Paper 377-2011

Implementing SAS® GRID computing in an Analytical Environment

Shivashanker Bitla, Discover Financial Services, Riverwoods, IL
Rich Kerwin, Discover Financial Services, Riverwoods, IL

ABSTRACT

This paper focuses on the value of implementing SAS Grid in a large analytical environment and uncovers the lessons learned in the course of the Grid implementation and subsequent rollout. The former environment consisted of 4 SAS v9.1.3 EBI installations on four server partitions, which served as Adhoc, Development, Production and Metadata environments, and were housed on one physically large server. The new SAS Grid is an internally hosted SAS 9.2 M 0 EBI installations housed on a SAS Grid with 8 identical server nodes using a shared file system.

Primary objectives for converting the SAS environment to a Grid with shared file system were to provide a cost effective, readily scalable in-house analytical environment for a rapidly expanding SAS User community that would replace an expensive, externally housed, difficult to upgrade environment that lacked the ability to distribute data across a shared file system. A pilot of SAS Grid on Linux was completed at an offsite vendor location before onsite implementation. Putting the SAS Grid on a shared file system promised to alleviate duplicate data, be more cost effective to upgrade and allowed us to more effectively utilize resources (cpu/memory) over the partition based environment. Secondary considerations included providing business users with all of the same functionality and access to SAS tools as was available in the old environment, including the ability to submit jobs on the server in both batch and Interactive mode, and to run jobs against databases.

INTRODUCTION

Our SAS EBI environment has users from across the company – Human Resources, Marketing, Risk, Card member Services, Finance, and Consumer Banking. The bulk of the processing on our SAS environment is from ad hoc related jobs – model development, reporting, campaigns, etc. Some of the processing is from running monthly production models and some processing takes place as Users perform new model development and the testing of new models before they are fully implemented in the production area. Most of our data is sourced from Relational Databases housed on separate servers and accessed by SAS through SAS/DBMS access engines. Additionally, along with SAS, other data mining and modeling products are also employed on the environment.

CHALLENGES IN OLD ENVIRONMENT

Our former environment was a SAS913 EBI Installation that was housed at an offsite vendor location. It consisted of a single P595 server that was partitioned into four LPARs as below.

- Ad hoc - assigned for business users
- Metadata
- Production – assigned for Model Run
- Development – assigned for model development (also acts a staging area)

The largest LPAR was the Ad hoc area, which had a majority of the CPUs and memory. This setup, while serviceable, presented us with a number of challenges and frequent processing bottlenecks, particularly in the Ad hoc environment where we had many concurrent SAS users who were sharing a single, large multi-CPU server. As our user base was continuing to grow and the demands on the server were continuing to increase, it was readily apparent that the lack of adequate processing power and frequent processing bottlenecks truly needed to be addressed. Our large multi-user environment had grown beyond what a single server could be expected to handle.

- The first challenge we faced in the old environment was with the availability of the servers: Because the four partitions sat on a single frame, any microcode upgrades on the hardware or OS level maintenance required downtime across the entire frame.

- The second challenge was with resource mobility across the partitions. The Production server was busy only during month end processing (usually the first 15 days of each month) and was available and very underutilized after month end processing had completed. The ad hoc server was almost always busy and was under near constant resource constraints. This was partially alleviated by allowing some of the CPUs on the Production and Development areas to be added to a dynamic pool which could be used by the Ad hoc area based on need and availability. This helped with the processing bottlenecks on the Ad hoc server, but only for approximately half of each month.

- A third challenge was the frequent need to create multiple copies of data across the Development, Production, and Ad hoc areas. The previous environment lacked functionality for sharing data via a shared file system. Business users had frequent need to access Production data in the process of performing analysis, requiring the month end data to be copied from the Production server to the other environments, especially the Ad hoc server. Also, there was often movement of data from the Production to the Development areas when the models in Production required changes or fixes. To address this challenge, a major component of our planned Grid infrastructure was the implementation of General Parallel File System (GPFS a shared file system that is available to all nodes on the Grid and is capable of sustaining the I/O bandwidth required by our SAS environment.

- The fourth challenge was the cost of upgrades – as we added more users and wanted to add CPU, memory, or storage, the cost was very high and required downtime on all partitions. Additionally, not only were the maintenance costs of the partitioned frame quite expensive, but the cost of replacing that server with something similar, but capable of adequately meeting our growing needs was prohibitively expensive.

WHY GRID?

The planned Grid environment offered several advantages over other architectures and addressed the challenges faced in our old environment. From the perspective of Business Technology, our primary objective in converting the SAS environment to a Grid was to provide a highly available, cost effective and readily scalable environment for our Business Users. And the addition of the GPFS shared file system minimized the duplication of the data by various departments, and across the Development, Production and Ad hoc areas. From the perspective of our Business partners, a Grid implementation offered several advantages as well. Our Business Partners desired to drive more business value with a new environment, and A SAS Grid offered this:

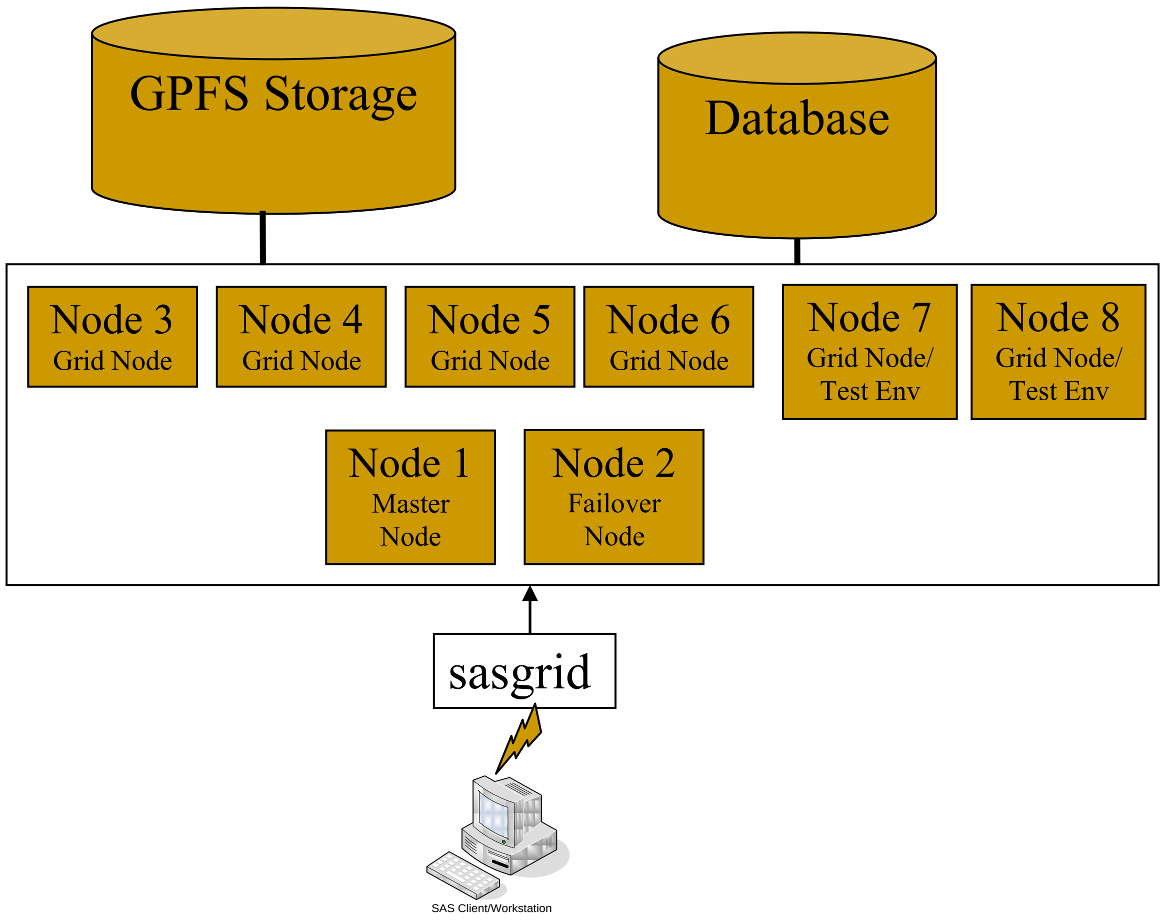
- A Grid/GPFS implementation offered cost-effective scalability to:
    - Absorb growth in the number of users
    - Absorb the ever-growing data needs of an analytical application
    - Accelerate ROI through improvements to the Model Development process and more efficient execution of end of month processing
    - Improve on ability to run larger and more complex analytics
    - Provide greater reliability to business processing while providing the flexibility to incrementally grow the environment without system downtime.

- Eliminate the single point of failure inherent in the current environment to guarantee continuous availability

Additionally, our former SAS environment was housed at a vendor site and was up for refresh. The decision to re-host at our in-house Production Center from an external vendor facility promised to save significant amounts of money long-term. Once we came to understand the refresh expenses associated with staying with an external vendor versus moving to an in-house solution, an internal Grid strategy became quite enticing.

GRID IMPLEMENTATION

We have implemented the SAS GRID for Production on eight servers/nodes with the following architecture.



All the eight grid nodes are connected via private network to a shared file system, which is using General Parallel File System (GPFS). The functionality of the nodes can be divided into three categories – Master node, Failover node and the Grid node. Master node is the essential node which controls the entire environment – from user connections to the jobs submissions. Failover node acts as a grid node while the Master node is up and running and will act as a Master node in the event of failover (when the master node dies or taken off line for maintenance). Grid nodes essentially act as worker nodes, taking their instructions from the master node which evenly distributes jobs to the Grid nodes for execution.

All the SAS EBI clients connect to the GRID through a virtual IP called sasgrid, which automatically connects them to the master node. The Master node has the entire suite of EBI server applications installed and configured including metadata, application tiers, and middle tiers and also the Platform suite for LSF. LSF is the back bone for the job scheduling, job submissions and the failover. To keep the Master node more available to manage the Grid, job slots on it are closed and the object spawners and work space servers are load balanced on to all other nodes. The object spawner on any one of these nodes is designated as a parent and all other object spawners on other nodes are designated as children. This eliminates the necessity for the master node to manage several hundred sessions from the EBI clients (from our business users).

Our business users also use Xclients like Exceed for job submissions. These applications tend to take up lot of resources on the servers and hence the Master node can't be used to manage these connections. A need to create a separate virtual IP which we named sasenv was identified and created. This virtual IP has been designated for all Xclient sessions. SAS users running Xclient sessions are instructed to utilize this virtual server, which routes traffic to all active grid nodes except the master node, in a round robin method. Compliance with this rule is enforced via a script that monitors for violations, and users who fail to follow this rule run the risk of having their X sessions terminated.

The Test environment is also mounted on two of the grid nodes, Nodes 6 and 7, which are also Grid nodes on the Production Grid. This was done by creating a separate virtual IP, called - sasgridtest. This is similar to the Production environment – Node 6 is designated as a master node for the Test Environment, while Node 7 acts as both the Master Failover node and the grid node. The sasgridtest virtual IP test environment was setup to be used by technology resources to be employed for activities like hot fix testing and maintenance upgrades. The 6th and 7th nodes can be removed from the Production Grid if necessary, or sasgridtest can be run in parallel with the full Grid, with the server nodes in the test grid remaining designated as part of the full grid environment.

As the three servers from our former environment – Adhoc, Production and Development were combined into one Production environment, there was need a need to prioritize actual Production month end jobs that need to meet SLAs. For this purpose, Queues have been implemented and all month end processing jobs were placed in the high priority queue, while ad hoc and development jobs remain a normal queue. The high priority queue jobs are always given precedence on the Grid and immediately submitted from the Master node to the grid nodes for execution. If resources are needed for the high priority jobs, jobs in the normal queue are moved to a pending state and only re-released for processing after the high priority jobs are completed and resources become available. Any new jobs submitted to the Grid during month end processing are also moved to the pending state until resources are available or the high priority jobs are completed. Jobs can be moved between these queues dynamically and several different queues with varying degrees of priority can be implemented based on need.

Existing SAS jobs do not need to be modified to be run on the Grid. There is a pre-code and an optional post-code that can be included in Enterprise Guide sessions or in the batch scripts that submit the jobs to the server. Job run time can be dramatically reduced by parallelizing jobs to take advantage of the Grid architecture. SAS provides a procedure called – *Proc scaproc*, which can be employed to automatically parallelize sas code to run on multiple grid nodes.

The job slots and the CPU thresholds also need to be configured after the install. The suggested configuration is to increase the number of job slots for each of the machines and also implement CPU utilization thresholds for each of the machines. Increasing the number of job slots will allow a larger number of simultaneous SAS sessions on each grid node. This is desirable because the workload of each session will be sporadic as compared to a large batch job that is I/O and compute intensive for a prolonged period of time. Setting the CPU utilization on each machine will ensure that if many users submit CPU intensive work at the same time, the SAS Grid

Manager will suspend some of the jobs while allowing other jobs to complete, and then resuming the suspended jobs when resources become available. This serves to prevent resource overload and avert potential hardware failures. All the modifications on resource allocations for queues, jobs or hosts can be dynamically modified and will not require a reboot of the nodes for changes to take effect.

The shared file system also minimizes the duplication of the data across our Business Partners data storage areas and across the Development and Production areas. The same file paths can now be utilized by the business users and the developers when they are developing models, or when business users validate the data, they can access data directly on the shared file system.

GRID EXPANSION

The Grid environment was expanded after the initial migration to accommodate the user growth and the demand for the storage. After the initial Grid deployment, the SAS Grid consisted of seven nodes. To address rapid growth in the size of our user community, and in order to test out the purported scalability of our new environment, a new node was added to our grid as a grid node. This 8th node was installed, configured and tested for all the connectivity with the databases, outside servers through firewalls etc. and successfully incorporated into the Grid seamlessly without any downtime on the environment or interruptions to user jobs or disruptions to processing. The same was true when a new storage appliance was added to double to storage capacity. The new appliance was setup, tested and then added to the cluster without the need to reboot the servers or remount the file systems. This ability to perform maintenance, enhancements and deploy additional server nodes and entire storage facilities without interrupting service to the environment displays one of the greatest advantages of the Grid architecture over other designs.

UPGRADING THE ENVIRONMENT

At the time of writing this paper, we are in the process of upgrading the SAS software from 9.2 M0 to 9.2 M3. We have taken the approach of installing the 9.2 M3 instance side-by-side with 9.2 M0 and running the two environments in parallel. Once we have completed testing and are satisfied with the M3 environments ability to replace M0, we will then switch our users over to the M3 environment. We should be able to seamlessly transfer our users from M0 to the M3 environment. On the sasgridtest virtual IP, M3 was installed, configured and tested on a side by side with M0 before we began implementing M3 on full SAS Grid.

LESSONS LEARNED

We have learned these lessons as we have implemented the Grid environment at our site. Some of these have greatly improved the performance on the Grid.

1) The Master node should be used exclusively for high availability and to manage the Grid environment. It should not be used to run any jobs on it, as running jobs on the Master node can create bottlenecks on the entire Grid and result in varying response times from the master nodes to the clients. And this could pose issues to the stability of the entire Grid environment. The Master node should be used to maintain connections to users' sessions submit jobs to the grid and maintain the metadata, which is an in-memory process. Slow response times on the Master node could corrupt metadata and increase the job submission times for the users, which could be frustrating for the users. Xclients should never be allowed to connect to the Master node, as the connections from these sessions take up an inordinate amount of resources.

2) Home directories on the shared file system are needed for the Grid environment. By default, operating systems create home directories on the local disk and when the users create ssh keys, they are stored in those home directories. We found that because we were using a shared file system on a grid, this could cause issues when users needed to transfer files from or to the Grid

using the ssh keys, as the users jobs could be running from any grid node based on the load on the Grid. So, having the home directories on the shared disk facilitates that users store their ssh keys in a centralized location which can be accessed from any grid node. This also enables the users who use Xclients for sas submissions to customize their operating system profiles on a shared home directory, so that they can have a uniform environment whenever they login to the Grid.

3) This suggestion is specifically for clients using a GPFS file system. On GPFS file systems, it is recommended to have one large file system and several filesets within it so that you can get more usable space from the disk. The advantage with the filesets is that you can set the quotas on them dynamically without the need for un-mounting or re-mounting. As the file system is large, the standard Unix commands will take lot of time to execute. Hence, we strongly recommend using the GPFS commands instead of standard Unix whenever possible. E.g. Use Mmlsquota instead of du.

4) Separate the access for SAS clients and Xcients to the Grid. Xclients like Exceed takes lot of resources on the servers and hence, should not be allowed to login to the Master node directly, as it could cause serious stability issues to the entire Grid environment. So, Xclients logins can be distributed across the grid nodes so that the terminal load on the servers is equally distributed on the Grid. SAS Clients could continue to use the master node for submitting the jobs.

5) A small Test Grid environment which is a subset of the Production Grid environment should be implemented in your Grid environment. This provides a place to test all changes and enhancements before implementing them into the Production instance, and it negates the necessity to have a separate Test environment for testing new and upgraded software. This is a very cost effective approach, as the test environment nodes are fully utilized at any point of time in the Production Grid and can be activated only when they are in need. Also, these nodes can be taken out of the Production Grid if there is a need to upgrade the Operating System (OS) or any software.

6) Do not combine Grid and Non-grid enabled applications – You may have other applications running on the Grid environment along with SAS. The Platform license that is distributed through SAS, can only manage SAS applications on the Grid. Other applications running on the Grid will give undesired results. It’s best to limit all Non-Grid applications to run on one node on your grid. And that node should not be either the Master or Fail-over nodes.

7) 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.

8) Crontab jobs are scheduled locally on the server. Since there are multiple nodes on the grid, the users need to keep track of where their jobs are scheduled.

CONCLUSION

Implementing a SAS Grid along with a GPFS file system has helped us to significantly increase productivity and improve cost. It has allowed us to exploit the previously under-utilized power of our computing resources while providing a more stable and scalable environment. Grid computing has proven to be a cost effective approach to providing large amounts of computer processing cycles to our Business Partners. While the implementation of the Grid and GPFS presented some significant challenges, given that grid computing is in its early stages of development, some of this was to be expected. But for applications like large analytical environments that lend themselves to a grid implementation, a SAS Grid can be used to tap unused capacity and reduce total execution time. Our SAS/GPFS implementation has provided our Enterprise with:

- GPFS Clustered storage that enables sharing of storage across platforms, reducing the overall amount of disk storage necessary for user community, and production model execution
- Scalability – Our initial outlay is far less than replacing the existing environment with 'like' hardware software. It has proven to be less expensive and easier to grow the environment
- Confidence that further refreshes will not incur environmental downtime. Individual nodes can be varied off-line and replaced one by one, as time and money permits
- Addition of a new compute node and additional storage has proven that computing resources can be scaled out to cost-effectively add new users and capacity with no impact to existing activities and user
- Load balancing and grid management across nodes leads to effective utilization of all available nodes while providing automated optimization for SAS application processing based on custom rules based routing logic
- Multiprocessing capabilities allow work flows to be processed in parallel on multiple machines, reducing SLA's of SAS model production and end of month processing
- The Enterprise SAS Environment refresh into our in-house Production Center from an external vendor facility was successfully implemented in a phased approach, with fewer risks, reducing impact and risk to SAS user community

The outcome of our production SAS Grid implementation is that it has been a success. We are able to run all of our current production SAS applications with no resource downtime during our last year. In that time, we have increased our user base by almost 50% and scaled our environment, adding a new compute node and an additional storage array, while suffering no disruption of service. Our organization has successfully migrated to an Enterprise SAS Grid Platform that ideally addresses our business requirements and technology goals.

Contact Information

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

Shivashanker Bitla
2500 Lake Cook Road, Riverwoods, IL 60015
Phone: (630) 212-4776
E-mail: sbitla@discover.com

Richard A. Kerwin Jr.
2500 Lake Cook Road, Riverwoods, IL 60015
Phone: (224) 405-5547
E-mail: rkerwin@discover.com