StarHPC - Teaching Parallel Programming within Elastic Compute Cloud

Ceraj Ivica*, Justin T. Riley*, Charles Shubert
(* equally contributing authors)
Massachusetts Institute of Technology, Cambridge, Massachusetts, USA
ceraj@mit.edu, jtriley@mit.edu, cshubert@mit.edu

Abstract. *The advancement of computer technology and the increasing complexity of research problems are creating the need to teach parallel programming in higher education more effectively. In this paper we present StarHPC, a system solution that supports teaching parallel programming in courses at the Massachusetts Institute of Techology. StarHPC prepackages a virtual machine image used by students, the scripts used by an administrator, and a virtual image of the Amazon Elastic Computing Cloud (EC2) machine used to build the cluster shared by the class. This architecture coupled with the no-cost availability of StarHPC allows it to be deployed at other institutions interested in teaching parallel programming with a dedicated compute cluster without incurring large upfront or ongoing costs.*

Keywords. Teaching, parallel programming, MPI, Amazon EC2, Eclipse parallel tools, Sun Studio, HPC, STAR project, StarHPC

1. Introduction

The rapid expansion of research data creates a need to develop tools that facilitate the understanding and analysis of data. Yesterday's tools were designed for a different scale of problems and are unable to cope with this expansion despite the advancement of available computational resources. Novel analysis tools designed with parallelism and scalability in mind will have a significant impact on how contemporary research is being conducted. To facilitate these needs, there is a desire in the educational community to help develop students with skills that allow them to tackle tomorrow's problems [1]. The Software Tools for Academics and Researchers (STAR) group of the Office of Educational Innovation and Technology (OEIT) at the Massachusetts Institute of Technology is creating practical solutions that fulfill this need. The STAR program seeks to address these needs by making software easier to use, easier to access, scalable for peak-period classroom usage (such as the night before the homework is due), and to have a sustainable support infrastructure that extends beyond the authors and beyond our institution.

One of the areas we concentrate on is making computational resources available to students in the classroom. A significant percentage of hands-on time in the classroom in our initial pilot project was spent setting up the software and system environment by getting UNIX command line options correct, modifying shell scripts, and navigating arcane user interfaces. It became clear to us that research software in the classroom needed to be easier to use. In responding to this need we developed StarHPC [2], in collaboration with MIT Earth, Atmospheric, and Planetary Sciences (EAPS) Research Scientist Constantinos Evangelinos. StarHPC streamlines development and running parallel programs on high performance computing clusters, thus allowing faculty to focus on teaching the concepts of parallel programming and algorithms.

2. StarHPC is a cost-effective way to bringing computational resources into the Classroom

In general, owning and managing computational resources is costly and involves a lot of overhead. In addition to hardware costs, there are costs associated with housing, powering, cooling, and administering a computing cluster. Aside from the cost, there are operational challenges to deliver a traditional computing cluster in a classroom setting. For research experts, classroom use is usually not at the top of the priority list when it comes to computing clusters, setting up user accounts, and providing remote access. Configuring the various software packages needed to teach effectively can prove to be a formidable, if not insurmountable, obstacle.

The classroom usage pattern typically has a one to two week "peak-period" during the course when the cluster experiences its largest load – usually this is when a problem set is due [3,4]. Traditionally, this pattern would require classroom users to request compute time donations from research computing clusters. The secondary nature of the classroom use and the possibility that multiple courses utilize the same resources can create an oversubscribing of the cluster during a peak-period. This can pose a problem as students have to compete for resources. Limited resources can determine what the student can accomplish depending on how intensive each course's calculations are.

3. Living with stars in clouds

StarHPC remedies the problems traditionally associated with bringing computational resources into the classroom by providing an on-demand, dedicated, and dynamic compute cluster hosted by Amazon's EC2 [5] web service. Amazon's EC2 allows a user to request a number of virtual machines to be started on computers in Amazon's data centers. Each running virtual machine incurs a cost of $0.10/hr/machine (January 2009) until it is shutdown. Comparing this to the cost of owning and operating traditional computing clusters presents a very appealing service with an affordable pricing model.

Amazon also provides an extensive API for managing virtual machines allowing the user to dynamically adjust the number of running virtual machines as necessary. The API can also be used to take a snapshot of the working state of a virtual machine into a virtual machine "image". The virtual machine image can then be reloaded at a later time to produce an identical system configuration. This allows for capturing the software and system configuration needed for a particular course. Using the Amazon EC2 API permitted us to automate most of the administrative tasks such as creating user accounts, setting up workspaces, configuring remote access, and so on.

The use of Amazon's EC2 and virtual machine images helped us address many of the issues we had with bringing computational resources into the classroom. First, we eliminated the need for concrete resources such as servers, storage, power, and cooling needed to facilitate a computing cluster. All of these concerns were shifted to Amazon by utilizing EC2 for hardware needs. Second, the need to inspect the system setup and configuration was eliminated by capturing it in a reloadable virtual machine image. This lowers the administrative overhead work, as it only needs to be done once and not every semester. Third, we were able to elevate classroom users to first-class citizens on the cluster and eliminate under-utilized resources since the user only pays for what is used during peak-period usage. Fourth, the problem of courses competing for limited resources during peak-period usage is eliminated by the ability to launch multiple independent clusters. And finally, by creating separate virtual machine images, each course can have its own set of configurations and software packages completely independent of each other.

4. Software solution contents and target classrooms

StarHPC solution Fig. 1 has 3 elements:

- A Virtual machine image used by students to develop software on their computers
- Administrative scripts that manage the Amazon EC2 cluster
- Amazon EC2 virtual machine image (AMI) that is run on the elastic compute cloud

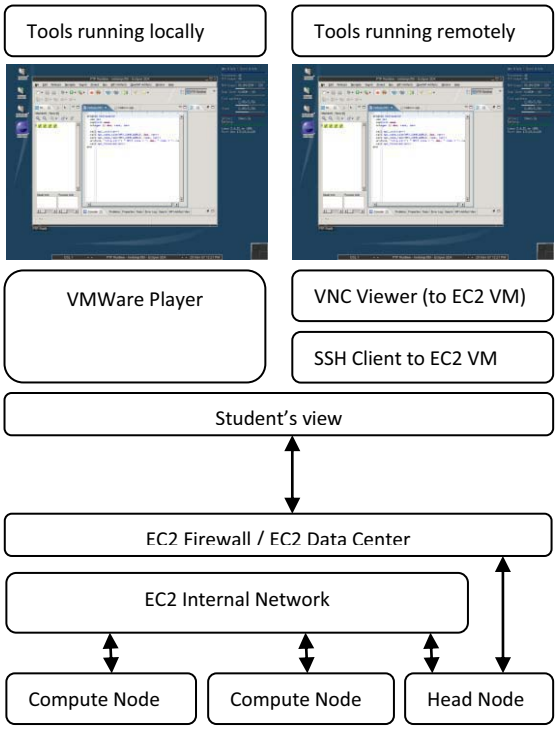

Figure 1: StarHPC solution architecture

The primary target audiences for StarHPC are advanced computer science courses that are designed to uncover the challenges of parallel programming practice. StarHPC can be also used in courses that teach numerical modeling and simulations.

4.1. Student's view

Virtual Machine images are distributed to students online or via burned DVDs. Students run a virtual machine using the freely available VMWare player [6]. The student's image Fig. 2 is built on Linux and it contains the following components: Photran Development Tools (Fortran) [7], Eclipse Parallel Tools Platform 1.1 [8], OpenMPI 1.2.4 [9] and the Sun Studio IDE [10]. This allows students to use the best of breed tools used by researchers and professional developers of high performance systems.

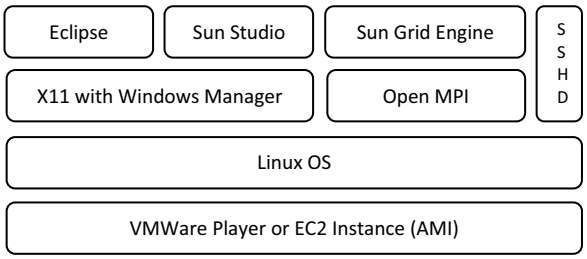

Figure 2: Software packaged in StarHPC VM

4.2. Educator's view

Educators using the virtual machine can have the same experience as the students while developing and testing parallel programs for the classroom. In addition to the virtual machine image educators use scripts and the Amazon EC2 virtual machine image (AMI) to provide a way to manage and administer a dynamic, on-demand, remote computing cluster. This eases the administration overhead as the STAR team provides the administrative scripts and the AMI. In turn, this frees the educator and the students from system administration tasks and allows them to focus on teaching the concepts of parallel computing and not be distracted by the details of setting up the hardware and software environment to support the course.

4.3. Remote cluster - Amazon Machine Image

When a remote cluster is created using the provided scripts, it features all the development packages available in the student's virtual machine as well as TightVNC [11] and OpenSSH [12] configured for remote access. This allows students and educators to test code developed locally and deploy it on remote cluster. The ability to run code on a dedicated remote cluster allows a meaningful discussion of network latency, communication overhead, and a whole slew of bugs that go unnoticed in non-distributed and/or non-symmetric multiprocessing environments. It also exposes students to the challenges of remote parallel debugging and allows them to appreciate the complexities of parallel software development.

7. Results

StarHPC was used in the "Parallel Programming for Multi-core Machines Using OpenMP and MPI" [13] course taught during an Independent Activities Period (IAP) at MIT. The course was a two-week crash course in both OpenMPI and OpenMP [14] programming techniques for writing software that runs on high performance computing systems. Throughout the course, students were guided through implementing an increasingly complex set of parallel programming exercises. Students logged-in remotely to the StarHPC cluster to develop, compile, and run their source code from virtually anywhere; even from a web browser if necessary. The course lasted two weeks and had a total of ten users actively developing on the cluster. The estimated total cost for using StarHPC hosted on Amazon's EC2 web service was around $150 for the course. Comparing this to the cost of a physical cluster, Amazon's EC2 web service presents an affordable solution to supporting peak-period computing needs in the classroom.

As with any solution, we needed to make trade-offs while developing StarHPC. To run StarHPC locally one needs to have administrative access to the machine (to install the VMWare player) and the fairly powerful computer with plenty of memory. We tried to mitigate risks and make it accessable to students who do have issues running StarHPC locally by providing a remote interface directly into the EC2 cluster. This, however, requires high

throughput and a low latency connection to the Amazon EC2 cluster. While broadband access is prevalent, network issues may prevent students from having an optimal experience while using our solution.

The StarHPC solution is available at the project's web site (http://web.mit.edu/star/hpc) and the solution can be replicated in classrooms worldwide by downloading the VMWare image and following the setup instructions on the the website.

6. Conclusions

With StarHPC, the Software Tools for Academics and Researchers (STAR) project has developed a framework for deploying technology in a streamlined way for teaching parallel programming that is both sustainable and scalable. StarHPC provides the development environment and computational resources necessary to teach parallel programming in OpenMP and OpenMPI. StarHPC represents the STAR team's first attempt to use Amazon's EC2 web service to address the issues with bringing computational resources into the classroom.

8. Acknowledgements

Dr. Constantions Evangelinos for being an outstanding collaborator and client; his enthusiasm helped StarHPC attain the success it has.

Dr. Vijay Kumar, Associate Dean and Director of the Office of Educational Innovation and Technology (OEIT), and Prof. Dan Hastings, MIT Dean of Undergraduate Education (DUE) for their ongoing support.

8. References

[1] Martin Bernreuther, Markus Brenk, Hans-Joachim Bungartz, Ralf-Peter Mundani and Ioan Lucian Muntean: Teaching High-Performance Computing on a High-Performance Cluster; Lecture Notes in Computer Science 2005; 3515; 1-9

[2] Riley J.T., Ceraj I, Shubert C. StarHPC; 2006. http://web.mit.edu/star/hpc [1/9/2009]

[3] Per Andersen: The Texas Tech Tornado Cluster: A Linux/MPI Cluster For Parallel Programming Education And Research; 1999. http://www.acm.org/crossroads/xrds6-1/tornado.html [1/9/2009]

[4] Riley J.T: StarBiogene and Game 5 of the ALCS; 2007. http://web.mit.edu/due/administration/Newsletter/DUENewsDec2007.pdf [1/9/2009]

[5] Amazon EC2. Amazon Elastic Compute Cloud. http://aws.amazon.com/ec2 [1/9/2009]

[6] Sugerman J, Venkitachalam G, Lim BH. Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor; Proceedings of the General Track: 2002 USENIX Annual Technical Conference. http://www.usenix.org/publications/library/proceedings/usenix01/sugerman/sugerman.pdf [1/9/2009]

[7] Photran - An Integrated Development Environment for Fortran. http://www.eclipse.org/photran [1/9/2009]

[8] PTP - Parallel Tools Platform. http://www.eclipse.org/ptp/ [1/9/2009]

[9] Gabriel, E. Fagg, G. E. Bosilca, G. Angskun, T. Dongarra, J. J. Squyres, J. M. Sahay, V. Kambadur, P. Barrett, B. Lumsdaine, A. Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation. Lecture Notes in Computer Science 2004; 3241; 97-104

[10] Sun Studio C, C++ and Fortran Compiler and Tools http://developers.sun.com/sunstudio/ [1/9/2009]

[11] TightVNC http://www.tightvnc.com/ [1/9/2009]

[12] OpenSSH http://www.openssh.com/ [1/9/2009]

[13] Evangelinos C. "Parallel Programming for Multi-core Machines Using OpenMP and MPI" http://stellar.mit.edu/S/course/12/ia08/12.950/index.html [1/9/2009]

[14] Dagum, L. Menon, R. Open MP: An Industry-Standard API for Shared-Memory Programming. IEEE computational science and engineering. 1998 5(1); 46-55