A Medical Image Retrieval Application Using Grid Technologies To Speed Up Feature Extraction

[Extended Abstract]

Xin ZHOU
Medical Informatics Service,
University and Hospitals of Geneva,
CH–1211 Geneva 14, Switzerland
xin.zhou@sim.hcuge.ch

Mikko Pitkänen
Helsinki Intstitute of Physics,
Technology Programme
CERN/PH CH–1211 Geneva, Switzerland
Mikko.Pitkaenen@cern.ch

Adrien Depeursinge, Henning Müller
Medical Informatics Service,
University and Hospitals of Geneva,
CH–1211 Geneva 14, Switzerland

ABSTRACT

Medical image data is produced in ever–increasing quantities and varieties. The digital production of these data makes them accessible for further automatic analysis and processing. Whereas automatic analysis is fairly common in the text domain, analysis of medical images in large quantities and in a large variety is a relatively new discipline. Computational limits are often restricting the possibilities to analyze the large amounts of data produced automatically. Grid computing opens new possibilities to use an intra–hospital computing infrastructure for research projects.

This article describes the griddification of a content–based image retrieval system called the GNU Image Finding Tool (GIFT). The goal of this study was to show the potential of grid computing and the benefits for the medical applications from this available computing power. We use the ARC (Advance Resource Connector) middleware for the distributed computing power available through the KnowARC research project funded by the European Union.

The feature extraction part of the GIFT was griddified. A hospital grid conception is explained. Grid performance was measured with the speed of the griddified system for several submission scenarios. Grid computing has the potential to help computer science researchers in medical institutions to better use an existing infrastructure. It shows that particularly computationally–intensive tasks such as the extraction of visual features from large image databases can be performed much faster. This allows to explore more complex feature spaces and also larger image datasets.

Categories and Subject Descriptors

H.3.1 [**Information search and retrieval**]: Indexing methods; J.3 [**Life and medical sciences**]: Medical information systems

Keywords

content–based image retrieval, medical image retrieval, grid computing

1. *INTRODUCTION*

Modern radiology departments are increasingly becoming digital, and at the same time the amount of data produced is rising [21]. As images are an important part of the diagnostic process many medical imaging applications have been developed over the last 20 years to help medical doctors in the analysis of the images. Most applications have concentrated on one very specific sort of images, anatomic region, and often one disease [18]. Content–based medical image retrieval in general had the goal to allow for retrieval of similar images or cases over very heterogeneous image collections [11, 15, 20] to help the diagnostic process. With modern radiology departments routinely producing tens of thousands of images per day [16] it became apparent that infrastructures are required to tread this extremely large amount of data.

Grid technologies are one approach to make available computing power for large–scale research projects [9]. The goal is in general to have a very large number of machines in various locations that can be shared for computationally intensive tasks. Many solutions have been prosed for technical approaches to grid computing. The roots of grids can be traced back to the late 1980s [10]. Large and complicated frameworks such as Globus [8] appeared in the late 1990s and created a basis for further middleware developments. Currently, a large number of grid related middlewares are in routine use, for example gLite, UNICORE [17] and ARC (Advance Resource Connector) [6]. Grid computing in the health domain was fostered by the healthGrid[1] initiative in

[1] `http://www.healthgrid.org/`

2002. The conferences of this initiative developed several white papers and a road map for grids in the life sciences [3, 2]. In general, most medical grid applications concentrate on compute–intensive problems [1, 7]. Functionalities such as data management and particularly security issues when treating medical data have not been widely addressed [4]. Most of these applications also concentrate on using large available clusters mainly from the physics domain or from Universities for the processing, rarely looking directly at the needs of clinical centers, which reduces the technology uptake in this sector.

In the University and hospitals of Geneva a grid project started in 2002 to identify challenges for this technology in hospitals [14]. Goal was to employ grid technology to use the large number of desktop computers (6'000 in case of the Geneva University hospitals) as a resource for research projects. Most hospitals do not have any research computing infrastructure and no personnel to maintain such a potential internal infrastructure. On the other hand would such an infrastructure limit the security problems closely linked to medical data. First concrete steps for such an infrastructure were presented in [16]. Several other authors propose the use of grid infrastructures for medical image retrieval with varying architectures [5, 12].

This paper addresses the challenges mentioned above and describes our approach for medical image retrieval using a grid infrastructure Section 2 describes the methods used for our implementation. Section 3 presents the grid deployment infrastructure and architecture of the griddified system, together with initial test results. Finally, a discussion concludes this paper.

2. *METHODS*

This section describes the medical image search engine used for our research and the environment in which the griddification was performed.

2.1 Operating system

The vast majority of computers in the Hospitals of Geneva are using Windows as their operating system. The software management of the Windows machines is uniform and software distribution for all clients is centralized. So far, Linux has only been used on research machines and as a server operating system. The ARC middleware, like much other scientific computing software, requires Linux as a host system. For our internal computing needs we created virtual machines using VMware[2] (Virtual Machine Ware) on our client windows machines to install a Linux environment for testing the middleware.

2.2 Network environment

The network policies inside the hospitals set strict constraints for the deployment of a grid infrastructure. The network addresses inside the hospital are distributed by using the Dynamic Host Configuration Protocol (DHCP) to assign IP (Internet Protocol) addresses based on the address of network card (MAC address, Media Access Control). A very restrictive firewall blocks all traffic to the outside world and only allows single ports to be opened selectively between two defined machines. To test our routines on the KnowARC[3] project resources we used two servers on the University network that we had access to from inside the hospitals.

2.3 The GNU Image Finding Tool

The griddification is based on the GNU Image Finding Tool (GIFT)[4]. In order to retrieve images similar to an example the entire collection of images needs to be indexed, meaning that visual features need to be extracted to represent each image. More on the GIFT can also be found in[19]. For computational reasons the features of GIFT are extremely simple color and texture features that computer very fast (1-2 seconds per image). Still, for very large collections this can take hours or even full days.

2.4 Dataset used

For this study we used a dataset made available by the ImageCLEF[5] medical image retrieval task[13]. ImageCLEF is part of the Cross Language Evaluation Forum (CLEF), which is a forum for benchmarking information retrieval research. This database contained 50'000 images in 2005 and 2006 and almost 70'000 images in 2007. The dataset of the ImageCLEFmed 2007 retrieval task containing in total 66'735 images was used to test the computation performance described in this article. These images are located in a server which is accessible from desktop machine. In principal the features of every image can be calculated in parallel and independently of other images. In our scenario, we group the images in blocks of 500–1000 to be computed on the same node.

3. *RESULTS*

This section describes the main outcome of a first pilot study for grid deployment in the University Hospitals of Geneva.

3.1 Application architecture

An important point for application developers is a simplification for modifying an existing application for using grid resources, which are usually run in batch jobs. A grid middleware is usually fairly complex to configure as it relies on a large number of components that need to be configured properly for optimal use. One solution is to use a relatively lightweight middleware client with the least possible configuration and to regroup the various parameters to automate part of the configuration. In Figure 1 we present the basic architecture for our griddified image retrieval application.

Another major bottleneck is the interface to manage the jobs running on the grid. In our solution, we emphasize the existence of a local job manager. Many middlewares have global job managers to provide the job state and execution details for all the jobs running on the grid at a certain moment. This allows a user to interrupt, restart, or re–submit jobs. This kind of interface is often preferred by system administrators as all the information linked to a single cluster is visible and configurable. However, the users do not always get the maximum benefit from such a situation. Users are often more interested in their jobs related to a particular

[2] http://www.vmware.com/

[3] http://www.knowarc.eu/

[4] http://www.gnu.org/software/gift/

[5] http://www.imageclef.org/

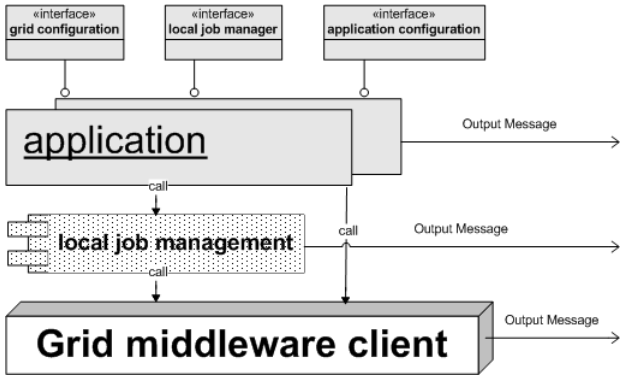

Figure 1: The basic architecture for the griddified application.

application. For the users, the manual control of the grid job execution is an overhead.

A local job manager (JM) concentrates on the jobs submitted for a single application session. It collects information by communicating with the global JM to avoid a duplication of work. No job execution details are generally provided, and job state information is visible through a graphical interface. No configuration is expected from the user for tasks such as optimization of resource usage, re–submission of jobs, and work directory cleaning. This reduces a major part of the complexity from the users and for the developers. The ARC community provides such a grid JM called GridJM[6].

Another important aspect is that feedback presented to the users needs to be intuitive. The output (especially error codes) from a grid middleware is often clear for experts but hard to understand for unexperienced users. Application users do also often not want to deal directly with the details of grid technology. To hide all the details of a grid middleware and to provide suitable messages based on the user's knowledge is important.

3.2 Deployment infrastructure

To exploit the desktop resources in the hospitals it is important to address several security issues. To protect the data from being read by unauthorized persons the resources used for computing are strictly separated from the host environment. Figure 2 shows a virtual network setup to enable grid connectivity. All the machines in blue are virtual machines providing CPU, memory, and disk space. The free version of VMware Server is used in our tests.

3.3 Comparison of computation speed

The deployment of a local grid inside the Hospitals requires the collaboration with the responsible network administrators in the Hospitals. The first tests were performed with resources of the KnowARC project to avoid any security problems A virtual organization of 37 CPUs was used to simulate the resources available as well on a local grid. In Table 1 a comparison between a single dedicated server (2xDualcore Xeon 2 GHz, 4 GB RAM, 700 GB disk) is performe with the use of the KnowARC grid.

[6]http://www.tcs.hut.fi/~aehyvari/gridjm/

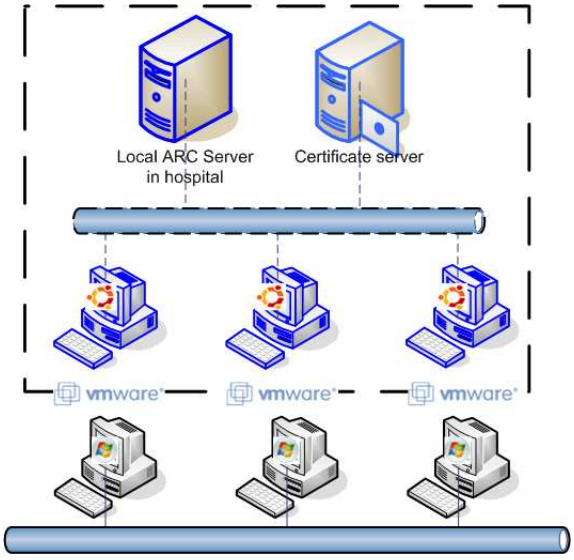

Figure 2: The infrastructure for the cluster deployment.

Table 1: Comparison of computing times between a large server and a small grid.

	Server	Small Grid
number of CPUs	4	37
total time	709.1 min	536.6 min

The total time listed corresponds the time required by GIFT to finish the indexation for the entire ImageCLEFmed database containing almost 70'000 images. The shown times are strongly influenced by the large amounts of data (67 packets of 1000 images in total over 4 GB) that need to be transported to the remote machines before computation. This transfer happens from a single machine in Geneva. As the partners are in several countries and sometimes with only slow connection to Switzerland this has a slowing effect on the execution. A time analysis for each submitted job is shown in Table 2.

In Table 2 the longest and average times are given for extracting the features of one packet of 1000 images. The execution time is the time actually used for computation, only. The waiting time in this case is the time lost before the actual computation starts (scheduling, queuing, security operations, and transfer of inputs and outputs).

4. *DISCUSSION*

Table 2: Comparison of execution time, time spent waiting, and the total time for executing the feature extraction of 1000 images.

	longest	average
execution time	63.43 min	44.78 min
waiting time	47.23 min	7.07 min
total time	94.68 min	51.85 min

This article describes an approach to griddify a medical image retrieval application. An architecture is described to use computing resources available inside the Hospitals for computation, using a virtualization–based approach for installing Linux on standard Windows desktops in the Geneva University Hospitals. First tests show that the computation time can be reduced but it also shows that the time for transferring the data is important. This is due to the fact that data transfer and storage to faraway resources creates an important time loss. First tests with local resources based on the same technology showed to improve computation times by a factor of almost 10. The latency problems should disappear with the use of local resources so much stronger improvements are expected here. Such an available computing infrastructure for research can help developing new and more complex features as well as keep up with the strong data production and start indexing a large part of medical images produced daily.

Acknowledgments

This study was supported by the EU in the 6th FP through the KnowARC project (Grant IST 032691).

5. REFERENCES

[1] I. Blanquer, V. Hernandez, D. Segrelles, and E. Torres. Trencadis – secure architecture to share and manage dicom objects in a ontological framework based on ogsa. In *Healthgrid 2007*, pages 115–124, Geneva, Switzerland, April 2007.

[2] V. Breton, I. Blanquer, V. Hernandez, N. Jacq, Y. Legre, and P. Wilson. Roadmap for a european healthgrid. In *Healthgrid 2007*, pages 154–163, Geneva, Switzerland, April 2007.

[3] V. Breton, I. Blanquer, V. Hernandez, Y. Legre, and T. Solomonidés. Proposing a roadmap for healthgrids. *Stud Health Technol Inform*, 120(iss):319–329, 2006.

[4] V. Breton, K. Dean, and T. Solomonidés. The healthgrid white paper. In *Proceedings of Healthgrid conference*, volume 112. IOS Press, 2005.

[5] M. Costa Oliveira, W. Cirne, and P. M. de Azevedo Marques. Towards applying content–based image retrieval in clinical routine. *Future Generation Computer Systems*, 23:466–474, 2007.

[6] M. Ellert, M. Grønager, A. Konstantinov, B. Kónya, J. Lindemann, I. Livenson, J. Langgaard Nielsen, M. Niinimäki, O. Smirnova, and A. Wäänänen. Advanced resource connector middleware for lightweight computational grids. *Future Generation computer systems*, 23(2):219–240, 2007.

[7] S. G. Erberich, J. C. Silverstein, A. Chervenak, R. Schuler, M. D. Nelson, and C. Kesselman. Globus medicus – federation of dicom medical imaging devices into healthcare grids. In *Healthgrid 2007*, pages 269–278, Geneva, Switzerland, April 2007.

[8] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115–128, Summer 1997.

[9] F. Gagliardi, B. Jones, M. Reale, and S. Burke. European datagrid project: Experiences of deploying a large scale testbed for e-science applications. In M. Calzarossa and S. Tucci, editors, *Performance Evaluation of Complex Systems: Techniques and Tools, Performance 2002*, Lecture Notes in Computer Science, pages 480–500. Springer–Verlag, 2002.

[10] M. Litzkov, M. Livny, and M. Mutka. Condor — a hunter of idle workstations. In *Proceedings of the 8th international conference on distributed computing*, pages 104–111, San Jose, California, USA, June 1988.

[11] H. J. Lowe, I. Antipov, W. Hersh, and C. Arnott Smith. Towards knowledge–based retrieval of medical images. The role of semantic indexing, image content representation and knowledge–based retrieval. In *Proceedings of the Annual Symposium of the American Society for Medical Informatics (AMIA)*, pages 882–886, Nashville, TN, USA, October 1998.

[12] J. Montagnat, V. Breton, and I. E. Magnin. Partitioning medical image databases for content–based queries on a grid. *International Journal of Supercomputer Applications*, 44(2):154–160, 2005.

[13] H. Müller, P. A. Do Huang, A. Depeursinge, P. Hoffmeyer, R. Stern, C. Lovis, and A. Geissbuhler. Content-based image retrieval from a database of fracture images. In *SPIE Medical Imaging*, 2007.

[14] H. Müller, A. Garcia, J.-P. Vallée, and A. Geissbuhler. Grid computing at the university hospitals of geneva. In *Proceedings of the 1st healthgrid conference*, pages 264–276, Lyon, France, January 2003.

[15] H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler. A review of content–based image retrieval systems in medicine – clinical benefits and future directions. *International Journal of Medical Informatics*, 73:1–23, 2004.

[16] H. Müller, M. Pitkanen, X. Zhou, A. Depeursinge, J. Iavindrasana, and A. Geissbuhler. Knowarc: Enabling grid networks for the biomedical research community. In *Healthgrid 2007*, pages 261–268, Geneva, Switzerland, April 2007.

[17] M. Romberg. The unicore grid infrastructure. *Scientific Programming*, 10(2):149–157, 2002.

[18] C.-R. Shyu, C. E. Brodley, A. C. Kak, A. Kosaka, A. M. Aisen, and L. S. Broderick. ASSERT: A physician–in–the–loop content–based retrieval system for HRCT image databases. *Computer Vision and Image Understanding (special issue on content–based access for image and video libraries)*, 75(1/2):111–132, July/August 1999.

[19] D. M. Squire, W. Müller, H. Müller, and T. Pun. Content–based query of image databases: inspirations from text retrieval. *Pattern Recognition Letters (Selected Papers from The 11th Scandinavian Conference on Image Analysis SCIA '99)*, 21(13–14):1193–1198, 2000. B.K. Ersboll, P. Johansen, Eds.

[20] H. D. Tagare, C. Jaffe, and J. Duncan. Medical image databases: A content–based retrieval approach. *Journal of the American Medical Informatics Association*, 4(3):184–198, 1997.

[21] M. W. Vannier, E. V. Staab, and L. C. Clarke. Medical image archives – present and future. In H. U. Lemke, M. W. Vannier, K. Inamura, A. G. Farman, and J. H. C. Reiber, editors, *Proceedings of the International Conference on Computer–Assited Radiology and Surgery (CARS 2002)*, pages 565–576, Paris, France, June 2002.