A Platform for Distributed Analysis of Neuroimaging Data on Global Grids

Scott Kolbe[1], Tianchi Ma[2], Wei Liu[1], Wee Siong Soh[1], Rajkumar Buyya[2] and Gary Egan[1]

[1]Howard Florey Institute; [2]Department of Computer Science and Software Engineering, The University of Melbourne

{s.kolbe, w.liu,w.soh, g.egan}@hfi.unimleb.edu.au; {raj,tcma}@cs.mu.oz.au

Abstract

This paper presents a Grid environment developed for analysis of MRI brain data on global Grids. In the current experiment, the MRI data analysis tasks are composed and formulated as a parameter sweep application. This application was deployed using the Gridbus Resource broker. The clusters in our Grid environment were managed using PBS and SGE. To execute tasks on remote resources, the Gridbus broker used SSH services for some resources and Globus middleware services for others for initiating execution and management. In this experiment, 16 x 8.1MB images and 28 x 16.3MB images were processed using a volumetric analysis method. The total waiting time was 1800s and the total processing time was 1279.30s. The broker overhead was 521s. Completing this analysis on a single machine could take over twenty hours. This Grid architecture provides a useful time-saving analysis tool for neuroimaging applications.

1. Introduction

Analyses of Magnetic Resonance Imaging (MRI) neuroimaging data involve computationally intensive, high data input/output (I/O) and memory-intensive tasks. These tasks include: modeling and quantitative analysis of brain macro-structure including gross brain shape; degree of gyrification; white matter axonal fibre tract connectivity; temporal correlations of co-activated brain regions (functional connectivity); covariance patterns of brain region networks (effective connectivity); modeling and analysis of brain microstructure, including cortical lamination structure; degree of white matter myelination; shape analysis of hippocampi; estimation of intracranial volume; and improved normalisation of functional MRI (fMRI) image sequences. Performing these analyses on a single machine for a single subject can take more than twenty hours. As neuroimaging data sets comprise groups of many individual subjects, much of the analysis can be performed simultaneously, with potential for further parallelization of processing algorithms that will reduce computation time significantly.

1.1. Background to Grid environment

The lack of computational power and the distribution of data and algorithms are two major problems in MRI research [1, 2]. Centralised high performance computing facilities address these problems by creating local area networks (LAN) of connected computer systems to solve computationally intensive problems. However this alone cannot offer the computational power demanded by MRI applications.

Geographically distributed computing resources need to be coupled together logically as a unified resource and the distributed MRI data and algorithms for data analysis need to be shared among multiple sites. Grid computing technology that enables sharing, exchange, discovery and aggregation of geographically distributed resources can assist in solving the problems in MRI research [3, 4].

To meet large-scale resource requirements of MRI data processing, we propose the development of a Grid environment, involving partners who bring varied expertise in MRI image processing, core Grid

This research was supported by a major National Research Facility grant to Neurosciences Victoria, and by the National Neuroscience Facility, Melbourne and partially supported by ARC Discovery Project Grant and StorageTek Corporation Fellowship.

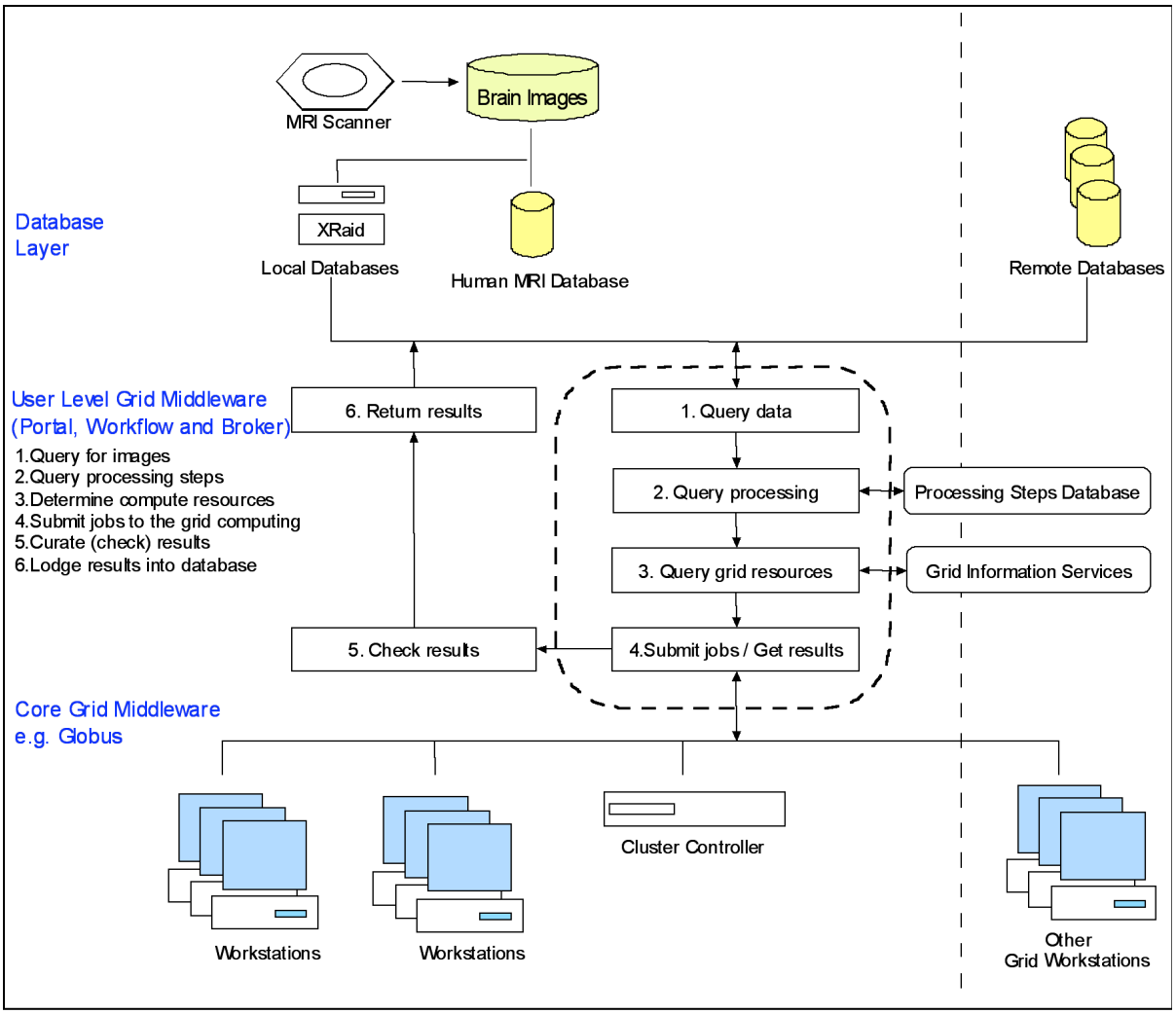

Figure 1. Grid architecture and users schematic view

computing, and infrastructure provision. The project aims to integrate local resources with state and national resources, which can interoperate with international infrastructure such as the Biomedical Informatics Research Network (BIRN) in the USA.

This paper details the use of a multi-site Grid computing facility to analyse the volumetric components of 44 structural brain images. This software enables grey and white matter and ventricular cerebro-spinal fluid (CSF) volumes to be calculated directly from MRI images. However the image processing operations are computationally intensive and require large investments of time and computing power for analysis of large groups of subjects. The following experiments demonstrate the time saved by distributing computations into a Grid environment.

2. Grid Architecture for Neuroimaging Analysis

The Grid environment addresses the problem of excessive wait time for analyses of large MRI image data sets, which currently lowers the productivity of researchers significantly. By targeting this research bottleneck, research results can be obtained sooner. By deployment of the Grid facility, other neuroscience researchers in Australia will have the opportunity to take part in collaborative research. This will benefit researchers by providing access to pooled data and computing resources while increasing collaboration by enabling sharing of results.

Figure 1 shows the schematic view of the Grid computing strategy for MRI analysis. This infrastructure is designed to enable the large-scale exchange of MRI

data, as well as the exchange, development and evaluation of MRI analysis software tools (algorithms).

To establish a Grid environment, middleware tools [5] are needed to support services such as security, uniform access, resource management, scheduling, application composition, computational economy, and accounting. One widely used Grid middleware software is the Globus Toolkit [6,7] which offers resource management (GRAM), data management (GridFTP), and information services (MDS).

Although the Globus Toolkit provides the basic services and libraries needed to build computational Grids, development and deployment of Grid applications are often knowledge-specific and therefore still present challenges for Grid system developers. Currently it is difficult to reuse software modules developed and deployed directly on top of the Grid service layer (core middleware layer) because of the close integration of application specific code. Therefore, the implementation of a user level middleware layer will provide a set of tools to simplify and optimize the development and deployment of future Grid applications. In particular the Grid middleware tools we propose will allow for an abstracted implementation by hiding Grid resource specific details and providing application programming interfaces (API) for most of the generic functions needed by Grid applications. Thus MRI or any other future Grid applications developed using proposed user-level middleware services will require predominantly application specific code. Gridbus Resource Broker can be used for composition and formulation of MRI analysis tasks for Grid application and deployment on multifarious platforms in Global Grid.

3. Data Analysis Methods

3.1. MRI parameters

MRI image data for forty-four participants were obtained. MRI scans were performed on a Signa 1.5-T scanner (General Electric, Milwaukee, WI, USA). T1-weighted Fast Spoiled GRASS images (TE =2.2ms, TR =11.3ms, 124 slices, 256 x 256 pixel matrix and 18.6 cm field of view). Images were converted to ANALYZE format [Mayo Clinic, Rochester, MN], which consists of two files per image, a meta file and a binary data file.

3.2. Data Analysis

Data was analysed using three software tools contained in the FSL brain image analysis software package (Analysis Group, FMRIB, Oxford University, UK). First the skull was automatically removed using the Brain Extraction Tool (BET) [8]. BET firstly analyses the intensity histogram of the image to find "robust" lower and upper intensity values and a rough brain/non-brain surface. This surface is used to calculate the centre of gravity of the head and approximate head size. From this centre of gravity a tessellated sphere is inflated and deformed to find a solution to the brain's external surface. This process is re-run with a higher smoothness constraint if the initial deformation does not produce a clean result. All extraneous matter is removed leaving an image of the brain only.

After extraction, the FLIRT linear image registration program [9] was used to register the skull to a reference image skull and then normalize the brain using the same transformation. FLIRT uses voxel intensities and various user selectable cost functions to produce a registration solution that is robust to local minima. Using the skull size as the registration constraint avoids losing volumetric data from the brain itself.

The FAST segmentation tool [10] was used to segment the brain into white matter, grey matter, and CSF and calculate the volumes of each segment. The FAST algorithm uses a hidden Markov random field (HMRF) model to create probabilistic maps of segmented volumes using neighbouring voxel intensity information in three dimensions. Figure 2 shows the steps involved in the analysis of a single subject.

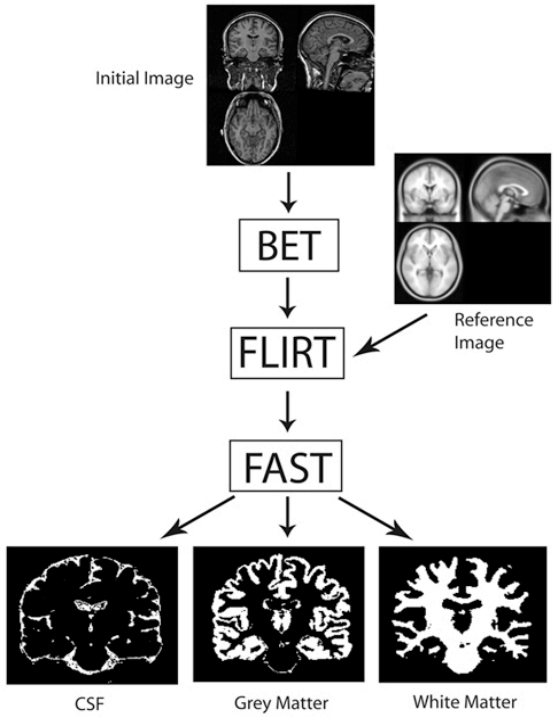

Figure 2. Schematic view of processing steps involved in volumetric analysis of MRI data.

Statistical analyses were performed using SPSS-11 for OS X (Mackiev Software, Kiev, Ukraine). To assess the determinates of total brain volume, a multivariate linear regression was used with white matter volume and grey matter volume as independent variables and total brain volume as the outcome variable. These results were then graphed as functions of age and regression variables calculated.

4. Application Composition and Analysis on Global Grids

4.1. Application Composition

The MRI data analysis tasks were formulated as a parameter sweep and workflow application using XPML (XML-based Parametric Modeling Language). The application involved processing of MRI data image files, represented by a parameter "MRI_ID". Figure 3 shows the XPML file describing the analysis operation.

A total of 44 MRI images with both image (.img) and header files (.hdr) were processed in the experiment. The files were distributed into a directory tree and a shell script linked the files into one directory with a standardised naming convention exp_[$N].[img/hdr] (e.g. exp_22.img and exp_22.hdr).

The five key parts of XPML code shown in Figure 3 are discussed below:

Part 1 defines the parameter-sweep, a variable *MRI_ID* that varies from 1 to 44 steps. For each value, a new job will be created and assigned a *task*.

Part 2 specifies the memory requirement of each job which is transformed to platform specific instructions by the Gridbus broker.

Part 3 comprises several *copy* commands with the data source marked as *local* and destination marked as *node*; "local" refers to a resource from which the broker launches executions and "node" refers a Grid resource selected by the broker's scheduling algorithm for job execution. These copy operations are called *stage-in*. The Gridbus broker transfers files from the local node to the remote execution node according to the stage-in commands. Note the variable *MRI_ID* is used with a *$* prefixed to indicate a substitution will occur using the value of *MRI_ID* for each job.

Part 4 comprises a set of *execute* commands which are carried out on the remote execution node. Variable substitution can also be used. The *execute* commands are executed sequentially.

Part 5 is comprised of *stage-out* commands responsible for collating results from various Grid resources and sending them to the user's workstation.

It requires the final output files of the application to be transferred back to the local node where the Gridbus broker sits. It can also be a third party node. Variable substitution is also used here.

4.2. Resources and Setup

The resources used in the experiments consisted of 3 clusters located in Howard Florey Institute (HFI) in Melbourne; GRIDS Lab in Melbourne, and the Australian Partnership for Advanced Computing (APAC) in Canberra. The Belle Grid server served as a client running the Gridbus Broker from which the experiment was launched. Table 1 shows configuration details of these resources. The APAC cluster was running Globus middleware, while other clusters were running Sun N1 Grid Engine (SGE) and Portable Batch System (PBS) respectively and were accessed via an SSH channel. The broker is designed to support management of job execution on remote resources using SSH-based connection and execution in addition to Globus-based access. For the SGE or PBS cluster, we established an SSH channel for staging the input images to a proper client node and ran the job submission commands remotely.

For defining the clusters, the user is required to give the remote hostname, as well as other platform relative configurations (e.g. the queue setting, using of SSH, firewall enabled, type of shell…) to assist the Gridbus broker in submitting jobs correctly.

Table 1: Australian Grid resources used in the experiment.

Grid Resource, Organisation	Cluster Details	CPU/Memory	OS	Middleware	Queue limit
Manjra Cluster, Uni. of Melbourne	16 nodes, x86	Intel P4 2.4 GHz / 512 MB	Linux RedHat 7.3	SSH-based Access, Open PBS	5
Mac Cluster; HFI	12 nodes, PowerPC	Power G5 2 x 1.8GHz/ 2 GB	Mac OS 10.3.9	SSH, Sun N1 Grid Engine 6 (SGE)	5
APAC, Canberra	154 nodes, x86	Intel P4 2.8 GHz / 1 GB	Linux RedHat 8.0	Globus Toolkit 2.4 – PBS job manager	2
Belle Grid Server, Uni. of Melbourne	Gridbus broker client node	Intel Xeon 2.8 GHz * 4 / 2 GB	Linux RedHat 8.0	Gridbus Broker 2.2	N/A

```
<!--Part 1: parameter definition -->
<parameter name="MRI_ID" type="integer"
domain="range">
  <range from="1" to="44" type="step"
   interval="1"/>
</parameter>
<!--Part 2: job resource requirements -->
<requirement type="job">
  <property name="minmemory" value="200"/>
</requirement>
<task type="main">
<!--Part 3: data staging -->
  <copy>
    <source location="local"
     file="sample/exp_$MRI_ID.img"/>
    <destination location="node"
    file="exp_$MRI_ID.img"/>
  </copy>
  <copy>
    <source location="local"
     file="sample/exp_$MRI_ID.hdr"/>
    <destination location="node"
    file="exp_$MRI_ID.hdr"/>
  </copy>
<!--Part 4: processing on Grid nodes-->
  <execute location="node">
    <command value="bet"/>
    <arg value="exp_$MRI_ID"/>
    <arg value="exp_$MRI_ID_brain"/>
  </execute>
  <execute location="node">
     <command value="flirt"/>
    <arg value="-in"/>
    <arg value="exp_$MRI_ID_brain"/>
    <arg value="-ref"/>
    <arg value="$refimg"/>
    <arg value="-out"/>
    <arg value="exp_$MRI_ID_brain_reg"/>
  </execute>
  <execute location="node">
    <command value="fast"/>
    <arg value="exp_$MRI_ID_brain_reg"/>
  </execute>
<!--Part 5: collating results to home node-->
  <copy>
    <source location="node"
    file="exp_$MRI_ID_brain_reg_seg.hdr"/>
    <destination location="local"
     file="exp_$MRI_ID_brain_reg_seg.hdr"/>
  </copy>
  <copy>
    <source location="node"
     file="exp_$MRI_ID_brain_reg_seg.img"/>
    <destination location="local"
     file="exp_$MRI_ID_brain_reg_seg.img"/>
  </copy>
```

Figure 3: XPML-based composition of MRI image Grid application.

For each resource, a queue limit is defined as the maximum number of jobs that can be executed on a particular resource in parallel. This depends on the resource's usage policy regulated by the resource provider. If a queue limit is not given, the broker will try to discover the resources' potential availability by submitting some test/query jobs. The broker will suspend the job submitting to a resource if the number of jobs currently executing on that resource has reached its queue limit.

5. Neuroimaging Results

The results of a multiple linear regression with total brain volume (TBV) as the dependant variable and white matter volume (WMV) and grey matter volume (GMV) as independent variables show that WMV (t= 1342) is slightly more highly associated with TBV than GMV (t= 1324). This analysis shows that variations in white matter and grey matter explain variations in total brain volume equally in this group.

Figure 4 shows the variables WMV and GMV graphed as functions of age. All regression parameters are listed on the graph. As can be seen from the graph, grey matter makes up a larger part of the total brain volume than white matter and it appears to follow a more age related pattern of volume loss than does white matter. This is reflected in both the R-squared values (0.51 for grey matter versus 0.01 for white matter) which are measures of vriability, and the regression coefficients (-3.32 for grey matter versus –0.39 for white matter) which are measures of gradient.

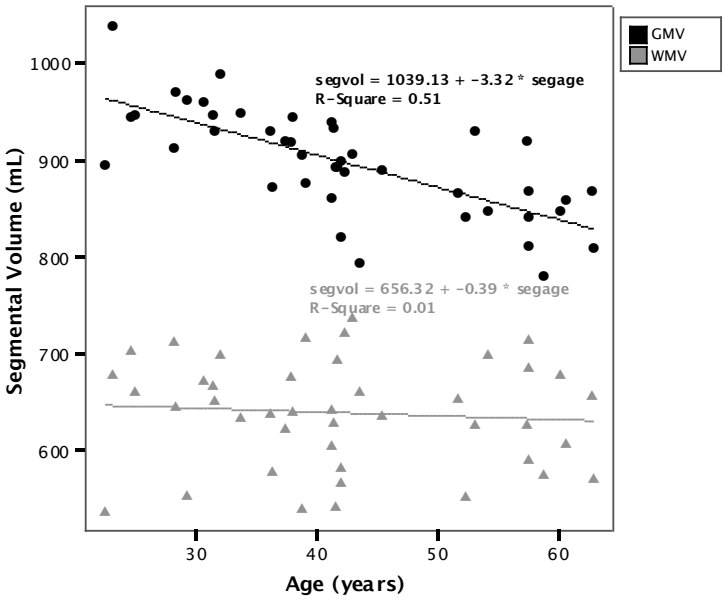

Figure 4: Grey and white matter volumes for 44 subjects as functions age. (Regression equations are listed above each fit line.)

6. Results of Grid computing performance

The maximum throughput of a resource at a certain time is estimated by $T_R \geq T_n - QL / P$, where T_R refers to the resource throughput, T_n refers to the processing ability (on the application) of each computing node provided by the resource, QL refers to the queue limit, and P refers to the maximum number of jobs allowed to be executed in parallel on each computing node. For the purpose of estimation, P was set to 1 in this experiment, so $T_R \approx T_n - QL$.

T_n was discovered for each cluster by running the *stage-in -> bet -> flirt -> fast -> stage-out* flow manually on each resource. The images in our experiment were divided into 2 sets, one set of 16 x 8.1MB 8 bit images, and one set of 28 x 16.3MB 16 bit images. This reflects the varying size and quality of neuroimaging data. One image from each set was selected and the result shown in Table 2.

For the 16.3MB images:

$$T_{n(SGE)} : T_{n(PBS)} : T_{n(Globus)} = (1/283.61) : (1/323.10) : (1/329.16) = 1.16 : 1.01 : 1$$

For the 8.1MB images:

$$T_{n(SGE)} : T_{n(PBS)} : T_{n(Globus)} = (1/235.92) : (1/297.79) : (1/306.82) = 1.30 : 1.03 : 1$$

Figure 5 shows the number of jobs submitted and finished as functions of time elapsed for each of the three platforms. Figure 6 shows the total jobs submitted and finished by different resources at different times.

Figure 7 shows the comparison of the number of jobs completed by three different resources. Eight jobs were finished on the Globus platform with queue limit set to two, while 18 jobs were finished on both the PBS and SGE platform with both queue limits set to 5.

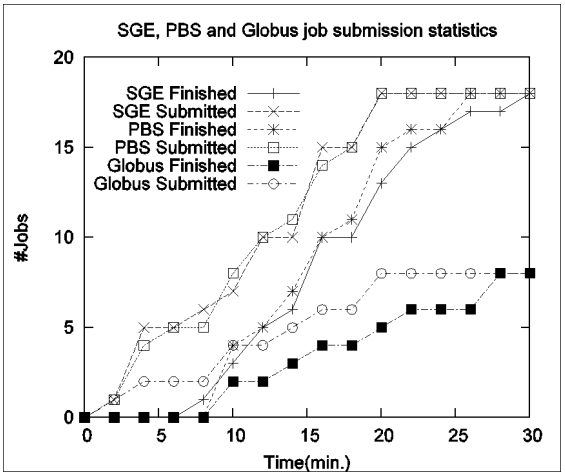

Figure 5. A Snapshot of number of jobs submitted and finished by on the resources at different time.

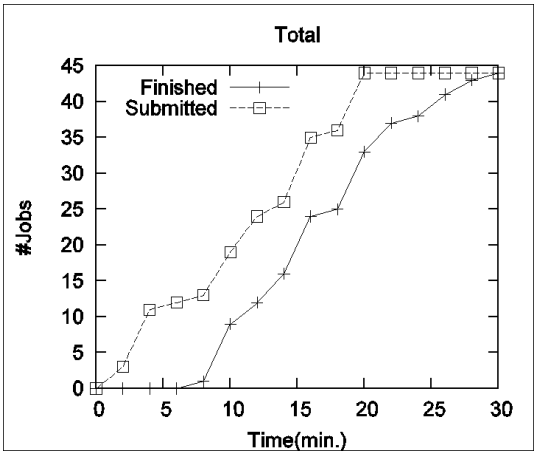

Figure 6. A Snapshot of number of jobs submitted and finished by all resources at different time.

7. Analysis of Grid performance

The total waiting time in this experiment was 30 minutes (1800 seconds), from which the broker overhead can be calculated. Figure 8 shows the scenario that jobs being scheduled resources. The ideal total waiting time of the experiment can be estimated to be $\max\{T_{SGE}, T_{PBS}, T_{Globus}\}$, where T_{SGE} can be calculated by aggregating all the execution time of all jobs (including the small and large images) together:

$$T_{[SGE/PBS/Globus]} = \sum T_{stage-in} + \sum T_{stage-out} + 1/5 \times \sum T_{exec}$$

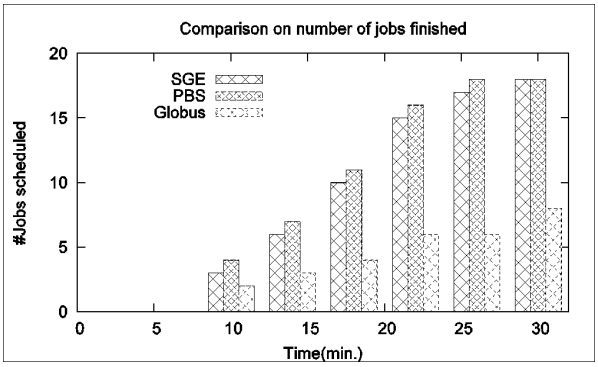

Figure 7: Number of jobs completed by different resources at different time.

Table 2: A single node performance comparison.

Image size (MB)	Resource	Time taken (sec) for different operations to complete			
		stage-in	Exec	stage-out	Total
8.1 MB image	Manjra Cluster, Melbourne	2.19	232	1.42	235
	Mac Cluster, HFI	1.72	294	1.25	297
	APAC, Canberra	4.13	300	2.25	306
16.3 MB image	Manjra Cluster, Melbourne	3.39	278	1.75	283
	Mac Cluster, HFI	2.25	319	1.59	323
	APAC, Canberra	7.01	318	3.53	329

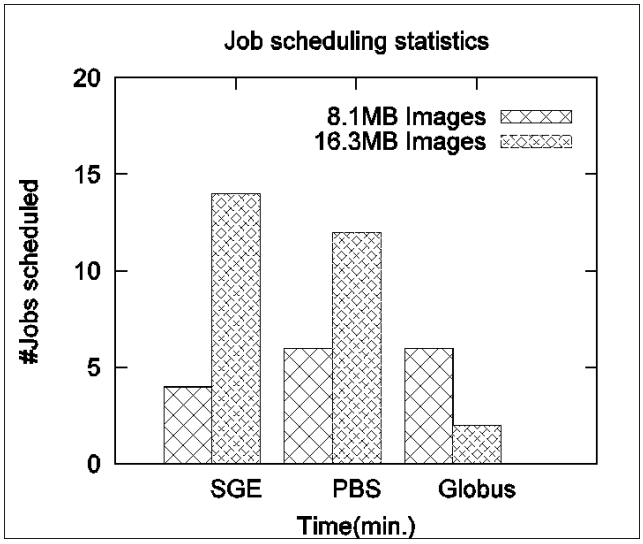

Figure 8: Number of small and large jobs scheduled to different Grid resources.

According to Table 2, we have:

$$T_{SGE} = (2.19 + 1.42) * 4 + (3.39 + 1.75) * 14 + (232.31 * 4 + 278.47 * 14) / 5 = 1051.96$$

$$T_{PBS} = (1.72 + 1.25) * 6 + (2.25 + 1.59) * 12 + (294.82 * 6 + 319.26 * 12) / 5 = 1183.91$$

$$T_{Globus} = (4.13 + 2.25) * 6 + (7.01 + 3.53) * 2 + (300.44 * 6 + 318.62 * 2) / 2 = 1279.30$$

The execution time can be parallelized, while the data staging cannot, because the stage-in and stage-out shares the same bandwidth. The estimated ideal waiting time is:

$$T_{total} = \max\{1051.96, 1183.91, 1279.30\} = 1279.30$$

Therefore the total overhead is about 1800-1279=521 seconds. This includes the cost of Gridbus broker startup, scheduling, remote job submission and monitoring, stdout/stderr transfer, as well as the local job submission/schedule time of PBS, SGE and Globus.

8. Conclusion and Future Work

The computational requirements of MRI neuroimaging data analysis application are presented in the context of Grid computing. The proposed Grid architecture develops the necessary mechanisms for composing MRI data analysis tasks as a Grid application. Data analysis was carried out on Grid resources using the Gridbus resource broker and various distributed clusters accessed using either SSH or Globus services. Experimental results of Grid time variables and scientific results derived from the analysis demonstrate the feasibility of a Grid computing environment for neuroscience applications that support collaborative scientific studies by sharing resources.

We plan to enhance the Grid environment by developing a portal based workflow environment with virtual organization-based authorization mechanisms at MRI data level including handling privacy concerns and curate results automatically. This will be achieved by making use of Gridbus workflow and Portlets hosted under GridSphere environment. We also plan to extend MRI databases access using Web services technologies.

9. References

[1] Egan, G. F. (2004). Neuroinformatics: the development of shared neuroscience databases and tools at the Australian National Neuroscience Facility. *The 11th International Conference on Neural Information Processing*.

[2] Amari, S., Egan, G. F., et al. (2002). "Collaborative neuroscience: neuroinformatics for sharing data and tools." *Integrative Neuroscience* **1**: 117-128.

[3] Buyya, R., Date, S., et al. (2004). "Neuroscience Instrumentation and Distributed Analysis of Brain Activity Data: A Case for eScience on Global Grids." *Concurrency and Computation: Practice and Experience*.

[4] Toga, A. W. (2002). "Neuroimage databases: the good, the bad and the ugly." *Nature Reviews Neuroscience* **3**(302-309).

[5] Biomedical Informatics Research Network (BIRN). - http://www.nbirn.net

[6] Foster, I. and Kesselman, C., *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1999.

[7] Venugopal, S., R. Buyya, et al. (2004). "A Grid Service Broker for Scheduling Distributed Data-Oriented Applications on Global Grids." *Proceedings of the 2nd International Workshop on Middleware for Grid Computing* (Toronto, Canada, October 18, 2004), ACM Press, 2004, USA.

[8] Smith, S. M., Zhang, Y., et al. (2002). "Accurate, robust, and automated longitudinal and cross-sectional brain change analysis." *Neuroimage* **17**(1): 479-89.

[9] Jenkinson, M. and Smith, S. (2001). "A global optimisation method for robust affine registration of brain images." *Med Image Anal* **5**(2): 143-56.

[10]Zhang, Y., Brady, M. and Smith, S. (2001) "Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm." *IEEE Trans. on Medical Imaging*, **20**(1):45-57.

[11]Buyya, R., Branson, K., Giddy, J., and Abramson, D., The Virtual Laboratory: Enabling Molecular Modeling for Drug Design on the World Wide Grid, *Concurrency and Computation: Practice and Experience*, Volume 15, Issue 1, Pages: 1-25, Wiley Press, USA, January 2003.

[12]Entropia, *Fight AIDS at Home Project: A joint effort of Entropia and Scripps Research Institute*, http://www.fightAIDSatHome.org/

[13]Smallen, S., Casanova, H. and Berman, F. , *Applying Scheduling and Tuning to On-line Parallel Tomography*, *Proceedings of the IEEE/ACM SuperComputing Conference* (SC 2001, Denver, CO, USA), IEEE Computer Society Press, Los Alamitos, CA, USA, 2001.