

SAS® GLOBAL FORUM 2018

USERS PROGRAM

Automation of Linux Multi-Tiered SAS® and
Load Sharing Facility (LFS) Services

Paper # 2757-2018

April 8 - 11 | Denver, CO
#SASGF

Automation of Linux Multi-Tiered SAS® and Load Sharing Facility (LFS) Services

Piyush Singh, Prasoon Sangwan, Ghiyasuddin Mohammed Faraz khan

TATA Consultancy Services Ltd

ABSTRACT

Have you ever faced the situation where your SAS Platform went down and you need to bring entire platform up and running as soon as possible? Have you experienced the situation when you missed a service to start and proceeded for next service in multi node environment? Alternatively, if you forgot to maintain the start-up/ shut down order in multimode and multi cluster environment. These things are very normal for a SAS Platform Administrator work and responsibilities.

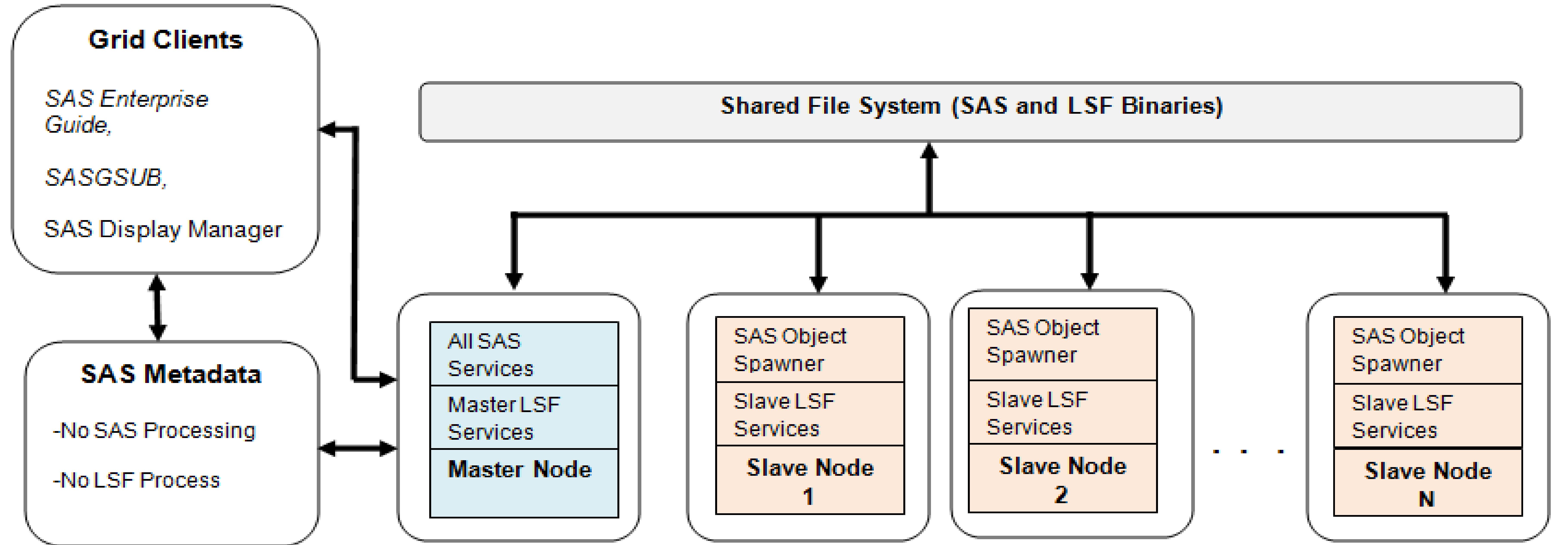
This paper explains how you can automate your platform services and no need to worry for sequence and order. This helps to reduce the effort and time to bring SAS Linux platform up or down quickly, without human error. Particularly this will be very helpful for SAS Grid environment, running in Linux environment.

Automation of Linux Multi-Tiered SAS® and Load Sharing Facility (LFS) Services

Piyush Singh, Prasoon Sangwan, Ghiyasuddin Mohammed Faraz khan

TATA Consultancy Services Ltd

SAS Grid (LSF + SAS) Service Architecture



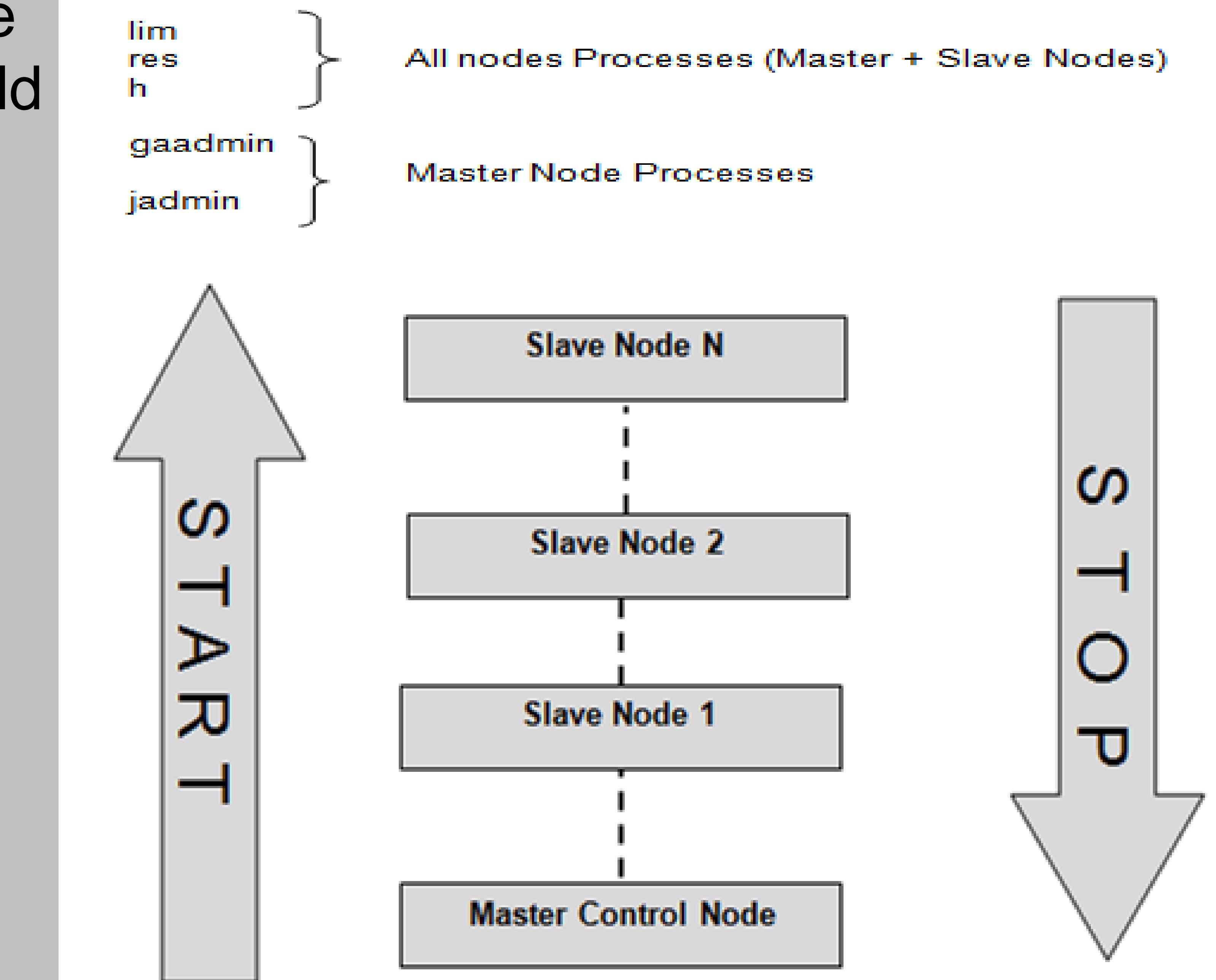
Automation of Linux Multi-Tiered SAS® and Load Sharing Facility (LFS) Services

Piyush Singh, Prasoon Sangwan, Ghiyasuddin Mohammed Faraz khan

TATA Consultancy Services Ltd

LSF Service Architecture

- LSF is the layer, which work for load balancing among the compute nodes defined in Grid cluster. LSF services should be up and running on all compute nodes to accept and execute the SAS jobs.
- If LSF is down on any node, it won't accept any job but if LSF is down on master one, it stops the entire SAS Grid environment.
- Master node need some extra services compare to other slave nodes, to monitor the entire Grid cluster. The other set of common services run on all nodes.



Automation of Linux Multi-Tiered SAS® and Load Sharing Facility (LFS) Services

Piyush Singh, Prasoon Sangwan, Ghiyasuddin Mohammed Faraz khan

TATA Consultancy Services Ltd

LSF Start Process (Sample Script for Three Nodes Grid)

```
#!/bin/bash
## Node and variable definition
## node1 is master and node2, node3 are slave
nodes

total_node=3
node1=gridnode1
node2=gridnode2
node3=gridnode3
lsf_conf=<lsf_home>/lsf/conf
## End of variable definition

for i in [1..3]
do
j=node$i
grid_node=`echo $echo $(!j)`

if [ $i -eq 1 ]
then
echo "$i. Starting lsf on node :
$grid_node"
. $lsf_conf/profile.lsf
lsadmin limstartup
sleep 5
lsadmin resstartup
sleep 5
badmin hstartup
sleep 5
else
echo "$i. Starting lsf on node :
$grid_node"
sudo -E ssh root@$grid_node '
lsf_conf=<lsf_home>/lsf/conf
. $lsf_conf/profile.lsf
lsadmin limstartup
sleep 5
lsadmin resstartup
sleep 5
badmin hstartup
sleep 5
exit
'
fi
Done
if [ $i -eq $total_node ]
echo "Remaining lsf services (gabd and jfd)
on master node"

/<lsf_home>/lsf/gms/gaadmin start
sleep 10
.
/<processmanager_home>/processmanager/conf/pro
file.js
jaadmin start
sleep 10
fi
## End of script.
```

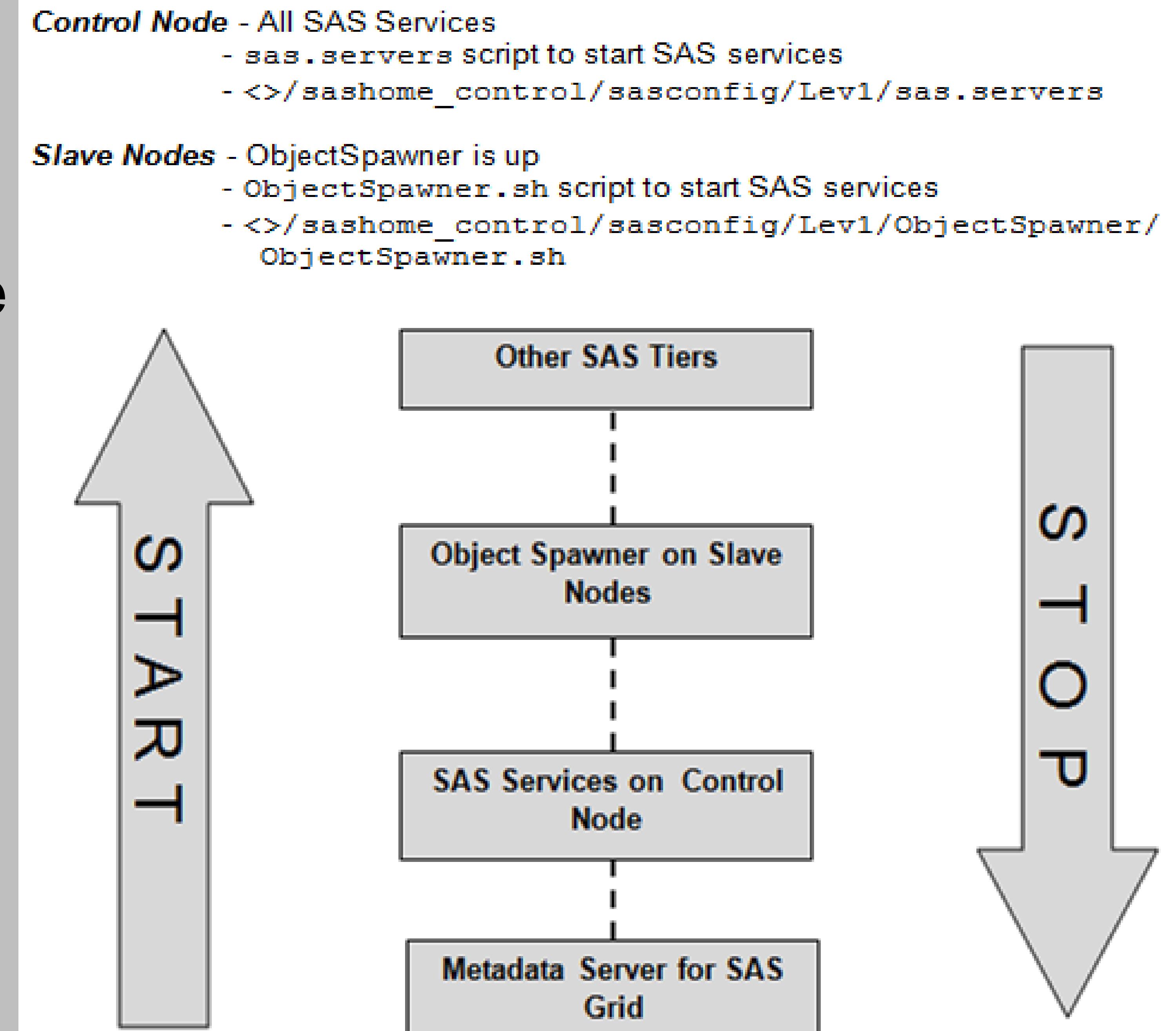
Automation of Linux Multi-Tiered SAS® and Load Sharing Facility (LFS) Services

Piyush Singh, Prasoon Sangwan, Ghiyasuddin Mohammed Faraz khan

TATA Consultancy Services Ltd

SAS Service Architecture

- While starting SAS services in SAS Grid environment you need to make sure that Metadata tier is the first tier to start.
- Until Metadata is up, you cannot start other tiers because all authentication takes place at Metadata server.
- Once Metadata is up, you can start other tiers (based on dependency order) accordingly.
- Right side is the flow chart for SAS services start process in Grid environment. This shows the order for different tiers startup order.



Automation of Linux Multi-Tiered SAS® and Load Sharing Facility (LFS) Services

Piyush Singh, Prasoon Sangwan, Ghiyasuddin Mohammed Faraz khan

TATA Consultancy Services Ltd

SAS Start Process (Sample Script)

```
#!/bin/bash
## Tier, Host and Variables Definition
tiers=3
tier1='Metadata'
host1=<Metadata_host>
tier2='Master'
host2=<Master_Node_host>
tier3='Slave'
host3=<Slave_ObjectSpawner_host>
##
for i in {1..3}
do
j=tier$i
tier_name=`echo $echo ${!j}`
k=host$i
host=`echo $echo ${!k}`
echo "Tier is : $tier_name and associated
host is : $host"
if [ $tier_name == "Metadata" ]
then
ssh <SASInstallerID>@$host '
sasconfig=<SASHomeMetadata>/sasconfig/Levl
$sasconfig/sas.servers start
'
else
if [ $tier_name == "Master" ]
then
ssh <SASInstallerID>@$host '
sasconfig=<SASHomeControl>/sasconfig/Levl
$sasconfig/sas.servers start
'
else
ssh <SASInstallerID>@$host '
sasconfig=/<SASHomeControl>/sasconfig/Levl/Obj
ectSpawner
$sasconfig/ObjectSpawner.sh start
'
fi
fi
done
## End of script.
```

Automation of Linux Multi-Tiered SAS® and Load Sharing Facility (LFS) Services

Piyush Singh, Prasoon Sangwan, Ghiyasuddin Mohammed Faraz khan

TATA Consultancy Services Ltd

Recommended Reading

For detailed information regarding:

- *SAS Grid Architecture*
- *Multi-tier SAS Dependency*
- *LSF Services Automation*
- *SAS Services Automation*

Please Refer Corresponding Paper 2757-2018

For additional information, you can refer:

- *Grid Computing in SAS® 9.4, Third Edition*
- *Platform LSF, Version 9 Release 1.3*



SAS[®] GLOBAL FORUM 2018

April 8 - 11 | Denver, CO
Colorado Convention Center

#SASGF

Automation of Linux Multi-Tiered SAS® Grid and LSF Services

Piyush Singh, Prasoon Sangwan, Ghiyasuddin Mohammed Faraz Khan

TATA Consultancy Services Ltd

ABSTRACT

Have you ever faced the situation where your SAS Platform went down and you need to bring entire platform up and running as soon as possible? Have you experienced the situation when you missed a service to start and proceeded for next service in multi node environment? Alternatively, if you forgot to maintain the start-up/ shut down order in multimode and multi cluster environment. These things are very normal for a SAS Platform Administrator work and responsibilities. This paper explains how you can automate your platform services and no need to worry for sequence and order. This helps to reduce the effort and time to bring SAS Linux platform up or down quickly, without human error. Particularly this will be very helpful for SAS Grid environment, running in Linux environment.

INTRODUCTION

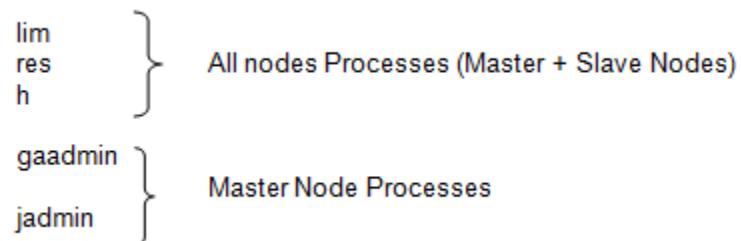
There are two key component of SAS Grid platform which are SAS and Load Sharing Facility (LSF). There are many operational activities when you need to start/stop or restart entire SAS Grid platform. In such cases you need to follow the sequence to execute the different services command on Grid servers. If sequence is not followed properly, Grid services may not start properly and you may experience some unexpected behavior. From SAS perspective, you can have single or multi-tier SAS deployment. If you have multi-tier SAS environment like additional Webtier then you need to follow sequence within the SAS tiers. For example, basic SAS services (`sas.servers`) should be up and running fine before we start Webtier services. For LSF, there are multiple internal LSF services which needs to be executed in specific order.

This paper explains the automation of entire start/stop process through Linux scripting where you can predefine the services order. Linux script will take care of the services order automatically and you no need to run any command manually and wait for one services to be up before starting others.

SAS GRID SERVICES ARCHITECTURE

LSF SERVICES

LSF is the layer, which work for load balancing among the compute nodes defined in Grid cluster. LSF services should be up and running on all compute nodes to accept and execute the SAS jobs. LSF services are the one who control and keep track of load on all nodes. Based on the load and resource basic, LSF dispatch the specific job to specific node for execution. If LSF is down on any node, it won't accept any job but if LSF is down on master one it stops the entire SAS Grid environment completely. That is the reason Master (also known as control) node is very critical for SAS Grid. Master node need some extra services compare to other slave nodes, to monitor the entire Grid cluster. There is other set of common services which are necessary for all nodes.



SAS SERVICES

There is no as such any significant changes in the operational process for SAS services in Grid or non-Grid traditional SAS environment. In Grid environment, nodes should have both LSF and SAS services running to accept and process the SAS job. In multi node Grid environment, all SAS services should be running (with `sas.servers` script) on Master control server while other slave nodes need to have Object Spawner only. You no need to run `sas.servers` script on all nodes except master node.

SAS services depends on different SAS tiers as well. For example, if your Grid has Webtier then SAS services should be up before you start Webtier. The reason to start SAS service prior to Webtier is; Webtier usage SAS from compute nodes to execute the stored process and SAS web services. When we call Webtier, actual SAS processing takes place at compute nodes, not on Webtier host. So, SAS services should be ready to accept SAS request from Webtier.

If you have other SAS tiers as well, you need to figure out the sequence and dependency of different tiers between each other. Based on the dependency, you can mention the services start/ stop order in Linux script. This is one time activity to define the service order and then script will take care of all of the service steps at backend automatically.

Basic SAS Services

This is the SAS product installed on control server's SAS home. This is the basic SAS component for Grid which needs to be up before starting other SAS tiers. Since we use shared file system, we don't install the SAS software on each nodes separately. Generally master server is used to install SAS and this install works for entire Grid cluster. All nodes share the same binaries installed with initial deployment. But each node need to have SAS services up to execute SAS program on that node. These are the SAS services which use the SAS binaries from shared file system to execute the SAS job.

Master Node Services

Master node need to execute `sas.server` script to start the SAS services. It starts all the SAS services on master server.

```
/<ControlInstallHome>/sasconfig/Lev1/sas.servers
```

Slave Node Services

Slave nodes need to have Object spawner up and running to accept any SAS request. Object spawner is the responsible to execute SAS programs on compute nodes.

```
/<ControlInstallHome>/sasconfig/Lev1/ObjectSpawner/sas.servers
```

LSF SERVICES AUTIOMATION

PREREQUISIT

Following are the key requirement for LSF services automation:

1. **ROOT Access** - Most of the LSF services runs as ROOT. Automated script needs to be executed as ROOT, so you have unrestricted ROOT access on your Grid nodes. You can have some work around but having ROOT access will make the process easy and less problem for your automation.
2. **Keyless Self-Login** - There should be keyless authentication for ROOT between Grid nodes. ROOT should be able to re-ssh on same servers as well, so that you can start the script from any of the node in cluster.
3. **Bash Shell** - Below given scripts are written in Bash shell.

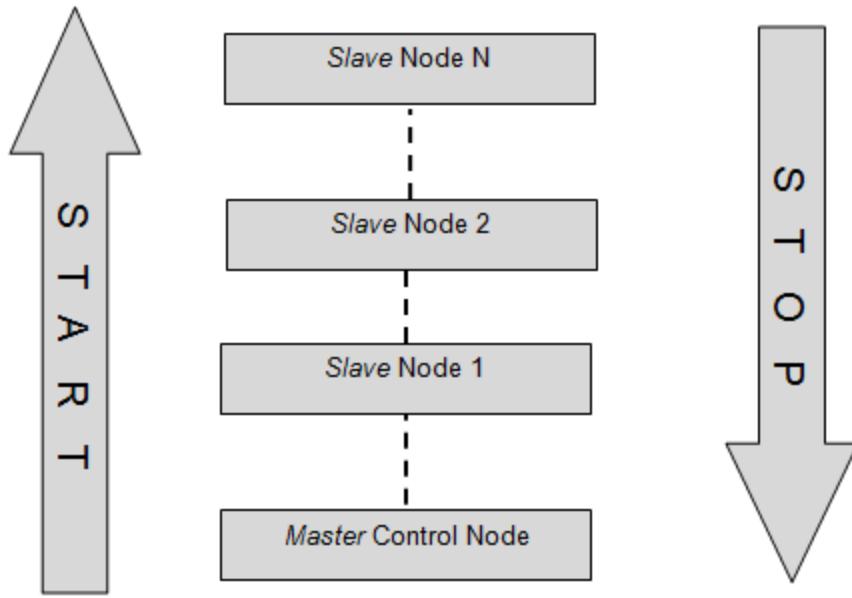


Figure 1. LSF Processes Order

LSF SERVICES START

To start the LSF services, we need to start the services from master control server. In below sample script, there are three nodes defined in the Grid cluster. Gridnode1 is the master one. Other two nodes are slave. As discussed earlier, there are two sets of services for LSF. First set of services should be started on all nodes before you start second set of services on master node. The second set of services helps master to have overall control over all the nodes in Grid cluster.

```
#!/bin/bash
## Node and variable definition
## node1 is master and node2, node3 are slave nodes

total_node=3
node1=gridnode1
node2=gridnode2
node3=gridnode3
lsf_conf=<lsf_home>/lsf/conf
## End of variable definition

for i in [1..3]
do
j=node$i
grid_node= `echo $echo $(!j)`

if [ $i -eq 1 ]
then
echo "$i. Starting lsf on node : $grid_node"
. $lsf_conf/profile.lsf
lsadmin limstartup
```

```

sleep 5
lsadmin resstartup
sleep 5
badmin hstartup
sleep 5

else
echo "$i. Starting lsf on node : $grid_node"
sudo -E ssh root@$grid_node '

lsf_conf=<lsf_home>/lsf/conf
. $lsf_conf/profile.lsf
lsadmin limstartup
sleep 5
lsadmin resstartup
sleep 5
badmin hstartup
sleep 5
exit
'

fi
done

if [ $i -eq $total_node ]
echo "Remaining lsf services (gabd and jfd) on master node"

/<lsf_home>/lsf/gms/gaadmin start
sleep 10
. /<processmanager_home>/processmanager/conf/profile.js
jaadmin start
sleep 10
fi
## End of script.

```

LSF SERVICES STOP

While stopping LSF services, you should follow the exact reverse order of what is used in above start-up script. In below script, the master node (gridnode1) is defined as last node (node3) while it was first node in start-up script. Because master node should be the last one to stop LSF after all slave nodes down. Below script the master node is the last sequence in loop to stop the LSF services. If you stop master service before slave nodes then communication between slave nodes and LSF is broken and services from slave nodes will not be stopped graciously.

```

#!/bin/bash
## Node and variable definition
## node3 is master and node2, node1 are slave nodes

total_node=3
node1=gridnode3
node2=gridnode2
node3=gridnode1
lsf_conf=<lsf_home>/lsf/conf
## End of variable definition

```

```

for i in [1..3]
do
j=node$i
grid_node= `echo $echo $(!j)`

if [ $i -lt $total_node ]
then
echo "$i. Stopping lsf on node : $grid_node"

ssh root@grid_node "
pltfmcomp=/<platformcomputing_home>

. $pltfmcomp/lsf/conf/profile.lsf
badmin hshutdown
sleep 5
lsadmin resshutdown
sleep 5
lsadmin limshutdown
sleep 5
exit
"
fi

if [ $i -eq $total_node ]
then
echo "$i. Stopping lsf on node : $grid_node"

ssh root@grid_node "
pltfmcomp=/<platformcomputing_home>

. $pltfmcomp/lsf/conf/profile.lsf
badmin hshutdown
sleep 5
lsadmin resshutdown
sleep 5
lsadmin limshutdown
sleep 5

##Stopping gabd and jfd on master node.
## This may take around a minute to stop these processes.
$pltfmcomp/lsf/gms/bin/gaadmin stop
sleep 10

. $pltfmcomp/processmanager/conf/profile.js
jaadmin stop
sleep 10
exit
"
fi
done
## End of script.

```

SAS SERVICES AUTOMATION

First, we need to list all the SAS tiers installed with SAS Grid platform. Based on different tiers we can define the service dependency between different tiers. Then we can mention the same order for start/stop commands in below given sample script. But irrespective of SAS tiers installed, SAS Metadata should be the first tier which needs to be up and running before we start any other tier. SAS Metadata is responsible for most of the authentication before launching SAS session for other SAS tiers.

PREREQUISIT

SAS installer id should have keyless SSH access on all the Grid nodes defined in the cluster. It should have keyless self-login access as well, so that we can execute the automated script from any of the node and it should take care of entire Grid cluster.

SERVICE START

While starting SAS services in SAS Grid environment you need to make sure that Metadata tier is started as first step. This is the first tier where our startup process should start from. Until Metadata is not completely up, you cannot start other tiers because all authentication takes place at Metadata server. Once Metadata is up, you can start other tier dependency and start accordingly. Below is the flow chart for SAS services start in Basic SAS Grid environment. This shows how the automated script starts the Grid environment in sequential mechanism.

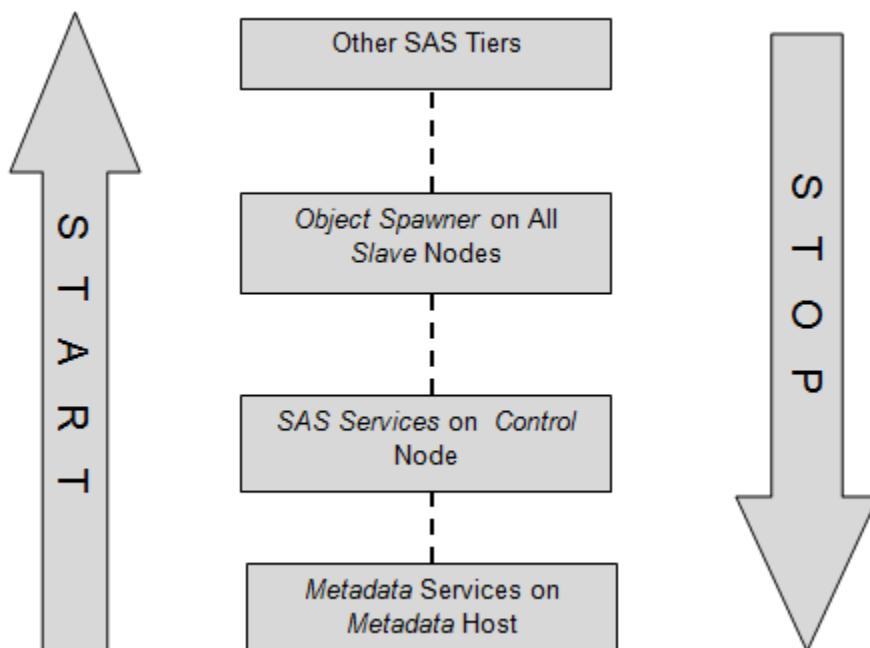


Figure 2. Multi-Tier SAS Services Order

Below is the sample script which can be used to start the SAS services in SAS Grid environment. This script is created to start the services on Metadata server and compute nodes (where SAS processing takes place). In this script, we have considered the different nodes as different tiers. Once metadata server is up, script starts the SAS services in control server.

SAS Grid uses the shared file system, so all the nodes use same binaries to start the services on all nodes. After starting SAS services on control server, ObjectSpawner needs to be started on slave nodes.

ObjectSpawner would be sufficient to execute the SAS job on slave nodes. If you have more nodes, you can keep on adding extra nodes as additional tiers in the script.

```
#!/bin/bash
## Tier, Host and Variables Definition
tiers=3
tier1='Metadata'
host1=<Metadata_host>
tier2='Master'
host2=<Master_Node_host>
tier3='Slave'
host3=<Slave_ObjectSpawner_host>
##
for i in {1..3}
do
j=tier$i
tier_name=`echo $echo ${!j}`
k=host$i
host=`echo $echo ${!k}`
echo "Tier is : $tier_name and associated host is : $host"

if [ $tier_name == "Metadata" ]
then
ssh <SASInstallerID>@$host \
sasconfig=/<SASHomeMetadata>/sasconfig/Lev1
# Starting Metadata Server
$sasconfig/sas.servers start
'
else
if [ $tier_name == "Master" ]
then
ssh <SASInstallerID>@$host \
sasconfig=/<SASHomeControl>/sasconfig/Lev1
# Starting SAS on control server
$sasconfig/sas.servers start
'
else
ssh <SASInstallerID>@$host \
sasconfig=/<SASHomeControl>/sasconfig/Lev1/ObjectSpawner
# Starting Objectspawner on slave nodes
$sasconfig/ObjectSpawner.sh start
'
fi
fi
done
## End of script.
```

SERVICE STOP

We discussed how can we start the SAS services automatically in previous topic. Before writing new script for stopping SAS services, you need to understand the concept of tiers dependency. Like start-up script, if you make few changes in above script (to start the SAS services), it can be used to stop the SAS services. There should be few logical changes, as below:

1. Change the tiers order exactly opposite to what it was mention in start-up script.
2. During stopping the SAS services, Metadata tier should be the last tier to stop.
3. Change the `sas.servers` (on control server) and `ObjectSpawner.sh` (on slave nodes) parameter from `start` to `stop`.

```
#!/bin/bash
## Tier, Host and Variables Definition
tiers=3
tier3='Metadata'
host3=<Metadata_host>
tier2='Master'
host2=<Master_Node_host>
tier1='Slave'
host1=<Slave_ObjectSpawner_host>
##

for i in {1..3}
do
    j=tier$i
    tier_name=`echo $echo ${!j}`
    k=host$i
    host=`echo $echo ${!k}`
    echo "Tier is : $tier_name and associated host is : $host"

    if [ $tier_name == "Metadata" ]
        then
            ssh <SASInstallerID>@$host \
            sasconfig=/<SASHomeMetadata>/sasconfig/Lev1
            $sasconfig/sas.servers stop
    '

    else
        if [ $tier_name == "Master" ]
            then
                ssh <SASInstallerID>@$host \
                sasconfig=/<SASHomeControl>/sasconfig/Lev1
                $sasconfig/sas.servers stop
    '

        else
            ssh <SASInstallerID>@$host \
            sasconfig=/<SASHomeControl>/sasconfig/Lev1/ObjectSpawner
            $sasconfig/ObjectSpawner.sh stop
    '
    fi
fi
done
## End of script.
```

CONCLUSION

Manual operation of Grid platform is very time consuming and error prone task for SAS administrators. Automating such activities not only help to manage entire platform smoothly and efficiently, in fact it helps to increase the platform availability for business users. Linux scripts given in the paper provides the basic understanding and high level concept for operation activities automation. This also explains how can these scripts be used to automate additional SAS tiers. Automation of big cluster operation is very cost effective values for the organizations.

REFERENCES

Singh, Piyush. and Pohl, Gerhardt M. Mar-2014. "Enhancing SAS® Piping Through Dynamic Port Allocation" *Proceedings of SAS Global Forum 2014*. Washington, DC, USA: SAS Institute. Available at: <http://support.sas.com/resources/papers/proceedings14/1826-2014.pdf>

Sangwan, Prasoon. Gupta, Tanuj. and Singh, Piyush. Apr-2016. "Key Requirements for SAS® Grid Users" *Proceedings SAS Global Forum 2016*. Las Vegas, NV, USA: SAS Institute. Available at: <http://support.sas.com/resources/papers/proceedings16/7140-2016.pdf>

Singh, Piyush. Sangwan, Prasoon. Kumar, Veera. and Sanger, Todd. Nov-2016. "SAS® Enterprise Guide for SAS® Visual Analytics LASR Server" *Proceedings of South Central SAS User Group 2016*. San Antonio, TX, USA: SCSUG. Available at: <http://www.scsug.org/wp-content/uploads/2016/11/SAS%C2%AE-Enterprise-Guide-for-SAS%C2%AE-Visual-Analytics-LASR-Server.pdf>

RECOMMENDED READING

- *Grid Computing in SAS® 9.4, Third Edition*
- *Platform LSF, Version 9 Release 1.3*

CONTACT INFORMATION

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

Piyush Singh

piyushkumar.singh@tcs.com

Ghiyasuddin Mohammed Faraz Khan

ghiyasuddin.farazkhan@tcs.com

Prasoon Sangwan

prasoon.sangwan@tcs.com