure occurs no metadata is lost or corrupted. High availability software like Sun Cluster is required to enable more predictable application service levels and availability of the SAS Metadata Server. The key steps in using Sun Cluster software for SAS Metadata Server high availability is:

- Cluster hardware configurations and cabling
- Installing the Solaris Operating System
- Installing and configuring a volume manager
- Installing, establishing and configuring Sun Cluster software
- Installing and configuring SAS Metadata Server software
- Creating a SAS Metadata Server data service for Sun Cluster

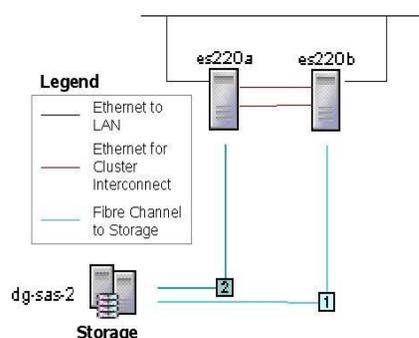


Figure 1: Two-node cluster topology

For this paper, the assumption is that all but the *Creating and installing a SAS Metadata Server data service for Sun Cluster* is complete. **Figure 1** represents the cluster topology that was used for this exercise. Important elements are: the cluster has two nodes – a primary and stand-by; the cluster interconnect for cluster communication; network connections to the LAN; and both nodes connected to a shared storage device. For information on how to setup the cluster, see the Reference section at the end of this paper.

Creating a SAS Metadata Server Data Service for Sun Cluster software

As mentioned earlier, each node in a cluster is a fully functional standalone system that has its own network identity – a physical hostname. In this case, the primary node's physical hostname is es220a and the stand-by node is es220b. However, SAS does not reference the cluster using the physical hostnames, rather SAS references the cluster by the logical hostname of sas-ms. During this last step of cluster configuration, this logical hostname will be created. This logical hostname will fail-over from the primary cluster node to the stand-by node when a failure occurs to the primary node; thereby, the nodes in the SAS cluster communicate with one another to form what looks like (to a network client) a single system. When the SAS administrator is using the SAS Management Console, the profile's server name would be sas-ms.

The summary of the steps to create a SAS Metadata Server data service for Sun Cluster are:

- Register a resource type
- Create a failover resource group
- Add logical hostname to resource group
- Add a Failover Application Resource to a resource group
- Bring the resource group and its associated resources online

The term “data service” describes 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, Sun Cluster configuration files, and management methods that control the starting, stopping and monitoring of the application. The Sun Cluster software provides a generic data service. Using this generic data service, a data service for SAS Metadata Server can be generated very quickly. The only information needed is the SAS Metadata Server start, stop and monitor methods. For this exercise, the standard SAS Metadata Server start and stop methods were used. To monitor the SAS Metadata Server, the OMRconfig file and the pingMetadataServer script were used (see Appendix A and B).

Register the resource type

```
# scrgadm -a -t SUNW.gds
-a Adds the specified resource type.
-t Specifies the type of resource we are using. In the case, the "generic" data service
```

Creating a Failover Resource Group

```
# scrgadm -a -g Metadata-harg -h es220a es220b
-h Specifies the primary and stand-by physical hostnames that make up the cluster
```

Adding a Logical Hostname Resource to a Resource Group

```
# scrgadm -a -L -g Metadata-harg -l sas-ms
-l The logical hostname that the SAS Management Console profile definition will use.
```

Add a Failover Application Resource to a Resource Group

```
# export SASMAIN=/global/sas/d1/CFG/Lev1/SASMain/
# export PROBE_HOME=$SASMAIN/MetadataServer/MetadataExamples/
# export META_HOME=$SASMAIN/MetadataServer
# scrgadm -a -g Metadata-harg -t SUNW.gds -j sas-rs \
-x Start_command="$META_HOME/MetadataServer.sh start" \
-x Stop_command="$META_HOME/MetadataServer.sh stop" \
-x Probe_command="$PROBE_HOME/pingMetadataServer.sh \
-y Port_list=8561/tcp -y Network_resources_used=sas-ms
```

```

Bringing a Resource Group online
# scswitch -Z -g Metadata-harg

```

Table 1: Create SAS Metadata Server monitor for Sun Cluster

Demonstrating High Availability of SAS Metadata Server

Let's demonstrate that the SAS Metadata Server is now being monitored by Sun Cluster and is highly available.

In **Table 2** section, the following actions will be taken:

- Display the status of the cluster
- Verify that the SAS Metadata Server is running
- Kill the SAS Metadata Server simulating a hardware or software failure
- Verify that the Sun Cluster software restarts the SAS Metadata Server

```

Display the status of the cluster; es220a is primary and online
root@es220a # scstat -g

          Group Name          Node Name          State
          -----          -
Group: Metadata-harg         es220a             Online
Group: Metadata-harg         es220b             Offline

Verify that the SAS Metadata Server is running
root@es220a # ps -ef | grep Meta
root 18402 ... /global/sas/d1/sas/sasexe/sas -log
                /global/sas/d1/CFG/Levl/SASMain/MetadataServer
...
root 18387 ... gds_svc_start -R sas-rs -G Metadata-harg
root 18392 ... /bin/sh ./MetadataServer/MetadataServer.sh start2

Kill the SAS Metadata Server simulating a hardware or software failure
root@es220a # kill 18402
root@es220a # ps -ef | grep Meta
root 18468  8722  0 10:22:53 pts/2    0:00 grep Meta

Verify that the SAS Metadata Server is running on es220b
root@es220a # scstat -g

          Group Name          Node Name          State
          -----          -
Group: Metadata-harg         es220a             Offline
Group: Metadata-harg         es220b             Online

```

Table 2: Demonstrating HA SAS Metadata Server using Sun Cluster

In the case of a simple software failure, Sun Cluster can be used to restart the Metadata Server on the same node of the cluster. However, for this paper the exit status, called the **probe status**, is set to 201, which causes the application to immediately fail over. See **Appendix B** for more details.

Conclusion

Business Intelligence is a business-critical, enterprise application. As customers move from a best of breed, departmental BI tools to an enterprise-class, intelligence platform from SAS, the SAS Metadata Server is central to that intelligence platform and provides centralized, enterprise-wide metadata management. Customers in every industry have concerns about any downtime of their SAS environment because downtime can mean lost productivity, revenue, customers and opportunities. The SAS Metadata Server provides services to all SAS applications in the enterprise supporting multiple repositories and thousands of concurrent users. For this reason alone, the SAS Metadata Server requires high-availability. This paper's purpose is to demonstrate how Sun Cluster Software, which extends the Solaris Operating System into a cluster operating system, is designed to deliver highly available service levels for the SAS Metadata Server. Sun Cluster improves the service-level and resilience of SAS as a mission-critical application making recovery from failures transparent to clients.

Appendix A: OMRconfig file

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<Redirect>
```

```
<LogicalServer Name="SAS Metadata Server" ClassIdentifier="2887E7D7-4780-11D4-879F-00C04F38F0DB">
```

```
<UsingComponents>
```

```
<ServerComponent Name="SAS Metadata Server" ClassIdentifier="2887E7D7-4780-11D4-879F-00C04F38F0DB">
```

```
<SourceConnections>
```

```
<TCPIPConnection Name="SAS Metadata Server" ApplicationProtocol="Bridge"  
CommunicationProtocol="TCP" HostName="sas-ms" Port="8561">
```

```
<Domain>
```

```
<AuthenticationDomain Name="DefaultAuth">
```

```
<Logins>
```

```
<Login Name="sasadm" UserId="sasadm" Password="sunone01" />
```

```
</Logins> </AuthenticationDomain> </Domain>
```

```
<Properties>
```

```
<Property Name="Repository" PropertyName="Repository"  
DefaultValue="Foundation" />
```

```
</Properties>
```

```
</TCPIPConnection>
```

```
</SourceConnections>
```

```
</ServerComponent>
```

```
</UsingComponents>
```

```
</LogicalServer>
```

```
</Redirect>
```

Appendix B: pingMetadata script

```
#!/usr/bin/ksh
# The location that the necessary jars were installed
export utildir=/global/sas/d1/CFG/Lev1/SASMain/MetadataServer/MetadataExamples/
#
# Run a quick program to get the status.
java -jar "$utildir/lib/sas.metadata.utilities.jar" -config "$utildir/config/OMRConfig.xml" \
-inxml "$utildir/pingMetadataServer/xml/getStatus.xml" \
-inxsl "$utildir/pingMetadataServer/styles/getStatusResult.xsl" \
-log "$utildir/pingMetadataServer/logs/getStatus.log" -exitStatus -debug

exitval=$?
echo $exitval
if (($exitval == 0));
then
    status=$exitval
else
    status=201
    #echo "SAS Metadata Server failure – Restarting Metadata Server"
fi

echo $status
exit 0
```

The pingMetadata is provided courtesy of SAS R&D. The pingMetadata command periodically checks the health of the SAS Metadata Server. The probe command returns with an exit status of 0 if the application is running correctly.

The exit status of the probe command is used to determine the severity of the application's failure. This exit status, called the **probe status**, must be an integer between 0 (for success) and 100 (for complete failure). The probe status can also be a special value of 201, which causes the application to immediately fail over unless **Failover_enabled** is set to FALSE. The GDS probing algorithm uses the probe status to determine whether to restart the application locally or fail it over. See the [scds_fm_action\(3HA\)](#) man page for more information. If the exit status is 201, the application is immediately failed over.

Appendix C: Solaris 10 Service Manager for Single-node cluster

Solaris 10 Service Manager for Single-node cluster: Self-healing for Applications

Solaris 10 Service manager is a feature to enable services to be more robust, reliable and easier to manage. It is a core part of the Predictive Self-Healing technology available in Solaris 10, which provides automatic recovery from software and hardware failures. Failed services are restarted in dependency order. The services can be started using an XML manifest. In the manifest you tell it how to start it, what its dependencies are, and how to stop it. If SAS has a software error and terminates, Solaris Service Manager will restart SAS using the XML manifest. If hardware fails SAS will restart when the system reboots.

References

Sun™ Cluster Overview for Solaris OS

<http://docs.sun.com/app/docs/doc/819-0579/6n30dc0ms?a=view>

Sun™ Cluster Software — Quality by Design for Advanced Availability

<http://www.sun.com/software/cluster/wp-advancedavail/wp-advancedavail.pdf>

Sun™ Cluster Software Installation Guide for Solaris OS

<http://docs.sun.com/app/docs/doc/819-0420>

Sun™ Cluster System Administration Guide for Solaris OS

<http://docs.sun.com/app/docs/doc/819-0580>

Sun™ Cluster Hardware Collection for Solaris OS

<http://docs.sun.com/app/docs/coll/1024.1>

SAS® 9.1.3 Open Metadata Architecture: Best Practices Guide, Second Edition

<http://support.sas.com/rnd/eai/openmeta/v9/bestpractices/index.html>

SAS® 9.1.3 Metadata Server: Setup and Administration Guide, Third Edition

<http://support.sas.com/rnd/eai/openmeta/v9/setup/index.html>

References

System Administration Guide: Solaris Containers—Resource Management and Solaris Zones

<http://docs.sun.com/app/docs/doc/817-1592>

Internal Zone configuration

<http://docs.sun.com/app/docs/doc/817-1592/6mhahuopn?a=view>

Solaris 10 - In a class by itself

<http://www.sun.com/software/whitepapers/solaris10/classbyitself.pdf>

Solaris Containers Whitepaper

http://www.sun.com/software/whitepapers/solaris10/grid_containers.pdf

Sun Solaris Containers vs. IBM Logical Partitions and HP Virtual Partitions

http://www.sun.com/datacenter/consolidation/docs/Sun_Solaris_Container_vs_IBM_HP_Partition.pdf

