Preface

MOTIVATION AND ORGANIZATION OF THE BOOK

Scientific applications generate increasingly large amounts of data, often referred as the "data deluge," which necessitates collaboration and sharing between national and international research institutions. Simply purchasing high-capacity, high-performance storage systems and adding them to the existing infrastructure of the collaborating institutions does not solve the underlying and highly challenging data handling problem. Scientists are often forced to spend a great deal of time and energy on solving basic data-handling issues, such as the physical location of data, how to access it, and/or how to move it to visualization and/or compute resources for further analysis.

In this book, experts on data intensive computing discuss the challenges imposed by data-intensive applications on distributed systems, and present state-of-the-art solutions proposed to overcome these challenges. This book is intended to be a reference for research scientists and application developers working with complex, data intensive, and data-driven applications on distributed environments. It can also be used as a textbook for advanced distributed systems, data management, and related courses.

This book is organized in four sections: i) New Paradigms in Data Intensive Computing; ii) Distributed Storage; iii) Data and Workflow Management; and iv) Data Discovery and Visualization.

Section 1, New Paradigms in Data Intensive Computing, focuses on new generation of computing paradigms proposed to overcome the challenges of complex, data intensive, and data-driven applications running on distributed environments. It includes three chapters: "Data-Aware Distributed Computing," "Towards Data Intensive Many-Task Computing," and "Micro-Services: A Service-Oriented Paradigm for Scalable, Distributed Data Management."

Chapter 1, "Data-Aware Distributed Computing," elaborates on how the most crucial distributed computing components such as scheduling, workflow management, and end-to-end throughput optimization can become "data-aware." In this new computing paradigm, called data-aware distributed computing, data placement activities are represented as full-featured jobs in the end-to-end workflow, and they are queued, managed, scheduled, and optimized via a specialized data-aware scheduler. As part of this new paradigm, the authors present a set of tools for mitigating the data bottleneck in distributed computing systems, which consists of three main components: a data-aware scheduler, which provides capabilities such as planning, scheduling, resource reservation, job execution, and error recovery for data movement tasks; integration of these capabilities to the other layers in distributed computing, such as workflow planning; and further optimization of data movement tasks via dynamically tuning of underlying protocol transfer parameters.

Chapter 2, "Towards Data Intensive Many-Task Computing," presents a new computing paradigm called many-task computing, which aims to bridge the gap high throughput computing and high performance computing. Many task computing denotes high-performance computations comprising multiple distinct activities, coupled via file system operations. The aggregate number of tasks, quantity of computing, and volumes of data may be extremely large. The authors also propose a "data diffusion" approach to enable data-intensive many-task computing. Data diffusion acquires compute and storage resources dynamically, replicates data in response to demand, and schedules computations close to data, effectively harnessing data locality in application data access patterns.

Chapter 3, "Micro-Services: A Service-Oriented Paradigm for Scalable, Distributed Data Management," defines micro-services, which are orchestrated into conditional workflows for achieving large-scale data management specific to collections of data. Micro-services communicate with each other using parameter exchange, in memory data structures, a database-based persistent information store, and a network messaging system that uses a serialization protocol for communicating with remote micro-services. The orchestration of the workflow is done by a distributed rule engine that chains and executes the workflows and maintains transactional properties through recovery micro-services. The authors discuss the micro-service oriented architecture, compare the micro-service approach with traditional service-oriented architectures (SOA), and describe the use of micro-services for implementing policy-based data management systems.

Section 2, Distributed Storage, focuses on design and implementation of advanced storage systems for sharing large amounts of data between distantly collaborating researchers. It includes three chapters: "Distributed Storage Systems for Data Intensive Computing," "Metadata Management in PetaShare Distributed Storage Network," and "Data Intensive Computing with Clustered Chirp Servers."

Chapter 4, "Distributed Storage Systems for Data Intensive Computing," presents an overview of the utility of distributed storage systems in supporting modern applications that are increasingly becoming data intensive. The coverage of distributed storage systems in this chapter is based on the requirements imposed by data intensive computing and not a mere summary of storage systems. To this end, the authors delve into several aspects of supporting data-intensive analysis, such as data staging, offloading, checkpointing, and end-user access to terabytes of data, and illustrate the use of novel techniques and methodologies for realizing distributed storage systems therein. The data deluge from scientific experiments, observations, and simulations is affecting all of the aforementioned day-to-day operations in data-intensive computing. Modern distributed storage systems employ techniques that can help improve application performance, alleviate I/O bandwidth bottleneck, mask failures, and improve data availability. The authors present key guiding principles involved in the construction of such storage systems, associated tradeoffs, design, and architecture, all with an eye toward addressing challenges of data-intensive scientific applications.

Chapter 5, "Metadata Management in PetaShare Distributed Storage Network," presents the design and implementation of a reliable and efficient distributed data storage system, PetaShare, which spans multiple institutions across the state of Louisiana. At the back-end, PetaShare provides a unified name space and efficient data movement across geographically distributed storage sites. At the front-end, it provides light-weight clients the enable easy, transparent, and scalable access. In PetaShare, the authors have designed and implemented an asynchronously replicated multi-master metadata system for enhanced reliability and availability. The authors also present a high level cross-domain metadata schema to provide a structured systematic view of multiple science domains supported by PetaShare.

Chapter 6, "Data Intensive Computing with Clustered Chirp Servers," presents Chirp as a building block for clustered data intensive scientific computing. Chirp was originally designed as a lightweight file server for grid computing and was used as a "personal" file server. The authors explore building systems with very high I/O capacity using commodity storage devices by tying together multiple Chirp servers. Several real-life applications such as the GRAND Data Analysis Grid, the Biometrics Research Grid, and the Biocompute Facility use Chirp as their fundamental building block, but provide different services and interfaces appropriate to their target communities.

Section 3, Data and Workflow Management, focuses on the challenges of managing and scheduling complex workflows and large-scale data replication for data intensive applications. It includes three chapters: "A Survey of Scheduling and Management Techniques for Data-Intensive Application Workflows," "Data Management in Scientific Workflows," and "Replica Management in Data Intensive Distributed Science Applications."

Chapter 7, "A Survey of Scheduling and Management Techniques for Data-Intensive Application Workflows," presents a comprehensive survey of algorithms, techniques, and frameworks used for scheduling and management of data-intensive application workflows. Many complex scientific experiments are expressed in the form of workñows for structured, repeatable, controlled, scalable, and automated executions. This chapter focuses on the type of workñows that have tasks processing huge amount of data, usually in the range from hundreds of mega-bytes to petabytes. Scientists are already using Grid systems that schedule these workñows onto globally distributed resources for optimizing various objectives: minimize total makespan of the workñow, minimize cost and usage of network bandwidth, minimize cost of computation and storage, meet the deadline of the application, and so forth. This chapter lists and describes techniques used in each of these systems for processing huge amount of data. A survey of workñow management techniques is useful for understanding the working of the Grid systems providing insights on performance optimization of scientific applications dealing with data-intensive workloads.

Chapter 8, "Data Management in Scientific Workflows," describes a workflow lifecycle as consisting of a workflow generation phase where the analysis is defined, the workflow planning phase where resources needed for execution are selected, the workflow execution part, where the actual computations take place, and the result, metadata, and provenance storing phase. The authors discuss the issues related to data management at each step of the workflow cycle. They describe challenging problems and illustrate them in the context of real-life applications. They discuss the challenges, possible solutions, and open issues faced when mapping and executing large-scale workflows on current cyberinfrastructure. They particularly emphasize the issues related to the management of data throughout the workflow lifecycle.

Chapter 9, "Replica Management in Data Intensive Distributed Science Applications," provides an overview of replica management schemes used in large, data-intensive, distributed scientific collaborations. Early replica management strategies focused on the development of robust, highly scalable catalogs for maintaining replica locations. In recent years, more sophisticated, application-specific replica management systems have been developed to support the requirements of scientific Virtual Organizations. These systems have motivated interest in application-independent, policy-driven schemes for replica management that can be tailored to meet the performance and reliability requirements of a range of scientific collaborations. The authors discuss the data replication solutions to meet the challenges associated with increasingly large data sets and the requirement to run data analysis at geographically distributed sites.

Section 4, Data Discovery and Visualization, focuses on techniques for mining, discovering, and visualization of large data sets. It includes three chapters: "Data Intensive Computing for Bioinformatics,"

"Visualization of Large-Scale Distributed Data," and "On-Demand Visualization on Scalable Shared Infrastructure."

Chapter 10, "Data Intensive Computing for Bioinformatics," discusses the use of innovative data-mining algorithms and new programming models for several Life Sciences applications. The authors particularly focus on methods that are applicable to large data sets coming from high throughput devices of steadily increasing power. They show results for both clustering and dimension reduction algorithms, and the use of MapReduce on modest size problems. They identify two key areas where further research is essential, and propose to develop new O(NlogN) complexity algorithms suitable for the analysis of millions of sequences. They suggest Iterative MapReduce as a promising programming model combining the best features of MapReduce with those of high performance environments such as MPI.

Chapter 11, "Visualization of Large-Scale Distributed Data," introduces different instantiations of the visualization pipeline and the historic motivation for their creation. The authors examine individual components of the pipeline in detail to understand the technical challenges that must be solved in order to ensure continued scalability. They discuss distributed data management issues that are specifically relevant to large-scale visualization. They also introduce key data rendering techniques and explain through case studies approaches for scaling them by leveraging distributed computing. Lastly they describe advanced display technologies that are now considered the "lenses" for examining large-scale data.

Chapter 12, "On-Demand Visualization on Scalable Shared Infrastructure," explores the possibility of developing parallel visualization algorithms that can use distributed, heterogeneous processors to visualize cutting edge simulation datasets. The authors study how to effectively support multiple concurrent users operating on the same large dataset, with each focusing on a dynamically varying subset of the data. From a system design point of view, they observe that a distributed cache offers various advantages, including improved scalability. They developed basic scheduling mechanisms that were able to achieve fault-tolerance and load-balancing, optimal use of resources, and flow-control using system-level back-off, while still enforcing deadline driven (i.e. time-critical) visualization.

Tevfik Kosar
State University of New York at Buffalo (SUNY), USA