Making Everything Easier!™

2nd Edition

SAS® Grid Computing For Dummies®

A Wiley Brand

Learn to:

- **Balance computing resources to meet demands**
- **Benefit from a highly available computing environment**
- **Accelerate decision making by processing data faster**

Cheryl Doninger
Steven Sober

SAS® Grid Computing For Dummies®

A Wiley Brand

Second Edition

by Cheryl Doninger
Steven Sober

For Dummies®
A Wiley Brand

SAS® Grid Computing For Dummies®, Second Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-119-25047-0 (pbk); ISBN 978-1-119-25049-4 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Publisher's Acknowledgments

For general information on our other products and services, or how to create a custom *For Dummies* book for your business or organization, please contact our Business Development Department in the U.S. at 877-409-4177, contact info@dummies.biz, or visit www.wiley.com/go/custompub. Some of the people who helped bring this book to market include the following:

Project Manager and Development Editor: Chad R. Sievers

Acquisitions Editor: Amy Fandrei

Editorial Manager: Rev Mengle

Business Development Representative: Karen Hattan

Custom Publishing Project Specialist: Michael Sullivan

Dummies Marketing: Lauren Noens

Production Editor: Shaik Siddique

Contents at a Glance

Table of Contents

Introduction

If your organization is like many out there, you're likely trying to do more every day. Your organization has service level agreements (SLAs) it needs to meet. You can't afford any setbacks, and you don't have hours to wait for complex computations to be executed. And if the system fails, you certainly can't afford to restart your processes.

The fast pace of business is par for the course, but if you're faced with a computing infrastructure that's quickly becoming obsolete and is consistently overloaded, you've got problems.

About This Book

SAS Grid Computing For Dummies, 2nd Edition, tells you all about SAS Grid Computing and its many benefits to organizations such as yours. By implementing a SAS Grid environment, your organization can better balance and manage the workloads, use its resources to its full scope and capabilities, and increase productivity and overall performance.

SAS Grid Computing allows you to perform more complex analysis in less time — and prioritize your most important work to match the requirements of your business. Using parallel processing and enterprise scheduling (we explain those in this book), SAS Grid Computing saves time and money so you can focus on your organization's business and not get bogged down in its IT needs.

Foolish Assumptions

In writing this book, we are going to make some assumptions about you. We assume the following:

- You're familiar with SAS Analytics and either use it or know how it's used in your organization.

- You understand that SAS users can be a mixture of people. For example, some are interactive and need immediate results, some are scheduling reports for specific times, and some are computer and resource demanding.
- You're either in technical management or line-of-business management with a title such as a Chief Information Officer, Director of IT, Operations Executive, and so on.
- You're interested in making the most of using SAS Analytics and want to know how SAS Grid Computing can help.

Icons Used in This Book

As you read through this book, you may notice these three icons, drawing your eye to certain pieces of information. Here's what they mean:

TIP

This icon indicates that the information is interesting or possibly useful, but not absolutely essential to know.

REMEMBER

When you see this icon, reading its neighboring paragraph is a good idea; it points out important information.

TECHNICAL STUFF

This icon indicates information that's a bit more specialized. Read it if you like, or skip it if you're short on time.

Where to Go from Here

Like all *For Dummies* books, you can read sequentially from cover to cover, or you can jump in, out, and around, depending on your preference. In an attempt to avoid repetition, some parts of the book reference other parts where there may be more detail on a particular area.

If this book persuades you that SAS Grid Computing is something that you're interested in and you want to know more, feel free to visit `www.sas.com/grid`.

Chapter 1

Increasing Performance with Analytics

In This Chapter

- Obtaining the right information, right away
- Exploring high-performance analytics
- Seeing how SAS Grid Computing works
- Taking a complementary approach to computing

Today's top organizations are looking for the best, most efficient ways to extract meaning from all the data they collect. They need to pull out what's relevant and find the underlying patterns, relationships, and ideas that will help them to improve their business. And with the speed at which the world operates today, this needs to happen quickly — almost in real time.

In this chapter, we explore the advantages of high-performance analytics and introduce SAS Grid Computing to show you how this tool can improve your organization's performance.

Getting Meaningful Data, Faster

Making meaning from massive amounts of data is not a simple process. It can take hours of complex computations. To speed things up, organizations must restrict the size of their analytical models and data sets accordingly.

A bank is a good example of an organization that needs to make sense of lots of data as soon as possible. Banks compute hundreds of financial models every year. To get the most accurate financial forecasting and information, they often need to change these models monthly, or even weekly. That's a lot of time and work. (Chapter 7 gives you an example of how SAS Grid Computing streamlined that process for a bank.)

But imagine: What if time weren't an issue, and models could be run hundreds of times faster? Then banks (for example) could run 10 to 20 times more models in the same amount of time. Well, it is possible — with *high-performance analytics (HPA)*.

Discovering HPA

With high-performance analytics (HPA), computations happen much faster — which speeds up the process of developing analytical models. The upshot is that — you guessed it — the organization gets the most salient information much faster.

The speed of HPA comes from using the power of hundreds, even thousands, of processors working in parallel.

HPA allows organizations to do some complex tasks more efficiently:

- Ask difficult questions, test multiple scenarios, and solve problems that could not be solved before
- Generate meaningful information quickly to make immediate decisions based on accurate results, even when an issue is complex
- Get highly accurate insights using more variables, more complex analytical methods, and more models than before

The advantage of using multiple processors is that a lengthy job gets done quickly, and complicated issues get solved as simply as possible. HPA allows organizations to get answers to complex questions — think billions of records with thousands of variables — in a fraction of the time needed by traditional computing environments. HPA can also perform more simulations and highly detailed modeling across all of an organization's data. With HPA, the limitation of working with only sample data sets is gone.

Approaching the Architecture

SAS applications that run HPA on big data sets have three complementary architectural approaches: SAS Grid Computing, SAS In-Memory Analytics, and SAS In-Database Analytics. The next sections explore these approaches.

SAS Grid Computing

SAS Grid Computing, as a shared, general-purpose computing environment for lots of SAS users running lots of different SAS applications, provides robust management capability. SAS Grid Manager does that job. You can greatly reduce the processing time for individual SAS jobs because jobs are load-balanced across multiple resources and can even be split up. Each piece of the job can be run in parallel, using several nodes across the grid that share physical storage.

TIP

SAS Grid Computing is an excellent tool for handling prep work, performing ad hoc data analysis, and doing production work. But you can also use it to improve performance across your entire workload.

SAS In-Memory Analytics

The key to the superior performance of SAS In-Memory Analytics is that it doesn't have to use disk-based processing across many SAS applications (including SAS High-Performance Analytics) to do its job. SAS In-Memory Analytics can run in either SMP or MPP mode, so it can discover patterns and build predictive models using big data instead of sample sets.

By utilizing SAS In-Memory Analytics, an organization can quickly find new opportunities to upsell or cross-sell products, by identifying small — but promising — customer segments.

REMEMBER

SAS Visual Analytics is a SAS in-memory product. It combines an easy-to-use interface with in-memory technology so users at various levels of sophistication can see how their analytical models work.

SAS In-Database Analytics

SAS In-Database Analytics uses a *massive parallel processing (MPP)* distributed architecture. It speeds up three essential processes: managing key data, developing analytics, and deploying tasks. Because the relevant tasks are moved closer to the data and the computations run inside the database, this approach saves the time that would be spent moving and converting data.

Working in a Complementary Way

Although some say three's a crowd, that's not true with the three SAS high-performance approaches — SAS Grid Computing, SAS In-Memory Analytics, and SAS In-Database Analytics — discussed in this chapter. One isn't better than another; they're complementary. All give organizations a better, faster option for creating and running SAS analytics to get meaningful results. Consider the following scenarios:

- If an organization needs to run production jobs, do ad hoc analytics, and give its analysts a shared platform, SAS Grid Computing is the right fit.
- If a business has millions of rows of observations in its data set and wants to drill down visually into different scenarios in real time, SAS In-Memory Analytics is a good choice.
- If a business needs more speed and processing power that uses its existing distributed storage system, the answer is SAS In-Database Analytics.

These are only three possible scenarios out of the many a business might encounter every day. Your own business workload and requirements will help determine which solutions are the best fit and how they'll be used.

Chapter 2

Eyeing the Features and Benefits of SAS Grid Computing

In This Chapter

- Understanding how grid computing works
- Checking out features of SAS Grid Manager
- Solving processing problems
- Assessing the benefits of grid computing
- Seeing how the system works
- Naming the parts of the grid
- Knowing what applications work

They say that many hands make light work. Well that's certainly the case with SAS Grid Computing. A *grid computing* environment distributes computing tasks across many computers on the same network; the computers doing the work form a grid — which SAS Grid Manager controls. Each computer within the grid is known as a *grid node.* All these computers share the workload, getting it done faster and more efficiently.

This chapter takes you through the features and benefits of a SAS Grid Computing environment, and introduces you to how this type of environment works.

Recognizing the Features

This grid computing environment and SAS Grid Manager have many features that offer advantages to organizations such as yours. The following sections take you through these features.

Managing the workload

SAS Grid Manager provides smarter workload management in two ways: by balancing the workload and making scheduling more efficient. (Chapter 3 goes into more detail about workload balancing and scheduling.)

Workload balancing

Nobody likes conflict, and SAS Grid Manager helps you to avoid that hassle with a feature called *workload balancing*. With multiple computers working at the same time and sharing tasks, several users in the same SAS environment can allocate jobs and SAS Grid Manager will balance the workload.

By spreading user jobs across the available resources — sharing the load — you don't experience resource conflicts, no matter which SAS products you deploy.

TIP

Workload balancing is critical when you have multiple users running multiple SAS applications.

Because SAS Grid Manager provides a centrally administered environment, it works with policies that identify high-priority users and/or jobs, and also designates which jobs get what portion of the computing resources. This helps you avoid a sudden onslaught of work requests that are all attempting to happen at the same time (which can slow down the system or even crash the server).

TIP

SAS Grid Manager prioritizes work by *mapping* the SAS work requests to the available resources. It can also prioritize the work, so that after high-priority work is executed, the remaining jobs get done as soon as computing resources are available. High-priority jobs can even pre-empt low-priority work, so that the most critical business processes are always executed first.

Enterprise scheduling

The ability to manage a complex flow of workloads in a robust and scalable way is called *enterprise scheduling*, and every enterprise-class organization must be able to do it. Using the Schedule Manager plug-in (see "Knowing What's Needed to Run SAS Grid Manager," later in this chapter) with the SAS Management Console, users can create SAS workflows and schedule them according to time of day, file events, and/or the completion code (success, warning, error, informational, or fatal) of any prior SAS job within the same workflow. Then, multiple workflows — as well as the jobs within a single workflow — are distributed to the SAS Grid environment.

A variety of SAS applications and solutions can create SAS jobs, or SAS programmers can write jobs. Many of the products and solutions — including SAS Data Integration Studio, SAS Marketing Automation, SAS Marketing Optimization, and SAS Web Report Studio — incorporate these scheduling capabilities. Multiple jobs, whether created by SAS products or user-written with SAS programs, can make for simple or complex workflows, which the Schedule Manager will understand and then schedule to the grid environment.

Enterprise scheduling gives your organization the structure needed for scheduling all production jobs, regardless of the complexity of the flows.

Performing distributed parallel processing

SAS Grid Manager supports *parallel processing*, the main benefit of which is that the whole application works efficiently and the job gets done much faster. SAS Grid Manager ensures that the parallel workloads are distributed evenly to keep the SAS programs and applications performing at their best.

Managing workloads in parallel is especially important for SAS applications that consist of independent *subtasks* — units of work that can be distributed across the grid and done at the same time.

Using distributed parallel processing

To allow a particular SAS program to use distributed parallel processing, the user adds RSUBMIT and ENDRSUBMIT statements around each subtask, and also adds the GRDSVC_ENABLE function call. SAS Grid Manager automatically assigns each identified subtask to a grid node. You can also use the SAS Code Analyzer to automatically create a grid-enabled SAS job. To use SAS Code Analyzer, add PROC SCAPROC statements to your SAS program, specifying the GRID parameter. When you run the program with the PROC SCAPROC statements, the grid-enabled job is saved to a file. You can then run the saved SAS job on the grid, and SAS Grid Manager automatically assigns the identified subtasks to the available grid nodes.

Several SAS products can generate code that recognizes opportunities for parallel processing. They generate the appropriate code in preparation to submit the flow to SAS Grid Manager. Then the appropriate jobs are executed in parallel across the grid. Products with the feature to parallelize include SAS Data Integration Studio and SAS Enterprise Miner. Additionally, programmers can modify existing SAS programs to identify the individual subtasks to run in parallel.

Offering a highly available analytical environment

When your computer resources are organized into a grid architecture, your company gets the benefits of a highly available analytical environment. It can perform much more work than the same resources could do when working independently. When a computer fails, the work stops on that machine. SAS Grid Manager provides the ability to detect when a system is down and to reschedule that work on another available resource. All without you having to worry about it!

With SAS Grid Manager, you can configure the services within your SAS Grid environment to be highly available. (See Chapter 4 for more detail about highly available grid environments and their benefits to your organization.)

In a nutshell, SAS Grid Manager does the following:

- Monitors the critical services
- Detects any failure of these critical services or the machine on which they're running
- Starts the services on a failover host or standby host automatically

When a component in a client/server system fails, a hardware load balancer or a DNS-name resolution program redirects clients to the service running on the failover host. This arrangement ensures that critical services remain available to clients without manual intervention.

Solving Problems with SAS Grid Manager

By running in a distributed, grid computing environment, SAS Grid Manager addresses and solves the challenges that fall into two general categories: business challenges and IT challenges.

Identifying business challenges

SAS Grid Manager addresses three main business challenges:

- **Increasing user and data growth:** As the number of users and the amount of data continues to increase, existing servers can get overloaded, which leads to sluggish processing times — not acceptable for any organization with deadlines to meet. You could change to a large-capacity server, but that costs money and interrupts services.
- **Running larger and more complex analysis:** You want to increase the complexity of the analysis to get more accurate or detailed results. But the server doesn't have the processing capacity so the SAS jobs that can be run are limited, or maybe getting the results you need takes too long.
- **Prioritizing user workloads:** Your organization has to prioritize the most important jobs to match business

requirements. But if you need to do more analyses than the existing infrastructure can support, business processes won't be accomplished in an acceptable amount of time.

Identifying IT challenges

To be a top performer, your organization has to meet some serious IT needs. Two such needs are the deployment of a highly resilient and high-performing infrastructure and the effective prioritization of work. The good news is that SAS Grid Manager can help — by addressing the following three main IT challenges. Perhaps you recognize your organization in these scenarios:

- **Needing to quickly adapt to changing workloads:** Your organization can be dynamic at times and so you need the ability to change priorities and rearrange the workloads. You require the flexibility to quickly change the allocation of resources to meet peak or changing workloads.
- **Demanding higher resiliency:** Faced with growing data volumes, more complex analytics, and tighter timeframes, IT can't afford any downtime. Some servers and long-running SAS programs are critical to your organization's operations, and those services must be available at all times — even if the servers that run them become unavailable. The SAS programs must complete their work in a timely manner, even if something causes them to fail.
- **Requiring a flexible IT infrastructure refresh:** Your existing server infrastructure has reached its capacity, is quickly becoming obsolete, and clearly won't be able to handle future computing requirements. That infrastructure needs to be replaced — and IT is looking for a flexible, cost-effective computing environment that can grow incrementally over time, keeping pace with the organization's needs. You must also be able to add needed resources quickly as the workload increases — and handle maintenance issues (such as adding or replacing resources) without disrupting your work.

Keying into the Benefits

Having a scalable, efficient, and flexible computing environment gives your organization a competitive edge. Here's where you get a look at how to get there by using SAS Grid Manager.

Using resources optimally

SAS Grid Manager provides a central point for administering policies. It allows you to prioritize and balance workloads, monitor jobs, schedule tasks, and suspend noncritical work. This enables you to help users and groups realize their business goals across the organization.

Maintaining business continuity

With multiple server and storage nodes, SAS Grid Manager eliminates single points of failure so the infrastructure is highly available — even during maintenance windows.

Fortunately, if a particular SAS program takes a long time to run, you can define checkpoints from which to restart the program if it's interrupted.

Improving productivity

Better productivity means a better bottom line for the organization. With SAS Grid Manager, individual SAS jobs get done faster when they're analyzed and distributed (as appropriate) to run simultaneously on multiple grid nodes. Result: You get quality results in a timely way.

In addition, more work gets done faster because workloads are managed in a smarter way. SAS Grid Manager ensures that jobs automatically execute on the best available computing resources, so valuable resources are being used to their full capability.

REMEMBER

SAS Grid Manager lets your organization focus on meeting its business goals, rather than wasting too much time, money, and effort dealing constantly with infrastructure and processes.

Kudos from a customer

SAS Grid Computing gives you the flexibility to rightsize your IT responses to emerging business opportunities. One SAS customer put it quite well when he said, "SAS Grid Computing gives us the ability to upgrade in increments of tens of thousands instead of our previous increments of millions of dollars."

Increasing flexibility

SAS Grid Manager gives increased flexibility to your organization in so many ways. It can

- Grow over time to meet increases in the number of users and the amount of data processed, adding servers to the grid as needed
- Leverage resources to reduce costs
- Run multiple applications simultaneously
- Allocate computing resources according to the demands of users and new business units
- Prioritize work to meet peak demands more efficiently

REMEMBER

SAS Grid Computing is so flexible that it can be implemented on a variety of operating systems and architectures.

Looking at SAS Grid Architecture

SAS Grid Computing architecture is a system of interconnected *nodes* — which are components that do the distributed work. The grid consists of the following main components, which Figure 2-1 illustrates:

- **Grid control server:** This node controls how jobs are distributed to the grid. Any machine in the grid can be designated as the grid control server.

- **Grid node:** These machines receive the work that is distributed to the grid. The number of grid nodes depends on the size, complexity, and volume of jobs that are run on the grid.
- **Central file server:** This machine stores data for jobs that run on the grid. It can be a single *storage node* (the grid's designated place to store data) or include multiple tiers of storage, which are typically managed by a shared, clustered file system.
- **Metadata server:** This machine stores the metadata definitions needed by SAS Grid Manager and other SAS applications on the grid.
- **Grid clients:** Grid clients submit jobs to the grid for processing, they don't actually execute the work. In effect, they pass the buck to the Grid nodes.

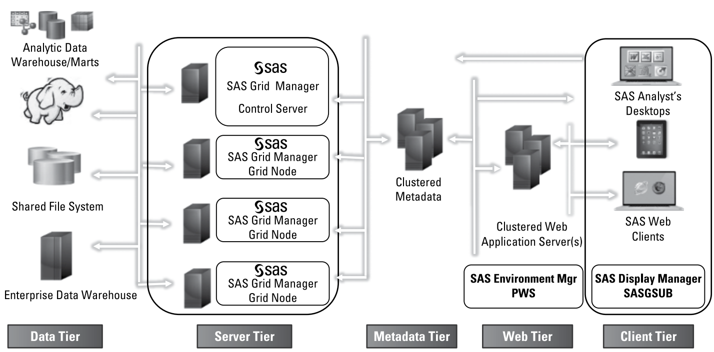

Figure 2-1: SAS Grid Computing architecture.

Knowing What's Needed to Run SAS Grid Manager

When SAS Grid Manager is deployed it uses the following components:

- **Grid enablement code:** This is the SAS code that directs the execution of the job to the grid.
- **Platform Suite for SAS:** This is a suite of components that is included with SAS Grid Manager and allocates the resources efficiently, manages policies, and balances the

allocation of SAS workload requests. For more details on the components that make up the Platform Suite for SAS, refer to the online SAS Grid Computing document on `support.sas.com/documentation/onlinedoc/gridmgr/index.html`.

- **SAS Environment Manager:** This is a web-based tool that shows users the status of the devices and the services running in the SAS Grid environment. It does this by using visual displays. It can also allow you to manage the policies and configuration of the grid.

Seeing the SAS Applications That Support Grid Processing

All SAS programs can participate in SAS Grid Computing. To increase performance of given SAS jobs, independent units of work (subtasks) can run in parallel. Figure 2-2 is a list of SAS applications that support grid processing, along with the type of processing that each supports.

SAS Application or Product	Scheduling	Workload Balancing	Parallel Workloads
SAS programs	✓	✓	✓
SAS Data Integration Studio	✓	✓	✓
SAS Web Report Studio	✓		
SAS Marketing Automation	✓		
SAS Marketing Optimization	✓		
SAS Enterprise Miner		✓	✓
SAS Enterprise Guide		✓	
SAS Add-in for Microsoft Office		✓	
SAS Workspace Server		✓	
SAS Stored Process Server		✓	
SAS OLAP Server		✓	
SAS Stored Processes		✓	✓
SAS Risk Dimensions			✓
JMP Genomics 4.0			✓
SAS Demand Forecasting for Retail 4.2			✓
SAS for Customer Experience Analytics	✓		

Figure 2-2: SAS applications that support grid computing.

Chapter 3

Managing the Workload

In This Chapter

- Ensuring a balanced workload
- Seeing how scheduling works
- Meeting the Process Manager
- Examining the architecture

When scheduling jobs in an enterprise environment, it's important to have a way to queue, prioritize, and dispatch the workload in an efficient manner. To be efficient, the work needs to be balanced and distributed to the right resources for each type of job. SAS Grid Manager is designed to accomplish this by providing the administrators and user with the tools and means to have the grid computing system make the best use of the available resources.

This chapter talks about how balancing the workload and submitting jobs are accomplished. This chapter also delves into how policies can be defined for managing the variety of work and discusses what happens when the workload gets heavy.

Balancing the Workload

When you have multiple users running multiple SAS applications — which is typical for most SAS enterprise-level analytic environments — managing the workload is absolutely critical.

SAS Grid Manager provides a centrally administered environment to enable policies that determine which users and/or jobs get priority and which jobs get which share of the computing resources. (Chapter 2 tells all about the SAS Grid

Manager and provides an overview of how jobs are submitted and prioritized.)

Policies prevent a huge amount of work requests from being submitted at the same time, which is essential because such a mass submission could slow down the servers or simply crash the system.

Submitting a job

The lifecycle of a grid-enabled job starts when the SAS application submits a job to SAS Grid Manager. The job is placed into a queue before being serviced by a targeted host.

Queuing up

Queues contain a series of pending jobs, lined up in a defined order, each waiting for its opportunity to use resources. At installation, the SAS Grid Manager creates a default queue called the NORMAL queue.

In a standard SAS Grid configuration, SAS jobs are automatically submitted to the default or NORMAL queue.

Queues implement different job scheduling and control policies. All jobs submitted to the same queue share the same scheduling and control policy. Queues don't correspond to individual hosts; each queue can use all server hosts in the grid — or a configured subset of the server hosts.

This flexibility enables grid administrators to configure queues and customize the grid environment. They can create multiple customized queues to provide complete administrative control over the workload submitted to the SAS Grid.

Queues can be assigned a priority to determine the order in which they are serviced. This allows an administrator to manage higher-priority work by automatically assigning it to the appropriate priority queue based on the originating user or application.

Queues can have many attributes, including the following:

- **Priority:** A larger integer indicates a higher priority.
- **Name:** This attribute uniquely identifies the queue.

- **Queue limits:** These attributes restrict hosts, number of jobs, users, groups, processors, and so on.
- **Load-sharing threshold conditions:** These attributes apply load sharing to the queue.
- **Job slots:** This attribute specifies the number of job slots that a queue can use concurrently. (See "Assigning job slots" later in this chapter.)

Queue priorities

The *queue priority* defines the order in which SAS Grid Manager searches the queues to determine which job will be executed next. The grid administrator assigns a priority to the queue — a higher number indicates a higher priority — and queues are processed in order of priority. If multiple queues have the same priority, SAS Grid Manager schedules all the jobs from these queues in first-come, first-served order.

The SAS Grid environment associates priority with queues, not jobs. For more urgent work requests to receive higher priority, they must be placed in a higher-priority queue whose jobs will be serviced ahead of lower-priority queues.

Scheduling and dispatching jobs

Submitted jobs sit in queues until they're scheduled and dispatched to a host for execution. When a job is submitted to SAS Grid Manager the following factors control when and where the job starts to run:

- Active time window of the queue or hosts
- Resource requirements of the job
- Availability of hosts
- Job dependency conditions
- Load conditions

Assigning job slots

A *job slot* is a placeholder for a single unit of work. Hosts are configured to have a number of job slots available; queues dispatch jobs to fill those slots. You can specify a maximum number of job slots per queue and per hosts. This arrangement provides the flexibility to limit the number of jobs a host will execute concurrently.

You apply the number of available job slots as a parameter when configuring hosts. The number of job slots assigned to a particular host will define how many jobs a host will execute concurrently.

The host's operating system, and not the job slot, controls the number of CPUs used to complete the job.

By default, all hosts are configured with the number of job slots equal to the number of physical CPUs on the host. When you apply the job-slots parameter to queues, it specifies the maximum number of job slots — across all hosts — that a particular queue can consume concurrently. Limiting the number of job slots on hosts and queues gives an administrator control over the amount of resources that a specific queue can consume across the entire grid.

Creating policies for sharing resource capacity

No organization runs only one single application. So the SAS Grid environment must allow multiple applications to share resources effectively and efficiently. These applications must be able to share the grid environment in a load-balanced and policy-based manner, so they can meet the *service level agreements (SLAs)* of multiple business units.

Several business groups can benefit from the greater processing speed that a grid environment makes possible. For example, a developer working on *extract, transform, load (ETL)* capabilities can load data from multiple sources in parallel. Data miners performing analysis to do model training that uses multiple modeling techniques can execute those analyses in parallel. The time savings are considerable.

The environment must be able to balance the workload across multiple groups of users according to their computing needs. For instance, suppose that different users must be able to define and enforce policies based on four factors:

- User priority
- Time of day
- Processor speed
- Temporary storage space

Enforcing policies based on these factors means that the infrastructure must be able to lend and borrow resources efficiently among the different workloads.

Scheduling with SAS

Enterprise scheduling of jobs is an important and necessary function in every organization. SAS Grid Manager does this with the Schedule Manager plug-in that comes with the SAS Management Console. With this you can create SAS workflows and schedule them based on time, file events, and the return code of the scheduled SAS workflow. Multiple workflows — as well as the jobs within a single workflow — can then be distributed to the SAS Grid environment.

With the SAS scheduling framework you can automate the scheduling and execution of SAS jobs across your SAS Grid. Scheduling requires three main components:

- **SAS applications:** SAS applications create the jobs to be scheduled. In addition, you can also schedule SAS programs that programmers have written.
- **Schedule Manager plug-in:** The Schedule Manager plug-in for the SAS Management Console sets or enables you to edit the specific schedule and conditions for a job's execution, and specifies which scheduling server to use. (Refer to Chapter 2.)
- **The Platform Suite for SAS:** This is part of SAS Grid Manager and determines when a job's schedule conditions have been met and then runs the job. It also manages dependencies between the jobs and determines when events specified by the Schedule Manager have occurred.

Perusing the SAS Scheduling

SAS Scheduling is an integrated job-scheduling program. It's specifically designed to manage your complex flows of SAS jobs in an efficient manner.

SAS Scheduling is directly integrated with SAS Data Integration Studio, SAS Marketing Automation, SAS Marketing Optimization, and SAS Web Report Studio.

You can also use the SAS scheduling framework to schedule and execute flows. (We discuss this more in this chapter in the "Executing jobs" section.)

Getting set up

The SAS Management Console handles and manages the SAS configuration. The SAS Metadata Server captures metadata for the servers, users, groups, flows, and jobs that are deployed from SAS applications. The Server Manager plug-in to the SAS Management Console does the scheduling and defines the three types of SAS *batch servers* that execute SAS jobs without user input or intervention:

- **SAS DATA Step Batch Server:** This process is used to execute jobs typically generated and deployed from SAS Data Integration Studio.
- **SAS Java Batch Server:** This process executes jobs submitted from Java applications such as SAS Web Report Studio.
- **SAS Generic Batch Server:** This environment supports execution of standalone commands that may be submitted from various SAS processes.

Creating and managing jobs

The Schedule Manager plug-in to the SAS Management Console — along with the integrated scheduling SAS applications such as SAS Data Integration Studio and SAS Marketing Automation — creates flows and manages tasks.

TIP

Some SAS applications (such as SAS Web Report Studio) can create and manage flows from within the application.

Executing jobs

A *flow* is comprised of one or more jobs deployed for scheduling. It's created in one of two ways: by SAS applications that feature the integrated scheduling capability or by the Schedule Manager that can create and maintain flows for applications and programs that don't have the integrated scheduling.

Each deployed job is associated with a batch server (refer to "Getting set up" a bit earlier in this chapter). SAS Grid Manager is responsible for managing dependencies between jobs, time, and files. When dependencies are satisfied, the batch server submits jobs to SAS Grid Manager so they can be executed.

To schedule SAS jobs, as used by SAS applications and programs, the jobs must be deployed to a SAS batch server. (Figure 3-1 shows the overall architecture.)

Jobs are executed in the following way: When SAS applications deploy jobs for scheduling, the information about the jobs are first saved in the SAS Metadata Server. The jobs are then loaded into the SAS Management Console where it uses the Schedule Manager to create the flows. Then Process Manager schedules these flows and submits the jobs to SAS Grid Manager for execution.

Schedule Manager gets metadata information about constructing the flow from the SAS Metadata Server, then converts it to metadata that the Platform Suite for SAS understands, and submits that metadata to the scheduling server. Each job can have its own combination of time and date return codes of prior jobs, or file dependencies. After all dependencies are met for each job, they can be executed.

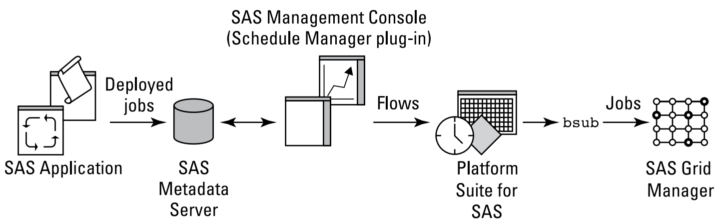

Figure 3-1: Overall scheduling architecture.

Meeting peak demand with added capacity

As the entire organization works to meet its SLAs, multiple applications are always running in a shared-computing infrastructure. Of course, when system loads peak at various times, the system reaches its resource limits. But how can you handle these peak loads?

You can answer the need if you dynamically add capacity to the SAS Grid environment when it reaches a pre-defined load condition on the system. Ideally, you need to be able to do the following:

- Bring additional hardware into the grid at peak application load times
- Dynamically detect the presence of the new resources
- Have the running applications start to use these additional resources automatically

An approach that works is to add grid nodes to the SAS Grid to get more capacity. However, growing the grid is typically done to add capacity for a sustained need for additional resources. A solution to address temporary peak demands for additional resources is to use the SAS Grid Manager dynamic host capability to add resources when needed and release them when they are no longer needed.

The SAS Grid Manager dynamic host capability allows you to automate adding more grid nodes to the cluster when they're needed and then give them back when the peak load is over.

In preparation the nodes are pre-loaded with software and identified by the system as candidates. When the grid approaches a full load condition, the new grid nodes can be added to the grid based on the rules that were set in advance. Conversely, when the heavy workload is gone, the additional grid nodes can be removed from the grid and go back to what they were doing. This is a great way to flex the capacity of the grid when it's needed. It also alleviates the manual tasks and saves the administrator.

Chapter 4

High Availability

In This Chapter

- Configuring the grid to be highly available
- Knowing what's needed for high availability
- Re-queuing and restarting in case jobs fail
- Seeing an example of SAS Grid Computing in financial services

No matter what happens, your organization needs its programs and servers to do what they're supposed to do in the right amount of time. In other words, your SAS programs and servers must be highly available, so that even if they fail, work will still be completed in a timely way. For example, if a particular SAS program takes a long time to run but ends prematurely, it can't be required to restart from the beginning. You don't have time for that.

In this chapter we look at how the SAS Grid environment can be configured to be highly available, so jobs get done on time.

Providing High Availability with SAS Grid Manager

With SAS Grid Manager, you can configure the critical services within your SAS Grid environment to be highly available. SAS Grid Manager can monitor the critical services, detect whether they fail — or the machine on which they're running fails — and automatically start the services on a failover host.

You want your critical services to stay open and working for clients without manual intervention. If a service fails, you can use the hardware load balancer or DNS name resolution, to redirect clients to the failover host where the service is restarted. (For details, see "Informing the client," later in this chapter.)

Avoiding the do-over

When a program ends before it's completed, starting over from square one costs valuable time that you probably can't afford. Using the SAS Grid Manager Client Utility (SASGSUB) or SAS Schedule Manager, you can specify that any SAS programs submitted to the SAS Grid restart automatically from the last completed checkpoint, if they end abnormally. The job restarts from the last completed procedure, labeled section, or *DATA step* (SAS code that typically performs data manipulation in preparation for analytic processing).

If a job takes a long time to execute, don't worry. You can use the restart capability with queue options that automatically re-queue jobs that end prematurely.

Fighting failure

With SAS Grid Manager, fear of failure is a thing of the past. SAS Grid Manager provides high availability thanks to several features. Each of these strategies is independent of the others; you can implement those that work best for your organization:

- **Multimachine architecture:** Because of how a SAS Grid is configured and operates, no single point of failure exists. Jobs are processed on the available grid nodes; if a node becomes unavailable, other nodes can take over the workload. (Refer to Chapter 1 for details.)
- **Automatic failover nodes:** The out-of-the-box configuration of the daemons for SAS Grid Manager provides high availability for the entire grid operation. The master process runs on a specified grid node (usually the grid control server), with a failover node identified. If the master process node fails, the failover node automatically takes over and broadcasts to the rest of the grid. The grid

immediately recognizes the new master process node and continues to run — no interruption necessary.

- **Failover for critical services:** Critical services and processes for SAS Grid applications (for example, the SAS Metadata Server) must always be available and running. With SAS Grid Manager, you provide a failover host for each service. You can then use SAS Grid Manager to monitor each service, restart it if it stops, and start the service on the failover host as needed. When the service has started on the failover host, you can use either hardware (a load balancer) or software (Domain Name Server [DNS]) to automatically direct clients to the failover host — they'll have no idea that a problem even existed.
- **Automatic SAS program failover:** You can use SASGSUB and the SAS Schedule Manager plug-in to specify that a job is *restartable* if it fails. The job will restart from the last successful procedure, DATA step, or labeled section. When you use the SAS checkpoint and restart functions, any failed jobs on the grid get done without delays. You can also use the queue in the grid to automatically restart and re-queue any job that meets these criteria:
  - Ends with a specified return code
  - Terminates due to a host failure

  Using the restartable capability and the queue together, you can ensure that critical SAS programs always run successfully and in the right amount of time.

Setting Up High Availability for Critical Applications

Certain services on the grid must always be available and accessible to clients. Examples of these critical services include:

- **SAS Metadata Server:** This process administers the "data about data," analytics, and processes within the overall SAS environment.
- **SAS Object Spawner:** This process generates "proxy" processes to fulfill user requests for SAS processing.

- **Web application tier components:** These Java applications make up SAS web-based applications such as SAS Web Report Studio.

Listing the required components

To configure a grid that ensures these services are highly available, you need to provide these components:

- **Failover hosts for machines:** With multiple machines, the grid has no single point of failure for critical applications. That means the applications can keep running.
- **A method for monitoring:** You need to be able to monitor critical applications on the grid and have them restart automatically in the event of a failure.
- **A way to let the client know to connect to the failover host instead of the regular host:** This can be done through software (DNS resolution) or hardware (the hardware load balancer), as we detail later in this chapter.

Here's how things normally work:

1. The client needs to access a service on a machine in the grid.
2. The client sends a query to the corporate DNS server. The DNS server looks up the address for the machine and returns that information to the client.
3. The client uses the address to connect to the machine and use the application.

To provide business continuity for the application, you must provide a failover host for the critical services running in the grid environment. Having a failover host offers an alternative location where the critical services can keep running and stay available to the applications on the grid.

REMEMBER

Both the main and failover machines must have access to a shared, or clustered, file server. This arrangement ensures that the application has access to the data required to operate, regardless of which machine is running the service. A shared/clustered file system maintains consistency of the data by controlling access and ensuring updates are applied properly. An example of a shared file system is the IBM Spectrum Scale environment, as we discuss in Chapter 6.

The failover capability must be automatic. SAS Grid Manager can be configured to monitor critical services running on the grid. If it detects that a critical application has failed or that the machine running it has gone down, it starts the application on the failover server automatically. When applications continue to run on the grid despite failure of individual components, you can ensure the consistent delivery of services.

Informing the client

After an application has started on the failover server, the client must have a way to know which server is running the application. You can provide that capability via a load balancer or through DNS resolution.

Using a hardware load balancer

The machine configured as a *load balancer* acts as an intermediary between the client and the services running on the grid. That way the grid's operation is separate from the physical structure of the grid. When the client wants to connect to the service, it connects to the load balancer, which then directs the request to the machine that is running the service. During normal operation, the request goes to the main machine. When failover occurs, the service is started on the failover host, and the load balancer forwards all client connections to it (because the failover host is now where the services are running).

DNS resolution

After SAS Grid Manager starts the application on the failover server, it sends the failover machine's address to the corporate DNS server, and it is updated. So the next time a client requests a connection to the application, the DNS server returns the address of the failover machine.

TIP

If you don't want SAS Grid Manager to update the corporate DNS directly (and most people don't), you can configure the DNS server to always point to SAS Grid Manager as the source that provides the IP address for each machine. When SAS Grid

Manager starts an application on the failover machine, it then points client requests to the failover machine's IP address.

Choosing between the two

Selecting to use a load balancer or a DNS solution depends on your organization's policies. Using DNS resolution means you don't have to buy additional hardware (in this case, a load balancer). However, your organization's policies might prohibit the corporate DNS from being changed by an outside DNS (in this case, SAS Grid Manager), or forbid that DNS requests be forwarded to an outside DNS (again, SAS Grid Manager). If such prohibitions are in place, the hardware load balancer provides a high-availability solution.

Restarting Jobs

A highly available grid can handle SAS jobs that fail or have to be restarted for some reason. If a long-running job fails, it can seriously affect productivity. After you notice the failure, you must manually resubmit the job and wait while the program starts over again from the beginning. For production SAS programs that take longer to run, this resubmission process can cause unacceptable delays.

When combined with the SAS Grid Manager queue policies and the SAS checkpoint restart feature, SASGSUB and the Schedule Manager in SAS Management Console provides support for two solutions to this problem:

- The capability to restart a job from the last successful job step
- The capability to set up a special queue to send failed jobs automatically to another host in the grid, where they can continue execution

Using SAS Checkpoint and Label Restart

SASGSUB and the Schedule Manager in SAS Management Console enable you to restart SAS programs from the last successful step within the SAS source code.

When the program runs, it records information about the SAS procedures and DATA steps or labels in the program — and tracks the ones that have completed successfully during execution.

If the program fails and has to be restarted, SAS first executes global statements and macros. Then it reads the checkpoint or *label library,* a list of completed sections of code, to determine which checkpoints or labels have been passed. When SAS determines where the program stopped, it restarts the program from that point. Program steps that have already successfully completed aren't re-executed.

REMEMBER The restart capability is available on the grid only if you're using SASGSUB or the Schedule Manager in SAS Management Console to run batch jobs. It isn't available if you're using other application interfaces to submit work to the grid.

TIP If you use the restart options, your SAS *WORK library,* the area that SAS uses for temporary results or "in progress" data, must be on shared storage. Using the checkpoint restart capability adds some overhead to your SAS program, so it is not recommended for every SAS program that you run.

Setting up automatic job re-queuing

Job re-queuing enables you to handle situations where the host or the system fails while the job is running. You can set up a queue that re-queues and re-dispatches any job that ends with a specified return code or terminates due to host failure. Using the re-queue capability ensures that any failed jobs are dispatched automatically to another node on the grid.

SAS Grid Computing in Action: Financial Services

Imagine a provider of financial services whose SAS Enterprise Business Intelligence environment has users from across the company — human resources, marketing, risk management, card member services, finance, and consumer banking.

The SAS environment processes mostly ad hoc jobs — model development, reporting, campaigns, and so on. Some of the processing runs monthly production models; some takes place as users develop and test new models before full implementation in production. Most of the financial-services data comes from relational databases housed on separate servers. The company uses SAS ACCESS engines to handle data from the databases, and employs other data mining and modeling products.

Challenges in the existing environment

Suppose that the financial services organization's former environment had the following technical features: a SAS 9.2 M3 EBI installation housed at an offsite vendor location, consisting of a single IBM POWER server configured in the following four logical partitions (LPARs):

- **Ad hoc:** Assigned for business users
- **Metadata:** Assigned for running the SAS metadata server
- **Production:** Assigned for model run
- **Development:** Assigned for model development (also acts as a staging area)

Here the largest LPAR was the ad hoc area, which almost monopolized the majority of the CPUs and memory. This setup resulted in frequent processing bottlenecks, particularly in the ad hoc environment where many concurrent SAS users had to share a single, large, multi-CPU server. As the company's user base continued to grow and demands on the server increased, it became obvious that the company's large, multi-user environment had grown beyond what a single server could be expected to handle.

Server availability

The first challenge posed by the old environment was the server's limited availability. Because the four partitions sat on a single frame, any firmware upgrades on the hardware — or operating system-level maintenance — required downtime across the entire frame. Result: less server availability.

Resource mobility

The second challenge was a lack of resource mobility across the partitions. Each partition had its own resources and didn't share them. Thus the production server was busy only during month-end processing and was underused the rest of the time. The ad hoc server was almost always busy and was under near-constant resource constraints. Result: some resources were always underused and others were always overused.

Multiple copies

A third challenge was the frequent need to create multiple copies of data to keep different areas — development, production, and ad hoc — up to date and on the same page. The file system could not make shared data available. Thus users who were analyzing production data needed to have month-end data copied from the production server to the other environments, especially the ad hoc server (which already had too much to do). Also, when the models in production required changes or fixes, data was often moved from production to development. Too many copies of data can breed confusion and inefficiency.

Cost of upgrades

The fourth challenge was the cost of adding more CPUs, memory, and storage to accommodate the growing number of users. These upgrades cost the company in productivity, because all the partitions needed downtime. The maintenance costs of the partitioned frame were mounting, and the cost of replacing that server was prohibitively expensive. The company was between a virtual rock and a virtual hard place.

SAS is the answer

By implementing SAS Grid Computing along with an IBM Spectrum Scale file system, the financial services organization significantly increased productivity and lowered costs. The company could now take advantage of previously underused computing resources while providing a more stable and scalable environment. In this case and many others, grid computing is a more cost-effective way to provide large amounts of computer-processing cycles to internal company units and to business partners.

The SAS Grid implementation provides the enterprise with many benefits:

- Workload management across nodes can now ensure a smoother and more efficient use of existing resources.
- Multiprocessing capabilities allow workflows to be processed in parallel on multiple machines, increasing processing speed without adding new hardware.
- IBM Spectrum Scale clustered storage enables the sharing of data across platforms, reducing the overall amount of disk storage necessary for servicing the user community and executing production models.
- Scalability is high, even though the initial outlay required is far less than it would be for replacing the existing environment with similar hardware and software.
- Confidence increases that further refreshes of the system won't need downtime.
- This system adds a new grid node and more data storage, showing that computing resources can be expanded affordably, adding new users and capacity without affecting operations.

The results of this change are significant: the ability to run all current SAS production applications for more than a year with no resource downtime, while increasing the number of users and adding new computation and storage resources with no disruption of service.

An organization that successfully migrates to an Enterprise SAS Grid platform is closer to ideally addressing its business requirements and technology goals.

Chapter 5

Running SAS Programs on Your SAS Grid

In This Chapter

- Identifying benefits of SAS Grid Manager
- Being interactive with SAS
- Using batch jobs
- Utilizing SAS client applications
- Exploring parallel processing

If you're a programmer or user of SAS client applications. You may be wondering, "What's in it for me?" In this chapter, we show why and how you, the user, can take advantage of SAS Grid.

Benefitting from SAS Grid

Programmers and users of SAS applications running on SAS Grid take advantage of the following benefits:

- More computing power than your local machine or a single departmental server
- Higher availability at all times
- Managed queues and priorities to avoid bottlenecks and give everyone the fastest throughput
- Parallel processing to speed up the time it takes a multi-step long running program to execute
- Automatic restart of a failed job

The SAS Grid Manager administrator sets up options and configures the grid. Check with your administrator to see how your grid is configured and what rules are in place for job queues, priorities, and so on.

Grid Enabling Your SAS Program

SAS programs can be submitted to the grid interactively using SAS Display Manager, SAS Studio, or any client application that allows code to be submitted. SAS jobs can be grid-enabled by adding a set of grid statements. When the job runs, it will send the program to the grid to execute and then return the log and results to you.

To grid enable your SAS code, add these five lines:

1. GRDSVC_ENABLE function to identify a SAS application server and set options.
2. SIGNON statement to start a session on the grid.
3. RSUBMIT statement to begin the block of code to be run on the grid.
4. ENDRSUBMIT statement to end the block of code to be run on the grid.
5. SIGNOFF statement to end the session on the grid.

The GRDSVC_ENABLE function tells SAS to use the grid and specifies the SAS application server. Optionally, you can set a job name and other grid options.

The SIGNON and SIGNOFF statements start and end a session on the grid. The only required option is a session-id.

The RSUBMIT and ENDRSUBMIT statements define the block of code that will be run in the grid session.

Say you want to look through a database for cars that have high miles per gallon (MPG). Here's an example of the statements needed to run a simple SQL query on SAS Grid using a parameter that's passed from the local session and returning the subset of data of high MPG cars.

```
options
   metaserver=sasserver.demo.sas.com
metaport=8561 metapass=xxxxx
metauser=xxxxx;

%let rc=%sysfunc( grdsvc_enable(
_all_, server=SASApp ));

signon grid1;

%SYSLPUT mpg = 30;

rsubmit;

   proc sql;
       create table WORK.HIGHMPG as
          select * from SASHELP.CARS
          where mpg > &mpg;
    quit;

    PROC DOWNLOAD data=WORK.HIGHMPG
             out=WORK.remote_results;
    run;

endrsubmit;
signoff grid1;
```

TIP

You can find examples for using remote sessions and passing data in the SAS/CONNECT 9.4 User Guide documentation.

Submitting SAS Batch Jobs

If you have access to the SAS Grid Manager Client Utility, you can run SAS jobs in batch mode using the SASGSUB command. To submit your job, open a command window on a grid client and type the following:

```
SASGSUB -GRIDSUBMITPGM sas-program-file
```

You can add several options to the SASGSUB command. The –GRIDWATCHOUTPUT option allows you to see the SAS log in the command window while the job is executing. It's an easy way to see problems that happen early in the code. Your command window will be taken over until the job finishes.

When you don't want to see the log as the job runs, you can use the –GRIDGETRESULTS command after the job finishes. This command will move the log and results of the job to the current directory or a directory of your choice. You can specify the job ID for a specific job or get the results for all jobs.

```
SASGSUB
-GRIDGETRESULTS job-id
-GRIDRESULTSDIR directory
```

You can find more information on syntax, options, and details in the SAS Grid Manager in SAS 9.4 documentation.

Using SAS Client Applications

How SAS client applications interact with SAS Grid can depend on the client application, the version of the client application, and how the grid is configured for load balancing.

If SAS is configured using grid-launched workspace servers, using the grid is mostly transparent to you. With grid-launched workspace servers, programs and projects in Enterprise Guide, and stored processes in SAS Add-in for Microsoft Office are automatically sent to the grid without any need to modify code.

If you're not using grid-launched workspace servers, you'll need to check an option in Enterprise Guide and SAS Add-in for Microsoft Office to specify that the code is to run on the grid.

All editions of SAS Studio can use the GRDSVC_ENABLE function and SIGNON/RSUBMIT/ENDRSUBMIT syntax to grid-enable and run code on the grid. The SAS Studio Enterprise Edition can take advantage of the load-balancing and grid-launched workspace server capabilities.

Refer to the SAS Grid Manager in SAS 9.4 documentation for more information on how these and other SAS client applications can take advantage of the grid.

REMEMBER

Enterprise Guide establishes a SAS Grid session the first time you access a workspace server. It keeps and reuses that grid session until you close the application. This provides continuity and avoids the overhead of starting and ending sessions.

SAS Grid Manager assigns the least busy server at the time. Over time, what was the least busy server could become the busiest server. If you have Enterprise Guide open for a long period of time and your jobs start taking longer to run, close the application. When you reopen the application, SAS Grid Manager will assign a new workspace on the currently least busy server.

Exploiting Parallel Processing

If your program has multiple independent steps, you can submit the steps to run at the same time as separate grid jobs. For example, your job might read in data from two different data sources, do some initial processing on each one, and then join them together to create a report. The steps that read and process the data from source 1 can be running in one grid session at the same time as the steps that read and process the data from source 2. After both of these steps are complete the report steps can be run.

In order to make the sessions run at the same time, there are options that you the user need to add to the SIGNON and RSUBMIT statements. There is some overhead in establishing a SAS Grid session, and the grid administrator may set limits on the number of sessions any one user can have at a time.

If you know the expected running time for each step, manually balancing the load is easy. For example, you know that Steps 1 and 3 together take about as long as Step 2. In your job, Steps 1 and 3 run in session A and Step 2 runs in session B. In the RSUBMIT statement for Step 3, you would set it to wait for Step 1 to complete.

Determining which steps to put together for the best utilization of computer resources can be difficult and can vary over time. You can use macros to dynamically determine which sessions are available and have steps wait for an available session. This technique provides the best overall reduction in job execution time that can adapt as needed.

If you want to know more and see example code of parallel processing techniques, have a look at this paper: SAS1935-2015 "Divide and Conquer – Writing Parallel SAS Code to Speed Up Your SAS Program" by Doug Haigh, SAS Institute Inc., Cary, North Carolina, at `http://support.sas.com/resources/papers/proceedings15/SAS1935-2015.pdf`.

Chapter 6

Managing the Grid

In This Chapter

- Meeting some SAS users
- Looking at ways to manage the grid
- Using job slots
- Being specific about resources
- Utilizing Grid Option Sets

Organizations that use SAS are made up of many different kinds of users. Each category of user has its own needs and expectations for getting work done. To keep your SAS grid environment running effectively, you need to manage where jobs run on the grid.

In this chapter, we look at a few different types of SAS users, and then explore ways that you can manage your organization's SAS Grid so work is executed properly and in a timely way.

Looking at Types of Users

Here are a few examples of SAS user categories. You might find these are common to your organization:

- **SAS Enterprise Guide and SAS Add-in for Microsoft Office users:** These users are usually running interactive programs — and expect immediate results.
- **SAS Enterprise Miner users:** These users might be using multiple machines to train models.

- **SAS Web Report Studio users:** These users might be scheduling reports to run at a specified time.
- **SAS Risk Dimensions users:** These users might be running jobs at night.

Some users in your environment might be running high-priority jobs. Others might be running jobs that require a lot of computing resources. A SAS Grid environment must be able to account for all these different needs, priorities, and workloads.

Considering Grid-Management Methods

To manage the SAS Grid environment with so many different users, you must be able to control when and where jobs can run on the grid. You can manage grid resources with these strategies, each of which we explain further in this chapter:

- **Job slots:** This setting enables you to control how many jobs can run concurrently on each machine on the grid. That way you can tune the load that each machine in the grid can accept. (For more about job slots, see Chapter 3.)

  For example, you can assign a higher number of job slots to higher-capacity machines, which specifies that those machines can process more jobs concurrently.
- **Queues:** *Queues* enable you to control when jobs can run and what resources are available to the jobs in the queue. You can create queues based on factors such as job size or priority. You can also define job *dispatch windows* (a block of time in which jobs dispatched from a queue are permitted to run) and *run windows* (times when jobs from a given queue can be dispatched for execution). (For more about queues, see Chapter 4.)
- **Resources:** *Resources* define specific capabilities of certain machines within the grid. For example, you may want SAS Enterprise Miner (SASEM) jobs to execute only on certain machines within the grid. By creating a SASEM resource and assigning it to those machines, you ensure

that only the jobs requesting the SASEM resource can be dispatched for execution there. You can specify where jobs are run on the grid by specifying resource names on hosts and matching resource names on jobs.

- **Grid Options Sets:** This is new with SAS Grid Manager 9.4. Different groups of users of a SAS application may need changes to the configuration and resource settings before they submit work to the queue. In the past, multiple application server contexts were created in the SAS metadata server for this, but it was sometimes complex to manage. Grid Options Sets provide a better way to allow these different user groups to run the SAS application tailored to their needs. (Refer to the later "Using Grid Options Sets" section for more information.)

Specifying Job Slots for Machines

SAS Grid Manager uses job slots to specify the number of processes that are allowed to run concurrently on a machine. The default number of job slots for a machine is the same as the number of processor cores in the machine.

A machine can't run more concurrent processes than it has jobs slots.

You can configure hosts given the many different types of SAS users, a single core can typically handle much more than a single job or process to handle more job slots than the number of cores. To do so, you set the MXJ value for the host to a fixed number of job slots. This enables more SAS jobs to run concurrently to increase throughput across the grid.

Using Queues

When a job is submitted for processing on the grid, it's placed in a queue and held until resources are available for the job.

SAS Grid Manager processes the jobs in the queues according to specified criteria such as which jobs are to be processed first, what hosts can process a job, and when a job can be processed.

All jobs submitted to the same queue share the same scheduling and control policy. By using multiple queues, you can control the workflow of jobs that are processed on the grid.

Defining and Specifying Resources

Defining resources enables you to specify where jobs are run on the grid. You can define resource names on grid nodes and then specify those same resource names on jobs in the grid. Instead of running on any available grid node, the job will be dispatched for execution only on grid nodes that have the matching resource. Administrators can contain jobs with high CPU or memory requirements, such as SAS Enterprise Miner, to specific grid nodes so that other SAS jobs are not impacted.

The resource names on grid machines indicate the type of job each machine runs (for example, jobs from specified applications or high-priority jobs). So you can direct specific types of work to the nodes that are best suited for processing them.

Using Grid Options Sets

Each SAS Application Server is associated with a single queue for submitting jobs but not everyone can use the default settings. There are times when different users — or situations — need changes to the configuration and resources for the jobs to run efficiently.

Grid option sets are a collection of grid options, SAS options, and required resources. The purpose of a grid options set is to allow a SAS Grid administrator to define a collection of options in SAS metadata that map to one or more SAS client applications. These options are automatically applied to the workload that is submitted to the grid based on the identity of the user accessing the client application.

TIP

The only applications that you can associate with a Grid Options Set are ones that have been identified as being grid capable. Applications are identified as grid capable either by the application's deployment process or by specifying the 'isGridCapable' keyword through the application's property values.

A Grid Options Set is a collection of SAS Options, SAS Grid Options, and Required resources. They're the changes from the default queue settings. When Grid Options Sets are created, their values are merged with the default option values to create the Logical Grid Server. The SAS Grid Administrator creates them in the SAS Management Console as a Logical Grid Server definition.

Chapter 7

Managing Storage

In This Chapter

- Defining a high-performance shared file system
- Knowing the benefits of IBM Spectrum Scale as a high-performance shared file system in grid for SAS Grid Computing

The benefits of SAS Grid Computing are many, and improvements to the overall performance of your SAS applications are always a very welcome benefit. If the storage subsystem isn't designed and implemented properly, however, you may be hard pressed to realize any of these performance gains. Having the best storage performance possible is absolutely critical.

What you need is a high-performance shared file system — an integral and required component of all SAS Grid Manager deployments. This chapter includes contributions from IBM and explores the IBM Spectrum Scale high-performance shared file system.

IBM Spectrum Scale File System

The IBM Spectrum Scale file system (formerly known as GPFS) is a full-featured set of file management tools that includes the following:

- Advanced storage virtualization
- Integrated high availability
- Automated tiered storage management
- The capability to manage large quantities of file data effectively

IBM Spectrum Scale gives a group of computers concurrent access to a common set of file data over a storage area network (SAN), a standard network, or a mix of connection types. IBM Spectrum Scale provides storage management, tools for information lifecycle management, and centralized administration. It also allows for shared access to file systems from remote clusters of IBM Spectrum Scale servers.

An IBM Spectrum Scale cluster can consist of a single node, two nodes, or thousands of nodes on the same grid. (For more about nodes, see Chapter 2.) The largest existing configurations exceed 5,000 nodes. In fact, IBM Spectrum Scale was purposefully designed to support high-performance parallel workloads. The bigger the workload, the more nodes you have to manage.

Performance and scalability

IBM Spectrum Scale provides exceptional high-performance input/output (I/O) to your SAS Grid environment by doing the following:

- Striping data across multiple disks attached to multiple nodes
- Scanning high-performance metadata (inode)
- Supporting a wide range of file-system block sizes to match I/O requirements
- Using advanced algorithms to improve read-ahead and write-behind I/O operations
- Using block-level locking based on a very sophisticated, scalable token management system, providing data consistency while giving concurrent file access to multiple application nodes

When all nodes in an IBM Spectrum Scale cluster are defined, any node can access a server's drive in two ways:

- Using a local connection to the drive.
- Using the network protocol specified on the IBM Spectrum Scale *network shared disk (NSD)*.

The NSD protocol allows for shipping data over a Transmission Control Protocol/Internet Protocol (TCP/IP) or via an InfiniBand connection.

Data availability

You can configure IBM Spectrum Scale to permit continued access to data even if a cluster node or storage system fails.

Clusters

IBM Spectrum Scale offers robust support for clustering and data replication. The IBM Spectrum Scale software can handle data consistency and availability. It doesn't rely on external applications to handle cluster operations such as node failover (described in Chapter 4).

This clustering support goes beyond managing who owns the data or who has access to the disks. In an IBM Spectrum Scale cluster, all nodes see all of the data. Any node in the cluster, provided it has a server license, can perform cluster operations. All nodes are capable of performing all tasks.

IBM Spectrum Scale continuously monitors the health of the file system's components, taking action automatically when it detects failures. It has extensive journaling and recovery capabilities so metadata consistency is maintained if a node fails.

Snapshots

Snapshots help protect the file system's contents against a user error. They preserve a point-in-time version of the file system or a subtree of a file system, called a *fileset*. IBM Spectrum Scale implements a snapshot mechanism that generates a map of the file system or fileset at the time the snapshot is taken. New data blocks are consumed only when the file system data has been deleted or modified after the snapshot was created.

Snapshot data is placed in existing storage pools (which we define in the next section). This approach simplifies administration of the file system and makes the best use of existing storage. The snapshot function can be used with a backup program, for example, to run while the file system is in use, and still obtain a consistent copy of the file system as it existed when the snapshot was created.

REMEMBER

In addition, snapshots provide an online backup for the file system that allows files to be recovered easily in case of common problems, such as accidentally deleting a file.

Information lifecycle management

To simplify administration, IBM Spectrum Scale is efficient at *information lifecycle management (ILM),* by using automation and tiered storage management. Storage pools, filesets, and user-defined policies allow the cost of storage to match the value of the data.

Storage pools

Storage pools manage groups of disks within a file system. Using storage pools, you can create tiers of storage — hierarchical levels that correspond to set priorities — by grouping disks based on performance, locality, or reliability characteristics. For example, one pool could contain high-performance *solid-state disks (SSDs)* while another pool could have more economical storage disks that spin at a modest 7,200 RPM. When data is placed in, or moved among, storage pools, IBM Spectrum Scale does all the data management.

Filesets

A *fileset* is a subtree of the file system namespace. It provides a way to divide the namespace into smaller, more manageable units. Using filesets, you can provide administrative boundaries to set quotas, take system snapshots, define Active File Management (AFM) relationships, and set user-defined policies to control where data is stored or where it's moved. Data within a single fileset can reside in one or more storage pools.

TIP

Active File Management (AFM) is remote file-caching technology that ensures high-performance access to files, regardless of their geographic location. AFM is useful for sharing large volumes of file-based data that must be highly available, no matter where the information resides. For more about IBM Spectrum Scale and AFM, go to `www-03.ibm.com/systems/storage/spectrum/scale/features.html`.

REMEMBER

Business users and your IT department will determine where the file data resides — and how it's managed after it's created — by creating a set of rules in a user-defined policy.

User-defined policies

IBM Spectrum Scale has two types of user-defined policies:

- **File-placement policies:** These policies determine the storage pool in which file data is initially placed. They're defined for each file by using a file's attributes, such as filename, fileset, or the user who created the file.
- **File-management policies:** When files exist in a file system, they must be migrated, deleted, modified, have their replication status changed, or be the subject of generated reports. For those purposes, you need file-management policies.

Data within a single fileset can reside in one or more storage pools. A set of rules in a user-defined policy determines where the file data resides — and how it's managed once it's created.

You can, for example, use a migration policy to move data from one storage pool to another without changing the file's location in the directory structure. It's a faster way to migrate. Similarly, you can use a policy to change the replication status of a file or set of files if copies are proliferating beyond what's useful or appropriate. *File deletion policies* provide rules for pruning the file system, getting rid of files that have outlived their usefulness. And with all this movement and deletion going on, you need *list policies* that let you quickly scan the file system's metadata and produce information on selected attributes of the files you want to manage.

Being efficient with metadata

File management only works well if you process file metadata efficiently. So IBM Spectrum Scale includes an interface that offers high-performance metadata scanning — just in case you need to process the metadata for billions of files, and do it efficiently. Using the tools available in IBM Spectrum Scale, you can automate file management on just about any scale. The secret is a high-performance metadata scan engine that allows one or more nodes in a cluster to concurrently identify files that need specific changes as well as handling data movement. IBM Spectrum Scale can get multiple nodes working in parallel on rule evaluation and data movement, no matter how big your system.

File-management policies can use the file attributes collected by file-placement policies. For example, file placement can use attributes such as last-access time, size of the file, or a mix of user name and file size. Any and all of those items of metadata can be useful in managing the file.

Using IBM Spectrum Scale with SAS Grid Manager

IBM Spectrum Scale is an ideal storage subsystem to support SAS Grid deployments. It's a powerful way to support some of the promises of SAS Grid Manager — such as better performance and availability.

Increasing performance

The rate at which IBM Spectrum Scale can transmit data is limited only by how fast the physical device can handle data. The system can work in top form and sustain excellent performance while the file system maintains relatively low overhead and stays manageable and responsive.

In addition, IBM Spectrum Scale can improve on sequential data access (the approach that SAS normally uses). The IBM Spectrum Scale file system can be created with *cluster allocation* that allocates data blocks close to each other, enhancing performance.

For SAS software, the access pattern to the file system's metadata is much different from the access pattern used for data. File system metadata tends to be randomly accessed in small blocks; file system data, on the other hand, is accessed sequentially in larger blocks. With IBM Spectrum Scale, you can separate metadata from data — and use parallel data movement to your advantage.

A best practice is to create separate volumes for file system metadata and use fast storage such as solid-state devices (SSDs), high-RPM drives, or memory-based drives to significantly improve the responsiveness of the file system.

Ensuring data availability

IBM Spectrum Scale's features eliminate single points of failure so data is always available, which is critical in any grid environment. IBM Spectrum Scale continuously monitors the various file system components and automatically takes action if a failure occurs. In addition, it uses snapshots to preserve past versions of the file system.

Managing the system is simpler

IBM Spectrum Scale information lifecycle management makes a shared- file system much less complex and arduous to manage than it used to be with earlier clustered file systems. This is important for handling metadata properly to ensure a high-performing, highly available system.

Chapter 8

Seeing SAS in Action: Customer Stories

In This Chapter

- Watching SAS working for public administration
- Seeing SAS work for the banking industry
- Finding out how SAS works for the tourism industry

This chapter takes you through three real-life examples of how three very different customers used SAS Grid Manager for big gains in performance.

SAS Grid Computing in Action: Public Administration

In the Piedmont region of northwest Italy, the region's public administration is making effective use of big data in a variety of ways by using SAS solutions. Public-sector bodies that routinely collect huge and complex data sets amass these huge and complex data sets.

Meeting the customer

The consortium — one of Italy's largest and most prominent IT entities — is leading the data collection. Owned by 96 organizations within the public administration but operating as a company, the consortium employs more than 3,700 people and has an income of more than 240 million euros annually.

The consortium helps public entities to cooperate, share best practices, and optimize their internal processes. As a result, they can save time, reduce costs, and satisfy the needs of their citizens. In addition, the consortium encourages the involvement of local companies in public-sector projects, helps the companies respond to calls for tender, and supports their drive to differentiate and cooperate.

The consortium had been using SAS solutions for over 30 years. Its range of SAS tools is among the broadest in Europe; it employs more than 70 SAS experts.

Making full use of big data

The consortium's approach to data management has gone through various stages — from printed records, operational databases, and data banks to web applications. In recent years, one of the consortium's main goals was to help public administration entities share data among themselves. This goal required setting up a single regional database for public administration.

Following the creation of this resource, the consortium began to distribute master data to public organizations and to develop joint use of data. It also shared open data online. However, the volume of data was growing faster than its rate of use. The consortium began to talk not only about data in general but also about *big data,* the flood of information that comes to — and describes — just about every citizen of the modern world via the media, entertainment, healthcare, video surveillance, and social media. How could they make sense of it all — and use it to support their goals?

Harnessing the power of new analytics

The public administration entities in the consortium's region have more than 2,000 conventional databases. Arranged by subject, the number increases to more than 3,000 because many of the databases are multi-subject.

The consortium has already begun to manage various new big data categories, such as the digital library, healthcare pictorial image data, streaming data, and sensor data on the environment. It also distributes this data for public organizations to use.

The consortium views traditional business intelligence as evolving into data science. This transformation is happening because of three factors:

- Data volumes are growing exponentially.
- A completely new type of data is becoming available for analysis.
- More advanced tools and considerably greater processing power are available for the analysis.

Thus the consortium requires new kinds of experts who can produce business value from the new type of data.

SAS infrastructure takes on big data

The goals in handling big data are to improve management of the platform and reduce platform costs. The business side also needs agile processes and the ability to comply with service-level agreements established with customers. It also needs to be flexible, perform critical services, and be protected from service interruptions.

The consortium chose the SAS analytical infrastructure as its solution, based on distributed network computing, in-database computing, and in-memory computing.

The consortium selected SAS Grid Manager as its practical implementation tool. The centrally managed SAS environment allows the organization to prioritize work based on policies. SAS applications aren't disrupted by maintenance measures. Different SAS applications can be used flexibly and simultaneously. Result: Users can get more data and receive more complex analyses in less time than before.

The consortium feels that public administration is in a particularly good position to make use of big data, which is why SAS will remain a tool in the consortium's toolbox as it pursues its mission.

SAS in Action: Banking

Today banks must be able to model the credit risk of portfolios. Loan products such as lines of credit, mortgages, and credit cards entail a high degree of risk for banks, and on a large scale, defaults produce difficult situations — with huge implications for both the lender and the borrower.

Banks regularly employ credit-risk management processes to monitor and assess credit portfolios, to make estimates, and to understand their risk position and value of assets at any given time. They need accurate credit-risk management processes — based on leading technology — to mitigate their exposure.

Meet the customer

This customer is a bank with approximately 59 million consumer and small-business relationships, 6,000 retail banking offices, and more than 18,000 ATMs. It's among the world's leading wealth management companies and is a global leader in corporate and investment banking and trading across a broad range of asset classes.

A large brokerage firm manages the bank's available-for-sale portfolio and is responsible for modeling and calculating the *probability of default (PD)* on the 9.5 million mortgages it services. In addition, the group calculates the market value, prepayment speeds, and sensitivities to change in interest rates, and hedges these risks for the $19 billion mortgage-service-rights asset. Recently this brokerage firm began assisting with forecasting loan losses for the bank's credit card portfolio.

Need for speed

The brokerage firm had been using analytics from SAS to model credit risks for many years. But the new responsibility of forecasting credit card loss forced the firm to reassess its internal siloed server environment for modeling and calculation processes.

Making such a change meant getting the tools that could help reduce processing time, increase access and availability of resources for ad hoc analysis, and ensure business continuity for this critical function of the bank's business.

"Unparalleled" performance

"We now have an environment that provides users with a robust platform on which to schedule and prioritize jobs, based on duration or computational requirements, so that ad hoc usage is not competing with scheduled work," says the managing director of the firm. "This advanced grid platform is giving us unparalleled performance. SAS is indispensable for its unique way of handling large data sets."

As an example, he adds, "We have to score a particular portfolio of 400,000 loans with our suite of models, using multiple scenarios, and we need to run it over the 360 months of the mortgages' life. That process used to take three hours, now it takes 10 minutes because of the parallelization capabilities of the grid. The ability to go from three hours to 10 minutes on a job demonstrates a tremendous increase in our ability to deliver information and make decisions."

"The bank has a strong desire to enable loss forecasting as accurately and quickly as possible, right up to the senior executive layers of the organization," says the managing director of the firm. "The only way we can do that is to have sufficient IT resources to score loans and appropriately assess risks. The partnership between SAS, IBM, and our internal technology group has provided a platform for us to demonstrate risk management leadership."

Enter SAS Grid Manager

To meet its performance requirements, the brokerage firm moved its processing to a grid platform consisting of SAS Enterprise Risk Management on SAS Grid Computing, and SAS Scalable Performance Data Server on a 224 core IBM BladeCenter grid, and the IBM XIV Storage System.

The initiative produced considerable results:

- The bank's time for calculating the probability of default shrank from 96 hours to just 4.
- The processing time for ad hoc jobs shrank by 90 percent.
- Processing speed increased to three times faster than the speed of the previous environment.

REMEMBER

Now the SAS platform pulls data from eight systems of record (SOR), amounting to hundreds of millions of records, or 30 terabytes of source data, and allows the SAS environment to consume 3.9 gigabytes of I/O throughput per second from the IBM XIV storage environment. Approximately 30 users now have unfettered access to the environment, compared to the siloed server environment of the past. In the previous environment, users had to compete for time on the system; response times varied dramatically due to the high number of jobs being executed.

SAS in Action: Tourism

In traditional real estate, the three most important factors might be "location, location, location." But in the world of vacation ownership exchange, the mantra is "value, value, value."

Meet the customer

For a tourism company, the goal is to analyze the preferences and patterns of its 3.7 million members — and to fairly estimate the value of their vacation ownership weeks within the company's exchange system. These analyses are critical to the business.

Each year, millions of member-customers can exchange the timeshares they own (at resort locations and for specific weeks of the year) for any other timeshare locations/weeks deposited within a program kept by the tourism company. Of course, not all locations/weeks have equal value within the exchange program. The tourism company must assign proper value to the items in its inventory and balance supply and demand.

Previously, the company used a legacy system that supported traditional one-for-one exchanges. However, recently the tourism company launched major enhancements to the program's product, providing members with some new advantages:

- The transparency they needed to know the value of their vacation weeks
- The capability to combine multiple deposits

- The capability to receive deposit credits when trading into a week of lesser value than that of the original deposit

To support these new product features, the tourism company's revenue management system had to be redesigned.

The tourism company's team created a program trading exchange application to manage its approximately 2 million distinct *stock-keeping units (SKUs).* The team used SAS to pull transaction data from a variety of data sources and raw logs to create time-series analyses, regressions, and other models: 6 million region/week forecasts for supply forecasts, 15 million region/week/product forecasts for usage forecasts, and another 15 million region/week/product forecasts for demand forecasts. These analyses and forecasts enabled the tourism company to assign — every day — an updated exchange value to each vacation week in its inventory. As a result, its members have access to more exchange opportunities than they ever had before.

Seeing great results with SAS Grid Computing

The company now has about 25 times more supply and usage forecasts, and 60 times more demand forecasts — all done using custom regression models in SAS. Altogether, the tourism company has done about 1.4 billion forecasts — creating them every day using SAS Forecast Server and SAS Grid Computing — which provide recommended values for its inventory. The process takes about four hours.

Previously, the company spent 80 percent of its time pulling data from disparate systems. With SAS, it spends about 95 percent of its time making business decisions on deriving the right value to fit items in the inventory. That change has increased member satisfaction and revenue.

SAS helped the tourism company spot trends and cross- and upselling opportunities. SAS segments customers by pulling data on travel patterns, and then the company used this intelligence to recommend vacations to other members with similar patterns. This merchandising logic is also applied to the tourism company's updated website.

During the first 12 months of operation, the tourism company's SAS Grid program exchange platform generated $11 million in revenue. Compared to recent historical volume trends, this result represents a profitable increase in exchanges:

- Average weekly rates (an internal metric that the tourism company uses to assess the network value generated by the exchange) increased 4.6 percent.
- Occupancy rates, which had been trending downward, rose 3 percent.
- Deposits (where members deposit a week of their timeshare so that the tourism company can offer it to others), which had also been declining, rose 6 percent compared to recent trends.

Developing the solution in-house and leveraging SAS — as opposed to working with a consulting firm — saved the company an additional $10 to $15 million in direct costs over three years.

A hit with executives, customers, and industry

According to the director of analytics, the tourism company's executives are enthusiastic about the system. "Our senior leadership has been raving about the system," he said. "They've told us the system is having a meaningful impact on the business and on customer satisfaction."

"The superior matching and superior operation of the network from using SAS leads to returns in revenues," said the vice president of analytics. "That matching is critical. If you don't make sure that members holding the scarce space get an equitable exchange, you prejudge the ability to maintain future streams of the supply. SAS allows us to have the right information in front of the right people at the right time, driving business decisions and company success."

Chapter 9

8 Things You Should Know about SAS Grid Computing

In This Chapter

- Figuring out what makes SAS Grid Computing so great
- Understanding the ins and outs of SAS Grid Computing

For those of you who can't spend the time reading the whole book, this chapter is for you. We have provided just the highlights of SAS Grid computing, which is a great way to increase productivity in a busy environment with lots of users. The following are eight important points you need to know about SAS Grid.

- **Balance your workload.** As your organization's use of business analytics grows, so does the need to scale for increasingly diverse workloads. High-priority jobs can run immediately, and computing resources are balanced and fully utilized. (Refer to Chapter 3 for more information.)
- **Be ready for anything.** Having multiple servers in a grid environment means that even if a server fails, you're always on and always connected. You can perform maintenance on servers without interrupting jobs and add resources without disrupting the business. (Chapter 4 provides more details.)
- **Divide and conquer.** Now you can get faster results by dividing individual jobs into subtasks that are run in parallel on the best available hardware. Make the most of all your available computing resources to keep IT spending in check. (Check out Chapter 8.)

- **Know what's going on.** With SAS Grid you can let users monitor their work in real time practically anywhere with web-based tools to track progress. Customize the displays to see only what you want with your jobs. Administrators can manage the grid to define roles, resources, and priorities. Alerts let you deal with everyday *and* critical situations. (Chapter 5 gives you more in-depth information.)
- **Flex your infrastructure.** When the pressure is on to produce and you need more computing power, SAS Grid allows you to quickly add extra resources and grow your grid. When the peak is over, you can give it back with no disruptions. (Flip to Chapter 3.)
- **Advantage the enabled.** Use the power of grid computing with SAS Applications that are ready to go. *SAS Scheduling* is directly integrated with SAS Data Integration Studio, SAS Marketing Automation, SAS Marketing Optimization, and SAS Web Report Studio. (Chapter 5 contains more information.)
- **Let the flows do the work.** With SAS Grid you can create job flows that describe all the files and dependencies for the task. Then you can reuse the flows anytime to schedule the work when it's needed. Tools can help analyze and generate grid-ready code. (Head to Chapter 2.)
- **Submit to the grid.** SAS Solutions and SAS user-written programs can be submitted to the grid where it's managed, scheduled, and executed with no intervention needed. After it's in, you can forget about it and reconnect later to retrieve your results. (We discuss this more in Chapter 8.)

Improve your organization's performance with SAS Grid Computing

Today, business is moving faster and faster. If your organization is faced with a computing infrastructure that's quickly becoming obsolete and is consistently overloaded and you're not willing to take the necessary action, then you have serious problems.

SAS® Grid Computing For Dummies, Second Edition, is your go-to resource for understanding in plain English how SAS Grid Computing works. This book explains the need-to-know information about how to prioritize your organization's most important work so you can keep ahead of your competition. By implementing a SAS Grid environment, your organization can better balance and manage the workloads, use its resources to its full scope and capabilities, and increase productivity and overall performance.

- ***Share the load* — spread user jobs across the available resources so you don't experience resource conflicts**
- ***Improve productivity* — better productivity means a better bottom line for the organization**
- ***Be a master scheduler* — use the scheduling feature to keep all projects on deadline**

Open the book and find:

- **How to maximize your resources to their full scope**
- **Get acquainted with important features and benefits**
- **Ways to automate the scheduling and execution of jobs across your computing environment**
- **Strategies for efficiently managing the grid**
- **Know how to restart jobs when doing so is needed**

Go to Dummies.com®
for videos, step-by-step examples, how-to articles, or to shop!

ISBN: 978-1-119-25047-0
Not for resale

For Dummies®
A Wiley Brand

Also available as an e-book