PART I

Grid Computing

In today's incredibly complex world of computational power, very high speed machine processing capabilities, complex data storage methods, next-generation telecommunications, new-generation operating systems and services, and extremely advanced networking services capabilities—we are entering a new era of computing. At the same time, industry, businesses, and home users alike are placing more complex and challenging demands on the networks.

In this book we explore all of these aspects in simple to understand terms as we unveil a new era of computing, simply referred to as "Grid Computing." The worldwide Grid Computing discipline involves the actual connections of a potentially unlimited number of machines within a grid, and can be most simply thought of as a massively large power "utility" grid, such as what provides power to our homes and businesses each and every day.

This part of the book unveils many of these powerful approaches to this new era of computing, and explores why so many are considering a Grid Computing environment as a single, incredibly powerful, and effective computing solution.

CHAPTER 1

Introduction

In today's pervasive world of needing information anytime and anywhere, the explosive Grid Computing environments have now proven to be so significant that they are often referred to as being the world's single and most powerful computer solutions. It has been realized that with the many benefits of Grid Computing, we have consequently introduced both a complicated and complex global environment, which leverages a multitude of open standards and technologies in a wide variety of implementation schemes. As a matter of fact the complexity and dynamic nature of industrial problems in today's world are much more intensive to satisfy by the more traditional, single computational platform approaches.

GRID COMPUTING EQUATES TO THE WORLD'S LARGEST COMPUTER ...

The Grid Computing discipline involves the actual networking services and connections of a potentially unlimited number of ubiquitous computing devices within a "grid." This new innovative approach to computing can be most simply thought of as a massively large power "utility" grid, such as what provides power to our homes and businesses each and every day. This delivery of utility-based power has become second nature to many of us, worldwide. We know that by simply walking into a room and turning on the lights, the power will be directed to the proper devices of our choice for that moment in time. In this same utility fashion, Grid Computing openly seeks and is capable of adding an infinite number of computing devices into any grid environment, adding to the computing capability and problem resolution tasks within the operational grid environment.

The incredible problem resolution capabilities of Grid Computing remain yet unknown, as we continue to forge ahead and enter this new era of massively powerful grid-based problem-solving solutions.

This "Introduction" section of the book will begin to present many of the Grid Computing topics, which are discussed throughout this book. These discussions in Chapter 1 are intended only to provide a rather high-level examination of Grid Computing. Later sections of the book provide a full treatment of the topics addressed by many worldwide communities utilizing and continuing to develop Grid Computing.

The worldwide business demand requiring intense problem-solving capabilities for incredibly complex problems has driven in all global industry segments the need for dynamic collaboration of many ubiquitous computing resources to be able to work together. These difficult computational problem-solving needs have now fostered many complexities in virtually all computing technologies, while driving up costs and operational aspects of the technology environments. However, this advanced computing collaboration capability is indeed required in almost all areas of industrial and business problem solving, ranging from scientific studies to commercial solutions to academic endeavors. It is a difficult challenge across all the technical communities to achieve this level of resource collaboration needed for solving these complex and dynamic problems, within the bounds of the necessary quality requirements of the end user.

To further illustrate this environment and oftentimes very complex set of technology challenges, let us consider some common *use case* scenarios one might have already encountered, which will begin to examine the many values of a Grid Computing solution environment. These simple use cases, for purposes of introduction to the concepts of Grid Computing, are as follows:

- A financial organization processing wealth management application collaborates with the different departments for more computational power and software modeling applications. It pools a number of computing resources, which can thereby perform faster with real-time executions of the tasks and immediate access to complex pools of data storage, all while managing complicated data transfer tasks. This ultimately results in increased customer satisfaction with a faster turnaround time.
- A group of scientists studying the atmospheric ozone layer will collect huge amounts of experimental data, each and every day. These scientists need efficient and complex data storage capabilities across wide and geographically dispersed storage facilities, and they need to access this data in an efficient manner based on the processing needs. This ultimately results in a more effective and efficient means of performing important scientific research.
- Massive online multiplayer game scenarios for a wide community of international gaming participants are occurring that require a large number of gaming computer servers instead of a dedicated game server. This allows international game players to interact among themselves as a group in a real-time manner. This involves the need for on-demand allocation and provisioning of computer resources, provisioning and self-management of complex networks, and complicated data storage resources. This on-demand need is very dynamic, from moment-to-moment, and it is always based upon the workload in the system at any given moment in time. This ultimately results in larger

gaming communities, requiring more complex infrastructures to sustain the traffic loads, delivering more profits to the bottom lines of gaming corporations, and higher degrees of customer satisfaction to the gaming participants.

- A government organization studying a natural disaster such as a chemical spill may need to immediately collaborate with different departments in order to plan for and best manage the disaster. These organizations may need to simulate many computational models related to the spill in order to calculate the spread of the spill, effect of the weather on the spill, or to determine the impact on human health factors. This ultimately results in protection and safety matters being provided for public safety issues, wildlife management and protection issues, and ecosystem protection matters: Needles to say all of which are very key concerns.

Today, Grid Computing offers many solutions that already address and resolve the above problems. Grid Computing solutions are constructed using a variety of technologies and open standards. Grid Computing, in turn, provides highly scalable, highly secure, and extremely high-performance mechanisms for discovering and negotiating access to remote computing resources in a seamless manner. This makes it possible for the sharing of computing resources, on an unprecedented scale, among an infinite number of geographically distributed groups. This serves as a significant transformation agent for individual and corporate implementations surrounding computing practices, toward a general-purpose utility approach very similar in concept to providing electricity or water. These electrical and water types of utilities, much like Grid Computing utilities, are available "on demand," and will always be capable of providing an always-available facility negotiated for individual or corporate utilization.

In this new and intriguing book, we will begin our discussion on the core concepts of the Grid Computing system with an early definition of grid. Back in 1998, it was defined, "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities" (Foster & Kesselman, 1998).

The preceding definition is more centered on the computational aspects of Grid Computing while later iterations broaden this definition with more focus on coordinated resource sharing and problem solving in multi-institutional virtual organizations (Foster & Kesselman, 1998). In addition to these qualifications of coordinated resource sharing and the formation of dynamic virtual organizations, open standards become a key underpinning. It is important that there are open standards throughout the grid implementation, which also accommodate a variety of other open standards-based protocols and frameworks, in order to provide interoperable and extensible infrastructure environments.

Grid Computing environments must be constructed upon the following foundations:

- *Coordinated resources*. We should avoid building grid systems with a centralized control; instead, we must provide the necessary infrastructure for coordination among the resources, based on respective policies and service-level agreements.

- *Open standard protocols and frameworks*. The use of open standards provides interoperability and integration facilities. These standards must be applied for resource discovery, resource access, and resource coordination.

Another basic requirement of a Grid Computing system is the ability to provide the quality of service (QoS) requirements necessary for the end-user community. These QoS validations must be a basic feature in any Grid system, and must be done in congruence with the available resource matrices. These QoS features can be (for example) response time measures, aggregated performance, security fulfillment, resource scalability, availability, autonomic features such as event correlation and configuration management, and partial fail over mechanisms.

There have been a number of activities addressing the above definitions of Grid Computing and the requirements for a grid system. The most notable effort is in the standardization of the interfaces and protocols for the Grid Computing infrastructure implementations. We will cover the details later in this book. Let us now explore some early and current Grid Computing systems and their differences in terms of benefits.

Early Grid Activities

Over the past several years, there has been a lot of interest in computational Grid Computing worldwide. We also note a number of derivatives of Grid Computing, including compute grids, data grids, science grids, access grids, knowledge grids, cluster grids, terra grids, and commodity grids. As we explore careful examination of these grids, we can see that they all share some form of resources; however, these grids may have differing architectures.

One key value of a grid, whether it is a commodity utility grid or a computational grid, is often evaluated based on its business merits and the respective user satisfaction. User satisfaction is measured based on the QoS provided by the grid, such as the availability, performance, simplicity of access, management aspects, business values, and flexibility in pricing. The business merits most often relate to and indicate the problem being solved by the grid. For instance, it can be job executions, management aspects, simulation workflows, and other key technology-based foundations.

Earlier Grid Computing efforts were aligned with the overlapping functional areas of data, computation, and their respective access mechanisms. Let us further explore the details of these areas to better understand their utilization and functional requirements.

Data

The data aspects of any Grid Computing environment must be able to effectively manage all aspects of data, including data location, data transfer, data access, and critical aspects of security. The core functional data requirements for Grid Computing applications are:

- The ability to integrate multiple distributed, heterogeneous, and independently managed data sources.
- The ability to provide efficient data transfer mechanisms and to provide data where the computation will take place for better scalability and efficiency.
- The ability to provide data caching and/or replication mechanisms to minimize network traffic.
- The ability to provide necessary data discovery mechanisms, which allow the user to find data based on characteristics of the data.
- The capability to implement data encryption and integrity checks to ensure that data is transported across the network in a secure fashion.
- The ability to provide the backup/restore mechanisms and policies necessary to prevent data loss and minimize unplanned downtime across the grid.

Computation

The core functional computational requirements for grid applications are:

- The ability to allow for independent management of computing resources
- The ability to provide mechanisms that can intelligently and transparently select computing resources capable of running a user's job
- The understanding of the current and predicted loads on grid resources, resource availability, dynamic resource configuration, and provisioning
- Failure detection and failover mechanisms
- Ensure appropriate security mechanisms for secure resource management, access, and integrity

Let us further explore some details on the computational and data grids as they exist today.

Computational and Data Grids

In today's complex world of high speed computing, computers have become extremely powerful as to that of (let's say) five years ago. Even the home-based PCs available on the commercial markets are powerful enough for accomplishing complex computations that we could not have imagined a decade prior to today.

The quality and quantity requirements for some business-related advanced computing applications are also becoming more and more complex. The industry is now realizing that we have a need, and are conducting numerous complex scientific experiments, advanced modeling scenarios, genome matching, astronomical research, a wide variety of simulations, complex scientific/business modeling scenarios, and real-time personal portfolio management. These requirements can actually exceed the demands and availability of installed computational power within an organization. Sometimes, we find that no single organization alone satisfies some of these aforementioned computational requirements.

This advanced computing power applications need is indeed analogous to the electric power need in the early 1900s, such that to provide for the availability of electrical power, each user has to build and be prepared to operate an electrical generator. Thus, when the electric power grid became a reality, this changed the entire concept of the providing for, and utilization of, electrical power. This, in turn, paved the way for an evolution related to the utilization of electricity. In a similar fashion, the computational grids change the perception on the utility and availability of the computer power. Thus the computational Grid Computing environment became a reality, which provides a demand-driven, reliable, powerful, and yet inexpensive computational power for its customers.

As we noted earlier in this discussion, a computational Grid Computing environment consists of one or more hardware- and software-enabled environments that provide dependable, consistent, pervasive and inexpensive access to high-end computational capabilities (Foster & Kesselman, 1998).

Later in this book, in the "Grid Anatomy" section, we will see that this definition has evolved to give more emphasis on the seamless resource sharing aspects in a collaborative virtual organizational world. But the concept still holds for a computational grid where the sharable resource remains a computing power. As of now, the majority of the computational grids are centered on major scientific experiments and collaborative environments.

The requirement for key data forms a core underpinning of any Grid Computing environment. For example, in data-intensive grids, the focus is on the management of data, which is being held in a variety of data storage facilities in geographically dispersed locations. These data sources can be databases, file systems, and storage devices. The grid systems must also be capable of providing data virtualization services to provide transparency for data access, integration, and processing. In addition to the above requirements, security and privacy requirements of all respective data in a grid system is quite complex.

We can summarize the data requirements in the early grid solutions as follows:

- The ability to discover data
- The access to databases, utilizing meta-data and other attributes of the data
- The provisioning of computing facilities for high-speed data movement
- The capability to support flexible data access and data filtering capabilities

As one begins to realize the importance of extreme high performance-related issues in a Grid Computing environment, it is recommended to store (or cache) data near to the computation, and to provide a common interface for data access and management.

It is interesting to note that upon careful examination of existing Grid Computing systems, readers will learn that many Grid Computing systems are being applied in several important scientific research and collaboration projects; however, this does not preclude the importance of Grid Computing in business-, academic-, and industry-related fields. The commercialization of Grid Computing invites and addresses a key architectural alignment with several existing commercial frameworks for improved interoperability and integration.

As we will describe in this book, many current trends in Grid Computing are toward service-based architectures for grid environments. This "architecture" is built for interoperability and is (again) based upon open standard protocols. We will provide a full treatment including many of the details toward this architecture throughout subsequent sections in this book.

CURRENT GRID ACTIVITIES

As described earlier, initially, the focused Grid Computing activities were in the areas of computing power, data access, and storage resources.

The definition of Grid Computing resource sharing has since changed, based upon experiences, with more focus now being applied to a sophisticated form of coordinated resource sharing distributed throughout the participants in a virtual organization. This application concept of coordinated resource sharing includes any resources available within a virtual organization, including computing power, data, hardware, software and applications, networking services, and any other forms of computing resource attainment. This concept of coordinated resource sharing is depicted in Figure 1.1.

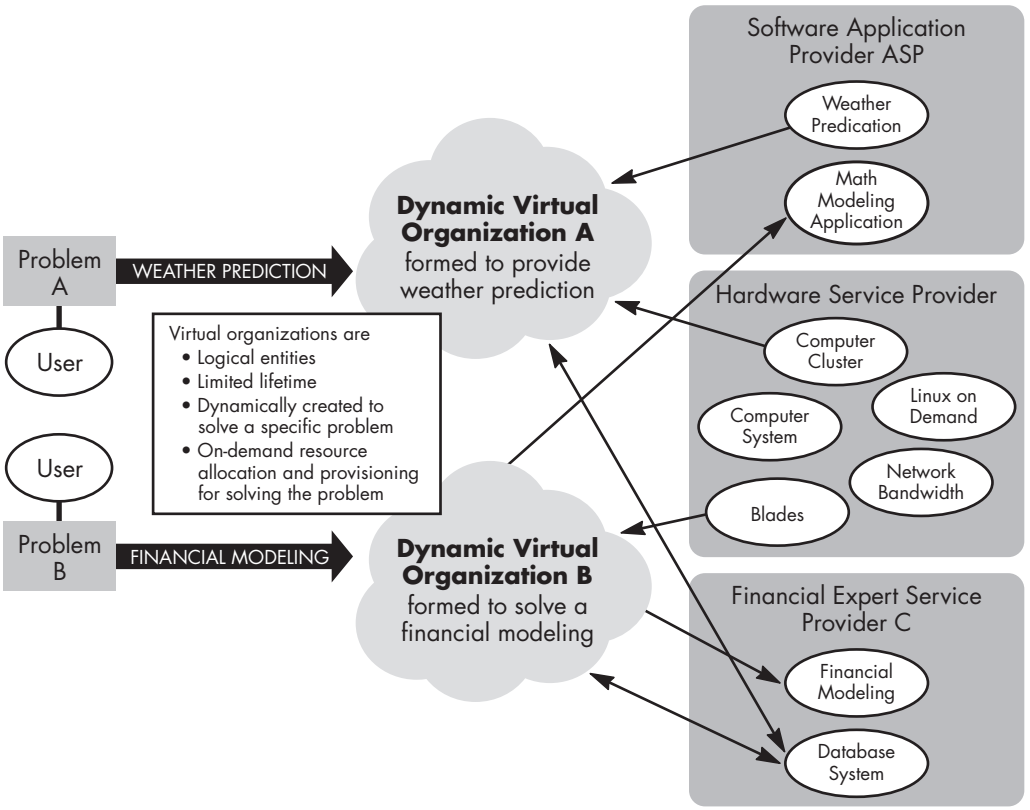

Figure 1.1
Dynamic benefits of coordinated resource sharing in a virtual organization.

As depicted in the previous illustration, there are a number of sharable resources, hardware and software applications, firmware implementations, and networking services, all available within an enterprise or service provider environment. Rather than keeping these resources isolated within an atomic organization, the users can acquire these resources on a "demand" basis. Through implementing this type of Grid Computing environment, these resources are immediately available to the authenticated users for resolving specific problems. These problems may be a software capability problem (e.g., modeling, simulation, word processing, etc.) or hardware availability and/or computing capacity shortage problems (e.g., processor computing resources, data storage/access needs, etc.). While on another level, these problems may be related to a networking bandwidth availability problem, the need for immediate circuit provisioning of a network, a security event or other event correlation issue, and many more types of critical environmental needs.

Based upon the specific problem dimension, any given problem may have one or more resolution issues to address. For example, in the above case there is two sets of users, each with a need to solve two different types of problems. You will note that one has to resolve the weather prediction problem, while the other has to provide a financial modeling case. Based upon these problem domains noted by each of the user groups, their requirements imply two types of virtual organizations. These distinct virtual organizations are formulated, sustained, and managed from a computing resource viewpoint according to the ability to access the available resources. Let us further explore this concept of "*virtualization*" by describing in more detail the usage patterns found within each of the virtual organizations.

- *A virtual organization for weather prediction.* For example, this virtual organization requires resources such as weather prediction software applications to perform the mandatory environmental simulations associated with predicting weather. Likewise, they will require very specific hardware resources to run the respective software, as well as high-speed data storage facilities to maintain the data generated from performing the simulations.
- *A virtual organization for financial modeling.* For example, this virtual organization requires resources such as software modeling tools for performing a multitude of financial analytics, virtualized blades[1] to run the above software, and access to data storage facilities for storing and accessing data.

These virtual organizations manage their resources and typically will provision additional resources on an "as-needed" basis. This on-demand approach provides tremendous values toward scalability, in addition to aspects of enhanced reusability. This approach is typically found in any "on-demand" environment. This capability is based upon a *utility* infrastructure, where resources are allocated as, and when, they are required. Likewise, their utility pricing scenarios are always based upon the capturing of usage metrics.

The following discussion introduces a number of requirements needed for such Grid Computing architectures utilized by virtual organizations. We shall classify these architecture requirements

into three categories. These resources categories must be capable of providing facilities for the following scenarios:

- The need for dynamic discovery of computing resources, based on their capabilities and functions.
- The immediate allocation and provisioning of these resources, based on their availability and the user demands or requirements.
- The management of these resources to meet the required service level agreements (SLAs).
- The provisioning of multiple autonomic features for the resources, such as self-diagnosis, self-healing, self-configuring, and self-management.
- The provisioning of secure access methods to the resources, and bindings with the local security mechanisms based upon the autonomic control policies.

Virtual organization must be capable of providing facilities for:

- The formation of virtual task forces, or groups, to solve specific problems associated with the virtual organization.
- The dynamic collection of resources from heterogeneous providers based upon users' needs and the sophistication levels of the problems.
- The dynamic identification and automatic problem resolution of a wide variety of troubles, with automation of event correlation, linking the specific problems to the required resource and/or service providers.
- The dynamic provisioning and management capabilities of the resources required meeting the SLAs.
- The formation of a secured federation (or governance model) and common management model for all of the resources respective to the virtual organization.
- The secure delegation of user credentials and identity mapping to the local domain(s).
- The management of resources, including utilization and allocation, to meet a budget and other economic criteria.

Users/applications typically found in Grid Computing environments must be able to perform the following characteristics:

- The clear and unambiguous identification of the problem(s) needing to be solved
- The identification and mapping of the resources required solve the problem
- The ability to sustain the required levels of QoS, while adhering to the anticipated and necessary SLAs
- The capability to collect feedback regarding resource status, including updates for the environment's respective applications

The above discussion helps us now to better understand the common requirements for grid systems. In the subsequent chapters in this section, and moreover throughout this book, we discuss the many specific details on the Grid Computing architecture models and emerging Grid Computing software systems that have proven valuable in supporting the above requirements.

The following section will provide treatment toward some of the more common Grid Computing business areas that exist today, and those areas that will typically benefit from the above concepts of Grid Computing. It is worthy to mention that these business areas are most often broadly classified, and based upon the industry sector where they reside.

An Overview of Grid Business Areas

One of the most valuable aspects of all Grid Computing systems are that they attract the business they are intended to address. In an "on-demand" scenario, these Grid Computing environments are the result of autonomic provisioning of a multitude of resources and capabilities, typically demonstrating increased computing resource utilization, access to specialized computer systems, cost sharing, and improved management capabilities.

IBM Business On Demand Initiative

Business On Demand (in the rest of the book we will refer to this as On Demand) is not just about utility computing as it has a much broader set of ideas about the transformation of business practices, process transformation, and technology implementations. Companies striving to achieve the Business On Demand operational models will have the capacity to sense and respond to fluctuating market conditions in real-time, while providing products and services to customers in a Business On Demand operational model. The essential characteristics of on-demand businesses are responsiveness to the dynamics of business, adapting to variable cost structures, focusing on core business competency, and resiliency for consistent availability. This is achieved through seamless integration of customers and partners, virtualization of resources, autonomic/dependable resources, and open standards.

There have been a significant number of commercialization efforts, which support Grid Computing in every sector of the marketplace. In general terms, the utilization of Grid Computing in business environments provides a rich and extensible set of business benefits. These business benefits include (but are not limited to):

- Acceleration of implementation time frames in order to intersect with the anticipated business end results.
- Improved productivity and collaboration of virtual organizations and respective computing and data resources.
- Allowing widely dispersed departments and businesses to create virtual organizations to share data and resources.
- Robust and infinitely flexible and resilient operational infrastructures.
- Providing instantaneous access to massive computing and data resources.

- Leveraging existing capital expenditures investments, and operational expenditure investments, which in turn help to ensure optimal utilization and costs of computing capabilities.
- Avoiding common pitfalls of overprovisioning and incurring excess costs.

Many organizations have started identifying the major business areas for Grid Computing business applications. Some examples of major business areas include (but are not limited to):

- Life sciences, for analyzing and decoding strings of biological and chemical information
- Financial services, for running long, complex financial models and arriving at more accurate decisions
- Higher education for enabling advanced, data- and computation-intensive research
- Engineering services, including automotive and aerospace, for collaborative design and data-intensive testing
- Government, for enabling seamless collaboration and agility in both civil and military departments and other agencies
- Collaborative games for replacing the existing single-server online games with more highly parallel, massively multiplayer online games

Let us now introduce and explore the analytics of each of these industry sectors by identifying some of the high-level business-area requirements for Grid Computing systems. In doing so, we will look at the facilities necessary for grid systems in order to meet these requirements.

Life Sciences

This industry sector has noted many dramatic advances in the life sciences sector, which have in turn provided rapid changes in the way that drug treatment and drug discovery efforts are now being conducted. The analytics and system efforts' surrounding genomic, proteomics, and molecular biology efforts provides the basis for many of these Grid Computing advancements in this sector. These advances have now presented a number of technical challenges to the information technology sector, and especially the Grid Computing disciplines.

Grid Computing efforts have realized that these challenges include huge amounts of data analysis, data movement, data caching, and data mining. In addition to the complexity of processing data, there needs to be additional requirements surrounding data security, secure data access, secure storage, privacy, and highly flexible integration. Another area that requires attention is the querying of nonstandard data formats and accessing data assets across complex global networks.

The above requirements presented by life sciences require a Grid Computing infrastructure to properly manage data storage, providing access to the data, and all while performing complex analysis respective to the data. The Grid Computing systems can provide a common infrastructure for data access, and at the same time, provide secure data access mechanisms while processing the data. Today, life sciences utilizes the Grid Computing systems to execute sequence comparison algorithms and enable molecular modeling using the above-collected secured data.

This now provides the Life Sciences sector the ability to afford world-class information analysis respective to this discussion, while at the same time providing faster response times and far more accurate results.

Financial Analysis and Services

This industry sector has noted many dramatic advances in the financial analysis and services industry sector. The technological and business advances are most noted in the information technology areas, the emergence of a competitive market force customer satisfaction, and reduction of risk as the most competitive areas financial communities continually strive to achieve. The requirements related to sophistication, accuracy, and faster execution are among the more salient objectives across financial communities. These objectives are now achieved by real-time access to the current and historical market data, complex financial modeling based on the respective data, and faster response times to user queries.

Grid Computing provides the financial analysis and services industry sector with advanced systems delivering all the competitive solutions in Grid Computing. These solutions exemplify the infrastructure and business agility necessary to meet and exceed the uniqueness that the financial analysis and services industry sector requires. This particular value statement is accomplished by the fact that many of these solutions in this industry are dependent upon providing increased access to massive amounts of data, real-time modeling, and faster execution by using the grid job scheduling and data access features. For this to be most successful, these financial institutions tend to form virtual organizations with participation from several different departments and other external organizations. In addition to the use of existing resources, a grid system can provide more efficiency by easily adapting to the rapidly changing algorithms pertaining to the financial analytics.

Research Collaboration

Research-oriented organizations and universities practicing in advanced research collaboration areas require the analysis of tremendous amounts of data. Some examples of such projects are subatomic particle and high energy physics experiments, remote sensing sources for earth simulation and modeling, and analysis of the human genome sequence.

These virtual organizations engaged in research collaboration activities generate petabytes[2] of data and require tremendous amounts of storage space and thousands of computing processors. Researchers in these fields must share data, computational processors, and hardware instrumentation such as telescopes and advanced testing equipment. Most of these resources are pertaining to data-intensive processing, and are widely dispersed over a large geographical area.

The Grid Computing discipline provides mechanisms for resource sharing by forming one or more virtual organizations providing specific sharing capabilities. Such virtual organizations are constituted to resolve specific research problems with a wide range of participants from different

regions of the world. This formation of dynamic virtual organizations provides capabilities to dynamically add and delete virtual organization participants, manage the "on-demand" sharing of resources, plus provisioning of a common and integrated secure framework for data interchange and access.

Engineering and Design

The enormous competitive pressure in the business and industry sectors today afford most engineering and design far less turnaround time. They need mechanisms to capture data, speed up the analysis on the data, and provide faster responses to market needs. As we already know, these engineering activities and design solutions are inherently complex across several dimensions, and the processing requirements are much more intense than that of traditional solutions of the past.

These complexities fall into several areas of solutions in Grid Computing that span across industry sectors all over the world. These complexities are described (but are not limited to) the following areas:

- The analysis of real-time data to find a specific pattern within a problem
- The parametric studies to verify different aspects of the systems
- The modeling experiments to create new designs
- The simulation activities to verify the existing models for accuracy

Grid Computing systems provide a wide range of capabilities that address the above kinds of analysis and modeling activities. These advanced types of solutions also provide complex job schedulers and resource managers to deal with computing power requirements. This enables automobile manufacturers (as an example) to shorten analysis and design times, all while minimizing both capital expenditures and operational expenditures.

Collaborative Games

There are collaborative types of Grid Computing disciplines that are involving emerging technologies to support online games, while utilizing on-demand provisioning of computation-intensive resources, such as computers and storage networks. These resources are selected based on the requirements, often involving aspects such as volume of traffic and number of players, rather than centralized servers and other fixed resources.

These on-demand-driven games provide a flexible approach with a reduced up-front cost on hardware and software resources. We can imagine that these games use an increasing number of computing resources with an increase in the number of concurrent players and a decrease in resource usage with a lesser number of players. Grid Computing gaming environments are capable of supporting such virtualized environments for enabling collaborative gaming.

Government

The Grid Computing environments in government focus on providing coordinated access to massive amounts of data held across various agencies in a government. This provides faster access to solve critical problems, such as emergency situations, and other normal activities. These key environments provide more efficient decision making with less turnaround time.

Grid Computing enables the creation of virtual organizations, including many participants from various governmental agencies (e.g., state and federal, local or country, etc.). This is necessary in order to provide the data needed for government functions, in a real-time manner, while performing the analysis on the data to detect the solution aspects of the specific problems being addressed. The formation of virtual organizations, and the respective elements of security, is most challenging due to the high levels of security in government and the very complex requirements.

Grid Applications

Based on our earlier discussion, we can align Grid Computing applications to have common needs, such as what is described in (but not limited to) the following items:

- Application partitioning that involves breaking the problem into discrete pieces
- Discovery and scheduling of tasks and workflow
- Data communications distributing the problem data where and when it is required
- Provisioning and distributing application codes to specific system nodes
- Results management assisting in the decision processes of the environment
- Autonomic features such as self-configuration, self-optimization, self-recovery, and self-management

Let us now explore some of these Grid applications and their usage patterns. We start with schedulers, which form the core component in most of the computational grids.

Schedulers

Schedulers are types of applications responsible for the management of jobs, such as allocating resources needed for any specific job, partitioning of jobs to schedule parallel execution of tasks, data management, event correlation, and service-level management capabilities. These schedulers then form a hierarchical structure, with meta-schedulers that form the root and other lower level schedulers, while providing specific scheduling capabilities that form the leaves. These schedulers may be constructed with a local scheduler implementation approach for specific job execution, or another meta-scheduler or a cluster scheduler for parallel executions. Figure 1.2 shows this concept.

The jobs submitted to Grid Computing schedulers are evaluated based on their service-level requirements, and then allocated to the respective resources for execution. This will involve

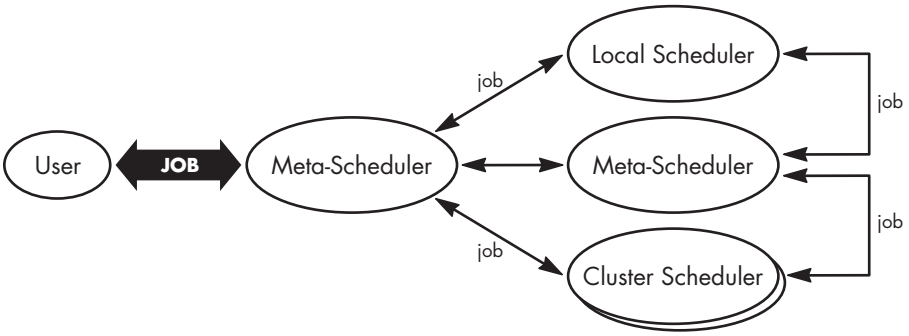

Figure 1.2
The scheduler hierarchy embodies local, meta-level, and cluster schedulers.

complex workflow management and data movement activities to occur on a regular basis. There are schedulers that must provide capabilities for areas such as (but not limited to):

- Advanced resource reservation
- Service-level agreement validation and enforcement
- Job and resource policy management and enforcement for best turnaround times within the allowable budget constraints
- Monitoring job executions and status
- Rescheduling and corrective actions of partial failover situations

Later in this book, full treatment is provided for many of the most notable scheduler and meta-scheduler implementations.

Resource Broker

The resource broker provides *pairing* services between the service requester and the service provider. This pairing enables the selection of best available resources from the service provider for the execution of a specific task. These resource brokers collect information (e.g., resource availability, usage models, capabilities, and pricing information) from the respective resources, and use this information source in the pairing process.

Figure 1.3 illustrates the use of a resource broker for purposes of this discussion. This particular resource broker provides feedback to the users on the available resources. In general cases, the resource broker may select the suitable scheduler for the resource execution task, and collaborate with the scheduler to execute the task(s).

The pairing process in a resource broker involves allocation and support functions such as:

- Allocating the appropriate resource or a combination of resources for the task execution
- Supporting users' deadline and budget constraints for scheduling optimizations

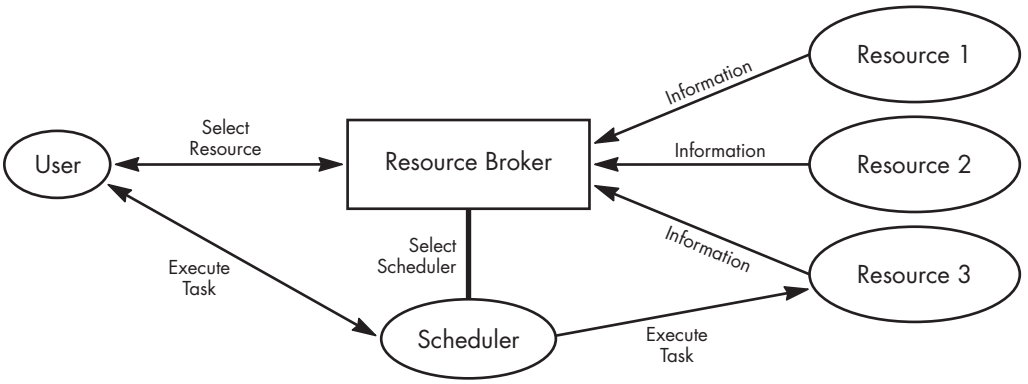

Figure 1.3
The resource broker collects information from the respective resources, and utilizes this information source in the pairing process.

Load Balancing

The Grid Computing infrastructure load-balancing issues are concerned with the traditional load-balancing distribution of workload among the resources in a Grid Computing environment. This load-balancing feature must always be integrated into any system in order to avoid processing delays and overcommitment of resources. These kinds of applications can be built in connection with schedulers and resource managers.

The workload can be pushed outbound to the resources, based on the availability state and/or resources, and can then pull the jobs from the schedulers depending on their availability. This level of load balancing involves partitioning of jobs, identifying the resources, and queueing of the jobs. There are cases when resource reservations might be required, as well as running multiple jobs in parallel.

Another feature that might be of interest for load balancing is support for failure detection and management. These load distributors can redistribute the jobs to other resources if needed.

Grid Portals

Grid portals are similar to Web portals, in the sense they provide uniform access to the grid resources. For example, grid portals provide capabilities for Grid Computing resource authentication, remote resource access, scheduling capabilities, and monitoring status information. These kinds of portals help to alleviate the complexity of task management through customizable and personalized graphical interfaces for the users. This, in turn, alleviates the need for end users to have more domain knowledge than on the specific details of grid resource management.

Some examples of these grid portal capabilities are noted in the following list:

- Querying databases or LDAP servers for resource-specific information

- File transfer facilities such as file upload, download, integration with custom software, and so on
- Manage job through job status feedbacks
- Allocate the resources for the execution of specific tasks
- Security management
- Provide personalized solutions

In short, these grid portals help free end users from the complexity of job management and resource allocation so they can concentrate more on their domain of expertise. There are a number of standards and software development toolkits available to develop custom portals. The emerging Web services and Web service portal standards will play a more significant role in portal development.

Integrated Solutions

Many of the global industry sectors have witnessed the emergence of a number of integrated grid application solutions in the last few years. This book focuses on this success factor.

These integrated solutions are a combination of the existing advanced middleware and application functionalities, combined to provide more coherent and high performance results across the Grid Computing environment.

Integrated Grid Computing solutions will have more enhanced features to support more complex utilization of grids such as coordinated and optimized resource sharing, enhanced security management, cost optimizations, and areas yet to be explored. It is straightforward to see that these integrated solutions in both the commercial and noncommercial worlds sustain high values and significant cost reductions. Grid applications can achieve levels of flexibility utilizing infrastructures provided by application and middleware frameworks.

In the next section we introduce and explain the grid infrastructure. Today, the most notable integrated solutions in the commercial and industry sectors are utility computing, on-demand solutions, and resource virtualizations infrastructures. Let us briefly explore aspects of some of these infrastructure solutions. We will provide an additional, more focused treatment in subsequent chapters of this book.

Grid Infrastructure

The grid infrastructure forms the core foundation for successful grid applications. This infrastructure is a complex combination of a number of capabilities and resources identified for the specific problem and environment being addressed.

In initial stages of delivering any Grid Computing application infrastructure, the developers/service providers must consider the following questions in order to identify the core infrastructure support required for that environment:

1. What problem(s) are we trying to solve for the user? How do we address grid enablement simpler, while addressing the user's application simpler? How does the developer (programmatically) help the user to be able to quickly gain access and utilize the application to best fit their problem resolution needs?
2. How difficult is it to use the grid tool? Are grid developers providing a flexible environment for the intended user community?
3. Is there anything not yet considered that would make it easier for grid service providers to create tools for the grid, suitable for the problem domain?
4. What are the open standards, environments, and regulations grid service providers must address?

In the early development stages of grid applications, numerous vertical "towers" and middleware solutions were often developed to solve Grid Computing problems. These various middleware and solution approaches were developed for fairly narrow and limited problem-solving domains, such as middleware to deal with numerical analysis, customized data access grids, and other narrow problems. Today, with the emergence and convergence of grid service-oriented technologies,[3] including the interoperable XML[4]-based solutions becoming ever more present and industry providers with a number of reusable grid middleware solutions facilitating the following requirement areas, it is becoming simpler to quickly deploy valuable solutions. Figure 1.4 shows this topology of middleware topics.

In general, a Grid Computing infrastructure component must address several potentially complicated areas in many stages of the implementation. These areas are:

- Security
- Resource management
- Information services
- Data management

Let us further examine the significance of each of these above components.

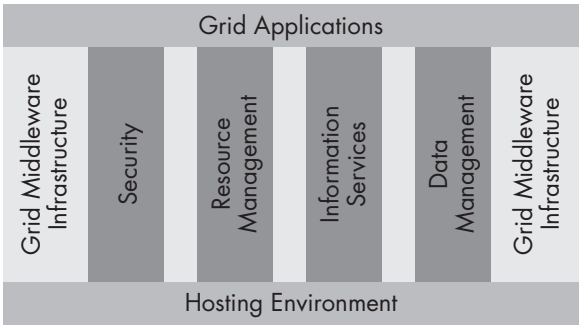

Figure 1.4
Grid middleware topic areas are becoming more sophisticated at an aggressive rate.

Security

The heterogeneous nature of resources and their differing security policies are complicated and complex in the security schemes of a Grid Computing environment. These computing resources are hosted in differing security domains and heterogeneous platforms. Simply speaking, our middleware solutions must address local security integration, secure identity mapping, secure access/authentication, secure federation, and trust management.

The other security requirements are often centered on the topics of data integrity, confidentiality, and information privacy. The Grid Computing data exchange must be protected using secure communication channels, including SSL/TLS and oftentimes in combination with secure message exchange mechanisms such as WS-Security. The most notable security infrastructure used for securing grid is the Grid Security Infrastructure (GSI). In most cases, GSI provides capabilities for single sign-on, heterogeneous platform integration and secure resource access/authentication.

The latest and most notable security solution is the use of WS-Security standards. This mechanism provides message-level, end-to-end security needed for complex and interoperable secure solutions. In the coming years we will see a number of secure grid environments using a combination of GSI and WS-Security mechanisms for secure message exchanges. We will discuss the details of security mechanisms provided by these standards later in this book.

Resource Management

The tremendously large number and the heterogeneous potential of Grid Computing resources causes the resource management challenge to be a significant effort topic in Grid Computing environments. These resource management scenarios often include resource discovery, resource inventories, fault isolation, resource provisioning, resource monitoring, a variety of autonomic capabilities,[5] and service-level management activities. The most interesting aspect of the resource management area is the selection of the correct resource from the grid resource pool, based on the service-level requirements, and then to efficiently provision them to facilitate user needs.

Let us explore an example of a job management system, where the resource management feature identifies the job, allocates the suitable resources for the execution of the job, partitions the job if necessary, and provides feedback to the user on job status. This job scheduling process includes moving the data needed for various computations to the appropriate Grid Computing resources, and mechanisms for dispatching the job results.

It is important to understand multiple service providers can host Grid Computing resources across many domains, such as security, management, networking services, and application functionalities. Operational and application resources may also be hosted on different hardware and software platforms. In addition to this complexity, Grid Computing middleware must provide efficient monitoring of resources to collect the required matrices on utilization, availability, and other information.

One causal impact of this fact is (as an example) the security and the ability for the grid service provider to reach out and probe into other service provider domains in order to obtain and reason about key operational information (i.e., to reach across a service provider environment to ascertain firewall and router volume-related specifics, or networking switch status, or application server status). This oftentimes becomes complicated across several dimensions, and has to be resolved by a meeting-of-the-minds between all service providers, such as messaging necessary information to all providers, when and where it is required.

Another valuable and very critical feature across the Grid Computing infrastructure is found in the area of provisioning; that is, to provide autonomic capabilities for self-management, self-diagnosis, self-healing, and self-configuring. The most notable resource management middleware solution is the Grid Resource Allocation Manager (GRAM). This resource provides a robust job management service for users, which includes job allocation, status management, data distribution, and start/stop jobs.

Information Services

Information services are fundamentally concentrated on providing valuable information respective to the Grid Computing infrastructure resources. These services leverage and entirely depend on the providers of information such as resource availability, capacity, and utilization, just to name a few. This information is valuable and mandatory feedback respective to the resources managers discussed earlier in this chapter. These information services enable service providers to most efficiently allocate resources for the variety of very specific tasks related to the Grid Computing infrastructure solution.

In addition, developers and providers can also construct grid solutions to reflect portals, and utilize meta-schedulers and meta-resource managers. These metrics are helpful in service-level management (SLA) in conjunction with the resource policies. This information is resource specific and is provided based on the schema pertaining to that resource. We may need higher level indexing services or data aggregators and transformers to convert these resource-specific data into valuable information sources for the end user.

For example, a resource may provide operating system information, while yet another resource might provide information on hardware configuration, and we can then group this resource information, reason with it, and then suggest a “best” price combination on selecting the operating system on other certain hardware. This combinatorial approach to reasoning is very straightforward in a Grid Computing infrastructure, simply due to the fact that all key resources are shared, as is the information correlated respective to the resources.

Data Management

Data forms the single most important asset in a Grid Computing system. This data may be input into the resource, and the results from the resource on the execution of a specific task. If the infrastructure is not designed properly, the data movement in a geographically distributed system

can quickly cause scalability problems. It is well understood that the data must be near to the computation where it is used. This data movement in any Grid Computing environment requires absolutely secure data transfers, both to and from the respective resources. The current advances surrounding data management are tightly focusing on virtualized data storage mechanisms, such as storage area networks (SAN), network file systems, dedicated storage servers, and virtual databases. These virtualization mechanisms in data storage solutions and common access mechanisms (e.g., relational SQLs, Web services, etc.) help developers and providers to design data management concepts into the Grid Computing infrastructure with much more flexibility than traditional approaches.

Some of the considerations developers and providers must factor into decisions are related to selecting the most appropriate data management mechanism for Grid Computing infrastructures. This includes the size of the data repositories, resource geographical distribution, security requirements, schemes for replication and caching facilities, and the underlying technologies utilized for storage and data access.

So far in this introductory chapter we have been discussing the details surrounding many aspects of the middleware framework requirements, specifically the emergence of service provider-oriented architectures[6] and, hence, the open and extremely powerful utility value of XML-based interoperable messages. These combined, provide a wide range of capabilities that deal with interoperability problems, and come up with a solution that is suitable for the dynamic virtual organizational grids. The most important activity noted today in this area is the Open Grid Service Architecture (OGSA) and its surrounding standard initiatives. Significant detail is recorded on this architecture, and will be given full treatment in subsequent chapters in this book. The OGSA provides a common interface solution to grid services, and all the information has been conveniently encoded using XML as the standard. This provides a common approach to information services and resource management for Grid Computing infrastructures.

This introductory chapter has discussed many of the chapters and some of their detail that will be presented throughout this book. This introductory discussion has been presented at a high level, and more detailed discussions with simple-to-understand graphics will follow.

Conclusion

So far we have been describing and walking through overview discussion topics on the Grid Computing discipline that will be discussed further throughout this book, including the Grid Computing evolution, the applications, and the infrastructure requirements for any grid environment.

In addition to this, we have discussed when one should use Grid Computing disciplines, and the factors developers and providers must consider in the implementation phases. With this introduction we can now explore deeper into the various aspects of a Grid Computing system, its evolution across the industries, and the current architectural efforts underway throughout the world.

The proceeding chapters in this book introduce the reader to this new, evolutionary era of Grid Computing, in a concise, hard-hitting, and easy-to-understand manner.

NOTES

1. The term "blades" refers to a smaller circuit inserted into a larger machine footprint, with many other blades, where each blade is performing, as it's own distinct computer. This notion is commonly referred to as "blade computing."
2. A *petabyte* is a term that indicates a unit of computer memory or data storage capacity equal to 1,024 terabytes (or 2^{50} bytes): one quadrillion bytes.
3. Please refer to the book *Business On Demand: Technology and Strategy Perspectives* (Fellenstein, 2004) for further details and precision on important technologies, Grid Computing, and key strategy perspectives.
4. XML (Extensible Markup Language) is a meta-language written in SGML (Standardized Markup Language) that allows one to design a markup language used to allow for the easy interchange of documents and data across the World Wide Web.
5. Please refer to Fellenstein (2004) for further details and precision on important technologies, Grid Computing, and key strategy perspectives.
6. Please refer to Fellenstein (2004) for further details and precision on important service provider technologies, Grid Computing, and key strategy perspectives on both topics.