Paper SAS3350-2019

Smart SAS® Environment Manager Agent for SAS® Grid Manager with a Shared Configuration Directory

Zhiyong Li, SAS Institute Inc.

ABSTRACT

A question frequently asked by SAS customers is how to run SAS® Environment Manager in a grid (or a shared configuration directory) environment. As of today, SAS has over seventy customer tickets that are related to this issue. You request, we listen. In SAS® 9.4M6, we re-designed the SAS Environment Manager agent to make it "smart": it detects whether the request to run the SAS Environment Manager agent is from a new host or a new grid node. If the request is from a new host, it automatically creates a new SAS Environment Manager agent instance for that host. With this design, we also introduced many capabilities to manage the smart agent to make it easier for customers. The new capabilities include support for update in place, hot fixes, other platforms, and all SAS® Deployment Manager tasks. This paper details the design, the implementation, and the process for how you can take advantage of this new smart agent feature. As a bonus, we also review what is new in the SAS® 9.4 middle-tier platform.

INTRODUCTION

In a SAS Grid environment, customers usually have a large number of grid nodes. On one hand, customers want to use SAS Environment Manager to monitor these nodes, but on the other hand, they don't want to run the SAS® Deployment Wizard just to install the SAS Environment Manager agent on each node. To address this challenge, SAS Technical Support has previously recommended the SAS technical paper *Installing and Configuring SAS Environment Manager in a SAS Grid Environment*. The paper includes a script that customers can use to generate multiple copies of SAS Environment Manager agents. Each agent can then be used by a grid node. However, there are many drawbacks of using this script, such as the following:

- It only supports Linux.
- It creates a fixed number of agents.
- It is not managed by the SAS deployment tools, so it might break if you update your environment through SAS Deployment Manager or if you apply hot fixes.

This paper introduces a new smart agent approach that can create and manage agents with the following advantages:

- automatically created without the need to pre-prepare additional SAS Environment Manager agents
- dynamic and on-demand, so there are no constraints due to a pre-defined number of agent
- created and managed by SAS Environment Manager, so no additional work is required from end-users
- works for all environments that use a shared directory
- supports update-in-place to upgrade all agent instances
- supports hot fixes to update all agent instances
- supports all SAS Deployment Manager functions, such as changing password
- supports all six platforms that SAS supports

This paper details the design, implementation, and use of this new smart agent feature. Deploying and using the agent is just the beginning of the process of monitoring and managing your grid environment.

This paper also shows you what you can monitor and manage with the help of smart agent in a grid environment, including live-time management and offline reports.

As the end of the paper, we will also briefly review what is new in SAS 9.4m6 middle tier platform.

THE DESIGN

This is the principal idea of the design: instead of providing a script to either create a list of agents after the SAS Deployment Wizard completes or pre-create a list of agents as part of the SAS Deployment Wizard process, we want to make the agent script be aware whether an agent instance for a specific host has been created.

In other words, we want to make the agent smart so that it doesn't simply start an agent instance, but that it's also aware of the existence of agent instances for other hosts. If an agent-related command (such as `start agent`) finds out that the command is for a host other than the primary agent machine, it performs these tasks:

- If the agent instance for the other host doesn't exist, it either creates a new instance or clones an agent, then starts the newly created agent instance.
- If the agent instance for the other host does exist, it starts the agent instance for the host.

The primary agent machine is defined as the machine on which the SAS Deployment Wizard has been run in order to install and configure an agent. In a multiple machine deployment, the SAS Deployment Wizard could be run on multiple machines, and the agent can be installed on all of these machines. Each of these agents can be a primary agent, and each one is used as a smart agent for the other machines that don't run the SAS Deployment Wizard. In a later section of this paper, we will discuss the best practices for deploying the smart agent and using it on a machine that does not run the SAS Deployment Wizard.

To achieve better customer experience for this new smart agent, we also enhanced other agent commands and maintenance functions such as update-in-place, SAS Deployment Manager tasks, and hot fixes. These are some of the enhancements.

- Agent commands: In addition to the `start agent` command described previously, the following commands are also smart-aware:
    - `stop agent`: checks the agent status first, and then shuts down the agent (if a cloned agent is present and running) or prompts that the agent is not running
    - `delete agent`: stops the cloned agent and removes its folder
    - `check agent status`: check the status of the cloned agent
- SAS Deployment Manager tasks:
    - Change host name: You can still use SAS Deployment Manager to change the host name for the primary agent. To change the name of the cloned agents, we provide the commands `hq-agent.sh(bat)` *`OLDNAME`* or `hq-agent.sh(bat) change` *`OLDNAME ALIAS`*.
    - Change password: You can run the change password process in SAS Deployment Manager to change the password for both the primary agent and all cloned agents.
    - Unconfigure: SAS Deployment Manager unconfigures the primary agent, the smart agent contents, and all cloned agents.
- Update in place: The update in place process upgrades the contents in the primary agent area. Cloned agents only need to be restarted.

- Applying hot fixes: Hotfixes are applied to the contents of the primary agent area. Cloned agents only need to be restarted to pick up the updated contents.

These are the details of what the smart agent `start` command does:

- Uses the `hostname` command to obtain the host name of the host where the command is running.
- Checks if there’s already an agent instance for this host name.
  - If there is, it starts the agent.
  - If there isn’t, it creates a new folder with the name of the host under the directory `cloned_agents`. It then either copies the content from the primary agent to the new folder or creates symbolic links from the new folder, and then modifies the agent configuration properties file with the host name. Finally, it re-creates the data and log folders and starts the new cloned agent.

There are several issues that are not addressed and automated with this design.

- Automatically starting agents on remote nodes. The current design requires customers to start a specific node’s agent from the node machine. If we want to automate remote starting of the agent, these are possible options
  - Consumers such as SAS Grid Manager can write an interface to automate the process of remotely starting agents for the node if SAS Grid Manager has a way to start remote processes.
  - Consumers can implement the SAS Environment Manager agent `exit` call interface to remotely start the agents.
- Converting the existing manually deployed multiple agents. If a customer used the script SAS previously provided, those agents are not managed by this new approach during any processes such as update in place or hot fixes. You can still use and manually manage these agents, but they are not automatically converted to the new smart agents.
- SSL support. We can only create a self-signed certificate for the clone agent. If you need to use a custom certificate for the agent, you must configure the certificate manually. Here are the detailed steps:
  - Case 1: If you have a jks format keystore, the alias name must be hq, and put the file somewhere that is accessible from the local machine.

1. Change the agent.properties file under the directory

```
$SASEnvironmentManager/cloned_agents/$NODE_NAME/agent-5.8.0-EE/conf
```

as follows:

```
  agent.keystore.path=<keystore file location>
  agent.keystore.password=<keystore password in plain text>
```

2. Clean up the *`agent_HOME/`*`data` folder.
3. Restart the agent.

  - Case 2: If you have crt + key format certificate, find the machine where the web server is configured (make sure you can use the OpenSSL and keytool on that machine). You must change the format from crt to jks format keystore. Run this command:

```
 openssl pkcs12 -export -inkey  "customercertificate.key" -in
"customercertificate.crt" -name hq -password pass:hyperic -out
hyperic.p12
 keytool -importkeystore -deststorepass hyperic -destkeypass hyperic
-destkeystore keystore -srckeystore hyperic.p12 -srcstoretype PKCS12
-srcstorepass hyperic -alias hq
```

You can now get the jks format keystore (which has a file name of `keystore`, and a password of `hyperic`), and then follow the steps in Case1 to configure a customized keystore for the agent.

THE IMPLEMENTATION

To support the above design, we developed the scripts for the smart agent and also enhanced the SAS Deployment Wizard agent configuration process in order to deliver the scripts. As part of the agent configuration process, all smart agent scripts are installed under the `smart-agent` folder, and an empty `cloned_agents` folder is created. This diagram illustrates the folder structure:

```
$SASEnvironmentManager/
  -- agent-5.8.0-EE (the primary agent)
  -- cloned_agents  (the parent folder of all cloned node agents)
  -- smart-agent/
  ........................hq-agent.sh (control shell script file for Linux and UNIX)
  ........................hq-agent.bat (control  batch script file for Windows)
  ........................sas.properties
  ........................agent.properties
```

The primary agent under the `agent-5.8.0-EE` folder is the same agent that the SAS Deployment Wizard is used to configure. You should continue use that agent instance for monitoring your current machine. That agent is also used in `sas.servers` scripts for starting all SAS servers on that machine.

The parent folder of all cloned agents is populated with the agent instances that you create from other machines using the smart agent scripts. However, some of the files (such as the ones in the `pdk/lib` folder in the cloned agents) might not have the real copies of the agent files. Instead, they might only contain symbolic links to the primary agent. The symbolic link saves space, but more importantly, it simplifies the update in place and hot fix processes. The symbolic link goes to the directory `$SASEnvironmentManager/agent-5.8.0-EE/bundles/agent-5.8.0/pdk/lib`. This folder is used to store the run-time libraries of the primary agent. Usually, plugin files or library files in this folder are updated only during the update in place and hot fix processes.

The `$SASEnvironmentManager/smart-agent` folder contains the newly implemented smart agent scripts and property files that are used by machines that do not run the SAS Deployment Wizard but that need to have a SAS Environment Manager agent for that machine. The file `hq-agent.bat` is the smart agent file for Windows and the file `hq-agent.sh` is the smart agent file for UNIX and Linux. All of the smart agent files are created using the parameters of actual environment (such as the JRE path and the SAS Environment Manager server host) by the SAS Deployment Wizard configuration process.

The following table lists all the options for `hq-agent.bat` for Windows. The `hq-agent.sh` for UNIX and Linux has the similar options, with a few exceptions. Please check *SAS Environment Manager User's Guide* for specific information.

Command options	**Description**
NONE	Output usage information.
`start`	Get the host name from the system environment variable - `%COMPUTERNAME%.` Check if there's already an agent for this host name. If yes, start it in no-wrapper mode. If not, create a new folder under the directory `cloned_agents` and use the host name as the new folder's name. Copy the server node agent into the new folder and modify the agent configuration properties file according to the host name. Finally, re-create the data and log folders and start the new cloned agent in no-wrapper mode.
`start` *`ALIAS`*	Check if there's already an agent for the specified alias name. If yes, start it directly. If not, create a new folder under the directory `cloned_agents` and use the specified alias name as the new folder's name. Then copy the server node agent into the new folder and modify the agent configuration properties file according to the alias name. Finally, re-create the data and log folders and start the new cloned agent in no-wrapper mode.
`stop`	Get host name from the system environment variable - `%COMPUTERNAME%.` Check if there's already an agent for this host name. If yes, shut it down. If not, prompt that the agent is not running.
`stop` *`ALIAS`*	Check if there's already an agent for the specified alias name. If yes, shut it down. If not, prompt that the agent is not running.

`restart`	Get host name from the system environment variable - `%COMPUTERNAME%.` Check if there’s already an agent for this host name. If yes, restart it in no-wrapper mode directly. If not, create a new folder under the directory `cloned_agents` and use the host name as the new folder's name. Then copy the server node agent into the new folder and modify the agent configuration properties file according to the host name. Finally, re-create the data and log folders and restart the new cloned agent in no-wrapper mode.
`restart` *`ALIAS`*	Check if there’s already an agent for the specified alias name. If yes, restart it in no-wrapper mode. If not, create a new folder under the directory `cloned_agents` and use the specified alias name as the new folder's name. Then copy the server node agent into the new folder and modify the agent configuration properties file according to the alias name. Finally, re-create the data and log folders and start the new cloned agent in no-wrapper mode.
`delete`	Get host name from the system environment variable - `%COMPUTERNAME%.` Check if there’s already an agent for this host name. If yes, shut it down, then remove its folder. If not, prompt that no agent needs to be deleted.
`delete` *`ALIAS`*	Check if there’s already an agent for the specified alias name. If yes, shut it down, then remove its folder. If not, prompt that no agent needs to be deleted.
`ping`	Get host name from the system environment variable - %COMPUTERNAME%. Check if there’s already an agent for this host name. If yes, check its communication port. If not, prompt that the agent is not running.
`ping` *`ALIAS`*	Check if there’s already an agent for the specified name. If yes, check its communication port. If not, prompt that the agent is not running.
`change` *`OLDNAME`*	Get host name from the system environment variable - `%COMPUTERNAME%.` Check if there’s already an agent for the specified old name. If yes, stop and remove the old agent, re-create a new agent using the host name, and start the new agent in no-wrapper mode. If not, prompt that the agent does not exist.

change *OLDNAME ALIAS*	Check if there's already an agent for the specified old name. If yes, stop and remove the old agent, re-create a new agent with the specified alias name, and start the new agent in no-wrapper mode. If not, prompt that the agent does not exist.
All other commands	Output usage information.

THE DEPLOYMENT PROCESS

The typical process of using the smart agent can be described as follows:

1. Create shared folders from one of your machines. Run the SAS deployment Wizard to install and configure SAS software into these shared folders. This software can either be only the SAS Environment Manager agent or an order that includes SAS Environment Manager.
2. Make sure all other machines that you want to use the smart agent can access the `$SASConfig` and `$SASHome` directories in the shared folders. The other machines must at least be able to access the `$SASConfig/LevN/SASEnvironmentManager` and `$SASHome/SASPrivateJavaRuntimeEnvironment` directories. In this paper, we call these machines **node machines** and call this deployment a **shared directory deployment**. For Windows machines, you must enable the policy of symbolic links from remote to remote. This paper includes examples of sharing directories on Windows and UNIX.
3. From a node machine, go to `directory $SASConfig/LevN/Web/SASEnvironmentManager/smart-agent` and run either `hq-agent.sh start` (Linux and UNIX) or `hq-agent.bat start` (Windows). You should be able to see that a SAS Environment Manager agent has been created and started.
4. Start SAS Environment Manager. You should be able to see that this node machine is discovered by SAS Environment Manager and is listed as a platform resource.

THE EXAMPLE DEPLOYMENT PROCESS

Due to restrictions on what can be shared on different platforms, Windows and UNIX environments need different topologies to use the smart agent feature.

On Windows, we recommend using at least three machines. The second server machine is used as a shared driver for other node machines, and only has the SAS Environment Manager agent configured on it. These are the details of the Windows deployment process:

1. First Server Machine: Install and configure all SAS Enterprise BI products on this machine.
2. Second Server Machine:
    a. Create a shared folder and map it as a mapped network drive. Use the following steps.
        i. Right-click the folder that you want to share and select the **Properties**.

        ii. On the Properties window, click the **Sharing** tab and then click **Share...**.
        iii. Choose an account and click **Share** to finish sharing.
        iv. Access the network address of the shared server to start the mapping process.
        v. Right-click the shared folder and select **Map network drive...**.
        vi. Specify a drive letter and click **Finish** to create the mapped network drive.

    b. Run the SAS Deployment Wizard installation and configuration, specifying the mapped network drive as the parent folder for both `SASHome` and `SASConfig`.
        i. Deselect all products except for SAS Intelligence Platform Object Framework and SAS Environment Manager Agent in the list of products to install.
        ii. Deselect all products except for SAS Environment Manager Agent Configuration in the list of products to configure.
        iii. Specify the address of the first server machine as the address of SAS Metadata Server and SAS Environment Manager.

    c. Finish the installation and configuration process.

    d. All installation and configuration can succeed, but the SAS Environment Manager agent Windows service might not be able to start automatically. Set the values of the system environment variables JAVA_HOME and HQ_JAVA_HOME to `<$SASHome>\SASPrivateJavaRuntimeEnvironment\9.4\jre` and manually start the SAS Environment Manager agent using the command `<$SASConfig>\LevN\SASEnvironmentManager\agent-5.8.0-EE\bundles\agent-5.8.0\bin\hq-agent-nowrapper.bat start`.

3. Node machine:

    a. Access the network address of the second server and map the shared folder as a mapped network drive with the same fully qualified path as on the second server. Use same mapping process as used on the second server.)

    b. Run the command `<$SASConfig>\LevN\Web\SASEnvironmentManager\smart-agent\hq-agent.bat start` to clone and start a new SAS Environment Manager agent.

    c. If you receive the error 'The symbolic link cannot be followed because its type is disabled', you must apply a workaround. (By default, remote to remote symbolic links are disabled on some versions of Windows.)

        d. Check the symbolic link policy by using the command `fsutil behavior query SymlinkEvaluation`.
        If the remote to remote symbolic links are disabled, enable them by using the command `fsutil behavior set SymlinkEvaluation R2R:1`.
        Run the command `smart-agent\hq-agent.bat start` again.

On UNIX and Linux platforms, all SAS products can be shared by using the same fully qualified path. Therefore, you don't need a separate machine for the SAS Environment Manager agent. Follow these deployment steps:

1. Install and configure SAS software at the server machine.
2. Use NFS to export both `$SASHome` and `$SASConfig` folders with read, write, and synchronize permissions from the server machine. Because NFS version 4 might cause a read-only issue, do not use NFS version 4. Follow these steps:

    - Edit `the /etc/exports` file.
    - Add the line`'/install/cfgsas1/SASConfig *(rw,sync,no_root_squash)'` to the `export SAS Config` area.
    - Add the line `'/install/cfgsas1/SASHome *(rw,sync,no_root_squash)'` to the `export SAS Home` area.
    - Edit the `/etc/sysconfig/nfs` file.
    - Change the value of property RPCNFSDARGS from `""` to `"-N 4".`
    - Stop the NFS service by using the command `'service nfs stop'`.
    - Start the NFS service by using command `'service nfs start'`.

3. Mount both `SASHome` and `SASConfig` as the same fully qualified path in the node machine by using the following commands:

```
mount -t nfs rdcesx50091:/install/cfgsas1/SASConfig
/install/cfgsas1/SASConfig -o rw
mount -t nfs rdcesx50091:/install/cfgsas1/SASHome
/install/cfgsas1/SASHome -o rw
```

You can now access the SASConfig directory from the node machine and run the agent `start` command from the smart agent folder: `.../smart-agent/hq-agent.sh start`.

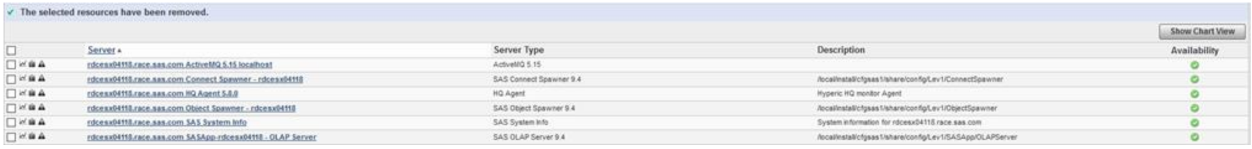
THE VALUE

When you use the SAS Environment Manager agent in the grid nodes, it brings all the SAS Environment Manager functions to those grid machines. For example, it can provide live monitoring of the grid machines and the processes running on those machines, it can take advantage of alerts defined by SAS Environment Manager service architecture, and it can generate reports that provide information about the grid nodes.

This is a list of interesting resources that can be monitored and the associated metrics that can be collected by the SAS Environment Manager Smart Agent:

- The node machine itself, including the metrics such as CPU and memory usage
- Processes running on the node machine:
    - SAS Workspace Server
    - SAS Pooled Workspace Server
    - SAS Stored Process Server
    - SAS OLAP Application server
    - SAS Object Spawner server
    - SAS Connect Server
    - SAS Workload Orchestrator
    - SAS Environment Manager Agent
- SAS System Info server and Log Collection service: SAS System Info is a SAS server that can collect the system-level information for the machine. Log Collection is a service of SAS System Info.

Figure 1 shows the availability of the processes discovered by the SAS Environment Manager agent on a shared directory machine.



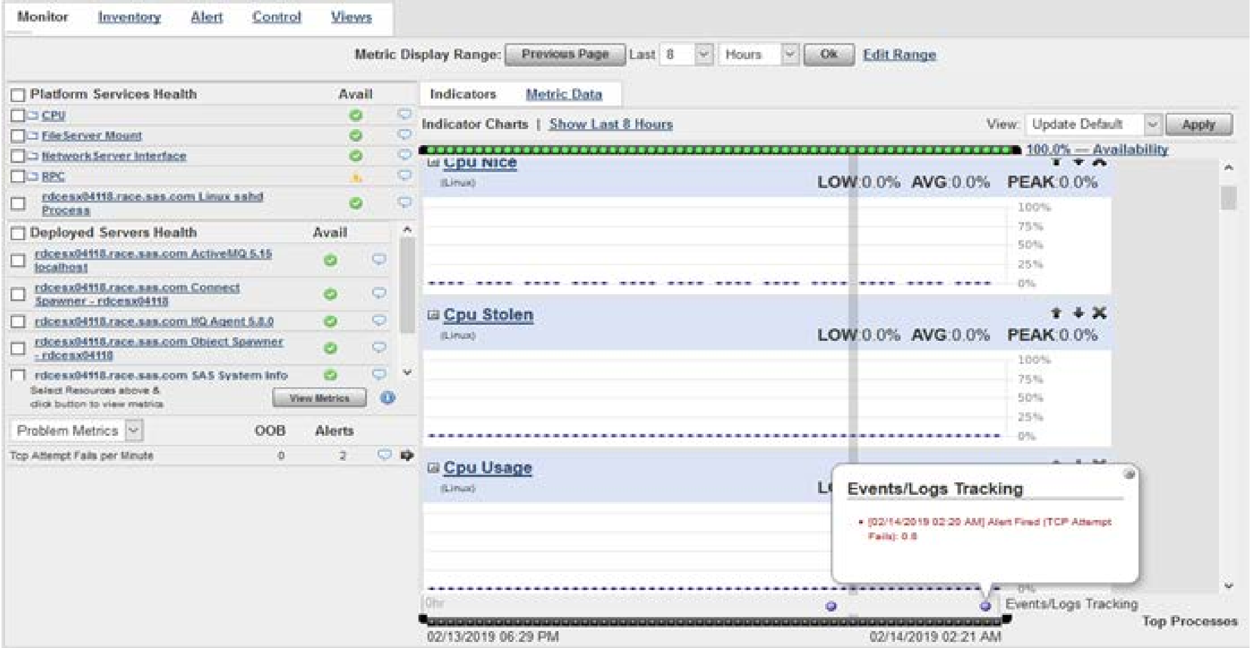
✓ The selected resources have been removed.

Show Chart View

□	Server ▴	Server Type	Description	Availability
□	rdcesx04118.race.sas.com ActiveMQ 5.15 localhost	ActiveMQ 5.15		●
□	rdcesx04118.race.sas.com Connect Spawner - rdcesx04118	SAS Connect Spawner 9.4	/local/install/cfgsas1/share/config/Lev1/ConnectSpawner	●
□	rdcesx04118.race.sas.com HQ Agent 5.8.0	HQ Agent	Hyperic HQ monitor Agent	●
□	rdcesx04118.race.sas.com Object Spawner - rdcesx04118	SAS Object Spawner 9.4	/local/install/cfgsas1/share/config/Lev1/ObjectSpawner	●
□	rdcesx04118.race.sas.com SAS System Info	SAS System Info	System information for rdcesx04118.race.sas.com	●
□	rdcesx04118.race.sas.com SASApp-rdcesx04118 - OLAP Server	SAS OLAP Server 9.4	/local/install/cfgsas1/share/config/Lev1/SASApp/OLAPServer	●

Figure 1. Processes on a Shared Directory Machine

Figure 2 shows the metrics for the shared directory machine.



Metric Display Range: Last 8 Hours

View: Update Default

□ Platform Services Health	Avail
□ CPU	●
□ File Server Mount	●
□ Network Server Interface	●
□ RPC	▲
□ rdcesx04118.race.sas.com Linux sshd Process	●

□ Deployed Servers Health	Avail
□ rdcesx04118.race.sas.com ActiveMQ 5.15 localhost	●
□ rdcesx04118.race.sas.com Connect Spawner - rdcesx04118	●
□ rdcesx04118.race.sas.com HQ Agent 5.8.0	●
□ rdcesx04118.race.sas.com Object Spawner - rdcesx04118	●
□ rdcesx04118.race.sas.com SAS System Info	●

Select Resources above & click button to view metrics

Problem Metrics	OOB	Alerts
Tcp Attempt Fails per Minute	0	2

100.0% — Availability

Cpu Nice (Linux) LOW 0.0% AVG 0.0% PEAK 0.0%

Cpu Stolen (Linux) LOW 0.0% AVG 0.0% PEAK 0.0%

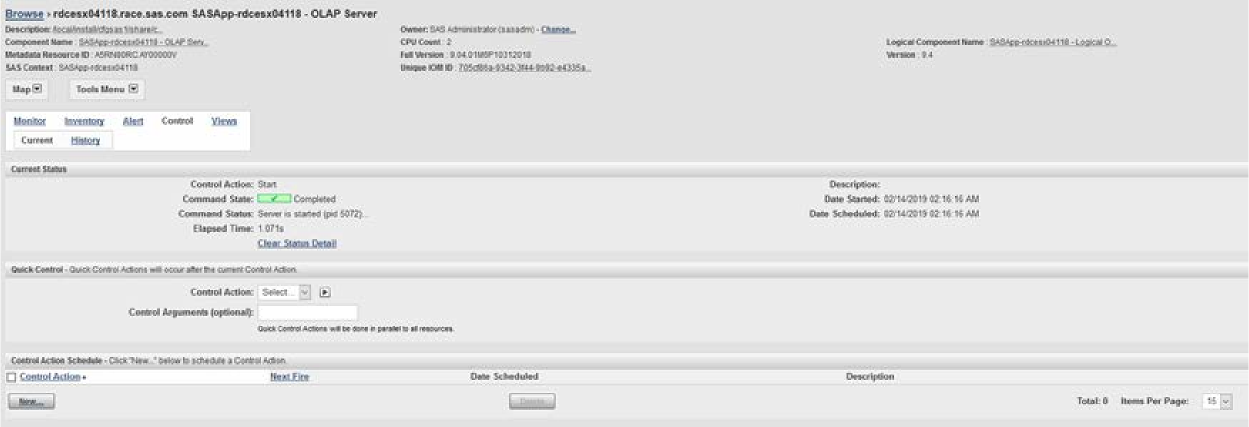
Cpu Usage (Linux)

Events/Logs Tracking

- [02/14/2019 02:20 AM] Alert Fired (TCP Attempt Fails): 0.8

Figure 2. Metrics on a Shared Directory Machine

Figure 3 illustrates the ability to use the smart agent to restart components such as an OLAP server using SAS Environment Manager.



Browse › rdcesx04118.race.sas.com SASApp-rdcesx04118 - OLAP Server

Description: /local/install/cfgsas1/share/c...
Component Name : SASApp-rdcesx04118 - OLAP Serv...
Metadata Resource ID : A5RN80RC.AY00000V
SAS Context : SASApp-rdcesx04118

Owner: SAS Administrator (sasadm) - Change...
CPU Count : 2
Full Version : 9.04.01M6P10312018
Unique IOM ID : 705cf86a-9342-3f44-9b92-e4335a...

Logical Component Name : SASApp-rdcesx04118 - Logical O...
Version : 9.4

Current Status

Control Action: Start
Command State: ✓ Completed
Command Status: Server is started (pid 5072)...
Elapsed Time: 1.071s
Clear Status Detail

Description:
Date Started: 02/14/2019 02:16:16 AM
Date Scheduled: 02/14/2019 02:16:16 AM

Quick Control - Quick Control Actions will occur after the current Control Action.

Control Action: Select...
Control Arguments (optional):
Quick Control Actions will be done in parallel to all resources.

Control Action Schedule - Click "New..." below to schedule a Control Action.

□ Control Action ▴	Next Fire	Date Scheduled	Description

New... Delete Total: 0 Items Per Page: 15

Figure 3. Restarting an OLAP Server Using the Smart Agent

The smart agent can discover the SAS System Info server and possibly the Log Collection service. The Log Collection service is used by EMI APM Log Centralization to identify an EMI client (in this case, the grid node machine). The Log Centralization service can gather log files from the discovered Log Collection service. However, the Log Centralization service gathers the log files only from a machine that has already discovered the Log Collection service usingSAS Deployment Agent.

We can use SAS Environment Manager to check and analyze the log events the log files are collected. Figure 4 and 5 show some of the details.

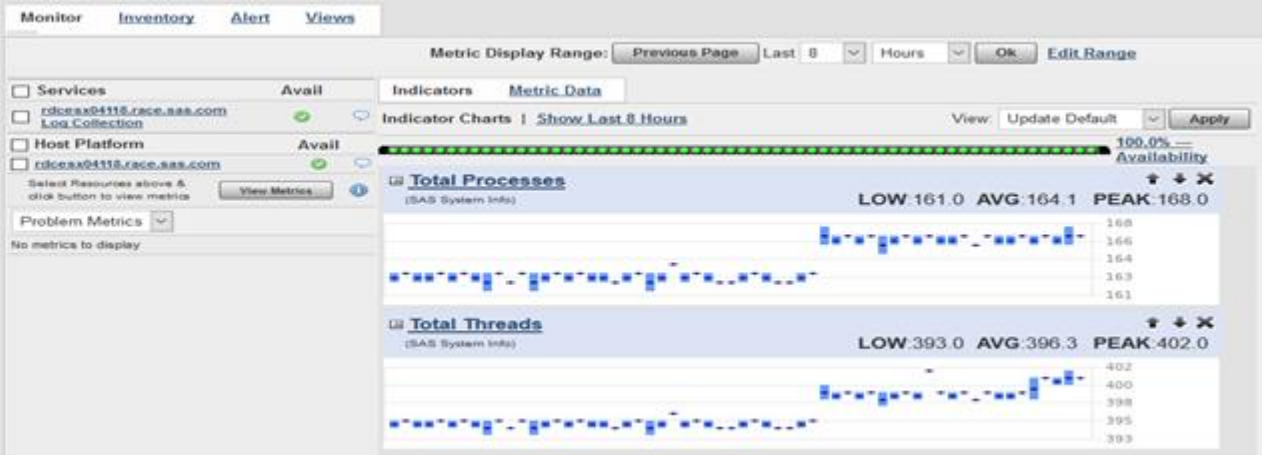
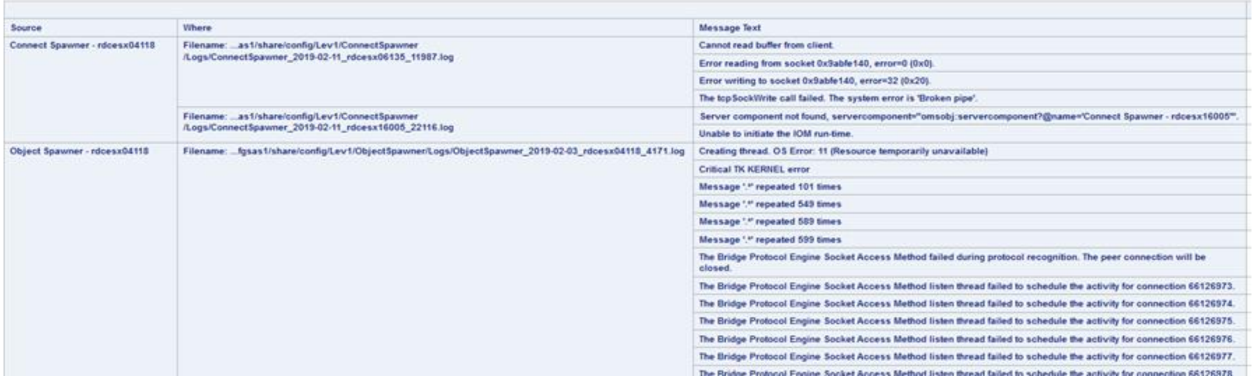


Figure 4. Log Collection Service Metrics

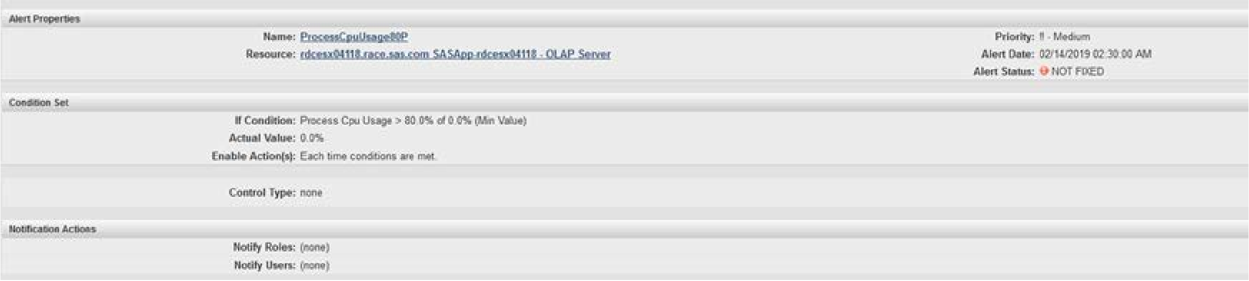


Source	Where	Message Text
Connect Spawner - rdcesx04118	Filename: ...as1/share/config/Lev1/ConnectSpawner /Logs/ConnectSpawner_2019-02-11_rdcesx06135_11987.log	Cannot read buffer from client.
		Error reading from socket 0x9abfe140, error=0 (0x0).
		Error writing to socket 0x9abfe140, error=32 (0x20).
		The tcpSockWrite call failed. The system error is 'Broken pipe'.
	Filename: ...as1/share/config/Lev1/ConnectSpawner /Logs/ConnectSpawner_2019-02-11_rdcesx16005_22116.log	Server component not found, servercomponent="omsobj:servercomponent?@name='Connect Spawner - rdcesx16005'".
		Unable to initiate the IOM run-time.
Object Spawner - rdcesx04118	Filename: ...fgsas1/share/config/Lev1/ObjectSpawner/Logs/ObjectSpawner_2019-02-03_rdcesx04118_4171.log	Creating thread. OS Error: 11 (Resource temporarily unavailable)
		Critical TK KERNEL error
		Message '.*' repeated 101 times
		Message '.*' repeated 549 times
		Message '.*' repeated 589 times
		Message '.*' repeated 599 times
		The Bridge Protocol Engine Socket Access Method failed during protocol recognition. The peer connection will be closed.
		The Bridge Protocol Engine Socket Access Method listen thread failed to schedule the activity for connection 66126973.
		The Bridge Protocol Engine Socket Access Method listen thread failed to schedule the activity for connection 66126974.
		The Bridge Protocol Engine Socket Access Method listen thread failed to schedule the activity for connection 66126975.
		The Bridge Protocol Engine Socket Access Method listen thread failed to schedule the activity for connection 66126976.
		The Bridge Protocol Engine Socket Access Method listen thread failed to schedule the activity for connection 66126977.
		The Bridge Protocol Engine Socket Access Method listen thread failed to schedule the activity for connection 66126978

Figure 5. Collected Log Messages

Some manual steps are required in order for the smart agent to discover all of these resources. Contact SAS Technical Support if you are interested in those manual steps.

Figure 6 shows the CPU Usage alert, which is atypical and useful alert.



Alert Properties

Name: ProcessCpuUsage80P
Resource: rdcesx04118.race.sas.com SASApp-rdcesx04118 - OLAP Server

Priority: !! - Medium
Alert Date: 02/14/2019 02:30:00 AM
Alert Status: ⊘ NOT FIXED

Condition Set

If Condition: Process Cpu Usage > 80.0% of 0.0% (Min Value)
Actual Value: 0.0%
Enable Action(s): Each time conditions are met.

Control Type: none

Notification Actions

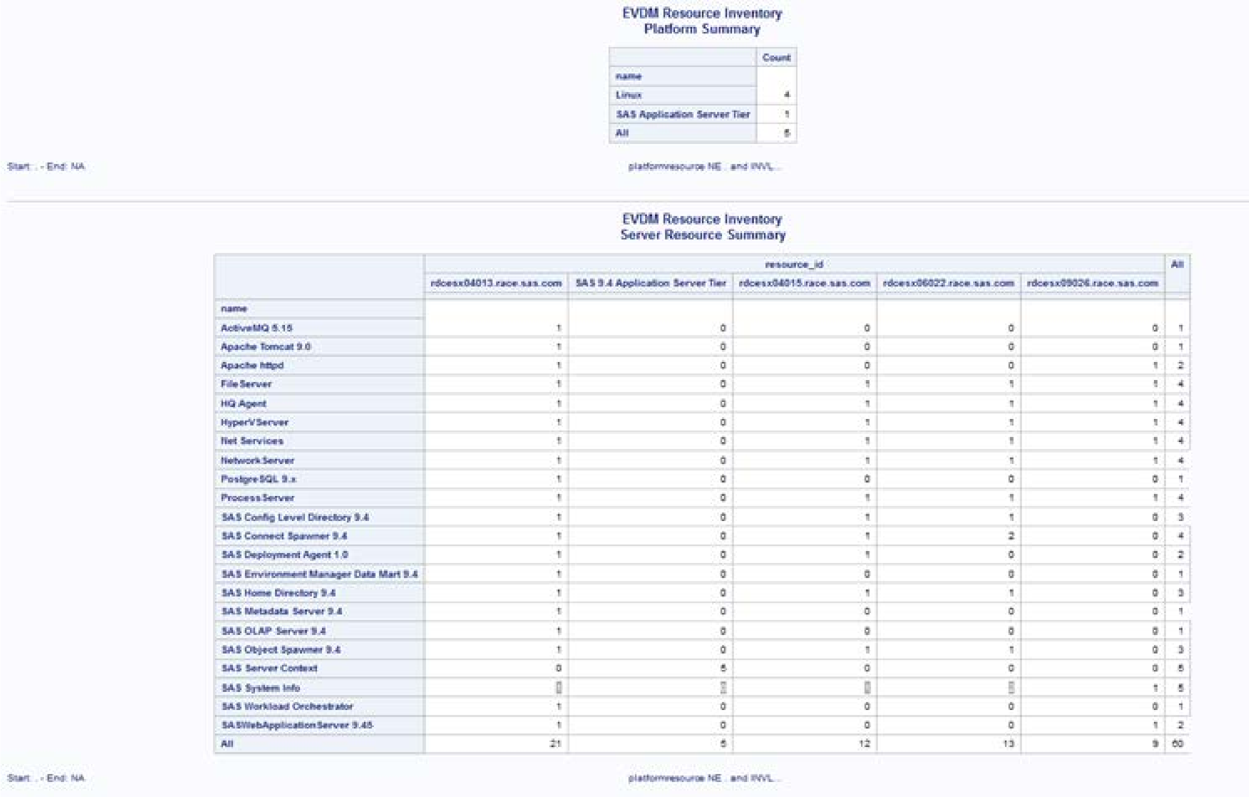
Notify Roles: (none)
Notify Users: (none)

Figure 6. CPU Usage Alert

Many reports related to the grid machines provide valuable insights about those machines and the grid environment, such as the following:

- Platform and Server Resource summary
- Platform Resources and Instances
- List of Alerts enabled by Resources
- Server properties defined in Metadata Server
- Event summary

Figure 7 shows the Platform and Server resource summaries.



EVDM Resource Inventory
Platform Summary

	Count
name	
Linux	4
SAS Application Server Tier	1
All	5

Start: . - End: NA

platformresource NE . and INVL...

EVDM Resource Inventory
Server Resource Summary

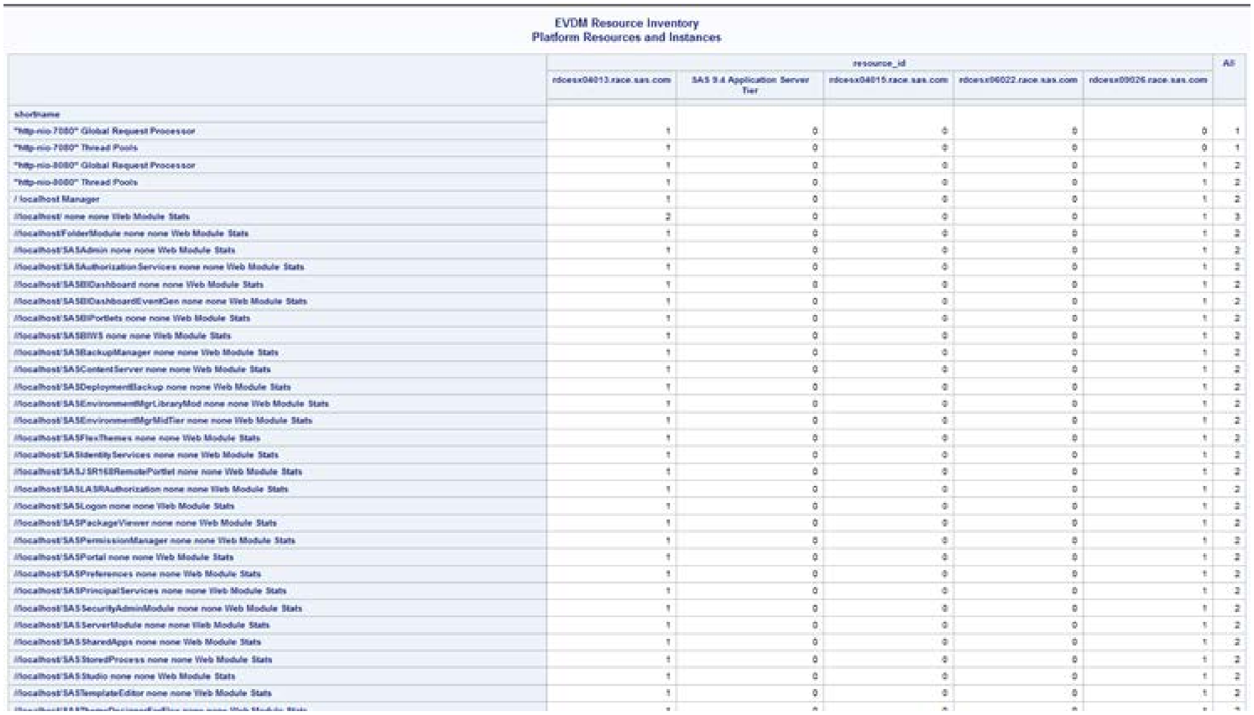
	resource_id					All
	rdcesx04013.race.sas.com	SAS 9.4 Application Server Tier	rdcesx04015.race.sas.com	rdcesx06022.race.sas.com	rdcesx09026.race.sas.com	
name						
ActiveMQ 5.15	1	0	0	0	0	1
Apache Tomcat 9.0	1	0	0	0	0	1
Apache httpd	1	0	0	0	1	2
File Server	1	0	1	1	1	4
HQ Agent	1	0	1	1	1	4
HyperV Server	1	0	1	1	1	4
Net Services	1	0	1	1	1	4
Network Server	1	0	1	1	1	4
PostgreSQL 9.x	1	0	0	0	0	1
Process Server	1	0	1	1	1	4
SAS Config Level Directory 9.4	1	0	1	1	0	3
SAS Connect Spawner 9.4	1	0	1	2	0	4
SAS Deployment Agent 1.0	1	0	1	0	0	2
SAS Environment Manager Data Mart 9.4	1	0	0	0	0	1
SAS Home Directory 9.4	1	0	1	1	0	3
SAS Metadata Server 9.4	1	0	0	0	0	1
SAS OLAP Server 9.4	1	0	0	0	0	1
SAS Object Spawner 9.4	1	0	1	1	0	3
SAS Server Context	0	5	0	0	0	5
SAS System Info	▯	▯	▯	▯	1	5
SAS Workload Orchestrator	1	0	0	0	0	1
SASWebApplicationServer 9.45	1	0	0	0	1	2
All	21	5	12	13	9	60

Start: . - End: NA

platformresource NE . and INVL...

Figure 7. Platform and Server Resource Summary Reports

Figure 8 shows Platform Types and Instances.

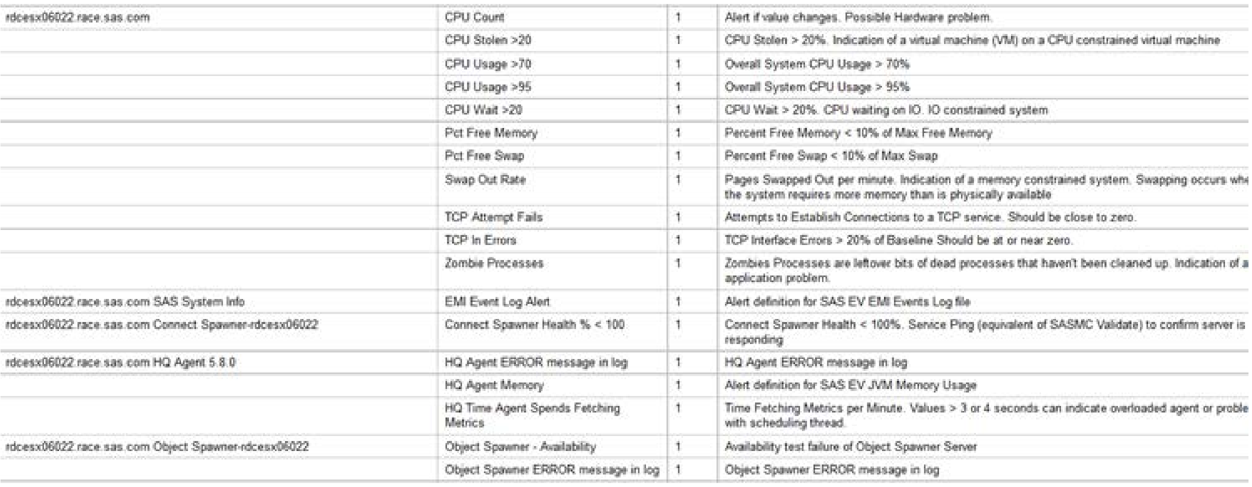


EVDM Resource Inventory
Platform Resources and Instances

	resource_id					All
	rdcesx04013.race.sas.com	SAS 9.4 Application Server Tier	rdcesx04015.race.sas.com	rdcesx06022.race.sas.com	rdcesx09026.race.sas.com	
shortname						
"http-nio-7080" Global Request Processor	1	0	0	0	0	1
"http-nio-7080" Thread Pools	1	0	0	0	0	1
"http-nio-8080" Global Request Processor	1	0	0	0	1	2
"http-nio-8080" Thread Pools	1	0	0	0	1	2
/ localhost Manager	1	0	0	0	1	2
//localhost/ none none Web Module Stats	2	0	0	0	1	3
//localhost/FolderModule none none Web Module Stats	1	0	0	0	1	2
//localhost/SASAdmin none none Web Module Stats	1	0	0	0	1	2
//localhost/SASAuthorizationServices none none Web Module Stats	1	0	0	0	1	2
//localhost/SASBIDashboard none none Web Module Stats	1	0	0	0	1	2
//localhost/SASBIDashboardEventGen none none Web Module Stats	1	0	0	0	1	2
//localhost/SASBIPortlets none none Web Module Stats	1	0	0	0	1	2
//localhost/SASBIWS none none Web Module Stats	1	0	0	0	1	2
//localhost/SASBackupManager none none Web Module Stats	1	0	0	0	1	2
//localhost/SASContentServer none none Web Module Stats	1	0	0	0	1	2
//localhost/SASDeploymentBackup none none Web Module Stats	1	0	0	0	1	2
//localhost/SASEnvironmentMgrLibraryMod none none Web Module Stats	1	0	0	0	1	2
//localhost/SASEnvironmentMgrMidTier none none Web Module Stats	1	0	0	0	1	2
//localhost/SASFlexThemes none none Web Module Stats	1	0	0	0	1	2
//localhost/SASIdentityServices none none Web Module Stats	1	0	0	0	1	2
//localhost/SASJSR168RemotePortlet none none Web Module Stats	1	0	0	0	1	2
//localhost/SASLASRAuthorization none none Web Module Stats	1	0	0	0	1	2
//localhost/SASLogon none none Web Module Stats	1	0	0	0	1	2
//localhost/SASPackageViewer none none Web Module Stats	1	0	0	0	1	2
//localhost/SASPermissionManager none none Web Module Stats	1	0	0	0	1	2
//localhost/SASPortal none none Web Module Stats	1	0	0	0	1	2
//localhost/SASPreferences none none Web Module Stats	1	0	0	0	1	2
//localhost/SASPrincipalServices none none Web Module Stats	1	0	0	0	1	2
//localhost/SASSecurityAdminModule none none Web Module Stats	1	0	0	0	1	2
//localhost/SASServerModule none none Web Module Stats	1	0	0	0	1	2
//localhost/SASSharedApps none none Web Module Stats	1	0	0	0	1	2
//localhost/SASStoredProcess none none Web Module Stats	1	0	0	0	1	2
//localhost/SASStudio none none Web Module Stats	1	0	0	0	1	2
//localhost/SASTemplateEditor none none Web Module Stats	1	0	0	0	1	2

Figure 8. Platform Types and Instances Report

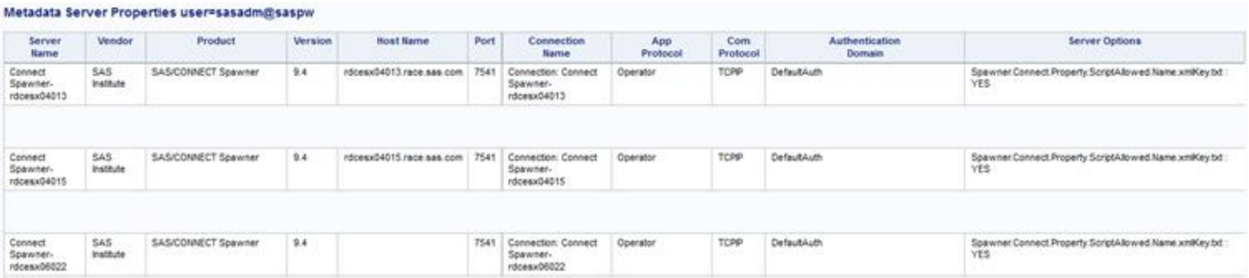
Figure 9 shows the Alerts Enabled by Resources.



rdcesx06022.race.sas.com	CPU Count	1	Alert if value changes. Possible Hardware problem.
	CPU Stolen >20	1	CPU Stolen > 20%. Indication of a virtual machine (VM) on a CPU constrained virtual machine
	CPU Usage >70	1	Overall System CPU Usage > 70%
	CPU Usage >95	1	Overall System CPU Usage > 95%
	CPU Wait >20	1	CPU Wait > 20%. CPU waiting on IO. IO constrained system
	Pct Free Memory	1	Percent Free Memory < 10% of Max Free Memory
	Pct Free Swap	1	Percent Free Swap < 10% of Max Swap
	Swap Out Rate	1	Pages Swapped Out per minute. Indication of a memory constrained system. Swapping occurs whe the system requires more memory than is physically available
	TCP Attempt Fails	1	Attempts to Establish Connections to a TCP service. Should be close to zero.
	TCP In Errors	1	TCP Interface Errors > 20% of Baseline Should be at or near zero.
	Zombie Processes	1	Zombies Processes are leftover bits of dead processes that haven't been cleaned up. Indication of a application problem.
rdcesx06022.race.sas.com SAS System Info	EMI Event Log Alert	1	Alert definition for SAS EV EMI Events Log file
rdcesx06022.race.sas.com Connect Spawner-rdcesx06022	Connect Spawner Health % < 100	1	Connect Spawner Health < 100%. Service Ping (equivalent of SASMC Validate) to confirm server is responding
rdcesx06022.race.sas.com HQ Agent 5.8.0	HQ Agent ERROR message in log	1	HQ Agent ERROR message in log
	HQ Agent Memory	1	Alert definition for SAS EV JVM Memory Usage
	HQ Time Agent Spends Fetching Metrics	1	Time Fetching Metrics per Minute. Values > 3 or 4 seconds can indicate overloaded agent or proble with scheduling thread.
rdcesx06022.race.sas.com Object Spawner-rdcesx06022	Object Spawner - Availability	1	Availability test failure of Object Spawner Server
	Object Spawner ERROR message in log	1	Object Spawner ERROR message in log

Figure 9. Alerts Enabled by Resources Report

Figure 10 is an example of Server Properties in the SAS Metadata Server.

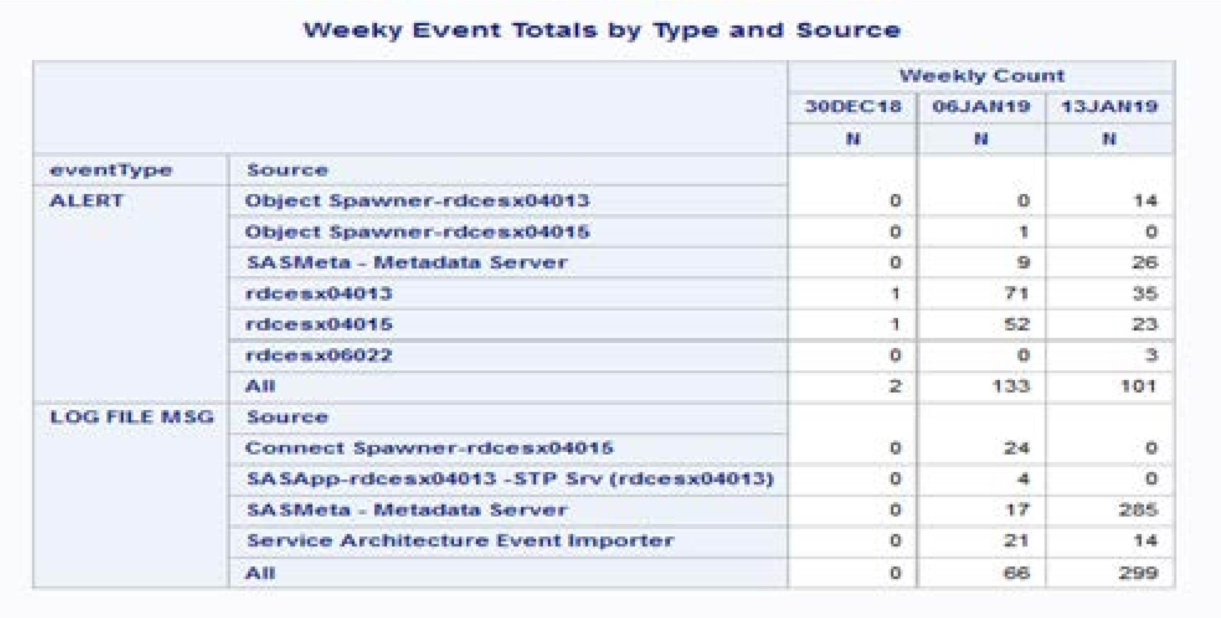


Metadata Server Properties user=sasadm@saspw

Server Name	Vendor	Product	Version	Host Name	Port	Connection Name	App Protocol	Com Protocol	Authentication Domain	Server Options
Connect Spawner-rdcesx04013	SAS Institute	SAS/CONNECT Spawner	9.4	rdcesx04013.race.sas.com	7541	Connection: Connect Spawner-rdcesx04013	Operator	TCPIP	DefaultAuth	Spawner.Connect.Property.ScriptAllowed.Name.xmlKey.txt : YES
Connect Spawner-rdcesx04015	SAS Institute	SAS/CONNECT Spawner	9.4	rdcesx04015.race.sas.com	7541	Connection: Connect Spawner-rdcesx04015	Operator	TCPIP	DefaultAuth	Spawner.Connect.Property.ScriptAllowed.Name.xmlKey.txt : YES
Connect Spawner-rdcesx06022	SAS Institute	SAS/CONNECT Spawner	9.4		7541	Connection: Connect Spawner-rdcesx06022	Operator	TCPIP	DefaultAuth	Spawner.Connect.Property.ScriptAllowed.Name.xmlKey.txt : YES

Figure 10. SAS Metadata Server Properties Report

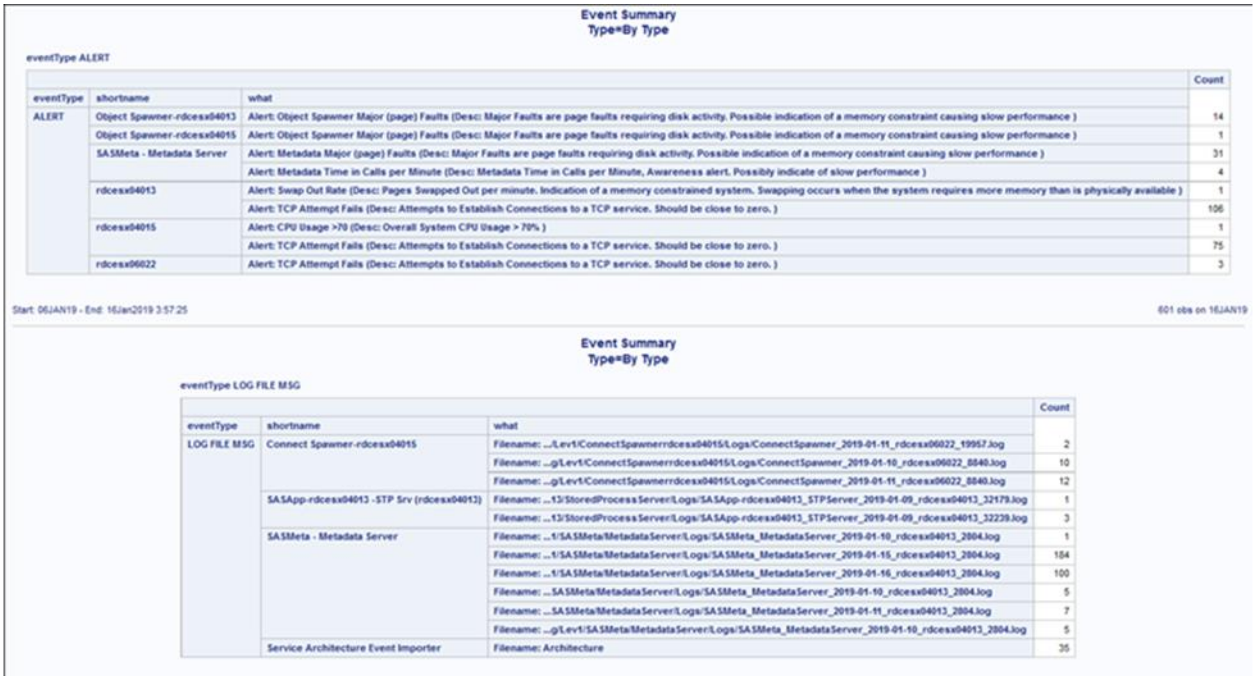
Figures 11 and 12 are the reports for Events.



Weeky Event Totals by Type and Source

		Weekly Count		
		30DEC18	06JAN19	13JAN19
		N	N	N
eventType	Source			
ALERT	Object Spawner-rdcesx04013	0	0	14
	Object Spawner-rdcesx04015	0	1	0
	SASMeta - Metadata Server	0	9	26
	rdcesx04013	1	71	35
	rdcesx04015	1	52	23
	rdcesx06022	0	0	3
	All	2	133	101
LOG FILE MSG	Source			
	Connect Spawner-rdcesx04015	0	24	0
	SASApp-rdcesx04013 -STP Srv (rdcesx04013)	0	4	0
	SASMeta - Metadata Server	0	17	285
	Service Architecture Event Importer	0	21	14
	All	0	66	299

Figure 11. Event Totals Report



Event Summary
Type=By Type

eventType ALERT

eventType	shortname	what	Count
ALERT	Object Spawner-rdcesx04013	Alert: Object Spawner Major (page) Faults (Desc: Major Faults are page faults requiring disk activity. Possible indication of a memory constraint causing slow performance)	14
	Object Spawner-rdcesx04015	Alert: Object Spawner Major (page) Faults (Desc: Major Faults are page faults requiring disk activity. Possible indication of a memory constraint causing slow performance)	1
	SASMeta - Metadata Server	Alert: Metadata Major (page) Faults (Desc: Major Faults are page faults requiring disk activity. Possible indication of a memory constraint causing slow performance)	31
		Alert: Metadata Time in Calls per Minute (Desc: Metadata Time in Calls per Minute, Awareness alert. Possibly indicate of slow performance)	4
	rdcesx04013	Alert: Swap Out Rate (Desc: Pages Swapped Out per minute. Indication of a memory constrained system. Swapping occurs when the system requires more memory than is physically available)	1
		Alert: TCP Attempt Fails (Desc: Attempts to Establish Connections to a TCP service. Should be close to zero.)	106
	rdcesx04015	Alert: CPU Usage >70 (Desc: Overall System CPU Usage > 70%)	1
		Alert: TCP Attempt Fails (Desc: Attempts to Establish Connections to a TCP service. Should be close to zero.)	75
	rdcesx06022	Alert: TCP Attempt Fails (Desc: Attempts to Establish Connections to a TCP service. Should be close to zero.)	3

Start: 06JAN19 - End: 16Jan2019 3:57:25

601 obs on 16JAN19

Event Summary
Type=By Type

eventType LOG FILE MSG

eventType	shortname	what	Count
LOG FILE MSG	Connect Spawner-rdcesx04015	Filename: .../Lev1/ConnectSpawnerrdcesx04015/Logs/ConnectSpawner_2019-01-11_rdcesx06022_19957.log	2
		Filename: ...g/Lev1/ConnectSpawnerrdcesx04015/Logs/ConnectSpawner_2019-01-10_rdcesx06022_8840.log	10
		Filename: ...g/Lev1/ConnectSpawnerrdcesx04015/Logs/ConnectSpawner_2019-01-11_rdcesx06022_8840.log	12
	SASApp-rdcesx04013 -STP Srv (rdcesx04013)	Filename: ...13/StoredProcessServer/Logs/SASApp-rdcesx04013_STPServer_2019-01-09_rdcesx04013_32179.log	1
		Filename: ...13/StoredProcessServer/Logs/SASApp-rdcesx04013_STPServer_2019-01-09_rdcesx04013_32239.log	3
	SASMeta - Metadata Server	Filename: ...1/SASMeta/MetadataServer/Logs/SASMeta_MetadataServer_2019-01-10_rdcesx04013_2004.log	1
		Filename: ...1/SASMeta/MetadataServer/Logs/SASMeta_MetadataServer_2019-01-15_rdcesx04013_2004.log	184
		Filename: ...1/SASMeta/MetadataServer/Logs/SASMeta_MetadataServer_2019-01-16_rdcesx04013_2004.log	100
		Filename: ...SASMeta/MetadataServer/Logs/SASMeta_MetadataServer_2019-01-10_rdcesx04013_2004.log	5
		Filename: ...SASMeta/MetadataServer/Logs/SASMeta_MetadataServer_2019-01-11_rdcesx04013_2004.log	7
		Filename: ...g/Lev1/SASMeta/MetadataServer/Logs/SASMeta_MetadataServer_2019-01-10_rdcesx04013_2004.log	5
	Service Architecture Event Importer	Filename: Architecture	35

Figure 12. Event Summary Report

WHAT IS NEW IN SAS 9.4M6 MIDDLE TIER PLATFORM

Here is a brief list of updates and new features that are implemented in SAS 9.4m6:

- SAS Private JRE: We now support Java 8 and we updated SAS Private JRE to Java 8 as well.
- SAS Web Server: We are no longer using Pivotal Web Server as our source for SAS Web Server. 9.4m6 SAS Web Server is now based on Apache Open Source HTTP server, and it is completely built and packaged by SAS. It has also been upgraded from Apache HTTP server 2.2.34 to Apache HTTP server 2.4.34. As part of this upgrade, FIPs compliance is now correctly supported.
- SAS Web Application Server: Upgraded from Pivotal tcServer 3.2.5 (or Tomcat 8.5.13) to Pivotal tcServer 3.2.11 (or Tomcat 8.5.32).
- SAS Messaging Engine: Upgraded to use ActiveMQ 5.15.5
- SAS Cache Locator: Upgraded to use GemFire 8.2.7. This is a big change, because the SAS 9.4m5 release used GemFire 6.3.3, which was released several years ago.
- Added a feature to support preserving SSL for ActiveMQ during upgrade in place
- Added a feature to support preserving SSL for Gemfire during upgrade in place
- Updated SAS Environment Manager plugins for all of the new middle tier infrastructure software (such as SAS Web Server, SAS Web Application Server, and SAS Messaging Engine - ActiveMQ)
- SAS Web Server is changed to use SAS built Apache httpd v2.4. To support that change, SAS Environment Manager is changed to use a new plugin that is packaged with apache-plugin.jar.
- Updated embedded Web Application Server, ActiveMQ, and support for new Postgres
- Lots of security upgrades

Further details can be found in "What's New in SAS® 9.4 and SAS® Viya® 3.4".

CONCLUSION

This paper discussed the design and the implementation of a new smart agent feature. The feature is mainly used for a SAS grid environment in which customers don't want to run SAS Deployment Wizard on every grid node. However, it can also be used in any shared-directory deployment. We also discuss the process of how you can take advantage of this new smart agent feature, and the real value that you can get with using the smart agent in your grid environment.

We also reviewed what is new in SAS® 9.4 middle-tier platform, which should help you to stay up-to-date about the changes of the SAS 9.4 middle tier infrastructure software.

REFERENCES

SAS Institute Inc. 2018, *What's New in SAS® 9.4 and SAS® Viya® 3.4*, Available at https://documentation.sas.com/?docsetId=whatsnew&docsetTarget=titlepage.htm&docsetVersion=9.4_3.4

SAS Institute Inc. 2018. *SAS Environment Manager 2.5 User's Guide.* Cary, NC: SAS Institute Inc. Available at https://documentation.sas.com/?cdcId=evcdc&cdcVersion=2.5_M1&docsetId=evug&docsetTarget=titlepage.htm

SAS Institute Inc. 2018. *SAS 9.4 Intelligence Platform: Middle-Tier Administration Guide.* Cary, NC: SAS Institute Inc. Available at https://documentation.sas.com/?cdcId=bicdc&cdcVersion=9.4&docsetId=bimtag&docsetTarget=titlepage.htm

Peters, Amy, Bonham, Bob, and Li, Zhiyong. 2013, "Monitoring 101: New Features in SAS 9.4 for Monitoring Your SAS Intelligence Platform." *Proceedings of the SAS Global Forum 2013 Conference*. Cary, NC: SAS Institute Inc. Available at https://support.sas.com/resources/papers/proceedings13/463-2013.pdf

Li, Zhiyong, and Fernandez, Alec. 2014, "Migrating SAS Java EE Applications from WebLogic, WebSphere, and JBoss to Pivotal tc Server." *Proceedings of the SAS Global Forum 2014 Conference*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings14/SAS357-2014.pdf

Li, Zhiyong, and Thorland, Mike. 2015, "Your Top Ten SAS Middle-Tier Questions." *Proceedings of the SAS Global Forum 2015 Conference*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings15/SAS1904-2015.pdf

Li, Zhiyong, and Chrzaszcz, Giles. 2016, "Advanced Topics in SAS Environment Manager." *Proceedings of the SAS Global Forum 2016 Conference*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings16/SAS5680-2016.pdf

Li, Zhiyong 2017, "Frequently Asked Questions about SAS Environment Manager on SAS 9.4." *Proceedings of the SAS Global Forum 2017 Conference*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings17/SAS0575-2017.pdf

Li, Zhiyong, and Qing, Gong. 2018, "Recent SAS® 9.4 Middle-Tier Platform Updates for Fun and Profit." *Proceedings of the SAS Global Forum 2018 Conference*. Cary, NC: SAS Institute Inc. Available at https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/1918-2018.pdf

ACKNOWLEDGMENTS

Thanks to Cheryl Doninger for encouraging us to conduct this project. Thanks to Scott Parrish, Ron Fleming, Taylor Shields, and Sam Chen for participating the initial design discussion. Chao Yang and Fred Li as SAS Beijing R&D did most of the implementation work.

CONTACT INFORMATION

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

Zhiyong Li
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
Zhiyong.Li@sas.com
http://www.sas.com