

LLgrid: Enabling On-Demand Grid Computing With gridMatlab and pMatlab

*Albert Reuther, Tim Currie, Jeremy Kepner, Hahn G. Kim,
Andrew McCabe, Michael P. Moore, and Nadya Travinin*

MIT Lincoln Laboratory, Lexington, MA 02420

Phone: 781-981-5699

Email Addresses: {reuther, currie, kepner, hgk, amccabe, moore, ntj@ll.mit.edu}

May 28, 2004

Introduction

The concept of grid computing – back-room computational resources that are as accessible and available as the electric grid – has gained momentum [1]. Numerous Grid computing projects such as NetSolve [2] and Legion [3] have provided infrastructure to enable the launching and monitoring of mostly parameter sweep applications. These Grid computing projects demand that users endure a steep learning curve to program and use the system. Also, these systems draw a strong distinction between the users' computers and the grid computing resources: users' work is done on their computers, while grid jobs are executed on grid resources. A computational power grid should have characteristics similar to the electric power grids: always available, ubiquitous throughout the organization, and easy to use.

The goal of the MIT Lincoln Laboratory Grid (LLgrid) project is to develop a On-Demand Grid Computing capability to address these characteristics and use MATLAB® – the dominant programming language for implementing numerical computations, widely used for algorithm development, simulation, data reduction, testing, and system evaluation – as its initial target application [4]. MIT Lincoln Laboratory has over one thousand MATLAB users; nearly two hundred users run very long jobs that could benefit from parallel processing. The LLgrid project has developed three technologies that allow these users to run parallel MATLAB jobs transparently on the LLgrid computational resources:

- *MatlabMPI* for point-to-point messaging;
- *pMatlab* for global array semantics (similar to High Performance Fortran); and
- *gridMatlab* for integrating user's computers into the LLgrid and automatically allocating grid computing resources.

These technologies have combined to create a unique on-demand, interactive Grid Computing experience, whereby running a parallel MATLAB job on LLgrid is identical to running MATLAB on the desktop. Users can use LLgrid from Windows, Linux, Solaris, and Mac OS X computers

with their desktop computer becoming a personal node in the LLgrid thereby establishing a transparent interface between the user's computer and the grid resources. LLgrid is enabling faster algorithm development, prototyping, and validation cycles for Lincoln staff. In addition, the initial creation and setup of user accounts is entirely automated, which minimizes system administration.

The LLgrid System

A number of components comprise the LLgrid On-Demand Grid Computing system, from the underlying hardware and network to MATLAB and the three Lincoln-developed MATLAB toolboxes: pMatlab, MatlabMPI, and gridMatlab. Figure 1 shows a conceptual depiction of the initial operational system including the LLgrid Alpha Cluster of Red Hat Linux nodes, the network configuration, the users' computers on the central Lincoln local area network (LLAN), and the gridsan network storage, which delivers a common file system to the LLgrid cluster and all of the users.

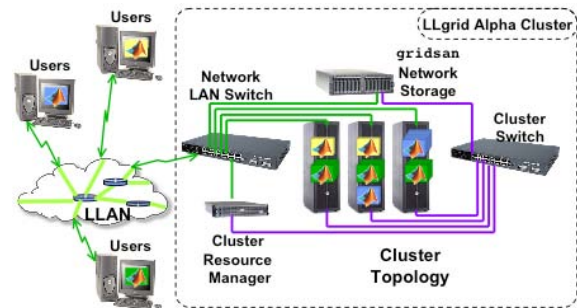


Figure 1: Conceptual diagram of LLgrid System.

The core of the users' experience with the LLgrid system occurs with MATLAB and the pMatlab toolbox, while the MatlabMPI toolbox provides the means for parallel MATLAB processes to communicate with each other and the gridMatlab toolbox provides the interface between MatlabMPI and the LLgrid resources for managing jobs. MatlabMPI [5] consists of a set of MATLAB scripts that implement a subset of MPI, allowing any MATLAB program to be run on a parallel computer. The key innovation of MatlabMPI is that it implements the widely used MPI "look and feel" on top of standard MATLAB file I/O, resulting in a "pure" MATLAB implementation that is exceedingly small (~300 lines of code). Thus, MatlabMPI will run on any combination of computers that MATLAB

This work is sponsored by the United States Air Force under Air Force contract F19628-00-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

supports. Overlaid on MatlabMPI, pMatlab combines operator overloading with parallel maps to provide implicit data parallelism and task parallelism [6]. pMatlab allows a MATLAB user to parallelize their program by changing a few lines.

The gridMatlab toolbox transparently integrates the MATLAB on each user’s desktop with shared grid clusters through a cluster resource manager; when a MatlabMPI or pMatlab job is run by the user in his or her MATLAB session, gridMatlab automatically amasses the requested LLgrid computational resources from the shared grid resources to process in parallel with the user’s MATLAB session. (In traditional grid computing the users must submit their jobs to batch job queues where the jobs execution must wait to be executed.) Using MatlabMPI, the underlying common file system (hosted on gridsan) becomes the communication fabric through which each of the parallel MATLAB processes communicate, including the MATLAB on each user’s desk. By integrating the user’s MATLAB session into the set of grid cluster MATLAB sessions working on his or her code, the user receives immediate feedback on the status of his or her job, thereby making parallel MATLAB execution virtually identical to running MATLAB on a single computer.

Results

As with any distributed, parallel computing system, it is expected to deliver high performance; the LLgrid system meets those expectations. For example, for a hyper-spectral imaging (HSI) analysis application [7], LLgrid is used to classify various characteristics in HSI signal returns using a normal compositional model (NCM). The application is composed of three components: abundance estimates of class at each pixel (up_abund), NCM class parameter updates (up_NCM), and NCM log-likelihood computations for given class abundance values (up_mll). Figure 2 shows the speedup of these three application components, each realizing near linear speedup.

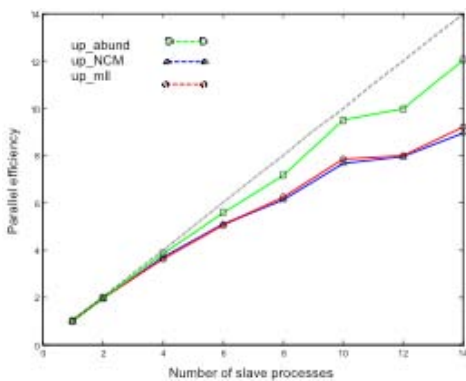


Figure 2: Speedup realized by Hyperspectral Analysis application

When it comes to modeling productivity, we turned to the Darpa IPTO High Productivity Computing Systems (HPCS) program. One of the challenges in the HPCS mission is defining, modeling, and measuring the productivity that a computer system delivers to its users and consequently to the users’ organization. To answer this challenge, the HPCS

Productivity Team has developed a high performance productivity framework and evaluation model [8]. Their model implies a profoundly different way of viewing HPC systems by including the user-associated costs in the model and by viewing innovative hardware as a key aspect to lowering the very high cost of high performance computing software.

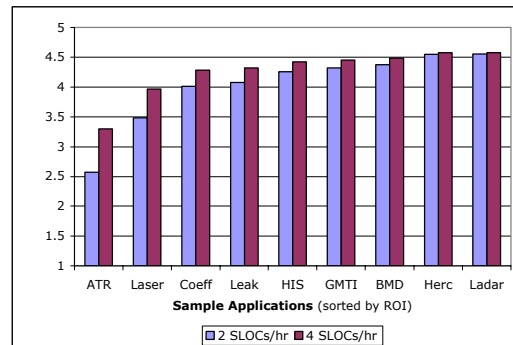


Figure 3: Range of Return of Investment for sample LLgrid applications.

Using the HPCS productivity model, Figure 3 plots two return on investment (ROI) values for the LLgrid system and nine sample applications that are currently being run on the LLgrid system, it shows a potential upper and lower bounds by calculating the ROI for average programming rates of 2 and 4 SLOCs per hour. The graph shows that the ROI range is between 2.6 and 4.6 across the nine sample applications and average programming rates, which means that for every \$1.00 spent on the LLgrid, between \$2.60 and \$4.60 is returned in user time saved.

References

- [1] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Second Edition, Morgan-Kaufman, 2004.
- [2] S. Agrawal, J. Dongarra, K. Seymour, and S. Vadhiyar, *Grid Computing – Making the Global Infrastructure a Reality*, F. Berman, G. Fox, and T. Hey (eds.), John Wiley & Sons, Ltd., 2002.
- [3] A. S. Grimshaw and W. A. Wulf, “The Legion vision of a worldwide virtual computer,” *Communications of the ACM*, Vol. 40, No. 1, Jan. 1997.
- [4] MATLAB, The MathWorks, Inc., <http://www.MathWorks.com/products/matlab/>.
- [5] J. Kepner, “Parallel programming with MatlabMPI,” *High Performance Embedded Computing (HPEC) Workshop 2001*, <http://www.ll.mit.edu/hpec/>, 2001.
- [6] J. Kepner and N. Travinin, “Parallel MATLAB: The next generation,” *High Performance Embedded Computing (HPEC) Workshop 2003*, <http://www.ll.mit.edu/hpec/>, 2003.
- [7] D. Stein, “A parallel implementation of the normal compositional model for hyperspectral analysis based on MatlabMPI,” *High Performance Embedded Computing (HPEC) Workshop 2001*, <http://www.ll.mit.edu/hpec/>, 2001.
- [8] J. Kepner, “HPC productivity model synthesis,” accepted for publication in the *International Journal of High Performance Computing Applications*, Vol. 18, No. 4, November 2004.