Real-Time High-Throughput Scalable MRI Reconstruction via Cluster Computing

E. A. Borisch[1], R. C. Grimm[1], P. J. Rossman[1], C. R. Haider[1], and S. J. Riederer[1]
[1]Department of Radiology, Mayo Clinic, Rochester, MN, United States

INTRODUCTION

Real-time MR imaging has been an area of active research for almost twenty years**,** with applications including interactive view control, fluoroscopic triggering, and cardiac fluoroscopy. While these applications originally focused on real-time imaging in two spatial dimensions, significant advances in parallel imaging techniques using multiple receive channels, such as SENSE and GRAPPA, introduce the possibility for real-time 3D MR imaging. In addition to the data sampling rates increasing significantly over this time, the total amount of raw data created also grows linearly with the number of receive channels in use. Compounding this increase in the rate and amount of data creation, parallel imaging techniques require additional processing steps to estimate the original image-space data, and often generate larger output image resolutions than the input k-space data sets. The desire to have real-time high-resolution three-dimensional reconstructions, enabled by accelerated acquisition techniques applied in multiple dimensions and received through multiple coils, places considerable demand on the communication and computational resources of a reconstruction system. The goal of this work is to provide a scalable reconstruction system capable of meeting these challenges, delivering real-time 3D reconstructions of accelerated (in multiple dimensions) parallel acquisitions.

METHODS

Hardware: To achieve the desired reconstruction performance, throughput, in both the communication and the computational sense, is paramount. To this end, we have implemented a scalable cluster-based parallel reconstruction system. Our current system consists of eight physical machines or nodes. Each node has two 3.4GHz Intel Pentium 4 Xeon processors, 4GB of RAM, and an 80GB drive, providing a total of 16 CPUs and 32GB of RAM. Within the cluster, three network connections are made with each node: one 1-gigabit per second (Gb/s) Ethernet connection and two 8Gb/s Infiniband connections. This private (local to the cluster) Ethernet carries network file system traffic and enables system administration tasks. The high-bandwidth ($2 \times 8 = 16\text{Gb/s}$) low-latency ($4\mu s$) Infiniband network is used for communication between the reconstruction processes (programs) running in parallel on the nodes of the cluster.

MRI System Connection: Data is provided to the reconstruction system during scans via an 8Gb/s optical Infiniband connection from the MRI system, while feedback control messages are sent to the MRI system over a dedicated 1Gb/s Ethernet connection. Both the scan data and feedback paths are controlled by a custom multi-threaded management class, which in turn incorporates vendor-provided (GE Raw Data Server) connection management software. The current 8Gb/s communication channel from the MRI system imposes a communication overhead, defined here as the percentage of one TR consumed transmitting one TR's worth of data, for a (as an example) 256 frequency encode, 8-channel, 2ms TR acquisition of under 1%. We have also developed tools enabling the cluster to perform non-real-time yet computationally intensive reconstructions for other MRI systems both on- and off-site.

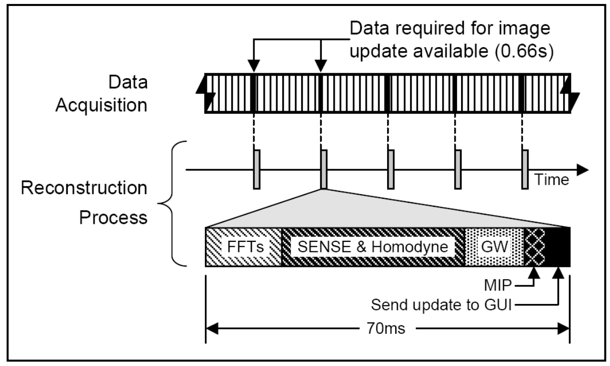

Figure 1: Acquisition and reconstruction timing for $256\times128\times32$ four-fold view-shared acquisition with R=4 2D SENSE (2×2) and gradient-warping correction with 0.66s image update rate. The completed image is available less than 70ms after the requisite data has been acquired.

Software: The distributed reconstruction routines executed on the cluster are implemented in C++ and use the Message Passing Interface **[1]** (MPI) model for intra-node and inter-node communication, and optionally OpenMP **[2]** for intra-node parallelism. The MVAPICH **[3]** library provides the Infiniband-based MPI implementation. An MRI-centric generic (templated) C++ class has been developed for management of distributed n-dimensional data arrays. This enables rapid development of parallel/distributed reconstruction algorithms, as the researcher is free to operate on the distributed data without concern for the vast majority of the communication and synchronization steps required by MPI (distributed) programming. In addition, many common MRI reconstruction operations (MIPs, targeted MIPs, FFTs, chops, shifts, rotations, etc.) are built into the class to further accelerate reconstruction software development. To shield the distributed reconstruction process from servicing the graphical user interface, the GUI is executed on a separate system located at the scan control console, and user input and reconstruction results are transported over an additional 1Gb/s Ethernet network. Similar to the MRI system control management class mentioned earlier, custom network management classes have been developed to facilitate communication between the GUI and the reconstruction software. The cluster is running the Rocks 4.2.1 **[4]** distribution of the Linux operating system.

RESULTS

Real-Time 3D MRI Reconstruction: We have successfully performed real-time reconstruction and display of 3D time-resolved data with an eight channel 3.0T (14.0 GE Medical Systems, Waukesha, WI) system. Data was acquired at an aliased resolution of $256\times64\times16$, with four-fold view sharing updates occurring every 0.66 seconds. During each update, FFTs, 2D SENSE unfolding, as well as 2D homodyne and gradient warp correction, as shown in Figure 1, were performed. After R=4 2D SENSE ($2\times$ in phase and $2\times$ in slice) unfolding, the reconstructed resolution at each view-sharing update was $256\times128\times32$. A MIP of this reconstructed volume was generated and sent to the GUI for display ~70ms after the completion of data acquisition for that timeframe. For a 3D time-resolved acquisition with higher acceleration (R=7.3 2D SENSE) and higher resolution ($400\times320\times132$ after unfolding) the full reconstruction time was ~800ms. In addition, the previously mentioned tools for utilizing the cluster remotely have been successfully used for reconstructions at multiple locations on- and off-site.

Computation and Communication Performance: The theoretical maximum communication throughput of the system as designed is 128Gb/s with four pairs of nodes, each exchanging data at $2\times16 = 32\text{Gb/s}$. In practice, we have measured system bandwidths in excess of 80Gb/s, or greater than a DVD's worth of data transmitted per second. The theoretical maximum HPLinpack **[5]** performance of the hardware is 105GFlops (105 billion double-precision floating point operations per second.) In practice, we have measured HPLinpack performance in excess of 85GFlops. More directly applicable to MRI, a distributed $256\times256\times256$ (single-precision) 3D FFT can be performed in 140ms.

CONCLUSIONS

A cluster-based high-throughput reconstruction system and associated MRI-centric distributed programming tools have been successfully designed and implemented to enable accelerated acquisition techniques and multiple-coil receive systems to be combined with the benefits of real-time reconstruction. The design of the cluster and software is such that future expansion, when needed to support higher coil counts, is not disruptive – the sixth fastest supercomputer in the world uses a very similar hardware and software platform. **[6]** Future work will enable user or programmatic modification of intra-scan acquisition parameters in real-time.

[1] http://www-unix.mcs.anl.gov/mpi/ **[2]** http://www.openmp.org/ **[3]** http://mvapich.cse.ohio-state.edu/ **[4]** http://www.rocksclusters.org/
[5] http://www.netlib.org/benchmark/hpl/ **[6]** http://www.cs.sandia.gov/platforms/Thunderbird.html