Building Grid based Application for the Management of Medical Image Data using Alchemi

M.Dhivya
PG Student
Dept of CSE
P.S.R Engineering College
Sivakasi

K.Ruba Soundar
Professor and Head
Dept of CSE
P.S.R Engineering College
Sivakasi

ABSTRACT

Grid computing is the combination of computer resources from multiple administrative domains applied to a common task, usually to a scientific, technical, business or Medical problem that requires a larger number of computer processing cycles. Many Grid Middleware distributions, such as Globus, NIMROD, Alchemi etc, are used to build the grids. This paper focuses on the grid application of the management and processing of digital medical images, in particular, functional magnetic resonance imaging (fMRI) by using Alchemi tool. Functional MRI is a non-invasive technique used to investigate the functions of the human brain. The computational demand for processing fMRI datasets is usually high. To make the comparison process of brain image data in single system requires more number of computer processing cycles. To cope with this computational demand, has to be developed Alchemi Grid. It provides a high level secured and reliable service environment significantly reduces the processing time and also reduces the CPU cycle utilization.

Keywords

Alchemi, Middleware, MRI, fMRI, NIMROD, Globus

1. INTRODUCTION

Recently many scientific experiments such as structural biology and chemistry, neuroscience, data analysis, disaster recovery, etc., are conducted through complex and distributed scientific computations that are represented and structured as scientific workflows [3]. Scientific workflows usually need to process huge amount of data and computationally intensive activities. Workflow management is emerging as one of the most important grid service[13]. Neuroscience data analysis is one such application that has been focus of much research in recent years (NIFTI, BIRN). Recent advances in Medical imaging technology, present several challenges to the management of digital medical images in terms of both storage and computational requirements. Medical imaging technologies such as magnetic resonance imaging (MRI), computed tomography (CT), among others, have progressed considerably over the past several years.

In recent years, publicly funded medical grid projects are developed. For example, MediGRID [2], EGRID show interest in the developing of grid technologies today. High resolution imaging techniques and advanced algorithms produce a rising demand on computing power and storage capacity in Medical image processing.

Functional magnetic resonance imaging is an imaging technique that can be used to characterize brain physiological activity usually presented as 3D volumes in function of time, So time series analysis imposes challenging requirements regarding computational power and medical image management. gLite middleware[14] are implemented for managing the brain images.

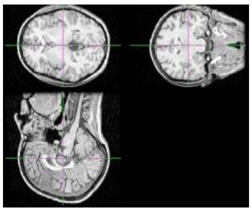
The fMRI data analysis application present in this paper has several tasks for correct execution. Both the data and computational requirements are very high. Given the typically large number of subjects' data being analyzed, it takes significant amount of time for this application to produce results when executed as a sequential process on limited resources. Moreover, scientists may need to re-run the application by varying run-time parameters. Often researchers and users around the globe may share the results produced. To facilitate these requirements the application may leverage the power of distributed resources presented by platforms such as Grids. By executing this application an Alchemi grid tool execution time can be minimized, repeated executions can be performed with little overhead, reliability of execution can be increased, and resource usage can be distributed, It also provides a high level secured and reliable environment.

2. RELATED WORK

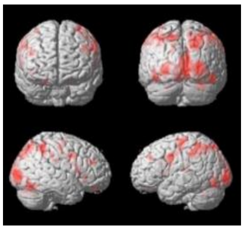
Functional MRI is a non-invasive technique used to investigate the functions of the human brain. The computational demand for processing fMRI datasets is usually high. To make the comparison process in single system requires high computation time. FMRI attempts to determine which parts of the brain are active in response to some given stimulus. For instance, a person (referred as subject), in the Magnetic Resonance (MR) scanner, would be asked to perform a task, e.g., finger-tap at regular intervals. As the subject performs the task, researchers effectively take 3-D MR images of his brain. The goal is to identify those parts of the brain responsible for processing the information the stimulus provides. Standard image is compared with the atlas image for identifying the tissues. Before making the comparison process we need image registration. When registering images we are determining a geometric transformation, which aligns one image to fit another. The aim is to establish a one-to-one continuous correspondence between the brain images of different individuals. The transformation will reduce the anatomical variability between high resolution anatomical brain images from different subjects. After registration we need to clearly identify the tissues so we required normalization process. Here using canny edge detection algorithm for normalization process.

This enables analysts to compute a single activation map representing the entire group of subjects or to compare the brain activation between two different groups of subjects. Figure shows an fMRI image processing. This Collection of fMRI images has to be stored into the database. Consider The raw fMRI data for a typical study would consist of three subjects with 20 subjects per group, five experimental runs per subject, and 300 volume images per run, yielding 90,000 volumes and over 60GB of data. A complete execution of the workflow of 40 subjects on a single CPU with single core takes two and a half days to complete. So we enter into grid environment for minimizing the processing time. Here we implemented Alchemi tool for fMRI processing.

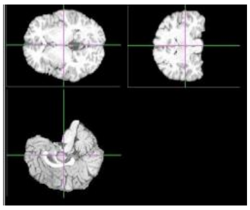
(a) fMRI scanner



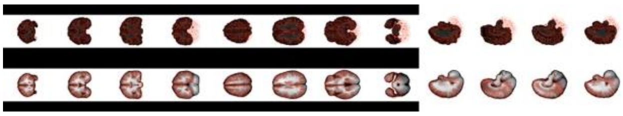
b) Before Image Registration



e) Further fMRI analysis



c) After image registration



(d) Comparison of Image fit with the Standard Image

Fig 1. fMRI Image Processing

Alchemi is one of the Software to enable grid computing has been primarily written for UNIX operating systems. Microsofts .Net framework provides a platform to implement windows based grid computing environment. It is shown in the following diagram. In particular, it also provides remote execution, security, multithreading,, asynchronous programming, disconnected data access, managed execution, making it an ideal platform for grid computing middleware.

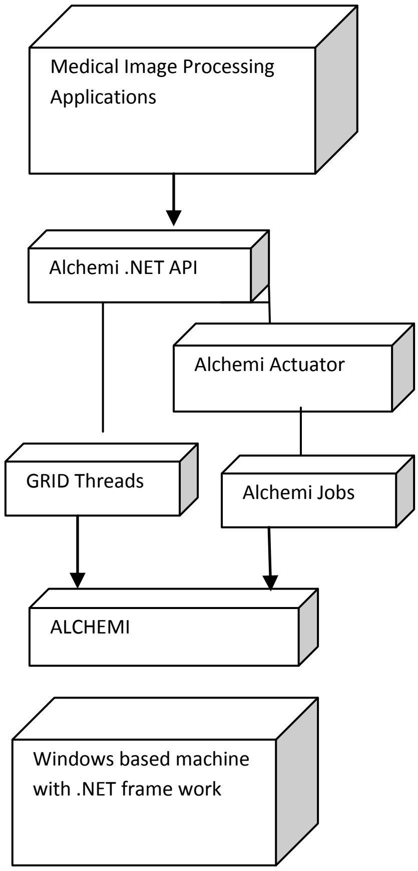


Fig 2. Layered Architecture of distributed Windows

2.1 ARCHITECTURE

Alchemi [1] follows the master-worker parallel programming paradigm in which a central component dispatches independent units of parallel execution to workers and manages them. This smallest unit of parallel execution is a grid thread. A grid application is defined simply as an application that is to be executed on a grid and that consists of a number of grid threads. Grid applications and grid threads are exposed to the grid application developer via the object- oriented Alchemi .NET API.

Figure 3 shows an Alchemi architecture interacts with components. Alchemi tool describes the four components. They are

- Manager
- Executor
- Cross-platform Manager
- Owner

These components allow Alchemi to be utilized to create different grid configurations desktop cluster grid, multi cluster grid, and cross-platform grid (global grid).

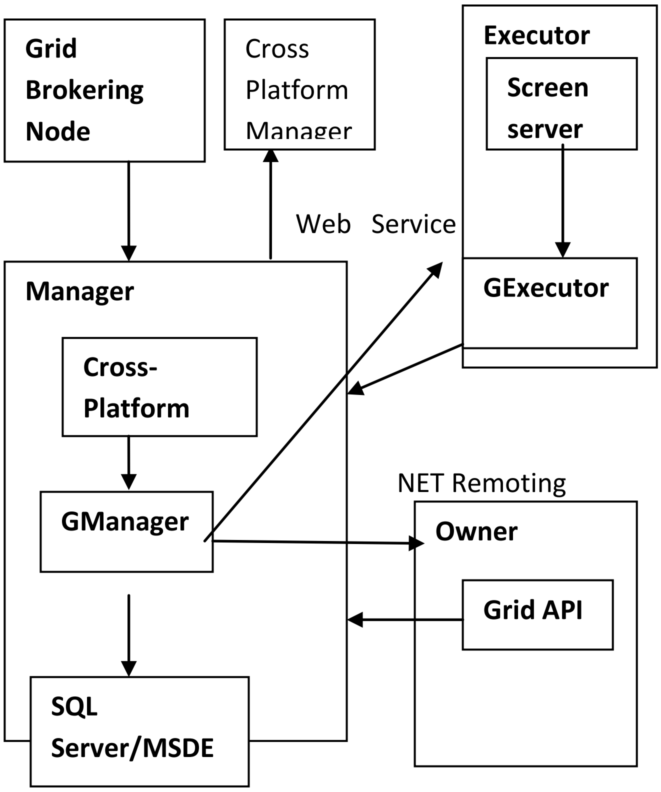


Fig 3. Alchemi architecture interact with components

2.1.1 Cluster Desktop grid

The cluster desktop grid as shown in figure 3, It consists of a single Manager and multiple Executors are configured, connect to the Manager. One or more Owners can execute their applications on the cluster by connecting to the Manager. Such an environment is appropriate for deployment on Local Area Networks as well as the Internet. The operation of the Manager, Executor and Owner components in a cluster is as described above.

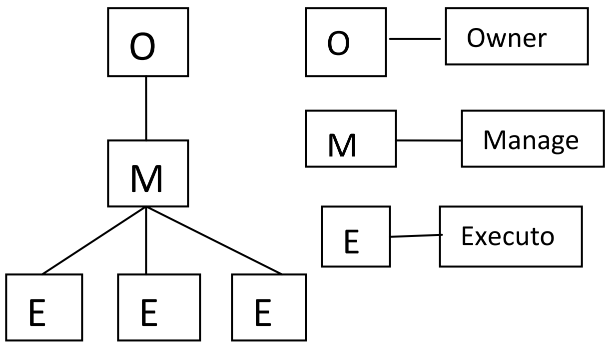


Fig 4. Cluster desktop grid

2.1.2 Multi Cluster deployment

A multi-cluster environment is created by connecting Managers in a hierarchical fashion (Figure 4). As in a single-cluster environment, any number of Executors and Owners can connect to a Manager at any level in the hierarchy.

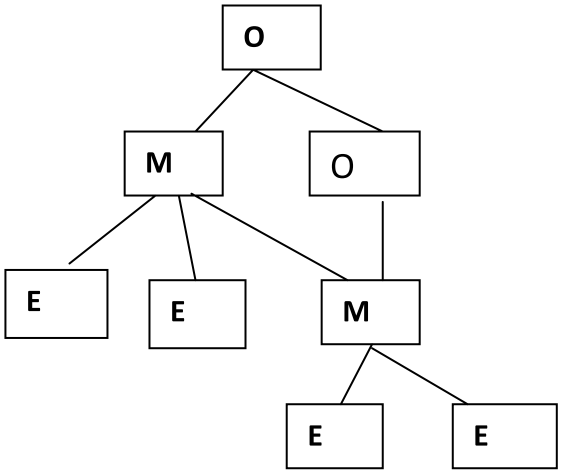


Fig 5. Multi cluster deployment

The key to accomplishing multi-clustering in Alchemi's architecture is the fact that a level Manager as an "intermediate" Manager and is treated by the higher level-Manager as an Executor. Such an environment is more appropriate for deployment on the Internet.

2.1.3 Cross-Platform Manager

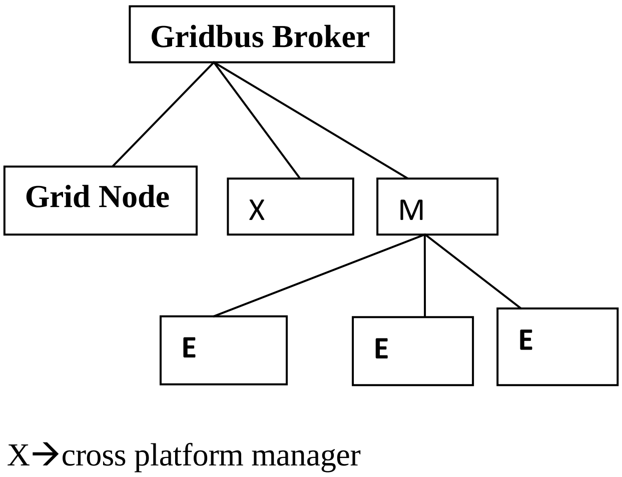


Fig 6. Cross-platform (global grid) deployment

A grid middleware component such as a broker can use the Cross-Platform Manager web service to execute cross-platform applications (jobs within tasks) on an Alchemi node (cluster or multi-cluster) as well as resources grid-enabled using other technologies such as Globus.

3. EXPERIMENTAL RESULTS AND DISCUSSION

Alchemi imitates traditional multi threaded programming. GThread is a grid thread and GApplication is application thread. Just in time scheduler algorithm is used for splitting the comparison job into different threads.

Assume that *T1, T2, T3….Tn.* Alchemi owner provides an interface with the grid application between the application developer and the grid.

The owner submits the completed threads to the Alchemi Manager (Figure 7). The Alchemi Manager manages the execution of images which is fit into another image or not, and provides services associated with managing thread execution.

The Executors register themselves with the Manager which in turn keeps track of their availability. Threads received from the Owner are placed in a pool and scheduled to be executed on the various available Executors. The Executor accepts threads from the Manager and executes them. Executor API is used to make an interface with the Alchemi Manager.

In this study, has been taken 3 systems, 1 system is going to be act as Manager and others are executors. The figure 9 shows the configuration of Alchemi Manager.

In the figure, setup connection enables or starts the alchemi manager and the value of own port is set as 9000. Advanced tab shows the server name and the database name.

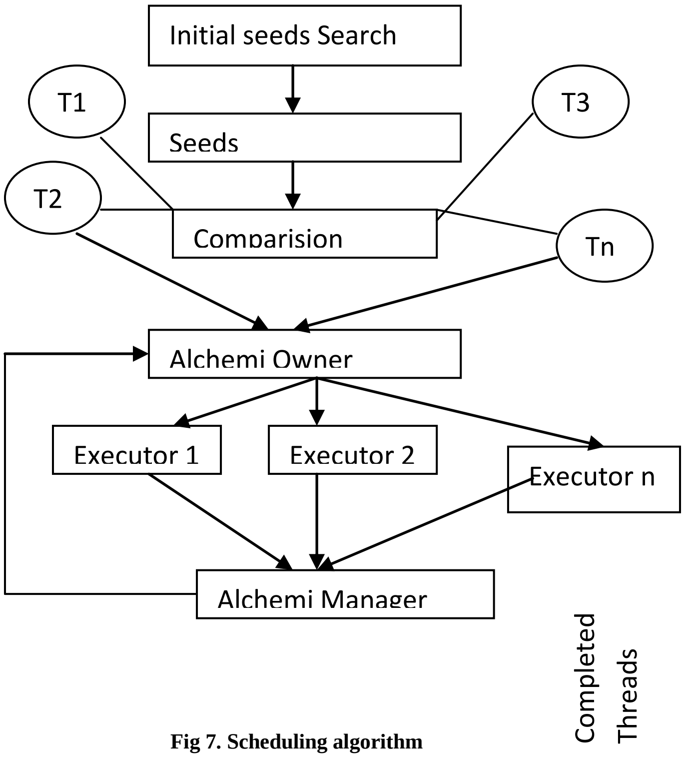


Fig 7. Scheduling algorithm

Fig.10 a) shows the Console parent form. This form is used to monitor the number of users, applications and executors. In this study, 2 systems CS32 and CS34 are executors. Here we can assign the rights for the executors.

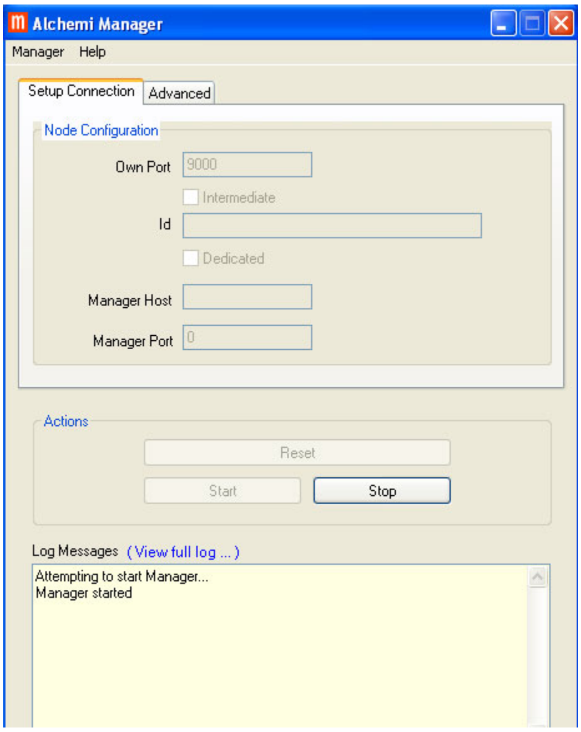


Fig 8. Alchemi Manager Configuration

Fig.8 shows the executor window. In this window, Own node id is used to identify the alchemi manager. By enabling the connect button the executor is connected with the alchemi manager

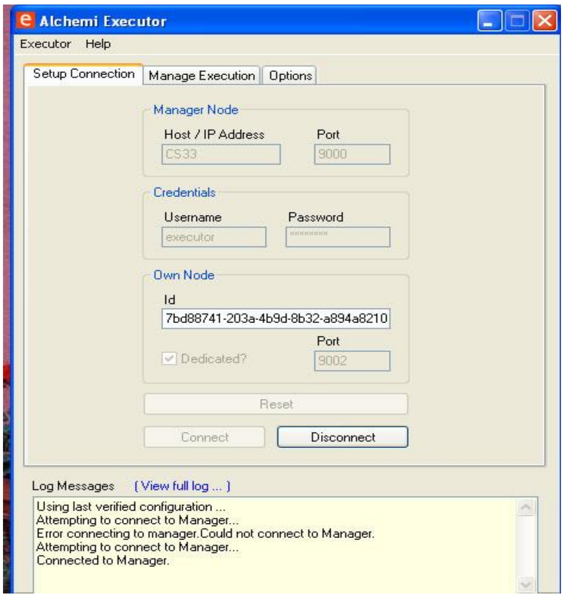


Fig 9. Executor

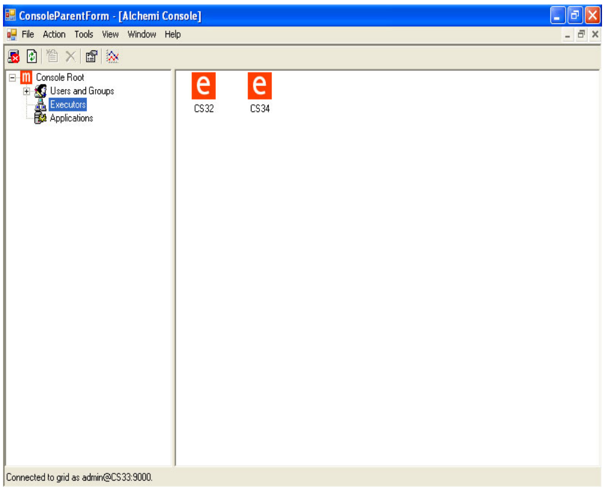


Fig 10 a). Console Parent Form

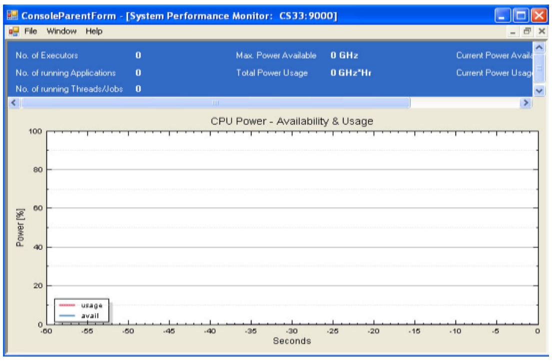


Fig 10 b) Before CPU Cycle Stealing

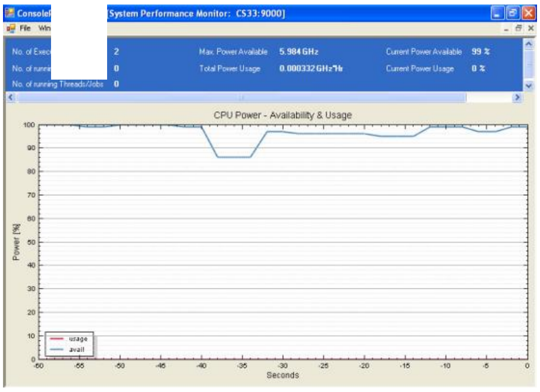


Figure 10 c) After CPU Cycle Stealing

Figure 10 (b) shows an Before CPU cycle stealing, and Figure 10 (c) shows an After CPU cycle stealing. Which represent the Executor detail, CPU utilization that is power usage of resources.

4. CONCLUSION

In this work presented the image registration process of brain image data. Then collected fMRI images and compared to the atlas images using Alchemi Grid tool. The time delay taken for comparing standard image into target image is too long. To make comparisons process, fast the iterations is done by the using of Alchemi. It provides the processing and managing the fMRI dataset, and reduces the CPU cycle utilization, and also provides secured and reliable service. In this study, has been installed the Alchemi tool and stole the CPU cycles from the neighborhood systems.

5. REFERENCES

[1] Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal "Alchemi: A .NET-based Grid Computing Framework and its Integration into Global Grids" , GRIDS Lab, Australia.

[2] Epifanio Bagarinao, Toshiharu Nakai, Yoshio Tanaka "Medical Grid: Using Grid Technology for Brain Studies" PHILIPPINE INFORMATION TECHNOLOGY, FEBRUARY 2008.

[3] Yolanda Gil, Ewa Deelman, Mark Ellisman, Thomas Fahringer, Geo!rey Fox, Dennis Gannon,Carole Goble, Miron Livny, Luc Moreau, and Jim Myers. Examining the challenges of scientific workflows. Computer, 40(12), 2007.

[4] Srikumar Venugopal, Rajkumar Buyya, and Lyle Winton. A grid service broker for scheduling e-science applications on global data grids: Research articles. Concurrency and Computation: Practice & Experience, 18(6):685–699, 2006. ISSN 1532-0626. doi:http://dx.doi.org/10.1002/cpe.v18:6

[5] Rajkumar Buyya, Susumu Date, Yuko Mizuno-Matsumoto, Srikumar Venugopal, and DavidAbramson. Neuroscience instrumentation and distributed analysis of brain activity data: a case for escience on global grids. Concurrency and Computation : Practice & Experience, 17, 2005.

[6] Ewa Deelman, James Blythe, and et al. Mapping abstract complex workflows onto grid environments. Journal of Grid Computing, 1, 2003.

[7] E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. S. Katz. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. Sci. Program., 13(3):219–237, 2005.

[8] C. B. et al. Neurobase: Management of distributed and heterogeneous information sources in neuroimaging. In Di- damic Workshop, MICCAI 2004 Conference, 2004.

[9] Ewa Deelman, James Blythe, and et al. Mapping abstract complex workflows onto grid environments. Journal of Grid Computing, 1, 2003.

[10] Christian Barillot et al. Neurobase: Management of distributed and heterogeneous information sources in neuroimaging. In Didamic Workshop, MICCAI 2004 Conference, 2004.

[11] Jinjun Chen and Yun Yang. A taxonomy of grid workflow verification and validation. Concurrency and Computation: Practice and Experience, 20(4):347–360, 2008. ISSN 1532-0626.

[12] Jia Yu and Rajkumar Buyya. A taxonomy of scientific workflow systems for grid computing.SIGMOD Record, 34(3):44–49, 2005. ISSN 0163-5808.

[13] Junwei Cao; Jarvis, S.A.; Saini, S.; Nudd, G.R.; C&C Res. Labs., NEC Eur. Ltd, Sankt Augustin, Germany Gridflow:workflow Management for Grid computing,