academicJournals

Vol. 6(7), pp. 144-148, July, 2013
DOI : 10.5897/AJMCSR11.154
ISSN 2006-9731 © 2013 Academic Journals
http://www.academicjournals.org/AJMCSR

African Journal of Mathematics and Computer Science Research

Review

Literature review on grid computing

Rahul Kumar[1]*, I. A. Khan[2] and V. D. Gupta[2]

[1]Institute of Environment and Management, Lucknow, Uttar Pradesh, India.
[2]Integral University, Lucknow, Uttar Pradesh, India.

Accepted 21 November, 2012

The concept of grid has emerged as a new approach to high performance distributed computing infrastructure. In general, Grids represent a new way of managing and organizing computer networks and mainly their deeper resource sharing. Grid computing has evolved into an important discipline within the computer industry by differentiating itself from distributed computing through an increased focus on resource sharing, coordination, manageability and high performance.

Key words: Computing infrastructure, grid computing, resource sharing, computer networks.

INTRODUCTION

The concept of grid computing originated in the early 1990s as a metaphor for making computer power as easy as to access an electric power grid. A computational grid is a collection of heterogeneous computers and resources spread across multiple administrative domains with the intent of providing users easy access to these resources. Technically speaking, Grid computing enables the virtualization of distributed computing and data resources such as processing, network bandwidth and storage capacity to create a single system image, granting users and applications seamless access to vast information technology (IT) capabilities. Just as an Internet user views a unified instance of content via the web, a grid user essentially sees a single, large virtual computer.

Grid computing pools processing cycles from multiple computers to maximize capacity, memory, power and other resources distributed across multiple systems. The concept of grid describes a framework in which heterogeneous and distributed computational, networking, memory and storage resources can be linked to serve the needs of particular user applications.

Grid computing has matured as it has moved from the realm of experimentation to mainstream technology. Today, there is much more to successful grid computing infrastructure than merely deploying racks of fast (or inexpensive) servers. Long embraced by scientific and educational organizations for its considerable impact on productivity and competitiveness, grid computing is now increasingly essential mainstream technology in the enterprise for a wide range of critical applications. These high-performance computing (HPC) applications are often at the very center of the organization’s mission or business, and asking the right questions and getting answers faster than the competition can directly impact key drivers such as quality, time-to-market, and product safety.

By pooling resources and applying very large amounts of compute power to their most strategically-vital tasks, grid computing lets organizations improve agility as they focus on their key priorities.

LITERATURE REVIEW

Recently, lot of work has appeared in the literature on the problems of the computational grid. A variety of problems have been studied which range from estimating capacity limits, optimal routing, and queuing behavior for grid computing to distributed molecular modeling for drug design on the World-Wide Grid. The summary of the articles published in the last 10 years is cited here under

*Corresponding author. E-mail: kumarahul971@gmail.com.

in chronological order.

Krauter and Maheswaran (1997) proposed a grid architecture that is motivated by the large-scale routing principles in the Internet to provide an extensible, high-performance, scalable, and secure grid. Central to the proposed architecture is middleware called the grid operating system (GridOS). This paper describes the components of the GridOS. The GridOS includes several novel ideas (i) a flexible naming scheme called “Gridspaces", (ii) a service mobility protocol, and (iii) a highly decentralized grid scheduling mechanism called the router-allocator.

Casanova (1998) introduced distributed computing research issues in grid computing. Ensembles of distributed, heterogeneous resources, or computational grids, have emerged as popular platforms for deploying large-scale and resource-intensive applications. They have focused on issues concerning the dissemination and retrieval of information and data on computational grid platforms.

Hernandez et al. (1999) described an approach to simplifying the development and deployment of applications for the grid. Their approach aims at hiding accidental complexities (for example, low-level grid technologies) met when developing these kinds of applications. To realize this goal, the work focuses on the development of end-user tools using concepts of domain engineering and domain-specific modeling, which are modern software engineering methods for automating the development of software.

Zhang and Wang (2000) presented a semantic grid infrastructure for distributed management of e-government resources across ubiquitous virtual governmental agencies. An ontology-based service-oriented approach to problem-solving in e-government is proposed in the semantic grid, enabling to provide, in an open, dynamic, loosely coupled and scalable manner, the service publication, discovery and reuse for connecting the customers and agencies of e-government services based on their semantic similarities in terms of problem-solving capabilities. The operation of the system is demonstrated using Protégé-2000, a widely accepted ontology-modeling tool to validate the implementation of the proposed approach towards effective ontological maintenance.

Ian et al. (2001) reviewed the “Grid problem,” which is defined as flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources-what we refer to as virtual organizations. In such settings, we encounter unique authentication, authorization, resource access, resource discovery, and other challenges. It is this class of problem that is addressed by grid technologies. They have also presented an extensible and open grid architecture, in which protocols, services, application programming interfaces, and software development kits are categorized according to their roles in enabling resource sharing. They have also discussed how grid technologies relate to other contemporary technologies, including enterprise integration, application service provider, storage service provider, and peer-to-peer computing.

Klaus et al. (2002) described the resource management system as the central component of distributed network computing systems. There have been many projects focused on network computing that have designed and implemented resource management systems with a variety of architectures and services. In this paper, an abstract model and a comprehensive taxonomy for describing resource management architectures is developed. The taxonomy is used to identify approaches followed in the implementation of existing resource management systems for very large-scale network computing systems known as grids. The taxonomy and the survey results are used to identify architectural approaches and issues that have not been fully explored in the research.

Fran et al. (2002) defined grid as the computing and data management infrastructure that will provide the electronic underpinning for a global society in business, government, research, science and entertainment. Grids integrate networking, communication, computation and information to provide a virtual platform for computation and data management in the same way that the Internet integrates resources to form a virtual platform for information. The grid is transforming science, business, health and society. Grid infrastructure will provide us with the ability to dynamically link together resources as an ensemble to support the execution of large-scale, resource-intensive, and distributed applications.

David et al. (2003) described the evolution of grid systems, identifying three generations: first generation systems which were the forerunners of the grid as we recognize it today; second generation systems with a focus on middleware to support large-scale data and computation; and third generation systems where the emphasis shifts to distributed global collaboration, a service oriented approach and information layer issues. They have discussed the relationship between the grid and the World Wide Web, and suggest that evolving web technologies will provide the basis for the next generation of the grid.

Geoffrey (2002) described the relevance of grids in education. The implication of grids and web services for any organization - we call this an Enterprise to stress the importance of the goal and application specific features in any grid deployment have also been discussed by him. The importance of grid in education where our organization involves learners, teachers and other stakeholders such as parents and employers is implied.

Craig and Domenico (2002) discussed issues, properties and capabilities of grid programming models and tools to support efficient grid programs and their effective development. Grid programming must manage computing environments that are inherently parallel, distributed, heterogeneous and dynamic, both in terms of the resources involved and their performance. Grid applications want to dynamically and flexibly compose resources and services across that dynamic environment. While it may be possible to build grid applications using established programming tools, they are not particularly well suited to effectively manage flexible composition or deal with heterogeneous hierarchies of machines, data and networks with heterogeneous performance.

Cao et al. (2003) proposed a workflow management system for grid computing, called GridFlow, including a user portal and services of both global grid workflow management and local grid sub-workflow scheduling. Simulation, execution and monitoring functionalities are provided at the global grid level, which work on top of an existing agent-based grid resource management system. At each local grid, sub-workflow scheduling and conflict management are processed on top of an existing performance prediction based task scheduling system. A fuzzy timing technique is applied to address new challenges of workflow management in a cross-domain and highly dynamic grid environment. A case study is given and corresponding results indicate that local and global grid workflow management can coordinate with each other to optimize workflow execution time and solve conflicts of interest.

Figueiredo and Dinda (2003) advocated a novel approach to grid computing that is based on a combination of “classic” OS level virtual machines (VMs) and middleware mechanisms to manage VMs in a distributed environment. The abstraction is that of dynamically instantiated and mobile VMs that are a combination of traditional OS processes (the VM monitors) and files (the VM state). They have given qualitative arguments that justify their approach in terms of security, isolation, customization, legacy support and resource control, and have showed quantitative results that demonstrate the feasibility of our approach from a performance perspective. They have described the middleware challenges implied by the approach and architecture for grid computing using VMs.

Ewa et al. (2003) discussed that grid computing provides key infrastructure for distributed problem solving in dynamic virtual organizations. It has been adopted by many scientific projects, and industrial interest is rising rapidly. However, grids are still the domain of a few highly trained programmers with expertise in networking, HPC, and OSs. This paper describes our initial work in capturing knowledge and heuristics about how to select application components and computing resources, and using that knowledge to generate automatically executable job workflows for the grid. Our system is implemented and integrated with a grid environment where it has generated dozens of workflows with hundreds of jobs in real time. The paper also discusses the prospects of using AI to improve current grid infrastructure.

Santos and Koblitz (2005) presented an interface for metadata access on the grid, designed to support flexible schema management, efficient retrieval of large result sets and to allow a broad range of implementations. They have also described an implementation of this interface, which supports a wide range of storage backends and two access protocols: Simple Object Access Protocol (SOAP) and a Transmission Control Protocol (TCP)-streaming based protocol.

Xiyan and Runtong (2005) described grid as one of the core technologies of the next-generation Internet. The essences of grid are resource sharing and service collaboration in virtual organization. This paper introduces some concepts and issues about grid computing. In the first part, the paper briefly describes the definition and some background knowledge of grid computing, and then presents its current and potential application, and in the end, it discusses the future trends in grid computing research and application.

Talia (2005) proposed a feasible approach for the development of OS facilities to integrate machines in grids. They have proposed a two-step strategy that in a first stage aims to integrate existing OSs with a grid through an interface based on a grid VM (GVM); in this phase, Linux can be used as a use case for developing a Gridified OS. The second stage should provide different versions of the GVM that can be configured on different devices implementing grid nodes.

Parastatidis et al. (2005) gave the fundamental tenets of service-oriented architecture (SOA) and their relevance to Internet-scale computing (or grid computing). SOA is the contemporary paradigm of choice for developing scalable, loosely coupled applications that span organizations. They have showed how to apply the principles of SOA to building Internet-scale applications using Web Services technologies and how to avoid software pitfalls by adhering to a number of deliberately simple architectural constraints.

An Oracle Technical White Paper presented on March, 2005 states that “Forrester Research reports that 37% of enterprises are piloting, rolling out, or have implemented some form of grid computing. IDC identifies grid computing as the fifth generation of computing, after client-server and multi-tier. Leading businesses, such as Dell and the Chicago stock exchange, have begun deploying enterprise grids. Grid computing has increased momentum as the enterprise IT architecture of choice. Grid computing is a new IT architecture that produces more resilient and lower cost enterprise information systems. With grid computing, groups of independent, modular hardware and software components can be connected and rejoined on demand to meet the changing needs of businesses.”

Sulistio et al. (2006) presented their work on a data grid simulation infrastructure as an extension to GridSim, a well-known computational grid simulator. The extension provides essential building blocks for simulating data grid scenarios. Since it is not possible to test many different usage scenarios on real data grid test beds, it is easier to use simulation as a means of studying complex scenarios. This technology is highly anticipated by scientific communities, such as in the area of astronomy, protein simulation and high-energy physics. Data grids are an emerging new technology for managing large amounts of distributed data. This is because experiments in these fields generate massive amount of data, which need to be shared and analyzed.

The Replica Catalogue (RC), a core entity of data grids, is an information service, which provides registry and look up service. Therefore, the work provides the possibility to organize RC entities in different ways, such as in centralized or hierarchical model. It provides the ability to define resources with heterogeneous storage components, and the flexibility to implement various data management strategies. Moreover, users and/or resources are able to query to a RC about the location of a particular file. It also allows authorized users to share, remove and copy files to resources.

Agustin et al. (2007) proposed an autonomic network-aware scheduling infrastructure that is capable of adapting its behavior to the current status of the environment. Grid technologies have enabled the aggregation of geographically distributed resources, in the context of a particular application. The network remains an important requirement for any grid application, as entities involved in a grid system (such as users, services, and data) need to communicate with each other over a network. The performance of the network must therefore be considered when carrying out tasks such as scheduling, migration or monitoring of jobs. Making use of the network in an efficient and fault tolerance manner, in the context of such existing research, leads to a significant number of research challenges. One way to address these problems is to make grid middleware incorporate the concept of autonomic systems. Such a change would involve the development of ``self-configuring'' systems that are able to make decisions autonomously, and adapt themselves as the system status changes.

Alexandre et al. (2007) proposed a semantic approach to integrate selection of equivalent resources and selection of equivalent software artifacts in order to improve the schedule of resources suitable for a given set of application execution requirements. A successful grid resource allocation depends, among other things, on the quality of the available information about software artifacts and grid resources. Scheduling parallel and distributed applications efficiently onto grid environments is a difficult task and a great variety of scheduling heuristics have been developed aiming to address this issue.

Romulo et al. (2007) focused on research methods to achieve efficient execution of parallel applications in a grid computing infrastructure. The paper presents WSPE, a grid programming environment for grid-unaware applications. WSPE's runtime system employs a new scheduling mechanism, called Round Stealing, inspired on the idea of work stealing. WSPE consists of a simple programming interface and a fully decentralized runtime system following a peer-to-peer organization. They have also demonstrated how an appropriate choice for a network overlay mechanism can further improve execution efficiency.

Singh and Srivastava (2007) devised a methodology to calculate Queue Length and Waiting Time utilizing Gateway Server information to reduce response time variance in presence of bursty traffic. The most widespread contemplation is performance, because gateway servers must offer cost-effective and high-availability services in the elongated period, thus they have to be scaled to meet the expected load. Performance measurements can be the base for performance modeling and prediction. With the help of performance models, the performance metrics (like buffer estimation, waiting time) can be determined at the development process.

Wang et al. (2008) presented a new service-oriented approach to the design and implementation of visualization systems in a grid computing environment. The approach evolves the traditional dataflow visualization system, based on processes communicating via shared memory or sockets, into an environment in which visualization Web services can be linked in a pipeline using the subscription and notification services available in Globus Toolkit 4. A specific aim of their design is to support collaborative visualization, allowing a geographically distributed research team to work collaboratively on visual analysis of data. A key feature of the system is the use of a formal description of the visualization pipeline, using the skML language first developed in the gViz e-Science project.

Conclusion

Computational grids are rapidly emerging as a practical means by which to perform new science and develop new applications. Grid computing appears to be a promising trend for three reasons: (1) its ability to make more cost-effective use of a given amount of computer resources, (2) as a way to solve problems that cannot be approached without an enormous amount of computing power, (3) it suggests that the resources of many computers which are not in use can be utilize for other computational task. The objective of this thesis is to present a homogeneous OS for the computational grid of Uttar Pradesh.

REFERENCES

Agustin C, Omer R, Maria C, Carmen C (2007). An Autonomic Network-Aware Scheduling Architecture for Grid Computing. pp. 444-458.

Alexandre V, Sergio K, Francisco da Silva e Silva, Fabio K (2007). Semantics-Based Grid Resource Management. 22(1):26-45.
Casanova H (1998). Distributed computing research issues in grid computing. pp. 506-515.
Cao J, Jarvis S.A, Saini S, Nudd G.R (2003). GridFlow: Workflow Management for Grid Computing. In Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'03) pp. 198-205.
Craig Lee, Domenico T(2003).Grid programming models: Current tools, issues and directions, In Grid Computing: Making The Global Infrastructure a Reality. John Wiley & Sons. pp. 555-578.
David De R, Mark A.B, Nicholas R.J, Nigel R.S (2003).The Evolution of the Grid. John Wiley & Sons. pp. 65-100.
Ewa D, James B, Yolanda G, Carl K, Gaurang M (2003).Mapping Abstract Complex Workflows onto Grid Environments. J. Grid Comput. 1:25–39.
Figueiredo R, Dinda P.A (2003).A Case For Grid Computing On Virtual Machines.In Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS), P. 550.
Fran B**,** Geoffrey F, Tony H (2002). The Grid: past, present, future John Wiley & Sons, Ltd.
Geoffrey F (2002). Education and the Enterprise with the Grid.
Hernandez F, Purushottam B, Rielly K (1999). End-user tools for grid computing. J. Grid Comput**.** pp. 156-165.
Ian F, Carl K, Steven T (2001). The Anatomy of the Grid: Enabling Scalable Virtual Organizations, Int. J. Supercomput. Appl.15:3
Krauter K, Maheswaran M (1997). Architecture for a Grid Operating System. pp.82-86
Klaus K, Rajkumar B, Muthucumaru M (2002). A taxonomy and survey of grid resource management systems for distributed computing. Softw. Pract. Exper. 32:135-164 (DOI: 10.1002/spe.432).
Parastatidis S, Watson P, Webber J (2005). Grid Computing using Web Services Technologies. published by Networkworld. pp. 377-389.
Romulo R, Patricia KV, Claudio G (2007). WSPE: A Peer-to-Peer Programming Environment for Grid-Unaware Applications.
Santos N, Koblitz B (2005). Metadata services on the grid. In Advanced Computing and Analysis Techniques (ACAT'05).
Singh LK, Srivastava R (2007). Estimation of Buffer Size of Internet Gateway Server via G/M/1 Queuing Model, proceedings of world academy of science, engineering and technology. 21:1307-6884.
Sulistio A,Uros C, Borut R, Rajkumar B(2005).Sulistio C, Robi¡c, Buyya (2006). A Toolkit for Modelling and Simulation of Data Grids with Integration of Data Storage, Replication and Analysis. http://www.cloudbus.org/reports/datagrid_fgcs.pdf
Talia D (2005). Towards GRID Operating Systems: from GLinux to a GVM, Workshop on Network Centric Operating Systems, Bruxelles.
Wang H, BrodlieK W, Handley JW, Wood JD (2008). Service-oriented approach to collaborative visualization. Concurr. Comput. 20(11):1289 - 1301.
Xiyan L, Runtong Z (2005). Grid: The Future of the Internet.
Zhang W, Wang Y (2000). Towards building a semantic grid for E-government applications. P. 31.