

DR&E LLGrid Portal: Interactive Supercomputing for DoD

Albert Reuther, William Arcand, Chansup Byun, Bill Bergeron, Matthew Hubbell, Jeremy Kepner, Andrew McCabe, Peter Michaleas, Julie Mullen, and Andrew Prout
{reuther, warcand, cbyun, bbergeron, mhubbell, kepner, amccabe, pmichaleas, jsm, aprout}@ll.mit.edu
MIT Lincoln Laboratory, Lexington, MA 02420

May 20, 2010

Introduction

High performance computing (HPC) uses supercomputers and computer clusters to solve large computational problems. To take advantage of HPC systems, users in general need training in parallel computing techniques. This is not particularly daunting for computational scientists. Others, who could well take advantage of HPC, prefer not to be distracted by needing to learn parallel computing techniques; they continue to focus instead on their particular fields of interest, thus leaving untapped the power of HPC for their work. This project is designed to overcome these traditional obstacles so that more DoD scientists and engineers can take advantage of the HPC resources available to them at the HPCMP's (DoD HPC Modernization Program) DoD Supercomputing Resource Centers (DSRCs) [1]. The goal of the DR&E Portal project is to make HPC resources easily accessible to all DoD scientists and engineers regardless of their computational background. The interactive, on-demand LLGrid computing system [2,3,4] at MIT Lincoln Laboratory represents a capability that has many elements of the objective DR&E Portal. The LLGrid system has radically changed how many Lincoln Laboratory project teams approach the development of algorithms and simulations. As part of the Defense Research and Engineering (DR&E) Portal program, Lincoln Laboratory in partnership with ARL is creating testbeds to demonstrate the feasibility of this capability. The intent of the Portal interface is to provide the look and feel of running applications on the desktop when in reality they are running on a supercomputer. This paper presents an overview of LLGrid, the DDR&E Portal goals, and how LLGrid has been adapted to meet these goals.

Lincoln Laboratory Grid (LLGrid)

One of the main design requirements of the LLGrid system was to make HPC usage accessible to the entire Lincoln technical staff by making use of HPC systems as easy as using a personal computer. In striving toward this goal, we developed a system that takes advantage of the desktop development environments (i.e., file system access and integrated development environments like Matlab) with which most engineers and scientists are already familiar. At Lincoln we currently have over 400 users, which is

This work is sponsored by the United States Air Force under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

greater than 20% of the entire technical staff. The vast majority of these users utilize LLGrid in an interactive supercomputing manner. As depicted in Figure 1, interactive supercomputing involves large jobs that require answers in minutes or hours and cannot tolerate waiting in a queue.

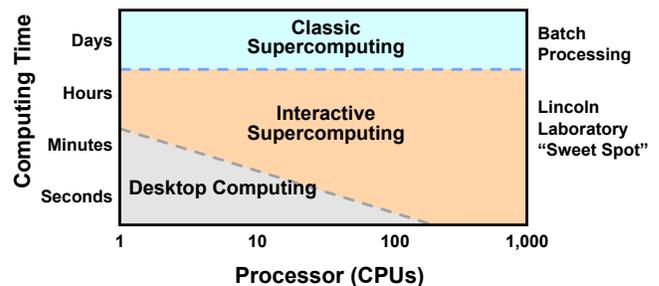


Figure 1: Interactive supercomputing vs. classic supercomputing and desktop computing. Interactive supercomputing jobs require answers in minutes or hours but cannot tolerate waiting in a queue. Classic supercomputing jobs take hours to days to execute and can tolerate waiting in a queue. Desktop computing jobs take minutes to run on a desktop (e.g. algorithm proof-of-concept).

DR&E LLGrid Portal Prototype

Several considerations must be made in order to adapt the LLGrid system to DoD supercomputing resources. These considerations are mainly concerned with security:

- DoD systems require authentication with common access card (CAC) credentials;
- Mounting network file systems via NFS, CIFS, etc. across organizations has historically been difficult;
- DoD networks only allow certain network ports to pass through routers and this varies by organization;
- The number of additional computer applications installed on the desktop should be minimized; and
- The applications that users can execute on the HPC system and file access of the HPC file system should be limited.

With these security requirements in mind, we have developed and/or implemented several unique LLGrid components that enable the portal technology:

- CAC-enabled Apache WebDAV server,
- Linux file system watcher,
- gridMatlab for Portal, and
- grecurity kernel patches.

The next paragraphs will describe each of these technologies.

Web-based Distributed Authoring and Versioning (WebDAV) is a set of web protocol extensions that enable remote file system access over the SSL web port 443. All modern operating systems including Windows 7, Linux, and Mac OS X provide WebDAV support natively by enabling WebDAV shares to be mounted as a network file system. The Apache web server can be configured to provide WebDAV access, but it did not include CAC authentication. We added CAC authentication to the Apache web server so that anyone with a CAC card can authenticate and mount the prototype LLGrid file system onto their computer via WebDAV.

We needed to avoid installing and enabling secure shell (ssh) on each of the user desktops (especially the Windows desktops), but we needed a way to get HPC system status, launch jobs, abort jobs, and get job status [4] without making a remote procedure call. By modifying the Apache WebDAV server, we designed and implemented a Linux file system watcher. The watcher monitors all file system activities and executes actions depending on the activity and filename. We have defined filename patterns and XML file contents for getting HPC system status, launching jobs, aborting jobs, and getting job status. When a matching filename appear in the file system, the watcher executes scheduler scripts and returns a response XML file into the same folder as the activity file.

The third part of this effort was to modify gridMatlab to write and read the aforementioned activity and response XML files. The gridMatlab toolbox enables desktop Matlab environments to interact with HPC schedulers to launch and abort parallel Matlab jobs, get HPC system status, and get job status.

The above technologies limit users to only execute the applications for which XML-file based launching is enabled. To further secure access to HPC system resources, we have implemented the grsecurity kernel patches [6]. Grsecurity provides white list role-based access control (RBAC) for all operating resources including file system access, process tables, process status (/proc), stack memory, socket access, etc. By white listing access to all resources on the HPC system, we have highly granular resource access control for each user and have monitoring capability that can indicate user errors and system exploitation. With these technologies, we have enabled the LLGrid system on a prototype DSRC, and we are capable of deploying HPC systems up to a protection level 2 (PL2), which enables access for multiple programs with different need-to-know levels at the same classification level on the same system.

Results

We have built a prototype Portal cluster at Lincoln Laboratory and are currently testing the capability on DREN with pre-alpha DoD users. Also, we have benchmarked two MatlabMPI/pMatlab [5] applications on the TX-2500 LLGrid system at Lincoln Laboratory that are typical applications for DoD engineers and scientists. The scaling results are shown in Figure 2. Application 1 (App1) is a smaller application that we scaled up to 36 processors,

and it scaled well up to that number. Application 2 (App2) is an application that we scaled up to 200 processors, and it scaled superlinearly up to that number of processors. This showed that for some typical DoD applications, there is promise for them to scale well on an LLGrid interactive HPC system.

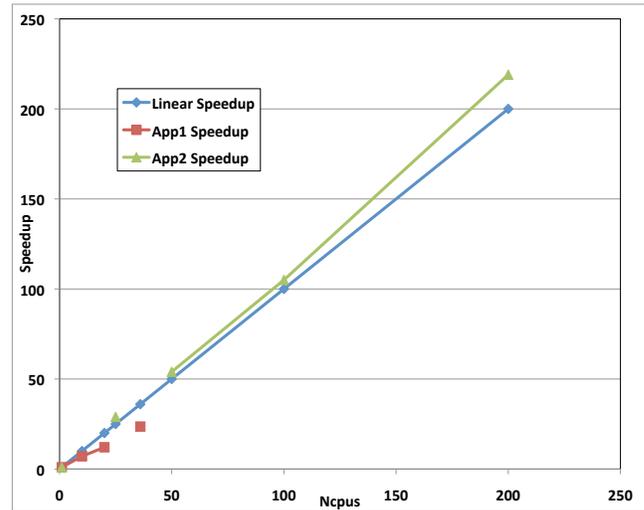


Figure 2: The speedup of two typical DoD pMatlab/MatlabMPI applications on the LLGrid TX-2500 system.

References

- [1] DoD Supercomputing Resource Centers, <http://www.hpcmo.hpc.mil/cms2/index.php/hpccenters>.
- [2] N. Travinin Bliss, R. Bond, J. Kepner, H. Kim, and A. Reuther, “[Interactive Grid Computing at Lincoln Laboratory](#),” Lincoln Laboratory Journal, Vol. 16, Number 1, 2006.
- [3] A. Reuther, B. Arcand, T. Currie, A. Funk, J. Kepner, M. Hubbell, A. McCabe, P. Michaleas, “[TX-2500 – An Interactive, On-Demand Rapid-Prototyping HPC System](#),” HPEC 2007, Lexington, MA, Sep. 2009.
- [4] A. Reuther, J. Kepner, A. McCabe, J. Mullen, N.T. Bliss, and H. Kim, “[Technical Challenges of Supporting Interactive HPC](#),” In Proceedings of the High Performance Computing Modernization Office (HPCMO) Users Group Conference (UGC) 2007, Pittsburgh, PA, 18-22 June 2007.
- [5] J. Kepner and N. Travinin, “[Parallel Matlab: The Next Generation](#),” HPEC 2003, Lexington, MA, Sept. 2003.
- [6] grsecurity, <http://www.grsecurity.org/>.