

Constructing a Grid Simulation for E-Governance Applications Using GridSim

¹Mrs. P. Sumathi and ²M. Punithavalli

¹Department of Computer Applications, PSG College of Arts and Science
Coimbatore - 641 014. Tamil Nadu, India

²Department of Computer Science, Sri Ramakrishna College of Arts and Science
for Women Coimbatore-641 006. Tamil Nadu, India

Abstract: This study provides a design framework for the adoption of grid computing for e-governance applications. **Problem statement:** E-Governance is the application of information and communication technology to achieve efficiency, effectiveness, transparency and accountability in Government to Government (G2G), Government to Employee (G2E), Government to Citizen (G2C) and Government to Business (G2B). It enables citizens to make best use of automated administration processes that are accessible on-line. Grid computing is an ideal solution to this type of applications and the study presents how grid computing can be used to effectively and efficiently handle such huge data. In this study, we illustrate the creation of a virtual environment by using existing Grid technologies to specific e-governance applications on distributed resources. **Approach:** A Grid generally refers to an infrastructure that involves the integrated and collaborative use of all computing resources into a single virtual computing environment. Grid applications often involve large amounts of data and/or computing resources that require secure resource sharing across organizational boundaries. Grid computing is an ideal solution to this type of applications and the study presents how grid computing can be used to effectively and efficiently handle such huge data. **Results:** The applications were run with the grid environment and without Grid environment. The results obtained were compared with the time and the number of jobs. The obtained results using grid environment were more significant and promising. **Conclusions:** Implementing an E-Governance solution will lower the cost of developing, deploying, managing government solutions and providing better services to citizens.

Key words: Grid computing, grid resource broker, scheduling

INTRODUCTION

Computational Grids^[1] enable sharing a wide variety of geographically distributed resources including supercomputers, storage systems, data sources and specialized devices owned by different organizations to create virtual enterprises and organizations. They allow selection and aggregation of distributed resources across multiple organizations for solving large-scale computational and data intensive problems in science, engineering and commerce. The parallel processing of applications on distributed systems provide scalable computing power. This enables exploration of large problems with huge data sets, which is essential for creating new insights into the problem. Citizen applying for a new water connection is one of the e-governance applications that require large computational and data storage capability.

We strongly feel that for the ultimate success of Computational Grids as a production-oriented commercial platform for solving problems, they need to support e-governance applications. In a Grid environment, a set of resources can dynamically team up to solve a given problem. This type of mutually agreed teaming up is quite useful for developing computational e-governance applications for executing parallel application tasks that have high degree of message communications for sharing partial results.

Related work: In the Data Grid Applications, it is valuable to point out the Storage Resource Broker^[3] from San Diego Supercomputing Centre which provides middleware for storing datasets over a network and accessing them. However, it does not deal with application execution directly. Cactus^[4] is a numerical problem solving environment for scientists

Corresponding Author: Mrs.P. Sumathi Department of Computer Applications PSG College of Arts and Science Coimbatore - 641 014. Tamil Nadu, India.

which supports Data Grid features through the use of MPICH-G and globus. However, applications in Cactus environment have to be written in MPI which implies that a legacy application cannot be adapted as such to be run on a grid.

The Application-Level Scheduling project builds agents for each application responsible for offering a scheduling mechanism^[2]. It uses a system-centric scheduling policy, which is targeted at minimizing the completion time and it does not take account of the economic cost of jobs processing while selecting applications.

Similar to^[2-4], our work focuses on a resource scheduling strategy within a Data Grid but we concentrate the same for heterogeneous resources that are implemented using the Gridsim simulator.

Requirements of E-governance based grid system

E-governance: E-governance is the use of information and communication technologies to support good governance.

The key characteristics of all e-governance projects are:

- The number of users of the system is enormous
- As time progresses, the number of applications will increase. Hence the system has to provide facilities for handling large loads.
- All e-governance applications must strictly adhere to specifications otherwise it is liable for legal prosecutions.
- The hardware and software heterogeneity exists in all spheres of e-governance. Hence there is a need for extensive integration.

Why E-governance?: According to Wikipedia, the important anticipated benefits of e-government include improved efficiency, convenience and better accessibility of public services^[5].

Some of the popular e-governance implementations in India include:

- Registration of property deeds in Andhra Pradesh,
- GYANDOOT: Community Owned Kiosks,
- State Border Check Posts, Gujarat

An Illustrative water connection example: The problem under consideration is how to obtain a water-connection from a municipal corporation or a municipality. It is assumed that both municipal corporation and municipality have computerized their operations and are web-enabled.

The stages in obtaining a water-connection are:

- The applicant has to submit an e-application for a new water-connection.
- The web server checks whether the Survey Field Number (SF number) of the land belonging to the applicant has been regularized by the concerned government authorities.
- Once the web server identifies that the SF number is a regularised one, the building plan is approved and the plan approval charges are calculated and levied.
- The server also verifies whether the building has obtained an electricity connection from the government electricity board. And all necessary charges for the same have been remitted.
- The server then calculates the house tax based on the plinth area and location of the house.
- The server then verifies whether the house tax and area development charges have been paid up-to-date.

Once all these are completed, the server then approves the water-service connection, allots a water-connection serial number and then calculates the appropriate water-connection charges.

The details of the Fig. 1 are given below:

- User submitting e-application for new water connection
- S.F number verification, plan approval verification & house tax calculation request to the server
- EB connection verification request to the server
- EB connection verification response by the server
- New water connection request to the Combine Water Supply Scheme (CWSS) Server
- New water connection approval number allotment.
- New water connection approval number allotment to the user

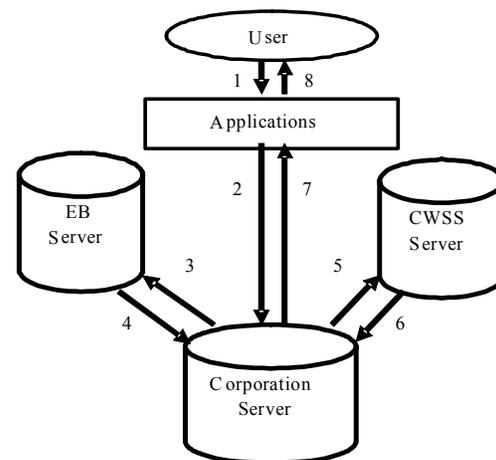


Fig. 1: Interactions that take place between various servers

Grid architecture for E-governance: The design features that are required by a Grid to provide users with a perfect computing environment are given below. Four main features describe a Grid are^[6]:

- Multiple Administrative Domains and Autonomy
- Heterogeneity
- Scalability
- Dynamicity or Adaptability

The steps necessary to realize a Grid include:

- The integration of individual software and hardware components into a combined networked resource.
The deployment of:
- Low-level middleware to provide a secure and transparent access to resources.
- User-level middleware and tools for application development and the aggregation of distributed resources.
- The development and optimization of distributed applications to take advantage of the available resources and infrastructure.

The Grid is made up of a number of components from enabling resources to end user applications. A layered Grid architecture of the Connection Approval Component is shown in Fig. 2. The key components of a Grid are^[6,7]:

- **Grid fabric:** This consists of all the globally distributed resources that are accessible from anywhere on the Internet. These resources could be computers running a variety of operating systems as well as resource management systems such as Load Sharing Facility, databases and special scientific instruments such as a radio telescope.
- **Core Grid Middleware:** This offers core services such as remote process management, co allocation of resources, storage access, information registration and discovery, security and aspects of Quality of Service (QoS) such as resource reservation and trading. managing resources and scheduling application tasks for execution on global resources.
- **User-Level Grid Middleware:** This includes application development environments, programming tools and resource brokers for
- **Grid Applications and Portals:** Grid applications are typically developed using Grid-enabled languages and utilities such as message-passing interface or Nimrod parameter specification language.

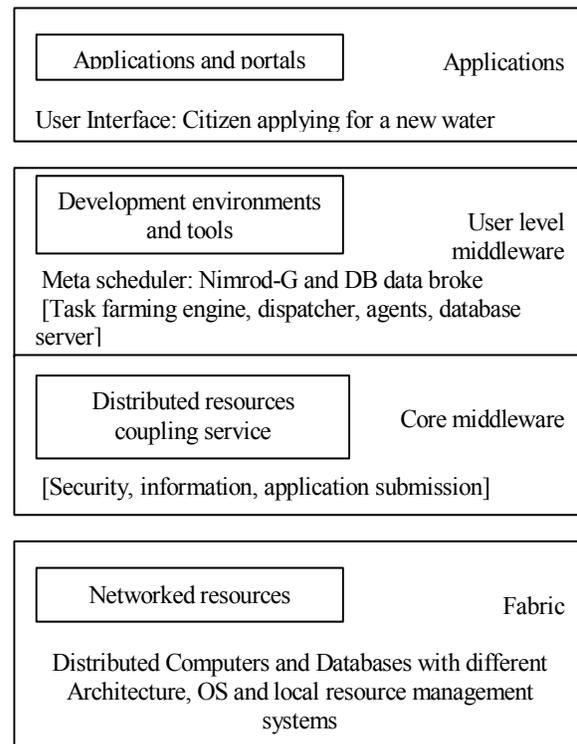


Fig. 2: A layered grid architecture and components

Grid resource broker is designed to operate in an environment that comprises a set of sites, each providing access to a set of Servers.

The Fig. 3 gives the interactions of Grid Resource Broker and the various components available in the e-governance applications.

Implementation of grid resource:

Broker simulator: The GridSim toolkit is used to simulate a Grid environment. The simulated Grid environment contains multiple resources and user entities with different requirements. The user and broker entities extend the GridSim class. All the users create application specification requirements and quality of services requirements. When the simulation starts, the user entity creates an instance of its own broker entity and passes a request for processing application jobs.

We briefly discuss features of the GridSim toolkit and its usage in the implementation of the economic broker simulator.

A Grid simulation toolkit: The GridSim toolkit presents a complete facility for simulation of different classes of various resources, users, applications, resource brokers and schedulers^[7,8,9]. It has facilities for

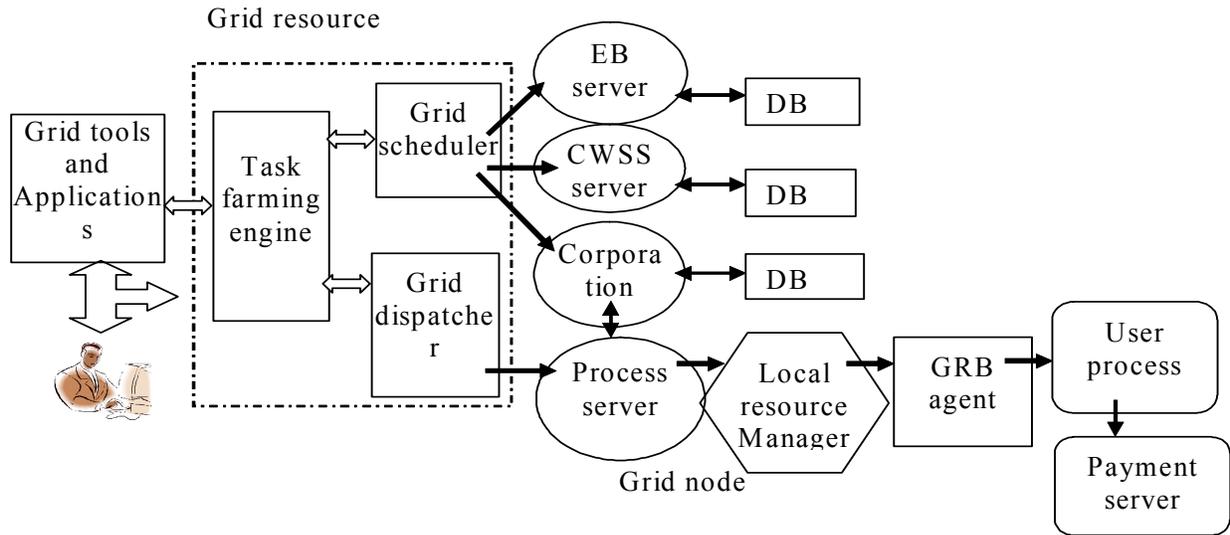


Fig. 3: Grid resource broker and the various components interactions

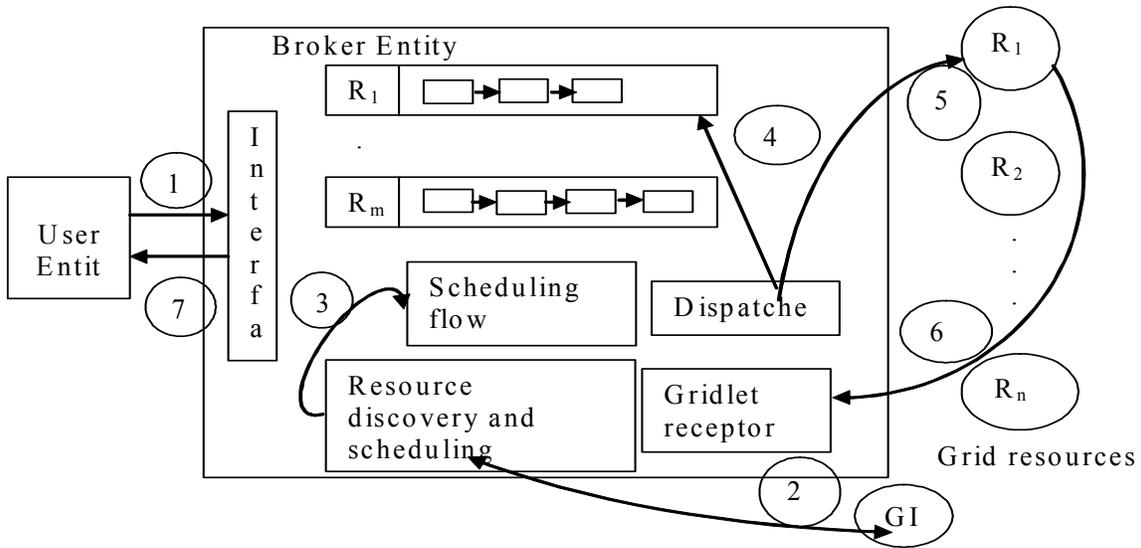


Fig. 4: Grid resource broker architecture and its interactions with other entities

the modeling and simulation of resources and network connectivity with different capabilities, configurations and domains. It supports primitives for application composition, information services for resource discovery and interfaces for assigning application tasks to resources and managing their execution. The GridSim toolkit resource modeling facilities are used to simulate the world-wide grid resources. The broker can be used for communication with other entities. In GridSim, application jobs are modeled as gridlet objects that contain all the information related to the job and its

execution management details such as job length in MI, disk input/output operations, input and output file sizes and the job originator. The broker uses GridSim's job management protocols and services to map a gridlet to a resource and manage it throughout its lifecycle. The broker also maintains full details of application scheduling trace data both at coarse and fine levels, which can be used in performance analysis.

Grid broker simulator architecture: A detailed discussion of the broker implementation using the

GridSim toolkit can be found in Buyya and Murshed^[7].The broker entity architecture and its interaction with other entities is shown in Fig. 4. The key components of the broker are: Interface, resource discovery and scheduling, scheduling flow manager, dispatcher and gridlets receptor:

- The user entity contains the application description and sends user requirements to the broker through the interface.
- The broker resource discovery and scheduling module interacts with the GridSim GIS entity to identify the contact information of the resources and then interacts with resources. It creates a broker resource list that acts as a placeholder for maintaining resource properties, a list of gridlets committed for execution on the resource and the resource performance data as predicted through the measurement and extrapolation methodology.
- The scheduling flow manager is used for mapping gridlets to resources depending on the user’s requirements.
- The dispatcher selects the number of gridlets for each of the resources that can be staged for execution according to the usage guidelines to avoid overloading resources with single user jobs.
- The dispatcher then submits gridlets to resources using the GridSim’s asynchronous service.
- When the gridlet processing completes, the resource returns it to the broker’s gridlet receptor
- Module, which then measures and updates the runtime parameter and it serves in predicting the job utilization rate for making scheduling decisions.
- Steps 3-6 continue until all the gridlets are processed. At the end, the broker returns updated experiment data along with processed gridlets back to the user entity.

RESULTS AND DISCUSSION

The experiments were carried out with two settings. The applications were run with the grid environment and without Grid environment.

A graph shows the comparison of resource performance for all the jobs is shown in Fig. 5 and another is the comparison of the total time taken for each server to execute all the jobs is shown in Fig. 6.

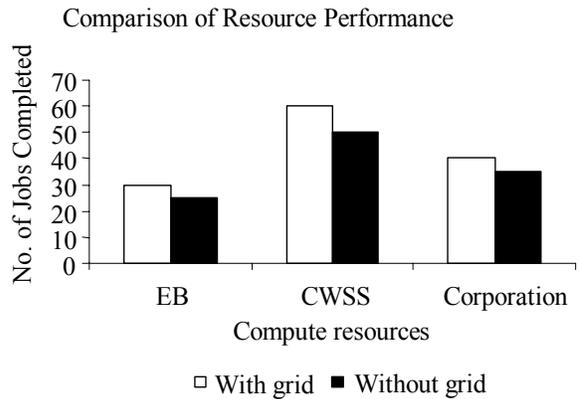


Fig. 5: Comparison of resource performance

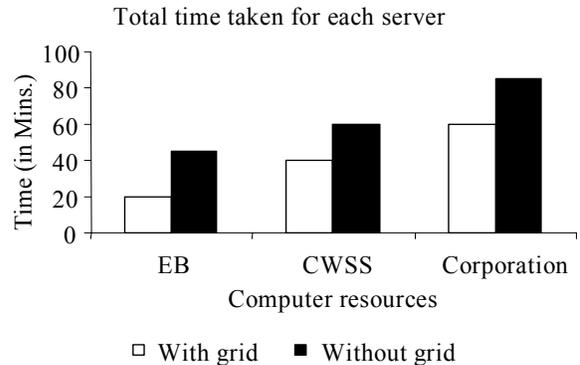


Fig. 6: Total Time taken for each server

In the first strategy, jobs were executed in the EB, CWSS and Corporation servers. As is displayed in the graph in Fig 5, all of the servers were executed different jobs. For example the EB server was executed 30 jobs within a given time in the grid environment and only 25 jobs without the grid environment. In same way all the servers were executed more jobs in the grid environment.

In the second strategy, jobs were executed in all the servers. For Example the CWSS server was executed the given jobs in 40 minutes in the grid environment and took 60 Minutes in the other case.

As can be seen from Fig. 5 and 6, in the grid environment the application took the least time and completed more jobs than without grid environment. This was implemented with the help of the GridSim Toolkit.

In this study, we have described how emerging grid computing technologies can be used to implement an e-Governance framework. We have presented a grid broker for executing distributed data-oriented jobs on a grid. The broker discovers computational and data

resources, schedules jobs based on optimization of data transfer and returns results back to the user.

CONCLUSION

Implementing an e-Governance solution will lower the cost of developing, deploying, managing government solutions and providing better services to citizens. The study has identified typical areas for e-governance. As a case study, we have presented a scenario of a citizen applying for a new water connection. We have implemented the application with the usage of grid and without the grid environment.

We are extending the current system into applying with the scheduling algorithm to develop the environment to increase speed and minimize cost.

ACKNOWLEDGMENTS

The authors wish to thank our management, secretary and principal for their constant encouragement and active support.

REFERENCES

1. Abramson, D., R. Sasic, J. Giddy and B. Hall, 1995. Nimrod: A tool for performing parametrised simulations using distributed workstations. Proceedings of the 4th IEEE International Symposium on High Performance Distributed Computing, Aug. 2-4, IEEE Xplore, Washington, DC, USA., pp: 112-121. DOI: 10.1109/HPDC.1995.518701
2. Berman, F., R. Wolski, 1997. The Apples project: A status report. Proceedings of the 8th NEC Research Symposium, Germany, May 1997. Elsevier Press, Amsterdam, The Netherlands. <http://apples.ucsd.edu/pubs/nec97.ps>
3. Baru, C., R. Moore, A. Rajasekar and M. Wan, 1998. The SDSC storage resource broker. Proceedings of the Conference of the Centre for Advanced Studies on Collaborative Research, Nov. 30-Dec. 3, IBM Press, Toronto, Ontario, Canada, pp: 5. <http://portal.acm.org/citation.cfm?id=783160.783165>.
4. Allen, G., *et al.*, 2000. The cactus code: A problem solving environment for the grid. Proceedings of the 9th International Symposium on High Performance Distributed Computing, Aug. 01-04, IEEE Press, Pittsburgh, PA, USA., pp: 253-260. DOI: 10.1109/HPDC.2000.868657.
5. Sameer Sachdeva, 2002. E-governance strategy in India, December 2002. <http://unpan1.un.org/intradoc/groups/public/documents/apcity/unpan014672.pdf>.
6. Rajkumar Buyya, David Abramson and Srikumar Venugopal, 2005. The Grid Economy, Special Issue on Grid Computing, Proceedings of the IEEE, Manish Parashar and Craig Lee (Eds.). IEEE Press, New York, USA., March 2005. pp: 698-714.
7. Rajkumar Buyya and M. Murshed 2002. Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Concurrance. Computat. Pract. Exp.*, 14: 1175-1220. DOI: 10.1002/cpe.710
8. Luis Ferreira, *et al.*, Grid Computing in Research and Education, IBM Red Books, April 2005. ISBN: 0738491756. <http://www.redbooks.ibm.com/redbooks/pdfs/sg246649.pdf>
9. The Globus Toolkit: <http://www.globus.org>.