INSTITUTE OF PHYSICS PUBLISHING PHYSIOLOGICAL MEASUREMENT

Physiol. Meas. **26** (2005) S209–S215 doi:10.1088/0967-3334/26/2/020

Using the GRID to improve the computation speed of electrical impedance tomography (EIT) reconstruction algorithms

J Fritschy[1], L Horesh[1], D S Holder[1] and R H Bayford[1,2]

[1] Departments of Clinical Neurophysiology & Medical Physics, University College London, UK
[2] School of Health and Social Sciences, Middlesex University, London, UK

E-mail: j.fritschy@ucl.ac.uk

Received 8 October 2004, accepted for publication 14 January 2005
Published 29 March 2005
Online at stacks.iop.org/PM/26/S209

Abstract

In our group at University College London, we have been developing electrical impedance tomography (EIT) of brain function. We have attempted to improve image quality by the use of realistic anatomical meshes and, more recently, non-linear reconstruction methods. Reconstruction with linear methods, with pre-processing, may take up to a few minutes per image for even detailed meshes. However, iterative non-linear reconstruction methods require much more computational resources, and reconstruction with detailed meshes was taking far too long for clinical use. We present a solution to this timing bottleneck, using the resources of the GRID, the development of coordinated computing resources over the internet that are not subject to centralized control using standard, open, general-purpose protocols and are transparent to the user. Optimization was performed by splitting reconstruction of image series into individual jobs of one image each; no parallelization was attempted. Using the GRID middleware 'Condor' and a cluster of 920 nodes, reconstruction of EIT images of the human head with a non-linear algorithm was speeded up by 25–40 times compared to serial processing of each image. This distributed method is of direct practical value in applications such as EIT of epileptic seizures where hundreds of images are collected over the few minutes of a seizure and will be of value to clinical data collection with similar requirements. In the future, the same resources could be employed for the more ambitious task of parallelized code.

Keywords: electrical impedance tomography, inverse problem, non-linear, e-science, GRID, condor, Matlab®

1. Introduction

Solving the forward and inverse problem in electrical impedance tomography (EIT) requires handling of large systems of equations (Polydorides *et al* 2002). Geometry of biological organs cannot be modelled accurately by simple symmetric shapes (Bagshaw *et al* 2003); hence, numerical (rather than analytical) formulation of the forward problem, such as the finite and boundary element method (FEM and BEM) has to be employed (Vauhkonen *et al* 1999, Polydorides 2004). Doing so, unfortunately, creates two difficulties: first, geometrically accurate models are likely to be large, as they include many elements and, secondly, the accuracy of numerical methods depends upon the mesh density as well, especially in regions where there are high field gradients (Molinari *et al* 2001). However, increasing the mesh density introduces a corresponding increase in the associated system matrices. A number of research groups have employed linear approximation (Uhlmann 2004), which requires a single calculation of the current field and measurement field forward solutions, and single formation of the a Jacobian matrix; however, the complexity of the problem has extended beyond the linear approximation domain, frequently, so that non-linear iterative reconstruction is needed (Yerworth *et al* 2004). Efficient Krylov subspace (Shewchuk 1994) techniques for solving large sparse matrices of the forward solution can be employed, yet the system matrix factorization requires a substantial amount of computational resources. In addition, as for most non-linear reconstruction methods, line search is required within each iteration. This process demands repetitive calculation of the forward problem, so that calculation of the forward solution becomes a severe bottleneck.

Our group is investigating the use of EIT to image the effects of stroke in the human brain (Yerworth *et al* 2004). There appears to be a clear requirement for a non-linear reconstruction algorithm due to the large and complex changes in impedance at different frequencies. Unfortunately, a single frequency reconstruction over a modest shelled head mesh of 31 111 elements requires about 2.5 h for a 2.8 GHz processor with 2 GB RAM, whereas the clinical requirement is that an image be collected and reported within 30 min. In other applications, such as imaging the source of epileptic seizures, we will need to collect about 1000 images over the several minutes of a seizure. Processing these serially—at 2 h per image—would be impractical and the use of a Condor cluster is of direct practical benefit as they could be processed in half a day and made available to the medical team on the usual timescale needed for reviewing data while the patient is on the ward.

Recently, a new research area (e-science) has seen the development of a resource known as the GRID. The concept is the development of coordinated resources that are not subject to centralized control, which use standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service (Foster 2002, Foster *et al* 2001, 2002). The GRID offers a wide range of resources; one specific middleware tool is Condor[3]. This organizes idle system resources, such as idle workstations, into clusters, called pools, or collections of clusters called flocks, that can exchange resources. Condor then searches for these idle workstations to run computational tasks. When the owner of the workstation resumes computing, Condor transfers the job to another machine.

In this paper we describe, for the first time, the application of Condor in reducing the processing time for EIT reconstruction. Before approaching the solution of the challenging EIT reconstruction algorithms, we tested the method in two simpler situations. We first ran a test algorithm to verify the configuration of our code and secondly we tested an algorithm used to analyse the electroencephalogram (EEG), which presents medium complexity. Finally, for the

[3] Condor Project http://www.cs.wisc.edu/condor.

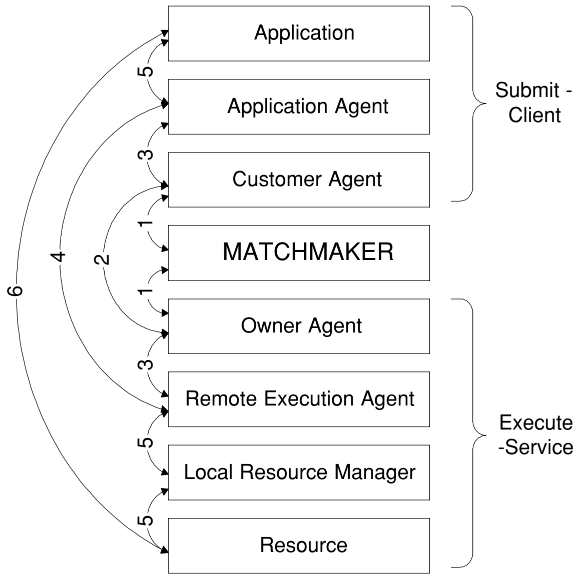

Figure 1. Key logical objects in Condor architecture.

EIT algorithm, a modified version of electrical impedance and diffuse optical reconstruction software (EIDORS) (Polydorides and Lionheart 2002) FEM generation functions have been used to produce the LHS of the forward solution system matrix. Incomplete Cholesky factorization has been chosen for preconditioning of the system matrix, and later plugged into a linear preconditioned conjugated gradients algorithm. The inverse solution framework was a Polak Ribiere non-linear conjugate gradients with regularized search direction scheme with Brent line search (Horesh *et al* 2004, Dorn *et al* 1999, Arridge 1999). The tests were run on a pool of 920 PCs running Condor in our local university (University College London).

The algorithm used was not parallelized internally. Multiple images were divided into individual tasks and sent to different nodes of the pool, which ran independent tasks at the same time. All the tasks were started at the same time. The ultimate potential of the GRID would clearly be best utilized with the parallel code which would permit rapid image reconstruction of individual images. Although this is our longer term goal, it was not practicable for this study. We have therefore elected to demonstrate the utility of the GRID in speeding up reconstruction of multiple images. This is of direct practical value in our work, as we are currently undertaking a trial of EIT in epilepsy, in which hundreds of images are collected over several minutes during a seizure. The purpose of this work was to demonstrate practical utility with the GRID in this preliminary fashion, with the hope that this would be of immediate practical value to other groups with a similar need during non-linear reconstruction of multiple clinical images, and also encourage activity into the parallelization of reconstruction algorithms in the future.

2. Methods

2.1. Condor architecture

The principles of the Condor method are as follows (Condor) (figure 1).

The three upper logical components (application, application agent and customer agent) are on the 'client-side'—in the workstation used by the user to submit the algorithm—and the four lower components (owner agent, remote execution agent, local resource agent and resource) are on the 'resource-side,' a workstation, which is part of the Condor system.

In the middle, between the two sides, there is the 'key piece' of the system: the matchmaker. On the resource-side, the information about available resources is being constantly updated in the four corresponding components, which have different levels of abstraction. At the end of this 'chain of information' is the 'owner agent,' which holds the knowledge regarding the amount of available nodes, their technical characteristics—such as memory and operating systems—and additional useful data about the resources. It regularly transmits this information to the matchmaker.

Once the client sends his tasks to the pool, the components on the client-side have information about the intended operation and the 'customer agent' sends the right information—such as number of tasks, amount of memory and operating system needed—to the matchmaker (number 1 in figure 1). The matchmaker then makes a decision as to which tasks will be executed, selects the resources, and allows direct communication between those (see action 2 in figure 1).

Afterwards, a 'chain reaction' is started, which allows the communication between peers on different sides of the matchmaker. At the end (action 6 in figure 1) the application's user contacts the resource.

2.2. *UCL-Condor pool specifications*

The UCL-Condor pool has 920 nodes (Pentium 3, 1 GHz. CPU, 256 MB RAM and 750 MB of free disk space, operating system: Windows 2000). These machines are distributed around the UCL campus.

2.3. *Matlab®-libraries*

Our second and third tests are executable files coded in Matlab®. To be able to run a Matlab® executable file in a machine that does not have Matlab® installed—which is the case for all the nodes in the pool—it is necessary to fulfil two requirements:

- Some DLL libraries have to be installed in a local directory. Matlab®'s executable file (mglinstaller.exe) does the deployment of these libraries into the local machine.
- Add the directory in which the mglinstaller.exe is deployed to the PATH variable of the machine.

2.4. *First test—a test algorithm*

This is a computationally undemanding algorithm, which takes less than 10 s in a well-specified PC, such as an Intel Pentium 4, 2.8 GHz and 512 MB RAM. (The term 'well-specified PC' will be taken to indicate a PC of this power in further discussion below.) This test algorithm was a simple piece of C code, which reads some data from the file *eegtestin.dat* and write a new file *eegtestout.dat*. We put the executable file and the input file in the root directory of the user. We also created the *submit.sub* (JDL code) file, which set the parameters to send the tasks to the pool, and put it in the same directory. A log file was created, which recorded when the file was submitted, started and finished.

2.5. *Second test*

In the previous example, just one job was run. In the more demanding second test, a set of 100 tasks, which normally take 74 min each in a well-specified PC (section 2.4), was sent to the Condor pool. Each was a Matlab® algorithm, which performed wavelet analysis on the EEG data for epileptic spike detection. The algorithm implemented a Daubechies 4 wavelet analysis with a window of 1024 samples. This was run on a single channel of EEG acquired in a patient with epilepsy lasting 22 min sampled at 200 Hz.

To run this Matlab® algorithm, it was necessary to deploy Matlab® libraries in each node. The executable was needed to: (1) deploy the necessary Matlab® libraries in the node (mglinstaller.exe), (2) set up the PATH variable, (3) run the desired executable file (4) retrieve the processed data and finally (5) clean up and restore the nodes to their initial condition.

2.6. Third test

The final test was a reconstruction of multi-frequency EIT data with a non-linear reconstruction algorithm. The whole dataset, measured from a realistic tank, consisted of 300 boundary voltages sets. The algorithm employed a shelled head shaped finite element forward model of 31 111 linear elements. Thirty one electrodes were positioned according to the extended 10–20 scheme over the scalp. The injection-measurement protocol included 258 entries, of which 21 were independent diametrically opposed injection pairs and 51 were independent adjacent measurement pairs. The inverse solution scheme was a non-linear Polak Ribiere conjugate gradients with a modified version of Kaczmarz filtered back-propagation regularization (Horesh *et al* 2004) and Brent line search. This inverse problem framework comprises a search for conductivity values which minimize a cost function of the weighted difference between the measured and the calculated boundary voltages. To produce a fair comparison, the algorithm was set to halt after 25 iterations. Each reconstruction in this scheme took approximately 2.5 h on the above well-specified PC.

3. Results

The first test was successfully accomplished once the submission format was clarified.

In the second test, while the execution of the 100 files (74 min each) requires 7400 min (5.1 days) in a well-specified PC (section 2.4), Condor finished the execution in 297 min. This was 25 times faster than serial execution in a well-specified PC, but, nevertheless, four times slower than the execution of a single task.

In the third test, the 300 tasks required approximately 750 h (31.2 days) in a well-specified PC whereas Condor executed it in 1128 min (18.8 h), which is 40 times faster than the normal execution.

4. Discussion

Overall, use of the Condor pool substantially reduced the time taken to process these programs by 25–40 times. However, this was not totally efficient, as the time taken was still considerably longer than the time for a single task on one machine. Some inefficiency seems inevitable, due to the overheads of the middleware in distributing and regulating the distributed processing. One specific design feature was probably largely responsible: one drawback of the implementation we used was the lack of checkpoints in Condor under Windows (Condor systems running under UNIX platforms have this feature). If a task is almost completed and the node is switched off, the task has to be migrated to another node and re-started anew.

We have described some characteristics of Condor running in a limited single-domain of 920 nodes. The developers of Condor have used the software provided by GT[4] (Globus

[4] Globus Project, http://www.globus.org.

Toolkit) to expand the capabilities of Condor to a multi-domain environment. The software that addresses this new multi-domain scope is called Condor-G and is potentially more powerful again.

In the future, we plan to analyse a promising Matlab® wrapper (Eres *et al* 2004) to avoid the compilation and manual submission in a multi-domain platform environment[5]. Eventually, we aim to analyse the possible migration of this technology to clinical applications. The tasks employed in this initial study were relatively simple, in that they comprised individual image reconstructions. Nevertheless, this is of direct practical benefit, as many hundreds of images will need to be produced rapidly for use of EIT in imaging epileptic seizures. We also plan to parallelize each individual reconstruction algorithm, so that an individual image could be processed more rapidly too.

In our group, we believe that more accurate anatomical EIT images are possible with the use of non-linear EIT reconstruction and finely detailed meshes in the forward solution. This work arose because of a simple practical limitation in that processing was taking far too long. It remains to be seen if this view will be supported by empirical testing of these advanced algorithms. However, if they are adopted, much more processing power will be needed to compute the reconstructions on a clinically useful timescale, and the GRID appears to offer exciting potential for researchers in the EIT community to achieve this.

Acknowledgments

The authors wish to thank Clovis Chapman, the UCL-Condor pool's administrator for providing support with the submission of the Matlab® code over the pool, Andrew Tizzard for providing the mesh, Andrea Romsauerova for providing the test data and the CT superintendent at UCLH, Jason Godbold, who scanned the tank for us. This work was supported by the BBSRC under an e-science grant.

References

Arridge S R 1999 Optical tomography in medical imaging *Inverse Problems*

Bagshaw A P *et al* 2003 Electrical impedance tomography of human brain function using reconstruction algorithms based on the finite element method *Neuroimage.* **20** 752–64

Dorn O, Bertete-Aguirre H, Berryman J B and Papanicolaou G C 1999 A nonlinear inversion method for 3D electromagnetic imaging using adjoint fields *Inverse Problems* **15** 1523–58

Eres M H, Pound G E, Keane A J and Cox S J 2004 User deployment of grid toolkits to engineers *All Hands Meeting (Nottingham, September)*

Foster I 2002 What is the GRID? *GridToday* **1** (July 22) Online at news.tgc.com/msgget.jsp?mid=286185&xsl=story.xsl

Foster I, Kesselman C, Nick J and Tuecke S 2002 The physiology of the grid. Globus project *Technical Papers* http://www.globus.org/research/papers/ogsa.pdf

Foster I, Kesselman C and Tuecke S 2001 The anatomy of the grid. Globus project *Technical Papers* http://www.globus.org/research/papers/anatomy.pdf

Horesh L, Bayford R H, Yerworth R J, Tizzard A, Ahadzi G M and Holder D S 2004 Beyond the linear domain—the way forward in MFEIT image reconstruction of the human head *EITS, XII ICEBI & V EIT Conf. (Gdansk EIT, Poland)*

Matlab® site. *Building Stand-Alone Applications* www.mathworks.com

Molinari M *et al* 2001 Adaptive mesh refinement techniques for electrical impedance tomography *Physiol. Meas.* **22** 91–6

Polydorides N 2004 Image reconstruction algorithms for soft-field tomography *Meas. Sci. Technol.* **15** 2263–74

[5] Enabling grids for e-science in Europe www.eu-egee.org.

Polydorides N and Lionheart W R B 2002 A Matlab toolkit for three-dimensional electrical impedance tomography: a contribution to the electrical impedance and diffuse optical reconstruction software project *Meas. Sci. Technol* **13** 1871–83

Polydorides N, Lionheart W R and McCann H 2002 Krylov subspace iterative techniques: on the detection of brain activity with electrical impedance tomography *IEEE Trans. Med. Imaging* **21** 596–603

Shewchuk J R 1994 Introduction to the conjugate gradient method without the agonizing pain *Technical Report, CMU-CS-94-125* School of Computer Science, Carnegie Mellon University

Uhlmann G 2004 *Developments in Inverse Problems Since Calderon's Foundational Paper* ed C Kenig and C Sadosky (Chicago, IL: University of Chicago Press) (ed M Christ)

Vauhkonen P J *et al* 1999 Three-dimensional electrical impedance tomography based on the complete electrode model *IEEE Trans. Biomed. Eng.* **46** 1150–60

Yerworth R J, Horesh L, Bayford R H, Tizzard A and Holder D S 2004 Robustness of linear and nonlinear reconstructions algorithms for brain *EITS, XII ICEBI & V EIT Conf. (Gdansk EIT, Poland)*