

Grid Computing: A Desirable Tool for Electronic Governance

Iyanda, D. G.[1*], Ogundele, L. A.[2], Oyedemi, O. A.[2]

1. Department of International Co-operation, State Planning Commission of Osun, New Secretariat, P. M. B. 4435, Osogbo, Osun State, Nigeria.
2. Department of Computer Science, Osun State College of Education, Ilesa, Osun State, Nigeria.

[*]E-mail of corresponding author: dgiyanda@fastmail.fm

Abstract

This paper explained how Government at different levels can apply Information and Communication Technology (ICT) 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). This application is referred to as Electronic Governance (e-Governance). The system 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 administrative processes. This paper therefore presents how Grid computing can be used to effectively and efficiently meet the yearnings of citizenry. Generally, we demonstrated the creation of a virtual environment by using Grid technologies to a specific e-governance application on distributed resources. We presented a framework for the adoption of grid computing for e-governance management using Electronic Bill server (EB server), Comprehensive Welfare and Social Services server (CWSS server) and Corporation sever (C server). Experiments were run with the Grid environment and without Grid environment by considering the number of jobs completed and the period to complete various jobs submitted for processing using MATLAB.

The numbers of jobs completed by EB server by using Grid are: 20, 40, 60, 80,100 and 120 while 15, 25, 33, 60, 72 and 90 were completed without Grid under the same condition. The numbers of jobs completed by CWSS server with Grid are: 30,50,70,90,120 and 130 while 22.5, 37.5, 52.5, 67.5 90 and 97.5 were completed without Grid. The numbers of jobs completed by Corporation server under Grid are: 30,50,70,90,120 and 130 while 24, 40,56,72,96 and 104 were completed without Grid. The period to complete various jobs submitted for processing by the EB server under Grid are: 18, 30, 42, 54,72 and 88minutes while 30,50,70,90,120 and 130minutes were required without Grid. For CWSS server, the period to complete various jobs submitted for processing under Grid are: 6.5, 19.5, 32.5, 45.5, 58.5,78 and 84.5minutes while 10, 30,50,70,90,120 and 130minutes were required without Grid. For Corporation server, the period to complete various jobs submitted for processing under Grid are: 6.4,19.2,32,44.8,57.6, 76.8 and 82.2minutes while 10,30,50,70,90 120, and 130minutes were required without Grid.

The result of simulation revealed that implementing an e-Governance solution was cost effective, efficient, consistent and reduced job processing time with high quality of result and providing better services to citizens.

Keywords: E-Governance, Grid Applications, Grid Computing, Grid environment, Grid Infrastructure and Grid Resource Broker.

1. Introduction

Grid computing 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. With particular reference to Rajkumar et al (2005), Rajkumar and Murshed (2002), the key components of a Grid are: Grid frame work, Core Grid Middleware, User-Level Grid Middleware, and Grid Applications and Portals. The Grid frame work 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. The Core Grid Middleware is a platform that 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. The User-Level Grid Middleware includes application development environments, programming tools and resource brokers for users. The Grid Applications and

Portals are typically developed using Grid-enabled languages and utilities such as message-passing interface or parameter specification language.

Following from Abramson et al (1995), Computational Grids 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 free health care scheme is one of the e-governance applications that require large computational and data storage capability.

The ultimate success of Computational Grids as a production-oriented commercial platform for solving problems can be utilized to support e-governance applications, Sumathi and Punithavalli (2008). 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. Detailed discussion of the broker implementation using the GridSim toolkit is available in Rajkumar and Murshed (2002). The key components of the broker are: Interface, resource discovery and scheduling, scheduling flow manager, dispatcher and gridlets receptor.

2. Previous Related Works

Numerical problem solving environment for scientists which supports Data Grid features through the use of MPICH-G and globus was presented in Allen (2000). Application-Level Scheduling project builds agents for each application responsible for offering a scheduling mechanism was proposed in Berman and Wolski (1997). This mechanism 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 into consideration while selecting applications. In the Data Grid Applications, the Storage Resource Broker from San Diego Supercomputing Centre provides middleware for storing datasets and accessing them, Baru et al (1998). It was observed that this framework does not deal with application execution directly. Similar to Berman and Wolski (1997), Baru et al (1998) and Allen (2000), this work focuses on a resource scheduling strategy within a Data Grid. Armstrong Ford (2005) used performance control of scientific coupled models in Grid environments for resource management. Eludiora et al. (2010) applied Load Balancing Policy to address problems associated with Distributed Web Service. However, in this paper we concentrate on utilization of heterogeneous resources in virtual environment for simulation purpose. This is the major different from other related works and this paper.

3. E-governance Architecture

As indicated in Rajkumar et al (2005), the design features that are required by a Grid to provide users with a perfect computing environment has four main features. These are: Multiple Administrative, Domains and Autonomy, Heterogeneity, Scalability and Dynamicity or Adaptability. However, the steps necessary to realize a Grid include the integration of individual software and hardware components into a combined resource. It also involved the deployment of Low-level middleware to provide a secure and transparent access to resources. Other features include; User-level middleware and tools for application development and the aggregation of distributed resources and the development and optimization of distributed applications to take advantage of the available resources and infrastructure. Figure 1 is the layered Grid architecture with a number of components.

3.1. E-Governance under Grid System

Because e-governance is the use of information and communication technologies to support good governance, the key characteristics of all e-governance projects are that the number of users of the system is enormous. As time progresses, the number of applications also increases in direct proportion. 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. The important anticipated benefits of e-government include improved efficiency, convenience and better accessibility of public services (Sameer, 2002).

4. The Proposed Framework

The proposed framework is based on the principle of utilization of heterogeneous resources virtual environment. Virtual environment or Virtual Organization (VO) is a concept that had emerged since 1990s yet, there is no single clear-cut definition of what a VO is. Different sources define it somewhat differently. According to Brian (2002), VO is a dynamic collection of individuals and institutions which are required to share resources to achieve certain goals. The proposed framework, therefore, is to test how to obtain free health care scheme from municipals health centres scattered in a virtual environment. It is assumed that those municipalities has computerized their health care system operations and are web-enabled. The architectural design of a Virtual Environment is as shown in Figure 2 while the process cycle of obtaining a free health care scheme is as depict below in Figure 3. As illustrated in Figure 3, prospective applicants need to submit e-applications for a new free health care scheme. The Corporation server checks whether the Survey Field Number (SF number) of the applicant has been regularized by the concerned authorities. Once the Corporation server identifies that the SF number is a regularized one, the applicant is allowed to have access to the service and passed the information to the Electronic Bill server (EB server). The EB server then verifies whether the beneficiary has paid for utility services provided by government and that all necessary charges for the same have been remitted. The server then calculates the beneficiary's tax and returns the information back to Corporation server. This server finally verifies whether the tax and other related charges have been paid up-to-date.

Once all the above processes are completed, the Corporation server sends the information to the Comprehensive Welfare and Social Services server (CWSS server). This server then approves the free health care system, allots a free health care scheme serial number and then calculates charges to be paid by the government on that particular beneficiary. This is also useful to track the total amount of the annual budget expended on such scheme and to project for future; hence promoting ease governance, transparency and accountability. The Information Management representation of e-governance is as shown in Figure 4.

4.1 Interaction between components of e-governance and grid resource broker

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 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. The GridSim toolkit presents a complete facility for simulation of different classes of various resources, users, applications, resource brokers and schedulers Rajkumar and Murshed (2002), Luis (2005). It has facilities for the modeling and simulation of resources and connectivity with different capabilities, configurations and domains. It supports 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 usually 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, 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. Figures 5 and 6 describe Interactions of Grid Resource Broker and the various components of e-governance applications as well as Grid Resource Broker and its interactions with other entities, respectively.

5. Result of the Experiment

The experiments were carried out with two settings using MATLAB. The applications were run with the assumption of being in Grid environment and without being in Grid environment. The comparison of number of jobs completed by each server under the above conditions are shown in Figures 7 , 8 and 9 while the comparison of the total time taken for each server to execute all the jobs are shown in Figures 10 , 11 and 12.

6. Discussion on Result of the Experiment

In the first scenario, as shown in Figures 7, 8 and 9, all the servers executed different numbers of jobs with and without Grid. For example, the number of jobs completed by EB server by using Grid: are 20, 40, 60, 80,100 and 120 while 15, 25, 33, 60, 72 and 90 were completed without Grid under the same condition. In the same manner, CWSS and Corporation servers executed more jobs in the grid environment. Under the time used to complete various jobs as shown in Figures 10, 11 and 12, all the servers executed jobs at different time lag. For example, the period to complete various jobs submitted for processing by the EB serve under Grid are: 18, 30, 42, 54, 72 and 88minutes while 30,50,70,90,120 and 130minutes were required without Grid. It is apparent from Figures 7, 8, 9, 10,11 and 12, that in the grid environment the application took the least time and completed more jobs than without grid environment. Through the above results presented, we have described how emerging grid computing technologies can be used to implement an e-Governance framework.

7. Conclusion

This paper has identified and discussed one of the social service areas of e-governance. As a case study, we have presented a scenario of a citizen applying for a new free health care scheme. We have simulated the application within and outside the grid environment. The result shows that rendering service in Grid environment is cost effective with reduced waiting time of the clients which is opposite in non-Grid environment. This is a pointer to the fact that when the framework is implemented, it will lead to e-Governance solution with lower cost of developing, deploying, managing government businesses and as well providing better services to citizens. Work is on-going in extending the current framework to implementation level with increased speed in job execution at minimum cost.

8. Recommendation

It is has been observed that in a Grid environment, a set of resources can dynamically team up to solve a given problem. This type of mutual agreement is quite useful for developing computational e-governance applications for the execution of parallel application tasks that have high degree of message communications. Since all the globally distributed resources are accessible from anywhere on the Internet, these resources could be utilized to solve resource management system problems as well as special scientific instruments. Due to this dynamic nature of Grid system, there is a need to design and develop an easy to use and secured Grid management infrastructure that will adequately manage the constituent users and resources.

References

Abramson, D., R. Sosic, J. Giddy and B. Hall (1995); Nimrod: A tool for performing parametised simulations using distributed paperstations. Proceedings of the 4th IEEE International Symposium on High Performance Distributed Computing, Washington, DC,USA.,pp:112-121.

Allen, G. (2000); The cactus code: A problem solving environment for the grid. Proceedings of the 9th International Symposium on High Performance Distributed Computing, IEEE Press, Pittsburgh, PA, USA., pp: 253-260.

Armstrong, C. W., and Ford, R. W. (2005). Performance control of scientific coupled models in Grid environments; Concurrency and Computation: Practice and Experience 17(4): 259-295.

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. IBM Press, Toronto, Ontario, Canada.

Berman, F., and R. Wolski (1997); The Apples project: A status report. Proceedings of the 8th NEC Research Symposium, Germany. Elsevier Press, Amsterdam.

Brian Matthews; Juan Bicarregui and Theo Dimitrakos (2002):"Building Trust on the GRID, Trust Isuues Underpinning Scalable Virtual Organizations", CLRC Rutherford Appleton Laboratory, Oxfordshire, OX11 0QX, UK.

Eludiora, S., Abiona, O., Aderounmu, G.A., Oluwatope, A.O., Onime,C., and Kehinde, L., (2010): A Load Balancing Policy for Distributed Web Service. International Journal of Communications, Network and System Sciences 2010, 3: 645-654

Luis F. (2005); Grid Computing in Research and Education, IBM Red Books.

ISBN: 0738491756.

Rajkumar B., D. Abramson and S. 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, pp: 698-714.

Rajkumar B. and M. Murshed (2002); Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. Concurrence. Computation Practical Experience, 14: 1175-1220.

Sameer, S. (2002); E-governance strategy in India, December 2002.

Sumathi, P. and Punithavalli, M. (2008): Constructing a Grid Simulation for E-Governance Applications Using GridSim. Journal of Computer Science 4 (8): 674-679, 2008. ISSN 1549-3636.

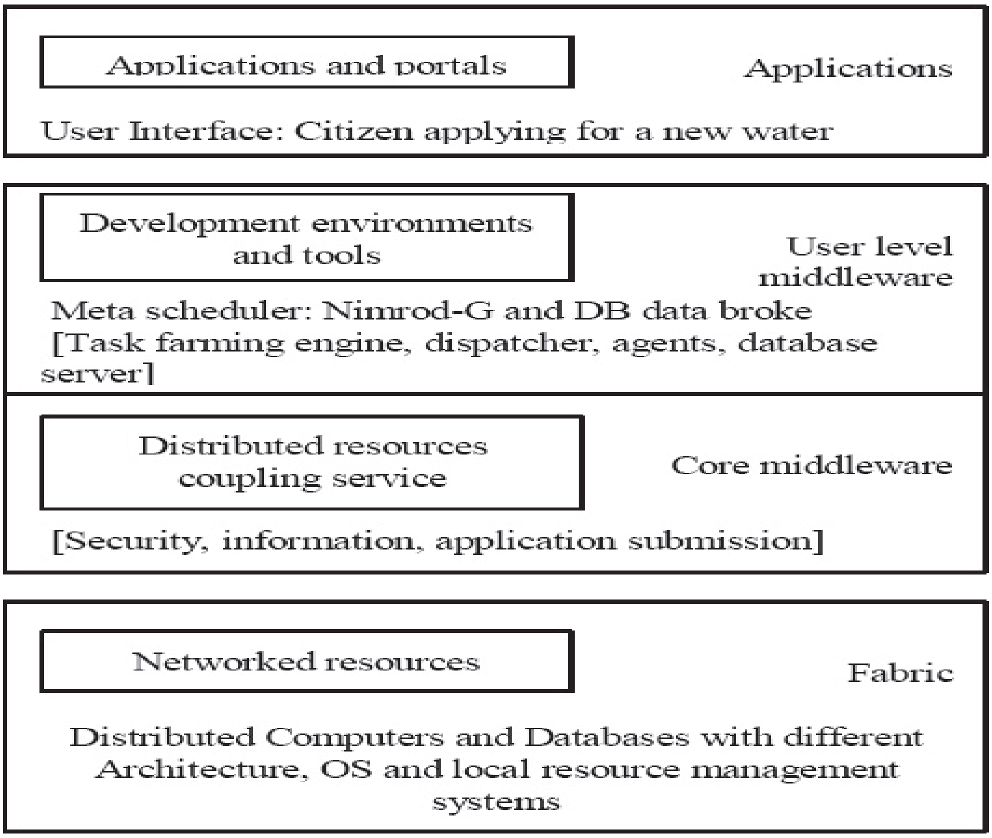


Figure 1: Layered Grid architecture

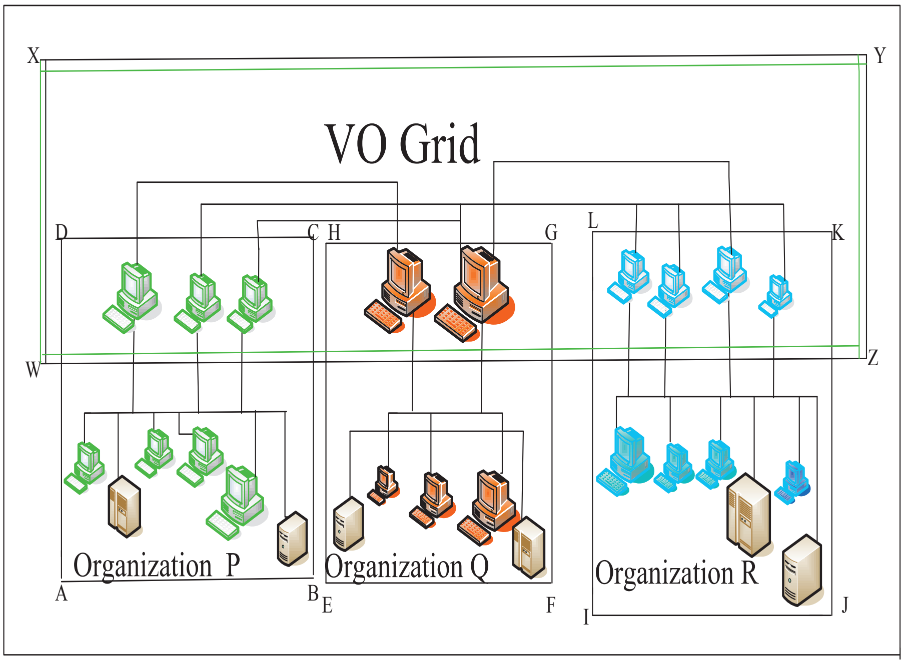


Figure 2: Architectural Design of Virtual Environment

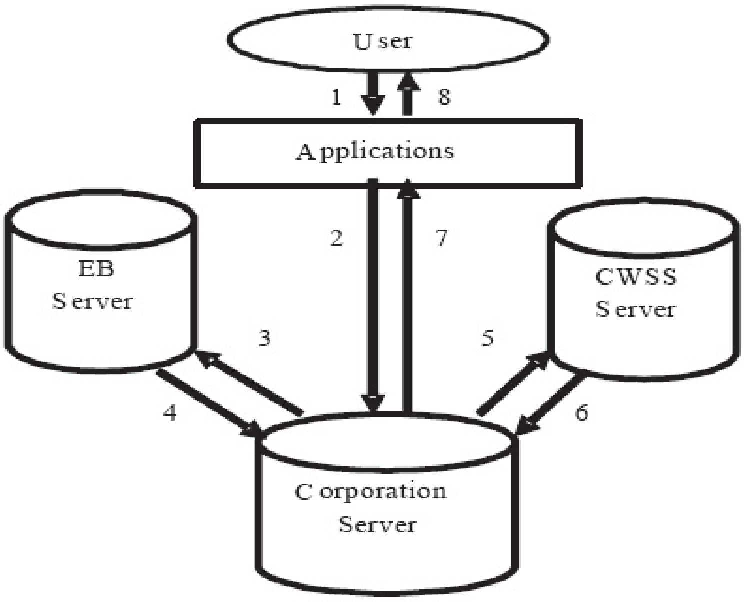


Figure 3: Architectural Design of a specific e-governance Grid system

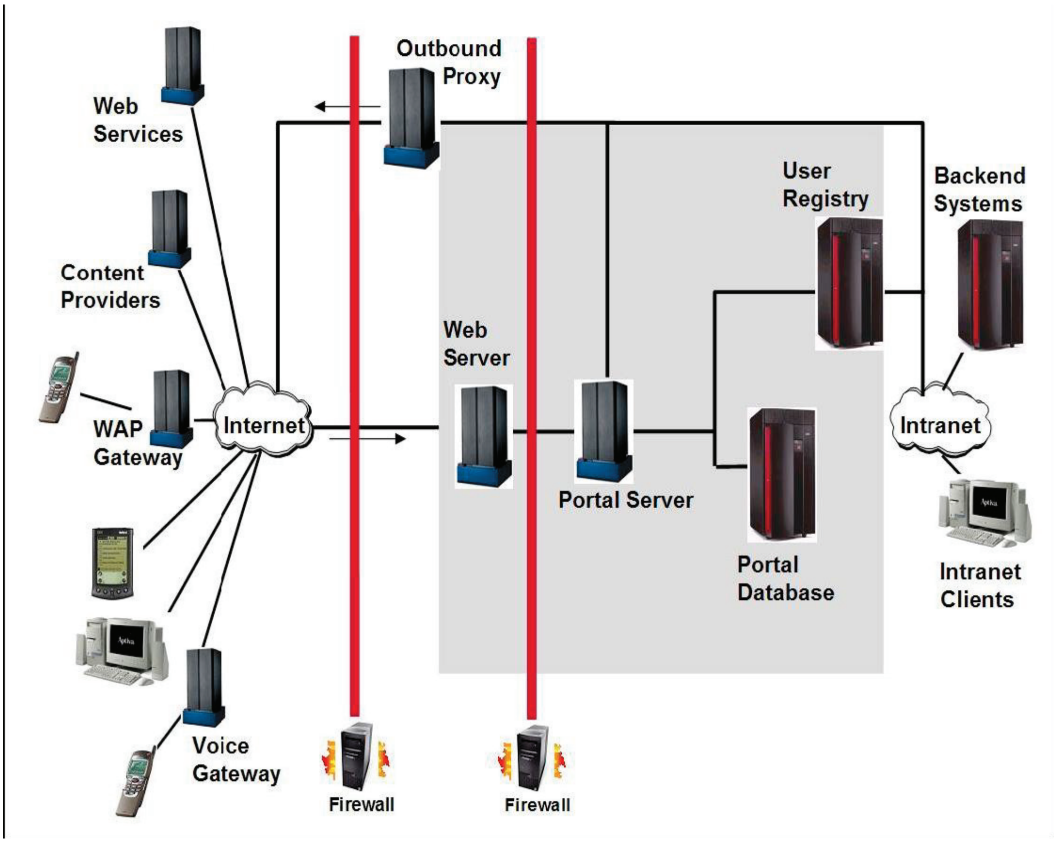


Figure 4: Information Management Model of e-governance

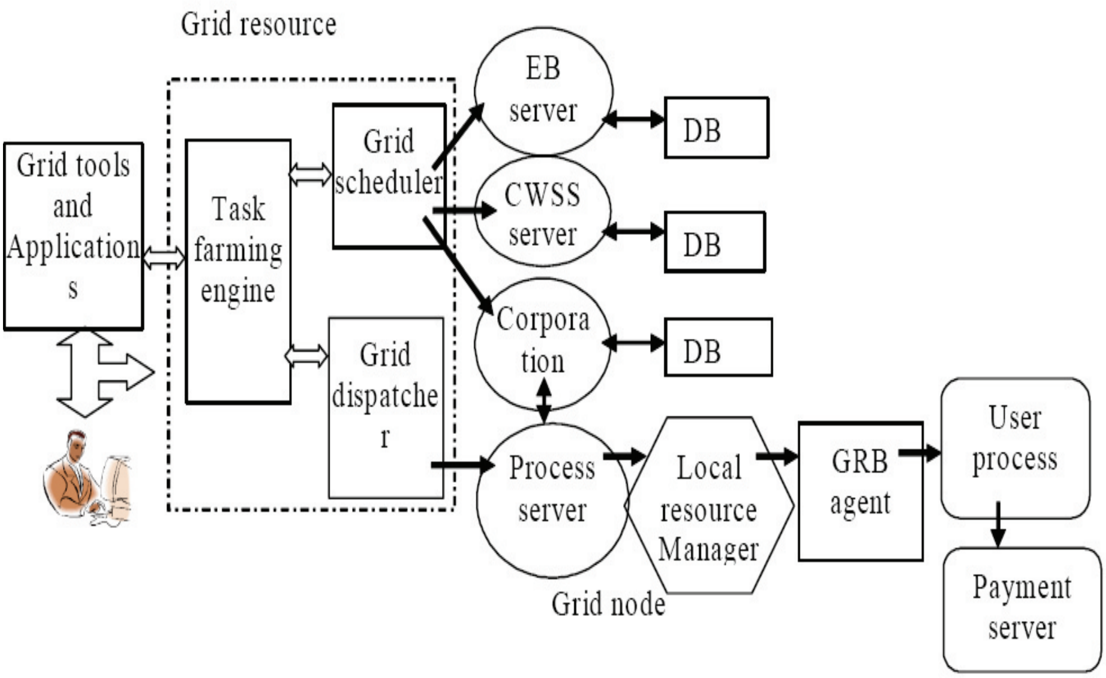


Figure 5: Interactions of Grid Resource Broker and the various components of e-governance applications.

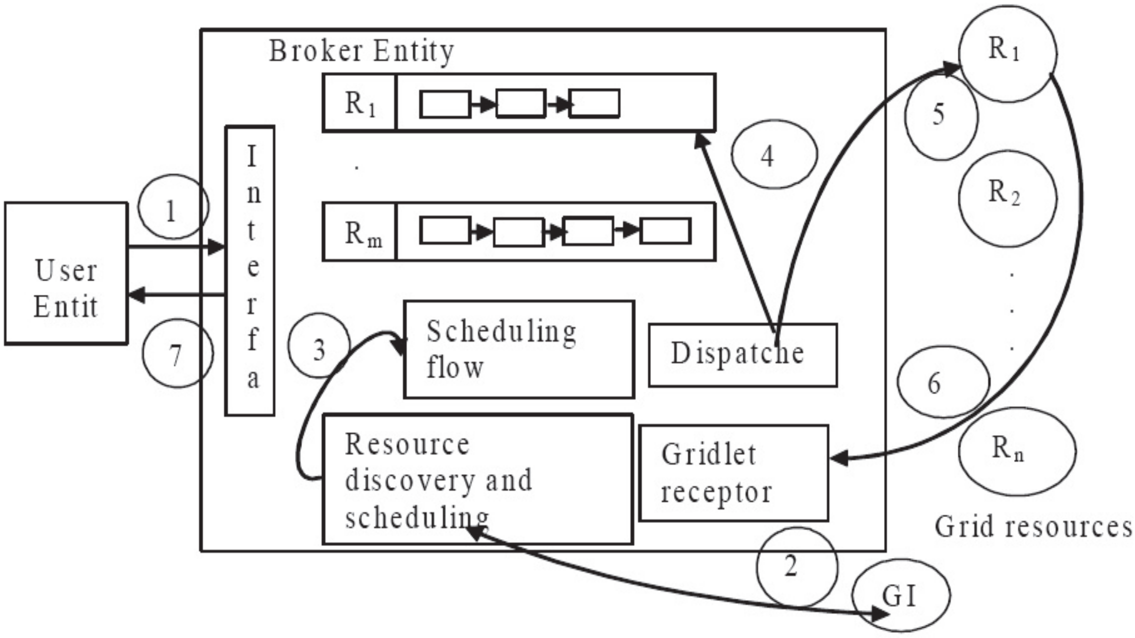
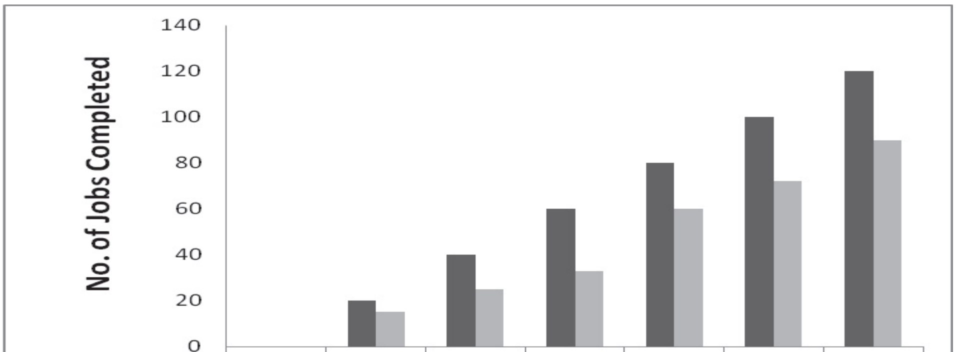


Figure 6: Grid resource broker architecture and its interactions with other entities



	1	2	3	4	5	6	7
With Grid	0	20	40	60	80	100	120
Without Grid	0	15	25	33	60	72	90

Figure 7: Comparison of number of jobs completed by Electronic Bill server (EB server)

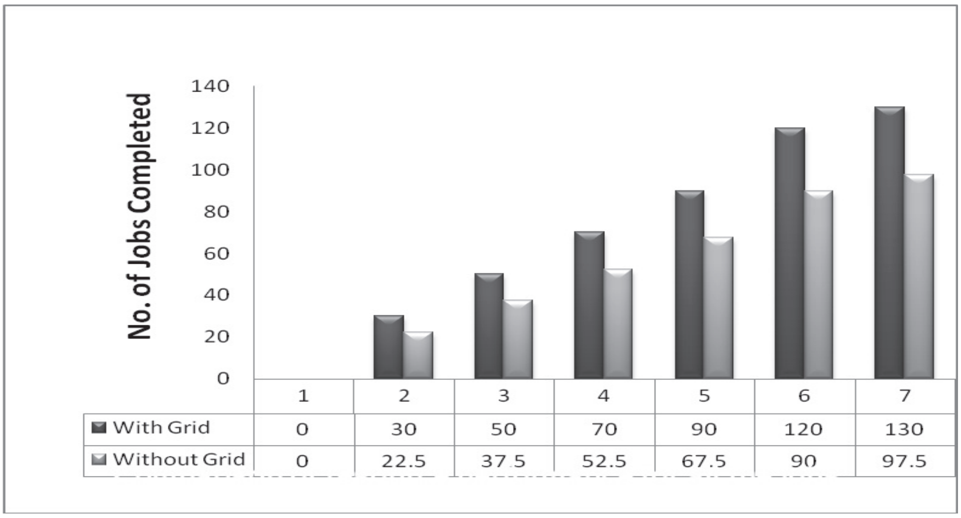


Figure 8: Comparison of number of jobs completed by Comprehensive Welfare and Social Services (CWSS) server

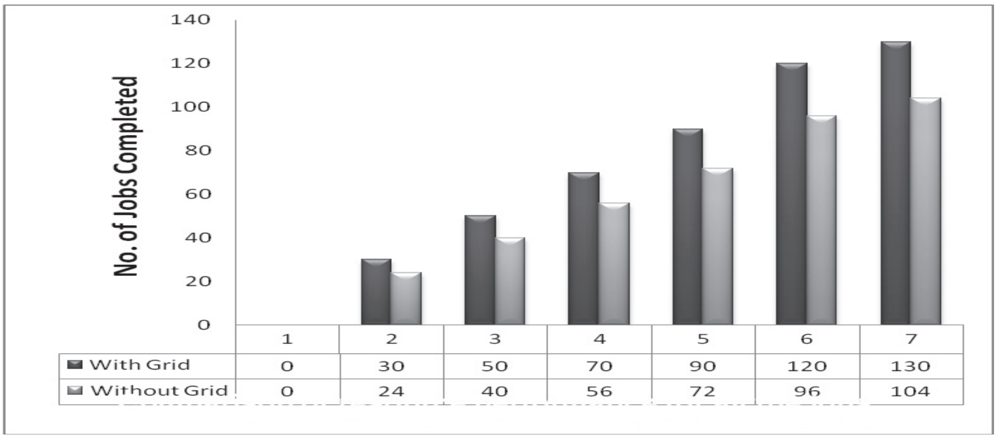


Figure 9: Comparison of number of jobs completed by Corporation server

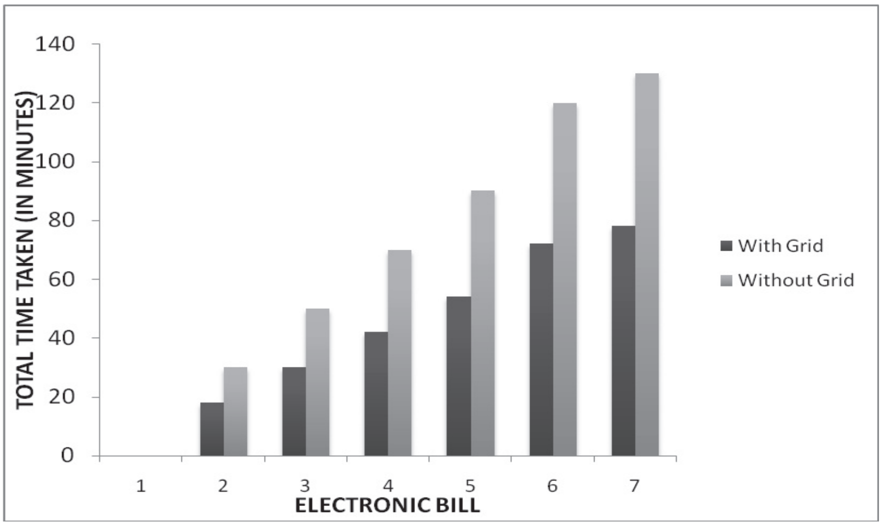


Figure 10: Total Time taken for Electronic Bill Server

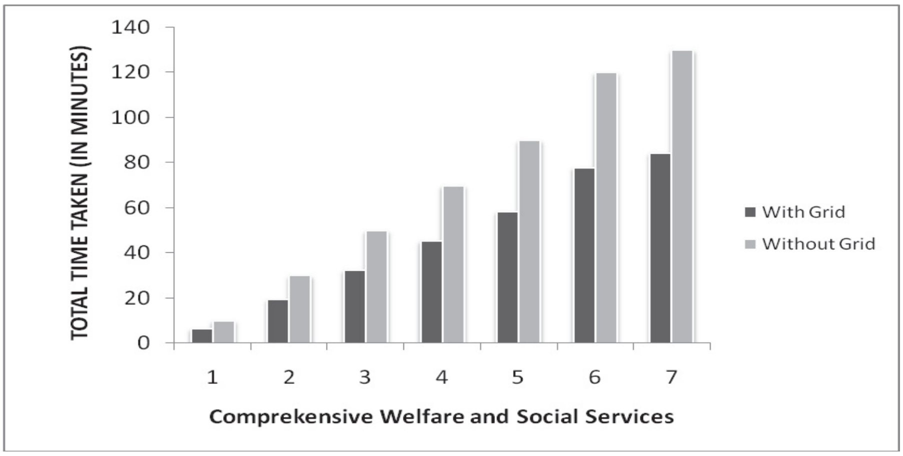


Figure 11: Total Time taken for CWSS Server

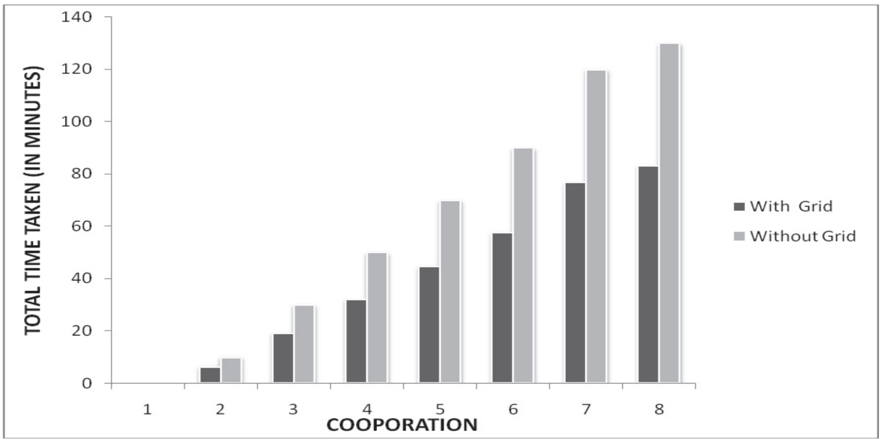


Figure 12: Total Time taken for Corporation