Vast Reduction in Reconstruction Time of 3D Time Resolved Inhomogeneity-Corrected Spiral MRA using Parallel Computing on a 32 Node Cluster

B. Kressler[1,2], P. Spincemaille[2], M. R. Prince[2], Y. Wang[1,2]

[1]Department of Biomedical Engineering, Cornell University, Ithaca, NY, United States, [2]Department of Radiology, Weill Medical College of Cornell University, New York, NY, United States

INTRODUCTION

In this work we demonstrate parallelizing a spiral reconstruction to run on a cluster constructed from standard commercially available components. Although spiral imaging is a technique that allows fast data acquisition in MRI, image reconstruction is fundamentally slower than in Cartesian scans due to regridding and inhomogeneity correction necessary to reduce blurring [1][2]. With multiple coil time resolved datasets, an inhomogeneity corrected reconstruction can take several hours. Parallelization of the reconstruction process results in drastic reductions in reconstuction time, enabling the use of spiral imaging in routine practice.

METHODS

A cluster was constructed with 32 compute nodes, a job submission/head node, and a disk server. All computers had dual 3.2GHz Pentium 4 Xeon EM64T processors and with 1MB L2 cache per processor and 2GB of RAM, and ran the Linux operating system. Networking was provided by two managed gigabit Ethernet switches interconnected via 6 load-balanced gigabit links. Data were transferred between the scanner and the cluster's disk server via an internal 100 megabit network. Message passing was implemented using the MPICH2 MPI library over the gigabit network. All code was compiled using gcc 3.4.3, and FFTs were performed using FFTW 3.0.1. Distribution of k-space and image files was performed using NFS to provide network mounted drives from the server. Topology of the cluster configuration is shown in Fig 1.

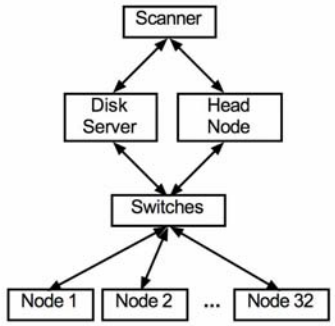

Fig 1. Cluster network topology

The ORC-VDS technique for conjugate phase inhomogeneity correction was implemented [3], then parallelized to run on the cluster. Parallelization was achieved by running each time frame on a separate processor. Each compute node ran two reconstruction processes, one per processor. Initially each process was assigned a frame to reconstruct. Once a process completed a frame, it was assigned the next available time frame to reconstruct, until all time frames had been completed.

The datasets reconstructed on the cluster were background subtracted time resolved spiral peripheral MRA cases (n=7) with 24 interleaves, a 256x256 matrix regridded to a 512x512 matrix, 24 slices, and a 4 channel coil. A separate inhomogeneity map was generated for each time frame to account for time varying off-resonance effects, and 11 frequency bins were used for frequency segmented conjugate phase correction [4]. Sliding window reconstruction [5] was used to produce 115 time frames from an acquisition lasting 2.6 minutes. Upon completion of the scan, data was automatically copied to the cluster, reconstructed, and maximum intensity projection images were inserted into the image database.

RESULTS

A typical resulting image is shown in Fig 2 both with and without inhomogeneity correction. The inhomogeneity correction reduces blurring in the arteries, especially near the edges of the FOV. Without use of the cluster, a reconstruction with inhomogeneity correction took approximately 80 minutes. Using the cluster, a reconstruction took 1.8 minutes. The entire process, including copying data to and from the cluster took approximately 3.2 minutes. Running times for just the reconstruction portion of the data processing chain are shown in Fig 3. These times are for a scan with 115 reconstructed temporal volumes.

DISCUSSION

In Fig 3, running time decreases in steps as the number of reconstruction processes used increases. This stepping is expected because each process constructs an entire temporal phase, and the reconstruction finishes once the last process has finished its last temporal phase. Therefore, significant jumps in running time occur only when number of phases is a multiple of the number of processes. Two conditions were tested, one in which each node ran a single reconstruction process for a total of 32 processes, and one in which each node ran two reconstruction processes (one per processor) for a total of 64 processes. The curve depicting two processes per node corresponds closely but not exactly to the curve for one process per node, indicating that contention for memory access between the two processors in a node is a limiting factor. Overall, the reconstruction system scales quite well from one process per node to two processes per node, and with the total number of processes used.

In conclusion, the use of parallel computing with a cluster can drastically reduce the time required to reconstruct large spiral datasets. Future work may include improving the granularity of work division among nodes and implementing real time retrieval of data from scanner memory, allowing near instantaneous reconstruction. It is hoped that these improvements will allow reconstruction to finish within 30s of completion of data acquisition.

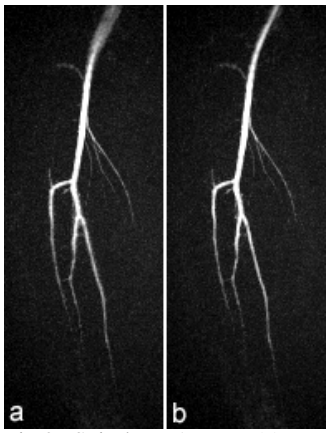

Fig 2. Spiral MRA scans (a) without and (b) with inhomogeneity correction. Note the marked improvement vessel sharpness.

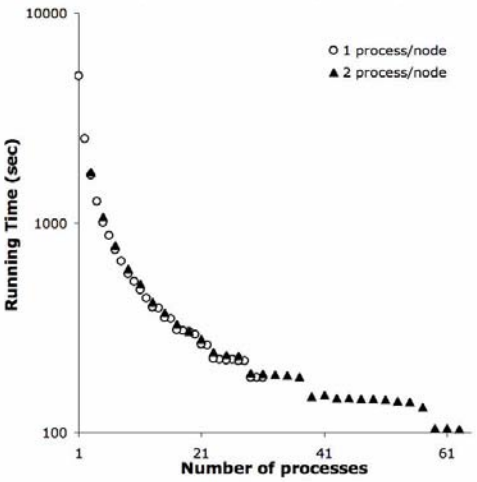

Fig 3. Running time versus number of reconstruction processes for 1 or 2 processes per node

REFERENCES

[1] Zhu H, et al. MRM: 52:14-18, 2004. [2] Börnert P, et al. MRM:172-175, 2001. [3] Nayak KS, et al. MRM 45:521-524, 2001. [4] Noll DC. Ph.D Thesis, Stanford University, 1991. [5] Riederer SJ, et al. MRM 8:1-15, 1998.