HIGH ANGULAR FIBER TRACKING ON THE GRID

Jasper van Leeuwen and Anca Bucur
Philips Research, High Tech Campus 37, 5656 AE, Eindhoven, The Netherlands

Keywords: High performance computing, GRID computing, Medical imaging, Brain imaging, Decision support.

Abstract: The grid carries the promise of seamless access to vast amounts of various resources (e.g. computational, data, special instruments, human, etc.) in a reliable and secure manner and at low costs, at any time and from anywhere. In this context, an increasing body of research focuses on enabling complex healthcare applications to make efficient use of grid technologies and remote resources in clusters and grids, while satisfying the high requirements in the healthcare domain with respect to performance, reliability, cost-effectiveness, privacy and security. Our research aims at developing an architecture that enables computationally intensive healthcare applications to transparently use powerful remote resources for significant performance improvement, while being used in actual clinical environments. In this paper we apply our grid-enabled architecture to High Angular Fiber Tracking (HAFT), an imaging application with very high computational requirements. We parallelize the HAFT application and evaluate its performance in terms of response time and scalability. Finally, we deploy the application at the Amsterdam Medical Centre, in The Netherlands, and validate it (and the underlying architecture) together with clinical users and on real patient data.

1 INTRODUCTION

The grid offers the possibility of transparently accessing resources, irrespective of the location of those resources. It has the potential to allow easy access to large amounts of resources without needing to own powerful computers. This lowers the financial barrier typically associated with the execution of computational intensive applications, enabling a large number of users from various domains (such as high energy physics, bio-informatics, etc) to exploit them. Typically the resources are not under centralized control and not (fully) owned, and mechanisms (based on open standards) are provided to integrate and coordinate those resources. This provides the opportunity of accessing powerful resources, possibly not available in the residing administrative domain. Beside extending the capabilities of current applications, this infrastructure also enables the development of novel applications previously not possible due to the lack of powerful resources on which these applications could be deployed.

In our current research, we focus on applications in the medical domain and on sharing of computational resources. We propose a service-oriented architecture for computationally intensive medical applications, enabling the exploitation of resources possibly located outside the administrative domain (e.g. typically the hospital). Several medical applications have been examined and typical decomposition patterns have been described in (Bucur et al., 2005). For the first decomposition pattern — computational decomposition — *white matter fiber tracking* was selected as a first use case in (Bucur et al., 2006). *High angular fiber tracking* (described in detail in (Hoogenraad et al., 2005)), which is the topic of this paper, has been selected as use case for the second decomposition pattern — domain decomposition —, but this application is suitable for computational and functional decomposition as well.

High angular fiber tracking (HAFT) uses a novel approach in order to track fibers, which achieves increased accuracy compared to the standard, single-tensor fiber tracking used as our first case study. The most computationally intensive part of HAFT is not the actual tracking of the fibers, but the precalculation of the data—determining the double fiber content for each voxel in the domain. The sequential version cannot be applied in a clinical setting due to the very large computation time.

In this paper, our Grid Architecture for Medical

van Leeuwen J. and Bucur A. (2009).
HIGH ANGULAR FIBER TRACKING ON THE GRID.
In *Proceedings of the International Conference on Health Informatics*, pages 168-174
DOI: 10.5220/0001512901680174

Applications (*GAMA*) is applied to high angular fiber tracking, employing domain decomposition for the parallelization of the application. Our results show that the application is very suitable for parallelization and execution on a grid, bringing the application within reach for clinical usage. In fact, the part of the application parallelized in this work scales linearly with the number of nodes.

We have deployed the grid-based HAFT application in an actual clinical setting at the Amsterdam Medical Centre, where its accuracy, performance and usability were evaluated on real patient studies.

2 GAMA OVERVIEW

In this section, we describe the context of the applications and of the domain targeted by the GAMA architecture. After introducing the main needs we want to address, we describe the devised architecture and motivate our choices.

2.1 Context

The arena for our work is the medical domain. Applications in this domain become increasingly complex, operating with high resolution images, large amounts of heterogeneous distributed data (e.g. clinical, images, genomic, etc.) and making use of significant computational power, while maintaining their high requirements with respect to interactivity, low response times, reliability, privacy and security, but also low costs. While parallel computing becomes a must for many such applications, the inherently distributed nature of the data and the need to maintain low costs motivate an increasing body of research in this area to focus on enabling applications to make use of grid technologies and resources. In (Breton et al., 2004) the need for research addressing the deployment of grid nodes in healthcare organizations and the connection of healthcare professionals to the grid in order to allow the deployment of grid solutions in real settings has been identified. We circumvent this issue with an innovative approach. With our architecture the applications are able to make use of external grid resources in a seamless way for the clinical user and without the need to deploy grid nodes in the hospital domain. Through a thin interface, the applications can securely connect to a (remote) service (described in detail in the next section), which transparently initiates the execution of the computational part of the application on available grid nodes or computer clusters and returns the results to the user.

While other research, such as (Frate et al., 2006), focuses on the use of the grid for enabling access to large amounts of distributed data, our work targets computationally intensive applications that can be efficiently parallelized. The GAMA architecture enables medical applications to use external, powerful resources transparently for the clinical user, and it is scalable for increasing problem sizes and number of users. The clinical user does not need to have any grid-related knowledge to use the applications built according to this architecture and does not need to be aware of the use of remote resources.

In (Bucur et al., 2005), a number of medical applications are analysed which currently cannot be used in a clinical setting due to the high computational demands. Three distinct decomposition patterns are suggested by which the applications can be efficiently parallelized through decomposition: functional, computational and domain decomposition.

In a functional decomposition, the system is divided into functional components. A performance improvement can be obtained when the execution architecture of the system allows pipelining of the components or when components can exploit previously not available resources. Computational and domain decomposition focuses on decomposing a functional component. In a domain decomposition, the data domain (denoted by A) is decomposed into (usually disjoint) parts (say n parts, named a_i with $0 \leq i < n$). Performance improvements can be obtained when ($F(A) = \oplus_{a_i \in A} F(a_i)$), as the $F(a_i)$ components can be executed in parallel. In a computational decomposition, the computation domain (C) is decomposed into parts (say n parts, named c_i with $0 \leq i < n$) and performance improvements can be obtained when $C(A) = \oplus_{c_i \in C} c_i(A)$, as the $c_i(A)$ parts can be executed in parallel. A real-world application can display a structure suitable for one or a combination of the three strategies. The benefit gained with the parallelization is a trade-off between the execution time achieved by parallel execution versus the overhead in communication.

2.2 Architecture

As previously mentioned, the GAMA architecture targets the medical domain and has the aim to enable a wide range of medical applications to access (remote) grid resources. The three decomposition patterns introduced in the previous section are simultaneously supported to ensure the facilitation of a wide range of medical applications. At the same time, the use of grid resources needs to be minimally invasive to the (clinical) user and to the associated workflow. This is achieved by maintaining a (thin) user

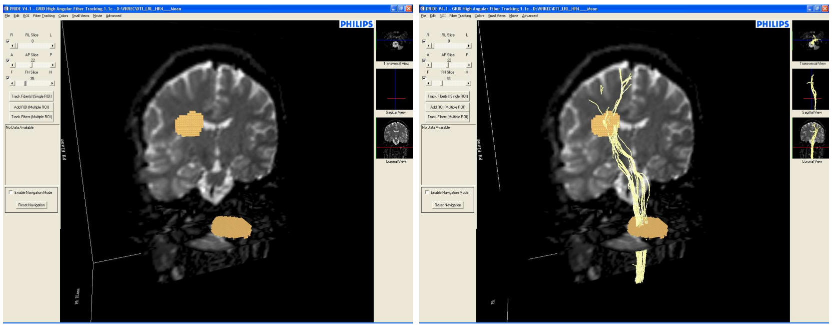

Figure 1: Visualization of HAFT output. Left: the scan with the selected ROIs, right: the resulting fibers.

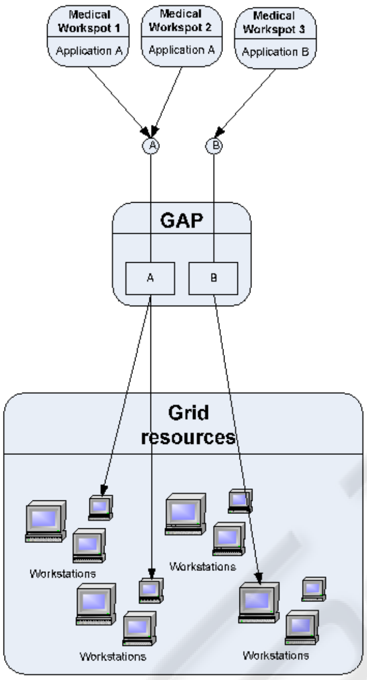

Figure 2: Overview of the GAMA architecture.

interface at the hospital side. The application components which form the bottleneck, preventing the system from reaching the required performance, are moved away from the hospital side to the grid-side, and modified in order to enable good exploitation of the available resources (see Figure 2).

In order to access grid resources from within a hospital, a system component called Grid Access Point (*GAP*) is introduced. This component acts as a bridge to the grid resources and offers application/computational services to workstations located in the hospital. In this way, the GAP also encapsulates the grid mechanisms (e.g. job-submission) of the specific grid implementation to which it is connected. This allows for an easy switch between different grid implementations. In the medical domain, there is also the need for interactivity. The GAP enables interactivity by relaying data streams between the hospital side and the grid side.

Besides providing benefits from an application (computational) point of view, the architecture also provides benefits from a business point of view. Instead of offering applications, services can be offered via the GAP. This allows for instance for a pay-per-use business model (e.g. compared to a pay-per-application model). In addition, different service implementations can offer the same application service, allowing for trade-offs between (for example) cost, performance, accuracy.

2.3 The Grid Access Point

The GAP offers multiple application services to multiple users and hides grid technology details from the (clinical) user. Via a plug-in mechanism, application service implementations can be provided to the GAP, that can run on the resources available to the GAP. Interactivity is enabled by relaying data streams between the hospital side and the grid side. The security of the datastreams between the hospital side and the grid side is ensured by the use of ssh tunnels[1].

The GAP can be seen as a single logical component, but it is designed as a distributed component in order to prevent it from becoming a bottleneck. Multiple GAPs can form a network and they can monitor each other's load.When a service is requested, an overloaded GAP can refer a user to a less loaded GAP. An alternative is to precede the GAP service by a GAP-discovery service that would transparently redirect the user request to the least loaded GAP. In the current implementation the load is based on the available network bandwidth for the GAP, since the bandwidth is the first bottleneck that would be reached by a GAP component. As the GAP hides grid technology from the user, it can be used to exploit resources available in multiple (disjoint) grids. This also includes

[1]http://www.openssh.com/

the possibility of exploiting a locally available cluster, providing a smooth transition path from locally owned resources to shared resources. In the current setup, globus[2] is used as grid fabric.

3 HIGH ANGULAR FIBERTRACKING

3.0.1 Fibertracking Introduction

Fiber tracking is an indirect medical imaging technique (see (Beaulieu, 2002; Mori and Zijl, 2002)). It uses diffusion-tensor magnetic resonance imaging and has as goal to reconstruct axonal tracts in the central nervous system that connect brain structures. Examples of the usage of fiber tracking are in the area of studying brain development, multiple scleroses, stroke and schizophrenia.

Fiber tracking relies on the observation that water molecules exhibit *Brownian motion*, i.e. random movement of molecules in a fluid or gas. The freedom of moving can however be restricted by, for instance, the underlying cellular microstructure of tissue. When molecules are in a pure liquid, the diffusion is the same in all directions (so-called *isotropic diffusion*). This is not the case with anisotropic diffusion (i.e. when barriers such as communication tracts connecting different brain areas exist). Diffusion Tensor Imaging (DTI) shows the extent and orientation of diffusion anisotropy or the averaged diffusion properties of the water molecules in a volume (typically a voxel with dimensions in the order of 1 to 5 mm). The amount of anisotropy is directly related to the cellular microstructure. This resolution is high enough to distinguish white matter tracts in the brain. A tract (called *fiber*) consists of a collection of axons (the extension of a nerve cell used to propagate information).

There are multiple ways to track the fibers. The first category of algorithms is based on line propagation. DTI provides a 3D tensor field, which is used to propagate a line from a seeding point. The second category is based on global energy minimization to find the least costly path between two points (energywise). The algorithm used in our application is a line propagation algorithm.

A common approach for tracking fibers is to create one or more regions of interest in the dataset in order to select fibers of interest. The purpose is to select those fibers that will go through all the regions of interest. Commonly, the voxels in a region of interest are used as seeding points for the line propagation algorithm. The extent of the anisotropy and the angle change are used as termination criteria. The line propagation algorithm is terminated when the angle change becomes too big, as the diffusion process is assumed to be Gaussian, or when the anisotropy becomes too low (as fibers are assumed to end).

The standard single-tensor algorithm assumes that there is clearly a principal axis in the voxel when fibers are present. That algorithm fails however when fibers branch or cross in a voxel (e.g. two large eigenvalues, which show two predominant directions of propagation). This can be countered by using every voxel as a seeding point for tracking and afterwards selecting only those fibers that intersect the regions of interest. This is however a time consuming approach and in (Bucur et al., 2006) we show a successful performance improvement by gridifying the algorithm and performing a computational decomposition.

In this paper fiber branching and fiber crossing are handled in a different way. Diffusion weighted data sets with many gradient encoding directions are used as input and instead of a single fiber model, a double fiber model is used. Each voxel can be described by a double fiber content, in terms of directions and volume fraction. In order to get the double fiber content, a double fiber model is fitted with the apparent diffusion coefficients for every voxel of the dataset (Hoogenraad et al., 2005). With the obtained double fiber content per voxel, branching and crossing fibers can be handled successfully during tracking. This calculation takes place when the data is loaded into the application. As the solution space for the fitting process is big, it is time consuming to determine the double fiber content. Therefore, this is the part of the application that we have chosen for efficient parallelization and grid execution.

3.1 Application Design

In the workflow of HAFT, first a dataset (a DTI scan) is selected and the double fiber content is computed for the complete dataset (so-called *dataloading*). Next, the user can interactively select regions of interest in the application. The fibers crossing all the regions of interest will be subsequently tracked.

Figure 1 shows the visual output of the application with a DTI scan loaded. Regions of interest are selected on the left side and the right side shows the output of the application with the resulting fibers, after the tracking.

The most computationally challenging part of HAFT is the calculation of the double fiber content. This calculation has to be performed for each voxel of the DTI scan. As there are no (computational) depen-

[2] http://www.globus.org/

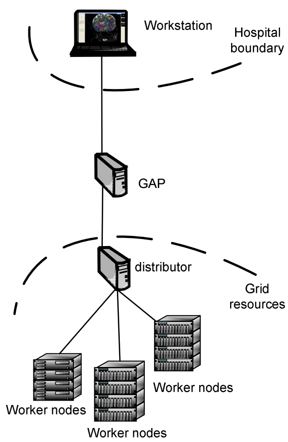

Figure 3: Deployment scenario of the grid-enabled HAFT application.

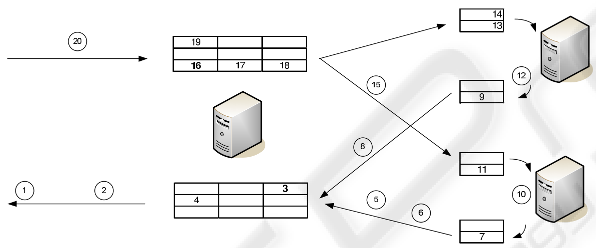

Figure 4: Topology of the grid-enabled HAFT application.

dencies between the voxels, a domain decomposition can be performed. This means that the entire dataset is split and distributed over the available worker nodes and each worker node will calculate the double fiber content for the part of the dataset assigned to it.

As grid resources may be used to deploy the application, it should be taken into account that the available resources can be heterogeneous. This implies that the calculation speed of the resources can differ. Therefore, we implement a dynamic load balancing approach where a distributor dynamically distributes the load over the available workers. The implementation uses MPI[3] as communication library. In our implementation (see Figure 3), the hospital-side workstation will still host a light-weight client and the visual interface of the original sequential application. The computational part of the application is extracted and its most computationally intensive component, which is in this case the *dataloading*), is parallelized with a distributor-workers approach. In the first stage of dataloading, the settings are sent to all the worker nodes. The client part of the application running on the workstation sends this data via the GAP to the distributor, which broadcasts it to all the worker nodes.

In the second stage, the dataset (the apparent diffusion coefficients) is streamed from the client to the distributor, via the GAP. The distributor forms on-the-fly work packages (consisting of the content of a specified number of voxels) out of the stream and sends the work packages to the workers. The workers calculate the results and send them back to the distributor. The worker nodes use asynchronous communication and buffering in order to hide as much (data transfer) latency as possible. The distributor sends a new workpackage to a worker as soon as it receives a result package from that worker. This approach (depicted in Figure 4) ensures that the workers are kept busy as long as there is still work to be distributed, even when the computational power of the worker nodes differs. The distributor receives the results in a buffer, indexed to preserve the order of the result items. The results may arrive out of order, as the workers might not work equally fast, and the size of the buffer dictates how out-of-sync workers can be. When the distributor receives a set of results, these are ordered and streamed back via the GAP to the client application.

[3]http://www-unix.mcs.anl.gov/mpi/mpich2/

4 EVALUATION

4.1 Scalability

In this section, the scalability of the algorithm is assessed. Experiments were performed on a cluster of computers. The cluster consisted of 35 nodes, having 4 3.0 GHz Intel Xeon CPU's processors each.

Nr of nodes	resource reservation time (s)	execution time (s)
2	77	22321
4	104	7675
8	81	3817
16	121	1720
32	128	857
64	137	439
96	159	304
128	188	246
140	175	227

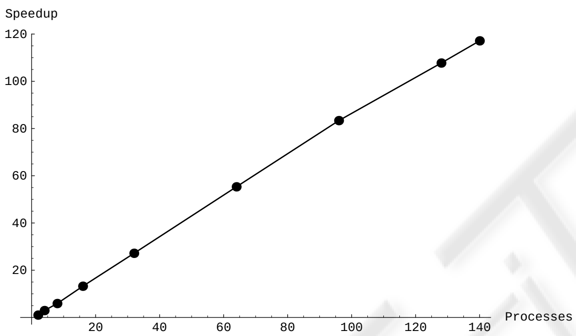

Figure 5: Measured response times and speedup of the HAFT application for 1 to 140 processing nodes

As can be seen in Figure 5, the speedup with respect to the number of nodes is very linear. Starting at a calculation time of 6 hours and 12 minutes, the execution time is reduced to 3 minutes and 47 seconds, bringing the application within clinical reach. Significantly larger scale-up is not desirable due to the incurred cost of adding nodes and little (noticeable) performance gain (as the steps with constant execution time and not parallelized, such as visualization, increasingly dominate the total execution time).

It is expected that eventually the distributor will become the bottleneck and will cause the speedup to flatten, as the distributor cannot provide the workers with enough work while gathering the results. The computation/communication ratio of the data loading phase is high and the latency is hidden (by using buffering and asynchronous communication), which yields that the flattening of the speedup will only happen for a very large number of nodes. At that point, the execution time gained by using more nodes will be spent in extra communication between distributor and worker nodes, and the number or sizes of work packages will be too small relatively to the number of worker nodes.

Table 5 shows the number of nodes versus the resource reservation time (sec.) and execution time (sec.). In the setup, a first-come first-served was used with respect to scheduling of jobs on the available resources, explaining the variations in the resource reservation times.

4.2 Requirements in the Healthcare Domain

The domain that triggered the development of grid middleware was high energy physics. Of course, also other domains saw the promises. Different domains may however bring in different requirements. Currently deployed grid middleware is very focused on long running jobs, with no interactivity or quick response time requirements, scheduled in a batch manner. The main goal of the administrators of the computational resources is to keep the load of the resources as high as possible, in order to maximize (depending on the ownership of the resources) the cost-effectiveness or the profit.

The medical domain brings in a whole range of applications with very different requirements. The requirement that is seriously hampering the effectiveness of applying currently available grid technology to the medical domain is predictability. Applications need to fit into the clinical workflow of the user. For interactive applications, this implies that the timespan between a request for execution and the actual execution and returning of the results is both short and predictable. For non-interactive (and mostly batch-oriented) applications, this implies that the time by which the computation will be finished is predictable (and short enough to be clinically relevant). Clinical applications may also have hard-time requirements and it is not always the case that the need for resources can be identified well in advance, which means that introducing reservations may not always solve the problem. Critical applications need to be served first - so the support of priorities may be needed, and the alternative to extend the number of available resources may need to be supported as well, as the load of the system is not even and it may unpredictably peak. Currently, this matter is resolved either by having dedicated resources available or by over-dimensioning the available resources. Clearly, this is not a cost-effective approach. Once Quality of Service guarantees (e.g. on scheduling times) become commonly available, application service providers can provide service level agreements suitable for the medical domain while having cost benefits compared to dedicated solutions.

5 CONCLUSIONS AND FUTURE WORK

In this paper we have proposed an architecture enabling computationally intensive applications to transparently access remote powerful resources. Our case study was High Angular Fiber Tracking, an application relevant in the healthcare domain, with very high resource requirements, too high for its deployment and its use in a real healthcare environment. Our parallelized version of the application, compliant with the GAMA architecture, shows linear speedup, yielding excellent performance during its execution on a remote cluster. The application was brought this way within the reach of the clinical practice, and was deployed and validated in a clinical setting.

This work shows the potential for using grid technology in the medical domain. It has been demonstrated that grid resources can be accessed from a clinical environment transparently for the (clinical) user. The benefits are clear, cost-effective access to a large amount of shared resources. Technically, the main bottleneck for accessing grid resources from within hospital boundaries is the lack of quality of service guarantees.

Several research issues are still to be addressed. As future work we foresee the use of the third decomposition pattern - the functional decomposition, resulting in an example in each of the decomposition patterns identified. On the GAP side, several issues are to be explored, such as incorporating quality-of-service guarantees. Another hot topic in the medical domain is privacy. The responsibility of the GAP plugins could be extended with ensuring that the data sent is anonymous, and to anonymise the data prior to sending it to the grid resources if this is not the case.

ACKNOWLEDGEMENTS

Part of this work was carried out in the context of the Virtual Laboratory for e-Science project[4]. This project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC and W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ). Philips Medical Systems provided the initial data sets used in our experiments and the sequential High angular Fiber-tracking application that constitutes the basis for our grid-enabled case study application. The validation of our HAFT application was carried out at the Amsterdam Medical Centre, The Netherlands. AMC Research has performed a clinical evaluation of the grid-enabled HAFT application.

REFERENCES

A.I.D. Bucur, R. Kootstra and R. Belleman. *A Grid Architecture for Medical Applications*. Healthgrid Conference, 2005.

A.I.D. Bucur, R. Kootstra, J. van Leeuwen and H. Obbink. *A Service-oriented Architecture for Grid-enabling Medical Applications*. Healthgrid Conference, 2006.

F. Hoogenraad, R. Holthuizen, R. Brijder. *High angular resolution diffusion weighted MRI*. International Patent, WO2005076030, 2005.

V. Breton, A.E. Solomonides and R.H. McClatchey. *A perspective on the Healthgrid initiative*. IEEE Int. Symp. on Cluster Computing and the Grid, 434-439, 2004.

C. Del Frate, J. Galvez, T. Hauer, D. Manset, R. McClatchey, M. Odeh, D. Rogulin, T. Solomonides, R. Warren. *Final Results from and Exploitation Plans for MammoGrid*. Studies in Health Technology and Informatics, 120:305-315, 2006.

C. Beaulieu. *The basis of anisotropic water diffusion in the nervous system - a technical review*. NMR Biomed,15:435-455,2002.

S. Mori and P. van Zijl. *Fiber tracling: principles and strategies - a technical review*. NMR Biomed,15:468-480,2002.

[4]http://www.vl-e.nl/