A case study in medical imaging and the grid

Cécile Germain-Reanud[1] Angel Osorio[2] Romain Texier[3]

[1]LAL
CNRS-Université Paris-Sud

[2]LIMSI
CNRS-Université Paris-Sud

[3]LRI
CNRS-Université Paris-Sud

Running head : A case study in medical imaging and the grid

Corresponding author : Cécile Germain-Renaud
LAL Bat 200
BP 34
91898 ORSAY Cedex
FRANCE
germain@lal.in2p3.fr

A case study in medical imaging and the grid

Cécile Germain-Renaud[1] — [1]LAL, CNRS-Université Paris-Sud

Angel Osorio[2] — [2]LIMSI, CNRS-Université Paris-Sud

Romain Texier[3] — [3]LRI, CNRS-Université Paris-Sud

Abstract

Medical images analysis is one of the domains where interactivity and resources at grid scale can be combined. In medical image processing, human perception and interpretation of complex structures is a key element. Visualization can then be included in a loop with the grid computing process, and this imposes a constraint of interactivity on the system. Interactive grid-enabled services dedicated to medical imaging will have to carry complex requests, either in data access and/or computing power. This paper first discusses the impact of real-time constraints on algorithms for data exploration and computing. The second part presents some preliminary experiments.

1. Introduction

Grid system target services requiring large computing, storage and network resources. One of the main goals is dealing with considerable quantities of data, produced in varied contexts. For some of these contexts, for example particles physics, extracting information from the data imply little man-machine interaction. For other contexts, the human visual system remains useful or even essential to recognize and interpret complex structures. Visualization can then be included in a loop with the grid computing process, and this imposes a constraint of interactivity on this system. To satisfy the interactivity constraint, the grid services must satisfy the requests in time limits compatible with the man-machine dialog specific to the considered application.

Medical images analysis is one of the domains where interactivity and resources at grid scale can be combined. A grid for medical images aims to gather highly specialized instruments, an interactive visualization tool, distributed storage at TeraBytes scale, or even PetaBytes, and parallel computing facilities. For the various modalities of multidimensional medical images (scanner, RMI, PET-Scan etc.), a standard workstation allows interactive navigation inside these images. On the other hand, this workstation is not sufficient for some more complex and extremely useful computations, such as volume reconstruction, multi-modal fusion (e.g. Scanner +

IRM + PET), and many other data-intensive medical tasks. This paper presents an analysis of what would be a grid-enabled framework for interactive medical applications (section 2) along with a preliminary experiment on a particular application (section 3).

2. Interactivity, medical imaging and the grid

This section discusses some new questions created by introducing interactivity applied to the medical imaging application domain with respect to existing grid middlewares. An in-depth discussion of visualization and the grid can be found in [1].

2.1. Visual interaction and medical images

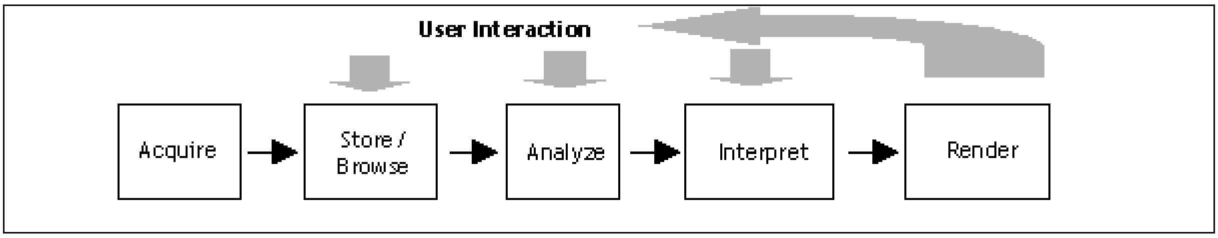


Figure1:The visualization pipeline

The steps of a visualization-based interaction are represented fig. 1. From the viewpoint of the final user (the radiologist), the three intermediary steps can be executed remotely: data storage and navigation, analyzes, and interpretation. User interaction is present in all these steps. Figure 2 shows a more practical view of the interaction between local and remote data storage and processing.

For medical images, *data acquisition* is realized by an instrument (Scanner, MRI, PET-scan) installed in hospitals, and encoded in DICOM format. The data can then be stored locally, or remotely (central or distributed database) and may as well be replicated (DataGrid [2]) or cached (DPSS/IBP [3]). For instance, in clinical practice, *selecting* the appropriate data is an highly interactive process, which uses the expertise of the radiologist, first to browse the various views included in a clinical exam, then to visually inspect the selected image. For other applications, e.g. epidemiology, this process can be automated

Analysis translates the initial data (typically 3D densities of physical quantities) to data relevant to the user request. This task can be simple, such as the planar projections used when navigating a radiological image, or very time-consuming, e.g. in semi-automatic segmentation, or multi-modal fusion.

Interprétation generates a visualizable representation of the data. In volume reconstruction, this is the most costly step.

The localization of the final step, rendering/rasterization, depends on the application and the target environment. For very heavy image, or if the target is a virtual reality center or collaborative work, this phase should be distributed. In most cases of clinical practice, a standard PC will suffice.

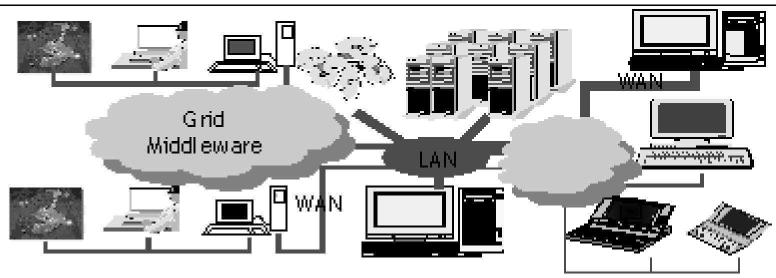


Figure 2: A Grid-enabled interactive medical imaging architecture

2.2. ***Heavyweight Interactive Services***

To comply with the interactivity constraint, the grid services must satisfy the requests in time limits compatible with the man-machine dialog specific to the considered application. For medical imaging, and more generally scientific visualization, such services carry complex requests, either in data access and/or computing power. For this reason, the term *heavyweight interactive services* (HIS) has been coined, although for a different application, in the framework of the Dv project [4].

An open question is the definition of the specificity of interactive services and their relationship with the current or emerging de facto standards (Globus, OGSA [5]), and also as with the existing middlewares (DataGrid, EuroGrid, NetSolve,…). This question is currently tackled in the CrossGrid project, with a strong emphasis on MPI parallelism and portals. As far as medical imaging is concerned, many other issues are of importance: interactive remote access to medical data and metadata (DICOM), and all questions related to security, authentication and privacy. Here, we focus on the impact of real-time constraints on algorithms for data exploration and computing.

A crucial question to deploy HIS lies in allocating adequate resources for the timely execution of requests, as well as the effective transfer of the data needed by these requests. Usually, nor the requirements of these requests neither the actual state of the Grid resources cannot be predicted well in advance. The usual approach to this issue is end-to-end QoS (e.g. the GARA architecture for Globus). In this view, the grid middleware provides monitoring and performance prediction tools; an algorithm is then able to guarantee a performance from its inputs parameters and its implementation. We think that this approach is probably not well suited to the relatively fine grain of the interactions considered here. We propose to consider dynamic adaptation at the algorithmic level. Such an approach has been experimented successfully for lossless adaptive compression [6];

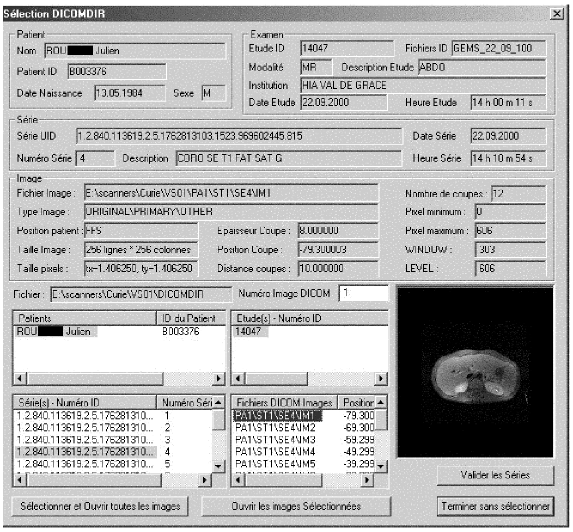
the next section provides preliminary experiments about user-level dynamic schedulers that adapt to the existing resources. A big issue here is how these schedulers can be embedded in grid middleware where the task submission of tasks is strongly mediated through batch systems.

Although leading to useful components, this rather classical adaptativity could be complemented by the more radical concept of *progressive fidelity algorithms*. For large parts of the interaction in the medical imaging applications, a non-optimal quality of information is sufficient at a given point in time. This comes from various reasons: only a cursory inspection can result in eliminating some data sets, when a DICOM examination is browsed for selecting the most relevant image; or, when an image is actually visualized, only a limited part of the actual information can actually be viewed at a given time by the radiologist (medical windowing). With the one order of magnitude increase in the data volume by image created by the new instruments (multi-resolution scanners), this phenomenon will become even more important. Thus, a key point is to focus the resource usage on the actually useful data., while the Grid system can exploit the human interaction delay in the background to refine the information finally delivered. A relevant analogy is video streaming: the combination of data access, computations and compression for the interaction with medicals image will play the role of the on the fly compression schemes for video. The specialization to medical imaging adds a supplementary constraint: the radiologist must be able to get the full information included in a medical image or examination, even if she can use only a fraction of this information at a time.

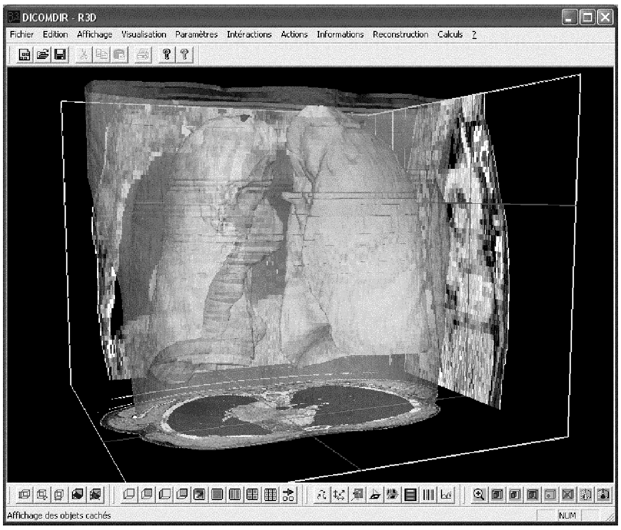
3. Remote volume reconstruction

Real-time volume reconstruction from tomographic images (Scanner, RMI, CT) remains a challenge for complex physiological objects. This section first presents the PTM3D software, an open-source fully featured DICOM images analyzer oriented towards volume reconstruction and instantaneous measurement. Next, we describe a distributed, but non yet grid-enabled architecture which achieves the delegation of these costly computations to high performance parallel computers. The goal of this work was to define the hardware and software requirements for a real-time execution of the reconstruction process. Along with many other conditions (storage, WAN network performance, security, etc.), our results contribute to the “big picture” of a grid infrastructure for medical imaging.

3.1. PTM3D



(a) DICOM interface



(b) 3D reconstruction

Figure 3: PTM3D snapshots

PTM3D is an open-source fully featured DICOM images analyzer developed at LIMSI by Angel Osorio and his team. PTM3D transfers, archives and visualizes DICOM encoded data (fig. 3 a); it provides computer-aided generation of three-dimensional representations from TDM, IRM, PET-scan, or echography 3D data (fig. 3 b);. The system currently runs on standard PC computers and it is used on line in radiology centers [7].

Besides moving independently along the usual three axes (axial, coronal, sagittal), the user is able to view the cross-section of the DICOM image along an arbitrary plane and to move it. A reconstructed volume (organ, tumor) is displayed inside the 3D view. The reconstruction also provides the volume measurement, required for therapeutic decisions. The reconstruction involves three main steps: segmentation, triangle mesh generation, and rendering. The segmentation process is semi-automatic, based on a few graphical hints from the user, and an active contour method.

In the mesh generation step, all surfaces joining two adjacent contours are generated as a set of triangles, which will be used for visualization and measurements. The triangles are defined by an appropriate correspondence between the points of the upper and lower contour, through a Djikstra-like algorithm. The resulting data are stored as OpenGL data structures.

3.2. Remote Computation Architecture

For the kind of application considered in this paper, the graphical capacities of a PC are sufficient for the rendering step, and the user interaction with the graphical display is essential; thus only the bulk of the computation must be deported on high-performance resources. The application can thus be considered as a four section program (fig. 4): the interactive section, the sequential one, the parallel computation, and the result re-composition and display. The interactive part is the user interface, which lets the user choose which structure she/he wants to see in 3D. Typical questions are, at the hardware level, the machine architecture (shared vs distributed memory) and the network requirements; at the software level, the parallel middleware (MPI vs OpenMP), and the scheduling scheme.

However, many questions must be answered at this point. For instance, if large shared memory machines were required for acceptable performance, due to the size of the data set, the remote computing center would be a high-end one; if moderate size cluster are acceptable, a more distributed scheme for computing centers is the most cost-effective solution. This has consequences on the data storage scheme, and in turn on performance, *e.g.* through replica management.

The mesh generation step dominates the cost of the reconstruction, consuming more than 95% of the total execution time. This step is thus the only one we have considered for parallelization. The mesh generation falls into the master-slave category: it can be divided in a fairly large number of independent tasks, which only need initial data; a task is the reconstruction of one slice, without inter-task communication. More complex schemes involving intra-slice parallelization should be considered only if this coarse grain one is not convenient. Linear acceleration is thus possible.

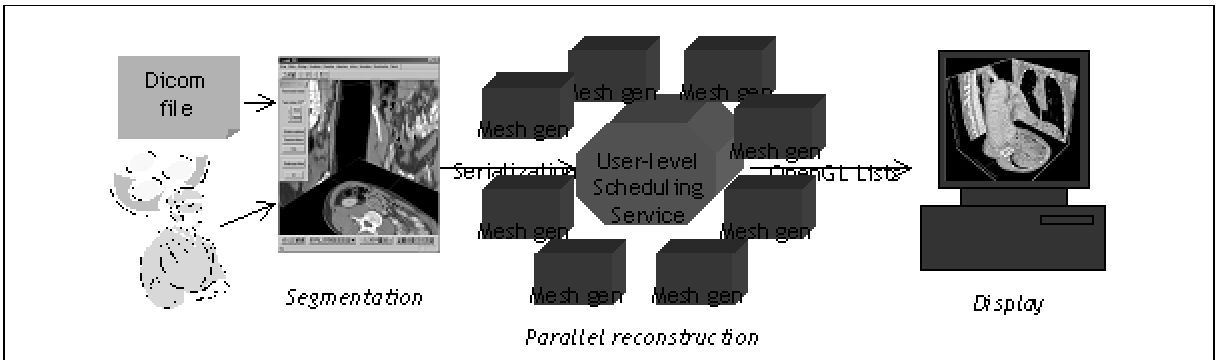


Figure 4: Remote volume reconstruction

In the master-slave scheme (as in the sweep parameter one), the key for efficient parallelization is the scheduling scheme. When it proves efficient, static scheduling should always be preferred: all the scheduling cost is paid at compile time, and the development effort is minimal. However, in our case, static scheduling is not efficient: the speedup is far from linear, ranging from 1.5 to 4 at 6 processors. Further analysis shows that the reconstruction cost widely varies across and 2) the reconstruction cost cannot be accurately predicted from the input data, precluding the use of sophisticated heuristics for static schedulers. We have thus considered self-scheduling. As shown in [8], centralized self-scheduling schemes outperform work stealing ones for independent tasks, and are far easier to implement. In centralized self-scheduling, a scheduling thread dispatches work to the slaves following an on-demand scheme: when a slave has finished its work, it asks the scheduler for a new one. A unit of work can be one task, or a subset of tasks, to spare network delay, at the expense of potential load imbalance.

The potential overheads are the communication of the initial data and result data and also the scheduler itself. Both can become bottlenecks. We have considered three self-scheduling schemes: pure, chunk and guided, which address this bottleneck in various ways. The overall result is described in the next section.

3.3. Performance results

The testbeds are 1)a cluster of biprocessors PC, based on 500MHz PIII, connected trough either a Myrinet switch and 2) the same cluster connected through a 100Mb Ethernet. The initial parallelization is based on MPI. For platform 1), we used the BIP implementation of MPICH, and for platform 2) the generic implementation. The DICOM inputs used for benchmarking range from 50MB to 100MB.

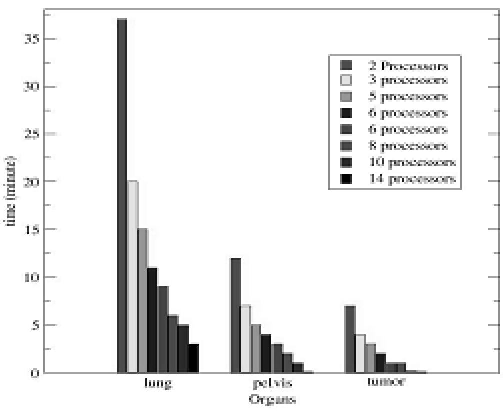


Figure 5: Parallel Speedup

It turned out that pure self-scheduling (PSS) is the most effective strategy. Figure 5 displays the reconstruction times for various experimental cases, on platform 2. The gain in load balancing obtained from this fine-grain scheduling outperforms the extra synchronization costs in the scheduler.

Another issue was the scheduling of data distribution. In the previous experiment, each request for work is answered with the corresponding data set. This cost time for each task, and this cost may be increased if the network or the scheduler become congested by simultaneous requests. Detailed execution traces showed that the network/scheduler congestion happened, but was limited: for our application, computation self-scheduling provides some network self-scheduling. Nevertheless, we have evaluated the remaining potential for gain with platform 1). With efficient broadcast hardware and software in platform 1), one can expect to limit the data distribution overhead by sending the whole data set to all processors at initialization time. The work request protocol involves only sending pointers to data, not the data themselves. Indeed, the experiments showed that the communication time decreases by 21% on average when comparing the on-demand data transmission with the initial broadcast scheme on platform 1).

The main results of our study are:

- small-scale parallelism is extremely effective for this application, with a quasi-linear speedup for large reconstructions;
- performance scales with the scanner size: the limiting factor is the size of the largest slice, not the number of slices;
- this efficiency requires a non-trivial self-scheduling scheme ; this dynamic scheme is compatible with a multi-user load.

The global result is that middle-range clusters, even equipped with 100 Mb/s Ethernet, provide the adequate level of parallel computing power.

4. Conclusion and future work

Grid-enabled solutions have been proposed for applications, such as High Energy Physics, with heavy but rather simple data processing requirements and no need for much user interaction. However, in some applications such as medical image processing, human perception and interpretation of complex structures is a key element. This paper has presented some ideas and very preliminary experiments on this subject. This is a part of the multidisciplinary project SPID-IMG, proposed by researchers in laboratories from parallel and distributed informatics, signal processing, human-computer interaction, grid computing, and medical imaging and involving hospital collaborations. This project has the ambitious goal to define a global architecture addressing the data distribution and time constraints related to interactive medical applications.

Bibliography

1 I. Foster, J. Insley, G. von Laszewski, C. Kesselman and M. Thiebaux. Distance Visualization: Data Exploration on the Grid. IEEE Computer Magazine, 32 (12):36-43, 1999.

2 The DataGrid Project. www.eu-datagrid.org

3 A. Bassi, M. Beck, J-P. Gelas. and L. Lefevre. Logistical Storage in Active Networking: a promising framework for network services . 3rd Int. Conf. on Internet Computing (IC 2002), 2002.

4 J. López and D. O'Hallaron, Evaluation of a resource selection mechanism for complex network services. 10th Symp. on High Performance Distributed Computing (HPDC-10), 2001.

5 The Globus project. www.globus.org

6 E. Jeannot, B. Knutsson and M. Bjorkmann Adaptive Online Data Compression, in IEEE High Performance Distributed Computing (HPDC'11), 2002.

7 S. Merran, A. Osorio, G. Andre and J. Couetil. PC-based software for organs volume measurements in living donors: Applications in organs transplantation. In RSN 2000, InfoRAD 9109DS-I

8 S. Flynn Hummel, J. Schmidt, R. N. Uma et J. Wein. Load-Sharing in Heterogeneous Systems via Weighted Factoring. In 9th Ann. Symp. on Algorithms and Architcture (SPAA), 1997