GRID COMPUTING

Techniques and Applications

PREFACE

BARRY WILKINSON

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group an **informa** business
A CHAPMAN & HALL BOOK

Preface

The purpose of this text is to introduce Grid computing techniques and form the basis for a practical senior undergraduate course or first-year graduate course on Grid computing. Grid computing uses geographically distributed computers collectively for high performance computing and resource sharing. The computers can be distributed locally, nationally, or across countries and continents. Grid computing often involves computers from multiple organizations and hence crosses organizational boundaries. Distributed teams can be formed to tackle major problems such as searching for new drugs and cures. Apart from computers, the shared resources might include expensive experimental equipment located at one site but shared by all in the team. The experimental equipment might generate huge amounts of data that need to be studied by members of the team using computers at other sites. The term *virtual organization* is used to describe the distributed team and the shared resources that are provided.

Grid computing first became viable with the widespread growth of high-speed networks and the Internet in the mid-1990s. The continued growth and performance of networks has fueled numerous Grid computing projects across the world. Very high performance Grid projects might use special dedicated high-speed networks but the general Internet also provides the backbone for many Grid computing projects. The use of the Internet enables everyone to become involved in Grid computing. A key aspect of Grid computing is the use of standard Internet protocols and techniques, which includes the basic transport protocols, but also Web services and associated technologies.

We capitalize the word Grid, not because there is only one Grid—there are many Grids that are set up locally, nationally and internationally—but because it is a common practice to do so.

Book Background. This book is the direct outcome of work done on introducing Grid computing into the undergraduate curricula by the author and Dr. Clayton Ferner from the University of North Carolina at Wilmington (UNC–Wilmington). Since Grid computing often involves computers at multiple sites on the Internet, to teach Grid computing, it is desirable to have cooperating geographically distributed sites. A Grid was set up using computing resources at several North Carolina universities crossing administrative boundaries as in many production Grids. We also took advantage of the existing state-wide televideo/teleconferencing network called North Carolina Research and Education Network (NCREN) to present the lectures to students at many universities simultaneously. NCREN is a telecommunications network that became operational in 1985 to interconnect universities, medical center, research institutions, and graduate centers in North Carolina and provides multi-way, face-to-face video and audio communications. "Teleclass" classroom facilities are provided at each site. Each student is provided with a microphone and multiple video cameras are used so that the instructor and students can hear and see each other.

With funding from the National Science Foundation (NSF) and the University of North Carolina Office of the President (UNC-OP), our Grid computing course was first offered in Fall 2004, and repeated in Fall 2005. In Fall 2004, eight institutions participated, and in Fall 2005 twelve institutions participated. The course was redesigned to use a more top-down approach with several new features such as the use of a production-style Grid portal and less reliance of centralized servers for student assignments. The new course was offered in Spring 2007 and again Fall 2008. Additional funding was received from the NSF in 2008 to incorporate a Grid computing workflow editor called GridNexus further into the course. The book is based upon this re-designed course.

Throughout the four years of development, students and faculty from a wide range of institutions were involved, including premier research universities, comprehensive state universities, private four-year colleges, minority-serving institutions, and a technical community college, fifteen institutions in total.[1] Apart from formal lectures given by instructors, internationally known guest speakers were invited to give presentations from different sites. Streaming video of the classes was provided by NCREN, which enabled students to watch the class from the Internet in real time or download the class for watching later. Depending upon the offering, computer systems were set up at between three and five universities to create a working Grid for the students. To do all of this, the instructors were assisted by many people (see Acknowledgements).

The course was designed primarily for upper-level undergraduate Computer Science students, although graduate students were accommodated by providing extra work. The course was recognized as the "Link of the Week" in the June 15, 2005

[1] Appalachian State University, Elon University, North Carolina A & T University, North Carolina Central University, North Carolina State University, University of North Carolina at Asheville, University of North Carolina Chapel Hill, University of North Carolina at Charlotte, University of North Carolina at Greensboro, University of North Carolina at Pembroke, University of North Carolina at Wilmington, Western Carolina University, Winston-Salem State University, Lenoir Rhyne College, and Wake Technical Community College.

issue of *Science Grid This Week*, and received further national attention in the feature article of *Science Grid This Week* in December 14, 2005 (repeated in *GridToday*). In addition to conference papers, a short article of the re-designed course was published in *International Science Grid This Week* in March 26, 2008. More information can be found at http://www.cs.uncc.edu/~abw/gridcourse/ including links to publications and all course materials in each offering of the course.

Structure of Materials. The book starts with Chapter 1 which is an introduction to Grid computing and its applications. Grid computing is about executing jobs on a distributed computing platform and the chapter leads onto using a Grid computing Web-based portal, which is used in real Grid computing projects. In our course, we get the students to register on a course portal after the first class and they immediately submit jobs to the Grid platform using the portal. In Chapter 2, the underlying action of job submission using a command-line interface is considered in some depth. Grid portals hide some of these details but the command-line interface is still needed to appreciate fully the underlying Grid infrastructure. We return to the portal later. Chapter 3 discusses the use of a job scheduler. Jobs usually enter a job queue and are sent to an appropriate compute resource as selected by a scheduler. Chapter 4 describes general Internet security techniques, which are the basis for Grid computing security. Chapter 5 describes the specific security mechanisms developed for Grid computing. Chapter 6 describes Web services technology. Grid computing middleware software is aligned to Web services. Chapter 7 describes how Web services are adopted for Grid computing. Chapter 8 focuses on graphical user interfaces. A user can interact with the Grid software either through a command-line interface or usually preferably through a graphical interface. A graphical interface offers several advantages. For example, it can offer scientists a domain-specific interface. It can make it easier for non-Computer Science users to access the Grid. We concentrate upon the GridSphere portal introduced in Chapter 1. Gridsphere conforms to currently agreed standards. The chapter describes how that portal can be customized to produce specific interfaces. Also in this chapter, we describe graphical workflow editors that enable the user to compose sequences of computational tasks visually using a simple drag-and-drop interface. We concentrate upon a graphical workflow editor developed at UNC–Wilmington called GridNexus which we also use in our class. The final chapter, Chapter 9, describes how to deploy applications on Grid. Although the last chapter, this chapter is very important and often not addressed in detail. Most applications can be run at one site if everything the application needs is installed at that site. However, the Grid computing platform offers much more than simply running an application at a remote site. It can also offer using multiple geographically distributed computers collectively to obtain increased speed and fault tolerance. It can offer resource discovery.

Each chapter concludes with a summary, further reading, bibliography, multiple-choice self-assessment questions, and programming assignments. The self-assessment questions are provided to check your understanding of the presented materials. The answers are given at the back of the book. Some of these programming assignments can be done on a PC once certain open-source software is installed as

explained in the assignments. Other assignments do need access to a Grid computing platform.

The appendices offer useful closely related background materials, especially for doing Grid computing assignments. Appendix A covers Internet and networking basics including IP addressing and Internet protocols. Appendix B focuses on operating system environments and covers commonly used Linux and Windows commands that are helpful in assignments. Appendix C covers the XML language, which is needed for Web and Grid computing services. Appendix C could be read prior to Chapter 6 and Chapter 7 if needed. Appendix D, written by Jeremy Villalobos, provides a tutorial on the Globus installation. This appendix is intended to supplement on-line instructions on installing the Globus toolkit and related software. It provides notes on practical experiences.

The material in the book can be used in the order presented, which is top down and back, i.e., begin with a portal and user job submission, delve into the Grid infrastructure and return at end to Grid-enabling applications. However, it can also be used in a different order to obtain a more bottom-down approach by covering Web services first. We use the material more-or-less in the order given, by starting with the users submitting jobs to a portal in the first class and then delve into what is behind the portal and study the command line interface. Security (Chapters 4 and 5) is covered later in the course. Some materials may already be known, for example that on Internet security (Chapter 4), and in that case could be skipped. If XML is not known, Appendix C would become part of the course. There are some dependencies as shown in Figure P.1. Chapter 2 is written to be done before Chapter 3. Chapter 4, if not already known, has to be done before Chapter 5. Chapter 6, if not already known, has to be done before Chapter 7. There are some partial dependencies, that is, for full appreciation of the materials, some previous materials are helpful. As one can see, after the Introduction, one could do the job submission sequence, the security

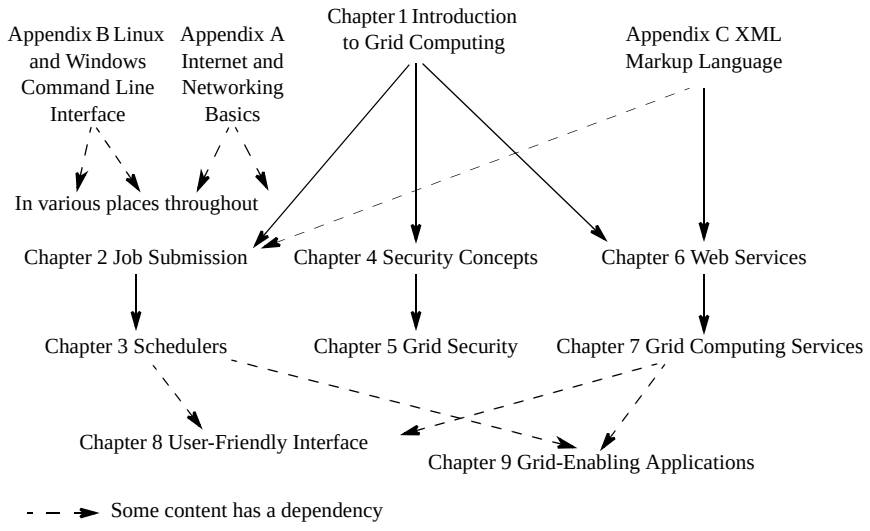

Figure P.1 Chapter dependencies.

sequence or the Web service sequence first, or interleaved combinations. Although the GridNexus workflow editor is described in a single chapter (Chapter 8), it can be introduced with schedulers (Chapter 3) for graphical job execution workflows and with Chapter 7 with graphical Web service workflows—in fact we do that in our class.

Grid Computing Platform. All the software to create a Grid computing platform is available by free download from various sites. The book concentrates on using the Globus toolkit as the central Grid computing software. Although ideally the course should use geographically distributed computers, the course can be presented to students at a single site as regular undergraduate or graduate-level Computer Science or Engineering programming course. It is usually most convenient to set aside computers/servers just for the course than attempt to use systems that are shared with other activities.

Using a centralized server for some Grid computing activities can be problematic. There are a number of educational activities that can be done on one's own personal computer (or laboratory computer) without access to a Grid platform and indeed it is much better to do so. For example, exercises using Web services (Chapters 6 and 7) can readily be done on a personal computer. Web services require a hosting environment (such as Apache Axis/Tomcat), which can be readily installed. Grid computing services can be hosted in the Globus container, a core software component of Globus. Being available in Java, it can be installed on a personal computer. Then, one's own services can be deployed and tested with local clients. This is much preferable to using a centralized server—with a single Web service hosting environment (container) or multiple hosting environments (containers). A large number of simultaneous containers is not practical because of the relatively large footprint of each container, which can cause operating system thrashing. Also, each container would need a separate port. Using a single container with multiple users is also problematic. Deploying/re-deploying services require the container to be re-started each time. Also all users would see all the deployed services and each service needs a unique name. In our early courses (Fall 2004 and Fall 2005), a script was provided that renamed students' services automatically but still continual restarting the container causes all users to be disrupted. Using one's own computer resolves all these issues and also provides users with a powerful learning experience and satisfaction in installing complex software components. Students have responded very positively to doing their software development on their own computer.

The Grid computing GUI workflow editor we use in GridNexus (Chapter 8) is freely available and can be easily installed on a personal computer to create workflows. Similarly, designing portlets within a Grid computing portal such as GridSphere can be done on a personal computer as portlet environments such as GridSphere/Tomcat can be installed there. In fact, portlets can be designed with a locally installed Grid portal that interfaces with either locally installed or remotely installed Web services (the latter with a network connection). Such activities combine materials in Chapters 6, 7, and 8, and relate to Chapter 9.

Activities relating to job submission, job scheduling, and security (Chapters 2, 3, 4, and 5) generally still need access to a Linux server or lab computer. GridNexus

workflows (Chapter 8) can include workflows that execute remotely, for example remote job submission, file transfers, and Web services on different servers.

Prerequisites. The materials in this book are designed primarily for upper-level undergraduate and first-year graduate Computer Science and Engineering students with knowledge of Linux, C, and Java. Most such Computer Science and Engineering students have this knowledge or can quickly learn it sufficiently. Grid computing actually brings together topics found in other contexts. For example, job scheduling may be part of a course on operating systems. Aspects of Internet security such as certificates, certificate authorities, and secure network protocols, are found in networking courses. Web services might be found on courses on distributed computing. The underlying technology of portals and portlets such as servlets might also appear in courses on network applications. This knowledge is not assumed.

Home Page. A Web site has been provided for instructors and students at http://www.cs.uncc.edu/~abw/GridComputingBook/. This Web site includes all the instructional materials needed including slides and programming assignments. We have designed our assignments so that many could be done without access to a Grid—instead students are asked to install open-source software on their own computers or on laboratory computers to do the work. This includes software environments to deploy and test Web and Grid services, a Grid computing workflow editor to design and test workflows, and a Grid computing portal for deploying portlets. We provide step-by-step instructions for students on the home page, which supplement the description of assignments in the book. Certainly, servers are still needed for running jobs and experimenting with job schedulers and assignments are provided that do require access to servers.

Acknowledgements. Partial support for this work was provided by the NSF's Course, Curriculum, and Laboratory Improvement (CCLI) program under awards #0410667/053334 and #0737318/0737269/0737208. Funding was provided by the UNC-OP under two major 2004–2006 awards. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the NSF or UNC-OP. I wish to record my appreciation to the NSF and UNC-OP for providing support to enable us to develop the Grid computing courses that this book is based upon.

The course required the cooperation of *many* people at many universities. I would particularly like to thank Clayton Ferner (UNC–Wilmington) who joined me in teaching the course from the beginning and contributed throughout its development and made the Grid course a success. Mark Holliday at Western Carolina University was instrumental in starting the work. He was the co-PI on the first NSF grant and one UNC-OP grant that supported the initial development of the Grid course, and I wish to record my appreciation to him.

The following provided direct assistance at their institutions: Barry Kurtz and Rahman Tashakkori at Appalachian State University, Dave Powell and J. Hollingworth at Elon University, Yaohang Li at North Carolina A & T University, Mladen Vouk and Gary Howell at North Carolina State University, Dean Brock at the Univer-

sity of North Carolina at Asheville, Shan Suthaharan at the University of North Carolina at Greensboro, and Dick Hull at Lenoir Rhyne College. Dr. Mark Holliday's undergraduate students James Ruff and Jeffrey House developed several of the early assignments. James Ruff maintained the Grid computing installation at Western Carolina University. Natasha Stracener, TV Media Services Coordinator, Broadcast Communications at UNC–Charlotte managed the NCREN teleconferencing facilities at Charlotte and provided continuous support.

Appendix D, a Globus installation tutorial, was written by Jeremy Villalobos, a Ph.D. student in the Department of Computer Science at University of North Carolina at Charlotte. Jeremy is responsible for maintaining the Grid site at Charlotte for our Grid course. His Ph.D. work is on aspects of computational Grid computing, in particular in developing a high-level framework for distributed computations on Grid platform. Ramya Chaganti, a Computer Science MS student at University of North Carolina at Charlotte, was extremely helpful in assisting in the preparation of the book, including providing some screen shots in Chapter 8.

The following kindly gave guest lectures to our classes between 2004–2008: Professor Daniel A. Reed, Chancellor's Eminent Professor, Director of Renaissance Computing Institute (at the time); Wolfgang Gentzsch, Managing Director at Microelectronics Center of North Carolina (MCNC) Grid Computing and Networking Services (at the time); Chuck Kesler, Director of Grid Deployment and Data Center Services, MCNC (at the time); Jeff Schmitt, genesismolecular.com; Jim Jokl, University of Virginia; Art Vandenberg, Georgia State University; Mary Fran Yafchak, Southeastern Universities Research Association (SURA); Lavanya Ramakrishnan, Renaissance Computing Institute; Purushotham Bangalore, University of Alabama at Birmingham; Joel Hollingsworth, Elon University; Carla Hunt, MCNC; Yaohang Li, North Carolina A & T University; Rahman Tashakkori at Appalachian State University; Sammie Carter, student at North Carolina State University; Melea Williams, graduate student at the University of North Carolina at Wilmington. In addition, many students gave project presentations to the class.

Professor Ian Foster, Argonne National Laboratory and University of Chicago kindly allowed me to use in my class a recorded presentation entitled "The Grid: Beyond the Hype" he made at Duke University in November 2004. He also kindly invited Dr. Ferner and me to give a presentation describing our re-designed 2007–2008 course at *Open Source Grid & Cluster Conference*, Oakland, CA, May 12–16, 2008.

I would also like to record my appreciation to Ron Vetter at the University of North Carolina at Wilmington who involved me in his large UNC-OP grant, which led to the collaboration with Dr. Ferner at University of North Carolina at Wilmington—without this collaboration, the work would not have been possible.

This book would not have been done without a chance meeting with Alan Apt at SIGCSE 2007 and without Randi Cohen, Computer Science Acquisitions Editor, Chapman and Hall/CRC Press/Taylor and Francis Group LLC, who quickly responded to my enquiry to write a book on Grid computing.

Barry Wilkinson
University of North Carolina
Charlotte