Paper 406-2013

Using SAS® Enterprise Guide®: A System Administrator's Perspective

Haibo Jiang, Joseph Buonomo and Sarah Shiue, Department of Biostatistics, Statistical Programming and Medical Writing, Allergan, Inc., Irvine, CA, USA

Thomas Barbour and Rickie Stevens, R&D Informatics Service, Allergan, Inc., Irvine, CA, USA

ABSTRACT

With the implementation of SAS Intelligence Platform, and the promotion of SAS Enterprise Guide as the main Statistical Programming and Data Analysis tool for Allergan's clinical studies, we are challenged every day with the support of this new environment used by SAS Programmers and Statisticians in our organization. From a technical support point of view, understanding how end users interact with various components of SAS Enterprise Guide, and how such activities are processed and managed behind the scene by SAS Enterprise Guide and SAS Intelligence Platform Servers, is critical to supporting users.

One of the key factors in diagnosing problems and providing resolutions quickly is being able to reproduce incidents and trace them down through system processes. This requires excellent knowledge of the underlying architecture of SAS Intelligence Platform: its various components and how they communicate and work with each other.

This paper shares our experience of supporting SAS Intelligence Platform server and client products with other Platform Administrators. We will focus on SAS Enterprise Guide as a client application on Windows desktop, and SAS Servers (SAS Metadata Server, Object Spawner, and SAS Application Servers) installed on Hewlett-Packard(HP) UNIX machine. The content of this paper will describe technical details related to user and system related activities in the following areas:
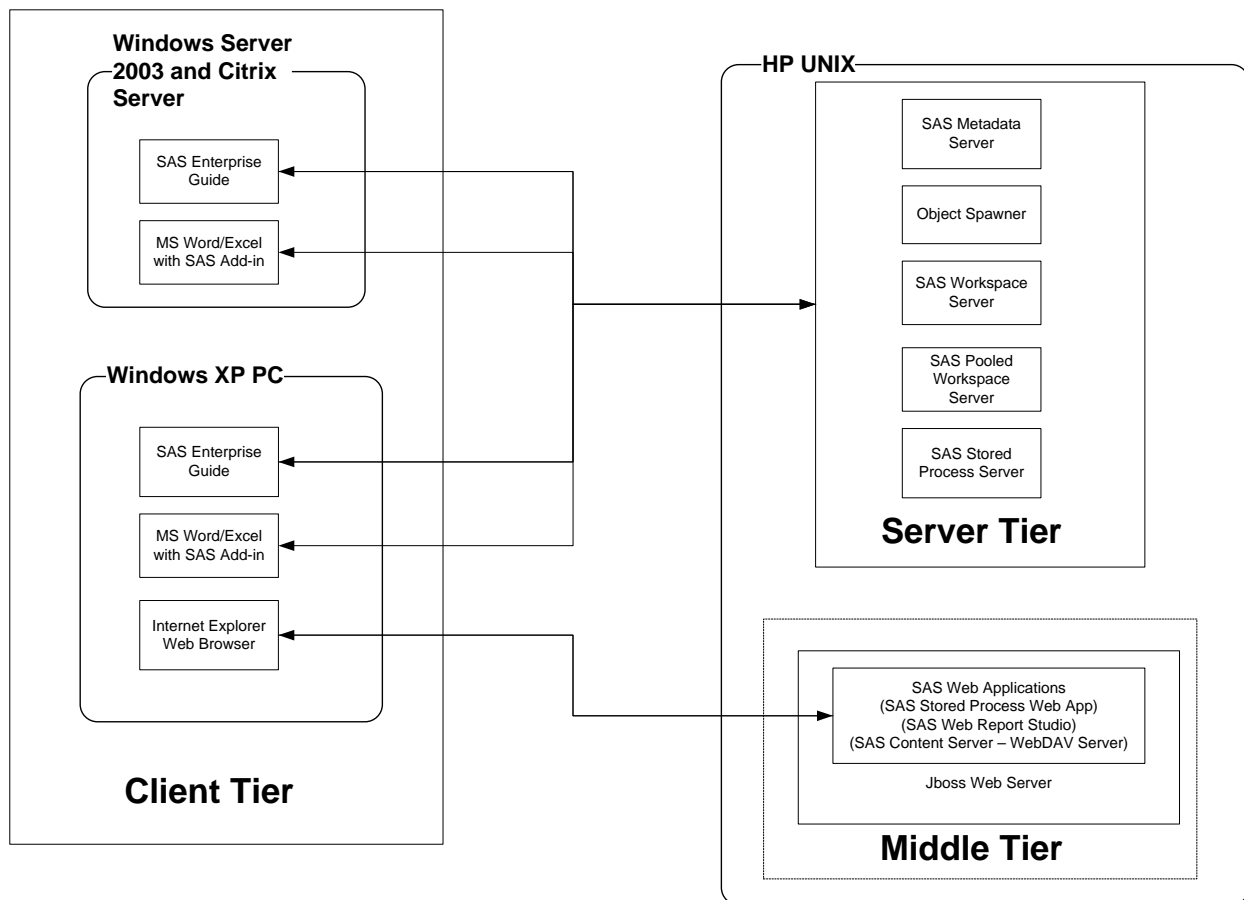- Starting SAS Servers and checking their status
- Connection to SAS Servers, user authorization and authentication
- Initiation of requests from SAS Enterprise Guide, and using SAS Workspace Server and Stored Process Server
- Performance considerations for the processing and presentation of analysis results in SAS Enterprise Guide

INTRODUCTION

Our organization has been using SAS Display Manager and SAS Foundation for many years. Transition to the new SAS Intelligence Platform client and server applications is challenging to both users and technical support staff. In contrast to relatively straightforward model of using SAS Display Manager and SAS Foundation deployed on the same machine, SAS Intelligence Platform is a complex environment, with many products designed to work with each other in a multi-tier architecture. Learning curves at various degrees are expected, and are usually overcome through user training, hand-on exercise and actual working experience . Our goal is that users gain more benefits and become more productive in their daily statistical programming and data analysis work using the new tools in the Platform.

THE ARCHITECTURE AND IMPLEMENTATION

The following is a simplified deployment diagram showing Allergan's implementation and use of key products in SAS 9.2 Intelligence Platform.

```
┌────────────────────────────────────────┐        ┌─HP UNIX──────────────────────────┐
│  Windows Server                         │        │                                  │
│  2003 and Citrix                        │        │   ┌──────────────────────────┐   │
│  Server                                 │        │   │   SAS Metadata           │   │
│    ┌──────────────┐                     │        │   │   Server                 │   │
│    │ SAS Enterprise│◄─────────┐         │        │   └──────────────────────────┘   │
│    │ Guide         │          │         │        │   ┌──────────────────────────┐   │
│    ├──────────────┤          │         │         │   │   Object Spawner         │   │
│    │ MS Word/Excel │◄────────┐│         │──────►  │   └──────────────────────────┘   │
│    │ with SAS Add-in│        ││         │         │   ┌──────────────────────────┐   │
│    └──────────────┘         ││          │        │   │   SAS Workspace          │   │
│                             ││          │        │   │   Server                 │   │
│  Windows XP PC              ││          │        │   └──────────────────────────┘   │
│    ┌──────────────┐         ││          │        │   ┌──────────────────────────┐   │
│    │ SAS Enterprise│◄───────┘│          │        │   │   SAS Pooled             │   │
│    │ Guide         │         │          │        │   │   Workspace              │   │
│    ├──────────────┤         │          │         │   │   Server                 │   │
│    │ MS Word/Excel │◄────────┘          │        │   └──────────────────────────┘   │
│    │ with SAS Add-in│                   │        │   ┌──────────────────────────┐   │
│    ├──────────────┤                    │         │   │   SAS Stored             │   │
│    │ Internet Explorer│◄──────┐        │         │   │   Process Server         │   │
│    │ Web Browser    │         │        │          │   └──────────────────────────┘   │
│    └──────────────┘          │        │          │            Server Tier           │
│                              │        │          └──────────────────────────────────┘
│         Client Tier          │        │          ┌──────────────────────────────────┐
└──────────────────────────────┘        └────────► │   SAS Web Applications           │
                                                    │   (SAS Stored Process Web App)   │
                                                    │   (SAS Web Report Studio)        │
                                                    │   (SAS Content Server – WebDAV)  │
                                                    │      Jboss Web Server            │
                                                    │            Middle Tier           │
                                                    └──────────────────────────────────┘
```

START SAS SERVERS AND CHECKING THEIR STATUS

Overview

On HP UNIX, we use the following scripts to start SAS Servers:

- SAS script sas.servers in <sas-config-dir>/Lev1: Metadata Server, Object Spawner, Share Server, Connect Spawner, Remote Services and Deployment Tester Server
- SAS script SASServer1.sh in <jboss-as>/bin: Jboss and SAS Web Server

The order of starting these servers is very critical. SAS Servers should be started first, followed by Jboss and SAS Web Server. The startup of these servers has to be very clean, meaning we need to completely terminate any old processes before starting/restarting new ones. In practice we often give the system enough time (often in minutes) to clear out old processes. Otherwise, old processes left not cleared will interfere with newly started processes, thus creating problems for SAS Servers and client applications.

SAS Workspace Server and Stored Process Server are not started by these scripts. Their instances are initiated on demand by client connection requests. These server processes are either owned by specific client and persistent until client connection ends (SAS Workspace Server), or shared and waiting for tasks from the same or different clients.

Once started, SAS Servers are running and listening for requests from client applications. Their status can be checked using following methods:

- /etc/services (on UNIX) and Services (on Windows Server)
- sas.servers script (on UNIX)
- netstat (on UNIX[-like] and Windows-NT based operating systems)
- ps command (on UNIX)
- Servers in SAS Management Console: validate (ping host/port and sometimes run a small program to check), and connect

SAS Metadata Server

SAS Intelligence Platform client applications rely on connection profile to communicate with SAS Metadata Server. Metadata Server checks user identity and enforces permissions on metadata objects in client applications. SAS Metadata Server is defined with a host and a port (hpusag26 and 8561 as an example) as shown from SAS Management Console:
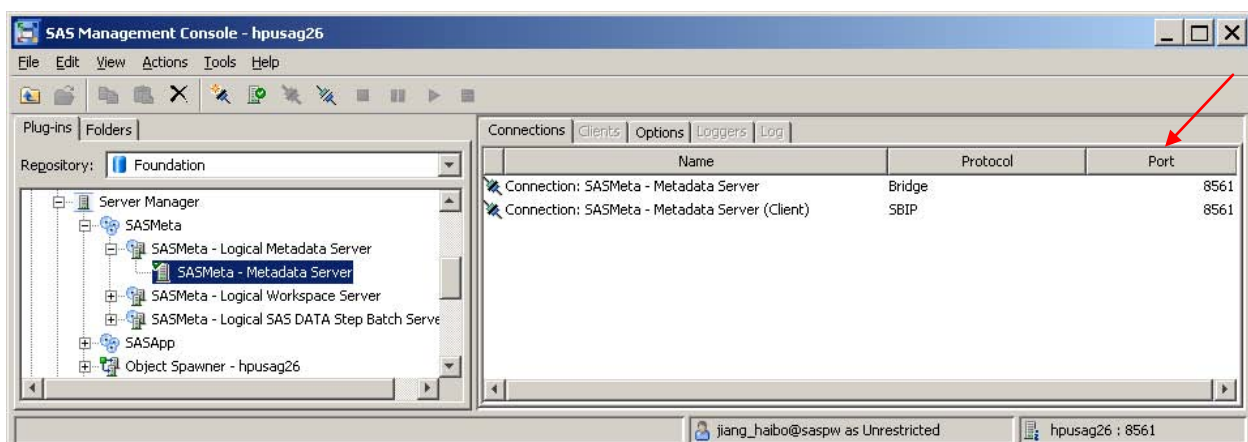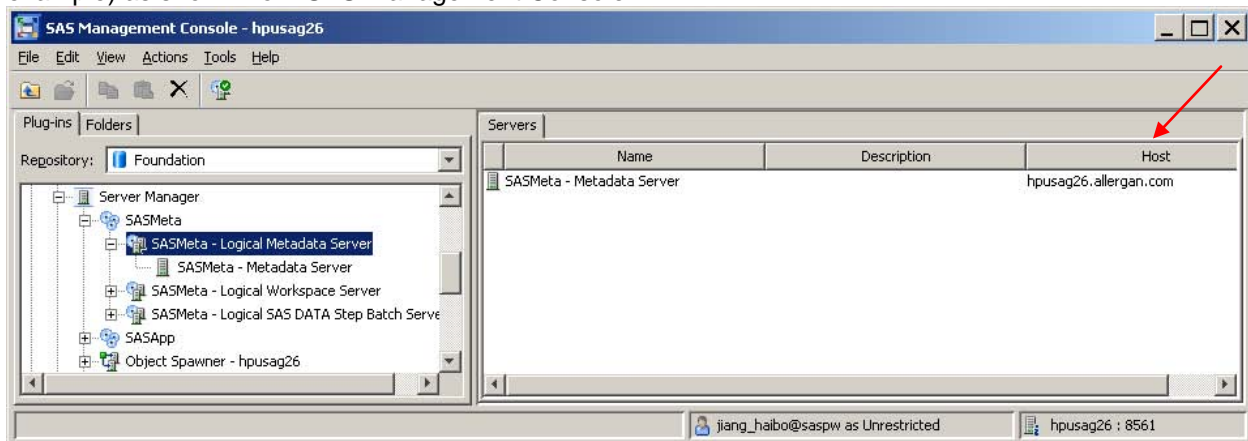
User authentication is performed by SAS Metadata Server using username/password on the host where the server is running (Direct Access in SAS terminology).
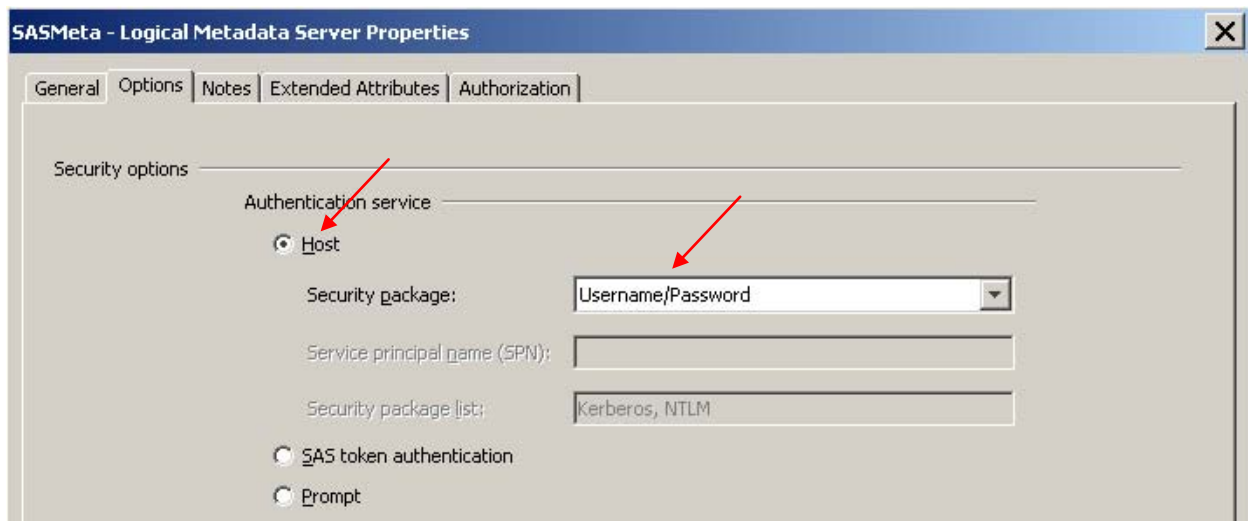
netstat command displays network statistics on UNIX(-like) and Windows-NT based operating systems. The following screen shot shows operating system information of our production HP UNIX machine (using command uname), and the port SAS Metadata Server runs and listens on this machine:

We can also use SAS script sas.servers and UNIX command ps to check the status of SAS Metadata Server and the UNIX process associated with it:

Object Spawner

In SAS Metadata Server, Object Spawner is defined with a host and four ports (hpusag26 and one Operator port 8581, three PortBank ports 8801, 8811, 8821 as an example) as shown from SAS Management Console:
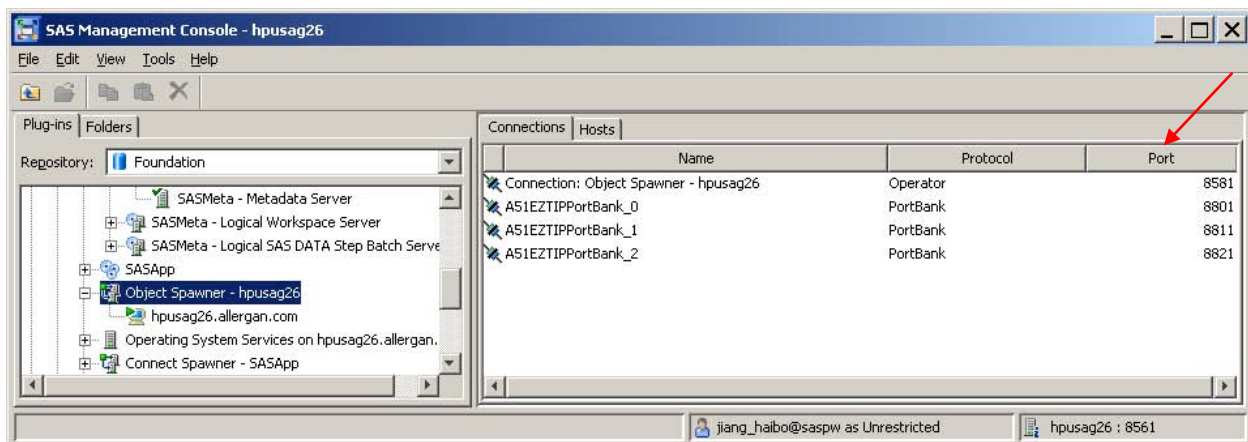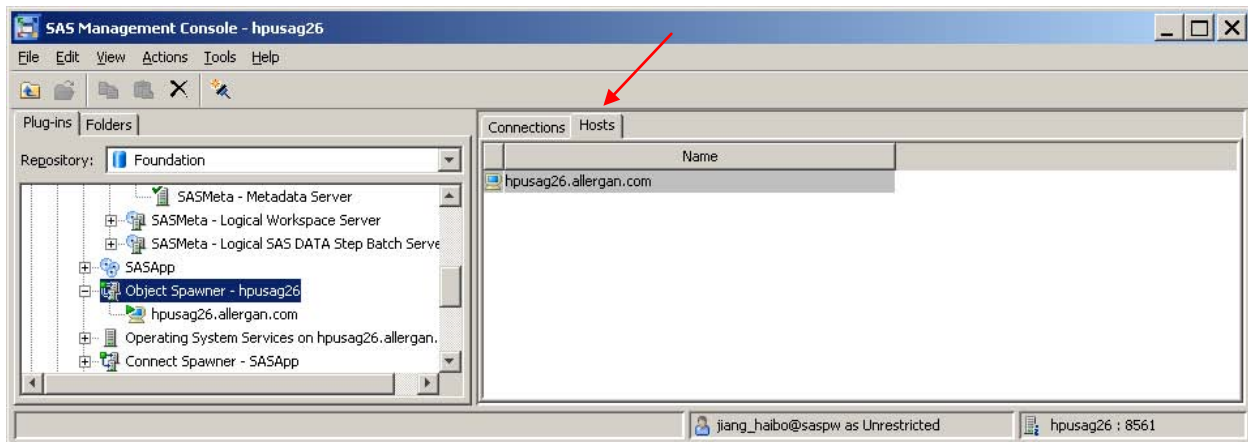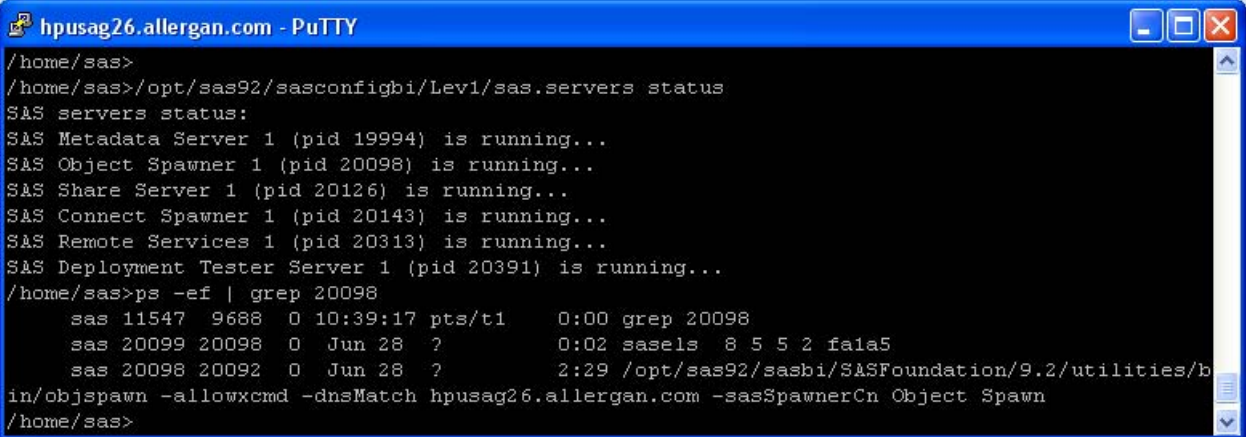
When SAS Intelligence Platform Servers starts, SAS Metadata Server starts first, followed by other SAS Servers and Object Spawner. The steps to start Object Spawner are described below:

1. Object Spawner reads a configuration file metadataConfig.xml (located in Object Spawner directory) containing information of how to access SAS Metadata Server.
2. Object Spawner then connects to SAS Metadata Server and reads configuration information for various SAS Application Servers
3. Object Spawner is up running and listening for requests for various spawner tasks on spawner operator port 8581 (such as client requests to connect to Workspace Server port 8591 and Stored Process Server Bridge port 8601). These tasks are managed by Object Spawner.

Using SAS script sas.servers and UNIX command ps shows the status of Object Spawner and the UNIX process associated with it (parent and child process id):

CREATING CONNECTION PROFILE, USER AUTHENTICATION AND AUTHORIZATION

Connection Profile

Many SAS Intelligent Platform client applications, such as SAS Enterprise Guide, use connection profile to establish connection to SAS Metadata Server, authenticate and authorize users.

A Connection Profile is created in SAS Enterprise Guide with a machine and a port, which shall match what's defined for SAS Metadata Server:

Authentication

1. User selects a Connection Profile and clicks on Set Active button. Connection Profile prompts user to enter their credential: User ID and Password.
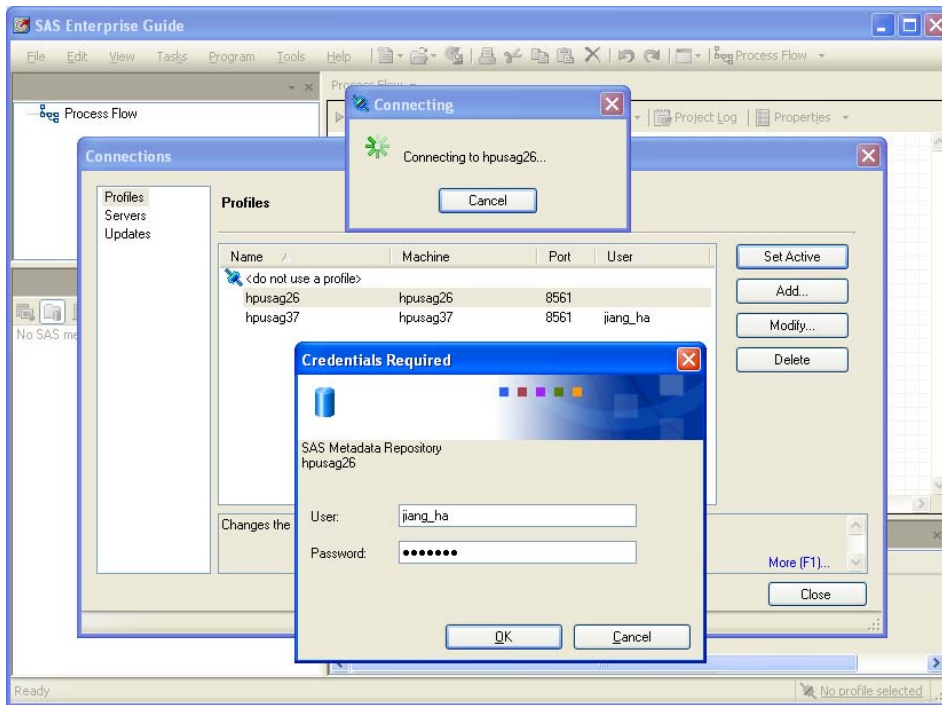
2. SAS Enterprise Guide initiates a connection request and forwards user credential to SAS Metadata Server.

3. SAS Metadata Server receives user credential, and passes it to the host of SAS Metadata Server for authentication.

    If host determines the user has a valid account, host sends User ID to SAS Metadata Server.

    If the account is invalid on host, host sends messages to Metadata Server which then requests SAS Enterprise Guide to re-challenge user. If user cancels or fails to provide a valid credential, no User ID will be returned from host to SAS Metadata Server.

4. SAS Metadata Server receives User ID from host, check the User ID against a list of SAS Users defined in SAS Metadata Server.

    If a matching SAS User ID is found, a connection is then established between SAS Enterprise Guide and SAS Metadata Server.
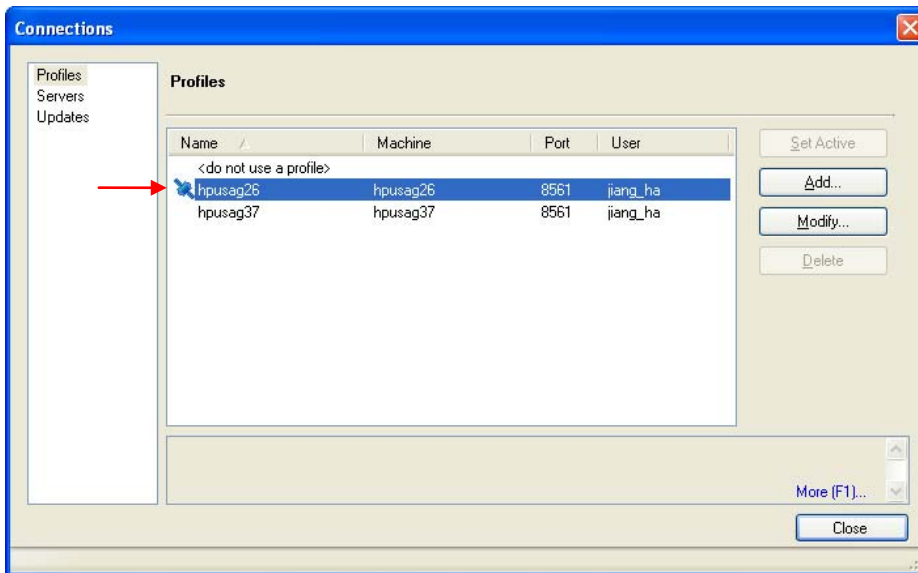
Checking network statistics on UNIX machine shows the connection is established between Widows XP PC (ir01632l) and UNIX machine hpusag26, port 8561 in our example:
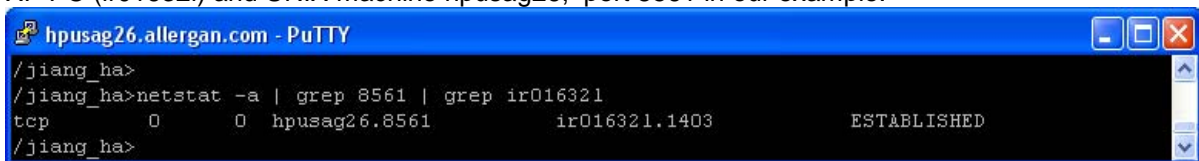
If a matching SAS User ID is not found, SAS Metadata Server sends message back to SAS Enterprise Guide which then notifies user, and will use Public for the default authorization.

Authorization

SAS Enterprise Guide uses the authenticated User ID or Public to find user associated groups, roles and permissions defined in SAS Metadata Server. This enforces metadata level security for metadata objects that are accessible by the user in the client application.

USING SAS WORKSPACE SERVER

SAS Workspace Server

In SAS Metadata Server, Workspace Server is defined with a host and a port (hpusag26 and 8591 as an example) as shown from SAS Management Console:
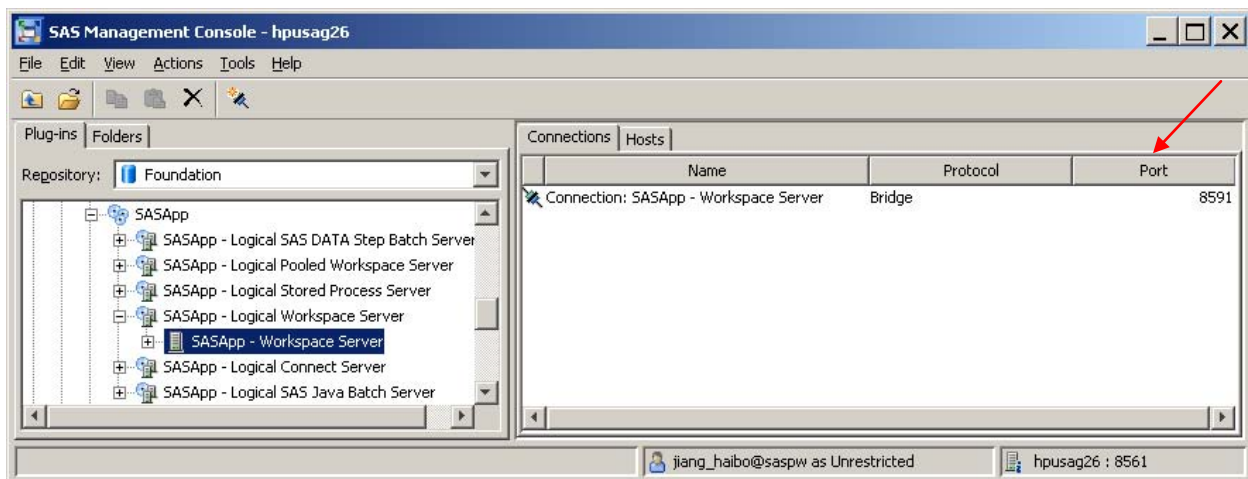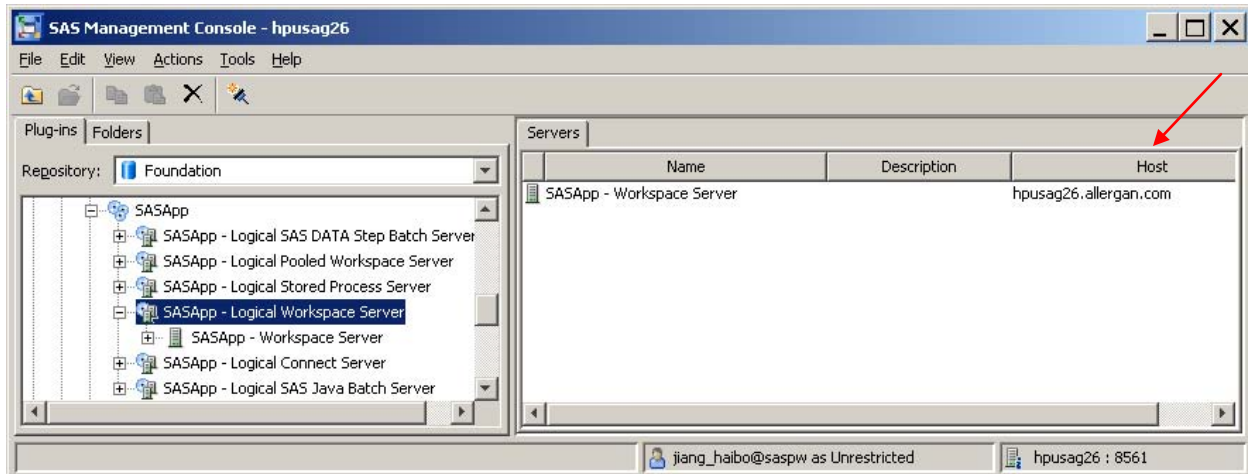
Checking network statistics on UNIX machine shows Workspace Server listens on port 8591:

Security for Workspace Server is set up using host username/password (Direct Access in SAS terminology):
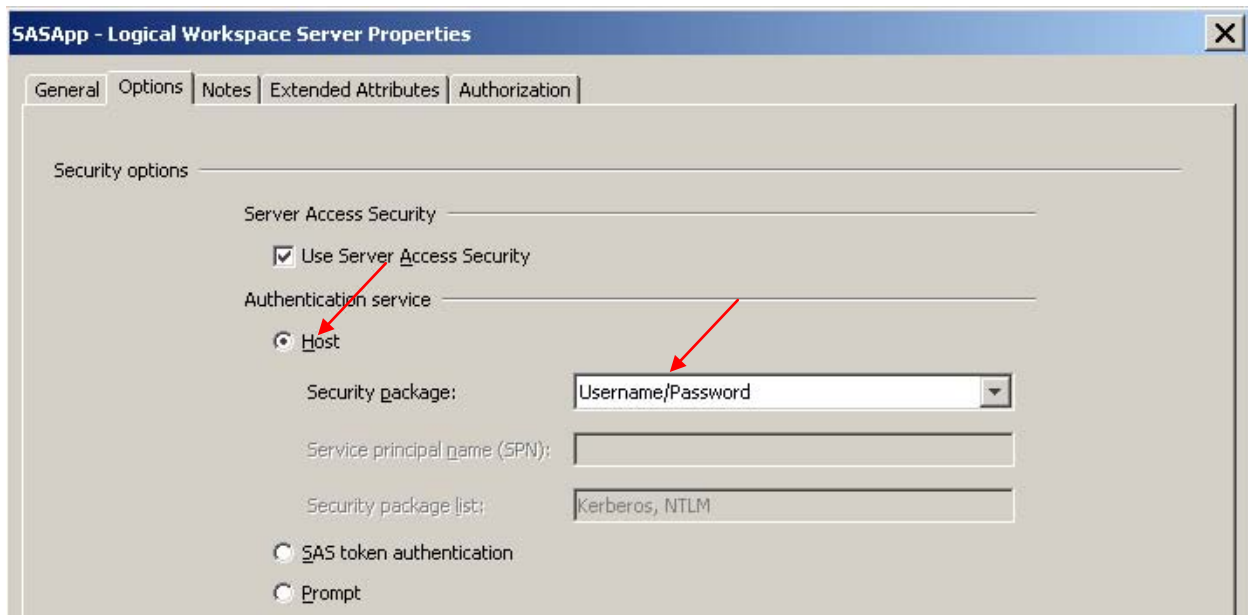
Masking request to use SAS Workspace Server

From SAS Enterprise Guide, there are several places that users can start a request to use SAS Workspace Server. The user request will initiate an instance of SAS Workspace Server if it is not already started for the client application. For example:
1. Submit an program to run on SASApp
2. Use task wizard to create a task and submit it to run on SASApp
3. Connect to SASApp from Resource Pane->Server List->Servers->SASApp or from File->Open->[Program|Data|…]->Servers->SASApp
4. Open certain items (such as the content of SAS tables) from Resource Pane ->SAS Folders->[Sub folders]
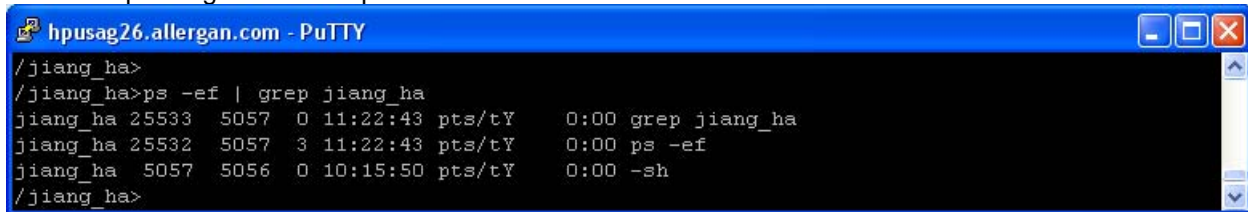
Initiating an instance of SAS Workspace Server

Here are the steps to start an instance of SAS Workspace Server (request initiated from SAS Enterprise Guide as an example):
1. With connection established to SAS Metadata Server, SAS Enterprise Guide will access the information of Object Spawner defined in metadata, and submit to Object Spawner an request of using SAS Workspace Server.
2. Object Spawner receives the request and accepts the connection from SAS Enterprise Guide.
3. Object Spawner authenticates the connection against the host machine of SAS Workspace Server (for stored process and pooled workspace servers, it will be authenticated using a common launch credential by default against SAS Metadata Server).
4. Object Spawner will then launch an instance of SAS Workspace Server under the client user's credential (or a common launch credential by default for stored process and pooled workspace servers), using command information stored in SAS Metadata Server.
5. The instance of Workspace Server is a basically a system process owned by the user on the host machine. Once it is created, it will be associated with/connected to SAS Enterprise Guide client that requests it (client specific), and ready for use.
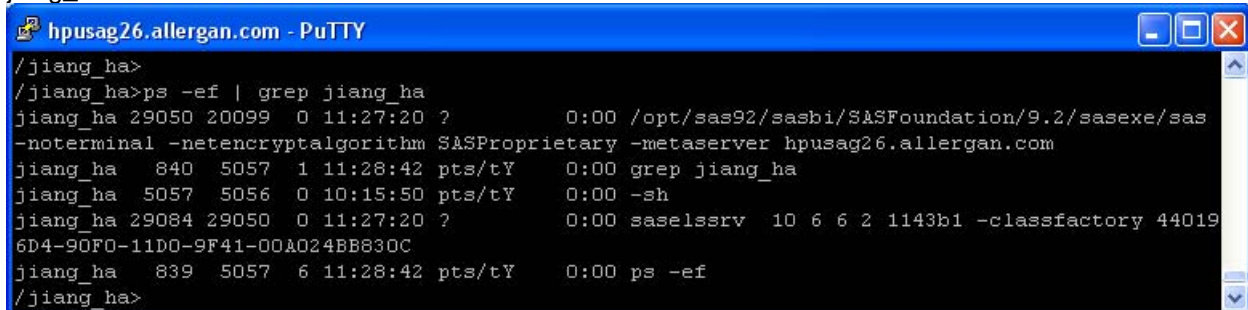
The following screen shots show an example of what happens when using a SAS Workspace Server. There are two instances of SAS Enterprise Guide launched on the same computer (ir01632l), each one has its own connection to its instance of SAS Workspace Server.
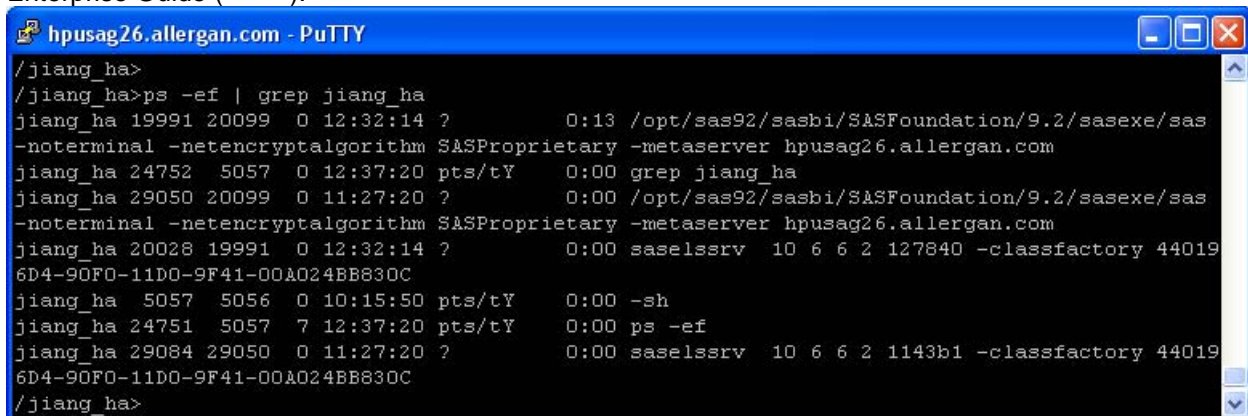
10

Before requesting SAS Workspace Server:

```
hpusag26.allergan.com - PuTTY
/jiang_ha>
/jiang_ha>ps -ef | grep jiang_ha
jiang_ha 25533  5057  0 11:22:43 pts/tY   0:00 grep jiang_ha
jiang_ha 25532  5057  3 11:22:43 pts/tY   0:00 ps -ef
jiang_ha  5057  5056  0 10:15:50 pts/tY   0:00 -sh
/jiang_ha>
```

Launching and connecting to the first instance of SAS Workspace Server from the first SAS Enterprise Guide (⟶)(Also see Object Spawner section of this paper): Object Spawner (PID 20098) -> Object Spawner child process (PID 20099) -> Instance of SAS Workspace Server (PID 29050), owned by user jiang_ha:
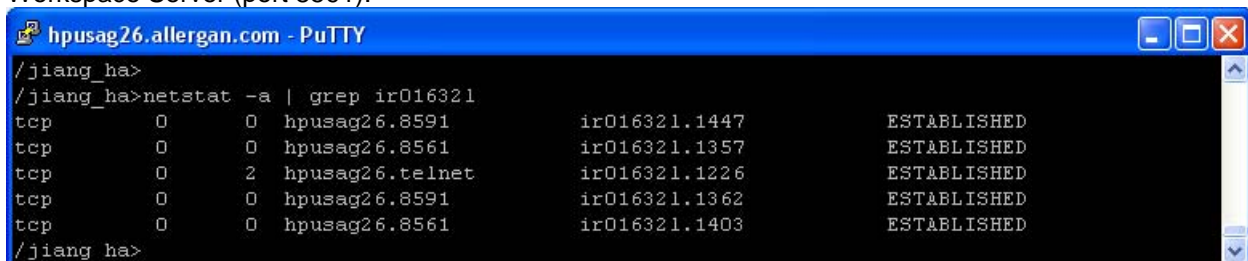
```
hpusag26.allergan.com - PuTTY
/jiang_ha>
/jiang_ha>ps -ef | grep jiang_ha
jiang_ha 29050 20099  0 11:27:20 ?       0:00 /opt/sas92/sasbi/SASFoundation/9.2/sasexe/sas
-noterminal -netencryptalgorithm SASProprietary -metaserver hpusag26.allergan.com
jiang_ha   840  5057  1 11:28:42 pts/tY   0:00 grep jiang_ha
jiang_ha  5057  5056  0 10:15:50 pts/tY   0:00 -sh
jiang_ha 29084 29050  0 11:27:20 ?       0:00 saselssrv  10 6 6 2 1143b1 -classfactory 44019
6D4-90F0-11D0-9F41-00A024BB830C
jiang_ha   839  5057  6 11:28:42 pts/tY   0:00 ps -ef
/jiang_ha>
```

Launching and connecting to the second instance of SAS Workspace Server from the second SAS Enterprise Guide (⟶):

```
hpusag26.allergan.com - PuTTY
/jiang_ha>
/jiang_ha>ps -ef | grep jiang_ha
jiang_ha 19991 20099  0 12:32:14 ?       0:13 /opt/sas92/sasbi/SASFoundation/9.2/sasexe/sas
-noterminal -netencryptalgorithm SASProprietary -metaserver hpusag26.allergan.com
jiang_ha 24752  5057  0 12:37:20 pts/tY   0:00 grep jiang_ha
jiang_ha 29050 20099  0 11:27:20 ?       0:00 /opt/sas92/sasbi/SASFoundation/9.2/sasexe/sas
-noterminal -netencryptalgorithm SASProprietary -metaserver hpusag26.allergan.com
jiang_ha 20028 19991  0 12:32:14 ?       0:00 saselssrv  10 6 6 2 127840 -classfactory 44019
6D4-90F0-11D0-9F41-00A024BB830C
jiang_ha  5057  5056  0 10:15:50 pts/tY   0:00 -sh
jiang_ha 24751  5057  7 12:37:20 pts/tY   0:00 ps -ef
jiang_ha 29084 29050  0 11:27:20 ?       0:00 saselssrv  10 6 6 2 1143b1 -classfactory 44019
6D4-90F0-11D0-9F41-00A024BB830C
/jiang_ha>
```

Checking network statistics on UNIX host machine shows that each SAS Enterprise Guide (⟶ is the first, ⟶ is the second) has a connection to SAS Metadata Server (port 8561) and a connection to SAS Workspace Server (port 8591):

```
hpusag26.allergan.com - PuTTY
/jiang_ha>
/jiang_ha>netstat -a | grep ir016321
tcp       0       0  hpusag26.8591      ir016321.1447      ESTABLISHED
tcp       0       0  hpusag26.8561      ir016321.1357      ESTABLISHED
tcp       0       2  hpusag26.telnet    ir016321.1226      ESTABLISHED
tcp       0       0  hpusag26.8591      ir016321.1362      ESTABLISHED
tcp       0       0  hpusag26.8561      ir016321.1403      ESTABLISHED
/jiang_ha>
```

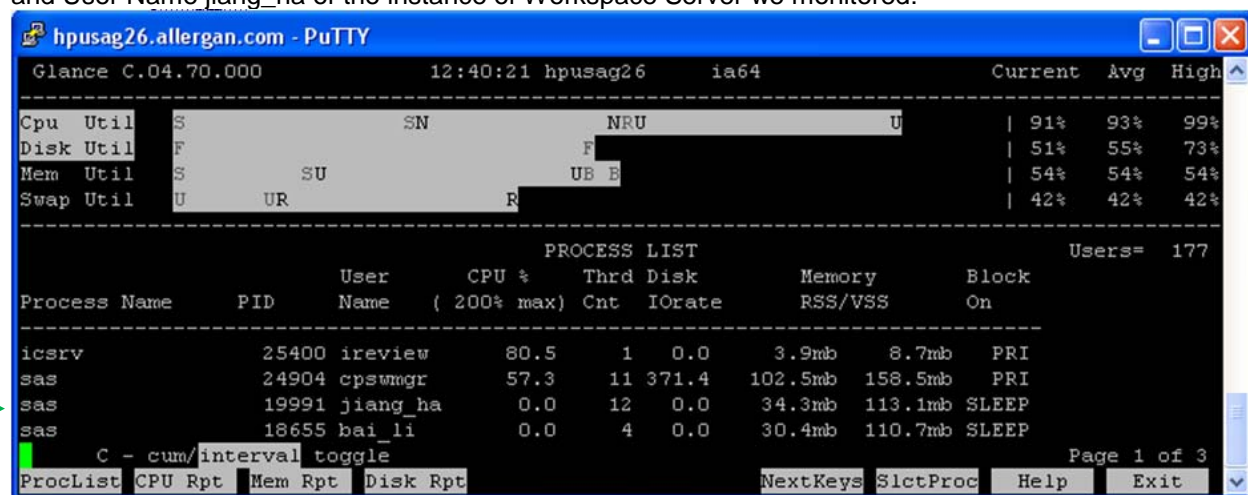**Submitting Tasks/Programs to Workspace Server for execution**

Tasks or programs submitted from SAS Enterprise Guide will usually be executed on an instance of SAS Workspace Server on SASApp. The instance of SAS Workspace Server is a system process, and is associated with SAS Enterprise Guide client until it is ended by disconnecting it from Server or existing the client application. The common use of SAS Enterprise Guide in our organization includes writing programs using enhanced editor, accessing datasets, running program/tasks on SAS Servers, collecting and viewing task status, data, results, log, etc. Understanding what happens behind the scene and knowing what problem areas to trouble shoot incidents is critical to our support of end users.

Basically an instance of SAS Workspace Server is a system process that is
- launched on demand by Object Spawner on behalf of client application
- owned by user who initiated the request
- persistent with the requesting client application session as long as an active connection is maintained
- reused for executing jobs submitted from the same client application session

The following screen shots build upon the example in the previous section. It demonstrates how an instance of SAS Workspace Server executes programs/tasks submitted from SAS Enterprise Guide. We used a monitoring tool called Glance on HP UNIX to view the performance metrics of system processes on host machine.

Before processing any jobs, the UNIX system process associated with an instance of SAS Workspace Server is in SLEEP mode and waiting for work from SAS Enterprise Guide. Please note the PID 19991 and User Name jiang_ha of the instance of Workspace Server we monitored:

Submitting a program from SAS Enterprise Guide to its associated instance of SAS Workspace Server will wake up the UNIX system process, which then runs the program. In the screen shot below, the system process (PID 19991) woke up to execute the task it received (PRI mode under Block On):

Once jobs submitted from client are finished by an instance of Workspace Server, the associated system process goes back to SLEEP mode and waits for new requests.

USING SAS STORED PROCESS SERVER

SAS Stored Process Server

In SAS Metadata Server, Stored Process Server is defined using the following host and ports (hpusag26 and one Bridge port 8601, 3 MultiBridge ports 8611, 8621, 8631) as shown from SAS Management Console:
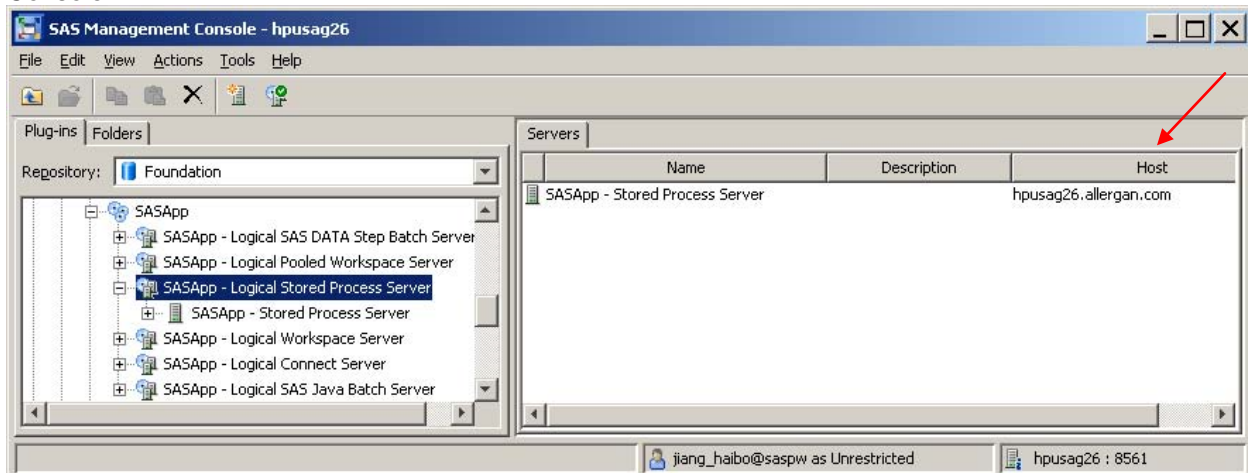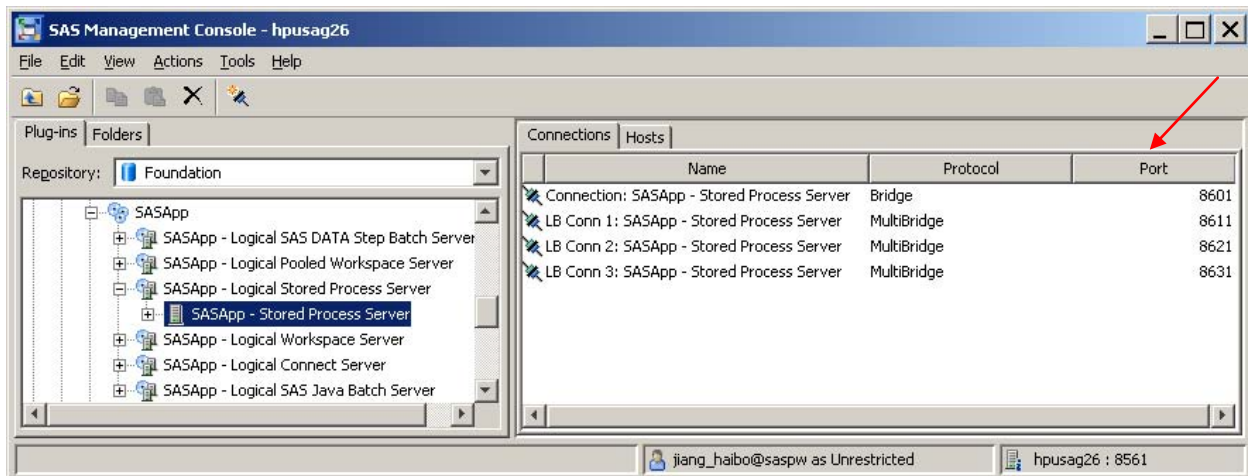
A Bridge connection through port 8601 is used for initial server request. A MultiBridge connection represents a Stored Process Server system process and runs on a specific port (8611 or 8621 or8631). When client application requests a bridge connection for Stored Process Server, Object Spawner determines which system process has the lease load, and redirects the client to the appropriate MultiBridge connection of the Stored Process Server system process selected.

Checking network statistics on UNIX machine shows Stored Process Server listens on ports 8601, 8611, 8621, 8631:

Security option is set up using Server Access Security with a common launch credential (sasserv):
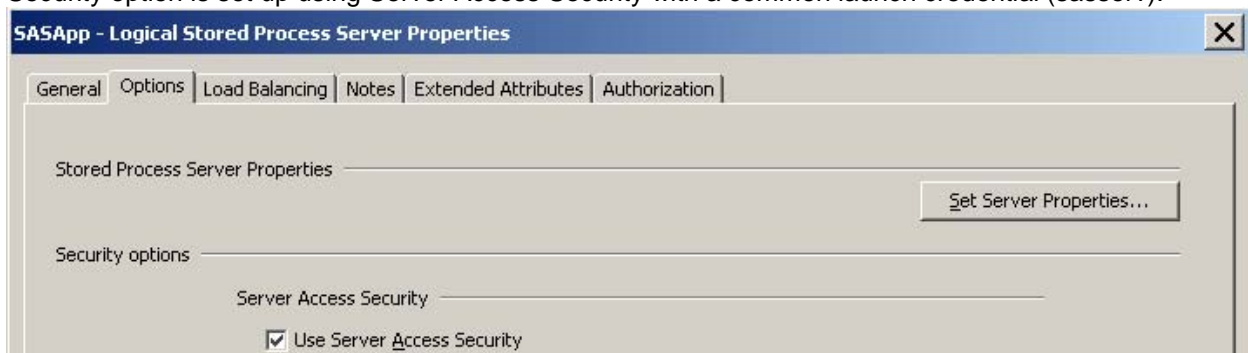
Making request to use SAS Stored Process Server

The use of SAS Stored Process Server is very different from that of SAS Workspace Server by design. From SAS Enterprise Guide or other SAS client applications, user creates or selects a stored process and submit it to run under SASApp, using either SASApp – Logical Workspace Server or SASApp – Logical Stored Process Server, based on which execution server is selected during the creation of the stored process.

Most of the time, stored processes are executed using Stored Process Server. There are certain scenarios (such as computing resource requirement, common launch credential vs. individual user identity, physical files created on file systems and their ownership and access control, etc.), users may decide to select Workspace Server as the execution server for a stored process. If that is the case, the execution of such stored process is no different from running a program/task on Workspace Server.

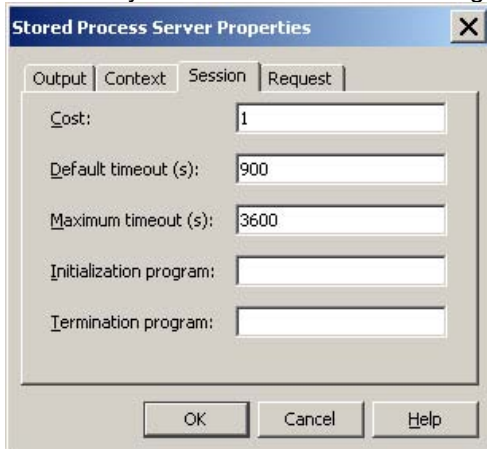**Initiating an instance of SAS Stored Process Server**

Here are the steps to start an instance of SAS Stored Process Server(request initiated from SAS Enterprise Guide as an example):
1. With connection established to SAS Metadata Server, SAS Enterprise Guide will access the information of Object Spawner and submit to Object Spawner an request of using SAS Stored Process Server.
2. Object Spawner receives the request and accepts the connection from SAS Enterprise Guide.
3. Object Spawner authenticates the connection using the common launch credential by default against Metadata Server.
4. Object Spawner then launches an instance of SAS Stored Process Server under the common launch credential, using command information stored in SAS Metadata Server.
5. The instance of Stored process Server is a basically a system process on the host machine. Once it is created, it is owned by common launch credential and ready for shared use by initial requesting client application/user and others submitting stored processes for executions.
6. Object Spawner uses its operator connection to perform load balance of stored process jobs among these system processes:
   - The maximum number of Stored Process Server instances/systems processes are 3 by default. They run on 3 different ports as defined in SAS Metadata Server for Stored Process Server.
   - Each system process can support up to 4 internal, independent stored process sessions which execute stored processes. The max number of concurrent stored process sessions is 3x4=12.
   - System processes and stored process sessions within a system process competes resources allocated to Stored Processes Server on the host machine. The performance of executing a specific stored process depends on the resource usage of other stored processes running on the same time.
7. Load Balancing for Stored Process Server
   Stored Process Server is designed to use load balancing by default. A few parameters are made available in SAS Management Console and can be configured by Platform Administrator.
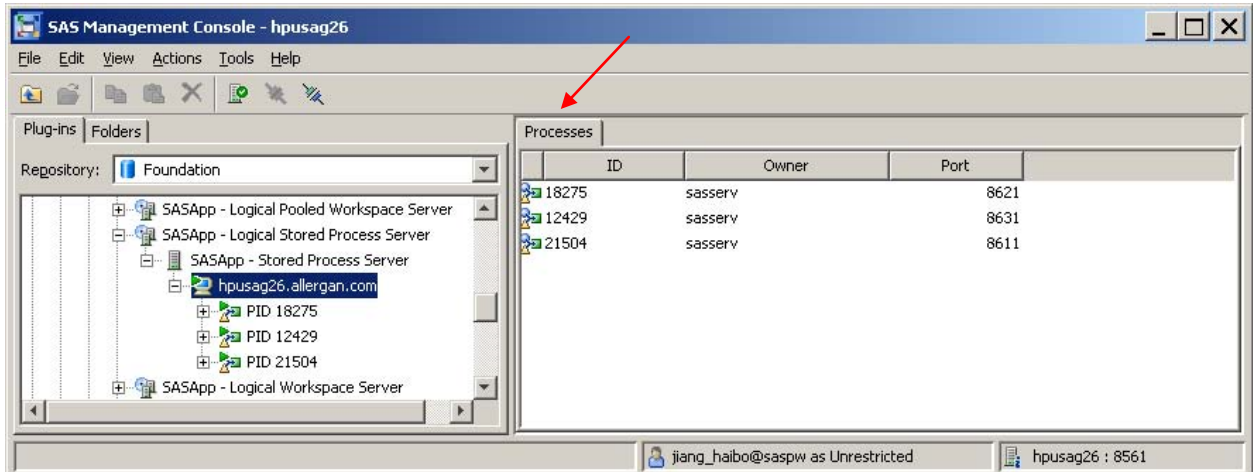
   On Load Balancing window under SASApp – Logical Stored Process Server properties, available timeout for Stored Process Server is set to 60 seconds. This means if Stored Process Server reaches its load limit (by number of instances and sessions), Object Spawner cannot launch new instance of Stored Process Server, and user must wait until there is a server available or it fails after 60 seconds.
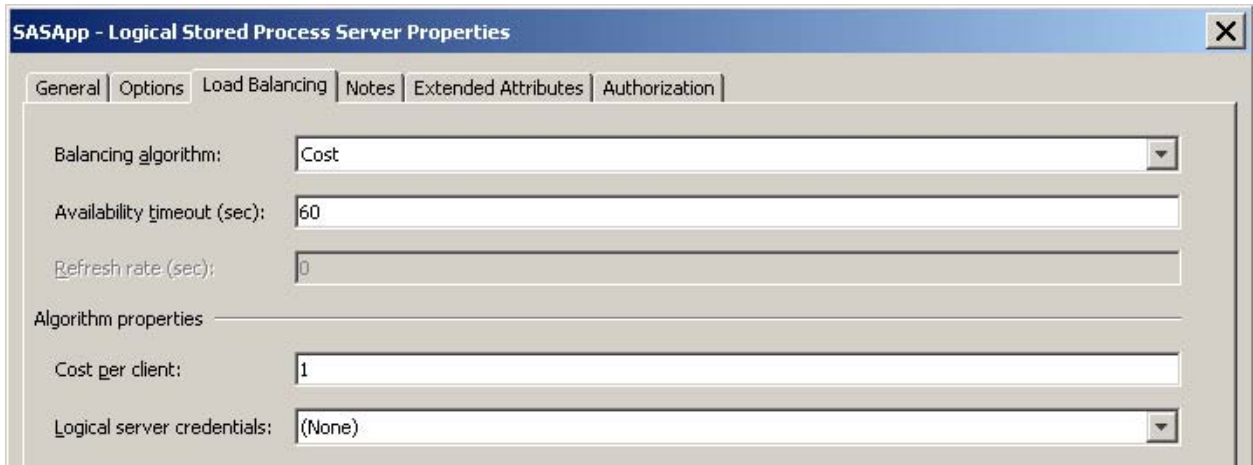
Stored process sessions within a system process expire after a period of inactivity. The default expiration time is 900 seconds but it can be increased to the maximum timeout, which is set to 3600 seconds by default and cannot be changed.

The system process for an instance of Stored Process Server can be viewed from SAS Management Console:

More details can be seen using ps command on UNIX:

16

SUBMITTING STORED PROCESSES TO STORED PROCESS SERVER FOR EXECUTION

A specific stored process is executed on an instance of Stored Process Server (a system process). The corresponding session within the system process can be viewed from Sessions tab in SAS Management Console. Session is usually very short (less than a few seconds). It will appear on the window and then be cleared once its work is finished.
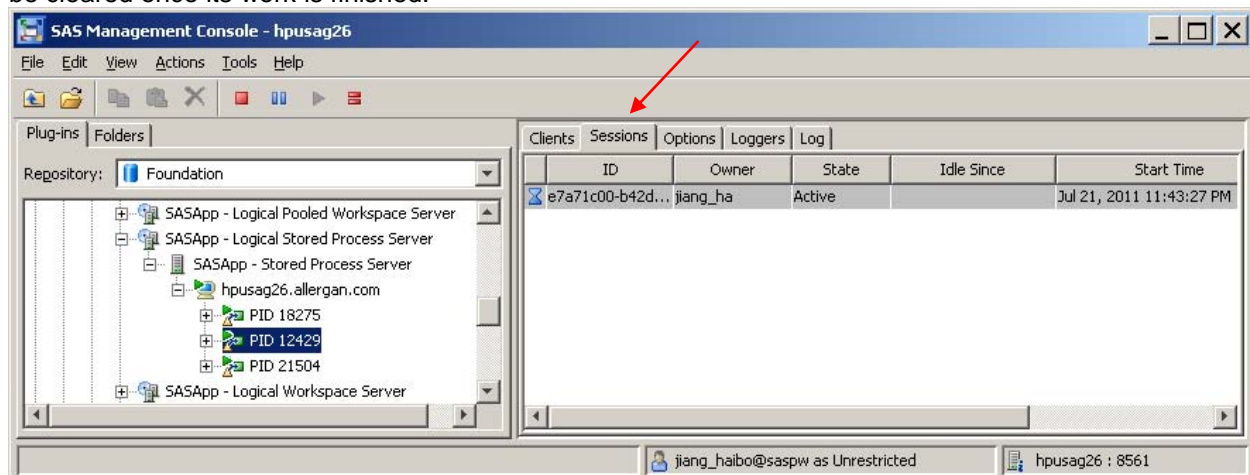
We used HP UNIX tool Glance to monitor system processes associated with the instances of Stored Process Server. Before processing any jobs, these systems processes are in SLEEP mode and waiting to run specific stored processes. In the example below, please note PID 12429 and User Name sasserv of the instance of Stored Process Server we monitored:

System processes representing the instances of Stored Process Server are monitored and load balanced by Object Spawner. When receiving a request to execute a stored process from a client application, Object Spawner will choose a system process and forward the request over for execution. In the following screen shot, the system process (PID 12429) was chosen and woke up to execute the stored process (PRI mode under Block On):

COLLECTING AND DISPLAYING ANALYSIS RESULTS IN SAS ENTERPRISE GUIDE

To SAS programmers and Statisticians who have been using SAS Display Manager for many years, SAS Enterprise Guide presents a new way of working and at the same time introduces challenges during initial transition period. What they experience interacting with the tool (user does something and the system

responds in its way by design) are often different from what they are used to see using SAS Display Manager.

The heavy-duty execution of a program/task/stored process is normally done by SAS Application Servers on the host machine, which usually has sufficient computing power. However, there is still a good portion of work performed by SAS Enterprise Guide on client machine, especially presenting results to end users (showing SAS Results, viewing datasets, displaying logs, creating listings, plotting graphs, etc.). The computing resource available to SAS Enterprise Guide is also critical to the satisfactory use of the tool. Changing some of the setting in Tools->Options will usually be the first area to try to improve user experience.

Performance is often examined and tuned on server side. For SAS Enterprise Guide to efficiently work with SAS Servers, client side performance is also very important and shall be examined. Here is a list of considerations based on our experience:

- SAS Application Servers captures tasks/programs execution information, and communicates them to SAS Enterprise Guide during (for task status) and after the execution. Sometimes a task is stuck and not progressing at all in SAS Application Servers for many reasons. In this case, what we described in early sections of this paper will help to diagnose the problem.

- Programs, Tasks and Stored Processes executed under SAS Applications Servers can include program code that specifically write output files to file systems on OS. Listings and graphs displayed in SAS Results tab in SAS Enterprise Guide may not have the same layout/format as listings and graphs saved as files on OS. Changing the format of certain results could improve performance.SAS Enterprise Guide requires different amount of computing resource on client machine to display different formats of Result and components within Result. Sometimes this is trivial and user does not notice any performance difference at all. However in some cases, the difference could be very large, especially for graph format.

- The amount of Random Access Memory (RAM) on the machine where SAS Enterprise Guide runs significantly affects user experience. As displayed in Performance tab in Windows Task Manager window, the following shall be monitored closely:

  □ Total, Available and System Cache Physical Memory (RAM)

  □ Total, Limit and Peak Commit Charge

  □ Page File Usage (Virtual Memory)

  If memory usage exceeds RAM and has to be extended to physical disk, applications in general will run much slower, or not perform at all in severe cases (frozen window or task status not changing using SAS EG). The rule of thumb is to have sufficient RAM and minimize the size of page file: the performance will be faster, the system cache will be increased and the hard drive won't be wasting space on virtual memory. Windows System Administrator should be able to provide help in this area to improve the performance of applications.
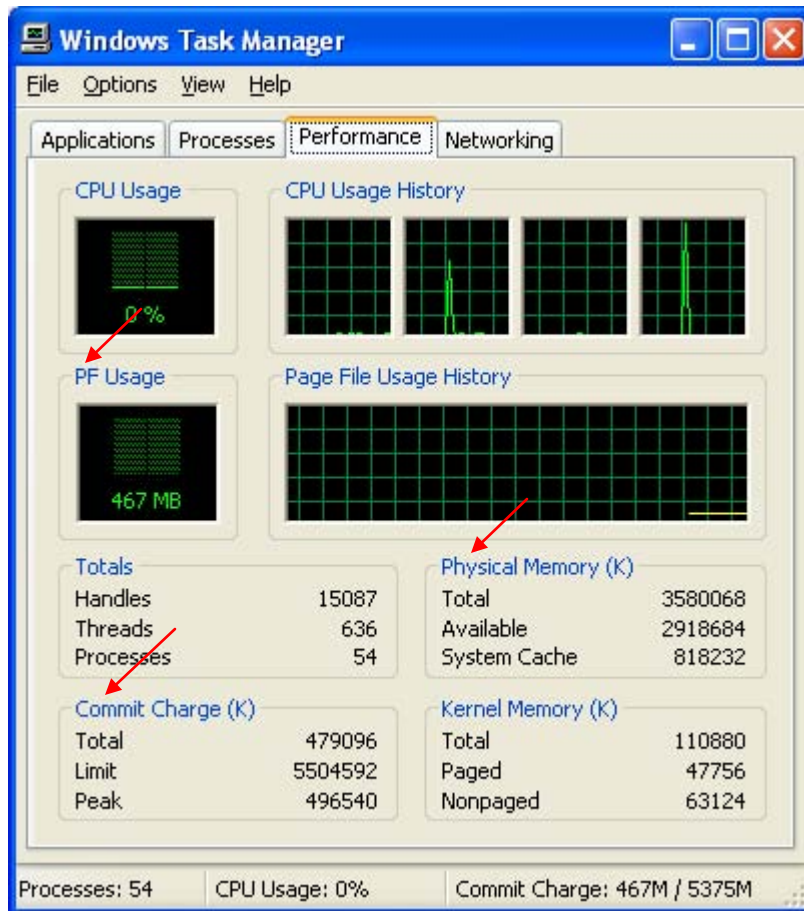
- Information sent back by SAS Application Servers is gathered by SAS Enterprise Guide and displayed in different tabs/windows as Results, Output Data, Log, etc. Depending on how programs are written and the settings in Tools->Options, users may see more or less tabs/windows shown up as results, and may receive pop-up warning or error windows (in addition to those in program logs) related to the following:

    □ Number of temporary datasets: The number of temporary dataset in WORK library can become very large since the same SAS session (started by the associated instance of SAS Workspace Server) is used for all program runs within the same SAS Enterprise Guide. The max number is 50 by default and can be set by user. User can remove temporary datasets manually or by executing SAS statement such as proc datasets kill.

    □ Size of a dataset: A dataset can be very large for SAS Enterprise Guide to effectively work with.

    □ Size of log: The log can become too large for SAS Enterprise Guide and new information will not be logged when running a program in a project. This can happen if project log is turned on and log is piled up. User can either turn off or clear project log.

    □ Size of SAS result: SAS result can become very large. The default is 5MB for SAS Enterprise Guide to display and can be set by user.

    □ Amount of data transferred between SAS Servers and SAS Enterprise Guide, network speed and computing resource on both server and client machine. These affect end user experience of "performance" – how fast they can see results in SAS Enterprise Guide after they submit a

task/program to SAS Servers to run. In addition to time spent by SAS Servers to process a task, time is also needed in the last step called Collecting in SAS Enterprise Guide – a step to gather analysis results from SAS Application Server and use local computing resource to process and display them.

CONCLUSION

This paper presents our experience of supporting the use of SAS Enterprise Guide and SAS Servers for statistical programming and data analysis work in our organization. It provides some technical insights of how client and server applications work together in SAS 9.2 Intelligence Platform. We hope this helps System Administrators to trouble shoot problems, monitor systems and tune their performance, and support more happy users of this Platform.

REFERENCES

Talley, B. and Kalich, T. 2011. Troubleshooting Tips and Techniques for a SAS® 9.2 Business Intelligence Environment. *Proceedings of the SAS Global Forum 2011 Conference*. Cary, NC: SAS Institute Inc.

Schneider, M., Bennett, D. and Robison, C. 2011. Understanding the Anatomy of a SAS® Deployment: What's in My Server Soup? . *Proceedings of the SAS Global Forum 2011 Conference*. Cary, NC: SAS Institute Inc.

Doninger, C. and Spruill, J. 2008. Balancing the Load - SAS® Server Technologies for Scalability. *Proceedings of the SAS Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc.

SAS® 9.2 Intelligence Platform: Application Server Administration Guide. 2010. Cary, NC: SAS Institute Inc.

SAS® Platform Administration: Fast Track. 2010. Cary, NC: SAS Institute Inc.

RECOMMENDED READING

SAS® 9.2 Intelligence Platform: Security Administration Guide. 2011. Cary, NC: SAS Institute Inc.

SAS® 9.2 Intelligence Platform: System Administration Guide. Second Edition. 2011. Cary, NC: SAS Institute Inc.

SAS® 9.2 Intelligence Platform: Desktop Application Administration Guide. 2010. Cary, NC: SAS Institute Inc.

SAS® 9.2 Intelligence Platform: Installation and Configuration Guide. 2011. Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

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

Haibo Jiang
Allergan, Inc.
2525 Dupont Dr.
Irvine, CA 92620 USA
Phone: 714-246-5169
E-mail: jiang_haibo@allergan.com

Joseph Buonomo
Allergan, Inc.
2525 Dupont Dr.
Irvine, CA 92620 USA
Phone: 714-246-5786
E-mail: buonomo_joseph@allergan.com