Paper 468-2013

Enhance Your High Availability Story by Clustering Your SAS® Metadata Server in SAS® 9.4

Bryan Wolfe and Amy Peters, SAS Institute Inc., Cary, NC

ABSTRACT

It can be challenging to use clustered file systems, backups, and system tools to ensure that the SAS® Metadata Server is always available. In SAS® 9.4, you can create and manage a clustered metadata deployment using SAS® tools to remove single-point-of-failure concerns for the SAS Metadata Server. The clustered SAS Metadata Server keeps its data in sync, balances its load, and continues to handle requests if a node fails, all while presenting a single face to the outside world. Once it is set up, you do not need to treat a clustered SAS Metadata Server any differently than you do a single, unclustered server.

INTRODUCTION

SAS® products are used in a wide variety of applications that are often critical to business processes. A frequent requirement for business-critical applications is high availability. While previous SAS releases provided high availability through failover techniques, many customers have requested more robust hot failover or cluster support for high availability. SAS 9.4 introduces metadata server clusters to meet this need. A metadata server cluster is a coordinated set of metadata servers that act as a single metadata server for a SAS® software deployment. While the primary driver for this new feature was automated failure recovery, clustering also provides a scalable metadata server that can better address larger deployments than previous releases.

ARCHITECTURE

A metadata server cluster is a group of three or more nodes configured as identical metadata servers, as shown in Figure 1. Each node typically runs on an independent physical or virtual machine and has its own server configuration directory, configuration files, journal file, and logs. The nodes work together so that they each have synchronized on-disk and in-memory copies of the metadata.

At any given time, one node in the cluster is designated as the master node. The master node is responsible for coordinating any metadata updates across the entire cluster. The master also coordinates the addition or removal of any node to or from the cluster due to configuration changes, failure, or administrator actions. The master node is chosen each time the cluster is started and can change over time. Normally the first node to start will become the master, although synchronization and failure recovery rules might cause another node to be chosen as master.

All running cluster nodes other than the master are slave nodes. Slave nodes have a direct connection to the master node, but they are generally unaware of other slave nodes. Slave nodes process read requests locally. Update requests are forwarded from a slave node to the master node. The master node serializes multiple update requests, performs constraint checks on update requests, creates a journal entry for each transaction, and broadcasts the transaction to all slave nodes. The master and all of the slave nodes apply the transaction to both their in-memory and on-disk metadata repositories. The slave node that originated the update request completes the request and responds to the client only when it receives the transaction back from the master node.

A client can connect to any node in the cluster. The cluster uses a built-in load balancing feature to control access to the cluster, which means that any client connection can be redirected to another node. Redirections to another node are generally invisible to the client, and clients are typically unaware that they are connecting to a cluster. Once connected, client applications and users interact with the cluster in exactly the same way that they would interact with a metadata server that is not clustered. The cluster load-balancing algorithm routes new connections to a running slave node using a round robin algorithm. Connections are always routed away from the master node. Therefore, in a normal cluster, the only connections to the master node are from the slave nodes and from administrative clients. Load balancing happens at connection time only. Once a client is connected, it can never be redirected to another node. If the node fails, the client must reconnect to another node. For certain administrative actions, a client can specify a special “noredirect” option and disable load balancing for that one connection.

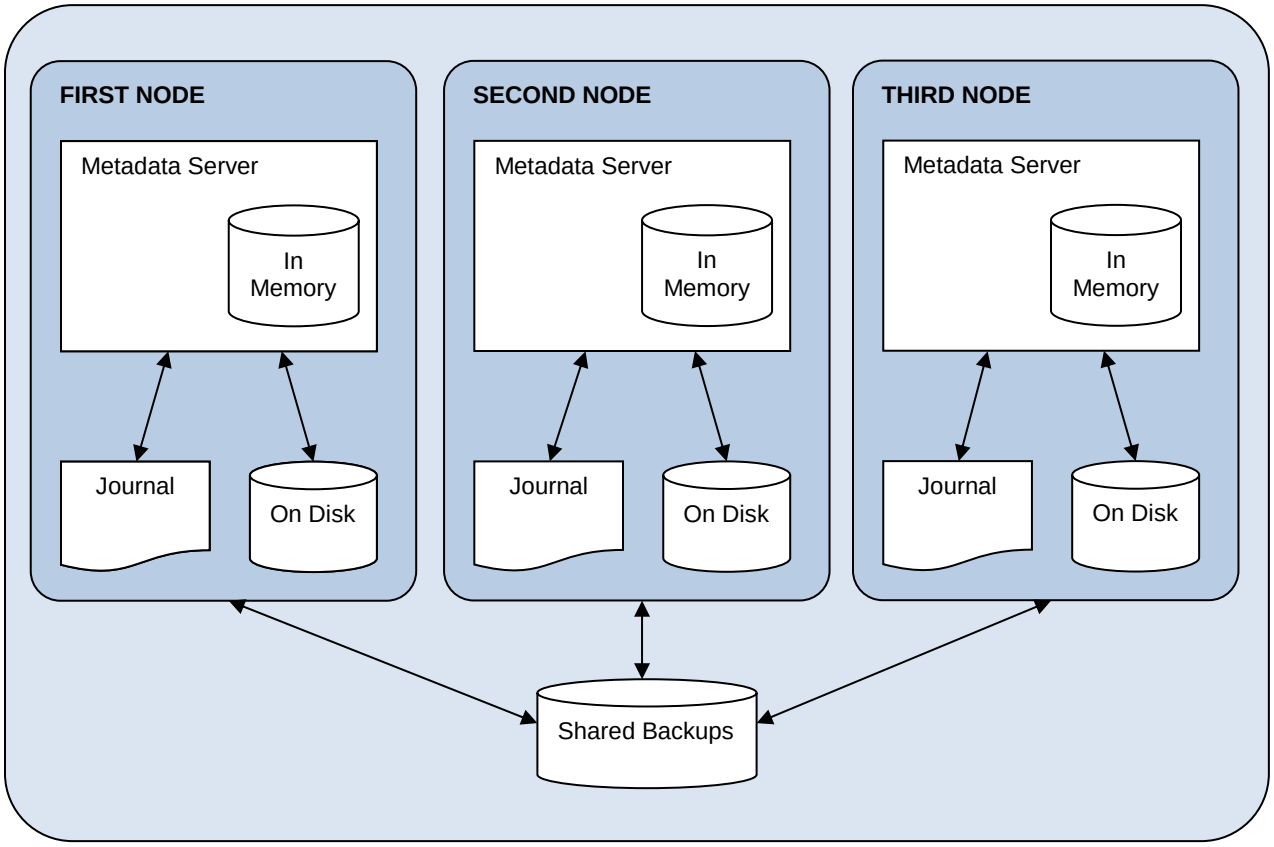

Figure 1. Metadata Server Cluster

Not all nodes in a cluster are required to be running. The cluster must be tolerant of individual node failures to provide high availability. The cluster does require a majority of the configured nodes to be running before the cluster is online and available for use. This is known as the quorum rule. The quorum rule is necessary to avoid synchronization failures after the recovery of a failed or stopped node. If the quorum rule is not met, either because the cluster is starting or because too many nodes have failed, the cluster is placed in an offline state. Clients can connect or remain connected, but they cannot access metadata until the quorum is established.

In addition to enforcing the quorum rule, the cluster must react to node failures. If the master node fails, another node is immediately promoted to serve as the master node, and the cluster resumes operation. If a slave node fails, it drops out of the cluster, and load balancing uses only the remaining slave nodes. In either case, client connections to a failed node are lost and the client must reconnect to the cluster. These failover processes occur automatically with no need for user or administrator intervention and, in most cases, with no noticeable effect on user activity.

CONFIGURATION

Metadata server clusters are fully supported by the standard SAS® Deployment Wizard and SAS® Deployment Manager configuration tools. A metadata server cluster will require the following:

- a standard or custom plan file with a Metadata Server Node step. By default, all SAS 9.4 deployment plans will include this step.
- three Windows or UNIX machines or partitions to run the cluster nodes. Each node is a complete metadata server instance and has the same hardware requirements as a normal metadata server installation. The metadata server node hardware can be used for other SAS software tiers or other processing loads but, as with unclustered metadata servers, care must be taken to avoid performance problems. The machines are not required to be identical, but they must use the same OS release and the same maintenance release of SAS. Common filesystem layouts are recommended to simplify maintenance tasks and troubleshooting.
- a network-accessible filesystem that is available to all cluster nodes for use as a shared backup area. The shared backup area must have the same path on all cluster nodes. The shared area is used only for backup, recovery, and node synchronization operations. It is not required to be a high-performance filesystem.

- a common account for use in starting each node. For Windows systems, a domain account must be used instead of the default LocalSystem account.

The cluster does not rely on any clustering, load balancing, or other high availability software from third parties. The SAS 9.4 cluster feature provides a high availability metadata server built on standard commodity hardware and software.

A typical cluster consists of three metadata server nodes, each on its own host system. More nodes can be added for large deployments with very high performance requirements. Three nodes are the minimum number required to provide a high-availability cluster.

Each node of the cluster is a full instance of the metadata server, with a complete copy of all metadata, and the normal system requirements for an unclustered metadata server apply. All nodes in a cluster must run on the same host OS and the same version of SAS. Hardware requirements will vary widely depending on the product mix and usage, but 8GB of RAM and a quad core CPU are a good baseline configuration. Many deployments will run with 4GB or less of RAM, but it is difficult to provide accurate estimates of resource requirements. Very large deployments can require 16GB or more of RAM, although this is fairly unusual.

Disk I/O performance is not as important for the metadata server as it is for most SAS processing. The metadata server uses an in-memory database, which means that most disk I/O processing is done in a background process and does not affect the performance of client requests. Any server-class disk I/O system is likely to be adequate for a metadata server deployment.

LICENSING

All customers who license products or solutions built on the SAS® Intelligence Platform or SAS® Integration Technologies are provided with a limited-use license that allows the SAS Metadata Server to be deployed either on the same machine as other SAS processing or on a machine dedicated to the metadata server. This license includes the ability to run a metadata server cluster on a set of machines not licensed for other types of SAS processing. A custom order might be required in some cases.

NEW DEPLOYMENTS

The first node of the cluster is configured as it would be for any SAS deployment. The only significant difference is that you must override the default backup location for the metadata server and specify an absolute path to a network-accessible filesystem. The path should be valid on all nodes of the metadata server. On Windows deployments that use services to run the metadata server, the default LocalSystem account cannot be used; a domain account valid for all nodes in the cluster must be used for the service definition.

The second, third, and any other nodes in the cluster are configured by running the SAS Deployment Wizard and choosing the Metadata Server Node step from the plan file. The SAS Deployment Wizard prompts for connection information to the first node and a few other options before configuring the additional node. The configuration process synchronizes the new node with the original node and brings the new node online in the cluster.

The recommended best practice is to use the same configuration directory path (including the same Lev*n* directory) and the same TCP port for all nodes in the cluster. Identical directory paths and ports are not required, but administration is simpler if the same paths are used for all files on all cluster nodes.

MIGRATION

Metadata server clusters are an exception to the limitation that normally prohibits deployment topology changes during migration using the SAS® Migration Utility and the SAS Deployment Wizard. Unclustered systems can be migrated to a cluster, a cluster can be migrated to an unclustered system, and cluster nodes can be added or removed during migration. Migration needs to move only one copy of the metadata, so the SAS Migration Utility is run only on the first node if the source system uses a cluster. On the target system, the SAS Deployment Wizard is run in migration mode only on the first node of the new system. The SAS Deployment Wizard migrates the metadata to the first node, and any cluster configuration information that is found in the source system's metadata is erased. As with a new deployment, a shared backup space and an appropriate account for running the server node must be specified. Cluster nodes can then be added using the Metadata Server Node step in the SAS Deployment Wizard in the same fashion as new deployment.

CONVERTING TO A CLUSTER

Converting an unclustered system to a cluster is straightforward. The existing metadata server deployment will become the first node in the cluster and must be prepared for operation in a cluster. The backup location for the metadata server must be set to a network filesystem that is accessible to all nodes in the cluster. This location can be

specified in SAS Management Console Metadata Manager. If the cluster is running as a service on a Windows system, the service must be running under a domain logon that is valid for all nodes in the cluster. If necessary, change the service logon identity in the Services control panel and restart the service. Note that the service logon must have access to the metadata server configuration area. This is usually done by making the service logon a member of the Administrators group.

Go to the machine that is to host the second node, and run the SAS Deployment Wizard from the same SAS Software Depot that was used to configure the original system. Choose the original plan file and the Metadata Server Node step. Configure the node as you would for a new cluster configuration. You will be prompted for a connection to the first node. You must also enter the backup location and the service logon identity (if applicable). The new node will be synchronized with the original node. Repeat this process for the third node.

CHANGING THE CLUSTER

Nodes can be added to or removed from an existing cluster at any time. New nodes can be added by running the SAS Deployment Wizard on the new machine and choosing the Metadata Server Node step from the original plan file. The prompting and configuration sequence is no different than for a new cluster configuration. The new node is synchronized with metadata from the existing cluster. Nodes can be removed from the cluster by using the unconfigure tool in SAS Deployment Manager. Unconfiguring all the nodes except the first node converts the deployment back to an unclustered metadata server.

Note: The initial release of SAS 9.4 requires that the entire cluster be restarted after any node is unconfigured. This requirement might be removed in a future release.

The configuration tools place special restrictions on the first cluster node to be configured. While the cluster high-availability feature allows the SAS deployment to continue running if any node (including the first node) fails, the first node must be running when configuration is being performed with the SAS Deployment Wizard or SAS Deployment Manager. The SAS Deployment Wizard and SAS Deployment Manager are not aware of the cluster; they know only the address of the first node. If the first node is not running, these tools are unable to access metadata. This restriction also means that the first node can never be unconfigured unless the entire SAS deployment is unconfigured.

Most metadata server clients in SAS 9.4 have the ability to automatically discover cluster configuration changes. This feature is very useful, because it means no manual reconfiguration of other tiers is required when nodes are added to or removed from a cluster. A notable exception to this is the SAS server tier. SAS application servers rely on the metadataConfig.xml file in the root configuration directory to know about the metadata server cluster. The object spawner and SAS/CONNECT® spawner configuration directories might also contain metadataConfig.xml files. These files must be updated manually on any SAS server tier machine whenever a cluster configuration is changed after the initial deployment. A batch tool called sas-update-metadata-profile is provided to simplify this process.

ADMINISTRATION

Many metadata administrative tasks are unchanged for the metadata server cluster. Tasks such as creating a repository, pausing and resuming the server, and performing a backup apply to the cluster as a whole and work just as they do for an unclustered metadata server. This section discusses administrative tasks that are unique to clusters.

STARTING AND STOPPING A CLUSTER

Starting and stopping a cluster is more complex than starting and stopping a single server because server processes are spread across multiple machines. The “start” option of the MetadataServer batch utility starts only the local node. As of the initial release of SAS 9.4, no utility is provided to start all nodes together. It would be easy to create such a script for a Windows configuration in which the nodes are installed as services, but this would be more difficult for deployments on other platforms. Note that SAS has always required a specific start-up sequence when a deployment is spread across multiple machines. A metadata server cluster must now be fully started (or at least reach quorum) before the other tiers can be started.

The MetadataServer batch utility in SAS 9.4 supports both a “stop” option and a “stopcluster” option. The “stop” option stops only the local node, while the “stopcluster” option stops all running nodes in the cluster. The Stop menu action in SAS® Management Console Metadata Manager stops the entire cluster.

INSPECTING THE CLUSTER

Surprisingly, the SAS Management Console Metadata Manager is almost unaffected by the addition of clusters. A new Cluster property tab (Figure 2) shows the current state of the cluster. All other actions and views in Metadata

Manager are unchanged, since they apply either to the metadata that is hosted by the cluster or to the overall state of the cluster.

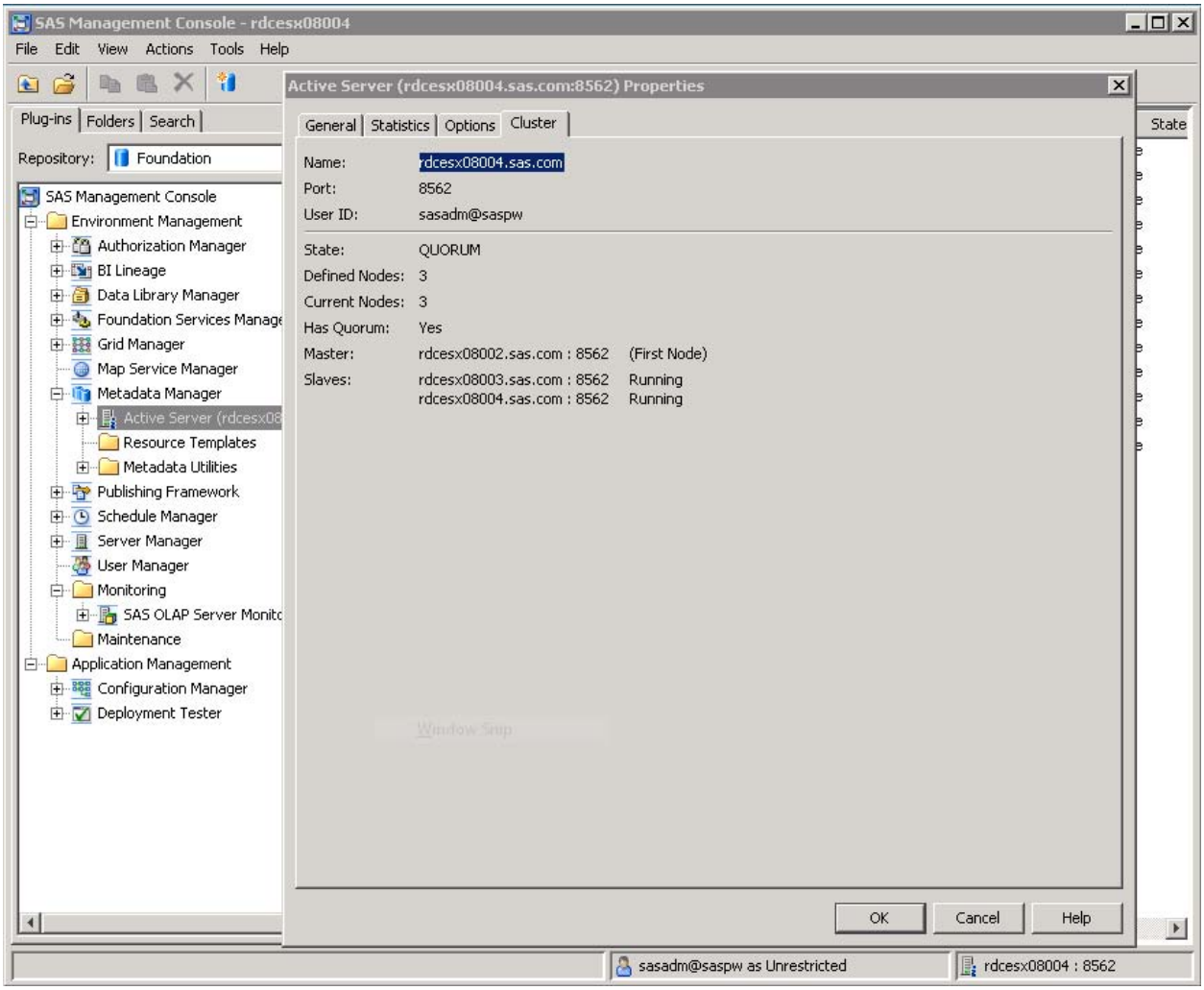

Figure 2. SAS Metadata Manager - Cluster Properties

SAS Management Console Server Manager can be used to inspect and control individual nodes in a cluster. Figure 3 shows performance counters for the third node of a three-node cluster. The nodes are located under the SASMeta – Logical Metadata Server node in Server Manager. From there, each node can be accessed, client connections can be listed, logging options can be changed, and counters and logs can be inspected.

Logs for metadata server nodes are located in the same SAS configuration subdirectory as in earlier releases. An example log excerpt for node start-up is shown in Listing 1. In this example, the node starts and finds its cluster definition in metadata. It then goes into OFFLINE state until the cluster quorum is reached. After failing an initial attempt to connect to another node, it receives a connection from the second node. The two nodes confer, and the first node is selected as master. The second node does not have all the metadata updates found on the first node and must be synchronized. Once the second node is synchronized, the cluster achieves quorum and goes ONLINE. The third node connects during this period, is synchronized to the master, and joins the cluster.

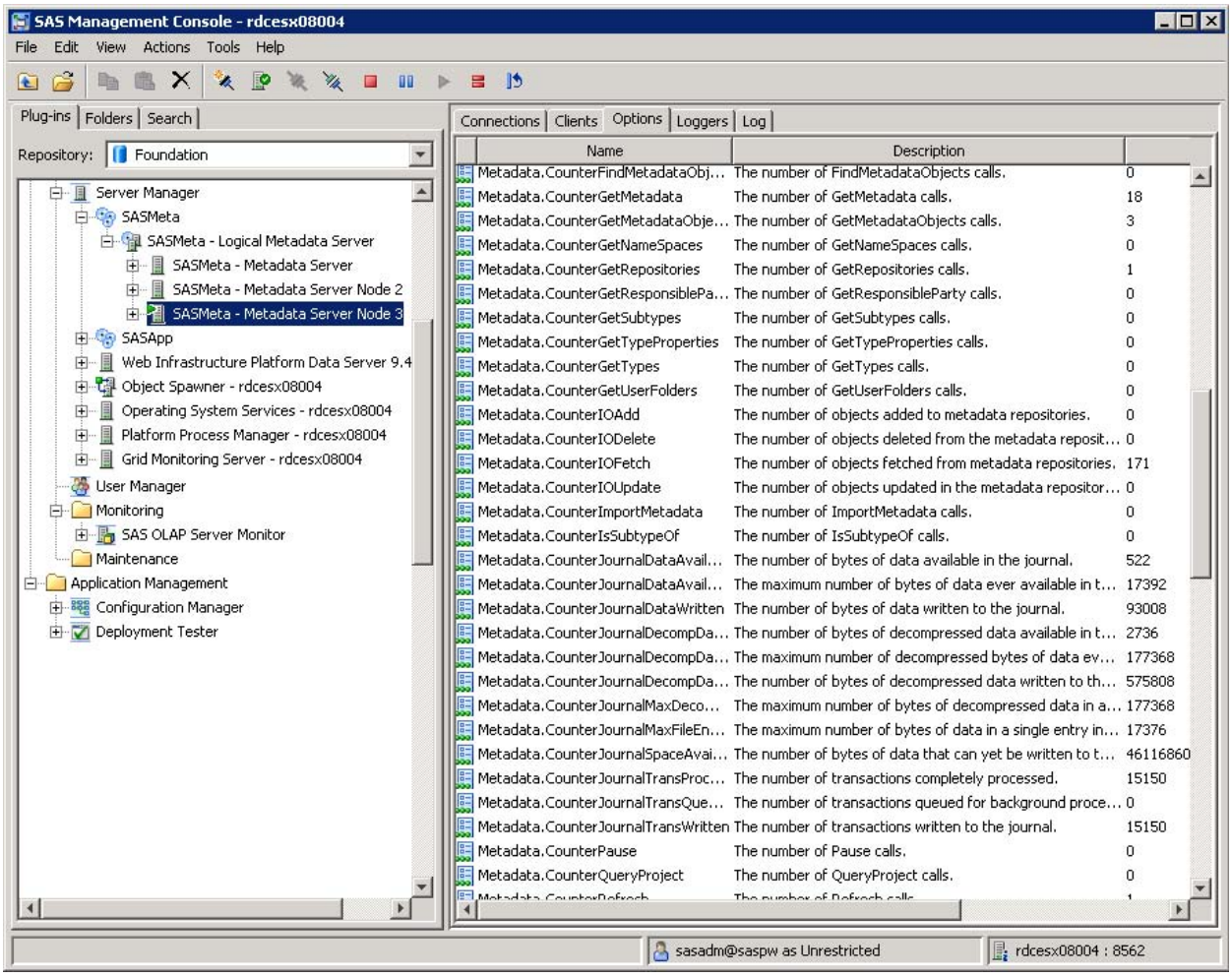

Figure 3. SAS Management Console Server Manager - Metadata Server Node Counters

```
2013-03-12T12:06:53,159 INFO - SAH011001I SAS Metadata Server (8561), State, starting
2013-03-12T12:06:56,035 INFO - Cluster SASMeta - Logical Metadata Server (A5LOAHYI.AX000001)
containing servers … found.
2013-03-12T12:06:56,035 INFO - The server is part of a cluster.  It is being switched OFFLINE
until we join the cluster and achieve quorum.
2013-03-12T12:06:56,097 INFO - Activated listen on IPv6 port 8561 (connection 1).
2013-03-12T12:07:07,660 INFO - Attempt to connect to peer …8562… failed …
2013-03-12T12:07:14,582 INFO - New client connection (2) accepted from server port 8561 for
IWA user sas. Encryption level is Credentials using encryption algorithm SASProprietary.  Peer
IP address and port are [::1]:50493 for APPNAME=SAS Metadata Server.
2013-03-12T12:07:14,582 INFO - Connecting server SASMeta - Metadata Server Node 2 to cluster
SASMeta - Logical Metadata Server.
2013-03-12T12:07:14,582 INFO - Setting the master node to SASMeta - Metadata Server.
2013-03-12T12:07:14,661 INFO - The metadata on the connecting server is sufficiently different
that the slave server will be recovered to synchronize its metadata with the cluster.
2013-03-12T12:07:17,973 INFO - Some updates are needed to make the metadata on server SASMeta
- Metadata Server Node 2 current.
2013-03-12T12:07:22,442 INFO - New client connection (3) accepted from server port 8561 for
IWA user sas. Encryption level is Credentials using encryption algorithm SASProprietary.  Peer
IP address and port are [::1]:50507 for APPNAME=SAS Metadata Server.
2013-03-12T12:08:20,726 INFO - The cluster has achieved quorum and is now ONLINE.
2013-03-12T12:08:23,460 INFO - Connecting server SASMeta - Metadata Server Node 3 to cluster
SASMeta - Logical Metadata Server.
2013-03-12T12:08:27,460 INFO - Update distribution complete.
```

Listing 1. Metadata Server Node Log at Start-up

BACKUPS

Backups are always performed by the master node and are saved to the shared backup area, which is accessible to all nodes in the cluster. The administrator interfaces for backup are nearly unchanged. Backups can be scheduled or performed manually in SAS Management Console Metadata Manager. A new sas-backup-metadata batch utility allows backups to be executed from the command line or a batch script. Backups can also be performed automatically as a result of some node synchronization and configuration processes.

Backup recoveries on a cluster require a manual process in SAS 9.4. The process is as follows:

1. Stop the cluster and any applications that rely on the metadata server.
2. Restart one node of the cluster by using the `MetadataServer startNoCluster` command. This command brings up the node in a stand-alone mode without waiting for other nodes to join the cluster. By convention, this command is normally executed on the first node, but any node can be used.
3. Use SAS Management Console Metadata Manager or the new sas-recover-metadata batch tool to recover the backup. Any valid backup can be recovered, and the roll-forward feature can still be used. You should be careful not to restore a backup that was taken before a cluster configuration change (for example, before the removal or addition of nodes) unless you want to undo that configuration change.
4. Stop the metadata server node.
5. Restart the cluster using your normal restart process. All other nodes will synchronize to the node with the recovered backup.

HIGH AVAILABILITY

The metadata server cluster can recover from node or network failures through detection and recovery features that are built into the cluster nodes and all metadata server clients. These features must handle a variety of issues, which are discussed in this section.

SLAVE NODE FAILURE

A failed slave node causes the loss of the connection from the slave to the master node. The master node detects the lost connection, removes the slave node from the active node list, and rechecks for a quorum. If a quorum of active nodes is still present, normal cluster operation continues. If quorum is lost, the cluster goes offline and waits for another node to join the cluster. If the cluster goes offline, existing client connections are not closed, but those clients will not be able to perform any normal metadata activities.

All of the client connections to the failed slave node are lost, and any active transactions might or might not complete. Clients are responsible for reconnecting to a new cluster node and for recovering from any failed transactions. In most cases, client applications simply report an error, reconnect to the cluster, and allow the end user to retry a failed action.

MASTER NODE FAILURE

A failed master node is a more complex problem than a failed slave node. Master node failure causes the loss of all connections from the master node to all slave nodes. Each slave node detects the lost connection, goes offline, and attempts to connect to all the other configured nodes in the cluster. The remaining nodes essentially drop into the same state as newly started nodes. They eventually connect, select a new master, and, if a quorum is present, go back online. If a quorum is not present, the cluster remains offline and waits for another node to join.

Master nodes normally have few, if any, client connections, and these connections are dropped when the master node fails. All of the existing client connections to the slave nodes remain connected, but clients cannot perform any normal metadata activities until the cluster returns to an online status. Requests will fail with a "server paused" error until the cluster achieves quorum.

NODE RECOVERY

A node must be synchronized with the cluster before it can perform as a fully functioning member. This is true for normal node startups, newly configured nodes, and nodes that are rejoining a cluster after a failure. The node connects to the current cluster and compares its current state to the state of the other nodes. Each node maintains a journal of metadata transactions that are guaranteed to be in the same order on every node, so it is relatively easy to determine if a node is synchronized with another node and, if not, how far out of sync it is. If the node is behind the cluster, the node can synchronize itself by requesting the missing transactions from the master and applying them. If

the node is far behind the cluster or is completely out of sync, the node can recover from the most recent backup before requesting missing transactions.

Start-up logic is performed if the cluster is starting or has lost quorum due to multiple node failures. Each newly started node attempts to connect to other configured nodes. Once two nodes connect, they compare transaction journals. The node with the most recent transaction becomes the master. New masters might be selected as each new node joins the cluster until a quorum is established. Synchronization happens each time a new master is selected.

CLIENT CONNECTIONS

Any metadata client can connect to any node in the cluster. The communication protocol includes a load balancing feature that can redirect a connection from one node to another. This redirection is normally hidden from the client application. Therefore, clients can connect to a cluster node even if they are unaware of the cluster. However, this is not sufficient for high availability. A client with no knowledge of the cluster has only a single node address, and if that node fails, the client is unable to connect. A high-availability system requires that each client must maintain a list of cluster nodes. If the first node is not available, another connection attempt must be made to the next node on the list, and so on, until a successful connection is made. The mechanism for creating and maintaining the cluster node list varies depending on the client technology. In most cases, the management of the node list is automatic and requires no administrator or end-user action.

In the event of a node failure, any clients that are connected to that node lose their connections. Any active transactions might or might not complete. Client applications must reconnect to a new cluster node and recover from any failed transactions. In most cases, clients reconnect automatically with no user action. In some case in which an active transaction fails, the client might report an error and allow the end user to retry the failed action.

Clients released before SAS 9.4 (such as older versions of SAS® Enterprise Guide® and SAS® Add-in for Microsoft Office) are unaware of metadata server clusters. These clients do not support automatic reconnection after a node failure. They also must have the host and port address of a running node, which might require client reconfiguration if the first node in a cluster fails.

PERFORMANCE

Performance of the metadata server is a complex subject. It depends heavily on user load, even before clustering is introduced. Performance of specific use cases can usually be determined only by testing those scenarios, but we can make some general statements about performance in a cluster.

For lightly loaded systems, the most interesting performance metric is request response time. The cluster imposes no overhead for read requests. Read requests are handled directly by the node the client is connected to and require no coordination with the master or other nodes. Therefore, the response time from a cluster should be very similar to the response time from an unclustered system. Update requests must be routed from the slave node to the master and back again. The processing required for a response to the client is exactly the same as for an unclustered system, but two additional network hops are required. This overhead is relatively small, and it becomes even smaller for more complex update requests. Of course, the update must be propagated to and performed on each node in the cluster, but these updates do not affect the response time to the client.

For heavily loaded systems, request throughput is the most interesting metric for cluster performance. In the best case for a standard three-node cluster, when all requests are read requests and are evenly distributed across the two slave nodes, the overall throughput of the cluster will be twice that of an unclustered metadata server running on equivalent hardware. At the other extreme, all requests are updates. In this case, they will need to be processed on all three nodes, and the overall throughput of the cluster will be similar to that of an unclustered metadata server. In practice, read requests usually outnumber update requests, and the cluster can be expected to support a noticeably higher request throughput rate, although well less than the theoretical maximum of twice the throughput of an unclustered system.

Adding nodes to a cluster beyond the default three nodes has the potential to improve overall throughput of the system, but only for read requests. As noted previously, read requests are distributed across all the available slave nodes, while update requests must be executed on every node. In the best case, adding a fourth node to a cluster might improve performance by 50 percent, assuming a pure read request load spread over three slave nodes instead of two. In practice, the improvement is likely to be much lower due to the presence of update requests.

There is no design limit on cluster size. There is a default configuration limit of eight nodes, although this limit can easily be increased by editing a configuration file. Almost all performance and high-availability testing for the initial release of SAS 9.4 has focused on the default recommended three-node configuration. The performance characteristics of larger clusters are not well documented.

CONCLUSION

A SAS metadata cluster provides a fault-tolerant metadata server, which is an important part of your high-availability story – automating failover recovery and removing any single point of failure. Its architecture is designed to minimize the impact on SAS users, because client applications interact with the cluster in the same way that they would interact with a metadata server that is not clustered. A cluster is largely self-managed – keeping its components in sync and handling failures automatically – further reducing its impact on SAS administrators. Metadata clusters help you to support your business-critical applications.

RECOMMENDED READING

This paper was written prior to the production release of SAS 9.4. Features and implementation details might change in the production release. Check the product documentation for more up-to-date information, including the following:

- SAS Institute Inc. *SAS 9.4 Intelligence Platform: System Administration Guide*. Available at a future date at http://support.sas.com/94administration.
- SAS Institute Inc. *SAS 9.4 Intelligence Platform: Installation and Configuration Guide.* Available at a future date at http://support.sas.com/94administration.
- SAS Institute Inc. *SAS 9.4 Intelligence Platform: Migration Guide.* Available at a future date at http://support.sas.com/94administration.
- SAS Institute Inc. *SAS 9.4 Language Interfaces to Metadata.* Available at a future date at http://support.sas.com/documentation/onlinedoc/base/index.html.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at Bryan.Wolfe@sas.com or Amy.Peters@sas.com.