



HAL
open science

Using grid technologies to face medical image analysis challenges

Johan Montagnat, Vincent Breton, Isabelle Magnin

► **To cite this version:**

Johan Montagnat, Vincent Breton, Isabelle Magnin. Using grid technologies to face medical image analysis challenges. Biogrid'03 workshop, May 2003, Tokyo, Japan. pp.588-593. hal-00691658

HAL Id: hal-00691658

<https://hal.science/hal-00691658v1>

Submitted on 26 Apr 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using grid technologies to face medical image analysis challenges*

J. Montagnat¹, V. Breton², and I.E. Magnin¹

¹ CREATIS, CNRS UMR 5515, INSA, 69621 Villeurbanne Cedex, France, <http://www.creatis.insa-lyon.fr/>

² LPC, CNRS-IN2P3, Campus des Cézeaux, 63177 Aubière Cedex, France, <http://clrwww.in2p3.fr/>

Abstract

The availability of digital imagers inside hospitals and their ever growing inspection capabilities have established digital medical images as a key component of many pathologies diagnosis, follow-up and treatment. To face the growing image analysis requirements, automated medical image processing algorithms have been developed over the two past decades. In parallel, medical image databases have been set up in health centers. Some attempts have been made to cross data coming from different origins for studies involving large databases. Grid technologies appear to be a promising tool to face the raising challenges of computational medicine. They offer wide area access to distributed databases in a secured environment and they bring the computational power needed to complete some large scale statistical studies involving image processing. In this paper, we review grid-related requirements of medical application that we illustrate through two real examples.

1 Context

Digital medical images [1] represent a tremendous amount of data. In industrialized countries, an hospital produces several Terabytes of medical image data each year, bringing the total production of the European Union or the USA for instance to thousands of Terabytes (Petabytes) a year. These data need to be properly archived for both medical and legal reasons. Beyond the outstanding issue of proper storage and long term archiving of such an amount of data, automated analysis is increasingly needed as manual inspection of medical images is a complex task, and may become extremely tedious and error prone when dealing with 3D or time sequence of 3D images. Moreover,

*This work is partly supported by the IST European Data-Grid project [6] and the French ministry for research ACI-GRID project [10]

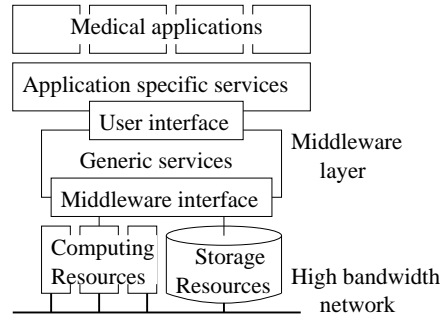


Figure 1. From resources to applications, layered architecture

many medical applications today benefit from computerized models and image processing techniques for which computer image analysis is mandatory. Computers have thus been introduced in hospital for diagnosis and pathologies follow-up (modeling and quantitative analysis of images), simulation (*eg* planning difficult interventions), and in real time treatments (*eg* computer assisted surgery).

Grid technologies have been developed for applications with large storage and computation requirements in mind. They offer a promising tool to deal with current challenges in many medical domains involving complex anatomical and physiological modeling of organs from images or large image databases assembling and analysis. However, grid technologies are still in their youth and the often propose only very generic services for deploying large scale applications such as users authorization, authentication and accounting, data replication and fast transfer, and transparent access to computing resources. These basic services are needed for any application domain but specialized higher level layers should be developed to take into account the application specific requirements (see figure 1).

The bottom of figure 1 shows the hardware: com-

puting nodes, storage nodes and fast network connections. The low level middleware services provide homogeneous access to the underlying heterogeneous components. The upper level middleware provides a user interface for deploying grid-enabled applications with data sharing and distributed computation capabilities. The application specific layer should provide high level and domain specific services such as data distribution taking into account the data semantics, access to heterogeneous database formats, high level computation services, support for parallel and interactive applications, etc.

Moreover, grid technologies offer a unique environment for sharing data and computing or storage resources. They will help federating research efforts over distant institutes by allowing to assemble very large datasets needed in statistical and epidemiological studies and by allowing to share computing resources between small laboratories that could not offer to maintain costly computing centers otherwise.

In the rest of this paper, we analyze and specific requirements of medical applications with an interest for grid technologies. Some of the components described below are currently under investigation within the European DataGrid project (EDG [6]) and the French MEDIGRID ACI project [10]. Some practical example are given.

2 Grid added value for medical applications

Obviously, grids provide a large computing power and data storage space for complex algorithms that require large amounts of disk, memory and processing resources, but there is more than that to grids that make them a potential tool to widen the actual limitations of medical applications. Indeed, grids are a vector for:

- allowing distribution of large datasets over different sites and avoiding single points of failure or bottlenecks;
- enforcing the use of common standards for data exchanges and making exchanges between sites easier;
- enlarging the datasets available for large scale studies by breaking the barriers between remote sites;
- allowing a distributed community to share its computational resources so that a small laboratory can proceed with large scale experiments if needed;

- opening new application fields that were not even thinkable without a common grid infrastructure (eg large scale epidemiology studies).

Grids are likely to have a deep impact on health related applications by playing a key federative role [2]. This is the opportunity to develop completely new applications. Grids provide a logical continuation to regional health networks [8] by allowing distant sites to collaborate and exchange their data for specific research purposes.

The technical requirements for building a medical-oriented grid can be roughly split into two broad categories: data-related requirements and computation-related requirements. A successful medical grid will necessarily take into account these two complementary aspects.

3 Data-related requirements

Medical data are very sensitive as they often carry information about individuals. Another particularity of medical data is their rich semantic content.

3.1 Medical data security

The primary concern when distributing medical data over a grid is privacy. Medical applications often deal with patient data that are private and should only be accessible to the patient himself, the medical team involved in his health care, and, under some restrictions, for research purposes. Therefore, a medical grid, opened to a wide community of users, should enforce strict access right checking. Privacy of medical data stored over a grid is particularly sensitive due to:

- The spread of data over many remote sites locally administrated.
- The replication mechanism triggering copies of data on any grid site without notification.

There are two confidentiality levels when considering medical data: (i) nominative data are the most critical and should only be accessible to the patient and a limited accredited medical team, while (ii) other data (such as image content) are restricted to authorized persons only but they do not allow the identification of a person from the data content. It is very important in a grid environment to ensure that no relation can be established between a person and its data except for a few accredited users. Also note that it may be difficult to determine whether some data is private (whether it carries nominative information or not). Indeed, a

medical image is usually not sufficient to recognize the patient it was acquired from but in some cases (*eg* high resolution image of the head) it might provide information that can indirectly be used to identify the patient.

We can consider four groups of users involved in medical data manipulation: (i) the patient from whom the data originates, (ii) the physicians or other specialists involved in health care, (iii) researchers needing an access to this data and (iv) any other grid user. In general, no private or personal data should ever be accessible to any grid user. This group includes system administrators of grid clusters who are not accredited to manipulate medical data. This makes security enforcement complex as they have full access to resources under their responsibility. Therefore, medical data should be encrypted when transferred onto the grid, and encryption key should only be stored on secured (trusted) sites with strict controls over who is allowed to retrieve them. Patients should always have full read access to their data. Physicians involved in the health care of a given patient need read/write access. However, any physician should not be able to access any patient data. Finally, researchers should be able to access personal data (not private ones) for research purposes if a physician with access to the data grants them this authorization.

Therefore, the grid authentication and authorization mechanism should allow individual authentication of people accessing the grid, and control individually the access right to the data. This individual authorization mechanism should be subtle enough to avoid that a system administrator can establish a connection between a personal data and the corresponding patient name. The content of private data should not be accessible to anybody but the authorized user and therefore kept secure on a limited number of trusted sites. The content of personal data may be distributed over the grid provided that is as been encrypted to make its content retrieval by unauthorized parties almost impossible.

3.2 Medical data semantics

Another particularity of Medical data is their strong semantic content. A medical image itself is often of low interest if it is not related to a context (patient medical files, other similar cases...). Therefore, grids should provide tools to organize, relate, and describe medical image contents. The medical information system should not only deal with data but also their semantics by providing standardized ways of describing their content. Tools to manipulate metadata attached to the data are a first step in this direction.

In general, medical applications do not manipulate

files (low level data elements with semantic description limited to the file name) but datasets that are of interest to the user because of some property (*eg* images of a given patient, images of some region of interest acquired on some device...). This is illustrated in the DICOM standard [5] format which organizes data around examinations rather than flat image files and stores a large amount of metadata in image headers. Moreover, metadata are distributed over different sites just like data are. For exactly the same reasons of scalability and fault tolerance that lead to replication mechanisms for managing grid flat files, metadata need to be replicated in multiple instances.

Therefore, the grid should provide a basic support for storing metadata (usually in relational databases) on different sites, make distributed queries over these data and trigger replicates when needed for efficiency or robustness reasons. Beyond the storage and retrieval of metadata, facilities are expected to make use of data easier from an application point of view. The middleware should allow description of datasets from queries on metadata and the definition of job inputs and/or outputs through metadata. A complete synchronization between the metadata management system and the data management system are needed to ensure coherence.

Another related requirement for a medical data management system is traceability. It should always be possible to know, for a given image where it originates from (which algorithm and which input image(s) were used to produce it). Indeed, physicians often need to come back to the unaltered data when studying a processed image. Conversely, for each input data it is of interest for optimizing computations to record which output has already been processed using various algorithms (computation results cache). Extending this idea further, the grid middleware could use this facility for storing either an output data or the description of how it can be obtained from some input data (and make space optimization based on the relative cost of results recomputation compared to the needed storage space). This is achievable not only with a data management system that stores metadata describing computations done, but also with an algorithm management service that make algorithms known to the grid middleware and reprocessing possible by picking algorithms out of a database where they have been registered.

4 Computation-related requirements

Like any grid application, medical applications first expect from a middleware a transparent access to distributed resources and data. It is extremely impor-

tant in the medical domain since, as already stated in the introduction, the medical community is spread and it is not only a way to obtain an increase computing power but also to foster collaborations and promote standards [2].

However, medical applications usually require more than a grid-wide scheduler and a batch job submission system. A medical experiment often involves not a single algorithm but a set of processings that can sometimes be executed concurrently. A support for the so called processing pipeline, described below, is expected from the grid middleware. Medical applications reliability is also sometimes critical and an expert need to assess the computation result or even to guide the algorithm during the computation. Therefore, the grid should allow user interaction with submitted processes, which implies that computations should be fast enough to allow a fluid interaction with a physician. This is made difficult by the remote execution of the computation whereas the application feed-back (visualization and user input) should be returned to the user desktop machine [11]. The most demanding cases are simulation applications for which real time constraints have to be enforced. Finally, a medical grid should be able to take into account emergency situations by allowing some users (surgeons, emergency services) to send high priority jobs, preempting running jobs when needed.

4.1 Pipelining computations

Processing pipelines are compound jobs composed of several elementary stages (each stage being an algorithm applied onto an input dataset and producing an output dataset). Several stages can be processed on different machines. Stages are chained (*eg* the output of stage A is used as input for stage B) but are not necessarily linear (*eg* both stages B and C can be processed in parallel). Therefore, the pipeline service should offer a pipeline description mechanism to draw the architecture of the pipeline and a smart scheduler able to exploit the pipeline intrinsic parallelism by distributing proceedings on various grid nodes (data-flow control, load balancing, synchronization...). Pipelines are of real interest when processing a large number of input data rather than a single input. Through pipelines, the user can describe once for all the chain of transformations that each element of the input dataset should undergo. The pipeline scheduler can process several elements in parallel on grid nodes (thousands of concurrent input images are expected for some medical applications). Synchronization barriers may be needed to extract statistics from several processed data at some point in the process flow. Therefore, pipelines

should provide additional services such as synchronization, logs of accomplished stages for a given input, restart from a failing job, automatic resubmission of stages that failed for user-independent reasons, etc.

4.2 Parallel computations

Some image processing, simulation, and modeling algorithms are very computation intensive and need a parallel implementation in order to get executed in a reasonable amount of time compatible with clinical practice constraints. It is even more critical with applications that require interactivity for which the user can only remain a reasonably short amount of time in front of his computer screen, waiting for the algorithm to process data and return an output. Support for parallel computations is mandatory for these applications. Local area parallelism is widely available today through message passing interfaces. However, grid-wide parallelism involving heterogeneous machines and networks is an area that still requires investigation.

4.3 Interactive applications

Interaction with the user may be needed for guiding the algorithm, to solve legal issues when dealing with medical data, or for the application itself (*eg* therapy simulator). Data compression and high-bandwidth network should insure a limited response time which is mandatory for interactive usage. Interactive feed-back often involves 3D visualization of medical scenes. This can become challenging due to the large size of 3D medical images and the complexity of meshes used for realistic 3D modeling [11]. To achieve user interaction with grid jobs, a communication should be possible between the computing node(s) and the user workstation. Firewalls are often opposing resistance to these communications and computing cluster may be isolated from the external network.

4.4 Accessibility to grid resources

The medical community is very large and in general not very much aware of computers and grid technologies. Therefore, a tremendous effort is needed to interface today existing grid middlewares and make them usable by this community. High level services such as data search engines, easy access to job submission, graphic pipeline construction tools, algorithms selection and execution from input data or metadata sets, are needed for grid-aware medical applications development. Comprehensive and easy-to-use interfaces built

on top of these services are mandatory for grid tools to become accepted by the medical community.

Another key point in the deployment of grid technologies is the trust the end users can have in the underlying architecture. Medical data being confidential in general, the deployment of many medical applications will only be feasible if the middleware is recognized as secured and controlled. This prevents the use of middlewares with a lazy security infrastructure or not enforcing strict checking on data and computation resources usage.

5 Medical application examples

Existing middlewares are far from responding to all the requirements described above. There is still a long way to go before grids can be widely used in the medical domain. However, we are developing prototypes and experimenting a first middleware dedicated to medical applications. Two typical grid-aware medical applications are detailed below.

5.1 Simple combined search and execution application

The first application presented here is very generic and corresponds to the routine manipulation a medical user should be able to accomplish using grid services [12]. A physician wants to select input images corresponding to some criterion (*eg* all images from Mr. X, all Magnetic Resonance Images of the heart with a given pathology) and perform some computations on the resulting dataset to obtain quantitative results for guiding its diagnosis or assessing a work hypothesis. Figure 2 illustrates this application that is currently investigated on the European DataGrid testbed:

1. The user queries the metadata tables to retrieve one, or a set of, medical images he is interested in.
2. The grid-wide file identifiers (LFNs) of the selected data are returned.
3. A Job is created for each input file and submitted to the grid resource broker.
4. The resource broker translates file identifiers to physical file instances (PFNs) and search for available resources to process the jobs.
5. The job is sent to a worker node where it is executed. Outputs are created.
6. The resulting files are returned to the user interface machine.

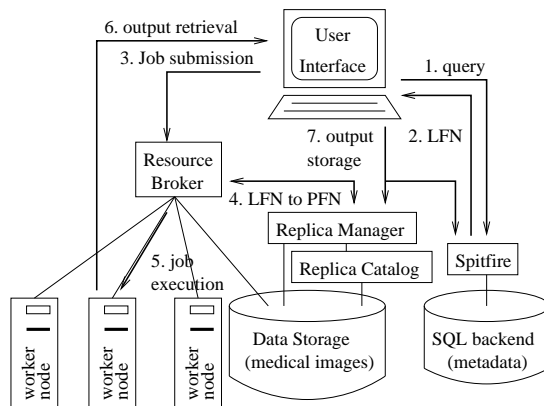


Figure 2. A generic medical application framework

7. Interesting files are sent to a storage space and the associated metadata are recorded into the SQL database for future use.

We rely on the middleware job submission service for job execution [13]. Data are stored on a disk storage element. Metadata are stored in an SQL database accessed through a secured interface [7]. The application framework presented above has been successfully implemented to compute similarity measures between a source image and a set of target images. The source image is evaluated against each target image independently. The use of the grid allows to parallelize the computations over hundreds of computation nodes, providing a near linear acceleration of the computation time.

5.2 Image processing for multiple sclerosis treatment trial

Multiple sclerosis is a severe brain disease that affects about 0.001% of the population in industrialized countries and for which there exists no complete redemption treatment. Few drugs are available on the market today that slow down the brain impairment caused by the disease, although their efficiency is difficult to quantitatively assess and their real effect is rather controversial. Assessments of these therapies have been proposed through serial Magnetic Resonance (MR) images of the head by measuring the brain white and gray matter atrophy resulting from the disease [4, 9]. However, this parameter extraction requires complex image analysis algorithms since very small volume variations are significant (the normal brain atrophy due to aging is in the order of 0.5% per year, while

the disease may lead to an accelerated atrophy in the order of 1% per year). Therefore, only studies involving a large number of patients over a long period of time prove to have a statistical significance.

Such an epidemiological study involves at least hundreds of patients (a group of placebo patients and several groups of treated patients following an experimental protocol) over years (an MR acquisition every few months is required to build time series). This kind of clinical protocol results in the acquisition of thousands of images, 10 to 20 Mb each, summing up to terabytes of data [3].

The processing of a single image implies a large number of basic image processing algorithms: each image is first registered in a common space frame (the Talairach frame) to compensate for head positioning differences in the scanner and scale. Then, images are resampled to compensate for different acquisitions resolution. These stages are preceded by various signal improvement and acquisition bias correction algorithms to improve the image quality and the subsequent processings reliability. Finally, classification of brain tissues and quantitative measurements of the brain volumes are extracted through anatomical modeling of the brain structures. From these volumes, atrophy measurements are estimated and the statistical study can begin. This complete processing chain involves 20 to 30 elementary processings on each image. The total computation time for one image is 30 to 60 minutes, leading to very long computations of the complete database except when considering a large number of processors to deal with all input images in parallel.

The gridification of this application over a 30 processors cluster with a custom middleware allowed to parallelize some stages of the processing chain and to process several images in parallel. It reduced the overall computation time to 10 days. This makes computations tractable but this is not sufficient for a real usage in clinical research. Many trial and error experiments are needed to optimize the image processing chain and discover the most discriminating algorithms from a statistical point of view. The overall computation time really needs to be reduced for a real extension of that kind of epidemiological study. Grids are a tool to achieve much more optimization and parallelization, opening new opportunities for automated medical image analysis.

6 Conclusions

Grids have known intensive developments over the past few years. Basic middlewares are today available and it is now time for grid developers to turn toward

specific applications. There is a tremendous potential in grid technologies to face today's medical application challenges. However, only secured and specialized infrastructures with high level services designed to meet the application requirements will really allow a large scale acceptance and deployment of grids in the medical community.

References

- [1] R. Acharya, R. Wasserman, J. Sevens, and C. Hinojosa. Biomedical Imaging Modalities: a Tutorial. *Computerized Medical Imaging and Graphics*, 19(1):3–25, 1995.
- [2] V. Breton, R. Medina, and J. Montagnat. Datagrid, prototype of a biomedical grid. *to appear in Methods of Information in Medicine*, 42, 2003.
- [3] D.L. Collins, J. Montagnat, A.P. Zijdenbos, and A.C. Evans. Automated estimation of brain volume in Multiple Sclerosis with BICCR. In *Information Processing in Medical Imaging*, Davis, USA, June 2001.
- [4] G. Comi, M. Philippi, V. Martinelli, G. Sirabian, A. Visciani, A. Campi, S. Mammi, M. Rovaris, and M. Canal. Brain Magnetic Resonance Imaging correlates of cognitive impairment in multiple sclerosis. *Journal of the Neurological Science*, 115:66–73, 1993.
- [5] DICOM: Digital Imaging and COmmunications in Medicine. <http://medical.nema.org/>.
- [6] European DataGrid project. <http://www.edg.org/>.
- [7] W. Hoschek and G. McCance. Grid Enabled Relational Database Middleware. In *Global Grid Forum 3*, Frascati, Italy, October 2001.
- [8] H. K. Huang. *PACS: Picture Archiving and Communication Systems in Biomedical Imaging*. Hardcover, 1996.
- [9] N.A. Losseff, L. Wang, H.M. Lai, D.S. Yoo, M.L. Gawne-Caine, W.I. McDonald, D.H. Miller, and Thomson A.J. Progressive cerebral atrophy in multiple sclerosis, a serial MRI study. *Brain*, 119(6):2009–2019, 1996.
- [10] MEDIGRID, ACI-GRID project, French Research Ministry. <http://www.creatis.insa-lyon.fr/medigrid/>.
- [11] J. Montagnat, E. Davila, and I.E. Magnin. 3D objects visualization for remote interactive medical applications. In *3D Data Processing, Visualization, Transmission*, Padova, Italy, June 2002.
- [12] D. Sarrut and S. Miguet. ARAMIS: A Remote Access Medical Imaging System. In *International Symposium on Computing in Object-Oriented Parallel Environments*, San Francisco, USA, December 1999.
- [13] M. Ruda *et al.* Integrating GRID tools to build a computing resource broker: activities of DataGrid WP1. In *Computing in High Energy and Nuclear Physics*, Beijing, China, September 2001.