

## Education

# Cluster and Grid Computing: A Graduate Distributed-Computing Course

Rajkumar Buyya and Srikumar Venugopal • University of Melbourne

**R**apid developments in the capabilities of computing, storage, and networking components and their manufacturing techniques in the last two decades have enabled commoditization of high-performance computing. This situation has substantially lowered the entry barriers, in terms of price and complexity, for potential users to implement these components. High-performance computing systems developed using these commodity components, called clusters, have been used for solving resource-intensive problems in many domains. Clusters have become a key part of many enterprise IT systems and, in the case of Google and Amazon, the building blocks of their technology infrastructure. On a wider scale, clusters and other resources such as storage servers have been coupled to create Grids to enable distributed high-performance computing for scientific research and business applications.

So, there is an increasing demand for trained personnel to develop and manage such infrastructures and applications. Consequently, the Department of Computer Science and Software Engineering at the University of Melbourne offers a master's-level Cluster and Grid Computing course. This course focuses on the technologies to realize such high-performance network computing systems and on the programming models and tools for working with them.

### Quick Facts

**Course:** Cluster and Grid Computing (CSSE 433-678)

**Units:** Computer Science and Software Engineering

**Institution:** University of Melbourne

**Instructor:** Rajkumar Buyya

**Level:** Graduate

**URL:** [www.csse.unimelb.edu.au/678](http://www.csse.unimelb.edu.au/678)

## The scope of the course

The Cluster and Grid Computing course is designed for students with little or no experience in network computing. By the end of the course, we expect students to be familiar and comfortable with designing applications for cluster and Grid systems, to have a good knowledge of the challenges of working with these systems, and to be exposed to the research topics in this domain. This seemingly large gap is covered in 12 weeks through lectures, term paper and programming assignments, and a team-based project.

The course admits senior undergraduates (honors and final-year software engineering students) and master's students (mostly international students) from different programs. Owing to this diversity, ensuring that all students have a good understanding of the foundations of network and concurrent programming is important. So, we spend the first week on socket programming and multithreading. The rest of the course builds on this foundation, to focus on core topics in cluster and Grid computing.

Cluster-computing topics include

- an introduction to parallel systems,
- cluster architecture and single system image,
- parallel-programming paradigms,
- parallel programming with the message-passing interface (MPI), and
- resource management and scheduling in clusters.

Grid-computing topics include

- an introduction to Grids and Grid technologies;
- programming models and parallelization techniques;
- standard application development tools and paradigms such as message passing and parameter parallel programming;
- Grid security infrastructure;
- data management;
- resource management and scheduling in Grids;
- Grid economy;
- setting up a Grid, deploying Grid software and tools, and application execution; and
- application case studies.

Because no textbook covers all the course topics, we've developed a comprehensive set of lecture slides. We derived the cluster-computing part from the first few chapters of *High Performance Cluster Computing*.<sup>1</sup> We derived the Grid-computing part mostly from our own research material<sup>2</sup> and that of many international colleagues.<sup>3</sup> Most of these papers are available online; a pointer to them is on the course Web page (for the URL, see the "Quick Facts" sidebar).

## Assessment and projects

The first assessment strengthens the foundation by requiring students to develop a simple, multithreaded server in Java that serves requests of remote clients for mathematical operations. This exercise introduces students with little or no experience in network-based programming to sockets and multithreaded-programming concepts, which are fundamental to network computing. Students who have basic familiarity with the foundations are assigned advanced work such as representing the messages exchanged between client and server programs in XML.

Students receive their second assignment after they become familiar with clusters and their programming using MPI. They must create a program that multiplies two matrices using the master-worker paradigm. During this assignment, they learn about important parallel-programming concepts such as dividing work between processes and managing intertask communication. The students execute their MPI programs on a shared computing cluster managed by the Grid Computing and Distributed Systems (GRIDS) Laboratory at the university, and obtain performance results.

The third assignment involves writing a term paper on an emerging research topic or surveying emerging trends in cluster and Grid computing. This team-based exercise aims to teach students to critically analyze existing systems and their characteristics. A typical paper contains an introduction to the problem and motivations, a description of current research and production systems attempting to tackle the problem, and a qualitative analysis and comparison of the systems that points out the strengths and weaknesses of the proposed solutions. Recent term papers have covered such topics as Grid resource brokers, thread-based applications on desktop Grids, distributed file systems, Grid programming models, and cluster and Grid resource information systems.

These three assignments constitute one-fourth of the course grade. Another fourth comes from the project that the students undertake during the final weeks of the course. The students form teams of three or four and are assigned a single, defined objective. This objective usually involves extending an existing system to gain new features or to tackle a problem different from what that system was

designed for. The project is related to the term paper topic; both of them are often tackled by the same team. This lets the students apply the lessons learned from their analysis to the design of their project. At the end of the project, the students present a demonstration and a comprehensive report on their experience. The project aims to tackle software engineering problems in network-based high-performance computing through teamwork and to impart some knowledge of research activities. Some recent projects have been to design a Grid resource broker for specific objectives, implement a bulk-synchronous-parallel model on desktop Grids, and implement a simple GUI-based Grid programming environment.

At the end of the course, the students take a written exam testing their knowledge and familiarity with cluster and Grid computing concepts. This exam constitutes the other half of their grade.

The assignments let students gain not just a theoretical overview but also practical experience with cluster and Grid computing systems. This means that at the end of the course, students will be able to effectively manage and utilize network and high-performance computing systems for the organization in which they're employed.

## **The evolution of the course**

The course, originally called "Advanced Topics in Computer Science," debuted in the second semester (July–Nov.) of 2002. The following year, we changed its title to the current one and shifted it to the first semester (Feb.–June), where we've offered it every year since. The class has increased from 22 students in 2002 to 67 in 2007.

Originally, each team worked on a different term paper and project, which we derived from activities that GRIDS Lab members identified as ancillary outcomes of their research work. Each team reported to a lab member, who provided regular feedback in terms of design and code reviews. In return, that lab member could pursue interesting side projects that complemented his or her main research. This led to the creation of several new Grid tools and applications developed using Gridbus middleware technologies.

However, to deal with the increasing class size, we now have each team work on the same project. We assign grades on the basis of the teams achieving a mandatory baseline and an optional enhanced set of features. For example, in the current iteration of the course, the project is to create a simple, Java-based Grid metascheduler for submitting jobs to resources with low-level Grid middleware based on the Web Services Resource Framework. We assign this baseline set of features:

- work with Globus 4.0 middleware;
- accept job requests from users continuously;
- discover appropriate resource information dynamically; and
- submit, query, and terminate jobs on these resources.

This set simplifies supporting a large number of students working with Grid middleware and evaluating their results. Possible enhanced features include a visual environment for the metascheduler, scheduling algorithms such as shortest job first and earliest deadline first, and priority queues based on job budgets.

This experience introduces students to the challenges of engineering software for network and high-performance computing systems. They're able to incorporate concepts such as fault tolerance and load balancing into their system designs. Because interacting with Grid middleware is still novel for the students, the teaching team created a library of common functions such as job submission, resource querying, and job monitoring. This lets students concentrate on designing and implementing the metascheduler and the features.

## The relationship with the GRIDS Lab

The GRIDS Lab was established in 2002, around the same time that this course debuted. The lab focuses exclusively on next-generation technologies for network computing systems. The synergy between the aims of the GRIDS Lab and the Cluster and Grid Computing course has enabled many students in the course to participate in the activities of the lab. Students can use the lab infrastructure, which consists of an Intel Xeon cluster and several server machines running Grid middleware, for their projects. As we mentioned previously, they also execute their MPI assignments on the cluster.

Students who have excelled in their term papers and projects have received scholarships to continue working on these during the vacation at the end of the semester. The goal is for them to produce work publishable in a peer-reviewed conference or journal, or as a chapter in an edited book on a related research area. This approach has led to the publication of four journal papers, nine conference papers, and three book chapters. This has enhanced the suitability of such students as prospective PhD candidates; a few of them have even joined the GRIDS Lab to pursue further research. Others are working in global companies such as IBM, local small and medium enterprises such as Intrepid Geophysics, and research organizations such as the Institute for High-Performance Computing in Singapore. Even better, some of them have started their own companies developing network and Web-based systems and applications.

The rapid growth of the course signals not only its success but also the growing importance of the topic and of the skills needed for developing network-based high-performance enterprise-computing systems and applications. The course has been at the forefront of the field and has served as a model for other courses at our university and elsewhere. Its success and the growing demand for Internet and distributed-computing skills in many industries have led to a completely new master's program dedicated to distributed computing.<sup>4</sup>

As we stated earlier, we've developed our own lecture material derived primarily from research papers and edited research books. Such material can be taught primarily by teachers who themselves are researchers in this field. So, we're developing a textbook so that teachers who aren't researchers can easily teach such courses, and we encourage others to do so too.

## References

1. R. Buyya, ed., *High Performance Cluster Computing*, vols. 1 & 2, Prentice Hall, 1999.
2. R. Buyya, "Economic-Based Distributed Resource Management and Scheduling for Grid Computing," PhD thesis, Monash Univ., 2002.
3. I. Foster and C. Kesselman, eds., *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999.
4. R. Buyya and R. Kotagiri, "An Innovative Master's Program in Distributed Computing," *IEEE Distributed Systems Online*, vol. 8, no. 1, 2007; <http://doi.ieeecomputersociety.org/10.1109/MDSO.2007.1>.



**Rajkumar Buyya** is an associate professor and reader of computer science and software engineering at the University of Melbourne. He's also the director of the Grid Computing and Distributed Systems Laboratory at the university. He serves as the chair of the IEEE Technical Committee for Scalable Computing. Contact him at [raj@csse.unimelb.edu.au](mailto:raj@csse.unimelb.edu.au); [www.buyya.com](http://www.buyya.com).

**Srikumar Venugopal** is a research fellow in the Department of Computer Science and Software Engineering at the University of Melbourne. He's a founding member of the Grid Computing and Distributed Systems Laboratory at the university and leads research in data grids, market-based resource management, and workflow scheduling. Contact him at [srikumar@csse.unimelb.edu.au](mailto:srikumar@csse.unimelb.edu.au).

## Related Links

- *DS Online's Software Engineering Community*, [http://dsonline.computer.org/software\\_engineering](http://dsonline.computer.org/software_engineering)
- "Teaching Condor Grid Computing to Beginning Programming Students," *IEEE Distributed Systems Online*, <http://doi.ieeecomputersociety.org/10.1109/MDSO.2007.23>
- "Teaching Grid Computing in North Carolina: Part I," *IEEE Distributed Systems Online*, <http://doi.ieeecomputersociety.org/10.1109/MDSO.2006.42>
- "Teaching Grid Computing in North Carolina: Part II," *IEEE Distributed Systems Online*, <http://doi.ieeecomputersociety.org/10.1109/MDSO.2006.46>

### **Cite this article:**

Rajkumar Buyya and Srikumar Venugopal, "Cluster and Grid Computing: A Graduate Distributed-Computing Course," *IEEE Distributed Systems Online*, vol. 8, no. 12, 2007, art. no. 0712-oz002.