Paper SAS1117-2017

Introduction to Configuration and Management for SAS® Grid Manager for Hadoop

Mark Lochbihler, Hortonworks Inc., Santa Clara, CA

ABSTRACT

"How can we run traditional SAS® jobs, including SAS workspace servers on Hadoop worker nodes?" The answer is SAS® Grid Manager for Hadoop. It has been integrated with the Hadoop ecosystem to provide resource management, high availability and enterprise scheduling for SAS customers. This paper will provide an introduction architect, configuration and management of SAS Grid Manager for Hadoop. Anyone involved with SAS and Hadoop should find the information in this paper useful. The first area to be covered will be a break down each required SAS and Hadoop component. From the Hadoop ecosystem, we will define the role of YARN Compute, HDFS Storage, and Hadoop Client services. We will review SAS metadata definitions for SAS Grid Manager, Object Spawner and Grid Workspace Servers. We will cover required Kerberos security, as well as SAS Enterprise Guide and the SASGSUB utility. YARN queues and the SAS Grid Policy file for optimizing job scheduling will be reviewed. And finally, we will discuss traditional SAS math running on a Hadoop Worker node, and how it can take advantage of SAS High Performance Math to accelerate job execution. By leveraging SAS Grid Manager for Hadoop, sites are moving SAS jobs inside a Hadoop Cluster. This will ultimately cut down on data movement and provide more consistent job execution. Although this paper is written for the SAS and Hadoop Administrators, SAS Users will also benefit from this session.

WHAT IS SAS GRID COMPUTING?

SAS Grid Computing has been offering SAS shops a lower cost, shared, multi-tenant, high performing computing environment to meet their advanced analytic and modeling needs. By implementing a SAS Grid, SAS administrators can centralize individual and or departmental SAS computing environments onto a SAS Compute Grid and better utilize IT resources, provide high availability and accelerated processing. A SAS Grid runs on two or more SAS Grid Compute Nodes. Each SAS Grid Compute Node is a candidate to execute SAS jobs submitted into a Grid queue by SAS user groups at a site.

ENTER HADOOP AND YARN

The benefits of SAS Grid Computing are **not new** to the SAS User Community. It has been successfully implemented and running in production at thousands of customers' sites around the world for well over a decade and provides significant benefits. What's new with SAS Grid Manager for Hadoop is that it offers YARN as an orchestration option in addition to the existing and well-proven use of the Platform Suite for SAS which includes LSF. SAS Grid Manager for Hadoop was designed to enable customers to co-locate their SAS Grid and associated SAS workloads on their new or existing SAS Hadoop clusters.

WHY MOVE SAS JOBS INSIDE HADOOP?

A decision to develop this type of solution typically starts by identifying business needs and associated usage cases to support these needs. Any customer interested lowering overall SAS storage and server costs, while at the same time consolidating and centralizing departmental SAS datasets is a candidate. An initial step can be to migrate selected SAS Workload data, including raw inbound and outbound files, SAS libraries and other RDBMS SAS data sources to Hadoop Storage. Just by taking the first step to move SAS storage to Hadoop, organizations can save over 50% in annual SAS storage costs. For new model develop or existing model optimization efforts that plan on using IoT Hadoop data sources (including Sensor, Click Stream, Web Log, Machine and IoT device), then moving existing SAS datasets to Apache Hive and Hadoop Storage (HDFS) could yield significant cost savings. To access these migrated SAS datasets, users can turn to SAS® Access to Hadoop. It offers a SAS libname engine for HDFS, as well as Hive, and a filename statement to HDFS. Once the data is migrated, carefully selected SAS Workloads can be moved onto Hadoop Worker nodes and SAS users can transparently leverage SAS Grid Manager for Hadoop to run their daily SAS jobs.

Once the decision has been made to move new and or existing SAS data and workloads to Hadoop, and leverage SAS Grid Manager, it is highly recommended that your site invest up front in SAS and Hadoop administrative training and professional services. It is also critical that you have a detailed understanding of how YARN configuration and scheduling works on Hadoop. In addition, involving SAS and Hortonworks Professional services during the project startup phase in important.

HOW IT WORKS

YARN 101

If you are new to Hadoop and YARN, let me provide a bit of background. YARN is the orchestration engine for Hadoop 2.x. Figure 1 below is a high-level view of an Enterprise Data Lake. With YARN as the as the central orchestrator and operating system for Hadoop, sites can run multiple Batch, Interactive and Real time compute engines within the same Hadoop cluster (including SAS Grid Manager as indicated by the arrow). Hadoop has been offering a Low Cost, Massive Scale Storage and Compute Architecture for over a decade. With SAS Grid Manager for Hadoop, SAS users can run traditional SAS jobs on Hadoop Worker nodes, inside the cluster.

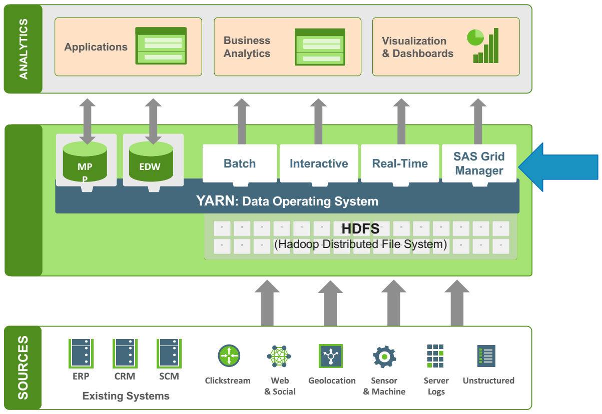


Figure 1: A Hadoop Cluster running Batch, Interactive and Real Time Engines, including SAS Grid Manager.

With SAS Grid Manager for Hadoop, a community of SAS users transparently leveraging SAS Clients and submit interactive and batch SAS jobs to the SAS Grid Computing infrastructure on Hadoop. These jobs are scheduled by YARN based on queues and site policies to run on an optimal SAS Grid Compute Node (Hadoop Worker Node). Below is a Conceptual View of the architecture:

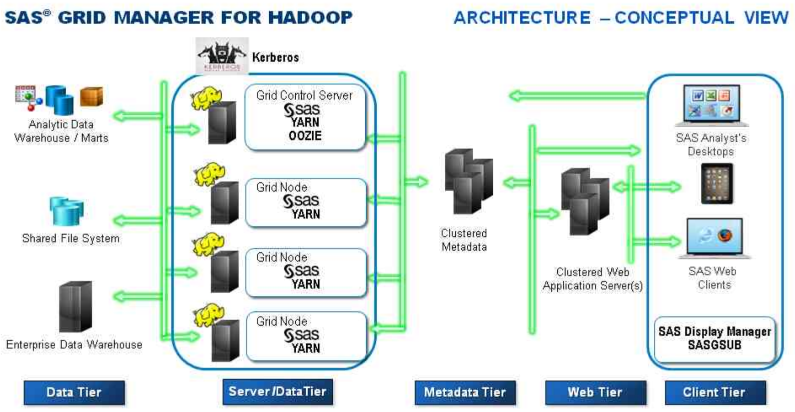


Figure 2: SAS Grid Manager for Hadoop Conceptual Architecture
(*Reference: SAS Grid Manager for Hadoop)*

BREAKING DOWN THIS SAS ARCHITECTURE ON HADOOP

With SAS Grid Manager for Hadoop, SAS jobs are scheduled by YARN's Resource Manager into a cluster running inside of the Hadoop firewall. SAS jobs can request additional Hadoop resources after they have been initially launched. This new architecture can reduce the complexity of configuration by simplifying port mapping between SAS jobs and the services they will need to complete. In this deployment model, SAS math is running closer to the Hadoop data and no longer requires negotiation with the Hadoop firewall to access, if requested, additional Hadoop and SAS High Performance service ports.

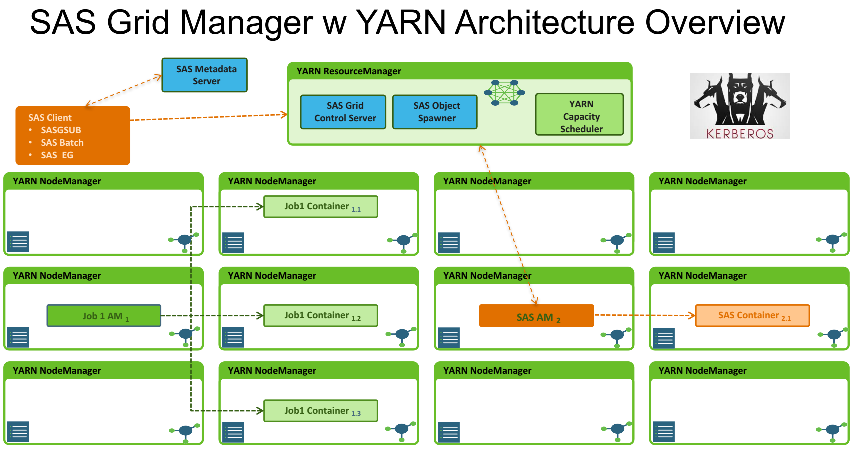


Figure 3: SAS Grid Manager for Hadoop w YARN Architecture Overview

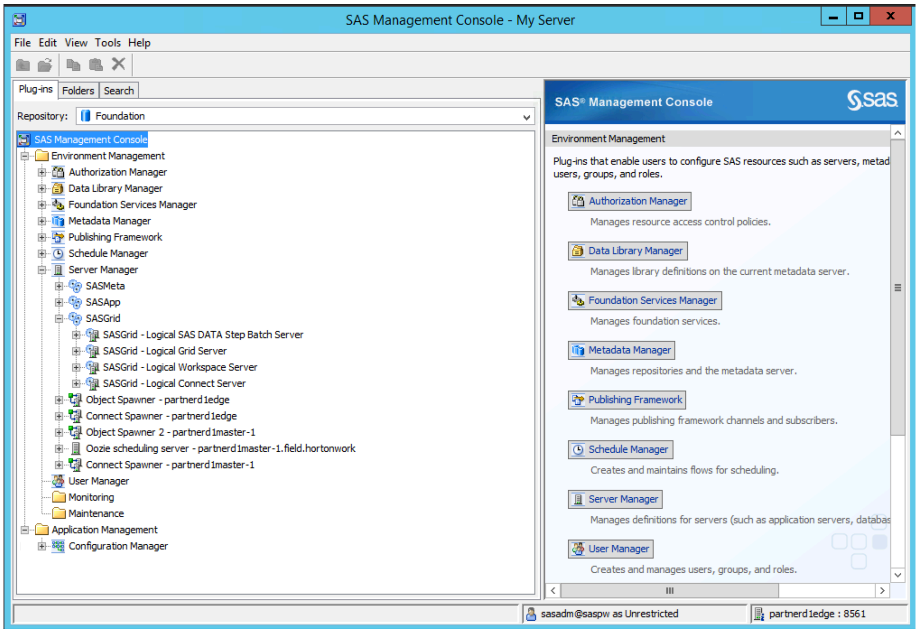
The YARN Resource Manager's role in Figure 3 above is to determine the most optimal Hadoop Worker Nodes to run a traditional SAS job or Workspace Server in the Hadoop cluster. In this illustration, you can see two YARN jobs already running on the cluster. Each of these Hadoop jobs has a single YARN Application Master (AM) Container assigned. Job1 is not a SAS job. It consists of AM1 along with three additional Job Task Containers (C1.1, C1.2, and C1.3). There is also a SAS job running on the same cluster. This SAS job has its own Application Master (SAS AM1) and one SAS Job Task Container (SAS C2.1). From an architecture and administrative perspective, lets take a look at the primary SAS and Hadoop services that will be configured.

KEY SAS GRID ARCHITECTURE COMPONENTS

SAS Metadata Server – A SAS service supporting, among other objects, the logical to physical mapping of SAS Logical Servers to YARN. In our examples, we will be using the local SAS Server, SASGrid.

SAS Grid Control Server – A SAS service running on the YARN Resource Manager node. It is called by SAS clients to communicate with YARN Resource Manager to negotiate resources for SAS jobs.

SAS Object Spawner – A SAS Service running on the YARN Resource Manager node. It is used to launch SAS contains with YARN.



```
SAS Management Console
  Environment Management
    Authorization Manager
    Data Library Manager
    Foundation Services Manager
    Metadata Manager
    Publishing Framework
    Schedule Manager
    Server Manager
      SASMeta
      SASApp
      SASGrid
        SASGrid - Logical SAS DATA Step Batch Server
        SASGrid - Logical Grid Server
        SASGrid - Logical Workspace Server
        SASGrid - Logical Connect Server
      Object Spawner - partnerd1edge
      Connect Spawner - partnerd1edge
      Object Spawner 2 - partnerd1master-1
      Oozie scheduling server - partnerd1master-1.field.hortonwork
      Connect Spawner - partnerd1master-1
    User Manager
    Monitoring
    Maintenance
  Application Management
    Configuration Manager
```

Environment Management

Plug-ins that enable users to configure SAS resources such as servers, metad users, groups, and roles.

Authorization Manager
Manages resource access control policies.

Data Library Manager
Manages library definitions on the current metadata server.

Foundation Services Manager
Manages foundation services.

Metadata Manager
Manages repositories and the metadata server.

Publishing Framework
Manages publishing framework channels and subscribers.

Schedule Manager
Creates and maintains flows for scheduling.

Server Manager
Manages definitions for servers (such as application servers, databas

User Manager
Creates and manages users, groups, and roles.

sasadm@saspw as Unrestricted | partnerd1edge : 8561

Figure 4: View of SAS Management Console, with expanded SASGrid Logical Server.

SAS Clients – includes SAS batch jobs, SASGSUB (a batch grid utility), and interactive Clients like SAS Enterprise Guide. SAS Clients will "Connect" and "Disconnect" from a SAS Logical Server, like SASGrid, defined in SAS Metadata and shown above in Figure 4.

KEY HADOOP INTEGRATION POINTS FOR SAS GRID

YARN Resource Manager – A Hadoop YARN Master Service responsible for controlling global Hadoop cluster resource usage. Resource Manager enables multi-tenancy and SLAs. It is also responsible for monitoring Node Manager State, submitting Application Master requests, verifying container launch and monitoring Application Master state.

YARN Node Manager – This Hadoop YARN Worker Node Service manages local resources on behalf of the requesting service. It also tracks node health and communicates status to the Resource Manager.

YARN Capacity Scheduler – A Hadoop YARN service, which can be configured to provide Job Scheduling policies for SLAs, Users, Groups, and Resources.

Hadoop Data Nodes – Hadoop Distributed File System (HDFS) storage nodes.

Kerberos Service – The Hadoop cluster must be Kerborized.

HADOOP MASTER NODE DECISIONS

A SAS Grid Control Server and SAS Object Spawner must be deployed on the same Hadoop Master Node as the YARN Resource Manager.

HADOOP WORKER NODE DECISIONS

SASHOME and SASCONFIG

For each Hadoop Worker Node which is a candidate to run SAS jobs must be configured so that SASHOME and SASCONFIG are available.

SASWORK and SASUTIL

It is critical that each Hadoop Worker node which is a candidate to run SAS jobs is configured correctly. A large part of the I/O required when running SAS analytics is to the scratch or temporary locations of SASWORK and SASUTIL. SAS required I/O throughput for these file systems, to provide the necessary performance to a heavily loaded system, is 100MB/sec/core. Adequate sizing for SASWORK is also necessary.

Traditional Storage and Compute verse Compute Only Worker Nodes

Within Hadoop, it is a common practice to have dual purpose worker nodes which run math or programs near on the same nodes where the Hadoop data resides. With Hadoop 2.x, the concept of dedicated Compute Only Hadoop Worker Nodes is an option. For SAS Grid Manager for Hadoop, both options are an option. For Compute Only, these Hadoop Worker Nodes will no longer host the required services and data for HDFS, giving more computing resources dedicated to the programs running on these nodes. The tradeoff for Compute Only Hadoop Nodes is the loss of HDFS data locality. Your sites SAS workload requirements will determine which type of Worker Nodes to deploy for SAS Grid Manager for Hadoop.

REAL WORLD CONFIGURATION EXAMPLE

In this section, we will walk through a SAS Grid Manager for Hadoop configuration exercise. For this deployment, our Hadoop Cluster currently has twenty-eight Hadoop Worker Nodes. Each of these nodes has 256GB of RAM. After subtracting the required RAM needed to run the OS and additional key services, there is 192GB of RAM left for Hadoop containers. This means that the total cluster wide RAM capacity for Hadoop Containers 5.376TB. We have decided to allocate 50% of this RAM to SAS jobs or 2.688TB.

Total RAM Per Cluster Node	Available Container RAM Per Node	# Worker Nodes in Cluster	Total Container RAM Available	Amount of Cluster RAM Allocated to SAS Queue	Total Container RAM for SAS Queue
256GB	192GB	28	5.376TB	50%	**2.688TB**

Table 1: Total Cluster YARN Container RAM Available for SAS Users

For our exercise, we had previously determined that the average number of SAS batch or interactive sessions for each active, concurrent user at our site is two. From a Hadoop perspective, for each of these SAS jobs, we know that YARN will spawns at least two containers (an Application Master and a Job Task Container). And once SAS jobs are running in Hadoop, they can call additional Hadoop (ie. Pig, Hive, MapReduce) and SAS SAS HP (ie. proc hpsummary, hpmeans, hpfreq) services. The average number of additional Hadoop Containers spawned from each SAS job running on Hadoop is four. This means we can anticipate each active, concurrent SAS user to be allocating, on average, eight YARN Containers.

Avg # of Batch Jobs or Interactive Sessions per SAS User	# Containers Per Job or Session	Avg # of Additional Hadoop Containers Spawned from initial SAS Job Container	Avg Total # of Containers per SAS user
2	2	4	**8**

Table 2: Average Number of Containers per SAS Users

Now that we know, on average, each active SAS user will allocate eight containers, and we have 2.688TB of total cluster RAM available, we need to determine the total number of concurrent SAS users who can run at any one time on our Hadoop cluster.

One additional factor to consider is that not all SAS users need the same amount of computing resources. In our exercise, we have three SAS user resource categories (Low, Medium, and High). Each user falling into the Low category, by default, will allocate 2GB YARN Containers for their jobs. The SAS Medium category will default to a YARN Container Size of 4GB. And the High category will default to 8GB Container size. With these numbers in mind, we will be to have 134 concurrent SAS users on our system as indicated below.

SAS AppType (User Type)	Container Size	Anticipated % of Users Type on Server	Available Cluster Memory for SAS jobs	Max # of Containers	Avg # Containers Per SAS User	Total # of SAS Users
Low (General/Analyst)	2GB	70%	1.881TB	940	8	117
Medium	4GB	20%	537GB	134	8	16
High	8GB	10%	268GB	33	8	4
		Totals	**2.688TB**	**1107**		**134**

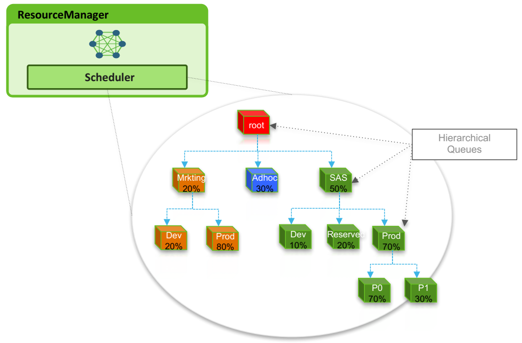
Table 3: Breakdown of SAS Application Types to be configured for SAS Users

YARN Capacity Scheduler

In this exercise, we are allocating 50% of the Hadoop Cluster's RAM to SAS Users. We will use the YARN Capacity Scheduler to setup this policy. We are able to setup Hierarchical YARN Queues, including sas94_queue, as a first cluster wide configuration step.

YARN Capacity Scheduler

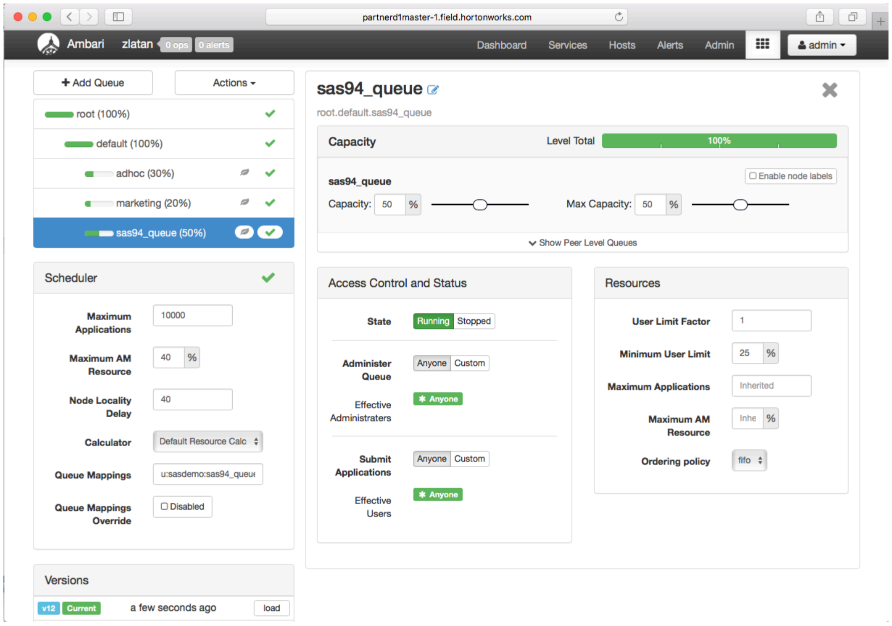
Example: 50% of Cluster RAM allocated to SAS Queue



Capacity Scheduler

Figure 5: YARN Capacity Scheduler Logical View - SAS Queue - 50% Hadoop Cluster RAM

Below is a view of the YARN Capacity Scheduler UI, we will see that the sas94_queue has 50% cluster RAM allocated. SAS Users Linux ids should also be mapped to this queue.



root (100%)
default (100%)
adhoc (30%)
marketing (20%)
sas94_queue (50%)

Scheduler

Maximum Applications: 10000
Maximum AM Resource: 40 %
Node Locality Delay: 40
Calculator: Default Resource Calc
Queue Mappings: u:sasdemo:sas94_queue
Queue Mappings Override: □ Disabled

Versions

v12 Current a few seconds ago load

sas94_queue

root.default.sas94_queue

Capacity — Level Total 100%

sas94_queue

□ Enable node labels

Capacity: 50 % Max Capacity: 50 %

Show Peer Level Queues

Access Control and Status

State: Running Stopped
Administer Queue: Anyone Custom
Effective Administraters: Anyone
Submit Applications: Anyone Custom
Effective Users: Anyone

Resources

User Limit Factor: 1
Minimum User Limit: 25 %
Maximum Applications: Inherited
Maximum AM Resource: Inhe %
Ordering policy: fifo

Figure 6: YARN Capacity Scheduler Admin View - SAS Queue - 50% Hadoop Cluster RAM

Our Hadoop cluster has **YARN Minimum Container Size** set cluster wide to 2GB. We are going to need to implement a mechanism to adjust Container size based on SAS User resource needs. We will use the **sasgrid-policy.xml** configuration file and SAS Metadata User and Group mappings to accomplish this task.

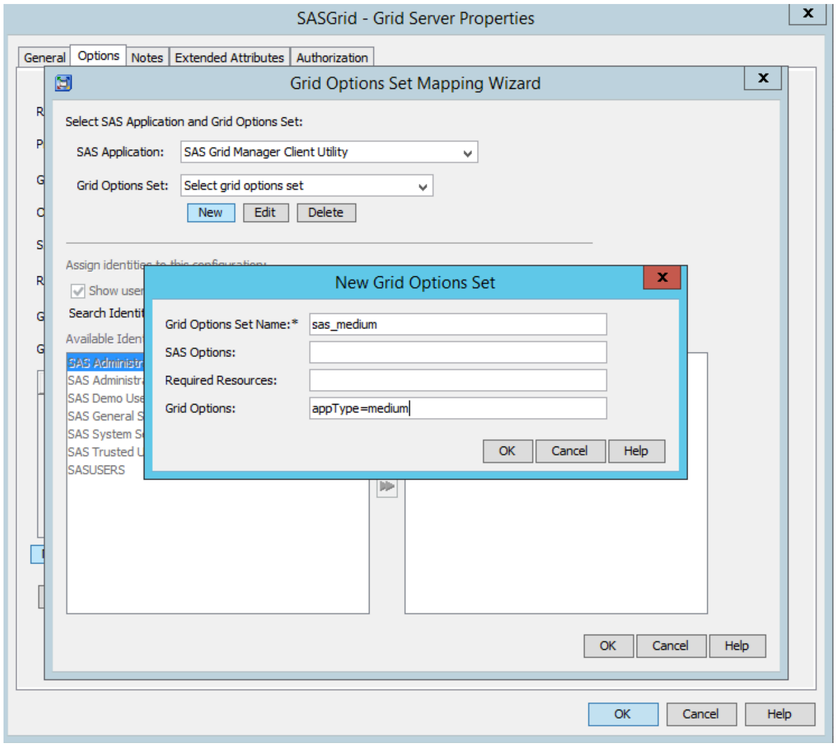
First, we will review a **sasgrid-policy.xml** file which will ensure that SAS users obtain the right amount of memory for their jobs. Within this SAS Grid configuration file, you will see three Grid Application Types: Low, Medium, and High.

SAS Grid Policy File Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<GridPolicy defaultAppType="low">

        <GridApplicationType name="normal">
                <jobname>Normal Job</jobname>
                <priority>20</priority>
                <nice>10</nice>
                 <memory>2048</memory>
                 <vcores>1</vcores>
                 <runlimit>480</runlimit>
                 <queue>default</queue>
                <hosts>
                        <hostGroup>development</hostGroup>
                </hosts>
        </GridApplicationType>

        <GridApplicationType name="low">
                <jobname>SAS Low</jobname>
                <priority>10</priority>
                <nice>0</nice>
                <memory>2048</memory>
                <vcores>1</vcores>
                <runlimit>480</runlimit>
                <queue>sas94_queue</queue>
                <hosts>
                        <hostGroup>sas94_work</hostGroup>
                </hosts>
        </GridApplicationType>

        <GridApplicationType name="medium">
                <jobname>SAS Medium</jobname>
                <priority>10</priority>
                <nice>0</nice>
                <memory>4096</memory>
                <vcores>1</vcores>
                <runlimit>480</runlimit>
                <queue>sas94_queue</queue>
                <hosts>
                        <hostGroup>sas94_work</hostGroup>
                </hosts>
        </GridApplicationType>

        <GridApplicationType name="high">
                <jobname>SAS High</jobname>
                <priority>10</priority>
                <nice>0</nice>
                <memory>8192</memory>
                <vcores>1</vcores>
                <runlimit>480</runlimit>
                <queue>sas94_queue</queue>
                <hosts>
                        <hostGroup>sas94_work</hostGroup>
                </hosts>
        </GridApplicationType>

        <HostGroup name="development">
                <host>partnerd1worker-1.field.hortonworks.com</host>
                <host>partnerd1worker-2.field.hortonworks.com</host>
                <host>partnerd1worker-3.field.hortonworks.com</host>
                <host>partnerd1worker-4.field.hortonworks.com</host>
                <host>partnerd1worker-5.field.hortonworks.com</host>
        </HostGroup>
        <HostGroup name="sas94_work">
                <host>partnerd1worker-4.field.hortonworks.com</host>
                <host>partnerd1worker-5.field.hortonworks.com</host>
        </HostGroup>

</GridPolicy>
```

The above sasgrid-policy.xml file is a sample. It illustrates how we can setup Grid Application Types for different SAS user groups. Note that we have specified a YARN Container memory and the SAS94 Queue for each AppType.

We that we have configured the appropriate Grid Application Types, we need a way to map SAS Applications (SAS Grid Manager Client Utility, SAS Enterprise Guide, others) to these SAS Grid Application Types. This can be done using the Grid Options Set Mapping Wizard within the SAS Management Console. This requires each SAS Application to be registered to the SAS Metadata Server and configured for **"IsGridEnabled"**. Once this is done, use the following **SAS Grid Server Properties Options Tab** for your Logical Grid Server (SASGrid in our case) and configure SAS users and groups to their appropriate default Grid AppType found in sasgrid-policy.xml above.



SAS Application: SAS Grid Manager Client Utility

Grid Options Set: Select grid options set

Grid Options Set Name:* sas_medium

SAS Options:

Required Resources:

Grid Options: appType=medium

Figure 7: Configuring SAS Metadata Groups to SAS Grid Application Types.

SAS Client Idle and Runtime Configurations

It is recommended that you set the following thresholds for SAS users as a default and adjust according to your sides workload needs. This will free up idle resources for other Hadoop users. It is recommended to be very conservative in these settings so that you do not inadvertently disrupt SAS user productivity.

- **SAS Client Idle Timeout** – in our example – we set to 120 minutes
  - You can control the amount of time a client such as SAS Enterprise Guide can stay connected but not active using the Inactive Client timeout setting.
- **SAS Run Limit Timeout** – in our example – we set to 480 minutes
  - Within the sasgrid-policy.xml file, you can control this using the "runlimit=xx" Grid Application Type parameter.

Also, a couple of other Hadoop parameters that we should pay close attention to are MapReduce Minimum Container Size, as well as the engine you are running for Hive. We recommend use Hive's Tez Engine in most cases as it provides in-memory query execution that is considerably faster than the MapReduce engine. We are currently investigating **YARN Node Labels** (Policy Scheduling) and the **Hive LLAP engine** (SQL Query Acceleration) for future configuration considerations.

SAS WORKLOAD MIGRATION CONSIDERATIONS

SAS Grid Manager for Hadoop should be leveraged as a compliment to your existing SAS infrastructure and not a replacement. While certain traditional storage architectures provide extremely fast IO for SAS workloads, there is a significant cost to these infrastructures. Selecting the right SAS workloads and data to move to Hadoop is a critically important step when moving SAS jobs to Hadoop. A key business driver is reducing overall storage costs. Another key driver accelerating the performance of SAS jobs that require access to large datasets in Hadoop. The most expensive operation for any big data job is moving large amounts of data from storage to a compute tier outside of Hadoop. With Hadoop, a primary goal design goal for the last decade has been to move a programs compute or math to where the data resides, in Hadoop. This enables avoiding costly data movement over the network.

With our business drivers in mind, SAS shops should initially identify a handful of SAS jobs running outside of Hadoop which have significantly large ephemeral(SASWORK) and permanent storage (SAS datasets, RDBMS, other inbound and outbound datasets) needs. If the SAS job data is currently stored on expensive SAN storage, cost savings will be significant. Any SAS jobs using Base SAS procedures like Proc Summary, Means and Freq which support SAS In-Database Pushdown should considered. Any SAS jobs planning to use Hadoop datasets should be on the list.

When moving existing SAS Workload data into Hadoop, there are several storage options. Existing SAS data can be moved into Hadoop Distributed File System (HDFS) and then, within Hive, a schema on ready can be established. If the existing SAS datasets and database tables are stored in Hive, as mentioned, SAS In Database capabilities can be leveraged. This will enable the SAS engine to have an opportunity to implicitly translate Base SAS Stat procedures to complex HiveQL, ultimately, moving most of the math to the data.

SAS Access to Hadoop is an add on SAS product that plays a critical role in this migration. SAS users will need to have some basic training on how to most effectively leverage the Hive and HDFS libname engines from their SAS jobs, as well as the SAS filename statement to HFDS. This basic syntax change in their programs will be required to take advantage directly of SAS data now residing in Hadoop.

- **Libname to Hive**
    - Uses a JDBC connect to Hive
    - Provides an opportunity for distributed parallel compute on Hive
        - Using SAS In Database Push Down capabilities

- **Libname to HDFS**
    - HDFS APIs are leveraged for direct, parallel read and write to Hadoop.

- **Filename to HDFS**
    - Supports inbound and outbound read and write of raw data to Hadoop.

SAS does include support for Hadoop Clients and RESTful APIs. The SAS Access to Hadoop can also allow a sas programmer to run explicit HDFS, Pig, Hive, and MapReduce code inline within their SAS job. In addition, SAS has an SQL Engine which supports both implicit and explicit Pass Thru HiveQL execution directly to the Hadoop cluster.

A DAY IN THE LIFE OF A SAS USER LEVERAGE SAS GRID

When using SAS clients, it should be transparent to the SAS users from a functionality perpective that they are interacting with a SAS Grid. Here is a detailed walk through of a SAS user leveraging SAS Enterprise Guide running SAS Process Flows and Tasks inside of Hadoop.

Below, a SAS user logs into SAS Enterprise Guide's UI, and opens an existing project. To execute work from this project, the user must launch a SAS Workspace Server. In this case, if the user expands the +SAS Grid on the lefthand side of the screen, SAS Grid Manager for Hadoop will request YARN's Resource Manager to launch a SAS Workspace Server (WSS) inside the Hadoop Cluster.

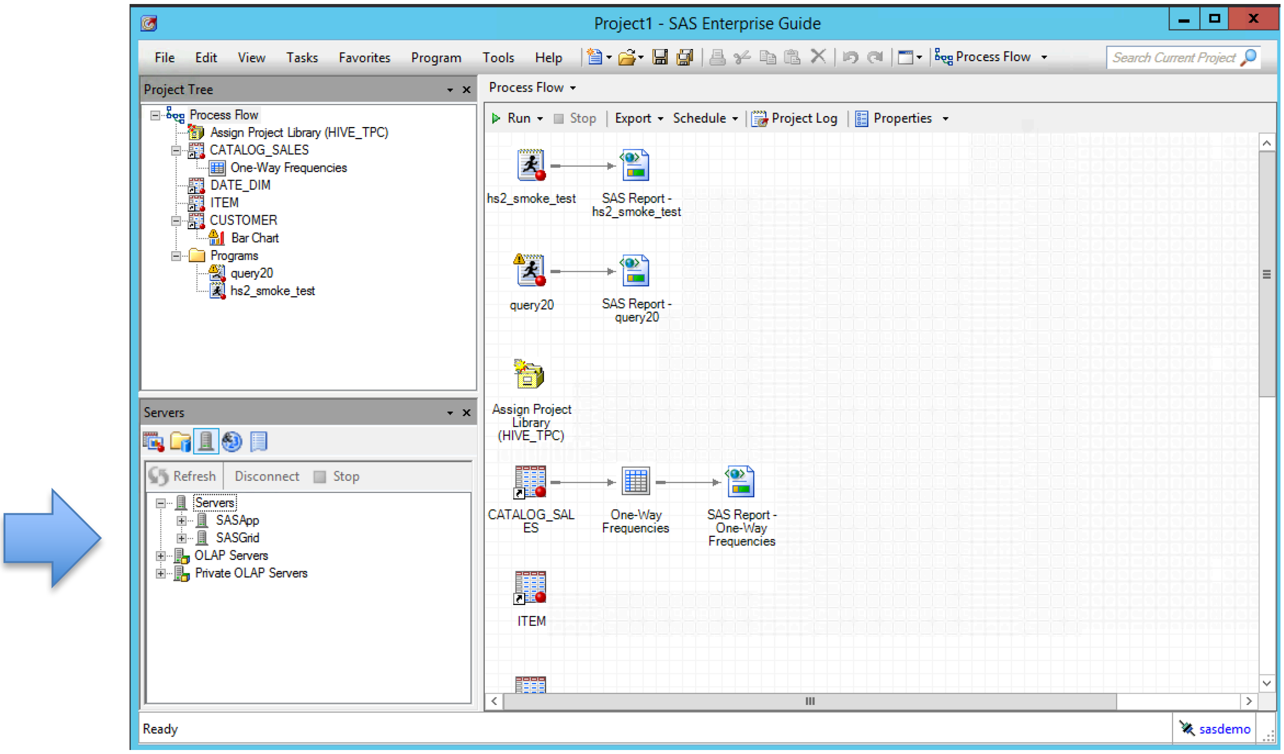


Figure 8: Click on +**SASGrid** to launch SAS Workspace Server on Hadoop Worker Node

The green check mark next to **SASGrid** below indicates that the SAS EG user can run any Process Flow or Task in Hadoop, because YARN has successfully launched the SAS WSS on a Hadoop Worker Node.

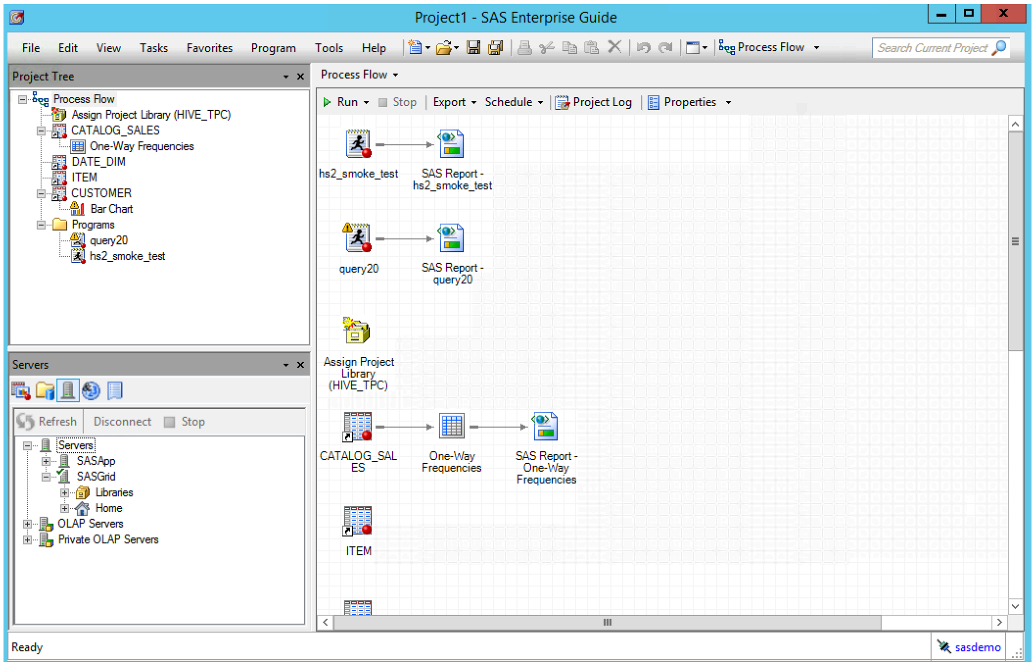
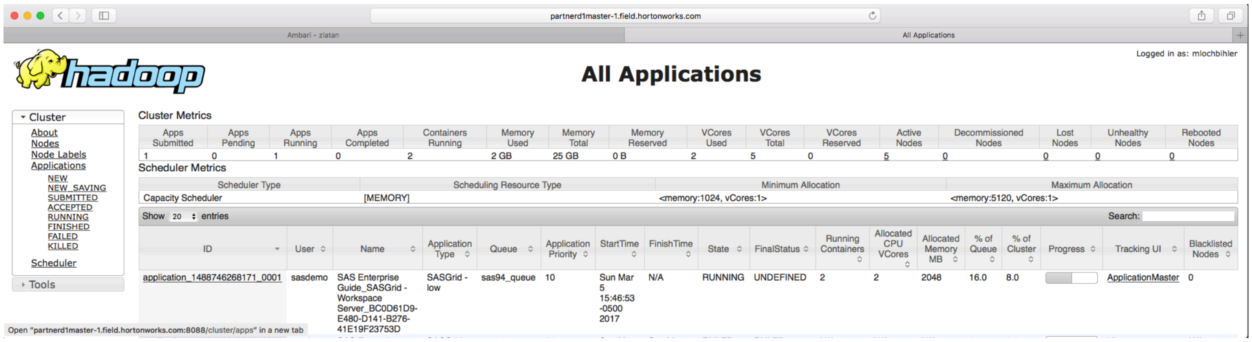


Figure 9: The green arrow (SASGrid) indicates SAS Workspace Server is established in Hadoop

If we look at the YARN UI below, typically reserved for Hadoop Administrators, we can see a SAS Enterprise Guide – Workspace Server in a RUNNING state. This indicates to the Hadoop Administrator that a SAS job is running and has reserved two YARN containers and is currently active on the Hadoop cluster:



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	1	0	2	2 GB	25 GB	0 B	2	5	0	5	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:5120, vCores:1>

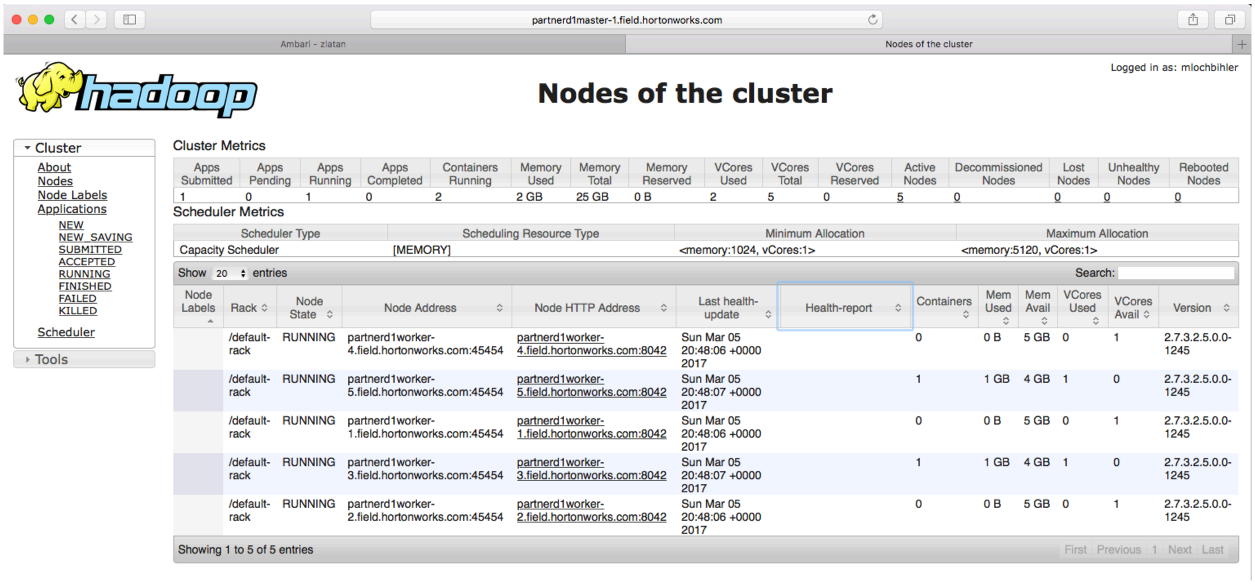
Show 20 entries Search:

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
application_1488746268171_0001	sasdemo	SAS Enterprise Guide_SASGrid - Workspace Server_BC0D61D9-E480-D141-B276-41E19F23753D	SASGrid - low	sas94_queue	10	Sun Mar 5 15:46:53 -0500 2017	N/A	RUNNING	UNDEFINED	2	2	2048	16.0	8.0		ApplicationMaster	0

Open "partnerd1master-1.field.hortonworks.com:8088/cluster/apps" in a new tab

Figure 10: YARN UI - SAS EG Workspace Server (WSS) running in sas94_queue on Hadoop.

Switching to the "Nodes" link in the YARN UI (this is a smaller sample cluster to illustrate our example only), you can see the two YARN containers associated with the SAS WSS, running in 1GB YARN Containers.



Nodes of the cluster

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	1	0	2	2 GB	25 GB	0 B	2	5	0	5	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:5120, vCores:1>

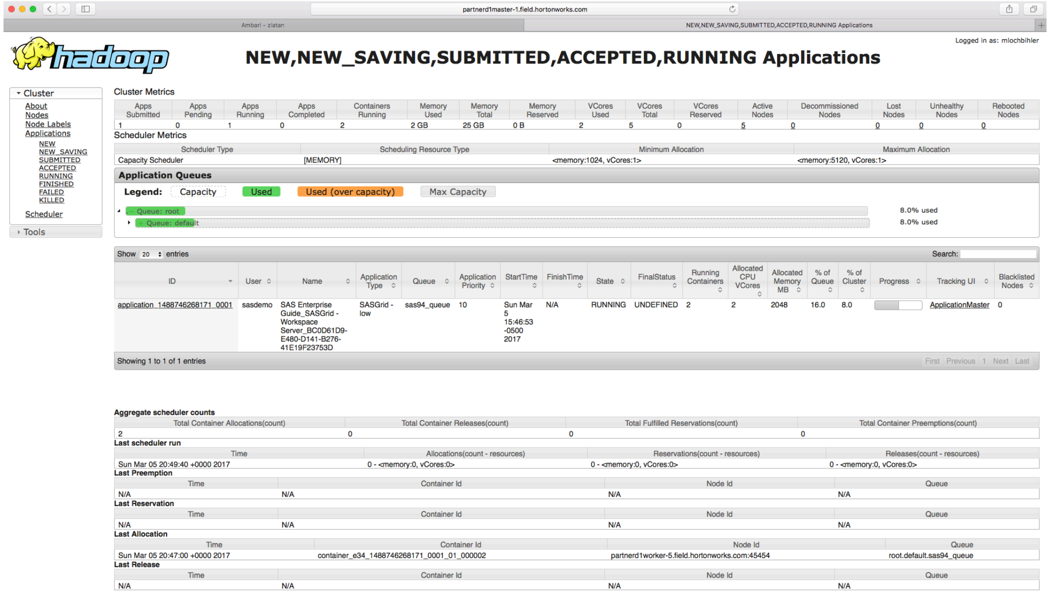
Show 20 entries Search:

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
	/default-rack	RUNNING	partnerd1worker-4.field.hortonworks.com:45454	partnerd1worker-4.field.hortonworks.com:8042	Sun Mar 05 20:48:06 +0000 2017		0	0 B	5 GB	0	1	2.7.3.2.5.0.0-1245
	/default-rack	RUNNING	partnerd1worker-5.field.hortonworks.com:45454	partnerd1worker-5.field.hortonworks.com:8042	Sun Mar 05 20:48:07 +0000 2017		1	1 GB	4 GB	1	0	2.7.3.2.5.0.0-1245
	/default-rack	RUNNING	partnerd1worker-1.field.hortonworks.com:45454	partnerd1worker-1.field.hortonworks.com:8042	Sun Mar 05 20:48:06 +0000 2017		0	0 B	5 GB	0	1	2.7.3.2.5.0.0-1245
	/default-rack	RUNNING	partnerd1worker-3.field.hortonworks.com:45454	partnerd1worker-3.field.hortonworks.com:8042	Sun Mar 05 20:48:07 +0000 2017		1	1 GB	4 GB	1	0	2.7.3.2.5.0.0-1245
	/default-rack	RUNNING	partnerd1worker-2.field.hortonworks.com:45454	partnerd1worker-2.field.hortonworks.com:8042	Sun Mar 05 20:48:06 +0000 2017		0	0 B	5 GB	0	1	2.7.3.2.5.0.0-1245

Showing 1 to 5 of 5 entries

Figure 11: YARN UI – "Nodes" link showing two 1GB containers supporting the SAS WSS.

And, switching to the YARN UI – “Scheduler” link, we can see Cluster Metrics below.



NEW,NEW_SAVING,SUBMITTED,ACCEPTED,RUNNING Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	1	0	2	2 GB	25 GB	0 B	2	5	0	5	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:5120, vCores:1>

Application Queues

Queue: root 8.0% used
Queue: default 8.0% used

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
application_1488746268171_0001	sasdemo	SAS Enterprise Guide_SASGrid - Workspace Server_BC0D61D9-E480-D141-B276-41E19F23753D	SASGrid - low	sas94_queue	10	Sun Mar 5 15:46:53 -0500 2017	N/A	RUNNING	UNDEFINED	2	2	2048	16.0	8.0		ApplicationMaster	0

Showing 1 to 1 of 1 entries

Aggregate scheduler counts

Total Container Allocations(count)	Total Container Releases(count)	Total Fulfilled Reservations(count)	Total Container Preemptions(count)
2	0	0	0

Last scheduler run

Time	Allocations(count - resources)	Reservations(count - resources)	Releases(count - resources)
Sun Mar 05 20:49:40 +0000 2017	0 - <memory:0, vCores:0>	0 - <memory:0, vCores:0>	0 - <memory:0, vCores:0>

Last Preemption

Time	Container Id	Node Id	Queue
N/A	N/A	N/A	N/A

Last Reservation

Time	Container Id	Node Id	Queue
N/A	N/A	N/A	N/A

Last Allocation

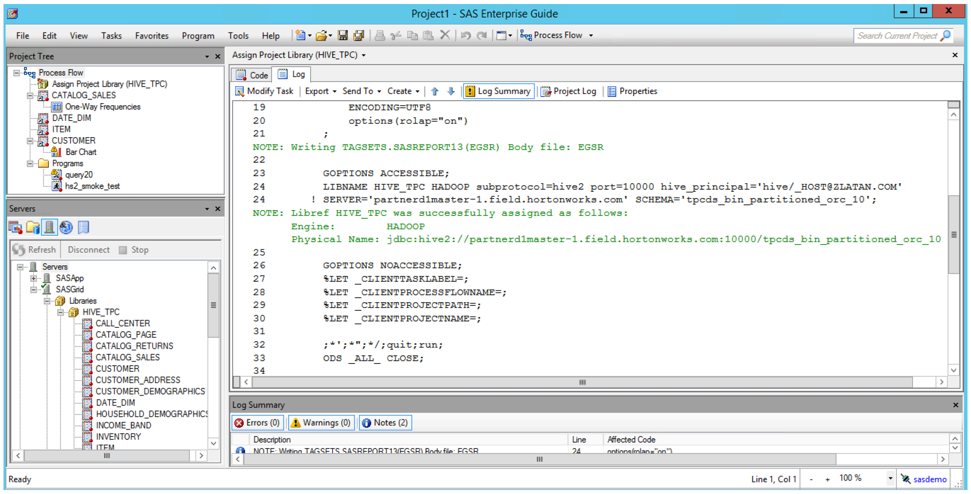
Time	Container Id	Node Id	Queue
Sun Mar 05 20:47:00 +0000 2017	container_e34_1488746268171_0001_01_000002	partnerd1worker-5.field.hortonworks.com:45454	root.default.sas94_queue

Last Release

Time	Container Id	Node Id	Queue
N/A	N/A	N/A	N/A

Figure 12: YARN UI – “Scheduler” link Hadoop Cluster Metrics.

Switching back to the SAS User EG UI, the SAS user can see that a SAS Library has been assigned pointing to Hive. We can also see in the SAS EG Explorer window on the bottom left hand side of the screen, all the Hive Tables available to the SAS user which are associated with SAS libref HIVE_TPC. At this point, the SAS EG user can run SAS analytic tasks directly against Hive tables using the Hive database schema tpcds_bin_partitioned_orc_10.

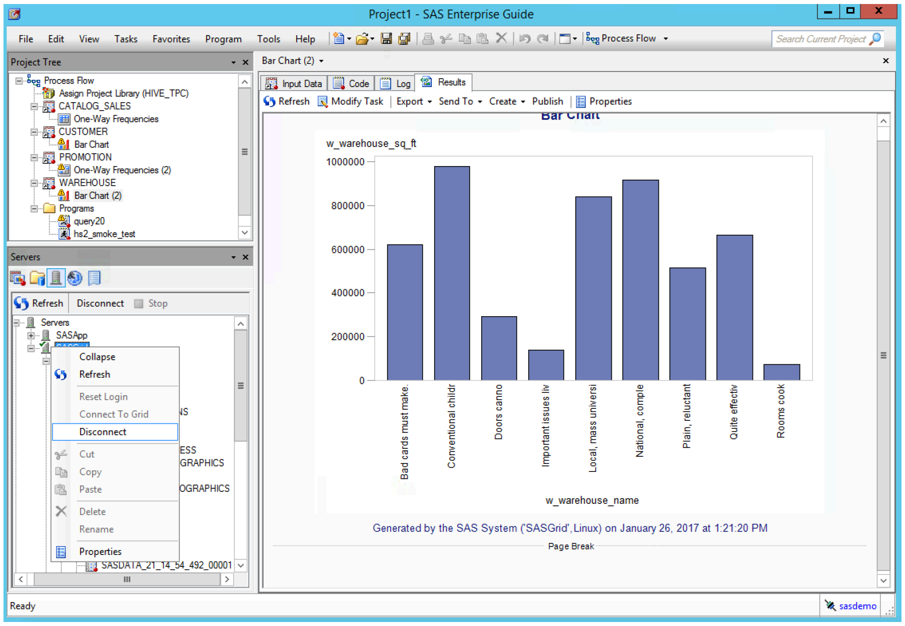


```
19                ENCODING=UTF8
20                options(rolap="on")
21           ;
NOTE: Writing TAGSETS.SASREPORT13(EGSR) Body file: EGSR
22
23           GOPTIONS ACCESSIBLE;
24           LIBNAME HIVE_TPC HADOOP subprotocol=hive2 port=10000 hive_principal='hive/_HOST@ZLATAN.COM'
24         ! SERVER='partnerd1master-1.field.hortonworks.com' SCHEMA='tpcds_bin_partitioned_orc_10';
NOTE: Libref HIVE_TPC was successfully assigned as follows:
      Engine:        HADOOP
      Physical Name: jdbc:hive2://partnerd1master-1.field.hortonworks.com:10000/tpcds_bin_partitioned_orc_10
25
26           GOPTIONS NOACCESSIBLE;
27           %LET _CLIENTTASKLABEL=;
28           %LET _CLIENTPROCESSFLOWNAME=;
29           %LET _CLIENTPROJECTPATH=;
30           %LET _CLIENTPROJECTNAME=;
31
32           ;*';*";*/;quit;run;
33           ODS _ALL_ CLOSE;
34
```

Description	Line	Affected Code
NOTE: Writing TAGSETS.SASREPORT13(EGSR) Body file: EGSR	24	options(rolap="on")

Figure 13: SAS Enterprise Guide – Library HIVE_TPC has been assigned to Hive.

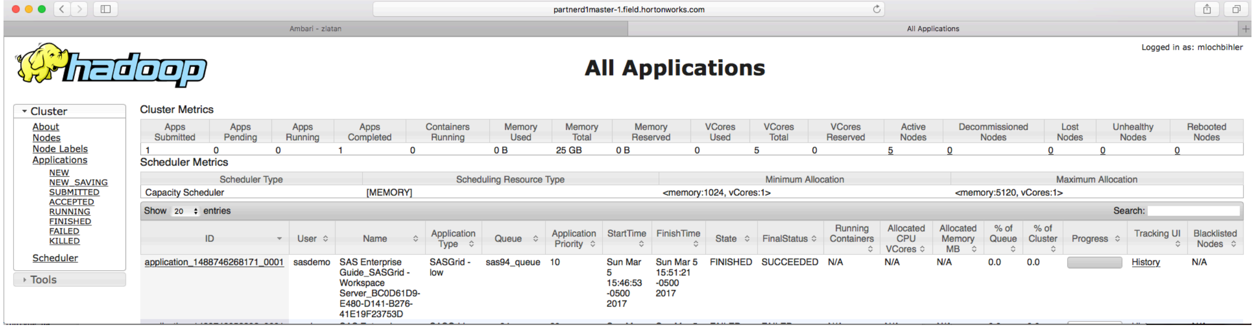
At this point, the SAS EG user can run any traditional SAS job or code. The SAS user can also call any HDP service (i.e. HDFS, Pig, Hive, MapReduce) in-line within this SAS workspace server. If installed, the SAS user can also call SAS High Performance Analytics, which can run on YARN in the cluster as well. When the SAS EG user issues a "Disconnect" (see below), this will initiate a request to the SAS Workspace Server to shutdown. Yarn will release the container that was used for running the SAS Workspace Server and reclaim the resources associated with that container.



Generated by the SAS System ('SASGrid', Linux) on January 26, 2017 at 1:21:20 PM

Figure 14: SAS Enterprise Guide – "Disconnect" from Hadoop.

Once the "Disconnect" is complete in EG, the Hadoop Administrator will see the SAS Enterprise Guide Workspace Server is now "FINISHED". (See below). A disconnect will also occur automatically if a SAS EG user shuts down the UI.



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
1	0	0	1	0	0 B	25 GB	0 B	0	5	0	5	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:5120, vCores:1>

ID	User	Name	Application Type	Queue	Application Priority	StartTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	% of Queue	% of Cluster	Progress	Tracking UI	Blacklisted Nodes
application_1488746268171_0001	sasdemo	SAS Enterprise Guide_SASGrid - Workspace Server_BC0D61D9-E480-D141-B276-41E19F23753D	SASGrid - low	sas94_queue	10	Sun Mar 5 15:46:53 -0500 2017	Sun Mar 5 15:51:21 -0500 2017	FINISHED	SUCCEEDED	N/A	N/A	N/A	0.0	0.0		History	N/A

Figure 14: YARN UI - SAS EG Workspace Server (WSS) FINISHED status on Hadoop.

For sites interested in moving traditional SAS workloads co-located onto Hadoop, SAS Grid Manager for Hadoop is an ideal solution to meet this need. In this paper, we have discussed the benefits of moving to SAS Grid Manager on Hadoop, required architecture and recommended configurations. We have also shared a seamless SAS user experience with you. If you would like to learn more about SAS Grid Manager for Hadoop, please see the following links below.

REFERENCES

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

SAS Institute Inc. "Introduction to Grid Computing."
Available at http://support.sas.com/rnd/scalability/grid/.

"Hortonworks Apache Hadoop YARN Overview Web Page."
Available at https://hortonworks.com/apache/yarn/.

"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.

"Introducing - SAS® Grid Manager for Hadoop." SAS Global Forum Paper SAS6281-2016
Available at http://support.sas.com/resources/papers/proceedings16/SAS6281-2016.pdf

CONTACT INFORMATION

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

Mark Lochbihler
Principal Architect, Hortonworks Inc.
mark.lochbihler@hortonworks.com