Paper 227-29

SAS® and Grid Computing – Maximize Efficiency, Lower Total Cost of Ownership

Cheryl Doninger, SAS Institute, Cary, NC

ABSTRACT

IT budgets are declining and data continues to grow at an exponential rate. SAS applications with high volumes of data can take many hours or possibly days or even weeks to complete. In some cases, a job is so big that it cannot be completed at all even given today's processor speeds. Whether you call it grid computing or not, you want to be able to complete your SAS applications in a reasonable amount of time without making a huge investment in new hardware. Grid computing is about leveraging your available resources and idle processor cycles to more quickly solve a problem while at the same time maximizing efficiency and reducing your total cost of ownership. SAS enables you to distribute long running jobs across a grid of any number of homogeneous or heterogeneous machines in order to get timely results and get information to the right people at the right time. This paper will discuss how SAS works in a grid, the types of applications that are well suited to grid computing, and success stories using SAS in a grid.

INTRODUCTION TO GRID COMPUTING

Grid computing began in the academic research community a few years ago when researchers needed to process large amounts of data as quickly as possible for data-intensive projects. The concept is led by Dr. Ian Foster and Dr. Carl Kesselman who pioneered the creation of the Globus Project for grid development software. Basically, grid computing is the next level of distributed computing. It is an innovative approach that leverages existing IT infrastructure to optimize compute resources and manage data and computing workloads. Grids can provide different levels of services. You may have an application which runs for a very long time because it is either very compute intensive, very data intensive, or both. If the problem is one that can be decomposed into sub-units of work these sub-units can be distributed across multiple remote machines, possibly heterogeneous machines, to be executed in parallel followed by an aggregation of the intermediate results. The benefit of this type of large scale parallel execution is a significantly shorter overall execution time. This type of grid is called a *compute grid.* On the other hand, you may have a collection of hardware resources that you would like virtualized so that multiple users can submit their individual applications to this pool of resources and have the benefit of load balancing, maximum use of resources, and optimal turnaround time for each individual application. This type of grid is often called a *utility grid.* With a utility grid each individual application executes on a single node in the grid but the client submitting the job would have no knowledge of which node actually executed the job.

SAS views grid computing as a means to apply the resources from a collection of computers in a network and to harness all the compute power into a single project. Grid computing can be a cost effective way to resolve IT issues in the areas of data, computing and collaboration; especially if they require enormous amounts of compute power, complex computer processing cycles or access to large data sources. SAS additionally believes that grid computing needs to be a secure, coordinated sharing of heterogeneous computing resources across a networked environment that allows users to get their answers faster.

Although the concept of grid computing has been generating a lot of new interest and attention in the IT world, it is not a new concept. However, there are new economic as well as business factors that are contributing to the attention around the development and implementation of grid computing solutions. Because of the Internet and the way business is conducted today, we are being inundated with data. As the data flood gates open wider, the window of opportunity for capturing and turning this data into information grows shorter and shorter. Computing problems in many industries involve processing large volumes of data and/or performing repetitive computations that exceed existing server platform capabilities. In addition, while sampling might have sufficed in the past for modeling and risk analysis applications, today's business requirements often demand a much larger sample size or perhaps use of the entire data source for maximum accuracy. Also the challenges that IT shops are facing today including budget cuts, underutilized hardware, server consolidation, hardware provisioning, and overall administration are all factors driving interest and implementation of grid computing. Grid computing addresses all of these factors and provides many benefits to lower your total cost of ownership.

Grid computing has grown out of the academic and scientific community's need to build very large cost effective computing systems. However, it is no longer a topic just in the academic community. Many different industries are seeing a need for grid computing and have begun to implement grid solutions. These industries include but are not limited to:

- Manufacturing. The requirement in both high-tech and industrial manufacturing is driven by the need to run many models over and over to optimize design.

- Financial. Banks deal with large amounts of data for their credit card processing, and investment banking runs applications that require fast response time to many simultaneous users.
- Life Sciences. The requirement is driven by the need to run large numbers of models for drug design and to perform genetic studies that involve potentially millions of gene combinations (BY groups).
- Public Sector. Many universities conduct highly compute intensive research projects as well as want to maximize the use of their existing hardware resources in the event of limited funding for new additional resources.

While these industries are early adopters of grid computing, other verticals are following suite including telecommunication, retail, pharmaceuticals, etc.

WHICH APPLICATIONS SHOULD BE GRID ENABLED

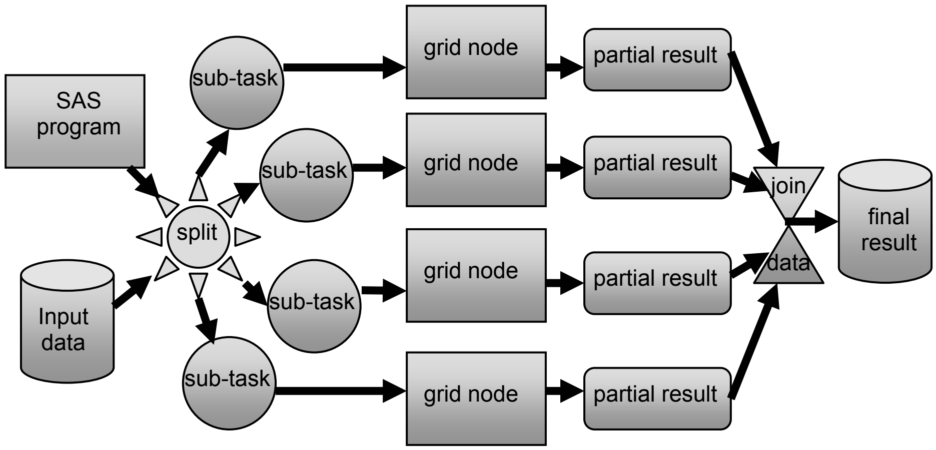
Typically, applications that are good candidates for a grid implementation take many hours and possibly even days or weeks to run. In some cases, the job is so big that it cannot be completed at all even given today's processor speeds. Jobs that take an extraordinary amount of time to execute usually do so either because they are processing very large amounts of data (on the order of many gigabytes or more), they require running the same steps over and over on either all of the data or many different subsets of the data, or both. A compute grid can be used to distribute and manage parallel pieces of such an application in order to execute the application in a fraction of the time that it would take on a single processor. The best part is that this reduced execution time is accomplished using idle resources across your network.

It is important to clearly define the types of applications that lend themselves to a compute grid implementation so that realistic expectations can be set and goals can be met. Not all applications can benefit from a grid implementation. In order for a single application to be grid enabled either the data or the execution logic must be decomposable into sub-units of work so that these sub-units of work can be distributed across the nodes in the grid and executed in parallel. In general an application would possess one or more of the following characteristics in order for a compute grid implementation to be considered:

- takes a long time to execute
- involves many replicate runs of the same fundamental task
- processes large amounts of data
- decomposable execution and/or data.

Figure 1 below illustrates how an application might be decomposed into sub-units of work that can be distributed in parallel across the nodes in a grid. The partial results are then aggregated to form the final result.

FIGURE 1: GRID ENABLING A SAS APPLICATION



Many applications involve repeating the same fundamental task many times against unique subsets of the data. In SAS terms, this equates to BY GROUP processing. While the execution of a single task against a single subset of the data may execute rather quickly, if you have to do this execution many times against hundreds, thousands, or even millions of subsets of the data, it can become extremely time intensive. These types of applications are often referred to as *embarrassingly parallel* and are very well suited to a grid implementation because the replicate tasks can be distributed across the grid to be executed in parallel and greatly reduce the total elapsed execution time. Other applications that fall into this category would be:

- Searching for optimal designs. For example, global integer optimization problems are often solved by steepest ascent (the fundamental task) with random restarts (the replicate runs).
- Statistical simulations. For example, Monte Carlo methods in statistics rely on a set of pseudo-random observations (the replicate runs) of one or more difficult-to-compute test statistics (the fundamental task).
- Applying any analysis or statistical model (the fundamental task) to subsets of the data or BY GROUPS (the replicate runs).
- Mining massive data sets.

Each of the fundamental tasks that get distributed across the grid must have access to all required input data. Sometimes the input data may be small (on the order of megabytes) and other times the data may be large (on the order of many gigabytes). In order to achieve the highest efficiency, the compute nodes should spend the majority of the time computing rather than communicating. Compute tasks that require substantial data movement generally do not perform well in a grid. Therefore the data must either be distributed to the nodes prior to running the application or much more commonly, made available via shared network storage. There have been many recent advances in data storage hardware that provide fast read access to data and help contribute to the success of a grid.

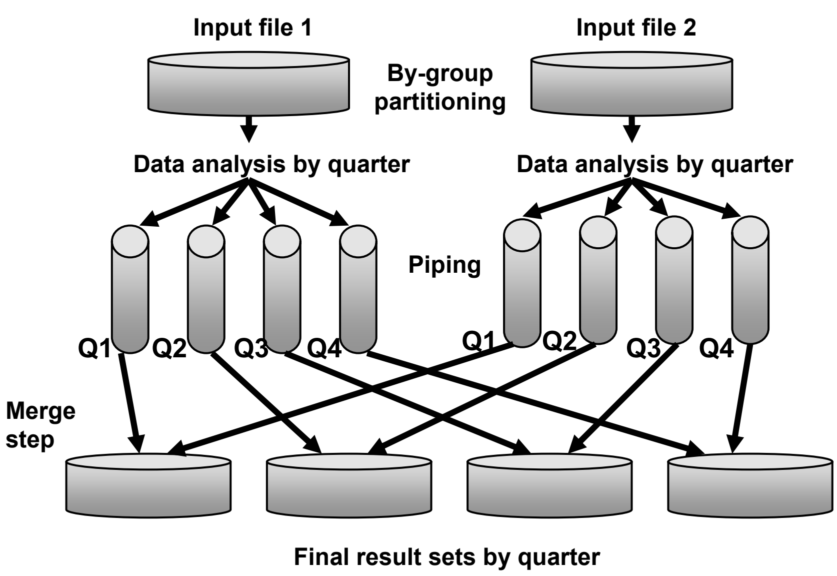
SAS AS THE GRID ENABLER

SAS/CONNECT software is a SAS-to-SAS client/server toolset that provides scalability through parallel SAS processes and provides users, applications developers, and SAS solutions the ability to manage, access, and process data in a distributed environment. SAS/CONNECT provides scalability to address the ever increasing time-to-market and time-to-customer needs for data delivery and business intelligence. It also provides interoperability between different SAS releases running on any number of heterogeneous platforms. All of these features make it an ideal middleware to enable SAS processing on a compute grid.

Since Version 8, SAS/CONNECT has provided a syntactic interface that can be used to partition large jobs into independent tasks that can be performed in parallel. This type of parallelism is called *independent parallelism*. By distributing these tasks across a grid and executing them in parallel, a job can be performed in a fraction of the time required for executing the job sequentially on a single machine. With SAS 9, SAS/CONNECT added support for *pipeline parallelism*. This allows dependent steps to overlap their execution by having the output of one process piped directly into the next process as input. This not only reduces overall elapsed execution time by allowing dependent steps to execute in parallel but also reduces disk space requirements by eliminating the need for the intermediate write to disk. Piping can be used to "chain" together any number of dependent processes.

Figure 2 below illustrates how independent parallelism allows the distribution of BY-group processing across the nodes in a grid. In addition, piping can be used to further overlap the processing and avoid unnecessary disk I/O.

FIGURE 2: INDEPENDENT ("BY-GROUP) AND PIPELINE PARALLELISM



The capabilities of SAS/CONNECT can be leveraged to coordinate the analytical power of the SAS System running simultaneously on any number of servers across a grid. A series of macros called %Distribute has been written to utilize SAS/CONNECT and provide the framework and middleware for a grid application. The following features are available using SAS/CONNECT for running in a grid environment:

- A mechanism for defining the grid nodes as well as the input data sources and output data locations
- A mechanism to define the processing that is to be distributed
- A load balancing scheme to improve the scaling efficiency of the parallel computation
- A report of the activity of each node in the grid
- The collection of partial result sets from the nodes in the grid back to a centralized location
- Optional additional analysis of the collected results

A major benefit of enabling SAS grid computing with SAS/CONNECT is that it puts the capability at the intelligence architecture layer as part of our toolset. This makes it available to be leveraged by both our SAS programmer customer base as well as all of the SAS solutions and vertical products developed internally by SAS. This not only provides a common implementation but also allows us to more quickly grid enable the SAS solutions that can benefit from a compute grid.

Running SAS in a compute grid requires that BASE SAS and SAS/CONNECT be pre-installed on all of the nodes in the grid. If any additional SAS products are required by the processing that is to be distributed, such as SAS/STAT, then those products would also need to be pre-installed on the nodes in the grid. Once this is done, SAS/CONNECT can be used to invoke SAS and enable the distribution of the work across the grid.

CASE STUDIES

SAS/CONNECT has been used by many customers to accomplish dramatic reductions in execution times. These applications have been run in a variety of environments including a grid of laptops, a Linux cluster, and a grid of over 200 heterogeneous Windows and Unix platforms. In addition, these applications have been run using Version 8, SAS 9, as well as a combination of the two.

OPTIMIZATION IN A GRID OF PC LAPTOPS

One case study is a SAS customer using SAS/CONNECT on a dedicated grid of 60 laptops (ranging from 266-400 MHz). The grid is running an optimization of 600 sales territories in about 2 hours. This same job previously took 26 hours on an NT server and 15 hours on a powerful Unix server. In this instance, computer hardware considered obsolete and slow was able to be used to significantly benefit an operating arm of this customer's company so they could get their reporting done in a time-frame consistent with their needs. Needless to say that by using obsolete hardware, this customer realized a very high return on investment and was able to do more with less.

MICROARRAY STUDIES IN A GRID

SAS/CONNECT has also been used for the purpose of parallelizing data analysis of a toxicogenomics microarray study. In this example, data was utilized from microarrays, a popular new scientific instrumentation platform enabling the assessment of the expression levels of thousands of genes simultaneously. The data are from an investigation conducted at the Microarray Center of the National Institute of Environmental Health Sciences (NIEHS), Research Triangle Park, NC.

After suitably normalizing the data from the microarrays, a common initial analysis is to fit a statistical model to data from each of the 1700 genes separately. While this kind of basic analysis is well-suited for parallelization, it can usually be completed on a single CPU in a few minutes using BY processing. Here, though, we consider models for (potentially all) possible pairs of genes, models which are both more complex and whose number in this case exceeds 1.4 million. Completing this analysis in a reasonable amount of time on a single CPU is impossible at today's clock speeds.

The folks at NIEHS were quite pleased with the 97% speed up that they were able to achieve using SAS in a grid. "Grid computing has enabled NIEHS to see an impressive breakthrough in statistical analysis computations, which aides our scientists in quickly examining their data," said Roy Reter, IT security officer and systems administrator for NIEHS' Division of Intramural Research. "SAS software has been critical in making this reduction in execution time possible," added John Grovenstein, computational scientist at the NIEHS. "Through SAS' involvement with the Global Grid Forum, the NIEHS will stay at the forefront of new developments in grid computing technology and standards. This will enable us to more quickly deliver critical results that impact our environmental health."
This project was run in two environments: the first is a 32 node Linux cluster on-site at the NIEHS, the second was on

a 100 node heterogeneous grid at SAS. The results of these two environments follow:

32 node Linux cluster

- 32 node Linux cluster running Mosix
- 1 Ghz Intel P3 processors, 1 G RAM per processor, 100 Mb backplane
- 14.5 hours total elapsed time
- 448 hours required to run on a single node
- 97% improvement in total elapsed time.

100 node heterogeneous grid

- 100 nodes running mixture of W2K, WXP, variety of Unix OS's
- combination of v8 SAS and SAS 9 on nodes in the grid
- 5.25 hours total elapsed time
- 469 hours required to run on average node in the grid
- 99% improvement in total elapsed time

NUMERICAL SIMULATION IN A GRID OF UNIX WORKSTATIONS

Finally, SAS software and software quality has greatly benefited from grid computing by using SAS/CONNECT to execute simulations in order to address important problems in statistical testing for analysis of variance. The accuracy and completeness of these simulation approaches would not have been feasible or even possible without the distributed parallel processing capabilities of SAS/CONNECT. Using SAS in a grid made it possible to condense three years' worth of computing time into just a month's worth of desktop time. This application was run on as many as 150 Unix servers simultaneously.

ADVANTAGES TO SAS GRID COMPUTING

Just as there are different uses of grids there are also different choices for implementing a grid computing solution. For a SAS application it is possible to implement your grid solution completely within SAS using SAS/CONNECT as the grid middleware. The advantages are:

- A SAS centric solution does not require a third party grid middleware so there are no additional products to purchase, configure, and support.
- SAS/CONNECT is able to leverage the complete application knowledge of the SAS System. This means that the distribution logic can be part of the SAS application rather than having to split the SAS program into multiple, possibly hundreds, of individual files. The disadvantage to splitting the SAS program into many files is that they would have to be individually maintained as well as defined through a third party grid middleware for scheduling and distribution. It would not only be very time consuming to manually sub-divide an application into possibly hundreds of individual files but would also be a substantial burden to maintain and update such a large number of files. Also, some SAS applications may be very difficult to split apart into individual files.
- In addition, SAS/CONNECT can provide any data transfer that may be necessary before, during, and after execution and the SAS environment is available for any post processing or aggregation of partial results.
- A SAS centric grid solution using SAS/CONNECT is the only solution that can leverage all of the platforms supported by SAS including: several flavors of Unix, Windows, Alpha/VMS and Z/OS.
- A SAS grid solution using SAS/CONNECT balances the execution load between faster and slower machines in the grid such that the faster machines get more of the work in order to minimize overall execution time.

FUTURE DIRECTION FOR SAS AND GRID COMPUTING

SAS/CONNECT is a proven grid middleware for enabling SAS processing across a compute grid in order to execute SAS applications faster. We have many customers using this technology today to lead the forefront of the grid computing wave and give them a competitive advantage over other organizations. However, there are features and capabilities that we need to add or improve in order to make SAS a more complete grid solution. One of the ways that we intend to accomplish this is by integrating with some of the industry leading grid middleware vendors to leverage the additional capabilities that they provide and surface these capabilities through a SAS application. Once this integration is complete, the solution will offer a user-friendly interface to configure the machines within the grid, let you set up your SAS application to utilize the parallel distribution capabilities, and support the execution of SAS and non-SAS applications across a grid environment.

In addition, SAS is also working on providing utility grid capabilities. This will enable an organization to create and optimize a virtual pool of resources allowing concurrent SAS users to submit jobs more efficiently and effectively. This type of grid would provide better turn around time for each individual job, sharing of resources, improved service levels, and more efficient use of these resources.

CONCLUSION

Grid computing is an emerging technology that will have a major impact on productivity and cost improvements at the enterprise level. It exploits un-utilized or under-utilized power of all computing resources within a network environment – including desktop PCs. SAS defines grid computing as a means to apply the resources from a collection of computers in a network and to harness all the compute power into a single project. Grid computing is a cost effective approach to resolve IT issues in the areas of data, computing and collaboration, especially if they require enormous amounts of complex computer processing cycles. While the development and implementation of grid computing is in the early phase, we expect the need and adoption of the grid computing market to increase rapidly over the next several years. For those applications that lend themselves to a grid implementation, SAS and SAS/CONNECT can be used in a grid to tap unused capacity and drastically reduce total elapsed execution time. This is proven by the customer results documented above. The good news is that SAS can leverage a grid environment today and we are in a great position to advance our capabilities as the grid computing industry matures.

ACKNOWLEDGMENTS

I would like to acknowledge the scientists and IT staff at National Institute of Environmental Health Sciences for sharing their experiences and successes of running SAS in a grid. I would like to specifically thank Roy Reter and John Grovenstein.

I would like to acknowledge Tho Nguyen, Marketing Strategist at SAS Institute Inc., for his help in developing a SAS grid strategy as well as for his input on this paper.

I would also like to acknowledge Randy Tobias, Manager of Linear Models at SAS Institute Inc., for authoring the %Distribute macro system.

FOR MORE INFORMATION

For more information about SAS and grid computing, visit the SAS Scalability Community at:
http://support.sas.com/rnd/scalability/grid

For more information about the %Distribute macro system and to obtain a copy of the macros visit the SAS Scalability Community at:
http://support.sas.com/rnd/scalability/papers
under the heading "The %Distribute System for Large-Scale Parallel Computation in the SAS System".

CONTACT INFORMATION

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

Cheryl Doninger
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
(919) 531-7941
Email: cheryl.doninger@sas.com
Web: http://support.sas.com/scalability/grid