

Introducing - SAS® Grid Manager for Hadoop

Cheryl Doninger, SAS Institute Inc., Cary, NC

Doug Haigh, SAS Institute Inc, Cary, NC

ABSTRACT

Organizations view Hadoop as a lower cost means to assemble their data in one location. Increasingly, the Hadoop environment also provides the processing horsepower to analyze the patterns contained in this data to create business value. SAS® Grid Manager for Hadoop allows you to co-locate your SAS® Analytics workload directly on your Hadoop cluster via an integration with the YARN and Oozie components of the Hadoop ecosystem. YARN allows Hadoop to become the “operating system” for your data; it is a tool that manages and mediates access to the shared pool of data, and manages and mediates the resources that are used to manipulate the pool. Learn the capabilities and benefits of SAS Grid Manager for Hadoop as well as some configuration tips. In addition, sample SAS Grid jobs are provided to illustrate different ways to access and analyze your Hadoop data with your SAS Grid jobs.

INTRODUCTION

Customers that want to co-locate analytic workload on their Hadoop clusters typically, if not always, require an integration of the workload with YARN. YARN is a resource orchestrator that makes a Hadoop cluster a better shared environment for running multiple analytic tools. SAS Grid Manager for Hadoop was created specifically for those customers who wish to co-locate their SAS Grid jobs on the same hardware used for their Hadoop cluster. It provides integration with YARN and Oozie such that the submission of any SAS Grid job is under the control of YARN. Figure 1 shows a high level architecture view of a SAS Grid Manager for Hadoop deployment.

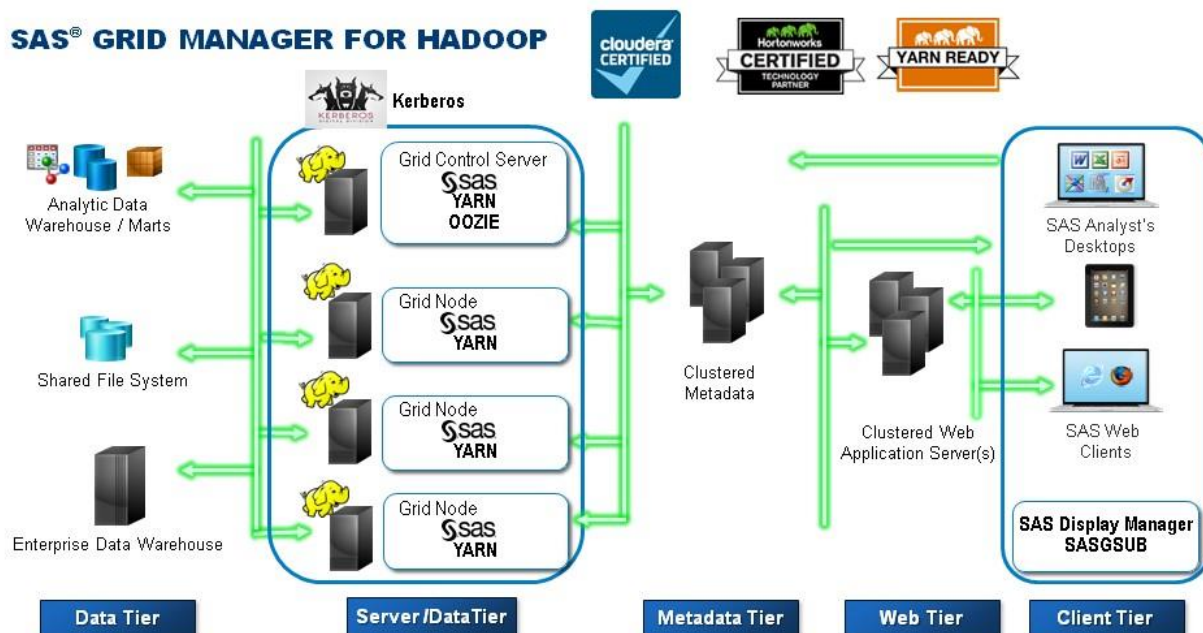


Figure 1. SAS Grid Manager for Hadoop High Level Architecture

A Hadoop cluster must meet several requirements in order to deploy SAS Grid Manager for Hadoop, including:

- Use of one of the enterprise Hadoop distributions supported by SAS Grid Manager for Hadoop (more details provided [here](#)):
 - Cloudera 5.2.x and higher
 - Hortonworks 2.1.x and higher
 - MapR 4.0.x and higher
- Kerberos must be configured on the cluster. This is required by YARN in order to launch a process as the user who submitted the request.

Additionally, if you want to deploy SAS Grid Manager co-located with all or some subset of the Hadoop data nodes then your cluster must also meet the following requirements:

- The cluster must have spare capacity to accommodate the introduction of additional SAS Grid workload.
- In addition to being data nodes, the nodes in the cluster must be architected to be compute nodes as well. This means appropriate number of CPUs and appropriate amount of RAM.

If the above two characteristics cannot be met and you still want to have your SAS Grid jobs running in your Hadoop cluster to avoid movement of data outside of the Hadoop cluster proper, then you could have a set of nodes that would be compute nodes only and not data nodes but still part of the overall Hadoop cluster. This is accomplished by running the YARN Node Manager (NM) on each of these nodes but they would not be data nodes. This provides the following advantages:

- Allows the data nodes to be pure data nodes
- Allows you to scale your compute and data nodes independently.
- Allows you to architect your compute and data nodes independently.

Notice that there is still a small, much reduced need for a shared file system even with SAS Grid Manager for Hadoop. This is because some SAS Grid features as well as other SAS product features, when running with SAS Grid, require a shared file system. This would include the use of SAS checkpoint/restart so that the SASWORK directory is on shared storage, the use of SASGSUB without the need to stage files and the use of SAS products like SAS Enterprise Miner that require shared storage to store projects and files when run in a SAS Grid. However, if all (or most) of the input and output data is stored in HDFS, the need for a shared file system is greatly reduced. Therefore, it is possible that NFS could meet the need for a shared file system with SAS Grid Manager for Hadoop. This would be determined during a SAS Grid Architecture Workshop to plan the architecture of your SAS Grid deployment.

HOW DOES IT WORK?

For the workload management capabilities, SAS Grid Manager integrates with YARN. As stated earlier, YARN is more of a resource orchestrator. The following diagram illustrates the steps involved from the time that a client submits a job to the SAS Grid to the time that the job starts to run under the control of YARN in the cluster.

SAS GRID MANAGER FOR HADOOP

INTEGRATION WITH YARN

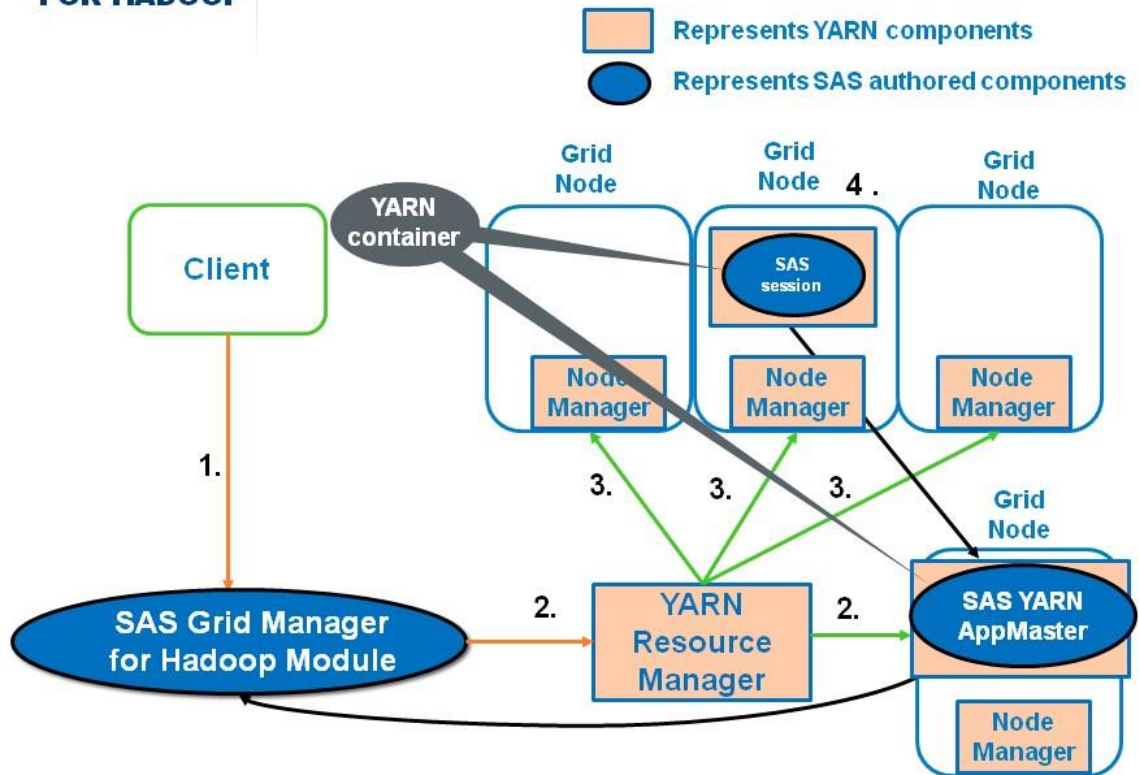


Figure 2. SAS Grid Manager for Hadoop Integration with YARN

The following steps correspond to the numbers in Figure 2 above:

1. A SAS client submits a SAS job (SASGSUB, SAS/CONNECT, grid-launched workspace server) to the SAS Grid Manager for Hadoop module.
2. The SAS Grid Manager for Hadoop module communicates with the YARN resource manager to request a container and to invoke the SAS YARN application master.
3. The SAS YARN application master requests a YARN container for the SAS Grid job with the specified resources. The resources are specified in a grid policy file read by SAS Grid Manager for Hadoop. (The grid policy file will be discussed in more detail later in the paper.) When the YARN Resource Manager determines which node has the requested resources available, the application master requests a YARN container on that grid node.
4. YARN runs the command it was given to invoke SAS under the control of the YARN container.
5. Once the SAS Grid job has started, the remaining SAS Grid behavior is unchanged. YARN knows which resources are used by the SAS Grid job, allowing you to build a multi-tenant or shared Hadoop cluster.

In order to create a process flow diagram (PFD) of one or more SAS jobs and schedule that flow to run based on a trigger, SAS Grid Manager integrates with Oozie. The following diagram illustrates the steps involved from the time that a flow is scheduled using the Schedule Manager plug-in in SASMC to the time

that the flow is triggered and each job runs as a MapReduce (MR) job which is under the control of YARN in the cluster.

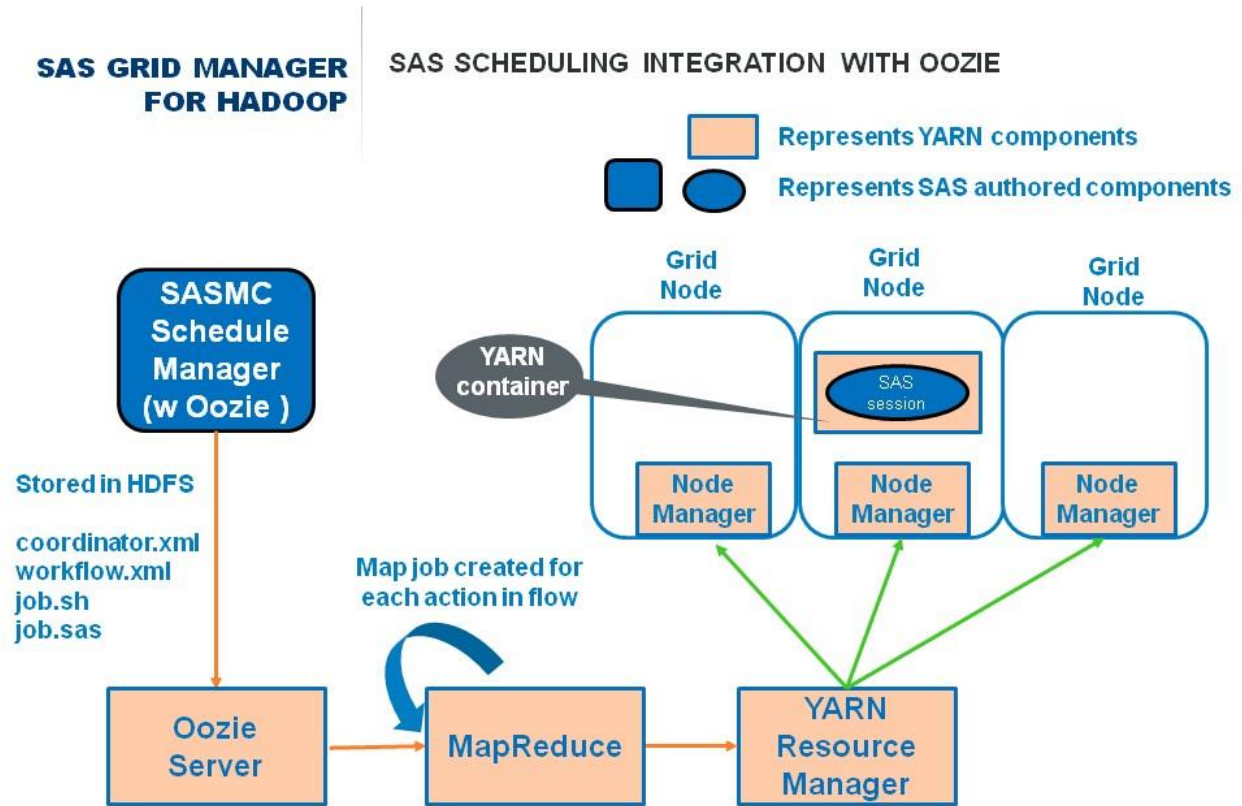


Figure 3. SAS Grid Manager for Hadoop Integration with Oozie

1. The SAS Schedule Manage plug-in in SASMC had been enhanced to support deployment and scheduling of SAS jobs using Oozie. This support has been added to the data step batch server only for .sas files.
2. Deploy step
 1. Schedule manager writes the .sas file (job.sas for example) to HDFS or a shared location (deployment directory defined at config time as usual)
3. Schedule step
 1. Flow metadata is transformed into Oozie XML and written to HDFS. This includes
 1. coordinator.xml
 2. workflow.xml
 3. job.sh
 2. The Oozie Server is called via a REST api to schedule job

4. When the trigger occurs, the Oozie server creates a map job for each action (.sas file) that is in the flow.
5. MapReduce interacts with YARN as usual to get a container on a node and then run the job.sh script which results in SAS running in a YARN container. If there are parallel actions in the flow, there would be multiple SAS jobs running across the cluster each in its own container.

WHAT BEHAVIOR CAN I EXPECT?

When SAS Grid Manager was originally architected, it was designed to be able to support different grid providers. The two providers supported by SAS Grid Manager today are the Platform Suite for SAS and the Hadoop YARN and Oozie components. While there are differences in the way the grid providers behave, the ways that jobs can be submitted to SAS Grid is consistent across all providers. This means that all of the methods of submitting work to a SAS Grid work the same way and have the same syntax. The submission methods include:

- Grid launched workspace and stored process servers
- Grid servers created with SAS/CONNECT syntax
- Batch submission with SASGSUB
- Batch submission with the Schedule Manager plug-in in SASMC

The fact that the above methods are supported and unchanged means that all of the SAS Grid integration that has been done with other SAS products and solutions, such as SAS Enterprise Guide, SAS Studio, SAS Data Integration Studio, SAS Enterprise Miner and others, also works in the same way with SAS Grid Manager for Hadoop. In addition, all of the usual SAS Grid related metadata applies as well. This includes things like grid logical server definition, grid options sets, and the data step batch server definition.

The following SAS Grid job will run successfully regardless of the provider being used by SAS Grid Manager.

```
/* assume metadata option values already set */  
  
%put rc=%sysfunc(grdsvc_enable(_ALL_, server=SASApp; jobname="SAS Grid  
Job"));  
signon grid;  
rsubmit grid;  
  data a; x=1; run;  
endrsubmit;  
signoff;
```

HOW WOULD I ADMINISTER THE ENVIRONMENT?

As previously stated, SAS Grid Manager for Hadoop was designed to run co-located on a shared Hadoop cluster. As such you would use the Hadoop management and monitoring tools for your respective distribution of Hadoop to administer your SAS jobs as well.

SAS Grid Manager for Hadoop also supports a grid policy file. The grid policy file is an xml file that is used to define different types of applications that will run on the Hadoop cluster and also associate resource information with each application type. The types of resources would include things such as the priority of the job, amount of memory required, number of virtual cores required, which queue the job should use, the hosts that can be used, as well as other types of resources. Each application type would have a unique name and that name would be associate with the job being submitted using any of the following methods:

- In the Grid Options field of the logical grid server definition
- In the jobOpts= parameter of the grdsvc_enable function
- As part of a grid options set
- As part of the –GRIDJOBPTS parameter on SASGSUB

A sample Grid Policy XML file follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<GridPolicy defaultAppType="normal">
  <GridApplicationType name="normal">
    <priority>20</priority>
    <nice>10</nice>
    <memory>1024</memory>
    <vcores>1</vcores>
    <runlimit>120</runlimit>
    <queue>normal_users</queue>
    <hosts>
      <host>myHost1.mydomain.com</host>
      <hostGroup>development</hostGroup>
    </hosts>
  </GridApplicationType>
  <GridApplicationType name="priority">
    <jobname>High Priority!</jobname>
    <priority>30</priority>
    <nice>0</nice>
    <memory>1024</memory>
    <vcores>1</vcores>
    <runlimit>120</runlimit>
    <queue>priority_users</queue>
    <hosts>
      <host>myHost2.mydomain.com</host>
    </hosts>
  </GridApplicationType>
  <HostGroup name="development">
    <host>myHost3.mydomain.com</host>
    <host>myHost4.mydomain.com</host>
  </HostGroup>
</GridPolicy>
```

WHAT IS DIFFERENT?

It is worth repeating that the behavior of submitting SAS Grid jobs to SAS Grid Manager for Hadoop is the same as it is with SAS Grid Manager using the Platform Suite for SAS. However there are several differences from the monitoring, configuration and provider functionality perspective. These differences include:

- The SAS Grid provider, in this case one of the supported Hadoop distributions, is not packaged with SAS Grid Manager for Hadoop. You must license this from your vendor of choice and work with that vendor to get Hadoop installed and configured.
- The monitoring and management of the SAS jobs is done through the tools specific to the Hadoop distribution being used, rather than through SAS Environment Manager. This is because it is expected that the SAS Grid jobs are sharing the cluster with other types of jobs and the cluster administrator is already using Hadoop tools to manage all jobs.

- Kerberos is required to be configured for SAS Grid Manager for Hadoop to interact with YARN to have jobs launched with the identity of the user submitting the work. This also means that Kerberos will have to be configured to interact successfully with any other SAS functionality being leveraged in the SAS Grid jobs such as SAS/ACCESS to Hadoop.
- The need for a shared file system is greatly reduced because the majority, if not all, input and output data will be stored in HDFS. There is still a need for a smaller shared file system as described previously in the “Introduction” section.
- It is important to understand that SAS data sets (sas7bdat) cannot be written or stored in HDFS. Therefore, if you have existing data it will need to be migrated to HDFS. We have found that SPDE provides good performance for moving data into HDFS and reading, writing and updating from your SAS Grid jobs.
- YARN is a resource orchestrator of static resources and does not dynamically monitor and adjust resource allotments over time. For a more detailed explanation of the considerations for using YARN to manage multiple workloads, see the two papers listed in the “References” section.
- It is also important to understand that each SAS Grid job submission results in the creation of (at least) two YARN containers – one to run the SAS YARN Application Master and one to run the actual SAS Grid job. (If the SAS Grid job contains SAS functionality that is also integrated with YARN, additional containers will be needed.) On heavily loaded clusters, this can result in deadlock if there are many concurrent job submissions which result in multiple AppMasters waiting for resources consumed by other AppMasters.

CONCLUSION

Today, as consumers, we have a choice between using a gas powered car or an electric car. In either case the way we use each car is the same – we open a door to get in, we use a steering wheel to steer, we use a GPS to navigate, etc. However, the engine that powers the car has many differences. Such is the case with SAS Grid Manager. Regardless of the provider that is being used the operation of sending SAS jobs to the Grid is the same. The differences will lie with the performance and design of the grid providers.

SAS Grid Manager for Hadoop, along with many of the other SAS technologies, provides the integration with YARN to enable SAS Grid jobs to run under the management of YARN in a shared Hadoop cluster. SAS Grid Manager for Hadoop is an important addition to the options for deploying SAS Grid. It is no longer a question of which options are possible but which option meets the goals and requirement of your organization.

APPENDIX 1

The following SAS Grid Manager for Hadoop job illustrates not only the `signon` and `rsubmit` syntax that will submit a job to the grid, it also uses many of the other SAS technologies that are integrated with Hadoop and YARN.

```
/* assume metadata option values already set */
%put rc=%sysfunc(grdsvc_enable(_ALL_, server=SASApp; jobname="SAS Hadoop
Job"));
signon hadoop;
rsubmit hadoop;
options set=SAS_HADOOP_RESTFUL=1;

proc options option=set; run;
```

```

/*****
** Use PROC HADOOP to create directory
*/
proc hadoop verbose;
  hdfs mkdir='/tmp/userid/spde';
  hdfs ls='/tmp';
  run;

/*****
** Use SPDE engine to play in directory
*/
libname dblib spde "/tmp/userid/spde" hdfs=yes;

data dblib.testout;
  x=1;
  run;

data _null_;
  set dblib.testout;
  run;

data _null_;
  set dblib.testout(where=(x=1));
  run;

libname dblib clear;

/*****
** Use PROC HADOOP to delete directory
*/
proc hadoop verbose;
  hdfs delete='/tmp/userid/spde';
  run;

/*****
** Play with HADOOP engine
*/
libname x hadoop server='host.xxx.yy.zzz';

proc delete data=x.test;
  run;

data x.test;
  x=1;
  c='str';
  run;

data _null_;
  set x.test;
  run;

```



```

data _null_;
  set x.test(where=(x=1));
  run;

/*****
** Play with DS2 to exercise Embedded Process (EP) on the cluster
*/

proc delete data=x.sasdemoa;run;
data x.sasdemoa; do x=1 to 100000; a='iuehgruiehgouqehrgohe'; output;
output; end; run;

proc delete data=x.sasdemob;run;
data x.testby;do x=1 to 10000;z='hellohadoop';output;end;run;

proc ds2 indb=yes ;
      thread threadby /overwrite=yes;
method run();
  set x.testby;
end;
endthread;
run;

/* ds2_options trace ; */
data x.sasdemob (overwrite=yes keep=(tot a)) ;
dcl thread threadby thb;
dcl double tot;
retain tot;
method run();
set from thb threads=2;
end;
enddata;
run;
quit;

proc print data=x.sasdemob(obs=10);run;
endrsubmit;

signoff;

```

REFERENCES

“SAS Analytics on Your Hadoop Cluster Managed by YARN.” July 14, 2015. Available at <http://support.sas.com/rnd/scalability/grid/hadoop/SASAnalyticsOnYARN.pdf>.

“Best Practices for Resource Management in Hadoop.” April, 2016. Available at <http://support.sas.com/rnd/scalability/grid/hadoop/BPResMgmt.pdf>.

SAS Institute Inc. “Introduction to SAS Grid Computing.” Available at <http://support.sas.com/rnd/scalability/grid/>.

“Apache Hadoop Yarn.” The Apache Software Foundation. June 29, 2015. Available at <https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html>.

CONTACT INFORMATION

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

Cheryl Doninger
Senior Director, SAS R&D
Cheryl.doninger@sas.com

Doug Haigh
Principal Software Developer
Doug.haigh@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.