Paper 098-2009

Data Integration in a Grid-Enabled Environment

Cheryl Doninger, SAS Institute Inc., Cary, NC
Nancy Rausch, SAS Institute Inc., Cary, NC

ABSTRACT

SAS® Data Integration Studio and SAS® Grid Manager add capabilities to the SAS® product suite to automatically and efficiently distribute workloads across a grid of computers to allow large processes to complete more quickly than previously possible. SAS Grid Manager has been incorporated into SAS Data Integration Studio to make it easy to use grid resources for any long-running task that can be processed in parallel. This paper discusses typical data integration workloads that can be made parallel and how they scale on typical grid computing hardware as well as the new capability to load-balance multiple data integration tasks across grid resources.

INTRODUCTION

SAS was originally developed as a 4GL programming language that could be used to write SAS applications. These applications executed in a single, sequential path that matched the single CPU capabilities at that time.

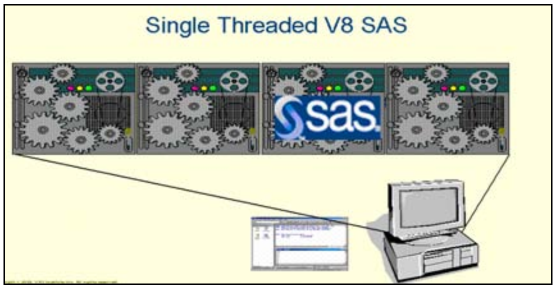


Figure 1. Single Threaded SAS® 8 Processing Capabilities.

As data volumes continued to grow and computing needs continued to increase, hardware vendors responded by creating symmetric multiprocessor (SMP) machines. SMP is a multiprocessor computer architecture in which two or more identical processors are connected to a single, shared main memory. Organizations also built networked computing environments with multiple, individual computing resources connected via networking protocols. SAS responded to these advances by developing multiprocessing capabilities. In SAS/CONNECT® 8, SAS applications could take advantage of the multiprocessors available in desktop and server platforms, and could multiprocess across platforms available in a network. With SAS/CONNECT you can spawn N SAS sessions or processes, simultaneously execute N tasks as independent processes, and coordinate the execution and results into the client or parent session. A major benefit of this technology is the flexibility for the multiple sessions to run on multiple CPUs within an SMP box, across multiple, distributed machines in a network, or a combination of both.

SAS® 9 brought multithreading capabilities to further leverage the growing adoption of SMP architectures. On hardware with more than one CPU, multithreading provides a mechanism for a program to exploit more than one CPU simultaneously. By creating multiple, simultaneously active threads, the program enables the operating system to schedule these threads concurrently on more than one CPU. Several SAS procedures, including SORT, SUMMARY, REG, and GLM, as well as several SAS servers and SAS/ACCESS® engines were rewritten with multithreading capabilities. It is important to understand that while multithreading capabilities allow an application to exploit the multiple CPUs in an SMP machine, the scalability gains are limited to a single SMP box and cannot leverage distributed computing resources.

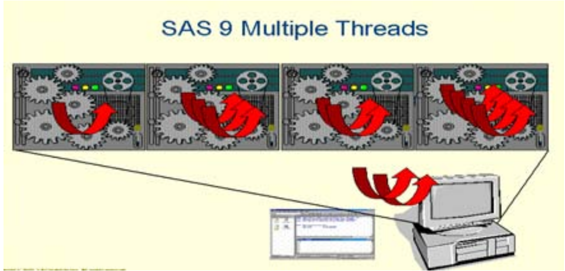


Figure 2. SAS® 9 Processing Capabilities.

The initial offering of SAS® 9 provided both multithreading and multiprocessing capabilities to allow SAS applications to scale-up to take advantage of multiprocessors available in SMP hardware. SAS also provided multiprocessing capabilities to allow SAS applications to scale-out to take advantage of any number of distributed computing resources. The next evolutionary step was for SAS to make it possible for customers to run their SAS applications in a grid or cluster environment. SAS Grid Manager was introduced in SAS® 9.1.3

to build upon the parallel capabilities of SAS/CONNECT and to add the many other requirements of enterprise grid deployments. SAS Grid Manager provides multiuser load balancing, policy enforcement, efficient resource allocation, and prioritization for SAS products and solutions running in a shared grid environment.

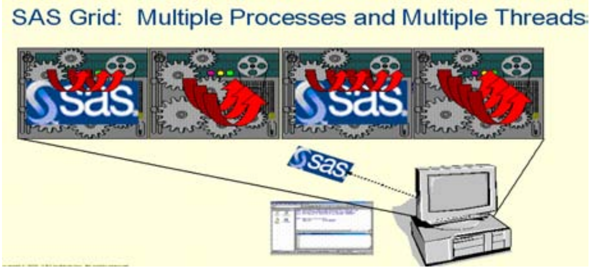


Figure 3. SAS® 9.1.3 Can Leverage Grid Processing.

SAS Grid Manager has been integrated with many SAS products and solutions to provide seamless grid capabilities to the users of these applications. One such application is SAS Data Integration Studio. This paper will discuss the multiple ways that SAS Data Integration Studio works with SAS Grid Manager to bring the benefits of a SAS grid infrastructure to the user in an easy-to-use, point-and-click development environment.

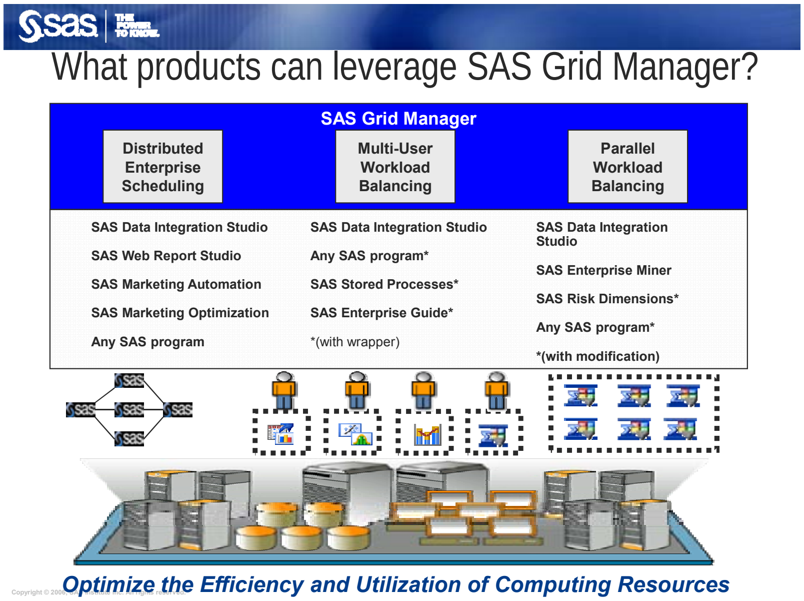
USING SAS GRID MANAGER IN SAS DATA INTEGRATION STUDIO

When running processes in a grid, SAS Grid Manager dynamically determines node availability and monitors grid nodes to determine which node is the best candidate to receive the next workload segment. This determination can be based on many factors, but it often considers the current load under which all grid nodes are running at any given time. The node that has the lowest CPU load becomes the best candidate on which to run the next workload segment. This dynamic capability greatly increases job runtime performance by distributing processes across a wider array of resources capable of handling the greater computing load.

There are three key ways that SAS programs and applications can exploit grid computing using SAS Grid Manager in SAS Data Integration Studio:

- Distributed Enterprise Scheduling—distribute scheduled jobs to a shared pool of resources in a grid
- Multiuser Workload Balancing—distribute user-submitted jobs interactively to a shared pool of resources in a grid
- Parallel Workload Balancing—distribute parallelized jobs, either scheduled or user submitted, to a shared pool of resources in a grid

Figure 4 below describes the SAS products that can exploit each of the features. SAS Data Integration Server can exploit all three capabilities.


What products can leverage SAS Grid Manager?

SAS Grid Manager

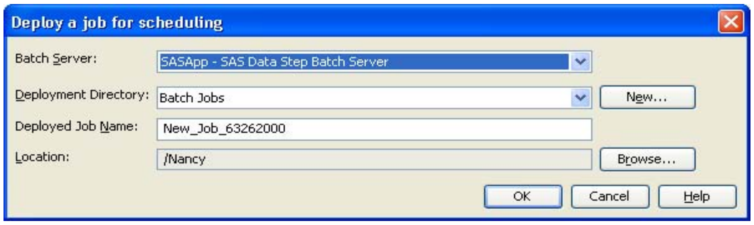
Distributed Enterprise Scheduling	Multi-User Workload Balancing	Parallel Workload Balancing
SAS Data Integration Studio	SAS Data Integration Studio	SAS Data Integration Studio
SAS Web Report Studio	Any SAS program*	SAS Enterprise Miner
SAS Marketing Automation	SAS Stored Processes*	SAS Risk Dimensions*
SAS Marketing Optimization	SAS Enterprise Guide*	Any SAS program*
Any SAS program	*(with wrapper)	*(with modification)

Optimize the Efficiency and Utilization of Computing Resources

Figure 4. Products in SAS That Can Exploit SAS Grid Computing.

DISTRIBUTED ENTERPRISE SCHEDULING

Using Distributed Enterprise Scheduling with SAS Grid Manager, scheduled jobs are targeted to run on the grid. This makes all resources in the grid available to run scheduled jobs. The scheduling server manages the resources in the grid workload so that jobs are efficiently distributed across the available machines in the grid. You can leverage this capability in SAS Data Integration Studio and other SAS products by deploying the job to be scheduled and then using the scheduling server that manages the grid to schedule and run the jobs.



Deploy a job for scheduling

Batch Server: SASApp - SAS Data Step Batch Server

Deployment Directory: Batch Jobs

Deployed Job Name: New_Job_63262000

Location: /Nancy

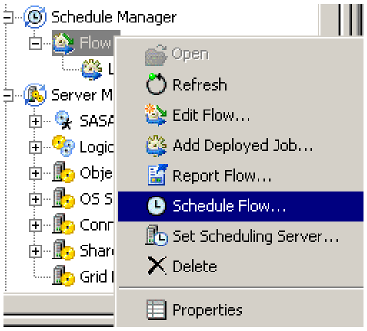


Figure 5. Using Schedule Manager for Grid Computing.

MULTIUSER WORKLOAD BALANCING

With Multiuser Workload balancing, individual users have the ability to submit jobs interactively and directly to the grid. For example, a site could have a number of users that do ad hoc development, such as model development, queries, and other sorts of discovery and analysis. SAS Grid Manager provides the ability to leverage the grid when submitting jobs. Using the grid provides all of the capabilities of load balancing, such as queuing, prioritization, workload balancing, and resource management, for this type of interactive submit. High-priority jobs can even preempt lower priority work so that the most critical business processes execute first. This enables users to leverage all of the available resources in their distributed environment for job processing, thereby speeding up long-running tasks and increasing user productivity.

SAS Data Integration Studio 4.2 supports this capability by enabling you to select the target server where you want to submit jobs or transformation steps in a job. The target server can be the grid server. When selected, SAS Data Integration Studio wrappers the submitted code with the appropriate statements to submit the interactive job to the grid.

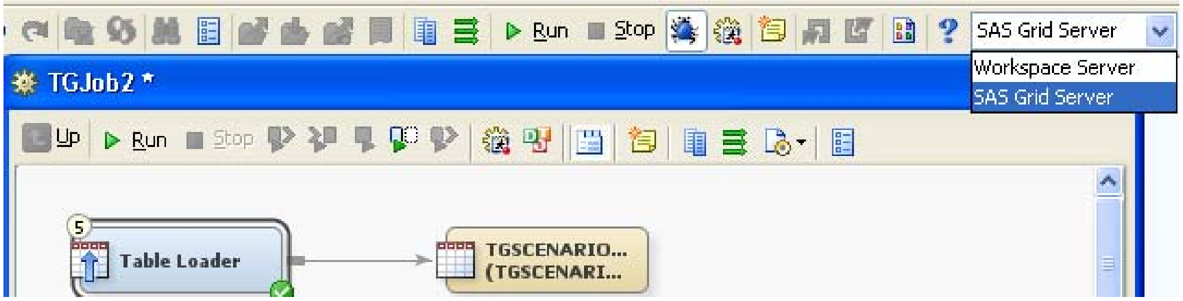
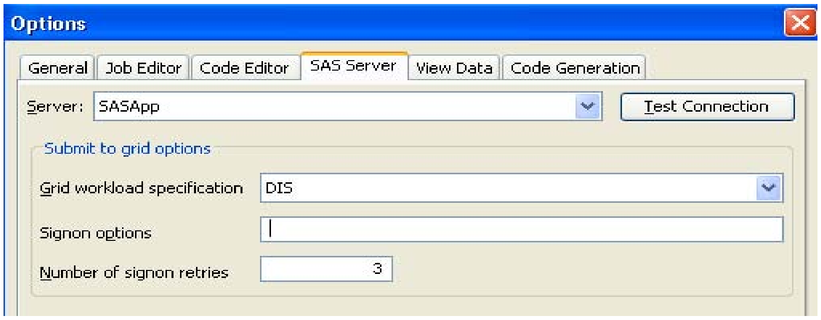


Figure 6. Using SAS Data Integration Studio's Server Selection Capabilities to Submit Jobs to a Grid.

SAS Grid Manager also supports the option to group like resources together into a group of nodes. For example, an administrator might want to configure one set of nodes for work with analytical applications, and another set of nodes for data integration processes. The administrator can assign a name to this partition so that users can specify it when submitting their processes. This partitioning ability allows the administrator to tailor the grid to better meet the needs of the user community. SAS Data Integration Studio allows you to specify an optional grid partition when you submit processes using workload balancing.



Server: SASApp

Grid workload specification: DIS

Signon options:

Number of signon retries: 3

Figure 7. SAS Data Integration Studio Supports Grid Partitions.

PARALLEL WORKLOAD BALANCING

SAS Data Integration Studio also supports the ability to parallelize processes, which can be submitted interactively using Parallel Workload balancing or can be scheduled as job flows using Distributed Enterprise Scheduling to run in parallel on a grid. Parallel execution of job flows is supported using iteration, a scenario frequently found in Data Integration processing. Iteration can be explained with a simple example.

Sometimes, it is desirable to execute the same process flow over and over again on different data. For example, suppose you have United States Census data as a set of 50 tables, one for each state. The table structure is identical, but the data is specific to each state:

- HouseholdsCA
- HouseholdsNC
- HouseholdsOR
- HouseholdsWA
- HouseholdsAZ
- HouseholdsNM

…additional tables

Now suppose you want to calculate the number of households that own more than one acre of land. For each state of data, you would run the same process on every source table. One way to process this data would be to run the process one run at a time per table. However, if you have the computing resources that would be available using a grid, you could run the same process in parallel on the different source tables. This is iteration, which means that the same process flow is run iteratively for each input in sequence. The scope of the iteration is called a loop. SAS Data Integration Studio includes a methodology for handling looping for jobs, both serially and in parallel. Each run is called an iteration of the job, and it can be submitted to run in parallel, either on a single machine or submitted to a grid to leverage the multiple computing nodes available on a grid.

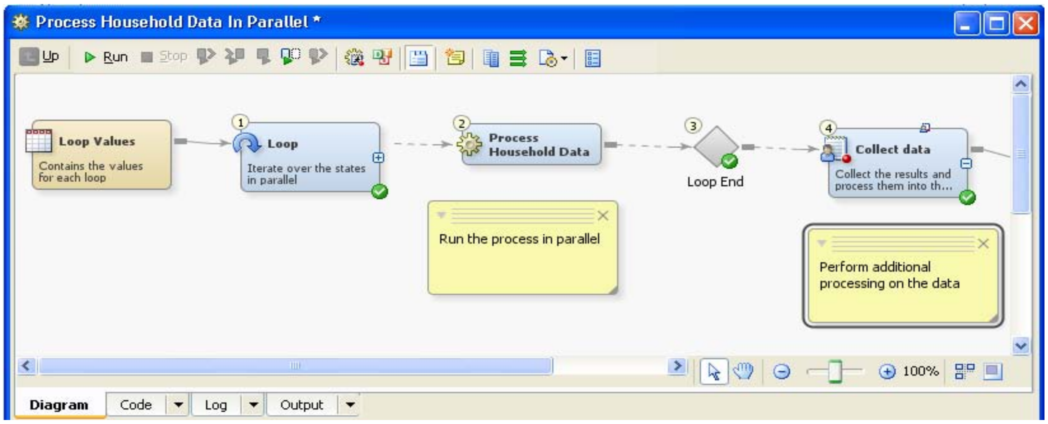
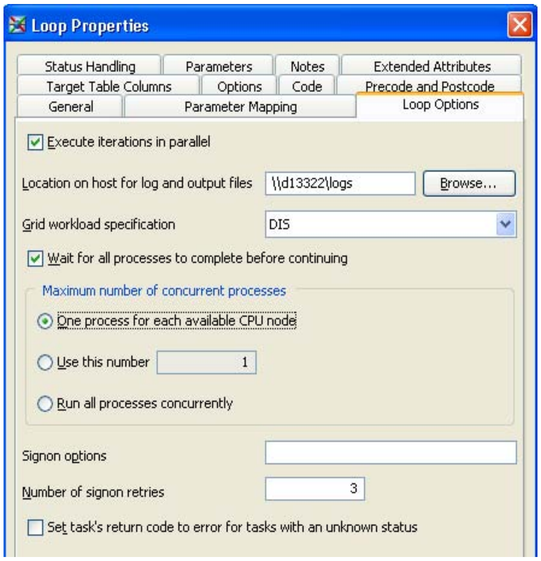

Process Household Data In Parallel *
Up
Run
Stop
Loop Values
Contains the values for each loop
1
Loop
Iterate over the states in parallel
2
Process Household Data
3
Loop End
4
Collect data
Collect the results and process them into th...
Run the process in parallel
Perform additional processing on the data
100%
Diagram
Code
Log
Output

Figure 8. SAS Data Integration Studio Job That Supports Parallel Processing of Data, One Process per Loop.

The loop transform lets you specify submit options to optimize the code to match the hardware that you will be submitting to. These options include dynamic discovery of available CPUs, optional grid workload configuration so that job submissions are sent only to the right set of available servers, sign-on retries, and other features.



☑ Execute iterations in parallel

Location on host for log and output files: \\d13322\logs

Grid workload specification: DIS

☑ Wait for all processes to complete before continuing

Maximum number of concurrent processes

● One process for each available CPU node

○ Use this number: 1

○ Run all processes concurrently

Signon options:

Number of signon retries: 3

□ Set task's return code to error for tasks with an unknown status

Figure 9. Options Available in the Loop Transform for Iteration.

SAS CODE ANALYZER

The SAS Code Analyzer is a tool that can help you run your existing SAS programs in a more efficient manner. The SAS Code Analyzer is run by calling a new SAS 9.2 procedure called PROC SCAPROC. This procedure will execute an existing SAS program and output information, as the program is running, about the input and output data sources and the interdependencies of the job steps. The output of the SAS Code Analyzer can be used to run and manage your SAS programs in a more efficient manner.

The SAS Code Analyzer can assist in:

- creating a grid-enabled parallel version of an existing SAS program
- importing SAS programs into SAS Data Integration Studio

It is important to note that the SAS Code Analyzer will actually run your original program to collect the detailed information. This allows the SAS Code Analyzer to work with any SAS programming statements including SAS macros and %INCLUDE statements. One best practice when using the SAS Code Analyzer is to consider using a subset of the data in your program. Using a smaller amount of data can speed up the analysis phase since the SAS Code Analyzer is executing the entire program.

CREATING A GRID-ENABLED PARALLEL VERSION OF AN EXISTING SAS PROGRAM

Some SAS programs contain multiple independent subtasks that can be distributed across the grid and executed in parallel. This approach can enable many applications to run much faster. To enable a SAS program to use distributed parallel processing, add RSUBMIT and ENDRSUBMIT statements around each subtask and add the GRDSVC_ENABLE function call to the beginning of the program. SAS Grid Manager automatically assigns each identified subtask to a grid node. However, depending on the length of your SAS program, how extensively it makes use of SAS macros and your familiarity with the structure of the code, this can be a time intensive process.

The SAS Code Analyzer can be used to create a grid-enabled parallel SAS job saving you the effort of manually analyzing individual procedure steps and interdependencies. When PROC SCAPROC is invoked in an existing SAS program, it will execute the logic of the existing SAS program and output information that identifies the input and output data sets and the interdependencies of the procedure and DATA steps in the program. It will then use this information to output a second grid-enabled version of the program that can be run across the grid via SAS Grid Manager. It is highly recommended that you review, run, and test the new grid-enabled version of the application before you consider it a production application.

Here is an example of how you would invoke PROC SCAPROC to create a grid-enabled version of your program:

```
proc scaproc; record 'out.txt' grid 'gridout.txt'; run;

data a;
     input x y z @@; cards;
1 2 3 4 5 6 7 8 9
run;

proc summary data=a;
     var x;
     output out=new1 mean=mx;
     run;
```

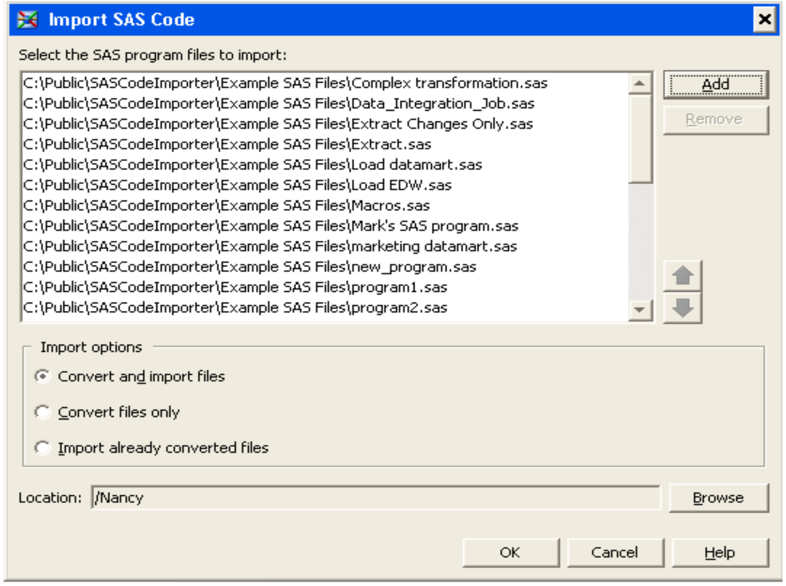
```
proc summary data=a;
     var y;
     output out=new2 mean=my;
     run;
proc summary data=a;
     var z;
     output out=new3 mean=mz;
     run;

proc scaproc; write; run;
```

The PROC SCAPROC step is inserted in front of the SAS code to be analyzed. This execution of PROC SCAPROC triggers the core components of the SAS System to provide information for activities such as opening and closing SAS data sets. The GRID statement of the SCAPROC step indicates which file to use in recording information.

After all the PROC SUMMARY steps have reached completion, PROC SCAPROC is re-invoked, indicating that it is time to analyze the information gathered from the previous steps. When this final SCAPROC step (with the WRITE statement) is seen, the grid.txt file will contain all of the necessary code to use SAS Grid Manager to execute portions of the SAS code on different nodes of the grid. In this example, the three separate PROC SUMMARY steps could be executed simultaneously on different nodes because none of the steps depends on output that is produced by any of the other PROC SUMMARY steps.

IMPORTING SAS PROGRAMS INTO SAS DATA INTEGRATION STUDIO

SAS Data Integration Studio uses the output from PROC SCAPROC to import existing SAS programs and generate jobs that include the steps in the programs. This gives you the ability to bring all sorts of existing programs that are currently in unmanaged code into a managed environment. For example, you can import a SAS program, then run that imported program in SAS Data Integration Studio, and view the performance characteristics of the program showing how each step in the program is performing and where the bottlenecks are. Each step in the program is visualized, and the code is enhanced with headers describing the inputs and outputs of each step, to improve its readability. Any tables and libraries in the programs are registered for you or connected to if the tables or libraries already exist in metadata. Mappings between steps are also created in the imported job flow.



Import SAS Code

Select the SAS program files to import:

C:\Public\SASCodeImporter\Example SAS Files\Complex transformation.sas
C:\Public\SASCodeImporter\Example SAS Files\Data_Integration_Job.sas
C:\Public\SASCodeImporter\Example SAS Files\Extract Changes Only.sas
C:\Public\SASCodeImporter\Example SAS Files\Extract.sas
C:\Public\SASCodeImporter\Example SAS Files\Load datamart.sas
C:\Public\SASCodeImporter\Example SAS Files\Load EDW.sas
C:\Public\SASCodeImporter\Example SAS Files\Macros.sas
C:\Public\SASCodeImporter\Example SAS Files\Mark's SAS program.sas
C:\Public\SASCodeImporter\Example SAS Files\marketing datamart.sas
C:\Public\SASCodeImporter\Example SAS Files\new_program.sas
C:\Public\SASCodeImporter\Example SAS Files\program1.sas
C:\Public\SASCodeImporter\Example SAS Files\program2.sas

Import options

● Convert and import files

○ Convert files only

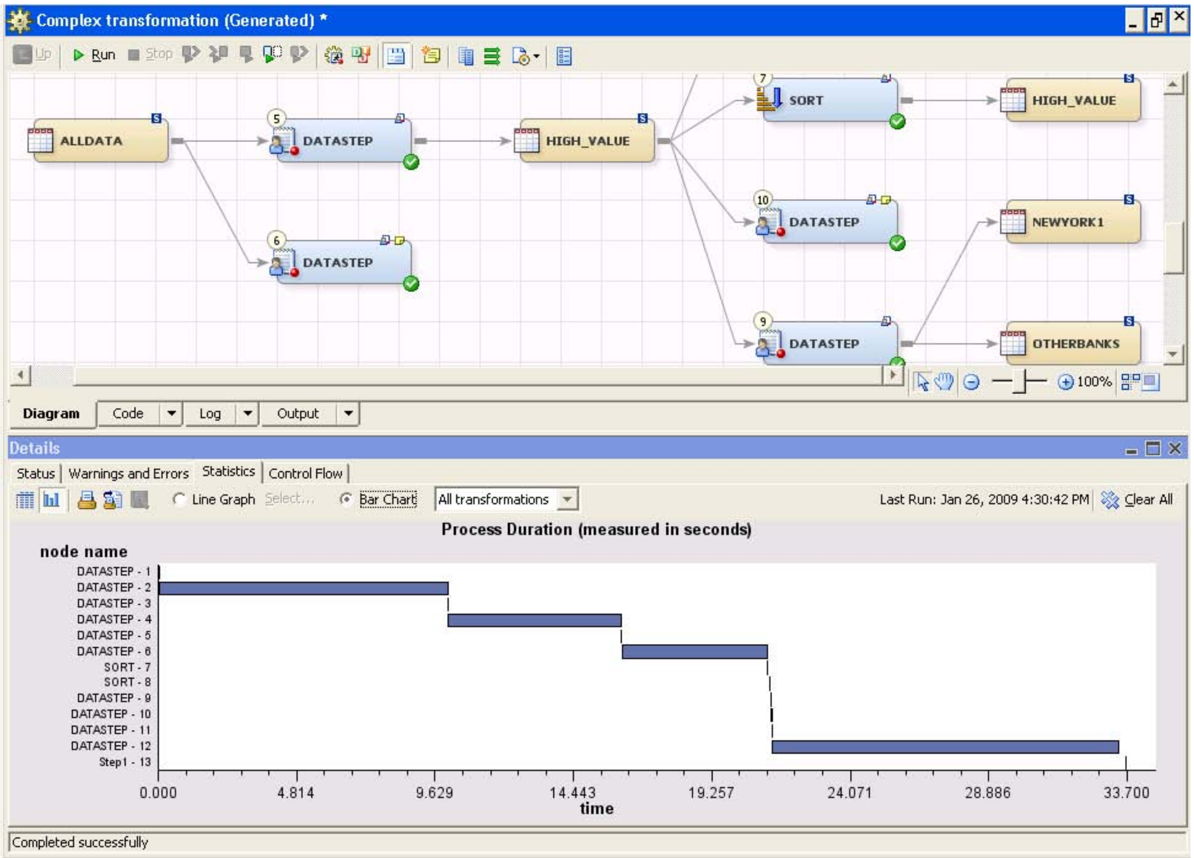
○ Import already converted files

Location: /Nancy

Figure 10: Using the SAS Code Analyzer, Any Existing SAS Program Can Be Imported into SAS Data Integration Studio.

SAS Data Integration Studio reads the output of PROC SCAPROC and creates jobs, tables, and libraries in the SAS metadata server from the output. You can also add additional syntax in your SAS program that provides hints to the SAS Data Integration Studio importer that will add notes and naming conventions to the steps in your jobs. Other options available include the ability to rerun the import and view the output iteratively.

Once the job is imported, you can use all of the SAS Data Integration Studio features including debugging and the ability to view performance metrics when the job is run. You can also submit this newly imported job to the grid by scheduling the job, using the loop/loop end transforms for parallel execution of the job, or submitting the entire job for multiuser workload balancing as described above.



Last Run: Jan 26, 2009 4:30:42 PM

Completed successfully

Figure 11: This Job Was Imported Using the Code Importer. Each Step in the Original Program Is Represented in the Job.

PERFORMANCE CONSIDERATIONS

It is useful to understand how a grid and parallelization can improve job-time performance. Taking the census example, some states have a larger number of households than other states. The performance of each job element therefore varies by state. We took a baseline test and benchmarked the runtimes for each state during sequential execution to come up with the following performance workload running in a test environment of 10 blade servers with 60GB of data.

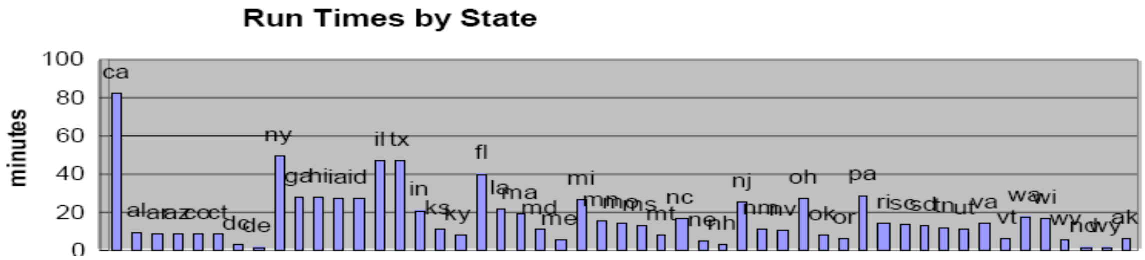
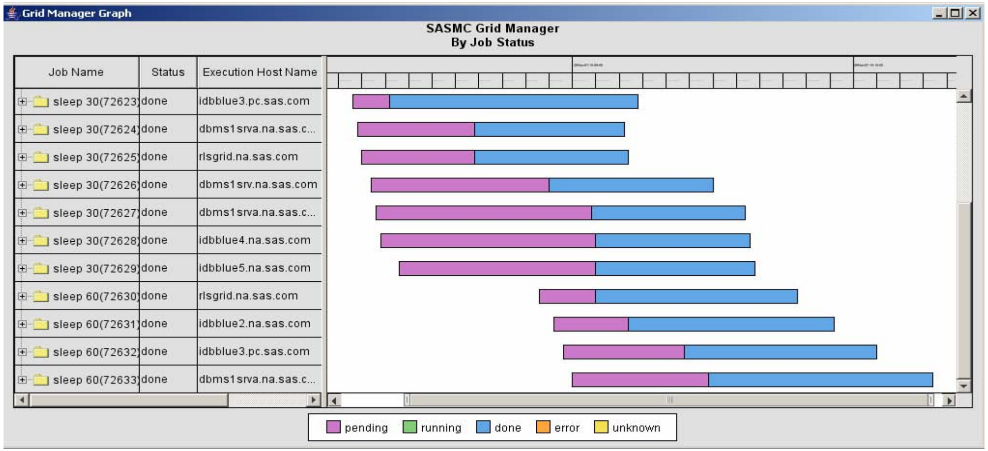


Figure 12. Runtimes for Processing Census Data by State.

Running these jobs serially took around 600 minutes of real time on our test environment. We then ran the same jobs in parallel using the Loop transform. We were able to achieve a best-case runtime in the test environment of approximately 100 minutes, a 6x performance gain over the serial case.

SAS has published a number of grid performance benchmarks for different scenarios; see the reference list at the end of this paper for details. Benchmark tests are available for a variety of customer-usage patterns, including a large, multiuser, ad hoc analytics environment, and an I/O-intensive scenario. These scenarios varied the number of available grid nodes and I/O capabilities to determine performance patterns with some encouraging results. In the computational scenario, the benchmarks were able to achieve linear scalability as workload increased simply by increasing the available computer resources; that is, adding nodes into the grid. Similarly with I/O-intensive processes, increasing the available I/O resources enabled the processes to scale linearly in the grid environment.

When using the grid for testing, we found it useful to visualize the workload on the grid to identify when processes were running and when they had completed. This is possible using the grid monitoring capabilities of the SAS Grid Manager plug-in available in SAS Management Console.



Job Name	Status	Execution Host Name
sleep 30(72623	done	idbblue3.pc.sas.com
sleep 30(72624	done	dbms1srva.na.sas.c...
sleep 30(72625	done	rlsgrid.na.sas.com
sleep 30(72626	done	dbms1srv.na.sas.com
sleep 30(72627	done	dbms1srva.na.sas.c...
sleep 30(72628	done	idbblue4.na.sas.com
sleep 30(72629	done	idbblue5.na.sas.com
sleep 60(72630	done	rlsgrid.na.sas.com
sleep 60(72631	done	idbblue2.na.sas.com
sleep 60(72632	done	idbblue3.pc.sas.com
sleep 60(72633	done	dbms1srva.na.sas.c...

Figure 13. Grid Monitoring Capabilities Available in SAS Grid Manager.

Other SAS Grid Manager features enable you to see the status of each job process on the grid, identify which nodes are in use and which nodes are available, withdraw a grid node from the set of available nodes, and perform other useful operations. You can tune your grid environment to achieve maximum efficiency.

TEXT MINING SCENARIO

Parsing unstructured data is one way you can readily leverage grid computing in the data integration environment. In some organizations, unstructured data, mostly in the form of text files, can account for a significant amount of useful information in an organization's knowledge stores. This information is difficult to access because it is not easily synthesized into tables that can be merged into existing processes, and the volume of information can require a significant amount of time to process. Grid processing, with its ability to manage workloads and bring many resources to bear, is ideally suited to handling this type of data.

SAS® Text Miner includes capabilities to assist in processing unstructured text data. You can wrapper the text-mining calls into SAS Data Integration Studio transformations to process unstructured data coming from text files. For example, suppose that you want to process text data coming from a Web page, and merge that data with other structured data in your data warehouse. Follow this process:

1. Generate a wordlist from a Web site.
2. Process the word list.
3. Merge the structured and unstructured data.

In SAS Data Integration Studio, we can create transforms that call SAS Text Miner to process the data on the Web pages.

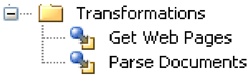
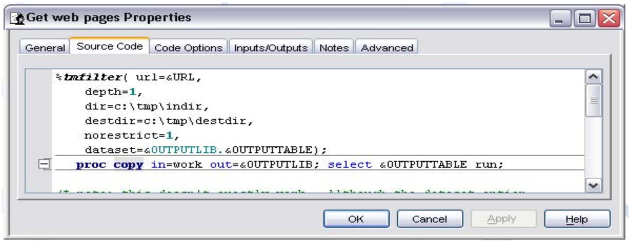


Figure 14. Transforms in SAS Data Integration Studio for Transforming Unstructured Text Data.

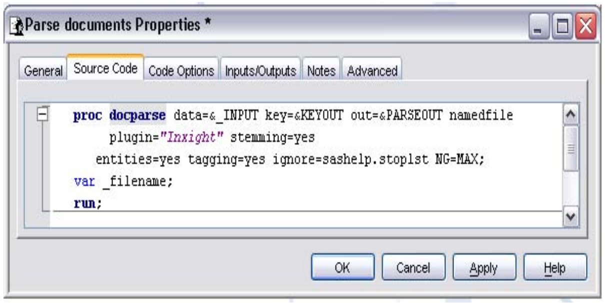
The Get Web pages transform uses the text miner %TMFILTER macro.



```
%tmfilter( url=&URL,
    depth=1,
    dir=c:\tmp\indir,
    destdir=c:\tmp\destdir,
    norestrict=1,
    dataset=&OUTPUTLIB.&OUTPUTTABLE);
  proc copy in=work out=&OUTPUTLIB; select &OUTPUTTABLE run;
```

Figure 15. Source Code Example from the Get Web Pages Transform in SAS Data Integration Studio.

This transform reads Web pages starting from a user-specified URL and creates a data set that can be used as input. Documents of different formats can also be read using this same technique. Then we parse the documents; this creates word lists by type and frequency. Parts of speech can also be identified during the parsing stage.



```
proc docparse data=&_INPUT key=&KEYOUT out=&PARSEOUT namedfile
     plugin="Inxight" stemming=yes
   entities=yes tagging=yes ignore=sashelp.stoplst NG=MAX;
var _filename;
run;
```

Figure 16. Source Code Example from the Parse Documents Transform in SAS Data Integration Studio.

Using these two transforms, we can read Web pages and parse them into two data sets, a list of keywords found, and the raw-word data.

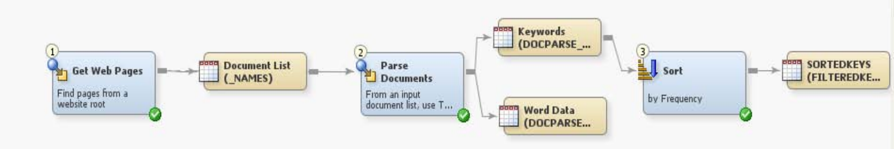
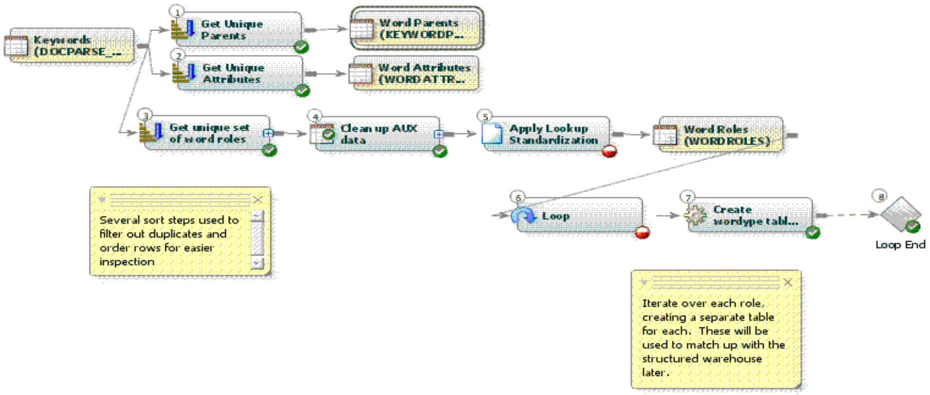


Figure 17. Sample SAS Data Integration Studio Job to Parse Text Data.

Using these two data sets, we can further process the data to pull out standard roles from the raw list of words, apply a standardization rule to place the words into known sets, and match the words to the roles so that we have a set of data that is structured to match up to additional warehouse data.

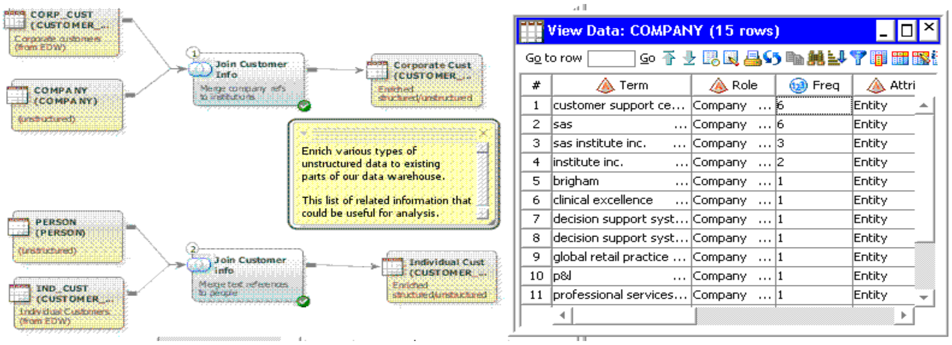


Several sort steps used to filter out duplicates and order rows for easier inspection

Iterate over each role, creating a separate table for each. These will be used to match up with the structured warehouse later.

Figure 18. Transforming Unstructured Textual Data To Enrich Existing Warehouse Data.

The output of this process is a set of tables that contain data that is structured to enrich the existing warehouse data with information pulled from the text files.


CORP_CUST (CUSTOMER_...
Corporate customers (from EDW)
COMPANY (COMPANY)
(unstructured)
1
Join Customer Info
Merge company refs to institutions
Corporate Cust (CUSTOMER_...
Enriched structured/unstructured
PERSON (PERSON)
(unstructured)
IND_CUST (CUSTOMER_...
Individual Customers (from EDW)
2
Join Customer info
Merge text references to people
Individual Cust (CUSTOMER_...
Enriched structured/unstructured
View Data: COMPANY (15 rows)
Go to row
Go

Enrich various types of unstructured data to existing parts of our data warehouse.

This list of related information that could be useful for analysis.

#	Term	Role	Freq	Attri
1	customer support ce...	Company ...	6	Entity
2	sas ...	Company ...	6	Entity
3	sas institute inc. ...	Company ...	3	Entity
4	institute inc. ...	Company ...	2	Entity
5	brigham ...	Company ...	1	Entity
6	clinical excellence ...	Company ...	1	Entity
7	decision support syst...	Company ...	1	Entity
8	decision support syst...	Company ...	1	Entity
9	global retail practice ...	Company ...	1	Entity
10	p&l ...	Company ...	1	Entity
11	professional services...	Company ...	1	Entity

Figure 19. Final Result of the SAS Data Integration Studio Job Showing Transformed Text Data Ready to Combine with the Existing Data Warehouse.

This processing can be performance-intensive because Web logs and word counts can be large. Using the SAS Data Integration Studio Loop transform to perform this processing in parallel on an SMP machine or on a Grid can significantly speed up the process.

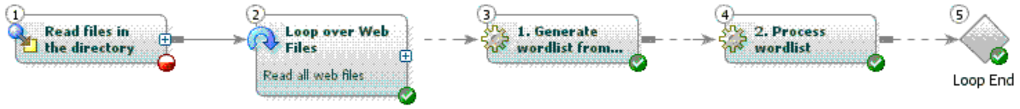


Figure 20. Performing the Text Transforms in Parallel Is Easy Using the Loop transform in SAS Data Integration Studio.

PERFORMANCE MONITORING USING JMP®

When monitoring the performance of grid nodes, you could find that you have to tune the environment to gain the best performance. Many variables can affect performance; for example, the number of machines in the grid, the amount of memory or cache per node, I/O capabilities, and the current workload. It is therefore easier to visualize the system performance rather than see individual node information or tables of performance numbers. JMP® is the ideal visualization tool for visualizing grid node performance, because JMP dynamically links statistics with graphics.

Using the census data sample scenario discussed earlier, JMP can help you see how various aspects of grid performance are affected by the range of tuning parameters that are available.

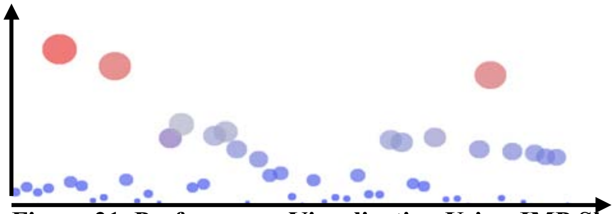


Figure 21. Performance Visualization Using JMP Shows That the Grid Performance Is Not Yet Optimized.

Figure 21 is a JMP bubble plot that shows an untuned grid deployment using the sample census data. The X axis represents the state being processed, and the Y axis shows the duration of time that is required for the state. We can also use bubble size as an indicator for a value such as memory use, and use bubble color to represent another value such as I/O required to process that state. Looking at a collection of values makes tuning easier because you can identify trade-offs. What we want is the smallest circles, lowest on the Y axis, and with the desired color. In the plot above, the largest circles belong to California, Florida, and Texas because data from their larger populations requires more processing time.

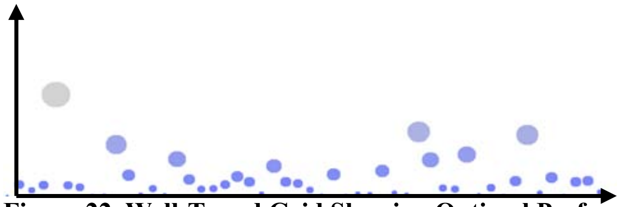


Figure 22. Well-Tuned Grid Showing Optimal Performance Visualization.

As tuning progresses, using the JMP software we can see as the network approaches the optimal configuration. Completed tuning results in the desired visual outcome: small circles (minimal memory used) that are lowest on the Y axis (smallest time requirement) for all states.

CONCLUSION

SAS parallelization capabilities have been continually enhanced with each release of SAS software. The introduction of SAS Grid Manager in SAS® 9.1.3 marked a significant milestone in performance computing support in the SAS System. SAS Data Integration Studio makes it easy to leverage parallel and grid computing through customized transforms that enable you to parallelize your processes easily and efficiently. Data Integration Studio also supports all aspects of SAS Grid Computing so that you can select the technique that best matches your computing needs. The introduction of the SAS Code Analyzer in 9.2 helps you parallellize your processes and import your existing SAS code into a managed environment. Grid computing lends itself to those scenarios that require a significant amount of computational and I/O resources. The processing of unstructured data using SAS Data Integration Studio is an excellent example of one scenario that can benefit from the performance optimizations provided by SAS Grid processing. For performance tuning, SAS JMP software can help visualize the many variables required to optimize grid performance.

REFERENCES

Hunley, Eric, and Nancy Rausch. 2009. “What’s New in SAS® Data Integration Studio.” *Proceedings of the SAS Global Forum 2009 Conference*. Cary, NC: SAS Institute Inc. Available at http://www2.sas.com/proceedings/forum2009.

Hunley, Eric, Gary Mehler, and Nancy Rausch. 2008. “A Whole New World: What’s New in SAS® Data Integration Studio 4.2.” *Proceedings of the SAS Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc. Available at http://www2.sas.com/proceedings/forum2008.

SAS Institute Inc. 2007. *SAS® Data Integration Studio 3.4 User's Guide*." Cary, NC: SAS Institute Inc. Available at http://support.sas.com/documentation/onlinedoc/etls/index.html

Hewlett Packard. 2007. "Benchmark Report for SAS® Grid Manager Running Linux on HP ProLiant BL460c Servers and HP StorageWorks." Available at http://h20219.www2.hp.com/ERC/downloads/4AA1-6058ENW.pdf

SAS Institute Inc. 2007. "SAS Demonstrates Superior Scalability with SAS® Grid Manager and Sun StorageTek QFS Shared File System." Cary, NC: SAS Institute Inc.
Available at http://www.sas.com/partners/directory/sun/SASGridMgr0907.pdf

SAS Institute Inc. 2006. "SAS® Enterprise Data Integration Server Building a 4.2 Terabyte Star Schema Using SAS Grid Technology." Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2007. "Introduction to Grid Computing. Scalability and Performance Community." Cary, NC: SAS Institute Inc. Available at support.sas.com/rnd/scalability/grid

SAS Institute Inc. 2007. "SAS Grid Computing. Technologies/Enterprise Intelligence Platform." Cary, NC: SAS Institute Inc. Available at http://www.sas.com/technologies/architecture/grid/index.html.

CONTACT INFORMATION

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

Cheryl Doninger
SAS Institute Inc.
Cary, NC 27513
Work Phone: (919) 677-8000
Fax: (919) 677-4444
E-mail: Cheryl.Doninger@sas.com
Web: www.sas.com

Nancy Rausch
SAS Institute Inc.
Cary, NC 27513
Work Phone: (919) 677-8000
Fax: (919) 677-4444
E-mail: Nancy.Rausch@sas.com
Web: www.sas.com