Undergraduate Review

Volume 10 | Article 13

2014

Advanced Computing Systems for Scientific Research

Jared Buckley

Jason Covert

Talia Martin

Follow this and additional works at: http://vc.bridgew.edu/undergrad_rev

Part of the Computer Engineering Commons

Recommended Citation

Buckley, Jared; Covert, Jason; and Martin, Talia (2014). Advanced Computing Systems for Scientific Research. *Undergraduate Review*, 10, 45-48.
Available at: http://vc.bridgew.edu/undergrad_rev/vol10/iss1/13

This item is available as part of Virtual Commons, the open-access institutional repository of Bridgewater State University, Bridgewater, Massachusetts.
Copyright © 2014 Jared Buckley, Jason Covert, and Talia Martin

Advanced Computing Systems for Scientific Research

Jared Buckley, Jason Covert, and Talia Martin

Jared Buckley, Jason Covert and Talia Martin collaborated on developing a more efficient technology resource for scientific computing under the mentorship of Dr. Robert Hellström (Geography). Jared ('14) majored in Physics at BSU and is continuing his interest in using computing systems to study weather and climate at UMass Dartmouth. Jason is majoring in geography and has an interest in weather studies and GIS mapping. Talia is a physics major at BSU and enjoys astronomy. She was awarded a 2013 NASA Space Grant to conduct astrophysics research.

An advanced computing system was constructed at Bridgewater State University to provide students access to computing machines tailored to the purpose of computational scientific research. This paper provides an overview of the construction, design, capability, and future potential of the computing system.

Project Reasoning

During the course of projects utilizing a numerical weather prediction (NWP) model, it became evident that the desktop computing machines provided to students in computer labs were not capable of yielding results in a reasonable amount of time for non-standard applications. The standard configuration of computers in the Conant Science and Mathematics Center at Bridgewater State University is sufficient for the majority of student purposes; however, these standard computers become insufficient for more comprehensive applications. Because of this insufficiency, it became necessary to seek better performing computer systems to support research applications.

Project Execution

To increase the computational power of computing systems on campus, two distinct steps were taken. The first step was the upgrade of a standard desktop computer. This upgrade provided a single desktop machine with superior computing capability to the standard machines available to students. The second step was the construction of a Linux computer cluster from several networked previous generation machines.

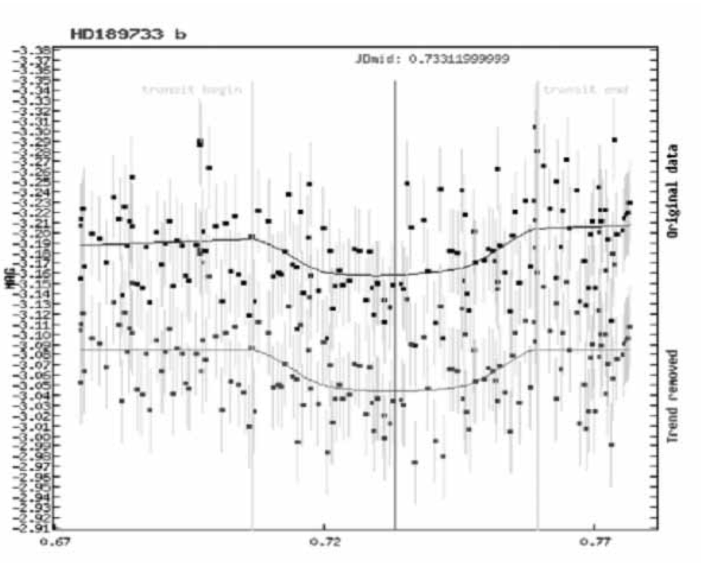
Standard Desktop Computer Upgrade

To reduce the execution time of computationally heavy tasks, a standard desktop computer was upgraded using conventional computer upgrade methods. These methods included Central Processing Unit (CPU) upgrading and the addition of Random Access Memory (RAM). The standard computer used had an original CPU configuration of the Intel Pentium G850 dual-core processor and a RAM configuration of 8 GB. Upon upgrade, the standard computer had a CPU configuration of the Intel Core i5-2500 quad-core processor and a RAM configuration of 16 GB. The upgraded specifications allowed for faster computation and larger tasks. The standard computer's main operating system was also changed from Windows 7 to CentOS 6 (a Linux distribution) to provide support for native Linux programs and development[1].

Application

To provide a stable computer working environment for image processing, the program Maxim DL was installed onto the upgraded standard desktop computer[2]. Maxim DL is an astrophotography software program used to analyze images in order to obtain information about the objects in the images[2]. This process, called photometry, can have many uses such as determining the brightness of a star in an image[3]. In working with exoplanets, photometry can be used to detect how bright a star is through several images over a period of time[3]. When an exoplanet crosses in front of its parent star it blocks some of the light coming from the star and this change in brightness can be detected in the images[4]. A light curve of this change is made to show when an exoplanet transits and can be used to gather other useful information, such as the size of the planet, the length of its orbit and in some cases, by using different filters (light curve shown in figure 1)[3]. Even what the planet is made of can be found[3].

Figure 1. HD 189733b Light Curve



Images of the transit of the known exoplanet HD 189733b were collected during the summer of 2013 at Bridgewater State University. The program Maxim DL was used to calibrate and analyze the images. This process requires using a series of calibration images that will remove background noise from the images to be analyzed[3]. Once this was completed for all images, Maxim DL was used to analyze the brightness of the main object as well as three comparison stars. The comparison stars act as a control, making sure that the brightness of all the stars in the image are not changing, only that of the main object[3].

The length of the transit for HD 189733b is approximately two and a half hours and an image is taken about once every minute for the duration of the transit and for ten minutes before and for ten minutes after. This results in well over 100 images being taken, each with a rather large picture file size. The computer originally used to calibrate and analyze the images taken over the summer had 4GB of RAM. It took approximately two weeks, working between 2 and 4 hours a day on average to complete the calibration of the images. The program would freeze if an attempt was made to calibrate more than 10 images at a time, so the whole process had to be done in batches of 10, otherwise the program stopped responding and anything done previously was not saved.

The transit of HD 189733b was once again captured at Bridgewater in the fall of 2013. This time, however, the upgraded standard computer was used to calibrate and analyze the data. It did stop responding when it was asked to calibrate all of the open images at once. Splitting the images into two groups of approximately 70 images worked well and it took less than 2 hours to fully calibrate all of the images. The same task had taken two weeks with the previous computer.

Using the upgraded standard computer made the image analysis process faster, as more images could be open at a time, but the greatest benefit was the reliability. Even if only 15 images were open while analyzing previously, the program would still shut down randomly, causing loss of work. The program would then have to be restarted and those images analyzed again, wasting time and causing a fair amount of frustration. The upgraded standard computer allowed for larger groups of images to be analyzed at one time and continued responding for as long as was required.

Construction of a Linux Computer Cluster

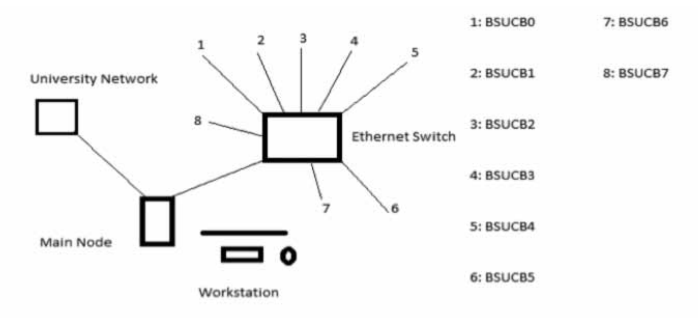
Using 8 idle, previous generation computers and the upgraded standard computer, a 9 node (computer) Beowulf-style Linux computing cluster was constructed during the fall semester of 2013[5]. The cluster is a distributed memory, heterogeneous computer cluster operating over a Fast Ethernet network consisting of a single network switch and 20 total CPU's[5]. Each individual machine runs the CentOS 6 Linux operating system[1]. Nicknamed the Bridgewater State University Crystal Ball (BSUCB), this cluster provides a dramatic increase in program execution speed through the use of parallel processing.

Bridgewater State University Crystal Ball (BSUCB) Design

The BSUCB is setup with 9 nodes in total. The main (control) node is a quad-core system with 16 GB of RAM and the compute nodes are dual-core systems each with 2 GB of RAM. Although the use of the cluster resources is dynamic, the BSUCB cluster design allows for a 20 CPU parallel processing device.

The BSUCB nodes are connected via a single Ethernet switch and communicate over a Fast Ethernet Network. The BSUCB is in general closed off from the University computing network, but the main node retains access to the University network for Internet connectivity when data is required to be downloaded. The main node is physically accessible via a workstation setup so that the BSUCB can be controlled. This workstation also allows the main node to be used as a standalone machine.

Figure 2. The BSUCB Design



Application

The construction of the BSUCB arose from the use of the numerical weather prediction (NWP) model the Advanced Regional Prediction System (ARPS)[6]. The ARPS was built/written in the Fortran computer language by members of the Center for Analysis and Prediction of Storms (CAPS) at the University of Oklahoma[6]. The ARPS was used during the summer of 2013 to provide a foundation for short range cloud cover forecasting. The general performance of the ARPS at Bridgewater State University has been poor due to computing restrictions. Because the ARPS is natively coded (designed) to run across a Linux computer cluster for advanced parallel processing, the BSUCB became the perfect plan to reduce ARPS execution time[6].

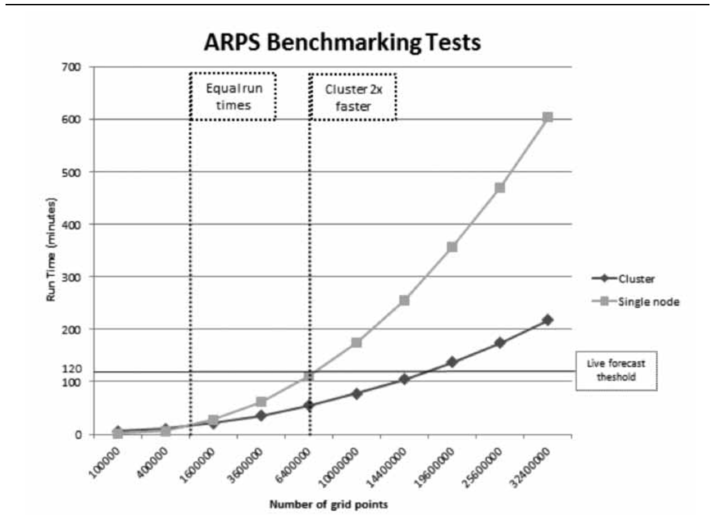
The BSUCB is capable of running the ARPS using Open Message Passing Interface (MPI)[6]. Open MPI is an open source adaptation of the MPI standard, a standard designed to allow networked computers to divide tasks and speed up program execution[7]. This execution time reduction will allow for more advanced projects using the ARPS to execute in a significantly more reasonable amount of time.

Benchmarking

Benchmarking, the assessment of the performance of computing systems, of the BSUCB was done using a 2 hour forecast supercell thunderstorm simulation run with the ARPS. To gauge the relative increase in performance of the BSUCB, the benchmarking of the BSUCB and the upgraded standard desktop computer were compared. The supercell simulation was run a total of 10 times on the BSUCB and on the upgraded desktop. Each run was made computationally more intense by increasing the size of the spatial domain of the simulation. The spatial domain determines the number of computational grid points in a simulation. Grid points are used by NWP models to hold information about the atmosphere[8,9]. This atmospheric information is used by NWP models in a series of calculations in order to predict a future state of the atmosphere[8,9]. Because the NWP models must perform calculations at each grid point, the larger the number of grid points, the more computationally intense the simulation. It is expected that an increase in the number of grid points increases the execution time.

Figure 3 displays the results of the ARPS benchmarking procedure. The line labeled "cluster" represents the results of the BSUCB benchmarking and the line labeled "single node" represents the results of the upgraded standard desktop computer benchmarking. Vertical lines are drawn to show where the BSUCB and upgraded desktop had identical performance and where the BSUCB and upgraded desktop performance were separated by a factor of 2. A horizontal line is drawn to show where the ARPS execution time is no longer faster than the weather it is simulating.

Figure 3. Benchmarking Results



The significance of figure 3 is that it proves the usefulness of the BSUCB as compared to a single computer. Once the number of grid points increases significantly, the BSUCB greatly outperforms the single computer. The BSUCB is also capable of performing more computationally intense live forecasting.

Despite the obvious advantages of the BSUCB, the beginning of the benchmarking graph reveals an important idea. The BSUCB is outperformed by the upgraded standard desktop computer when the number of grid points is small. This is because it takes physical time for the nodes in a cluster to speak with one another and transfer tasks over a network. As constructing/writing computer code to execute across a cluster can be a time consuming project, this slow down during less intense operations suggests that such operations are better suited to run on a conventional computing setup.

Future Work

Projects that utilize the BSUCB and upgraded standard computer are planned for the future. The ARPS will continue to be run on the BSUCB for weather prediction projects. There is also some consideration for the installation of alternate NWP models, such as the Weather Research and Forecasting Model (WRF), and a long term climate modeler[10]. Physics students are also planning to use the BSUCB to perform computationally intense simulations, including space-time simulations. The upgraded standard desktop computer will also continue to be used to provide a stable computing system for astronomical image processing.

Personal Acknowledgements

Thanks are extended to Dr. John Santore (Bridgewater State University Computer Science Department) for allowing the use of idle computers for the construction of the BSUCB, to James Govoni for providing networking supplies and support, and to Dr. Robert Hellström for his support as mentor of this project.

Software Acknowledgements

Thanks are extended to CAPS at the University of Oklahoma for providing the ARPS for free and open source public use, the CentOS development team for providing a free, open source operating system, and to the developers of Open MPI for their free, open source adaptation of the MPI standard. Without open source software, projects such as the construction of the BSUCB may not be possible.

References

1. *www.centos.org - The Community ENTerprise Operating System*. Retrieved December 13, 2013, from http://www.centos.org/

2. dl. (n.d.). *Cyanogen Imaging Products from Diffraction Limited*. Retrieved December 13, 2013, from http://www.cyanogen.com/maxim_main.php

3. Berry, R., & Burnell, J. (2006). *The Handbook of Astronomical image Processing: Includes AIP4Win Software 2.0* (2. Engl. ed.). Richmond, Va.: Willmann-Bell.

4. Carroll, B. W., & Ostlie, D. A. (2007). *An Introduction to Modern Astrophysics* (2nd ed.). San Francisco: Pearson Addison-Wesley.

5. Zomaya, A. Y. (1996). *Parallel and Distributed Computing Handbook*. New York: McGraw-Hill.

6. "CAPS - Center for Analysis and Prediction of Storms." (n.d.). *CAPS - Center for Analysis and Prediction of Storms*. Retrieved December 13, 2013, from http://www.caps.ou.edu/

7. "Open MPI: Open Source High Performance Computing." (n.d.). *Open MPI: Open Source High Performance Computing*. Retrieved December 13, 2013, from http://www.open-mpi.org/

8. Giordano, N. J., & Nakanishi, H. (2006). *Computational physics* (2nd ed.). Upper Saddle River, NJ: Pearson/Prentice Hall.

9. Vasquez, T. (2002). *Weather Forecasting Handbook* (5th ed.). Garland, TX: Weather Graphics Technologies.

10. "The Weather Research & Forecasting Model." (n.d.). *The Weather Research&Forecasting Model Website*. Retrieved December 13, 2013, from http://www.wrf-model.org/index.php