A Parallel Computing Solution for Rapid Reconstruction of Highly-Accelerated Volumetric Parallel MRI Data

S. Cohen[1], A. K. Grant[1], E. N. Yeh[2], S. Joshi[3], D. K. Sodickson[1,2]

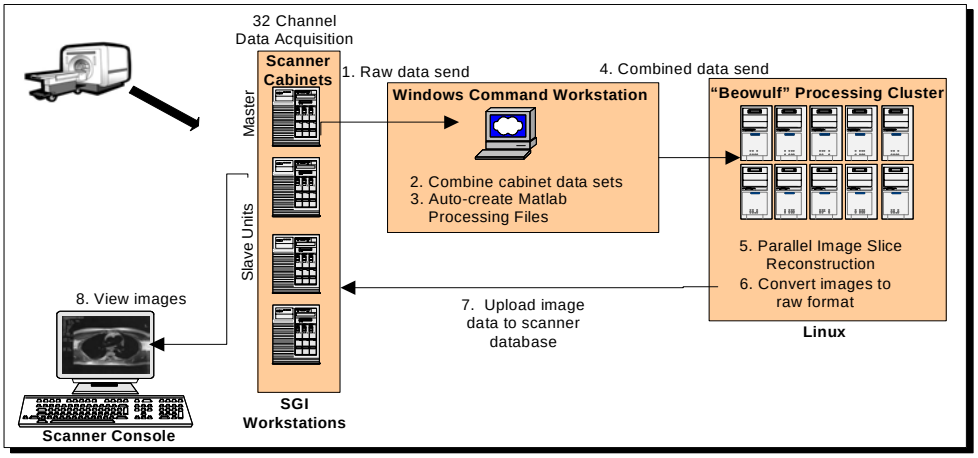
[1]Beth Israel Deaconess Medical Center, Boston, MA, United States, [2]Harvard-MIT Division of Health Sciences and Technology, Boston, MA, United States, [3]General Electric Medical Systems, Milwaukee, WI, United States

Introduction: The use of large coil arrays for highly parallel volumetric imaging [1] creates very large data sets that stress the capability of current off-line reconstruction algorithms and application development tools. The typical single CPU implementation of parallel image reconstruction algorithms on an off-line workstation results in long time delays before an image set can be presented to clinicians and/or research scientists so that they may refine their studies. Our goal in this work was to reduce the image reconstruction turn-around time for acquired images by use of a low cost cluster of computers, so that the processed images can be delivered to the clinical workstation before the patient leaves the MRI magnet, while maintaining the use of high-level software (MATLAB) to facilitate the rapid prototyping of new applications and algorithms.

Problem: Parallel MR imaging using the GE 32-channel system at Beth Israel Deaconess Medical Center generates large data sets. For example, a current volumetric data acquisition of 32 component coil signals at a modest 3D matrix size of 128 x 128 x 128 generates a combined signal data file that is 262 megabytes in length, with an accompanying sensitivity calibration file containing as much as an additional 64 megabytes of data. This size of course decreases with increasing acceleration, but increases again if multiple dynamic data sets are acquired. Given the large size of these data sets, and given that all processing has occurred offline in our first 32-channel implementations, there has generally been no possibility for the study to be modified based on the outcome of the reconstruction while the patient is still available. Anticipated future growth in the number of receiver channels will only exacerbate the problem, even for commercial inline reconstruction hardware.

Theory: In this work, parallel computing was used to increase the system throughput. Even irregularly-sampled two-dimensionally-accelerated parallel MR data sets have a natural division boundary in the unaccelerated readout direction. If a Fourier transformation is performed along this direction, the resulting data "slices" are independent from one another, and this fact can be exploited by restructuring the normal sequential image reconstruction problem into a parallel one, with separate CPU's dedicated in turn to individual readout-direction slices.

System Description: The data acquisition portion of the 32-channel research scanner is comprised of four 8-channel receiver cabinets networked together. There is an SGI Unix host computer connected to each receiver cabinet. One host is designated the "master" and the other three systems are "slave" units, obtaining their timing/synchronization information from the master unit. Signal data are acquired simultaneously in the four 8-channel cabinets. Upon completion of the data acquisition, Unix shell scripts on the master host gather the raw data files from the slave hosts and post them via a 100 megabit Ethernet link to a Windows workstation to begin the processing. A control application on the workstation, written in C#, updates records in a processing database and automatically generates Matlab™ processing scripts for the job according to the type of the reconstruction desired (SMASH, SENSE, GEM, etc.). The four individual raw data files are then combined into one large raw data file and a Fourier transform is performed along the readout direction. The raw data and Matlab scripts are sent across a gigabit Ethernet LAN to a 10-node Linux "Beowulf" processing cluster made up of inexpensive commodity servers and freely available clustering software. A parallelized MPI version of Matlab [2] is used to simultaneously process images in a "round-robin" fashion. The processed image data is re-combined on the master node of the Linux cluster and sent back to one of the SGI workstations where standard GE post-processing algorithms are run before the final image is posted back into the scanner database, enabling the clinician or researcher to view the reconstructed images.



Results and Discussion

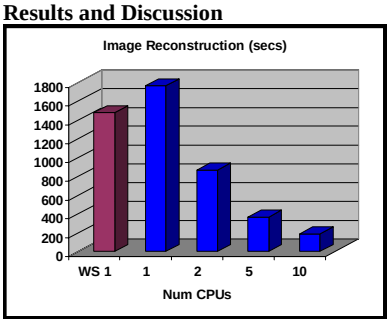


Figure 2 shows the processing speedup achieved through readout-direction parallel processing for a GEM [3] reconstruction of a typical volumetric dataset with sixteen-fold acceleration. The column labeled WS1 indicates the timing with the reconstruction performed entirely on the original fast workstation with a single CPU; the remaining columns show the results with the reconstruction ported to the cluster, with a varying number of slower individual nodes. The results show that significant improvements in processing speeds can be achieved through inexpensive commodity hardware (our cluster cost less than $5000 USD to build), returning results in an acceptable timeframe, while at the same time utilizing a high-level language familiar to researchers to enable algorithm development. We must note that as more receiver coils are added in the future to enable further acceleration, more sophisticated techniques to parallelize the image reconstruction process (beyond simple slice partitioning) will become necessary as the data sets continue to grow. Future enhancements will add additional processing units into the cluster, as well as new parallel intra-slice sub-block parallelization algorithms [3,4], in order to reduce the round-trip time to under a minute. This will enable real-time modifications to the parallel MR imaging protocols and reconstruction algorithms for both clinical and research applications.

[1] Y Zhu et al., ISMRM 2003, 22 [2] R Choy and A Edelman, Proc. of the IEEE Special Issue on Program Generation, Optimization, and Platform Adaptation 2005 [3] DK Sodickson et al. Med Phys 2001;28(8):1629-43 [4] EN Yeh et al, ISMRM 2002, 2399