Real-Time Functional MRI Analysis Using Grid Computing

Epifanio Bagarinao,[1] Luis F.G. Sarmenta,[2] Yoshio Tanaka,[3] Kayako Matsuo,[1] and Toshiharu Nakai[1]

[1]Photonics Research Institute, National Institute of Advanced Industrial Science and Technology
AIST Kansai Center, Ikeda City, Osaka 563-8577 JAPAN
[2]Department of Information Systems and Computer Science, Ateneo de Manila University
Quezon City 1108, Metro Manila, PHILIPPINES
[3]Grid Technology Research Center, National Institute of Advanced Industrial Science and Technology
Tsukuba City, Ibaraki 305-8566 JAPAN

Introduction

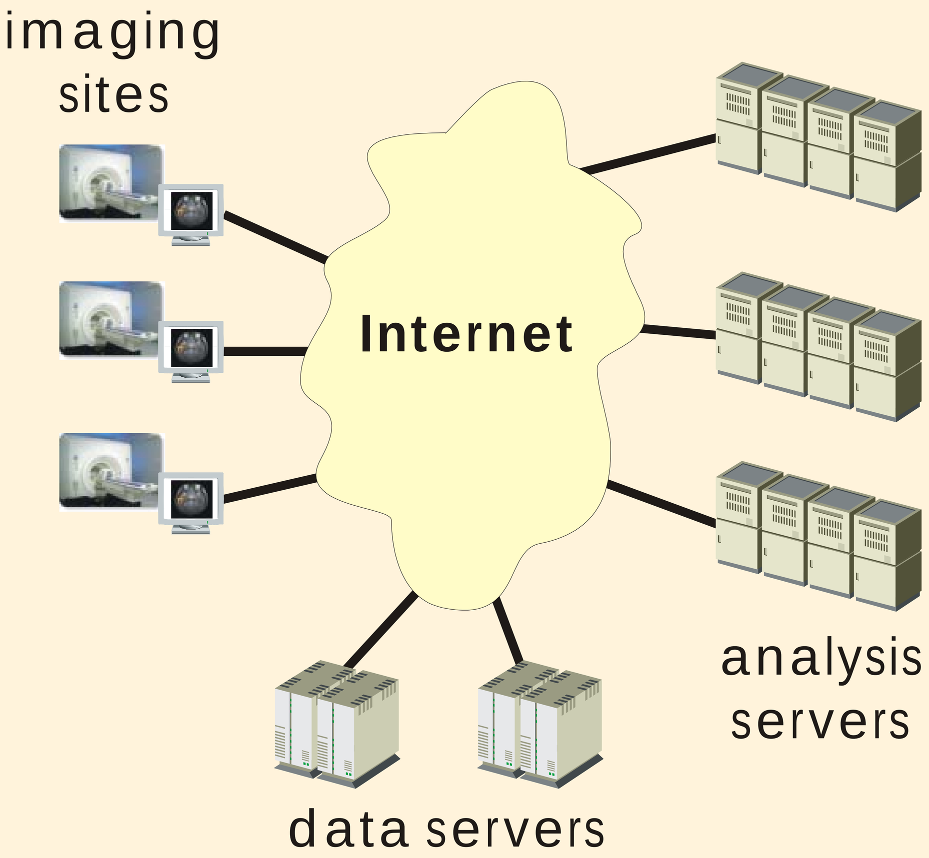
Real-Time fMRI (functional magnetic resonance imaging) is an evolving technique used for the immediate analysis of fMRI data with results available within a few seconds after data acquisition [1]. In an fMRI examination, the subject's brain is repeatedly scanned while the subject is performing certain tasks. The resulting brain scans are then analyzed to produce an activation map indicating regions in the brain that are active during the task. At present, most hospitals and medical imaging facilities do not have enough computing power to perform fMRI analysis in real-time. Thus, most fMRI examinations are analyzed offline and the results are only available hours or days after the examination.

Our Goal: To use **Grid Computing** to provide medical imaging facilities on-demand real-time fMRI capability.

Design

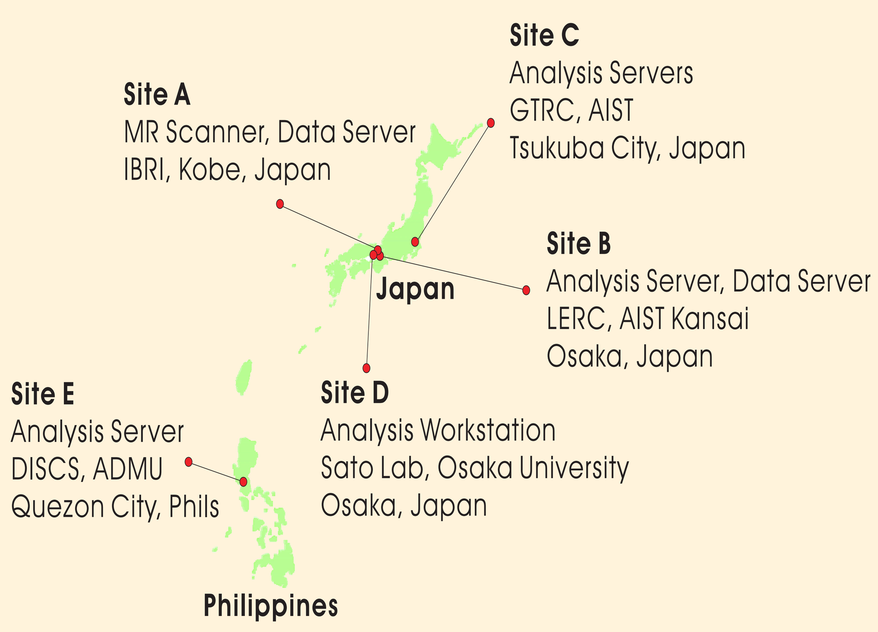
System Organization

We propose a computational grid consisting of several MRI acquisition systems, several data servers, several analysis servers, and their interconnection via the Internet for the real-time analysis of fMRI data.



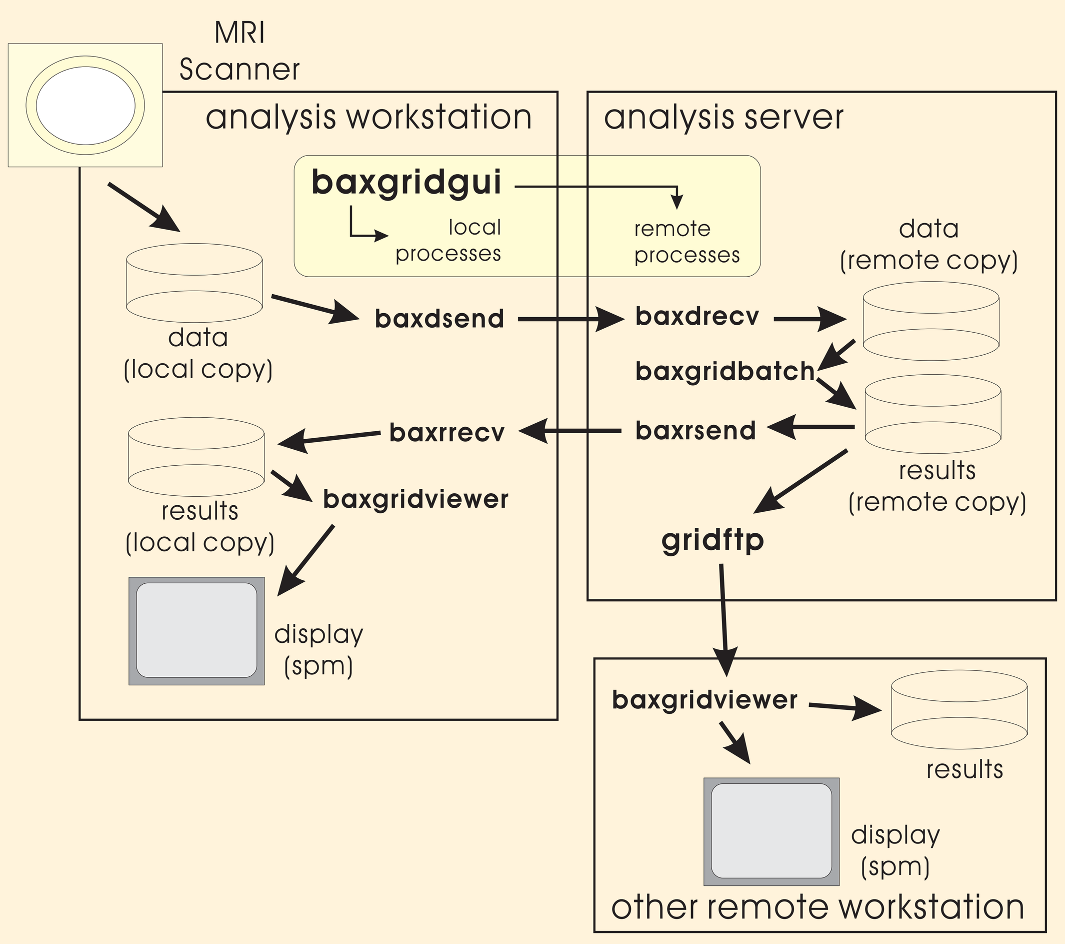
The Medical Grid Project Testbed (medgrid.org)

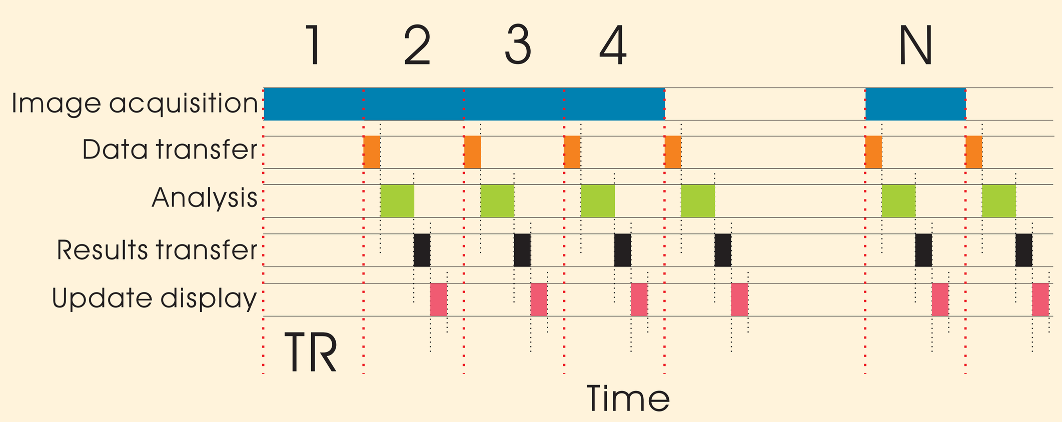
At present, we have interconnected several sites in Japan and the Philippines.



Flow of Real-Time Analysis

After the first image volume is acquired, it is immediately sent to the analysis server for processing. After the needed computation, the results are then sent back to the analysis workstation where the displayed activation map is then updated using the newest values. The whole process is repeated for the next volume until the last volume is processed.

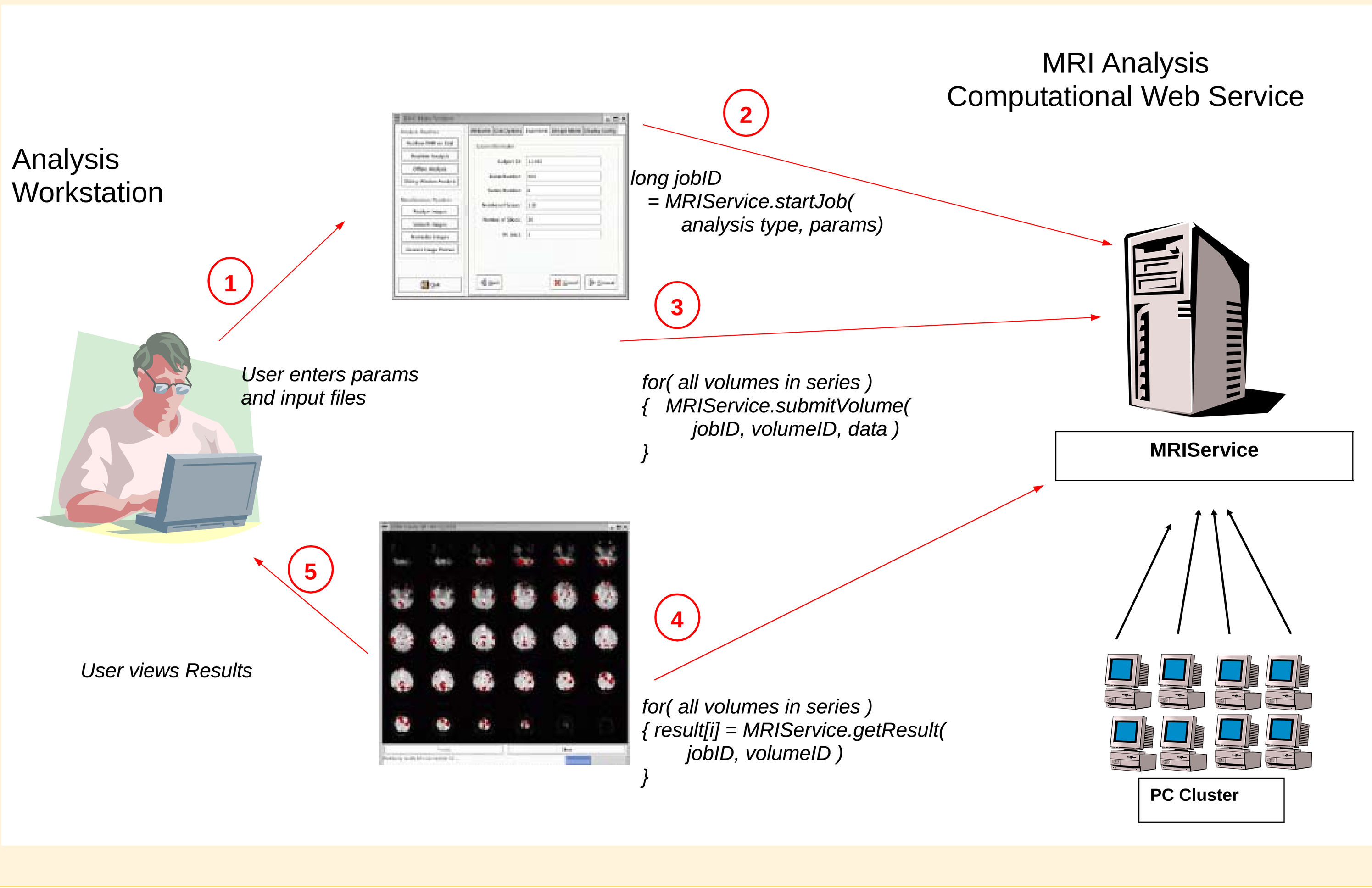




To achieve fully real-time operation, we must achieve a processing *throughput* of one new volume every TR seconds, where TR is the *repetition time* of the experiment. (Ideally, *total latency* for each volume should also be less than TR, but for a pipelined execution, it is sufficient to ensure that each of the 4 steps is less than TR.) In our experiments, TR = 3 s.

Computational Web Services

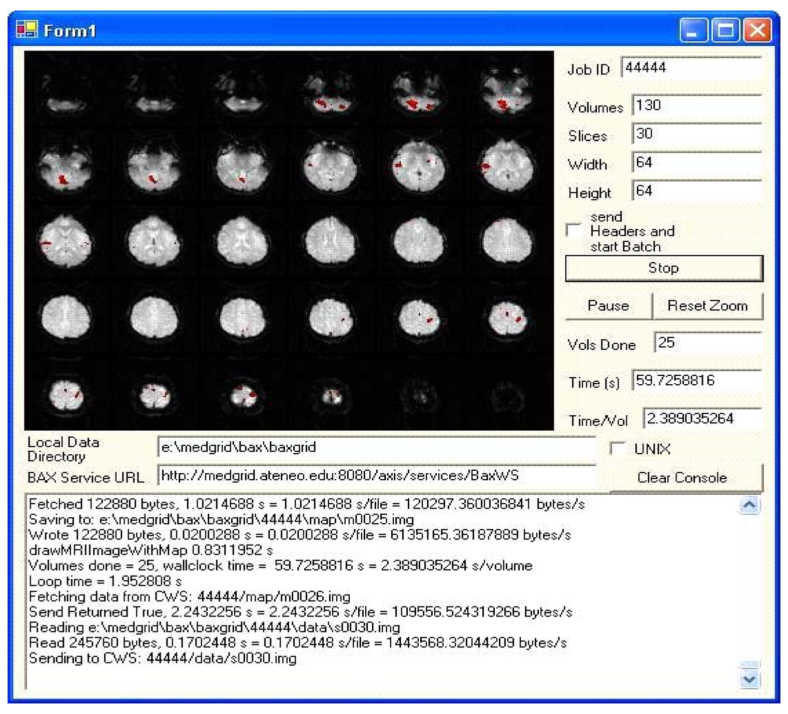
Computational XML Web Services make it possible to tap the computational grid resources such as the analysis servers through HTTP and XML-based protocols [2] This provides wide accessibility and interoperability, making it easier to access the services from different platforms and different programming languages, and across firewalls.



Implementation

Core System Software

The core system software was written in C, using MPI [5] for the parallel code running on the analysis servers. A GUI front-end was built using the gtk+ library [3], and the Globus v 2.0 APIs [4] for remote job submission and execution. A C program was written for transferring the data as binary streams through TCP/IP sockets. The GUI was designed such that the remote analysis is transparent to the user, as if all computations are done locally.



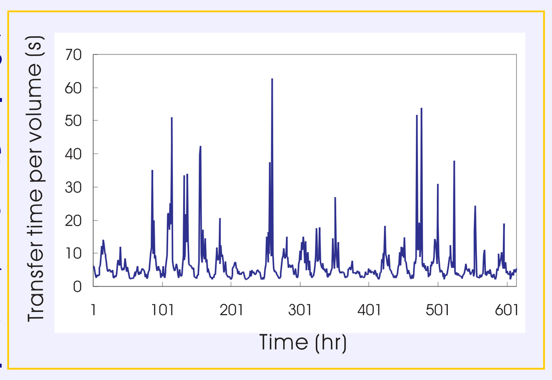
Computational Web Service-based Software

Using Java and Apache AXIS, we also implemented and deployed a prototype computational web service, running on the PC Cluster in the Philippines. The CWS made it easy for us to write new analysis workstation client software, including a text-based version written in Java, and a GUI-based version using Microsoft .NET (shown above).

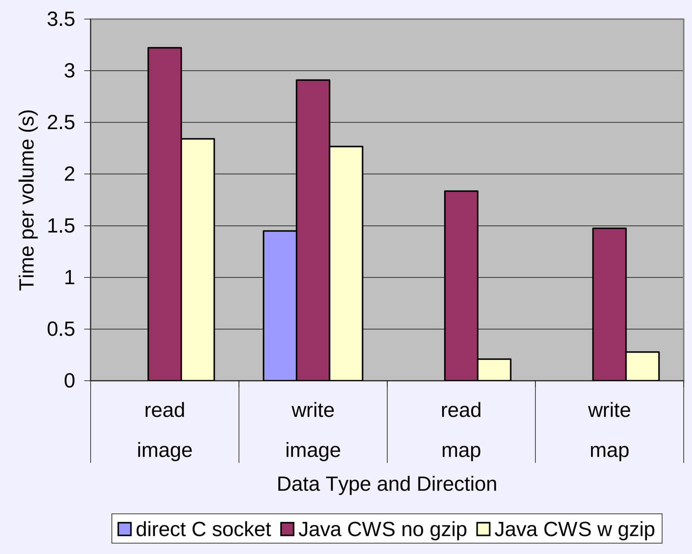
Results

Simulated real-time runs were performed to test the system. The estimated data transfer time from Osaka University (Site D) to the PC cluster in the AIST Grid Technology Research Center in Tsukuba (Site C) was about 0.215 s. The analysis time per volume using a 16-CPU PC cluster (1.4-GHz PIII processors, Gigabit Ethernet link) was about 0.745 s. The time to transfer back the results (activation maps) was about 0.094 s per volume and 0.035 s to update the local display. We therefore obtained a total analysis time per volume equal to 1.089 s, which is much less than the image acquisition time (TR) set to 3.0 s demonstrating fully real-time performance.

To test the feasibility of using grid computing for fMRI analysis across large distances, we measured data transfer times between Osaka (Site D) and the Philippines (Site E). Transfer time varied dramatically due to usage during school hours at Ateneo de Manila. However, the data transfer time during off-peak hours was less than 3 s, thus indicating the possibility of real-time operation.



Finally, we also compared the performance of using computational web services versus direct TCP/IP sockets. As expected, the web service-based version was slower due to the fact that it uses text-based XML as its encoding format. However, it was only 2x slower than the direct socket version. Moreover, using GZIP compression to compress the data before encoding it as XML reduces the slowdown to less than 2x (around 1.5x in this case), and allows us to get transfer times of less than 3s between Osaka and Manila during off-peak hours, again demonstrating the possibility of real-time performance even when using XML web services.



References:
[1] E. Bagarinao, K. Matsuo, T. Nakai and S. Sato, Estimation of general linear model coefficients for real-time application, *Neuroimage* 19 (2003) 422–429; E. Bagarinao, K. Matsuo, and T. Nakai, Real-Time Functional MRI Using a PC Cluster, *Concepts in Magnetic Resonance Part B* 19B(1) (2003) 14–25.
[2] L.F.G. Sarmenta, S.J.V. Chua, P. Echevarria, J.M. Mendoza, R. Santos, S. Tan, and R. Lozada. Bayanihan Computing .NET: Grid Computing with XML Web Services, in *Global and Peer-to-Peer Computing on Large Scale Distributed Systems Workshop (GP2PC), 2nd ACM/IEEE International Symposium on Cluster Computing and the Grid (CCGrid '02)*, Berlin, Germany, May 2002.
[3] http://www.gtk.org/ [4] http://www.globus.org/ [5] http://www-unix.mcs.anl.gov/mpi/