PharmaSUG 2015 - Paper IB04

SAS Grid: Simplified

Rajinder Kumar, inVentiv International Pharma Services Pvt. Ltd., Pune, Maharashtra (India)

ABSTRACT

For most organizations, huge volume of data means big trouble, because it has great impact on speed and accuracy of data analysis. Due to high demand of reliable environment and faster response time for huge data and large number of users from same organization, SAS Grid is in great demand. This paper puts light on some of common features of SAS Grid. It also provides some of the challenges in SAS Grid and alternate solution for them. It shares different examples related to SAS Grid's challenges and their solutions. For example, few options or functionalities which work fine in normal SAS, may need slight modification before getting used in SAS Grid.

This paper provides details about all these problems, root cause of the problem and their possible solutions (if any). I am sure, as use of SAS Grid is becoming more popular, these will help all the end users in finding answers to their most common questions.

INTRODUCTION

It is becoming a big challenge to analyze volumes of data in a timely manner. Due to huge volume of data, in general SAS takes more time in running and providing analysis results, which if needed to run again with any modifications, means more time will be required. Sometimes due to this requirement of high running time, interests of programmers/analysts also get impacted and it has direct impact on the efficiency of users. When high number of users with high volume of data competes for resources, it has severe impact on the speed of SAS. As a result maintenance of system will be required more frequently, which again means system outage or unplanned downtimes periodically. Any downtime planned or unplanned comes with a cost involved. SAS Grid is the answer to all these. It avoids costly infrastructure upgrade cycles and manages multiple jobs, applications and users in a better way. Here we will touch on some of the features related to SAS Grid, which are main reason behind its high demand and great performance. SAS Grid also needs to be thought of differently than SAS and some things that can easily be accomplished in a traditional implementation of SAS, has to be programmed differently in a Grid environment. There are also some untraditional ways to get some programs to run faster (ie pieces of the program being submitted to processor simultaneously).

SAS GRID

SAS Grid is a combination of Grid Client, SAS Metadata Server, Central File Server, SAS Control Server and multiple Grid Nodes. Figure 1 given below provides details about all these and their relationships. Before proceeding further, it really becomes important to have a basic idea about all these components of SAS Grid.

SAS Grid is a cluster of computers, where SAS Grid manager controls distribution of computing task among these computers. Workloads are distributed across a Grid cluster of computers and as a result it provides following functionalities:

- Workload balancing: Workload is distributed in a balanced manner to all the available resources of SAS Grid, making high availability of resources to multiple users in a SAS environment.
- Accelerated processing: SAS Grid divides entire task in subtasks and then distributes them to shared pool of resources. Due to running these subtasks in parallel on different part of Grid, entire tasks completes faster.
- Scheduling jobs: Jobs can be scheduled in SAS Grid, which will run on trigger (on specified time or after occurring specified event).

Due to separately handling SAS applications and infrastructure used to execute the applications, SAS Grid transparently helps in adding or removing hardware resources as per need. It also provides tolerance of hardware failures within the Grid infrastructure.

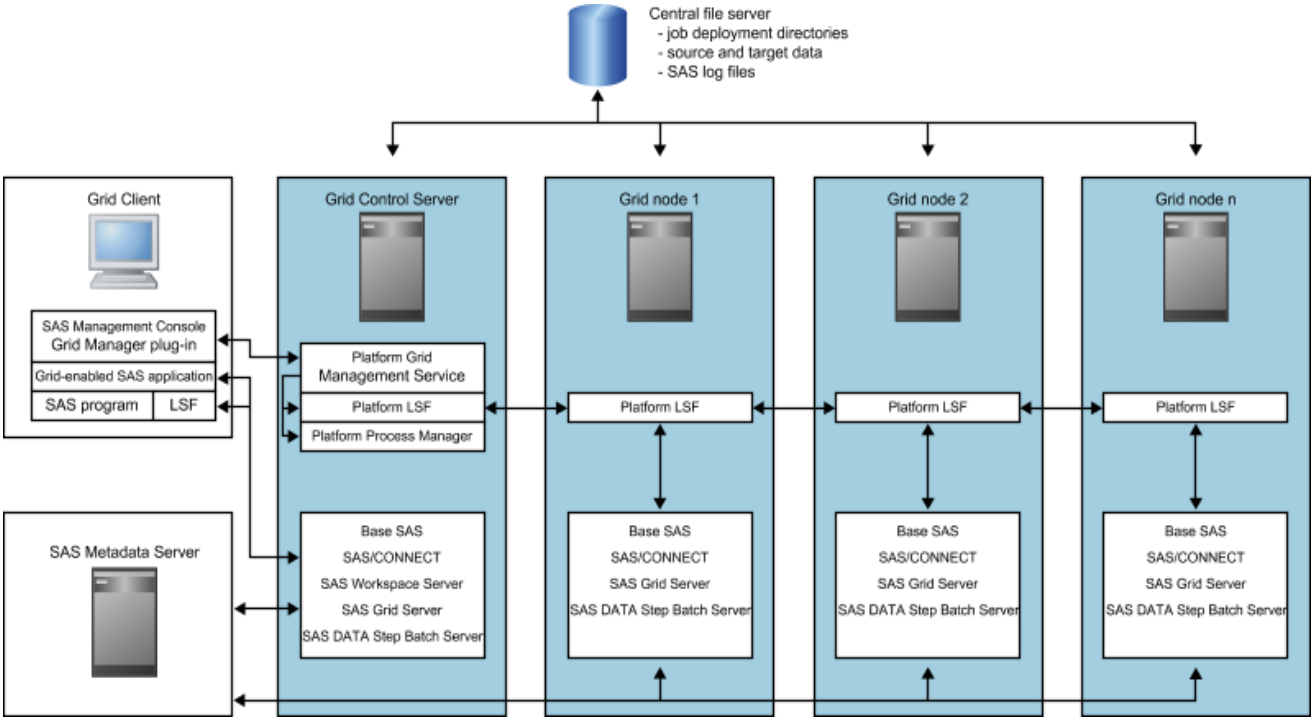

Figure 1: SAS Grid topology

GRID CONTROL SERVER

This machine is one of the nodes from Grid, which controls distribution of jobs to the Grid. It can be any machine in Grid, which can be designated as the Grid control server. This allows for flexibility, should this server fail, control can be passed to another server and allows the environment to continue to function. It is up to one, whether to configure the Grid control server as a Grid resource capable of receiving work or not. It is recommended not to use this node for receiving work or running any job, which ensures high availability in managing the Grid environment. Grid control server machine must contain Base SAS, SAS/CONNECT, and Platform Load sharing Facility (LSF) along with Process Manager (PM) and Platform Grid Management Service (GMS). SAS workspace server can be configured for SAS applications(SAS Data Integration Studio, SAS Enterprise Miner, SAS Enterprise Guide, and SAS Add-In for Microsoft Office) to run programs that take advantage of the Grid.

GRID NODES

Grid node is a machine which is capable of receiving and executing work that is distributed to a Grid. All but Grid control server, machines in Grid are called Grid nodes. Each Grid node must contain Base SAS, SAS/CONNECT, Platform LSF, and any applications and solutions needed to run Grid-enabled jobs for your business needs. As per your business needs and depending on size, complexity and volume of the jobs to be run on Grid, number of nodes in Grid are decided. As per requirement new node can be added or existing nodes can be removed.

CENTRAL FILE SERVER

It is a critical component of a Grid environment which is used to store data for jobs that run on the Grid. For each application on a Grid node it becomes really critical to efficiently access the data, in absence of which its performance causes slowdown and reduces entire benefit of using a Grid. Type of file storage system is determined with the help of amount of storage required and type of input/output (I/O) transactions.

SAS METADATA SERVER

Metadata definitions needed by SAS Grid Manager and other SAS applications and solutions that are running on the Grid are stored in metadata repository. SAS metadata server contains the metadata repository. It can be on Grid control server, but having it on a dedicated machine provides better results. Multiple users can access this server simultaneously to read metadata from or write metadata to one or more SAS Metadata Repositories.

SAS MANAGEMENT CONSOLE

It manages the definitions in the metadata repository, to submit jobs to the Grid through the Schedule Manager plug-in, and to monitor and manage the Grid through the Grid Manager plug-in. There can be real power here to have flexibility and knowledge on what is going on in the environment and have the functionality to have jobs run when certain events happen, either that a specific time occurs or if a file is updated or appears on the system.

GRID CLIENTS

This is not a part of Grid resources which are available to executing work, but it submits jobs to the Grid for processing.

LOAD SHARING FACILITY (LSF)

This is really the functionality that makes Grid work. LSF provides the flexibility to close any node at any time so that no jobs are submitted to that node. This can be utilized to do maintenance on any of the Grid nodes and test separately while the production remains running. The node can be added back to the Grid after the testing is complete. This can be utilized not only for SAS software fixes/license upgrades but also for OS upgrades.

GRID MAIN FEATURES

With all above details now we are curious to know, why SAS Grid is more effective. Below are main functions of SAS Grid which are responsible for its high speed and efficiency.

MULTI-USER WORKLOAD BALANCING

As it is quite common, most of the organizations have many SAS users who are working on SAS and accessing and performing SAS related tasks on same resources. SAS Grid distributes the workload submitted by multiple users to available machine in Grid. Here SAS Grid ensures no machine is overloaded, which as a result ensure faster outputs. Sometimes due to multiple requests, some tasks may need to be forced to wait. SAS Grid also manages queues of waiting tasks as per their priority and requirement of resources. Priority can be of many types, such as normal queue, priority queue and night queue etc. Where a normal queue is in general runs its programs when it finds required resources available after waiting in queue, in case needed. While priority queue is, a queue where programs are given higher priority than normal queue, when there are enough jobs to cause the system to hold off processing some jobs. Tasks from normal queues are put on waiting and priority queue tasks are allowed to use the computing resources on priority. Whereas night queue is queue, whose tasks doesn't need any priority and should be run in night time, when there are fewer individuals submitting jobs and waiting for the results. A separate queue for small tasks which takes very less time in running is also managed and it ensures smaller tasks which can execute quickly, runs without waiting for long, it also helps in keeping waiting queue small.

PARALLELIZED WORKLOAD BALANCING

Traditionally SAS programs get executed sequentially. Hence if a large program is submitted then it takes longer time in executing. Because it executes step by step and next step starts only after finishing execution of the previous step. Due to this dependency and sequential order it takes longer time to process. While in SAS Grid the same program can be divided into small pieces and then run in on different nodes. This ensures parallel running of sub programs of a large program. This can mean that the results can be achieved in lesser time. Due to this parallelized workload balancing larger programs which usually takes more time in processing can be executed in lesser time freeing up the resources to process other jobs sooner.

HIGH AVAILABILITY

Due to multiple nodes present, performing same task in SAS Grid, it helps in ensuring if a node fails, the system recovers to provide another available node for performing same task quickly with very minimal downtime. This way SAS Grid provides high availability. As there are many nodes, if a particular node requires maintenance then also it doesn't mean there will be outage of entire system. Although there can be outage for this, but it will be minimal for performing de-attaching and attaching back to that node, after performing maintenance task on that particular node.

SCALABILITY

Usually in most of the organizations as work grows, number of users and load on SAS resources also grows. Sometimes its reverse can also happen. Then need arises to grow or shrink SAS resources. SAS Grid provides this feature of scalability for growing or shrinking with minimal effort and downtime. Usually when we want to grow in hardware vertically then existing hardware becomes of no use and totally new hardware needs to be purchased. But

SAS Grid makes scaling easy with overcoming from hardware vertical scalability limits. This also allows for a smaller impact on IT hardware budgets as you can add fewer machines to have a bigger impact on helping the system grow.

SCHEDULING OF JOBS

SAS management console of SAS Grid provides the ability to schedule the jobs on SAS Grid. The jobs can be scheduled to run when specified trigger occurs (time or file events). After the trigger jobs are put into queue for running, based on the priority set and availability of resources from SAS Grid manager these jobs are executed. This helps in managing load on resources (time based trigger) and better execution of plans (event based trigger).

Best thing about SAS programs on SAS Grid is that they support all the functionality of scheduling, workload balancing and parallel workloads.

GENERAL CHALLENGES IN SAS PROGRAMS IN USING SAS GRID

In a regular SAS (i.e., non-Grid) environment, user submits a program either by pressing a key from keyboard (usually F3) or by pressing RUNNING MAN icon on the task bar. Due to being on single server or single node for executing the submitted program, both options (pressing key and RUNNING man icon) provides same results. But in SAS Grid it has different meaning. As it being on local and remote server, pressing key from keyboard(F3) submits it locally, same is done with the help of pressing SINGLE RUNNING MAN icon at task bar, while submitting it with the help of DOUBLE RUNNING MAN submits it on remote server. Pressing key(F3) from keyboard along with RSUBMIT and ENDRSUBMIT statements also submits it on remote server. Due to local and remote server submission there are some challenges faced, in this section we will have a look at some of the common challenges due to migration to SAS Grid.

When programs from normal SAS get moved to SAS Grid environment, some of the programs which were running on SAS absolutely fine earlier, now has some error or warning or note in log. These challenges are not issues in SAS Grid, but they may be challenges due to shift from one environment to other (SAS Grid). It may be due to change in bit size of system (64bit from 32 bit of application) or may be some system options settings or may be possibly due to un-purchased any specific license. Let's discuss some of these challenges and their possible solutions.

DDE FUNCTIONALITY ISSUE

DDE (Dynamic Data Exchange) functionality of SAS doesn't work in SAS Grid. Below SAS code works perfectly in normal SAS:

Using DDE to read data from Microsoft Excel:

```
/* The DDE link is established using Microsoft Excel SHEET1, rows 1 through 10 and
columns 1 through 3  */

filename monthly dde 'excel|sheet1!r1c1:r10c3';

DATA monthly;
  infile monthly;
  input var1 var2 var3;
run;


PROC PRINT;
run;
```

But when same is used in SAS Grid, it doesn't work in SAS Grid. PROC IMPORT needs to be used for this.

One of the possible reason for this is, Grid nodes do not have MS products licensed on them which is what DDE needs to run. MS products are licensed only on the control server. So when the job is run on the nodes it does not have the ability to launch MS products which is what DDE needs.

DBMS = EXCEL OPTION IN PROC IMPORT

In PROC IMPORT, DBMS = EXCEL option doesn't work, although it works fine, in normal SAS. This is a challenge related to configuration, with 64-bit SAS, and 32-bit MS applications. In SAS Grid, this needs to be replaced with DBMS = XLS option.

CONCEPT OF SINGLE AND DOUBLE RUNNING MAN

In normal SAS, for submitting SAS program only SINGLE RUNNNING MAN is available. In SAS Grid, for submitting programs locally SINGLE RUNNING MAN can be used while for submitting programs on remote server DOUBLE RUNNING MAN can be used.

For remotely submitting task, RSUBMIT and ENDRSUBMIT statements can also be used as an alternate. Where RSUBMIT should be first statement of program and ENDRSUBMIT should be last as given below.

RSUBMIT;

******Normal Code which needs to be submitted remotely***;

ENDRSUBMIT;

Programs submitted with SINGLE RUNNING MAN goes and get executed at control server while programs submitted with DOUBLE RUNNING MAN are processed on Grid nodes. A midway approach, where some part of the code is submitted on control server and rest on Grid nodes can also be used for better and quicker results.

For example below piece of code submits DDE part of the code to control server and then some part of the code to Grid nodes with the help of RSUBMIT and ENDRSUBMIT statements and then pass the processing back to the control server.

```
/* The DDE link is established using Microsoft Excel SHEET1, rows 1 through 10 and
columns 1 through 3  */

filename monthly dde 'excel|sheet1!r1c1:r10c3';

DATA monthly;
  infile monthly;
  input var1 var2 var3;
run;


PROC PRINT;
run;
```

RSUBMIT;

code for Grid nodes**;

ENDRSUBMIT;

normal code for submitting at Control server;

Above code can be divided into three parts, first one containing DDE functionalities is executed at control server and then second part is submitted at Grid nodes with the help of RSUBMIT statements. After its execution ENDRSUBMIT passes the processing back to the control server. This way existing code can also be used without any/much modifications.

CONCLUSION

SAS Grid helps significantly increase productivity and improve cost. It allows to exploit the previously under-utilized power of computing resources while providing a more stable and scalable environment. It is also cost effective. While the implementation of the Grid presents some significant challenges, given that Grid computing is in its early stages of development, some of this is to be expected.

SAS Grid is really very powerful and provides high speed and high availability. As this paper puts light, on some of main components of SAS Grid and its main features, after understanding its features and getting introduced to its

main components, it becomes easier to understand reason of its high demand. Discussion about some of basic challenges and their resolutions helps in crossing initial hurdle in use of SAS Grid. Concept of SINGLE RUNNING MAN and DOUBLE RUNNING MAN helps in understanding difference between submitting a program locally and remotely. Overall this paper helps new user of SAS Grid in getting familiar to one of the high demanding SAS product.

REFERENCES

SAS Documentation. Copyright© 2014. Grid Computing in SAS® 9.4, Third Edition page 3-7. Cary, NC, USA: SAS Institute Inc.,
SAS Documentation, “SAS Grid Computing.” Available at http://support.sas.com/documentation/

ACKNOWLEDGMENTS

I would like to thank my manager Neelam Shinde for her support. I would also like to thank Sandeep Sawant, Nancy Brucken and all my team members for their support and valuable inputs.

CONTACT INFORMATION

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

Name: Rajinder Kumar
Enterprise: inVentiv International Pharma Services Pvt Ltd.
Address: Building no. 4, VI floor, Commerzone, Jail road, Yerwada
City, State ZIP: Pune, Maharashtra, India - 411006
Work Phone: +91 20 3056 9217
E-mail: Rajinder.kumar@inventivhealth.com