Paper 378-2012

From Solo to Symphony: Scaling Up Service Using SAS® Stored Process Server Cluster

Qing Ye, Richard Nardin, Romon Williams, SAS Institute Inc., Cary, NC

ABSTRACT

System architects today often face a difficult challenge: how to scale up existing services to serve more clients, who not only have more requests but also demand faster responses. The answer is to create a server cluster with the support of load balancing. Clusters created using SAS Stored Process Server provide a way to define a service in a metadata server, balance its load for various clients by using Object Spawners, and execute the implementation across different hosts to achieve the desired performance. This paper shows how to set up a server cluster by connecting multiple stored process servers and how to configure the load balancer.

INTRODUCTION

As more business applications centralize their software, system and software architects often face the challenge of providing quick responses to the rapidly increasing number of client requests. For example, software to help make real-time optimal pricing decisions for a worldwide hotel chain might need to answer thousands of concurrent price queries from clients all over the world. Without adding more powerful CPUs and more memory to an existing server, you can solve this scalability challenge by a horizontal scaling approach: connecting servers to form a cluster to increase the service limit. In this way, the additional client requests can be distributed and served by additional machines. This strategy has been implemented well by SAS Stored Process Server software. In this paper, a server that has SAS Stored Process Server software installed is called a stored process server.

SAS STORED PROCESS SERVER

SAS Stored Process Server is designed to execute stored processes and deliver the results to software clients. Interacting with stored process servers, clients can run powerful SAS programs in the reliable server-side environment without needing to install the whole environment on their own machines. SAS Stored Process Server software is designed to serve multiple concurrent client requests. It can be installed on any host that has SAS® Foundation and SAS® Integration Technology software installed.

STORED PROCESSES

A stored process is any SAS program that consists of DATA steps, procedures, and macros that are written between two special stored process macros: %STPBEGIN and %STPEND. The following simple “hello world” stored process prints the input user name:

```
%global username;
*ProcessBody;

%STPBEGIN;
%macro HelloWorld(username=);
    %put Hello &username;
%mend HelloWorld;

%HelloWorld(username=&username);
%STPEND;
```

In SAS 9.3, stored processes provide the following benefits to software applications that are hosted in the centralized cluster to publish results and deliver packages to multiple clients in parallel.

- A variety of clients: Without additional code changes, the same stored process can be executed by many kinds of clients, including clients that have installations of SAS® Enterprise Guide®, SAS® Web Report Studio, SAS® Data Integration Studio, SAS® Add-In for Microsoft Office, and customer clients.
- Multiple concurrent client requests: The same stored process can be run at the same time by multiple clients with or without using sessions.

- Consistent software upgrades: Clients can always execute the latest version of the software, because stored processes are centrally maintained and managed. Code changes are needed in only one place, and these changes are transparent and consistent among all clients.
- Secure access to software: Before SAS 9.3, a stored process usually consisted of two parts: a physical .sas program file and a definition of its metadata in SAS® Metadata Server. Both IO-level and metadata-level security policies were needed to protect the stored process. Starting in SAS 9.3, the contents of the physical program file can also be stored as part of its metadata. In this way, access to the code is consistently controlled by the SAS Metadata Server alone.

INSTALLING A SAS STORED PROCESS SERVER CLUSTER

PREPARING THE SOFTWARE DEPOT

In the SAS installation package that you receive from your SAS representative, make sure that the following SAS products are included, along with any additional SAS software needed to build your own application:

- Base SAS, which provides all the core functionalities in SAS® Foundation
- SAS Integration Technologies, which provides the software that enables you to build a secure client/server infrastructure on which to implement your applications
- SAS Metadata Server, which provides the capacities to store, manage, and maintain the metadata

If any of this required software is missing, you cannot set up a working stored process server cluster.

PREPARING THE PLAN FILE FOR MULTIPLE MACHINES

SAS® Deployment Wizard is the official tool for installing SAS software. A plan file is a critical component that serves as the "driver" for SAS Deployment Wizard to list all the machines that are included in the installation, declare the environments in each machine, and state which software will be installed on which machine. Clearly, a well-defined plan file can make the overall installation easier.

There are two approaches to adding multiple machines in the plan file. The more straightforward approach is to specifically include each machine. But obviously this will quickly become a tedious job for system architects if there are more than 10 machines in the cluster. Every time the cluster needs to be expanded, a new plan file for the additional machine must be created.

A better approach is to create a plan file that is based on the roles and platforms of each machine and to organize them into several categories. As shown in Figure 1, an example of the plan file has the following categories: a metadata server, stored process servers on the Windows platform, and stored process servers on the UNIX platform. In this way, the same plan file can be used repeatedly when a new machine needs to be added to the cluster.

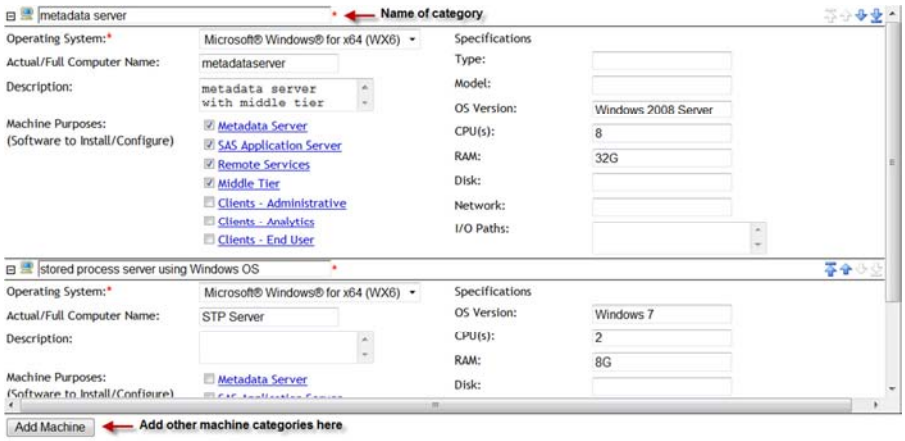

Figure 1. Creating a Plan File by Categorizing Machines in the Cluster

INSTALLATION PROCESS OVERVIEW

After you have the software depot and the plan file, you are ready to create the stored process server cluster. The easiest cluster topology to install is the following:

1. Designate one machine as the metadata server, and install SAS Metadata Server, the Object Spawner, and SAS Stored Process Server.
2. On all other machines, install the Object Spawner and SAS Stored Process Server; these servers will work together with the Metadata Server to form the cluster.

To form a fully functioning cluster, all the installed SAS Stored Process Servers must be put under the same logical stored process server, which is a logical definition that serves as a "container" ("folder") of the real stored process servers. Installation details are discussed later in this paper.

CONFIGURING THE CLUSTER WITH PROPER LOAD BALANCING

Without a proper load-balancing strategy, a cluster that contains a large group of servers cannot serve many client requests. Some servers in the cluster become the hot spots, processing heavy loads of requests, while others remain idle. This is a waste of system resources. A good load-balancing strategy shares client requests equally throughout the cluster, based on the current workload of each server and the current system resources of each machine.

OBJECT SPAWNER

In a SAS Stored Process Server cluster, load balancing is handled by the SAS Object Spawner. SAS Object Spawner is a software component that instantiates SAS servers, including stored process servers. In the case of a single server, the Object Spawner is configured to listen to a certain port for incoming client requests. When it receives a client request to execute a stored process, the Object Spawner starts an instance of the stored process server to fulfill this request. When there is a cluster of servers, each server must have its own Object Spawner installed.

PARENT AND CHILD SPAWNER

Object Spawners communicate with each other via peer-to-peer connections and automatically designate one of them as the "parent" spawner. All the others are "child" spawners. If the current parent spawner is offline, the online spawners communicate again and automatically designate a new parent spawner. The original parent becomes a child spawner even if it is back online. A new spawner that joins the existing cluster is a child spawner by default.

ROUTING CLIENT REQUESTS

When a client requests that a stored process be executed, it can send that request in one of two ways: to the Metadata Server or directly to a Stored Process Server. In either case, the request is first routed to the parent spawner in the cluster. Based on the configured load-balancing algorithm, the parent spawner decides which server will execute this request and forwards the request to the Object Spawner installed on that machine. That spawner then forwards the request to a process on the Stored Process Server to have it executed.

MULTIBRIDGE CONNECTION

By default, each stored process server has three multibridge connections. A multibridge connection is a process that serves multiple client requests. As Figure 2 shows, you can view these multibridge connections in SAS Management Console.

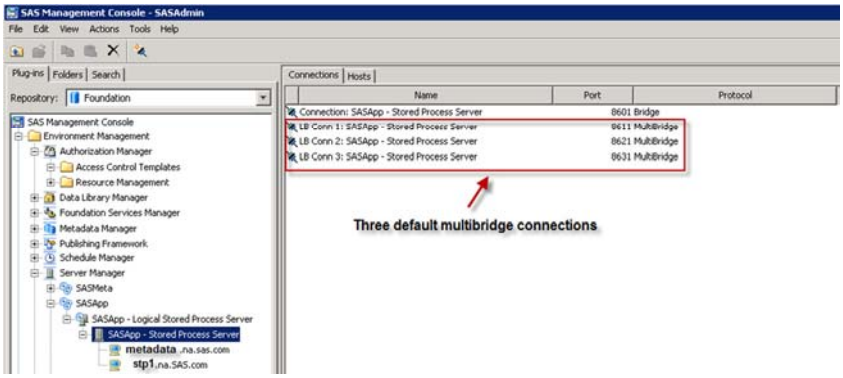

Figure 2. Multibridge Connection

The more multibridge connections a server has, the more client requests it can handle. It is highly recommended that a server have up to four or five multibridge connections per CPU core.

CHOOSING SAS LOAD-BALANCING ALGORITHMS FOR STORED PROCESS SERVER CLUSTERS

SAS 9.3 has three algorithms for conducting load balancing in a SAS stored process server cluster, which are listed in order of the amount of control that they offer: response time, cost, and grid algorithms. These algorithms enable the administrator to control load balancing at different levels.

Response Time Algorithm—Server-Level Control

In the response time algorithm, each Object Spawner maintains a list of available servers in the cluster by communicating with the other servers. It sorts the list based on the current response time of each machine. The list is refreshed at an interval that is specified and controlled by the cluster administrator. When a new client request arrives, the parent spawner redirects the request to the top machine in its list. As shown in Figure 3, you can also decide how often to refresh the response time of each server in the cluster.

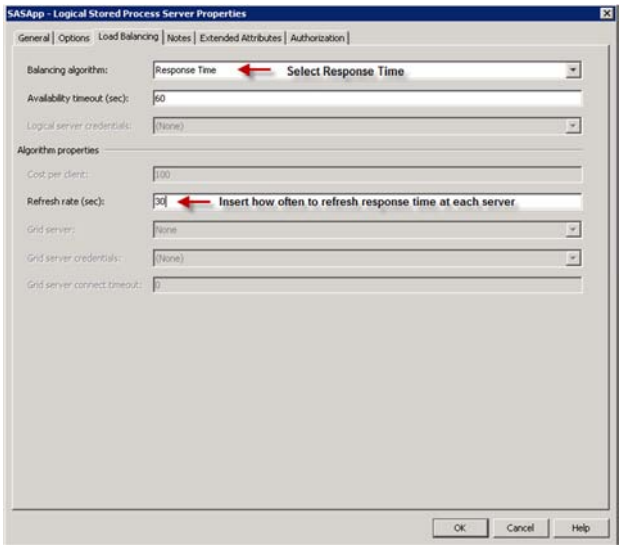

Figure 3. Selecting the Response Time Algorithm

You can further control the maximum number of client requests to be handled by a server, as shown in Figure 4. When this limit has been reached at a server, an incoming client request is not routed from the parent spawner to the server until the server finishes with one of the client requests it is currently handling.

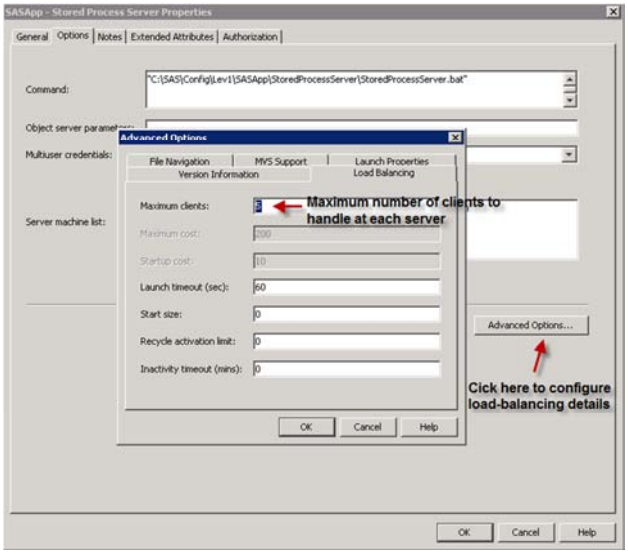

Figure 4. Configuring the Maximum Number of Clients for the Response Time Algorithm

Cost Algorithm—Process-Level Control

The basic idea of the cost algorithm is to charge a fee for responding to each client request; a cost value is assigned to a server by the cluster administrator. By maintaining the current cost at each server, the parent spawner always routes a new client request to the server that has the lowest cost. You can configure the cost value as shown in Figure 5.

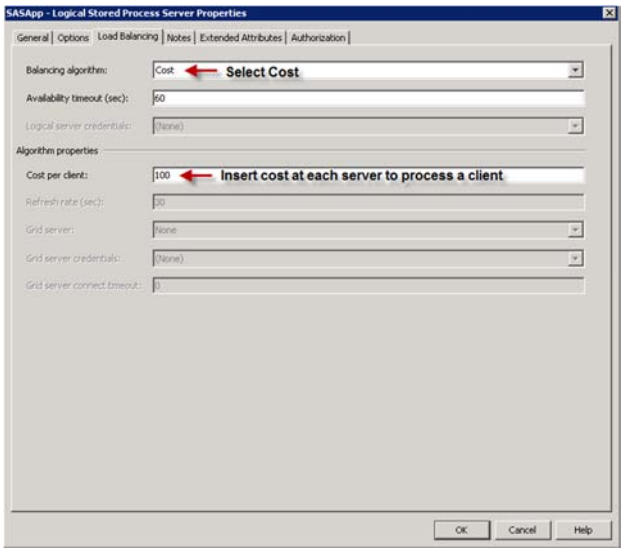

Figure 5. Configuring the Cost Algorithm

To control the maximum number of concurrent client requests a server can handle, you can also define the maximum cost at each server, as shown in Figure 6. This configuration is for each multibridge connection on the Stored Process Server. So if the maximum cost is 500 and the cost per client is 100, a multibridge connection can serve at most five client requests.

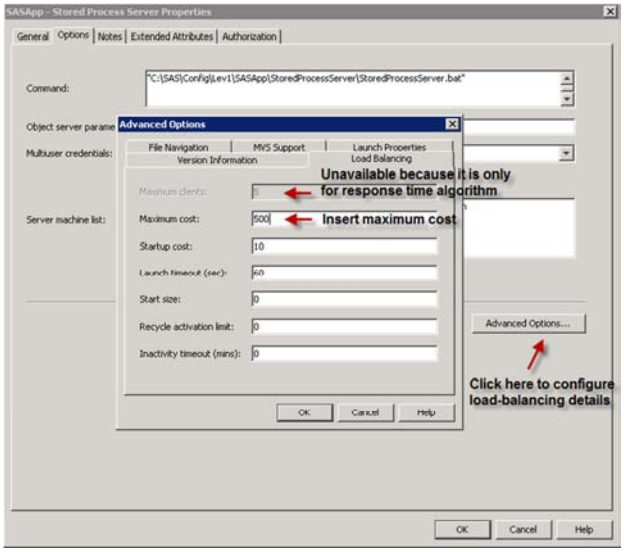

Figure 6. Configuring Maximum Cost for the Cost Algorithm

Grid Algorithm—Control Based on CPU Load Level

The grid algorithm is designed to solve the following problem, which can arise in the cost algorithm, which itself provides one more degree of control in load balancing compared to the response time algorithm: Different stored processes can consume different amounts of computation resources on a CPU. For example, a multibridge connection might already use 90% of the CPU on a server to execute one client request, when the multibridge connection is configured to serve five client requests using the cost algorithm. When a new request arrives, the multibridge connection accepts the request according to the algorithm even though there are not enough system

resources on the server to handle this request in a timely manner.

The grid algorithm communicates with a SAS® Grid Server to allow load-balancing access to grid-related load information on each server. This information is used by the Object Spawner to find the least-loaded server that will accept the client request. This algorithm requires that a SAS grid server be deployed together with the cluster. Deploying a SAS Grid Server together with the cluster provides a better and more sophisticated way for administrators to conduct load balancing, but it is beyond the scope of this paper.

CONFIGURING DIFFERENT LOAD-BALANCING CONFIGURATIONS IN A CLUSTER

Your choice of load-balancing algorithm affects all the servers in the cluster. Sometimes you might want to fine-tune the configurations, such as to allow different maximum number of clients on different servers, according to the capabilities of the hardware in different machines. You can fine-tune configuration by creating a new SASApp – Stored Process Server definition in SAS Management Console and adding servers that have different configurations to it. An example is shown in Figure 7. In this case, server stp2.na.sas.com is able to have a different configuration from stp1.na.sas.com.

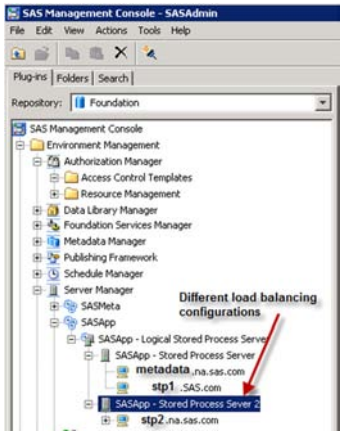

Figure 7. Using Different Load-Balancing Configurations on Different Servers

INSTALLATION DETAILS

The following installation steps are not complete; they highlight the steps that are critical when you create a cluster.

INSTALL THE METADATA SERVER

The first step is to install the Metadata Server. This paper uses `metadata.na.sas.com` as the server's host name. In this step, a stored process server is also installed on the same machine. During the installation process that uses the SAS Deployment Wizard, ensure that you select the following settings:

1. On the **Select Deployment Step and Products to Install** page, from the list that contains all the categories that have been defined in your plan file, select **Metadata Server** as shown in Figure 8.

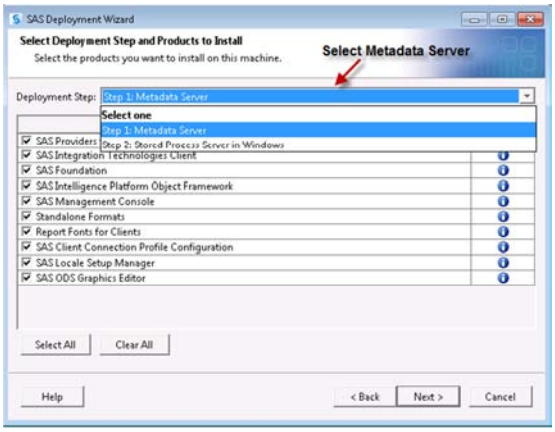

Figure 8. Select the Metadata Server

2. From the **Specify Configuration Information** page shown in Figure 9, record the configuration directory and the configuration level that you need to configure. It is recommended that you keep these configurations consistent throughout the entire cluster.

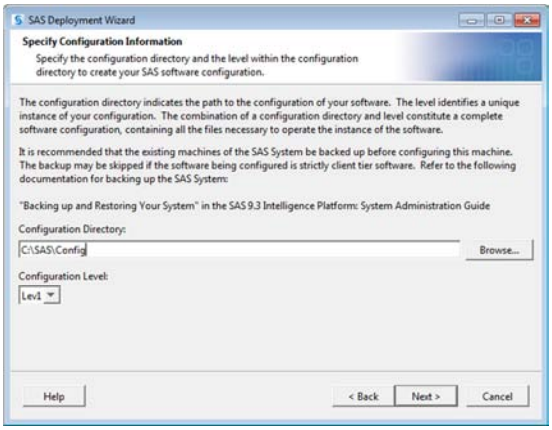

Figure 9. Configuration Directory and Level on the Specify Configuration Information Page

When the installation is done, you will see one SAS Metadata Server and one SAS Stored Process Server in the server list on the SAS Management Console. The next step is to install SAS Stored Process Server on other machines.

INSTALL STORED PROCESS SERVER ON OTHER SERVERS

Assume that the machine where you are installing another stored process server is called **`stp1.na.sas.com`**. During the installation process, ensure that you select the following settings:

1. On the **Select Deployment Step and Products to Install** page, from the list that contains all the categories that have been defined in your plan file, select **Stored Process Server in Windows,** as shown in Figure 10.

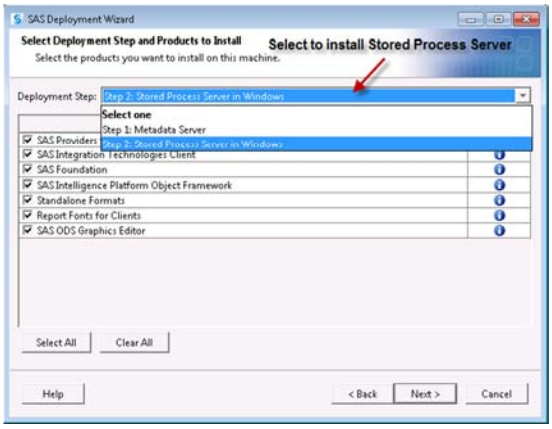

Figure 10. Select the Stored Process Server

2. On the **Specify Configuration Information** page, as shown in Figure 11, be sure to enter the same information for the configuration directory and configuration level as you entered when you installed the Metadata Server. This makes creating the cluster much easier.

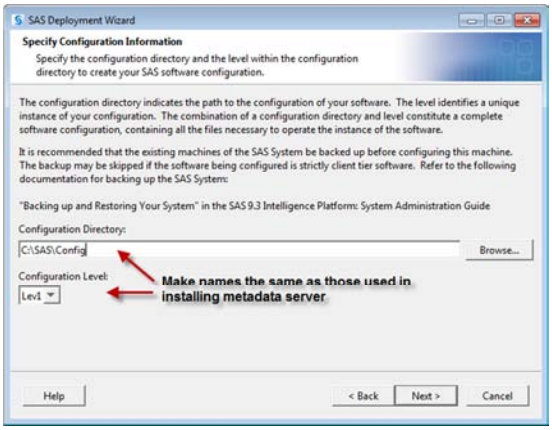

Figure 11. Configuration Directory and Level for Each Individual Stored Process Server

3. On the **Create SAS Application Server Context** page, clear the **Create a SAS Application Server Context** check box, as shown in Figure 12. To form a cluster, you must put all the stored process servers under the same SAS Application Server context. Therefore on the next page, make sure you select the existing **SASApp** application server.

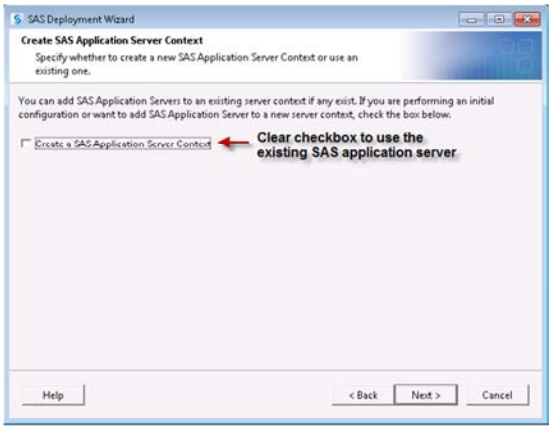

Figure 12. Using the Existing SAS Application Server Created When the Metadata Server Is Installed

4. On the **Specify SAS Stored Process Server information** page, remember to use a new name in the **Logical Stored Process Server Name** field, as shown in Figure 13. In this way, you can avoid the conflict of having the same logical server name as the name you used when installed the Metadata Server. After the installation is finished, you can merge this new Stored Process Server into the existing Logical Stored Process Server that you installed the Metadata Server.

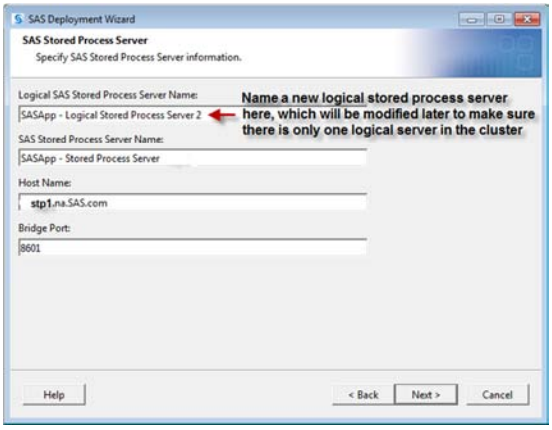

Figure 13. Using a New Logical Stored Process Server Name

During the installation, if you see a message that indicates that you are missing a SAS Application Server content folder on the new machine, the solution is to locate the SAS Application Server content folder in the already-installed SAS Metadata Server (which by default is `C:\SAS\Config\Lev1\SASApp`) and then copy and paste this folder to the same place on the machine where you are installing the Stored Process Server. This automatically copies the SAS Application Server context definition and makes it the same as the definition you already created when you installed SAS Metadata Server.

After the installation is finished, you see an additional Logical Stored Process Server on the SAS Management Console, as shown in Figure 14.

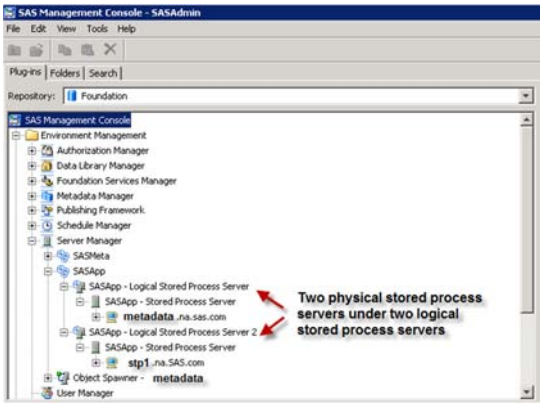

Figure 14 Newly Installed Stored Process Server in SAS Management Console

The next step is to use these two servers to create a cluster.

CREATE THE CLUSTER

To create the cluster:

1. Remove the newly installed Stored Process Server from its Logical Stored Process Server definition. In SAS Management Console, find the Logical Stored Process Server for the newly installed server. On the **Options** tab, remove the newly installed stored process server from the **Selected** list and add it to the **Available** list, as shown in Figure 15.

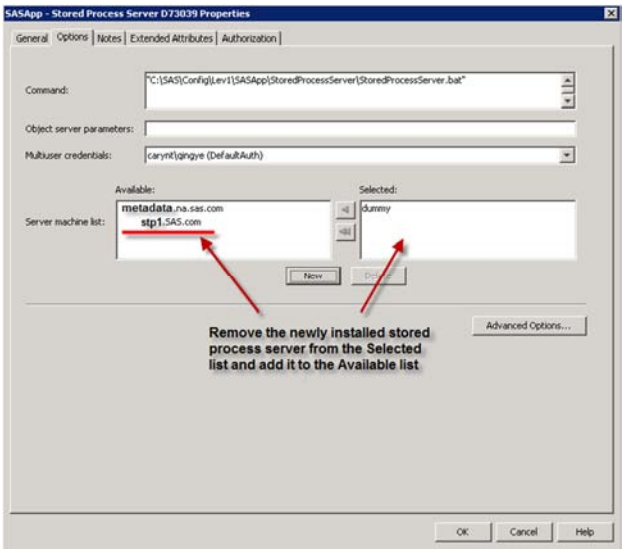

Figure 15. Removing the Newly Installed Server from Its Logical Stored Process Server Definition

2. Add the newly installed Stored Process Server to the existing Logical Stored Process Server that was created when you installed the Metadata Server. In SAS Management Console, find the Logical Stored Process Server installed together with the Metadata Server. On the **Options** tab, add the newly installed server to the **Selected** list, as shown in Figure 16.

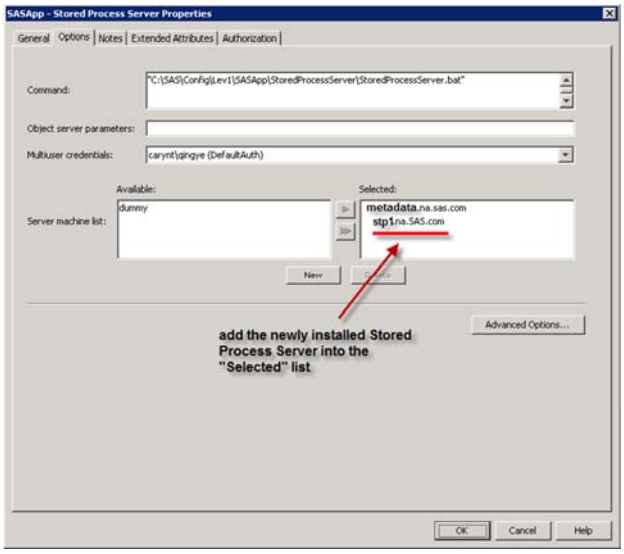

Figure 16. Adding the Newly Installed Server into the Common Logical Stored Process Server Definition

3. Delete the Logical Stored Process Server definition that was created with the newly installed Stored Process Server.

4. Modify the Object Spawner configuration file for the newly installed server. In this example, locate the **`ObjectSpawner.bat`** file in the new server, open the file, and make the modifications shown in Figure 17. Remember to back up this file before you make any changes.

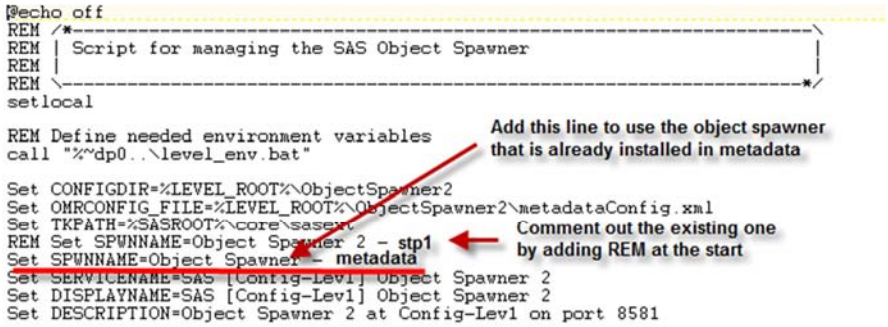

Figure 17. Using the Object Spawner Installed on the Central Metadata Server

5. Add the newly installed server to the spawned server list of the Object Spawner in the Metadata Server. In SAS Management Console, find the Object Spawner that was created together with the Metadata Server. On the **Options** tab, add the newly installed server to the **Selected** list, as shown in Figure 18.

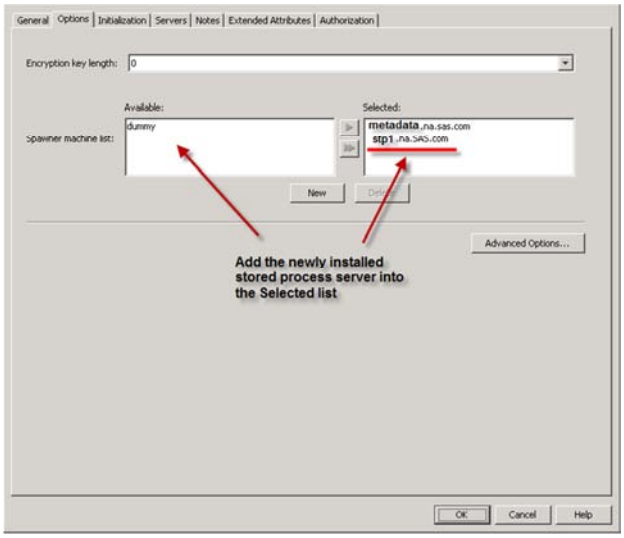

Figure 18. Configuring the Object Spawner of Metadata Server to Spawn the New Server

The final configurations should look like Figure 19 on the SAS Management Console: two stored process servers running on two different physical machines in the same cluster. To add more servers to the cluster, repeat the preceding steps. Whenever you add or remove a server from the cluster, remember to restart all the servers to make sure that every server reads the latest metadata in the cluster.

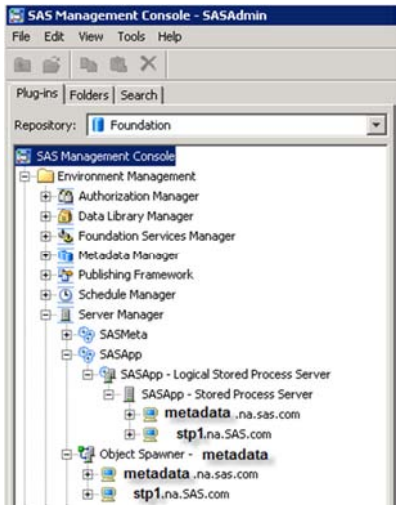

Figure 19. A Cluster with Two Stored Process Servers

CONCLUSION

Clusters created using SAS Stored Process Server can respond quickly and consistently to a large number of client requests. This paper discusses how to create and expand a cluster and how to choose and configure load balancing properly. By providing an easy way to add more servers and to balance the load throughout the whole cluster, SAS Stored Process Server clusters can help software and system architects solve the scalability challenge of rapidly growing numbers of clients and their requests. Solving the scalability challenge enables them to focus more on providing better functionality from their applications for their customers.

ACKNOWLEDGMENTS

The authors acknowledge Sylvia Monaco, Anne Baxter, and Kevin Scott for their contributions to this paper.

CONTACT INFORMATION

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

Qing Ye
SAS Institute Inc.
500 SAS Campus Drive
Cary, NC 27513
(919) 531-1052
qing.ye@sas.com

Richard Nardin
SAS Institute Inc.
500 SAS Campus Drive
Cary, NC 27513
(919) 531-5424
richard.nardin@sas.com

Romon Williams
SAS Institute Inc.
500 SAS Campus Drive
Cary, NC 27513
(919) 531-2605
romon.williams@sas.com