

Preface

OVERVIEW

Grid computing is one of the most innovative aspects of computing techniques in the last decade. Distinguished from conventional parallel and distributed computing, Grid computing focuses on resources sharing among geographically distributed sites and the development of innovative, high performance oriented applications. Computational Grid can present users with pervasive and inexpensive access to a wide variety of resources. Qualities of services (QoS) fall into the most important research topics of Grid computing with following concerns:

- **Heterogeneity:** Computational Grid is a highly heterogeneous environment. Different computing sites may have different types of resources. Even the resources of the same type, located at different sites, may have different configurations, capacities and performance profiles. Application users may expect to meet various programming interfaces and user interfaces from different computing sites.
- **Large-scale distribution:** Computational Grid enables resource sharing among geographically distributed sites. These sites are linked in the Internet. Network communication delay may be extremely high when some communication intensive applications are running among these sites. In this scenario, network performance has an important effect on resource management.
- **Dynamic environment:** Computational Grid is a highly dynamic environment in that computing capacities may vary in time; computing resources could join or withdraw the VO (Virtual Organization) base on their own interests.

The research on QoS mainly includes the qualitative analysis method and the quantitative analysis method. The qualitative QoS characters the Grid QoS aspects such as service availability, reliability and user satisfaction. This book mainly focuses on the Quantitative QoS.

To help develop complex Grid systems and software, the layered model is built to abstract the architecture of Grid systems. Each layer provides various services for upper layers. This book studies the Grid QoS aspects in various layers:

- **Fabric layer:** The Grid fabric layer provides the basic Grid protocols that enable Grid applications to share resources, which can be, for example, computational resources, storage systems, network resources and sensors. In the fabric layer, QoS of computing, storage and network are considered (Chapter IX, chapter XV).
- **Connectivity layer:** The Connectivity layer defines the core communication and authentication protocols required for Grid-specific network transmission. The communication protocols enable the exchange of data between Fabric layer resources (Chapter II).

- **Resource layer:** The resource layer builds the communication and authentication protocols of Connectivity layer to define protocols (and APIs, SDKs) for the secured negotiation, initiation, monitoring, control, accounting and payment of sharing operations on individual resources. In this layer, the research of Grid QoS focuses on local resource management policies and algorithms, resource allocation and reservation and resource access control (Chapter V, Chapter VI, Chapter VIII, Chapter XIII, Chapter XVII, Chapter XVIII, Chapter XIX, and Chapter XXII).
- **Collective layer:** The collective layer in the architecture contains protocols and services (and APIs, SDKs) that are not associated with any specific resource but rather interactions across collections of resources. Many research efforts have been developed in this layer, such as, resource co-allocation & co-reservation, resource brokering (Chapter I, Chapter III, Chapter IV, Chapter VII, Chapter XIV, and Chapter XX).
- **Application layer:** The application layer in the Grid architecture comprises the user applications that operate within a VO environment. In this layer various Grid services provided by other layers are employed to fulfill different Grid application requirements (Chapter X, Chapter XI, Chapter XII, Chapter XVI, and Chapter XXI).

This book defines and characterizes the latest research achievement in the QoS aspects for Grid computing. This book is supposed to be a milestone to summarize recent research works on Grid QoS. It is expected to be an important reference of Grid computing for the academia, especially for the research field of parallel & distributed computing, high performance computing, and Grid computing. All chapters of this book are based on recent research work of Grid experts and researchers. Expected readers include researchers, engineers, and IT professionals who work in the fields of parallel computing, distributed computing, cluster computing, Grid computing and high-performance computing. This book could also be employed as the reference book for postgraduate students who study computer science.

ORGANIZATION OF THE BOOK

This book includes 22 chapters contributed by 55 scholars. These book chapters cover the recent advances in QoS aspects of Grid computing: Grid infrastructure, resource management, workflow organization and scheduling, service-oriented architecture, and Grid applications.

Chapter I introduces two approaches that can provide QoS features at the workflow scheduling algorithm level in the Grid. One approach is based on a workflow rescheduling technique, which can reallocate resources for tasks when a resource performance change is observed. The other copes with the stochastic performance change using pre-acquired probability mass functions (PMF) and produces a probability distribution of the final schedule length, which will then be used to handle the different QoS concerns of the users.

Chapter II makes a study on dynamic network optimization for effective QoS support in large Grid infrastructures. At the entity Grid network layer, queuing strategies and shaping can be configured to allow for a certain treatment of packets. This needs administrative access to entities and can only applied in a limited scope like a local network. More generally, at the network layer, advanced network services like MPLS, GMPLS or DiffServ can be used to acquire committed bandwidth, specific transport features or QoS for applications exchanging data. In particular, with the evolution of MPLS technology, GMPLS can become the unified control plane technology to provide reliable transportation, efficient resource utilization and end-to-end QoS in Grid infrastructures.

Grid workflows are becoming a mainstream paradigm for implementing complex Grid applications. In addition to existing Grid enabling techniques, various Grid ensuring techniques are emerging, for

example workflow analysis and temporal reasoning, to probe potential pitfalls and errors and guarantee QoS at a design phase. A new state π calculus is proposed in **Chapter III**, which not only enables flexible abstraction and management of historical Grid system events, but also facilitates modeling and verification of Grid workflows. Some typical patterns in Grid workflows are captured and both static and dynamic formal verification issues are investigated, including structural correctness, specification satisfiability, logic satisfiability and consistency. A Grid workflow modeling and verification environment, GridPiAnalyzer, is implemented using formal modeling and verification methods proposed in this work. Performance evaluation results are included using a Grid workflow for gravitational wave data analysis.

In **Chapter IV**, a cost-based resource management and scheduling strategy is presented for the computational Grid, which borrows the idea from economic principles. The main idea is that the usage of heterogeneous resources such as CPU speed, memory capability, and network bandwidth is converted into a homogeneous cost based on some rule, although these resources are measured in unrelated units. According to the goal of better QoS, tasks are scheduled conveniently in the computational Grid.

Consistency control is important in replication-based Grid systems because it provides QoS guarantee. However, conventional consistency control mechanisms incur high communication overhead and are ill suited for large-scale dynamic Grid systems. In **Chapter V**, the authors propose CVRetrieval (Consistency View Retrieval) to provide quantitative scalability improvement of consistency control for large-scale, replication-based Grid systems.

Chapter VI elaborates the QoS aspect of load sharing activities in a computational Grid environment. This chapter defines QoS based performance metrics for evaluating job scheduling and resource allocation strategies. According to the QoS performance metrics appropriate Grid-level load sharing strategies are developed. The developed strategies address both user-level and site-level QoS concerns. A series of simulation experiments were performed to evaluate the proposed strategies based on real and synthetic workloads.

Chapter VII focuses on presenting and describing an approach that allows the mapping of workflow processes to Grid provided services by not only taking into account the QoS parameters of the Grid services but also the potential business relationships of the service providers, which may affect the aforementioned QoS parameters. This approach is an integral part of the QoS provisioning, since this is the only way to estimate, calculate, and conclude to the mapping of workflows and the selection of the available service types and instances in order to deliver an overall quality of service across a federation of providers. The added value of this approach lays on the fact that business relationships of the service providers are also taken into account during the mapping process.

Opportunistic techniques have been widely used to create economical computation infrastructures and have demonstrated an ability to deliver heterogeneous computing resources to large batch applications; however, batch turnaround performance is generally unpredictable, negatively impacting human experience with Grid resources. Scheduler prioritization schemes can effectively boost the share of the system given to particular users, but to gain a relevant benefit to user experience, whole batches must complete on a predictable schedule, not just individual jobs. Additionally, batches may contain a dependency structure that must be considered when predicting or controlling the completion time of the whole workflow; the slowest or most volatile prerequisite job determines performance. In **Chapter VIII**, a probabilistic policy enforcement technique is used to protect deadline guarantees against Grid resource unpredictability as well as bad estimates. Methods to allocate processors to a common workflow subcase, barrier scheduling, are also presented.

Grids can form the basis for pervasive computing due to their ability of being open, scalable and flexible to various changes (from topology changes to unpredicted failures of nodes). However, such environments are prone to failures due to their nature and need a certain level of reliability in order to

provide viable and commercially exploitable solutions. This is causing a significant research activity which is focused on the topic of achieving certain levels of QoS in highly unreliable environments (such as mobile and ad hoc Grids). **Chapter IX** focuses on the state-of-the-art analysis of the QoS aspects in Grids and how this is achieved in terms of technological means.

Web knowledge flow provides a technique and theoretical support for the effective discovery of knowledge innovation, intelligent browsing, personalized recommendation, cooperative team work, and the semantic analysis of resources on Internet, which is a key issue of Web services and knowledge Grid. In **Chapter XI**, the authors introduce some basic concepts related to Web knowledge flow and illustrate the concepts of interactive computing, including the Web interaction model, the implementation of interactive computing and the generation of Web knowledge flow. In this chapter, the applications of Web knowledge flow are also given.

Chapter XII mainly introduces some recent researches of reputation evaluation methods in Grid economy. The GRACE (Grid Architecture for Computational Economy architecture) is adopted to explain some mechanisms in the Grid economy for its clearly inner modules architecture. In addition, several new developed modules based on GRACE architecture are detailed discussed and two of them are laid more emphasis on by us, which are the RCM (Reputation Control Module) and distributed reputation control architectures based on VOD (Virtual Organizational Domain). The inner communication and workflow of them are shown in this chapter. Furthermore, through experiments results, the authors discover the profit of Grid nodes and tasks execution success rate are all improved by adding these new modules.

A major design challenge in wireless sensor network application development is to provide appropriate middleware service protocols to control the energy consumption according to specific application scenarios. In common application scenarios such as in monitoring or surveillance systems, it is usually necessary to extend the system monitoring area as large as possible to cover the maximal area. The two issues of power conservation and maximizing the coverage area have to be considered together with both the sensors' communication connectivity and their power management strategy. In **Chapter XIII**, Fu and Wang propose novel enhanced sensor scheduling protocols to address the application scenario of typical surveillance systems. The protocols take into consideration of both power conservation and coverage ratio to search for the balance between the different requirements. Chapter XIII proposes both centralized and de-centralized sensor scheduling versions, and compared the performance of different algorithms using several metrics. The results provide evidence of the advantages of the proposed protocols comparing with existing sensor scheduling protocols.

In scientific computing environments such as service Grid environments, services are becoming basic collaboration components which can be used to construct a composition plan for scientists to resolve complex scientific problems. However, current service collaboration methods still suffer from low efficiency for automatically building composition plans because of the time-consuming ontology reasoning and incapability in effectively allocating resources to executing such plans. **Chapter XIV** presents a QSQL-based collaboration method to support automatic service composition and optimized execution. With the method, for a given query, abstract composition plans can be created in an automatic, semantic and efficient manner from QSQL (Quick Service Query List) which is dynamically built by previously processing semantic-related computing at service publication stage. Furthermore, concrete service execution instances can be dynamically bound to abstract service composition plans at runtime by comparing their different QoS values.

It will become increasingly popular that scientists in research institutes will make use of Grid computing resources for running computer simulations and managing data. Although there are some production Grids available, it is often the case that many organizations and research projects need to build their own Grids. However, building Grid infrastructure is not a trivial job as it involves sharing and manag-

ing heterogeneous computing and data resources across different organizations, and involves installing many specific software packages and various middleware. This can be quite complicated and time-consuming. Building a Grid infrastructure also requires good knowledge and understanding of distributed computing and Grid technology. Apart from building physical Grid, how to build a user infrastructure that can facilitate the use of and easy access to these physical infrastructures is also a challenging task. In **Chapter XV**, Yang and Chiang summarize some hands-on experience in building an institutional Grid infrastructure.

The emergence of Grid technologies provide exciting new opportunities for large scale simulation over Internet, enabling collaboration and the use of distributed computing resources, while also facilitating access to geographically distributed data sets. **Chapter XVI** presents HLA_Grid_RePast, a middleware platform for executing large scale collaborating RePast agent-based models on the Grid. This chapter also provides the performance results and analysis on Quality of Service from a deployment of the system between the United Kingdom and Singapore.

Due to the rapidly increasing number of mobile devices connected to the Internet, a lot of research is being conducted to maximize the benefit of such integration. The main objective of **Chapter XVII** is to enhance the performance of the scheduling mechanism of the mobile computing environment by distributing some of the responsibilities of the access point among the available attached mobile devices. To this aim, the authors investigate a scheduling mechanism framework that comprises an algorithm that provides the mobile device with the authority to evaluate itself as a resource. The proposed mechanism is based on the “self ranking algorithm” (SRA), which provides a lifetime opportunity to reach a proper solution. This mechanism depends on an event-based programming approach to start its execution in a pervasive computing environment.

Web services’ discovery mechanism is one of the most important research areas in Web services because of the dynamic nature of Web services. In practice, UDDI takes an important role in service discovery since it is an online registry standard to facilitate the discovery of business partners and services. However, QoS related information is not naturally supported in UDDI. Service requesters can only choose good performance Web services by manual test and comparison. In addition, discovery among private UDDI registries in a federation is not naturally supported. To address these problems, **Chapter XVIII** proposes UDDI extension (UX), an enhancement for UDDI that facilitates requesters to discover services with QoS awareness.

The purpose of **Chapter XIX** is to investigate the requirements of knowledge management (KM) services deployment in a Semantic Grid environment. A wide range of literature on Grid Computing, Semantic Web, and KM have been reviewed, related, and interpreted. The benefits of the Semantic Web and the Grid Computing convergence have been enumerated and related to KM principles in a complete service model in Chapter XIX.

The increasing ability for the sciences to sense the world around us is resulting in a growing need for data-driven e-science applications that are under the control of workflows composed of services on the Grid. The focus of **Chapter XX** is on provenance collection for these workflows that are necessary to validate the workflow and to determine quality of generated data products. The challenge addressed in this chapter is to record uniform and usable provenance metadata that meets the domain needs while minimizing the modification burden on the service authors and the performance overhead on the workflow engine and the services.

Chapter XXI introduces an ontology-based framework for automated construction of complex interactive data mining workflows as a means of improving productivity of Grid-enabled data exploration systems. The authors first characterize existing manual and automated workflow composition approaches

and then present their solution called GridMiner Assistant (GMA), which addresses the whole life cycle of the knowledge discovery process. GMA is specified in the OWL language and is being developed around a novel data mining ontology, which is based on concepts of industry standards like the predictive model markup language, cross industry standard process for data mining, and Java data mining API. The ontology introduces basic data mining concepts like data mining elements, tasks, services, and so forth. In addition, conceptual and implementation architectures of the framework are presented and its application to an example taken from the medical domain is illustrated.

In **Chapter XXII**, two algorithms have been presented for supporting efficient data transfer in the Grid environment. From a node's perspective, a multiple data transfer channel can be formed by selecting some other nodes as relays in data transfer. One algorithm requires the sender to be aware of the global connection information while another does not. Experimental results indicate that both algorithms can transfer data efficiently under various circumstances.