Neuroimaging Analysis using Grid Aware Planning and Optimisation Techniques

Irfan Habib, Ashiq Anjum, Peter Bloodsworth, Richard McClatchey
University of the West of England, Bristol, UK
[Irfan.Habib, Ashiq.Anjum, Peter.Bloodsworth, Richard.McClatchey]@cern.ch

Abstract

Neuroimaging research is increasingly shifting towards distributed computing architectures for the processing of ever growing neuroimaging datasets. At present compute and data intensive neuroimaging workflows often use cluster-based resources to analyse datasets. For increased scalability however, distributed grid-based analysis platforms may be required. Such an analysis infrastructure necessitates robust methods of grid-aware planning and optimisation in order to efficiently execute often highly complex workflows. This paper presents the approaches used in neuGRID to plan the workflow gridification and enactment for neuroimaging research. Experiments show that grid-aware workflow planning techniques can achieve significant performance gains. Turn-around time of a typical neuroimaging workflow reduces by 30% compared to the same workflow enacted without grid-aware planning. Data efficiency also increases by more than 25%. The use of workflow planning techniques in the neuGRID infrastructure may enable it to process larger neuroimaging datasets and therefore allow researchers to carry out more statistically significant analysis.

1. Introduction

Large-scale computing infrastructures such as clouds and grids enable the execution of applications that are both data and compute intensive. Such computing paradigms have played a part in accelerating advances in e-Science research. One domain that has started to leverage such infrastructures is that of neuroimaging analysis. Neuroimaging techniques extract features or information from brain scans. Recent progress in neuroimaging has led to a rapid growth of digital data and opportunities for sophisticated image analysis. This is particularly true in the case of images such as magnetic resonance (MR) imaging, positron emission tomography (PET), ultrasound imaging and others. Currently, large datasets of MR images are being collected worldwide that will eventually lend themselves to full scientific study. Such data repositories include the Alzheimer's Disease Neuroimaging Initiative [1] and the NIH MRI Study of Normal Brain Development [2]. These repositories and various specialised neuroimaging research centres around the world hold thousands of images and cumulatively hold dataset sizes of hundreds of terabytes.

The wealth of data acquired from the brain with state-of-the-art scanners has necessitated the development of a number of semi-automatic processing algorithmic toolkits. These have enabled the analysis of specific image features [3] [4] and provided collections of algorithms that chain together to form a neuroimaging workflow. Such workflows can analyse 2D or 3D maps of the brain according to features of interest. Often a number of hours are required to process each brain scan using traditional personal computers. Some toolkits [5] [6] can enact neuroimaging pipelines using more powerful cluster-based resources. This may well improve the overall processing time but a significant processing period is still often necessary.

The neuGRID Project[7] is an initiative which builds on existing research in neuroimaging based Grid infrastructures. The infrastructure will initially incorporate a base dataset of around 5000 brain scans. Processing this number of images using complex data and compute intensive neuroimaging workflows places a strong emphasis on efficient workflow planning. As an example, the CIVET cortical thickness analysis pipeline will be considered in this paper. CIVET may consist of up to 120 tasks depending on the configuration that is used. The turn-around time for this using a modern dual-core processor is around 8 hours for a single brain scan. The workflow also

produces ten times more data than it consumes. To enable neuroimaging researchers to quickly extract knowledge from thousands of brain scans and ensure short workflow turn-around times, an efficient use of computational and data resources is necessary and this cannot be achieved without smart workflow planning techniques

The focus of workflow planning in neuGRID is to increase the compute and data efficiency of a workflow. There are several workflow planning techniques. Some approaches focus on reducing the granularity of a workflow by clustering or grouping tasks. Other approaches concentrate on enhancing the data efficiency of a workflow by reducing or eliminating data transfers in a workflow. Some approaches are integrated with components of a given Grid middleware to make them application-aware [8] [9] [10]. The neuGRID project has adopted several workflow planning approaches that are intended to improve scalability and speed up the neuroimaging analysis. The component in the neuGRID architecture that is responsible for the authoring, planning and enactment of task based neuroimaging workflows is the Pipeline Service[8] .

This paper proceeds as follows. Section 2 briefly details related work in this area. Section 3 discusses the neuGRID Pipeline Service architecture. Section 4 considers the pipeline planning, and optimisation techniques that are employed to scale up the neuroimaging analysis process in the Pipeline Service. Section 5 presents results of the planned and optimised CIVET workflow compared with the existing CIVET implementation. Section 6 presents the conclusions and briefly highlights the future directions for this work.

2. Related Works

In terms of workflow planning and optimisation for neuroimaging analysis related work includes [11-13]. NeuroLOG uses MOTEUR [10] to plan service-based neuroimaging workflows. MOTEUR supports various service workflow planning techniques such as asynchronous service execution, data parallelism and workflow parallelism based techniques. Service-based workflows introduce some limitations to planning. For instance, services are stationary and they cannot be staged to any other Grid site. Hence input data has to be transferred towards the services. In the bioinformatics ^{my}Grid[14] project, this problem is solved by streaming data rather than transferring the entire input data set. In the case of task-based workflows none of this is required, as tasks can be migrated to the source of the data. Hence tasks based workflows enable the use of a full spectrum of planning techniques, which often lead to a more efficient optimisation.

Another related effort is the Computational Neuroscience Applications Research Infrastructure (CNARI) [12]. In the CNARI workflows are defined in the parallel scripting language SwiftScript [15]. Swift enables the declaration of abstract workflows and supports abstract to concrete workflow transformations. Richer planning is required for more complex neuroimaging workflows. Another related work is from S. Kolbe [13]. The paper details a Grid platform for neuroimaging analysis. In this infrastructure the neuroimaging workflows are formulated as a parameter sweep application. The focus in this infrastructure is highly parallel execution of relatively simple pre-processing workflows on a large number of images. In neuGRID optimisation for large custom-built workflows is required.

3. neuGRID Pipeline Service

The neuGRID Pipeline Service enables the authoring, specification, planning and enactment of neuroimaging pipelines. These workflows consist of a large number of atomic tasks performing several types of operations. These include segmentation, extraction of specific image features, linear or nonlinear normalisation, basic arithmetic on brain scans and others. The neuGRID Pipeline Service supports task-based neuroimaging workflow planning and enactment. Service-based workflows will be supported in the neuGRID infrastructure in future. This paper deals primarily with the task-based workflow planning, distribution and optimisation issues. Figure 1 shows the Pipeline Service architecture.

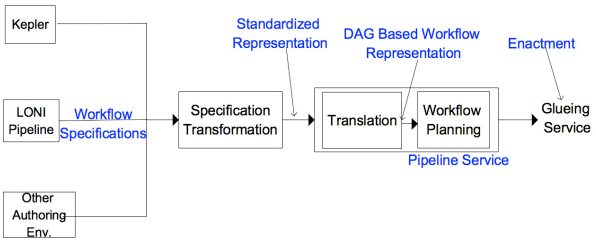

Figure 1: Pipeline Service Architecture

The user-facing component of the Pipeline Service supports numerous workflow-authoring environments, such as the LONI Pipeline [5] and Kepler [16]. These workflow-authoring environments use the Pipeline Service Translation component to interact with the Pipeline Service and the neuGRID infrastructure. Users may author workflows in one of these environments. The Pipeline Service then plans the workflow for

efficient enactment and submits it to the Glueing Service for execution.

This service shields the Pipeline Service and other neuGRID generic services from direct lock-in to a specific Grid middleware. The Glueing Service is a web service binding of the OGF SAGA specification[17]. The OGF SAGA specification provides a unified API for accessing heterogeneous Grid middleware. It promotes an adaptor-based architecture for accessing different Grids platforms in an interoperable way. The Glueing service is detailed in [7].

4. Workflow Planning in the neuGrid Pipeline Service

In order to execute user-authored pipelines efficiently in a Grid environment the workflow needs to be transformed into a grid executable state. This transformation is often termed as "planning", "grid-aware distribution" or "grid-enabling". The workflow formalism used in the Pipeline Service is the directed acyclic graph. Other workflow formalisms such as Petri-nets enable richer workflow specifications such as iteration structures however neuroimaging workflows rarely require them. This is because the main purpose of neuroimaging workflows is to string together processing algorithms in order to extract and compute certain features from brain scans.

The workflow planning approach that is appropriate for a given case depends on the nature of application specific workflows. Some applications have large and complex workflows and the input data set sizes are relatively small such as a single brain scan. Other applications in e-Science however have relatively small workflows but the input data set sizes are in the giga-byte range. The planning techniques that are applicable for both of these types of applications are different. Therefore to determine the applicable workflow planning techniques relevant to neuroimaging anlaysis, a study of a workflow that will be extensively used in neuGRID was carried out.

In the neuGRID infrastructure neuroimaging researchers can use CIVET to calculate cortical thickness at each vertex on both registered and native spaces on hemispheric surfaces that have been non-linearly registered. Moreover, CIVET is used to retrieve a number of related measurements from brain scans. It is a compute intensive workflow that generates up to 1000 percent more data than it consumes and in a modern dual core CPU it takes around 8 hours to process. CIVET uses the PMP framework[6] for execution on cluster resources managed by the Sun Grid Engine[18]. Figure 2, maps the distribution of run-times of the tasks in the CIVET pipeline when enacted with a T1 weighted brain scan. From the results, apparently CIVET consists of a significant number of granular tasks. Granular tasks are tasks that have a very short execution time.

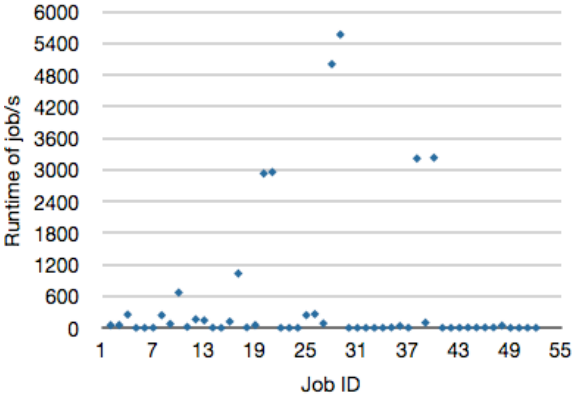

Figure 2: Distribution of the runtime of jobs in CIVET

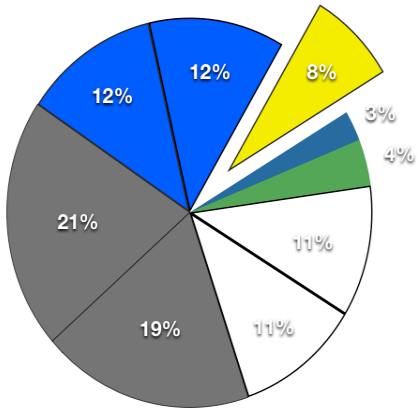

Figure 3: Visualization of the run-times of coarse grained jobs and cumulative fine grained jobs

These tasks number 44 in total. As numerous overheads [19] are associated with each task submission in a Grid environment, these granular tasks will severely affect the runtime performance of the workflow. Figure 3 shows the processing times of various tasks in the workflow as a percentage of the whole workflow turn-around time. The most compute intensive tasks are the grey matter extraction from the right and left cortex hemispheres, taking respectively 19% and 21% processing times (coloured in grey) of the entire workflow. Surface registration tasks for both hemispheres take 12% of the processing each (coloured in blue), while white matter extraction in each hemisphere takes 11% of the processing time (in

white). These tasks are coarse-grained tasks. Fine grained tasks on the other hand, number 44 and cumulatively perform 8% of the computations (highlighted in yellow).

In order to avoid unnecessary latencies for granular tasks, the granularity of the workflow needs to be reduced. Another requirement from the neuroimaging researchers is that neuGRID should support data efficient enactment of workflows. Workflow should be enacted which leverage existing data available in the Grid environment efficiently.

Given the requirements of the application the following two types of workflow planning approaches were identified as suitable.

4.1 Task Reduction Based Approaches

The primary approach that is used to reduce the granularity of the workflow is task clustering or grouping approaches. Various workflow planner/enactment engines [8] [20] provide task clustering methods. Two primary types of task clustering have been implemented in state-of-the-art planners. Both approaches are evaluated in section 6.

1. *Automated Horizontal Clustering:* Horizontal clustering is efficient when multiple tasks are associated at each level of processing. The planner determines the levels in a workflow by performing a breadth-first based graph traversal of the workflow starting from the root nodes. For each level, the jobs are grouped and each group is submitted to a single site for execution. Planners provide capabilities to customise the behaviour of automated horizontal clustering. For example, in Pegasus [8] users can define two factors to control the behaviour of the automated clustering. The user can define a collapse factor, which specifies the maximum number of jobs that can be part of a single cluster. The other parameter is the bundle factor, which denotes the number of clusters to be generated per level. Both of these parameters can be used to configure the granularity of the automated clustering.

2. *User Defined Clustering:* In this type of clustering, the user manually defines the task cluster in the workflow. All jobs that are manually grouped into a cluster by the user are dispatched as a single job to the Grid. This clustering method is useful when users are expert in their domain and are aware of the behaviour of specific workflow tasks and can formulate appropriate clusters manually.

4.2 Data Efficiency based techniques

Data efficiency based techniques try to minimise data transfers. This is achieved by either eliminating tasks which produce data that already exist in the Grid environment or schedule computations in a manner that is more efficient in terms of data. A widely deployed approach, and one which is used in neuGRID, is to determine if the data outputs to be generated in a workflow already exist in the Grid environment. A planner determines the availability of the output based on the logical file names supplied by the user when defining a workflow. The planner queries the Grid replica catalogue to find instances of data outputs that will be generated in the workflow. If such instances are found, the tasks that lead up to the generation of the data are eliminated. Another approach is termed as "data diffusion" [9]. During execution of a workflow, a planner that is integrated with a Grid scheduler, acquires compute and storage resources dynamically and stages data sets which may be required by the workflow in future. When a relevant task is to be scheduled it is sent to a resources which is close to a replica of the data, hence reducing data transfer latency and wait times for the task.

Data efficiency based approaches are often beneficial for planning neuroimaging workflows. This is because numerous workflows in a neuroimaging analysis have related operations. For instance in neuGRID, the pre-processing steps required for both creating the cortical mask for a T1 weighted brain scan and extracting white matter hemispheric masks have the same pre-processing steps that span ten tasks. If both workflows are executed separately on the same image, the latter workflow will only execute three tasks as opposed to the thirteen step execution process that otherwise may be required for final results. Many of these tasks can be eliminated and data generated by the execution of a previous workflow could be reused. This leads to more efficient use of the available data in the Grid infrastructure.

5. Planning, Optimization and Gridification of the CIVET Pipeline

CIVET is a typical neuroimaging workflow that is being deployed in the neuGRID infrastructure. The planning techniques highlighted in section 4 were used to plan and optimise the workflow. The aim of the experiments is to determine the efficacy of various workflow planning techniques. Table 1 highlights the infrastructure that was used to carry out the experiments. A set of 10 virtual machines run on VMware Server 2 was used to create 3 pools of Condor clusters and a 10 node SGE cluster. The specification of each virtual machine was identical and is stated in Table 1 . The planned workflow was executed in a Grid of three Condor cluster pools. The workflow was planned with Pegasus and enacted through DAGman. The standard CIVET workflow was executed in a 10-

node SGE Cluster. The enactment was performed by PMP. Pegasus is a suitable planner for neuGRID, because it supports user defined and automated task clustering as well as task elimination based on data availability. Apart from the compute nodes, a NFS server was used for shared storage in case of both the SGE cluster and the Condor cluster pools.

Figure 5 shows the turn-around time results for the example workflow. The workflow was executed using automated horizontal clustering. Figures 5, 6 and 7 respectively show the results of automated clustering, user defined label based clustering and the standard CIVET. In the automated clustering the workflow was executed with collapse and bundle factors of 2, 3 and 4. The factor limit of 4 was chosen based on the structure of the workflow. The maximum number of tasks at any single level in the workflow was eight. Based on this the significance of the results would diminish with increased factor values, as either very few jobs would be submitted per level or a large number jobs would be generated. In the subsequent sections, an analysis of the results of the automated clustering as well as the label based clustering is presented.

Pegasus Hardware setup	3 Condor Clusters Each Cluster: 1 Master and compute Node 2 compute Nodes	Each node, a virtual machine consisting of: 2.4Ghz single core 1GB RAM 20GB dedicated Hard disk (7200RPM, SATA II) 1Gbps Ethernet connectivity
SGE Hardware setup	10 Node SGE Pool 1 SGE Master Node 9 SGE Exec Nodes	
NFS Shared Storage	500GB, on a System of AMD Phenom 9850 1 Core 8GB RAM, 1GBps connectivity	

Table 1: Experimental infrastructure details

5.1 Automated Workflow Planning

From Figure 5, 6 and 7 it is apparent that there are significant differences in terms of the behaviour of workflows when enacted with a bundle or a collapse factor.

As we can observe increasing the value of the collapse factor from 2 to 3 increases the workflow turn-around time by more than 30 minutes or 14.7%. Increasing the collapse factor further increases the turn-around time by a further 4 minutes. Collapse factor based automated clustering created more computationally inefficient workflows than the standard non-workflow planned CIVET workflow. At the same time, increasing values of the collapse factor produces increasingly more data efficient workflows. As we can see in figure 6, the workflow consumes 402 MB of data per image when enacted with a collapse factor of 2, increasing this value to 3, reduces the amount of data read by the workflow by 8.8%. At a collapse factor of 4 the amount of data read drops further by 2.9%. The scheduling latency decreases as well with an increasing collapse factor value, a reduction of 12.5% is noted at a collapse factor of 3 and at a collapse factor of 4 the scheduling latency decreases by a further 6.7%. The results show that increasing the value of the collapse factor has direct correlation with reduced scheduling latency and reduced data retrieval at the expense of increased workflow turn-around time. Bundle factor, as previously stated, defines the number of clusters per level to be created. As we can observe increasing the bundle factor has an opposite effect on the workflow enactment to that of the collapse factor. A bundle factor of 2 produces a workflow which is efficient in terms of computation and has a reduced turn-around time at close to 139 minutes.

By increasing the values of the bundle factor the turn-around time of the workflow increases moderately by 3.5% (at a bundle factor of 3) and then by a further 4.1% (at a bundle factor of 4). In terms of data efficiency increasing the bundle factor creates less data efficient workflows. A total amount of 384 MB is retrieved from the NFS server at a bundle factor of 2, while 412MB is retrieved at a bundle factor of 4, an increase of 7.2%. The scheduling latency increases significantly as the bundle factor grows. An increase of 33.3% is noted at a bundle factor of 4 compared to the scheduling latency at a bundle factor of 2.

Larger values of the collapse factor increase the size of clusters, and hence reduce the workflow granularity. Clustering, groups a number of tasks into a single Grid submission, and adversely affects the parallelism and potential distribution of a workflow. Jobs that are clustered together are executed sequentially by Condor. For instance compute intensive operations such as surface registration and gray matter extraction occur in the same horizontal level of the workflow. Due to the value of the collapse factor these tasks are grouped together and executed sequentially.

Both of these tasks are compute intensive and hence executing them sequentially increases the turn-around time of the workflow. However both of these sets of compute intensive operations retrieve a number of common data sets, such as the cortical mask or linear transformation of the original brain scan.

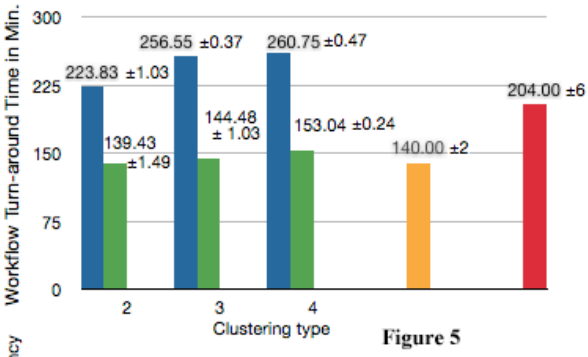

Figure 5

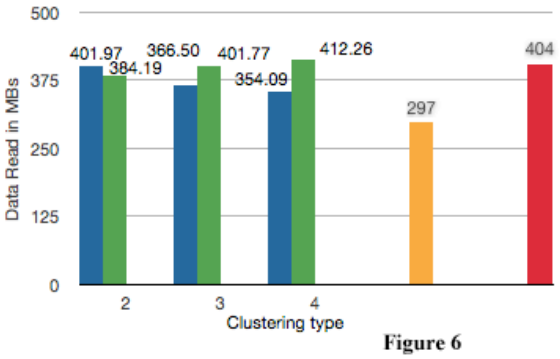

Figure 6

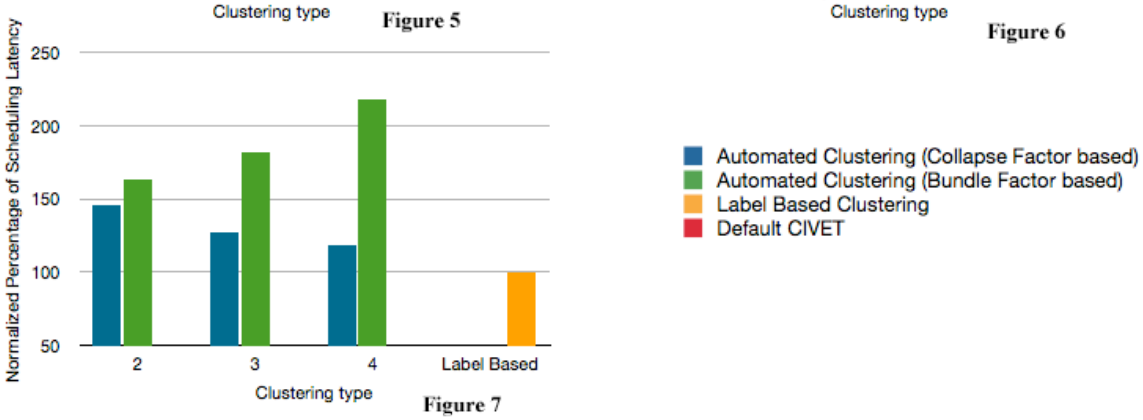

Figure 7

These data sets are only retrieved once form the central storage and are read from the cluster pool cache for subsequent jobs. Hence increasing cluster sizes produces more data efficient workflows. However, due to increased sequential execution of tasks within a cluster the turn-around time of the workflow is also increased.

The bundle factor on the other hand apparently creates computationally efficient workflows at the expense of both data efficiency and scheduling latency. A bundle factor of 2 creates at most 2 clusters per level, while a bundle factor of 4, defines the clustering granularity of 4 per level. Increasing the bundle factor, creates more and more clusters per level, increasing the parallelism in a workflow and potential for distribution. Compute intensive tasks occurring at the same level such as surface registration and gray matter extraction are grouped in separate clusters and executed in parallel. However, because these clusters are executed in parallel, the input data required by both clusters is retrieved independently by each cluster, increasing the amount of data retrieved during the execution of the workflow. As the workflow granularity decreases with increasing values of the bundle factor, more and more jobs are scheduled to the Grid, which affects the cumulative scheduling latency of the workflow.

It is apparent that in order to deploy automated clustering a trade-off between compute efficiency and data efficiency must be considered. If data efficiency is of more value then a workflow must be executed with larger clusters, hence an increased collapse factor value. However, when a better workflow turn-around time is required, a large number of clusters are more suitable.

6.2 User defined Workflow Clustering

Current planners support only automated horizontal clustering. In order to explore planning techniques which support both horizontal and vertical clusters, label based clustering in Pegasus was used to set up 11 clusters in the workflow. Most clusters group fine-grained tasks together as well as tasks which have a high number of data interdependencies. Fine grained tasks were grouped with compute intensive tasks in order to eliminate scheduling latencies and improve parallelism. The label based workflow is compared with the standard CIVET workflow.

As shown in figure 5, the turn-around time of standard CIVET workflow was 204 ±2 minutes of execution, while the turn-around time for the manually planned workflow was 140 ± 6 minutes of execution. In terms of raw turn-around times the CIVET workflow that has been planned using user defined clustering is more efficient on the same computing infrastructure. Figure 8 depicts the number of jobs that were being executed over time. The manually planned

workflow was able to execute more jobs in a shorter time span when compared to the unplanned workflow. However the workflow turn-around time is similar to the turn-around time of the workflow using automated horizontal clustering at a bundle factor of 2. In the planned workflow, an efficient use of the resources has enabled the planner to quickly execute a workflow. This allows us to process more imaging data in a relatively shorter period of time. Such methods therefore may play a key role in facilitating the e-science community during large-scale data analyses.

Figure 6 shows the amount of data retrieved from the shared NFS server. A shared NFS storage was used to store input data sets as well as intermediate data generated during the execution of the workflow. The standard CIVET Workflow read 404 MB of data during the execution life-time. The clustered workflow on the other hand read 297 MB of data from the central storage, while other required data was read from local cacheFS cache in the condor hosts. The most data intensive cluster was the cluster which grouped quality assurance tasks. The quality assurance tasks consume a great amount of data and perform fine-grained computations on them. For example, the task create_verify_image, the primary verification task in the CIVET pipeline, consumes about 50MB data, and outputs a single PNG graphic file which concatenates all intermediary processed brain scans in the workflow. Researchers use the final output to determine the correctness of the workflow output. Due to the clustering of tasks with high data interdependencies the total amount of data read from a shared storage was 107MB less for a single image.

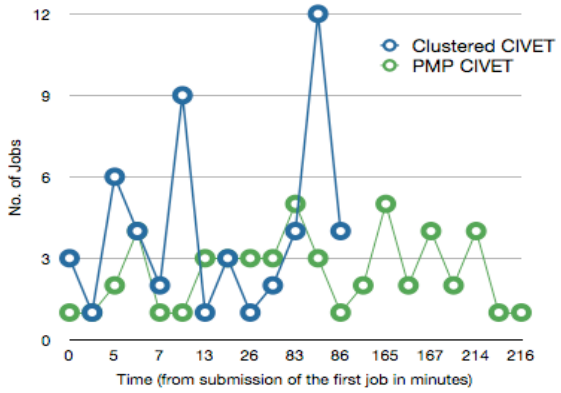

Figure 8: Number of tasks executing over time

In terms of computation, the turn-around time of the clustered workflow was more than 1 hour less than the standard PMP based workflow. In the neuGRID infrastructure the CIVET pipeline processes hundreds of images simultaneously, and this can lead to huge savings in terms of data transfers and turn-around times. The infrastructure deployed for these experiments was purpose built, and dedicated for the use of these experiments. Mean queue waiting time for a workflow was 19 seconds, which is not representative of queue waiting times in Production Grids. Production Grids such as EGEE, according to a recent study [21] have a scheduling latency of 121 seconds, with the median of 119 seconds, while some jobs may take up to 45 minutes of scheduling latency. Figure 7 shows the normalised results of the scheduling latencies compared to all clustering approaches. The user defined clustered workflow was efficient both in terms of data efficiency and compute efficiency. Automated clustering did produce a similarly compute efficient workflow, however the same workflow was less data efficient.

Creating a user defined concrete workflow plan requires a rigorous effort. Each task in a workflow needs to be studied in order to measure the compute and data footprint of a task. After such a study users author a workflow and label each task as part of an appropriate cluster. Various iterations of this process are then required. The new cluster instances must be studied and possible strands for parallel data processing identified and may be split into another cluster instance. Some clusters may be merged in order to optimise scheduling latencies and enhance data reuse. This process is impractical for large workflows or for workflows which consist of dynamic tasks which change their behaviour in response to user input or data sets. The automated workflow planning techniques, however, result in workflows that are computationally efficient or data efficient but not both. Hence a lot of challenges still remain in workflow planning. Scalable planning of large workflows, planning of workflows for enhanced data efficiency and enhanced decision making at the planner level to merge user, performance and application requirements still remain open research issues [22].

7.Conclusions and Future Work

This paper investigated the use of grid-aware planning techniques for neuroimaging analysis in the neuGRID project. The neuGRID project is an effort that builds on existing research on neuroimaging-based Grid infrastructures. It aims at providing an infrastructure that is designed to support and enhance research, which is necessary for the analysis of neuro-degenerative diseases. The paper discussed the Pipeline Service and its role in enabling parallelization, distribution and gridification of neuro-imaging pipelines. An analysis of the CIVET workflow, a compute intensive neuroimaging pipeline to measure

cortical thickness, was presented. It has been discovered that significant performance gains can be achieved if jobs are clustered together in cases where a number of small granularity jobs are destined for the same computational resource. The use of workflow planning techniques such as data-reuse, data-aware site selection, resource and data-aware parallelization, job grouping and clustering may led to huge savings in terms of turn-around times and data transfers of workflows.

In the future, we aim to explore the use of dynamic and intelligent planners for neuroimaging analysis and other related domains. As can be determined form the results, existing automated workflow planning approaches offer a trade-off between compute and data efficiency. Achieving both is a multi-dimensional search problem as a number of parameters are involved that need traversing across a number of possible optimisation techniques. This may lead to more intelligent workflow planning approaches envisioned by researchers in the domain [22].

8. References

[1] C. R. Jack Jr., M. A. Bernstein, N. Fox et al., "The Alzheimer's disease neuroimaging initiative (ADNI): MRI methods," Journal of Magnetic Resonance Imaging, Jan 1, 2008.

[2] "NIH MRI Study of Normal Brain Development " 23/7/2009; http://www.bic.mni.mcgill.ca/nihpd/info/.

[3] M. Smith, M. Jenkinson, M. W. Woolrich et al., "Advances in functional and structural MR image analysis and implementation as FSL," NeuroImage, vol. 23, no. S1, pp. 11, 2004.

[4] P. Neelin, D. MacDonald, D. L. Collins et al., " The MINC file format: From bytes to brains,", Fourth International Conference on Functional Mapping of the Human Brain, Montreal, 1998.

[5] M. J. Pan, D. Rex, and A. W. Toga, "The LONI Pipeline Processing Environment: Improvements for Neuroimaging Analysis Research," 11th Annual Meeting of the Organization for Human Brain Mapping, Jan 1, 2005.

[6] "Poor Mans Pipeline," 23/7/2009, 2009; http://wiki.bic.mni.mcgill.ca/index.php/PoorMansPipeline.

[7] A. Anjum, P. Bloodsworth, I. Habib et al., Reusable Services from the neuGRID Project for Grid-based Health Applications: HealthGrid Research Innovation and Business Case, IOS Press, 2009. pages 283-288.

[8] G. Singh, M. Su, K. Vahi et al., "Workflow task clustering for best effort systems with Pegasus," MG '08: Proceedings of the 15th ACM Mardi Gras conference, Jan 1, 2008.

[9] I. Raicu, Y. Zhao, I. T. Foster et al., "Accelerating large-scale data exploration through data diffusion," First international workshop on Data-aware distributed computing, Dec 31, 2008.

[10] T. Glatard, J. Montagnat, D. Lingrand et al., "Flexible and efficient workflow deployment of data-intensive applications on grids with MOTEUR," International Journal of High Performance Computing Applications , vol. 22, no. 3, pp. 347-360, Jan 1, 2008.

[11] J. Montagnat, A. Gaignard, D. Lingrand et al., "NeuroLOG: a community-driven middleware design," Stud Health Technol Inform, vol. 138, pp. 49-58, 2008.

[12] S. Small, M. Wilde, S. Kenny et al., "Database-managed Grid-enabled analysis of neuroimaging data: The CNARI framework," International Journal of Psychophysiology, vol. 73, no. 1, pp. 62-72, 2009.

[13] S. Kolbe, T. Ma, W. Liu et al., "A platform for distributed analysis of neuroimaging data on global grids," e-Science and Grid Computing, Jan 1, 2005.

[14] T. Oinn, M. Greenwood, M. Addis et al., "Taverna: Lessons in creating a workflow environment for the life sciences," Concurrency and Computation: Practice and Experience, vol. 18, no. 10, pp. 1067-1100, Jan 1, 2006.

[15] Y. Zhao, I. Raicu, I. Foster et al., "Realizing Fast, Scalable and Reliable Scientific Computations in Grid Environments," Grid Computing Research Progress, Jan 1, 2008.

[16] I. Altintas, C. Berkley, E. Jaeger et al., "Kepler: an extensible system for design and execution of scientific workflows," Scientific and Statistical Database Management, Jan 1, 2004.

[17] T. Goodale, S. Jha, H. Kaiser et al., "SAGA: A Simple API for Grid Applications. High-level application programming on the Grid," Computational Methods in Science and Technology, Jan 1, 2006.

[18] "Sun Grid Engine," 23/7/2009, 2009; http://gridengine.sunsource.net/.

[19] R. Prodan, and T. Fahringer, "Overhead analysis of scientific workflows in grid environments," IEEE Transactions on Parallel and Distributed Systems, Jan 1, 2008.

[20] G. von Laszewski, I. Foster, J. Gawor et al., "A Java commodity grid kit," Concurrency and Computation Practice and Experience, Jan 1, 2001.

[21] T. Glatard, J. Montagnat, and X. Pennec, "A Probabilistic Model to Analyse Workflow Performance on Production Grids," Cluster Computing and the Grid, Jan 1, 2008.

[22] Y. Gil, E. Deelman, M. Ellisman et al., "Examining the Challenges of Scientific Workflows," COMPUTER, vol. 40, no. 12, pp. 24-35, Jan 1, 2007.