Preface

Many today’s applications follow the distributed computing paradigm in which parts of the application are executed on different network-interconnected computers. Examples include Web browsing and searching, Internet banking, enterprise applications (for accounting, production scheduling, customer information management), and grid applications (for data intensive or compute intensive processing). The extension of these applications in terms of number of users or size led to the unprecedented increase of the scale of the infrastructure that supports them in terms of number and geographic dispersion of resources, number of services supported, number of administrative organizations involved, etc. The large scale distributed computing, which encompasses the concepts, models, patterns, technologies, systems, platforms, and applications, is the subject of this book.

Several motivations for using distributed systems exist: reducing the program execution time, increasing systems’ reliability and fault tolerance, realizing the functional specialization, and exploiting the inherent parallelism of applications. There are some other advantages of using distributed systems such as price/performance ratio (it is cheaper to share common resources than buy equipment and software for exclusive use), easier user access to remote resources (the network supports the interconnection of users and resources), incremental growth (permits adding to existing infrastructure rather than completely replacing the existing resources with more powerful ones).

Designers face several challenges when conceiving and developing distributed systems. Most of them are related to what is called transparency, which means “make the distributed system appear as a single computer” (Tanenbaum & van Steen, 2007) to the user or to the application developer. Transparency has several facets that have been approached by specialists, and solutions have been proposed for hiding the resource replication and location, concurrent access to resources by several users, resource failure, resource migration, and others. Another challenge is the openness, which means the interoperation of components and systems by respecting the same set of rules. This allows easily extending the system with new services or moving the applications from one system to another without major modifications. Security is another important issue, which is related to the preservation of confidentiality, integrity and availability of the distributed systems and of their components. Finally, the scalability is the ability of the system to extend itself without dramatic performance penalties. This is the most problematic issue. The difficulties are due to the explosive growth of the number of users, the geographic dispersion of users, and the size of the applications. It’s enough to look at distributed applications like searching the Web with Google, e-commerce applications on Amazon and eBay or the photo sharing application with Flickr to get an idea of this phenomenon. The solutions are based on extending the capacity and / or the number of resources to respond to the new higher requirements. In order to keep the performance unchanged, scaling is associated with new models and techniques such as: data replication, partitioning and
distribution to support parallel transactions, caching to reduce the communication time required in data access, fast parallel transfer for file distribution, load balancing, code migration, and service replication for reducing the response time, etc. While several issues related to scalability are still waiting for adequate solutions, some of successful large scale distributed systems and platforms incorporate stable, proved, innovative concepts and implementations which might constitute the subject of a book on this domain.

Performance improvement is just one subject of the actual research in large scale distributed systems. Another important one refers to extending the functionality. Web 2.0, the second generation of web-based communities and hosted services (such as social-networking sites, wikis, blogs, etc.) aims to facilitate creativity, collaboration and knowledge sharing between users and opens up an incredible number of options for flexible web design, creative reuse and easier updates. Applications such as those found on Google, Amazon, and eBay are driving Web 2.0 processing to the end users computer, leveraging the idle distributed computing capability of web clients. We are entering the era of Distributed Internet Applications, which have all the benefits of desktop applications yet leverage the Internet while remaining easily deployable to a mass market.

Web 3.0 makes steps ahead towards the Semantic Web (W3C, 2009), which uses technologies like Resource Description Framework (RDF), The Extensible Markup Language (XML), and Web Ontology Language (OWL) to make the Web content more meaningful to programs (machines) not only to people. Another dimension is offered by the Internet of Things (Dodson, 2003), which aims at extending the action of the Internet from people to any thing. This is based on the micro-miniaturization of Radio-Frequency Identification (RFID) tags that are able to associate Ids with the Things they are attached to, and of sensors that are able to detect changes in the physical status of those Things.

Many applications that support scientific research in astronomy, biology, medicine, engineering, high-energy physics, environment and other fields are also data- and compute-intensive (Foster and Kesselman, 2004) at large scale. To respond to their requirements, which overpass the capabilities of single high performance computers, Grid infrastructures have been conceived and developed. A Grid encompasses resources of various kinds and sizes that are linked together in a large area network and are used collaboratively by people working on common projects. The policy is to use the resources on demand, with some restrictions related to resources’ availability and applications’ QoS (quality of service) requirements. Since this might claim for moving large volume of data over the network and running long jobs on Grid resources, the good operation of the Grid is achieved by the use of complex coordinated resource sharing strategies, very fast data transfers, very efficient error recovery procedures, and very easy user access to storage and computing facilities or services. Finding new, efficient solutions for these complicated problems has been and still is a challenge for the specialists in the domain of large scale distributed systems.

As an alternative to sharing their own resources, users can pay for and use resources offered by a provider. This is the idea behind the cloud computing concept. The cloud encompasses hardware and software computer resources in data centers, which are accessible remotely through the Internet. For example, Amazon (2009) sells different services such as storage, databases, queue, Web applications, and others. Support for Amazon Web Services development is also provided. Cloud computing faces problems that are similar to Grid computing. The difference is that resources in a cloud belong to a single authority, which makes simpler the administration of resources and services. On the other hand, companies can collaborate to offer services in a cloud. For example, Amazon has partners that build solutions using Amazon Web Services (Amazon, 2009).
Other large scale distributed applications are developed on peer-to-peer infrastructures that include nodes with similar capabilities that communicate directly to each other for exchanging information (data) and performing collaborative tasks. Examples include Gnutella (2009), Kazaa (2009), BitTorrent (2009), and Skype. Peers voluntarily join specific systems to offer some services or resources and look for the services and resources offered by other peers. Consequently, important issues of peer-to-peer systems relate to searching for specific resources or services, information monitoring, security, and reliability.

OBJECTIVES

The book has three overall objectives: offer a coherent and realistic image of today’s research results in large scale distributed systems; explain state-of-the-art technological solutions for the main issues regarding large scale distributed systems, such as resource and data management, fault tolerance, security, monitoring, and controlling; and present the benefits of using large scale distributed systems and the development process of scientific and commercial distributed applications.

The book will also make readers familiar with new concepts and technologies that are successfully used in the implementation of today’s large scale distributed systems or have a good chance to be used in future developments. The approach is to not separate the theoretical concepts concerning the design of large scale distributed systems from their impact in real-world environment. For each important topic that one should master, the book plays the roles of bridge between theory and practice and of instrument needed by professionals in their activity. To this aim, the topics are presented in a logical sequence, and the introduction of each topic is motivated by the need to respond to the claims of new distributed applications. The advantages and limitations of each model or technology in terms of capabilities and areas of applicability are presented as well. The case studies included in each chapter offer models of how to use these instruments in solving the problems of some large scale distributed systems.

CHAPTER BY CHAPTER PRESENTATION

Chapter 1, Introduction, addresses the definition, goals, and fundamental issues related to large scale distributed computing. The presentation takes a pragmatic approach. It starts from typical examples of actual large scale distributed systems, which cover the well known categories, such as Enterprise Information Systems, Peer-to-Peer Systems, Grids, Utility and Volunteer Computer Systems. For each category, the motivation of use, the requirements and the problems raised by their implementation are taken as a base for introducing specific concepts, models, paradigms, and technologies. The presentation, which follows a historical perspective on large scale distributed computing, creates the framework for introducing future trends in the domain and paves the way to approach the convergence issues toward the future Cyberinfrastructure. In the same time, the chapter introduces a comprehensive set of concepts that are developed in the next chapters.

Chapter 2, Architectures for Large Scale Distributed Systems, introduces the macroscopic views on the components and their inter-relations in distributed systems. The importance of the architecture for understanding, designing, implementing, and maintaining distributed systems is presented first. Then the currently used architectures and their derivatives are analyzed. The presentation refers to the client-server (with details about Multi-tiered, REST, Remote Evaluation, and Code-on-Demand architectures),
hierarchical (with insights in the protocol oriented Grid architecture), peer-to-peer (with its versions: hierarchical, decentralized, distributed, and event-based integration architectures), and service-oriented architectures including OGSA (Open Grid Service Architecture). For each category, the chapter describes the model, presents the main issues and the actual research trends. Also it provides concrete cases of use in the actual distributed systems and platforms and clarifies the relation between the architecture and the enabling technology used in its instantiation. In addition, Chapter 2 frames the discussion in Chapters 3 to 10, which refer to specific components and services for large scale distributed systems.

In Chapter 3 we analyze current existing work in enabling high-performance communications in large scale distributed systems, presenting specific problems and existing solutions, as well as several future trends. By their nature, communication is an inherent aspect of every distributed application. Applications running in Grids, P2Ps and other types of large scale distributed systems have several specific communication requirements. For this reason we present the problem of delivering efficient communication in the case of P2P and Grid systems. The chapter starts with the review of high performance networks and technologies, where we analyze existing state-of-the-art solutions to enabling high quality communication over high speed networks. We present next peer-to-peer communication issues and solutions, moving on next to the specific requirements of the communication technologies in Grid systems. Also several patterns are analyzed from the point of view of semantics, methods and technologies. The chapter concludes with a presentation of the challenges in developing multicast and very high-speed communication software components.

Resource management is a central component in large scale systems. It can be implemented for a variety of architectures and services. Chapter 4 considers the management of distributed physical and virtual resources, and provides the requirements that are specific to large scale distributed system. Taxonomy of resource management methods is used to identify approaches followed in the implementation of actual systems, including Grids, and to discuss the solutions adopted in research and commercial platforms. The resource management system can support different users and resource owners’ constraints, according to different policies. Obeying to one policy could ask the resource allocation mechanisms to solve a multi-criteria optimization problem. An important subject is related to agent frameworks for resource management, which offer mechanisms for distributes resources management. An important subject presented in this chapter is Agents Frameworks for resource management that offers a mechanism for distributed resources management. The chapter ends with presentation of WSRF (Web Services Resource Framework) that is the new solution for resources management based on SOA (OGSA – Open Grid Service Architecture).

Chapter 5 presents the task scheduling problem in large scale systems (with examples from Grid and Web-based systems). The scheduling models are analyzed based on systems architecture described in chapter 2. The chapter presents scheduling algorithms for independent and dependent tasks, and provides a critical analysis of the most important algorithms. The workflow scheduling algorithms are presented for complex application management in large scale systems. The new scheduling mechanisms, like resources co-allocation and advance reservation, multi-criteria optimization mechanisms for user and system constraints (e.g. load-balancing, minimization of execution time) are described and analyzed in this chapter. The Implementation issues for scheduler tools are also presented.

Chapter 6, Data Storage, Retrieval and Management, introduces specific issues related to data handling in distributed systems. The chapter approaches topics related to the challenging problem of storing large amounts of data in distributed environments and of retrieving them for further analysis. In this context, the problem of ensuring a fast and reliable data transfer becomes crucial and is extensively explored. The
chapter further discusses key features for an efficient transfer solution to be used in large scale distributed systems and matches them with existing protocols and tools. The main problems of replication and consistency (that are related to issues like performance and fault tolerance) are discussed in the specific context of large scale distributed systems, highlighting particular models and solutions used there.

Chapter 7, Monitoring and Controlling Large Scale Systems, approaches the role, models, technologies and structure of a distributed monitoring platform. Monitoring is effectively used in many systems but has specific roles in Grids and other large scale distributed platforms. Monitoring data is used not only for services related to the past activity in these systems but also for prediction and learning purposes (e.g. scheduling further jobs according to a registered execution profile, avoiding bad schedules, and so on). The chapter discusses the challenges and requirements, the models used, the current architectures and specific solutions for all phases of the monitoring process: data production, data dissemination, data collection and presentation.

Chapter 8, Fault Tolerance, approaches the fault tolerance and other techniques used in the design of dependable distributed systems, a crucial characteristic for deploying highly-available, life-critical applications in modern large scale distributed systems. The chapter presents the special requirements for fault tolerance in large scale distributed systems, analyses the models used in representing the failures at different levels, and continues with the presentation of the most important strategies used in designing fault tolerance large scale distributed systems and applications. Emphasis is put on the special case of reliable communication in such systems. The chapter concludes with the techniques used to enable recovery from failures in large scale distributed systems.

Chapter 9, Security, starts by presenting the threats and vulnerabilities in large scale systems, and the difficulties encountered in preserving their confidentiality, integrity and availability. The Chapter is organized around the concept of security architecture and addresses three important problems: secure communication (with emphasis on the secure group communication), access control (more specific the access control in distributed multi-organizational platforms), and security management (especially key distribution management and trust management). For each problem, specific security models, mechanisms and protocols are described. The case studies used in this chapter refer to the security in Web, Grid, and peer-to-peer systems.

Chapter 10, Application development tools and frameworks, presents the engineering aspects of the distributed software development, from requirements to deployment and use. Specific tools, frameworks and portals for different development phases are introduced. The first section discusses the evolution of web applications and makes an overview of the current development platforms, especially Java EE and .NET. The chapter continues by presenting programming tools for Grid environments (such as Cog Kit and Grid MPI), which are mainly targeted at scientific applications. Clouds and peer-to-peer systems are addressed in the following sections, while the last part of the chapter is dedicated to distributed workflows – as the usage of workflow management platforms has been constantly increasing during the last years.

Chapter 11, Applications, start with a description of current projects and applications in large scale distributed systems, like those from OSG projects in USA, EGEE applications in Europe and Asia, and DEISA (Distributed European Infrastructure for Supercomputing Applications DEISA) initiative. We present the requirements for application development in Grids and P2P systems. Grid applications and Web-Based applications offer a reference for application development and related issues presented in the book such as application scheduling, application monitoring, data management, and application security.
TARGET AUDIENCE

The work is a scholarly book addressed mainly to researchers, professors, and teaching assistants who can find here a quick reference to the actual issues and research results in the domain of large scale distributed systems. The book could also represent an important help for PhD Students when documenting their research and looking for appropriate references to specific problems in this field.

This book is also well-suited for non-IT researchers and specialists from other data and intensive processing fields (physics, biology, etc.) who use large scale systems to run their applications and need a better understanding of the technologies involved. In this respect, the presentation of the specific concepts, research subjects, case studies, and distributes systems applications and application development tools is beneficial.

Individuals outside universities wishing to learn more about this important topic might also find this book useful. It targets software architects and developers, solution designers, IT specialists from professional environments interested in distributed systems for seeking appropriate solutions to their specific problems. For them, the book includes an informative introduction to the domain with emphasis on the design and implementation solutions of the large scale distributed systems and their applications.

After reading the book, the reader will be able to identify and use the concepts and technologies related to large scale distributed systems, from models to the implementation of technological solutions addressing the scheduling, monitoring, dependability, security, and other issues. The book introduces the reader to the high-level architectural view of large scale distributed systems and then to the technological solutions of the real-world implementation. The book facilitates the understanding of the new concepts used in a comprehensive set of real-world case studies. The reader will be able to easily recognize the concepts and structure of large scale distributed systems and will master the up-to-date technological solutions supporting the implementations of such systems.

CONCLUSION

The book presents the actual large scale distributed systems that are becoming more and more attractive in academia and industry as preferred computing infrastructures to be used for a wide-range of actual and next-generation applications. Most IT vendors and enterprise solutions adopters view distributed systems and their characteristics (such as virtualization, resource reallocation, automation and self-management) as foundation of the technology of the future, involving new kinds of IT procurement, delivery and usage models such as service oriented and utility models. Large scale distributed systems are presently helping many organizations to dynamically integrate their disparate, heterogeneous compute and storage resources. For these reasons the book presents the advantages of using large scale distributed systems and the development process of scientific and commercial distributed applications for the benefit of academic and business professionals as well.

The book also presents up-to-date technological solutions to the main aspects regarding large scale distributed systems, a highly dynamic scientific domain that gained much interest in the world of IT in the last decade. Distributed systems have matured from the large scale distributed computing science projects of the ‘90s to commercially viable business computing and network infrastructures. The book discusses nowadays computational large scale distributed systems that are used in solving some of the thorniest business problems affecting today’s networked economy: supply chain integration, virtual or-
ganizations, collaboration, and more. Along with covering the architecture and components behind the large scale distributed computing paradigm, the book introduces readers to the technologies that make up today’s large scale distributed platforms.

REFERENCES


