Learning and Teaching Grid Computing

Rahul Thakur, Suruchi Gondkar, John Rose Santiago and Fabian T. R. Barreto

***Abstract*—Academic institutions developed the first-generation technologies and architectures that formed the basis of grid computing. Presently grid protocols and technologies are being adopted in many industrial and government environments. Grid technology involves large-scale computations, virtualization of resources and open, standard interfaces for enabling interoperability among systems. Grids present the opportunity to aggregate, share, compute and store data across geographically dispersed computer systems, and therefore increase the utilization of the latent resources. The quest continues as academic institutions, scientific organizations and commercial firms attempt to build a secure, reliable and robust Grid infrastructure and reengineer a simpler middleware solution. This paper introduces the concepts and components of the Grid, its services and the effective grid computing courses in academic institutions essential to the training of students who will be the builders of the next-generation grid.**

***Keywords* — Globus Toolkit, Grid computing, Grid services, Service-oriented architecture, Virtualization paradigm.**

I. Introduction

OVER the past three decades, computer engineering has emerged as a distinct discipline. Computer engineering exemplifies the technology of design, construction, implementation, and maintenance of the hardware and the software components of computing systems. However, the characteristic of this discipline is such that as it grows more and more mature, the body of knowledge compounds daily and this poses a problem in deciding and defining what will go into the recommended curriculum to produce computer engineers competent to further the technology and its applications. Surely, the curriculum cannot contain everything; future computer engineers must be equipped with essential knowledge and well-tested methods and techniques, not just transient technologies. The computer engineering curriculum [1], by Computer Curricula 2001, indicated "the Grid" as an important emerging technical area pointing to further developments in computer engineering. It mentioned that Grid infrastructure would provide various kinds of computing power as well as an information infrastructure and associated networking capability that would support many aspects of future activities of research, science, government and industry. Grid computing which builds on knowledge of Web services technologies, distributed computing, network security, and network architecture and programming is often referred to as the next revolution after the Internet and the World Wide Web [2].

The paper is organized in the following way. In Section II some basic background about the grid is given. Section III explains the evolution of the grid. The grid architecture with its different layers is briefly described in Section IV. Then, in Section V, some of the Grid Technology standards are introduced. Section VI deals with two models used for teaching grid computing in academic institutions. Finally, some concluding remarks are given in Section VII.

II. What is the Grid?

The term, "the Grid" was coined by Ian Foster in the mid-1990s to denote a proposed distributed computing infrastructure for advanced science and engineering. The metaphor "grid" came from power grids, where it was possible to access electrical current in a seamless way using a standardized plug. Ian Foster, Carl Kesselman and Steve Tuecke, widely regarded as the "fathers of the grid" brought together the ideas of the grid, distributed computing, object oriented programming, cluster computing, web services to give a blueprint [3] of the grid.

According to Foster's Grid checklist [4], the Grid is a system that is able "to coordinate resources that are not subject to centralized control", use "standard, open, general-purpose protocols and interfaces" and finally "to deliver nontrivial qualities of service". The coordinated resources sharing is a direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource brokering strategies emerging in industry, science, and engineering. A set of individuals and/or institutions defined by sharing rules form a virtual organization (VO). The virtualization paradigm provides the end users with a logical, unified view of the resources and services in the Grid.

Grid technologies enable the sharing, exchange, discovery, and aggregation of resources like data, processors, storage and scientific devices across geographically distributed sites. Grid computing [5] provides highly scalable, highly secure, and high performance mechanisms for negotiating access to remote computing resources in a seamless manner.

III. The Evolving Grid

Grid computing has evolved with the explosive growth in microelectronics, computer networks, internet technology, and software development. The driving force behind the formation of grids was the need of academic communities for collaborative research. The research centres achieved a

Rahul Thakur is with Dept. of Information Technology, Suruchi Gondkar and John Rose are with Dept. of Computer Engg., and Fabian Barreto is with Dept. of Electronics and Telecommunication Engg., at Xavier Institute of Engineering, Opp. Raheja Hospital, Mahim, Mumbai 400 016. Tel: (022)24451961, 24469670, e-mails: rahthakur@xavierengg.com, suruchi. gondkar @ xavierengg.com, johnrose@xavierengg.com, fabbar@ xavierengg.com

more refined and informative level of quality in collaboration by efficiently sharing vast amounts of information and computational resources. Managing access, processing, distribution and storage of the data were some of the challenges of these environments. Institutions such as the Globus Alliance, and the e-Science Grid core program were some of the first to incubate and grow grid solutions to maturity, preparing them for commercial adoption.

We can identify three generations [6], which led to the evolution of grid systems. The first generation systems from the academic institutions were the forerunners of the Grid. The second generation focused on middleware to support large-scale data and computation. The present third generation systems is concerned with distributed global collaboration and information layer issues, and follows a service oriented approach.

The objective of the early grid environments was to provide computational resources to a range of high performance applications. Two representative projects were FAFNER (Factoring via Network-Enabled Recursion) and I-WAY (Information Wide Area Year) in 1995. The second generation core technologies were Globus, Legion, Jini, Nimrod, P2P (Peer-to-Peer) and UNICORE (UNIform Interface to COmputer REsources). The third generation of grid initiatives that implement a more holistic view of grid computing, deal with autonomic computing, business-on-demand and infrastructure virtualization, service-oriented architecture and semantic grids.

IV. Grid Architecture

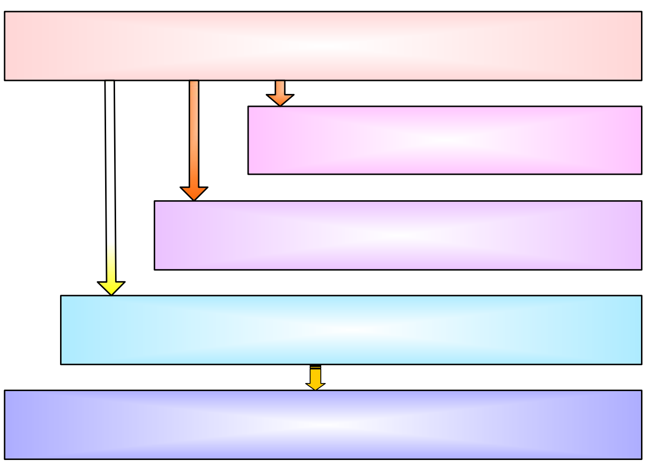

Fig. 1. Grid architecture

The Grid architecture, as described in Fig. 1, defines common mechanisms, interfaces and protocols at each layer, by which users and resources can negotiate, establish, manage, and share resources [7]. It is a standards based protocol architecture, which facilitates extensibility, interoperability, portability, and code sharing. The protocols, services, application programming interfaces (API), and software development kits (SDKs) are categorized according to their roles in enabling resource sharing.

The grid components are arranged into five layers: Fabric, Connectivity, Resource, Collective and Application. Each layer builds on the services offered by the lower layer in addition to interacting and co-operating with components at the same level. The *Fabric* layer defines the shared resources. It implements enquiry mechanisms that permit discovery of their structure, state, and capabilities, and resource management mechanisms that provide some control of delivered quality of service. The *Connectivity* layer defines core communication and authentication protocols required for Grid-specific networking service transactions. The *Resource* layer defines protocols for the secure negotiation, initiation, monitoring, control, accounting, and payment of sharing operations on individual resources. The *Collective* layer is responsible for all global resource management and interaction with a collection of resources. The *Application* layer comprises of user applications constructed by utilizing the services defined at each lower layer.

V. Grid Technology Standards

Standardisation [8] developed with the expansion of Grid computing. This ensured interoperability and enhanced integration. The Global Grid Forum (GGF), presently known as the Open Grid Forum (OGF), led the global standardisation effort and produced standards for almost all aspects of Grid technology. The OGF, a community of users, developers, and vendors, follow open standards for grid software interoperability. Within this Forum, an ongoing initiative which was started in 2006 attempts to consolidate a Grid Education and Training Community Group (ET-CG) with the goal of sharing and developing best practice in grid-related education and training.

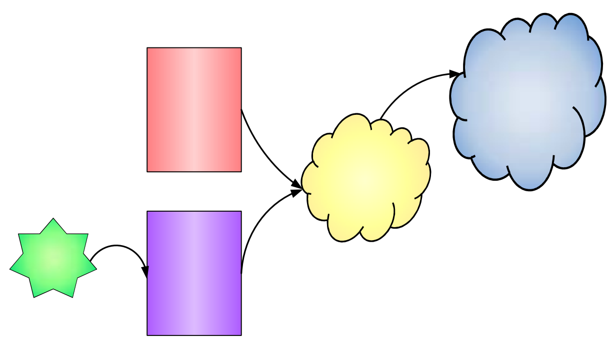

Fig. 2. Relationships between OGSA, OGSI, Grid Services, Web Services, and the Globus Toolkit GT3.

OGF developed the Open Grid Services Architecture (OGSA) and created the Open Grid Service Infrastructure (OGSI) specification and its successor, the Web Services Resource Framework (WSRF), which led to integration of Web Services within Grid architecture. Grid Services were built on Web Services, using the Web Service Description Language (WSDL), Simple Object Access Protocol (SOAP) and Web Services security layer. Web services (WS) provide flexible, extensible, and widely adopted XML (eXtensible Markup Language) based mechanism for describing, discovering, and invoking network services. The relationships between OGSA, OGSI, Grid Services, Web Services, and the Globus Toolkit GT3 is shown in Fig. 2.

The Globus Toolkit [9] is a community-based, open-architecture, open-source set of services and software libraries that support Grids and Grid applications. The toolkit includes software for security, information infrastructure, resource management, data management, communication, fault detection, and portability. The core services, interfaces and protocols in the Globus toolkit allow users to access remote resources seamlessly while simultaneously preserving local control over the resources.

The Globus architecture [10] has three main groups of services accessible through a security layer, namely: Information infrastructure, Data management and Resource management. The *Security layer* uses Generic Security Services Application Programming Interface (GSS-API) which is a security infrastructure interface independent of programming language and underlying security mechanism. Grid Security Infrastructure (GSI) is used as the security mechanism, which is based on Secure Sockets Layer (SSL), Public Key Infrastructure (PKI) and X.509 Certificate Architecture. The *Information infrastructure* includes the MDS (Globus Monitoring and Discovery Service), and a Lightweight Directory Access Protocol (LDAP)-based index service: the Grid Resource Information Service (GRIS) allows querying resources for their current status, while the Grid Index Information Services (GIIS) knits together arbitrary GRIS services. *Data management* is done though Grid File Transfer Protocol (GridFTP), a management protocol for data access, or the Reliable File Transfer Service (RFT) for GT3. For *Resource management* GT2 uses the Grid Resource Allocation Management (GRAM) for allocation of computational resources and for monitoring and control of computation on those resources while GT3 uses a collection of OGSI services, MJS, MJFS and MMJFS (the Master and Managed Job Factory Service), whose task is resource allocation, submit jobs, and manage job status and progress.

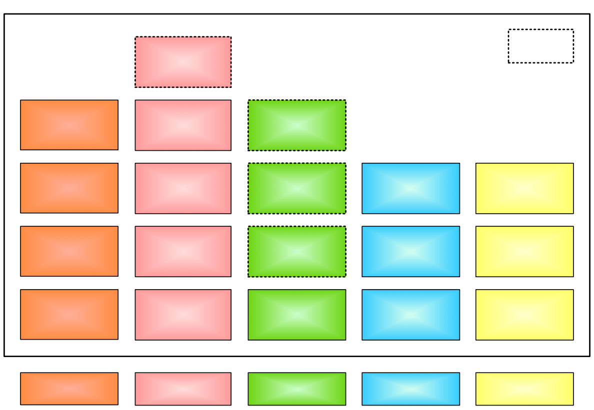

Fig. 3. Primary GT4 components

Version 4 of Globus Toolkit (GT4) was released in April 2005. In Fig. 3 the primary GT4 components [11] are shown. The principal GT4 elements include Service-oriented architecture, Infrastructure services, Web services, GT4 containers, Security and Standards. With GT4, the main interface switched from OGSI to WSRF, as extensive use was made of Web services to define the interfaces and structure the components. Web Service standard provides a platform-independent method for messaging-based interaction of applications. Web services architecture consists of protocol used to transport messages (typically HTTP), message encoding (SOAP), and interface description (WSDL). A client interacts with a Web service by sending it a SOAP message; the client may subsequently receive response message(s) in reply. In GT4 remote job execution is supported by GRAM4 service, which defines mechanisms for submitting requests to execute jobs (defined in a job description language) and for monitoring and controlling the resulting job executions.

GT4 service-oriented architecture (SOA) supports applications in which sets of services interact via standard protocols. A Service-Oriented Architecture is a design model for linking computational resources, data and applications to perform services and deliver results to service consumers. Globus Toolkit v4 software has components that can be used to construct GT4 containers for hosting Web services written in Java, C, and Python.

VI. Teaching Models

Jens Mache and Amy Apon [12], describe topics, exercises, and experiences of teaching grid computing to graduate and advanced undergraduate computer science students at two different universities in the United States of America. They identify five knowledge areas for grid computing: Cluster and Grid Administration, Security, Grid Programming, Grid Usage and Application and Parallel Computing. *Cluster and Grid Administration* deals with the installation and configuration of the cluster and the administration of a grid development and execution environment. *Security* is concerned with the use of public key encryption and certificates. *Grid Programming* looks into the skills required to develop a service and to use tools and languages to develop grid infrastructure and for grid-enabling applications. *Grid Usage* prepares the students to use commands and tools to submit and execute different grid applications and to locate, access, and manage data repositories and other grid resources. *Application and Parallel Computing* deals with the use of tools to partition applications or the use of parallel programming languages and libraries to develop and execute grid applications.

They used the Open Grid Service Infrastructure standard Globus Toolkit 3 (version 3.0.1), Java 2 Platform Standard Edition (j2sdk-1.4.2), Open Source Cluster Application Resources (OSCAR version 2.2.1), A grid enabled Message Passing Interface (MPICH-G2 version 1.2.5-1), and Red Hat Linux suite of software packages. The students were also directed to the online training and grid development materials at the Globus website.

The authors stated that sophisticated knowledge of Linux was required for implementing security, for configuring the network, for locating various configuration files, and for installing software, little of which was taught in a typical undergraduate curriculum. This difficulty can be circumvented by forming Research or Hobby Clubs where undergraduate students can experiment and learn Linux. This project of training students in Linux has been successful at Xavier Institute of Engineering.

Bina Ramamurthy [13] describes a comprehensive model to learn and teach grid computing. Here project entitled Grid For Research, Collaboration and Education (GridFoRCE) was initiated and developed by the Department of Computer Science and Engineering, University at Buffalo, with the primary objective of spreading grid awareness and improving the technical preparedness of the current and future scientific work force. She also maintains an excellent website on the many universities all over the world who have implemented courses in grid computing [14].

They organized GridFoRCE into five major activities: Adapting grid technology (architecture, protocol, services, and applications) to undergraduate curriculum through a series of laboratory exercises. Building laboratory prototypes that supported grid application development. Conducting workshops for strengthening local industry work force, Completing a detailed assessment of the project so that the adaptation, implementation, progress, and the outcome are measured and appropriate actions taken to make improvements. Disseminating adaptation details (course plans, lab descriptions, and implementation), project experience, and assessment methods to educator and developer communities.

The students were taught to design, implement and test the grid services, and to finally deploy the integrated system. Their laboratory infrastructure consisted of two different experimental research and development grids. The first grid was a 40 Sun Microsystem's Sparc 4 with an Ultra Sparc 5 as front-end gatekeeper, all running Solaris 8.0 operating system and the Ultra Sparc running Condor 2.0 grid software. The second had newer Dell blades 1650 hardware, a combination of FreeBSD (Berkeley Software Distribution) and Red Hat Linux 9.0 operating systems, all running Globus Toolkit 3.0.2. FreeBSD was chosen as an operating system for the utility server, designed to provide network gateway/firewall services and basic Unix-level account authentication and file services. The three computing nodes had RedHat-9 Linux operating system and Globus GT3. The Lab sessions for the CSE486 course consisted of Webservices, Grid Infrastructure and Grid Programming. The second course, CSE487, involved applying grid technology to solve problems in specific application domains.

She mentions an interesting point as to how they used discarded computers (originally used for graduate students desktops) to set up the grid clusters. This is a big advantage for academic institutions to extend the use of their resources and at the same time experiment in grid computing.

VII. Conclusion

We have introduced the key concepts of grid computing, the associated grid and web services and the role that academic institutions can play in training the students in grid computing. The challenge of teaching grid computing in academic institutions is that standards, frameworks, implementations and applications are evolving rapidly. However, good methodologies help in setting valuable pedagogical paradigms. The Grid Technology is a crucial enabling technology to achieve the productivity and growth challenges for our world today. As grid computing makes colossal processing power on-demand widely available, it offers a whole new generation of services for society. In addition, science and industry can look forward to solving problems never previously imagined. A global grid infrastructure is developing to be readily available in the near future. It will make computing available as a utility just as one accesses the telephone grid for voice communication. Hence, the next generation of computer engineers need to prepare for a technological workplace where grid computing will play a key role. The academic institutions can surely show the way by guiding the students and preparing them to be the builders of the next-generation grid.

VIII. Acknowledgment

The authors would like to thank the reviewers of this paper for their careful reading, critique and insightful comments.

IX. References

[1] McGettrick A., Theys D. M., Soldan L. D., and Srimani K. P., "Computer Engineering Curriculum in the New Millennium" *IEEE Trans. Educ*., vol. 46, no. 4, pp. 456-462, Nov. 2003.

[2] Touriño Juan, "Grid Education and Grid-Based Technologies Applied to Education: Ongoing Activities," *IEEE Trans. Educ*., vol. 50, no. 1, pp. 1-2, Feb. 2007.

[3] Foster I., Kesselman C. (editors), *The Grid: Blueprint for a New Computing Infrastructure* (2nd edition), Morgan Kaufmann, 2003.

[4] Foster I., "What is the Grid? A Three Point Checklist", *http://www.fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf*

[5] Joseph J., Fellenstein C., *Grid Computing*, Delhi: Pearson Education, 2004.

[6] De Roure D., Baker M. A., Jennings N. R. and Shadbolt N. R., "The Evolution of the Grid", http://www.semanticgrid.org/documents/ evolution / evolution.pdf

[7] Foster I., Kesselman C., and Tuecke S., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", *International Journal of High Performance Computing Applications*, vol. 15, pp. 200-222, London: Sage Publishers, UK, 2001.

[8] IBM, Grid Computing: Past, Present and Future — An Innovation Perspective. Available: *http://www.ibm.com/grid*

[9] Foster I., Kesselman C., Nick J., and Tuecke S., The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Available: *http://www.globus.org/research/ papers/ogsa.pdf*

[10] Wolfgang L., Shahrzade M., Jørn I., Bent Ø., Bjørn N, "Grid and Related Technologies", *Norwegian Computing Center*, 2004, *http://www.nr.no*

[11] Foster I., "Globus Toolkit Version 4: Software for Service-Oriented Systems", *Journal of Computer Science and Technology*, vol. 21, no. 4, 513–520, July 2006.

[12] Mache J. and Apon A., "Teaching Grid Computing: Topics, Exercises, and Experiences", *IEEE Trans. Educ*., vol. 50, no. 1, pp. 3-9, Feb. 2007.

[13] Ramamurthy B., "GridFoRCE: A Comprehensive Resource Kit for Teaching Grid Computing", *IEEE Trans. Educ*., vol. 50, no. 1, pp. 10-16, Feb. 2007.

[14] Ramamurthy B. Available: http://www.cse.buffalo.edu/~bina/ resources.html

X Biographies

Rahul Thakur received the B.E. in Computer Engineering from Mumbai University, India in 2007.

He is currently a Lecturer at Xavier Institute of Engineering.

His main research interests are in grid computing, data mining and warehousing, web technologies and game designing.

Suruchi Gondkar received her B.E. from Shivaji University, India in Computer Science Engineering in 2003.

She is currently a Lecturer at Xavier Institute of Engineering.

Her main research interests are in parallel processing, distributed computing, system programming and web technologies

John Rose Santiago received his M.Sc. (Electronics) degree from Madras University in 1994.

He is the Systems Manager at Xavier Institute of Engineering. He has set up the Xavier's Linux Research Lab, at Xavier Institute of Engineering, where he has put together some free-and-open-source software to meet the requirements of Engineering Colleges.

His research interests include computer networking, distributed systems, and free-and-open-source Software based systems.

Fabian T.R. Barreto (M'07) received his B. E. degree in Electronics and Telecommunication Engineering from Mumbai University, India in 1991. He is currently working toward his M. E. degree in Electronics under Mumbai University.

His industry experience includes working with Zenith Computers, Mandovi Pellets Limited, both in India, and working on UMS ships for Neptune Orient Lines, Singapore. Currently he is the Administrator at Xavier Institute of Engineering and teaches Digital Communication.

His research interests include grid computing, computer networking, sensor networks, digital communications over wireless channels and information theory.