In Search of Clusters for High Performance Computing Education

Paul Gray and Sarah Miller[1]

Dept. of Computer Science
University of Northern Iowa
Cedar Falls, Iowa 50614-0507
gray@cs.uni.edu sarahm@uni.edu

Abstract. This paper examines the issue of High-Performance Computing (HPC) platforms in undergraduate education and the Bootable Cluster CD (BCCD) to facilitate hands-on, classroom immersion in HPC topics. There is measurable benefits derived from teaching HPC topics through providing hands-on clustering experience to the students. Supporting an HPC environment for instruction, however, is accompanied by the commensurate demand in administrative tasks associated with staffing costs or personnel to maintain a dedicated high-performance computing environment–from maintenance of user accounts and configuring software components, to keeping ahead of the latest security advisories. In this document, the authors survey several HPC computing environments in the context of HPC education: The Open Source Clustering Application Resources (OSCAR), ClusterKnoppix, and NPACI Rocks, and we will focus on the unique aspects of the BCCD in breadth of HPC software, ease of configuration, and ease of administration.

1 Introduction

In 2001, IEEE and ACM released a joint recommendation for undergraduate curricula[4]. Included in their recommendation are several courses that speak directly to the topics of high-performance computing, including:

- Parallel Algorithms
- Parallel Architectures
- High Performance Computing
- Net-Centric Computing
- Specialized Systems Development

In addition, the number of graduate programs and research projects which revolve around high performance computing topics–such as high-availability systems, parallel I/O, grand-challenge areas and more–have increased signifi cantly in recent years. These programs, in turn, bring about a demand for graduate students that have been trained with both theoretical and practical knowledge in HPC.

Instructors at primarily undergraduate institutions (PUIs) wanting to explore implementation associated with the corresponding discipline are faced with several signifi cant impediments: namely

1. The signifi cant fi xed cost of system hardware and infrastructure support for a dedicated cluster.

[1] Work supported in part by a grant from the National Computational Science Institute (NCSI)

2. The ongoing person-hours required for system administration, software configuration, policy enforcement, system monitoring, and aspects of security.
3. The development of curricular modules to augment existing course materials and material for new HPC-centric courses.

One alternative to purchasing a state-of-the-art clustering environment available to faculty at primarily undergraduate institutions is to apply for an account on a cluster at one of the super-computing centers[2]. Another approach is to leverage clustering distributions to build an evaluation cluster for HPC education, using new or second-hand computer workstations. A third approach is taken by the Bootable Cluster CD project (BCCD), which provides a non-invasive clustering environment that runs exclusively off of a node's CD drive and system RAM, leaving the underlying operating systems(s) on the hard drives untouched.

The outline for the remaining portion of the paper is as follows: Section 2 discusses clustering distributions and their suitability for HPC education. Section 3 discusses various self-contained bootable CD distributions that are available for clustering environments. Section 3.1 details the composition of the BCCD ([9]) image, how to customize it, and some of the BCCD features that are unique to CD-clustering environments.

2 Clustering Distributions

It should be noted that there continues to be strong support for vendor-supported clustering packages. Today's HPC vendors offer clustering options that include all of the hardware, compilers, software, and support necessary to provide a high-performance computing environment with minimal setup time. To most undergraduate institutions, however, the cost and commitment are well beyond budgetary consideration.

A more modest, yet viable approach to setting up a clustering environment that continues to be available to researchers and educators is to purchase commodity off-the-shelf (COTS) components and to build a cluster entirely from scratch. After sufficient testing, troubleshooting, and tuning, this often produces a very capable computational environment. Others–the authors included–have sought out capable "second-hand" systems that have reached the "end-of-life" in terms of production use. Systems that have performed well in production for one to three years still possess considerable *educational* value for HPC education.

Whether one takes a COTS or second-hand approach to building a cluster for HPC educational use, "*clustering distributions*" such as the Open Source Cluster Application Resources ([5]) (OSCAR) and NPACI-Rocks ([16]) provide a balanced, integrated approach to the installation of clustering tools and packages. Clustering distributions integrate (e.g. NPACI-Rocks) or augment (e.g. OSCAR) the standard GNU/Linux operating system installations with specialized and pre-configured clustering tools and utilities.

The "clustering distributions" approach has been proven to be a very effective approach to the instantiation and long-term support of dedicated clustering resources. The viability in this approach is reflected by the adoption of OSCAR and NPACI-Rocks by several hardware vendors as an avenue to providing packaged clustering environments to research labs and

[2] `allocations@ncsa.uiuc.edu`, for example.

institutions. By augmenting or integrating clustering distributions with existing GNU/Linux distributions, extensive hardware support for both legacy systems and for cutting-edge hardware is realized. Consequently, there are very strong arguments for adopting clustering distributions for dedicated clustering environment.

The situation in many community colleges and primarily undergraduate institutions is one where dedicated clustering resources are out of the question due to budgetary, administrational, or space reasons. In the situation where a clustering environment must co-exist with the day-to-day computing needs, dual-booting workstations with Microsoft Windows and clustering distributions is one approach to meet both needs.

OSCAR has made tremendous progress toward providing "network booting," or PXE-booting of the OSCAR client nodes as well as booting an OSCAR cluster under Microsoft Windows systems using VMWare ([6]). While these approaches require a bit more configuration than the CD-based approach described below, one could successfully leverage these flexible booting approaches to support the dynamic clustering needs associated with HPC education in the situation where dedicated hosts are not available.

3 CD-based Clustering Solutions

With the number of bootable Linux *distributions* growing lately, such as Knoppix ([11]), SuSE Live ([17]), etc. There has been a corresponding recent increase in the number of approaches to live clustering CD images. Some of these clustering CD images include ClusterKnoppix ([18]), CHAOS ([13]), and Warewulf ([12]). In addition to compact, CD-based clustering solutions, there has been some projects that have even looked to floppy-based clustering, such as openMosixLOAF ([2]). The remainder of this section discusses the CD-based clustering solution, BCCD.

3.1 The Bootable Cluster CD

The BCCD is a self-contained clustering environment in a bootable CD format (see Figures 1, 2, 3, and 4). The project was motivated by the need for a drop-in clustering environment for parallel computing education at community colleges and primarily undergraduate institutions (PUI's). Unlike research institutions, where dedicated clustering resources have the benefits of full-time support and administration staff, the support for hardware and the ongoing administration of clustering environments is a significant resource drain on community colleges and PUIs. However, these days student labs with Windows operating systems are becoming universally available for class-time use. The Bootable Cluster CD project was put forward to leverage these environments so as to provide a convenient, non-destructive (no installation required), preconfigured clustering environment–providing PVM, MPICH, openMosix, compilers, and graphical debugging tools–and to "leave no trace" when the host computer system is allowed to reboot normally.

In this way, the BCCD project offers a drop in solution that requires no dual-boot configurations, minimal system configuration, no ongoing administration, and an extremely robust on-demand development environment. This paper will describe the flexibility inherent in building the BCCD, describe its use at SuperComputing for supporting workshops on High Performance Computing Education, and tips for effective usage.

When running the Bootable Cluster CD, a full clustering environment–including compilers (C/C++/Fortran/Java), parallel computing environments (PVM [8], MPICH [10], LAM [3]), debugging tools (gdb, Electric Fence, ddd), clustering tools (C3 tools [7], omdiscd [1], ssh keys) and graphical visualization tools (upshot, xmpi, xpvm, openMosixView [15]) can be established in the time it takes to boot a workstation from an ordinary CD drive. The "live" clustering environment of the BCCD lives only in system RAM and allows the workstation to revert back to the operating system(s) on the hard drive upon system reboot. Over the past two years, the Bootable Cluster CD has been used for several week-long workshops on cluster computing education for Primarily Undergraduate Institutions (PUIs), and has been shown to be quite useful as a drop-in clustering environment where dedicated clustering resources are not readily available, but computer labs with Microsoft Windows operating systems are.

3.2 The Components of a BCCD Cluster.

By design, a "Bootable Cluster CD" is inherently customizable. The entire system is dynamically built from web-fetch sources from a BSD Ports-style dependency tree (in a build process referred to as "*gar*," used for the development of the LNX-BBC ([14]) project[3]). The build process brings together source code and local customizations in order to produce the final raw cdrom image.

Once the BCCD image is built, transferred to a cdrom disc and booted on a workstation, the BCCD initialization process begins by soliciting a user-level password for the booted image. User-level accounts are one aspect of the BCCD that differs from other clustering images (c.f. [13]). Initialization continues with the configuration of the network. If a DHCP server is readily available, the BCCD image leverages the existing services to configure the network. The BCCD also allows for manual configuration of the network parameters and includes a DHCP and DNS server for situations in which no such services exist on the local network (on a dedicated cluster, for example).

Another unique aspect of the BCCD is the manner in which ssh is dynamically configured to facilitate remote process instantiation—which is a requisite for the process-based parallel computing environments. Once the system has initialized, the student or instructor can log in using the user account *bccd*, with the password specified during the boot process. Upon login, the user is given the opportunity to start the *pkbcast* process, which broadcasts public ssh keys of the individual user across the local network. Workstations running the BCCD environment adopt an *opt-in* approach to resource utilization. The only way that a workstation can enter into a clustering environment is by way of the local user's intentional review and acceptance of public keys that have been collected from the local ssh-key broadcast program mentioned above, *pkbcast*.

Using the public ssh-key broadcast, key review, and volunteering of resources paradigm, no user's machine can be accessed or utilized without the permission of the local user. In a classroom of students, this also means that no student is accessing the instructor's workstation or other student's systems without having expressed permission to do so. Once public keys have been reviewed and accepted, the students now have an environment that facilitates distributed process instantiation across the clustering resources through ssh, and a

[3] Dr. Gray is an active maintainer and developer of the LNX-BBC project as well.

full clustering environment is now available for writing, compiling, running, and debugging parallel applications[4].

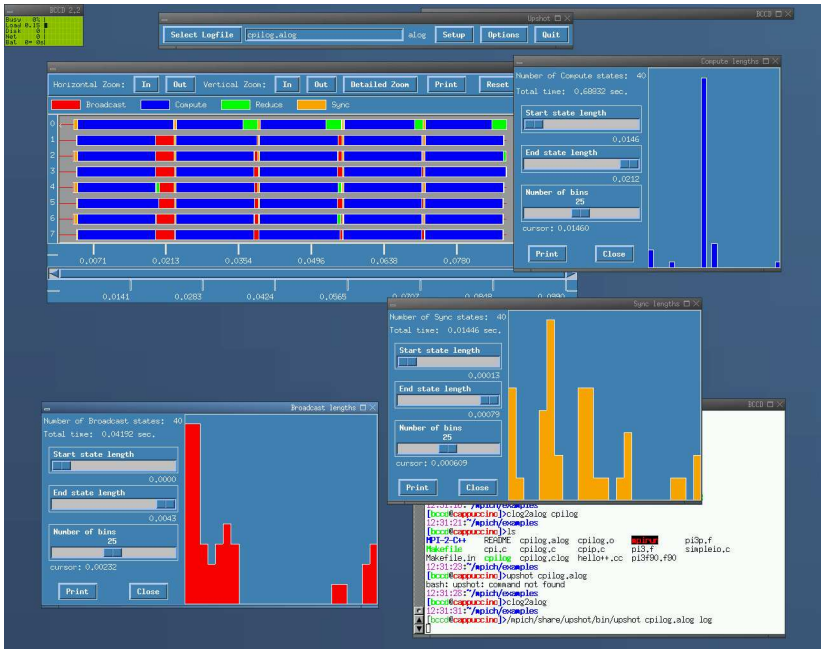

Fig. 1. MPICH and the core profiling and debugging libraries (via mpe) are included on the BCCD. "upshot" is also included for the analysis of clog and alog files.

– **MPICH on the BCCD:** The default MPI environment on the BCCD distribution is MPICH (see Figure 1). However, changing MPI environments between LAM and MPICH is trivial to do, and can be done while the cluster is "live," or even when the BCCD cdrom image is being built. The current version of MPICH on the BCCD image is 1.2.5[5]. In addition to the standard MPICH environment, the BCCD includes the MPE extensions for profi ling and visualization. The Tcl/Tk version of upshot is provided on the standard image. However, due to licensing issues with the Java runtime environment, the Java-based visualization tools (jumpshot, for example) are unfortunately not available on the standard BCCD.
– **LAM-MPI on the BCCD:** With a simple change to the "live" runtime system, or by changing a simple build-time environment parameter, the MPI runtime environment defaults to LAM-MPI support. With LAM-MPI enabled, one has access to the *recon*, *lamboot* and corresponding environmental tools associated with LAM. In addition, graphical debugging though *xmpi* is also available to the users (see Figure 2).

[4] This manual process of reviewing and accepting ssh keys has been a very compelling student exercise: providing insight into how process-based clustering environments are put together. However, the BCCD also supports *init* (boot) options which automate the tasks of public key broadcasts, collection, and the persistence of monitoring for new or dead hosts in the pool.

[5] MPICH version 2.0 should be standard on the BCCD image by the time of the LCI conference.

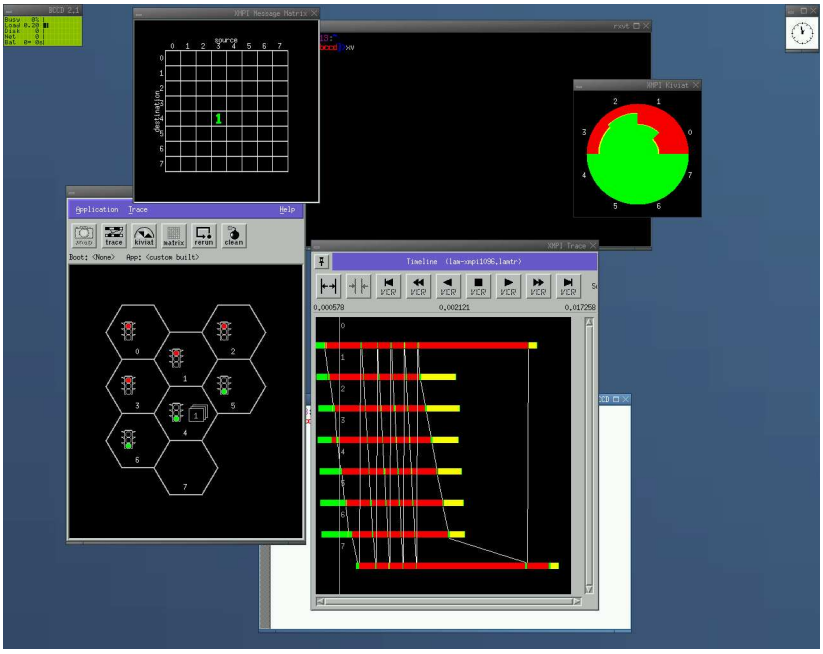

Fig. 2. The BCCD provides tools for creating, building, debugging, and profiling LAM-MPI applications. The figure above shows `xmpi` being used to trace the execution of a distributed communication ring process from the standard LAM examples (which are also included in the image).

– **OpenMosix on the BCCD:** OpenMosix support provides kernel-level process migration and clustering capabilities across a BCCD-based cluster. With the alternative kernel-supported distributed processing paradigm of openMosix comes an additional educational opportunity. Using visualization tools such as *openmosixview, openmosixmigmon* (the openMosix migration monitor), *openmosixanalyzer, mosmon*, and *mtop* (openMosix "top"), students can explore the characteristics of kernel-based clustering (see Figure 3). The version of openMosix running on the BCCD matches the running kernel; the openMosix-tools is version 0.3.5; the openMosixView distribution on the BCCD is version 1.5.
– **PVM on the BCCD:** Support is also available for creating, compiling, running, and debugging programs under the PVM environment (see Figure 4). The PVM version on the BCCD is 3.4.4; XPVM is version 1.2.5. Users can run PVM programs from the command line, from the pvm console, or from the graphical tool "*xpvm*."

  In addition to the core clustering environments described above, all of the example codes from the respective distribution packages — LAM-MPI, MPICH, and PVM, for example — are included on the standard BCCD image. This allows for a very straightforward and immediate "boot-and-run-code" classroom atmosphere.

4 Summary and Future Work

In the context of cluster computing education, the Bootable Cluster CD approach has many advantages: ease of installation (boot it up), ease of administration (automated ssh-key gen-

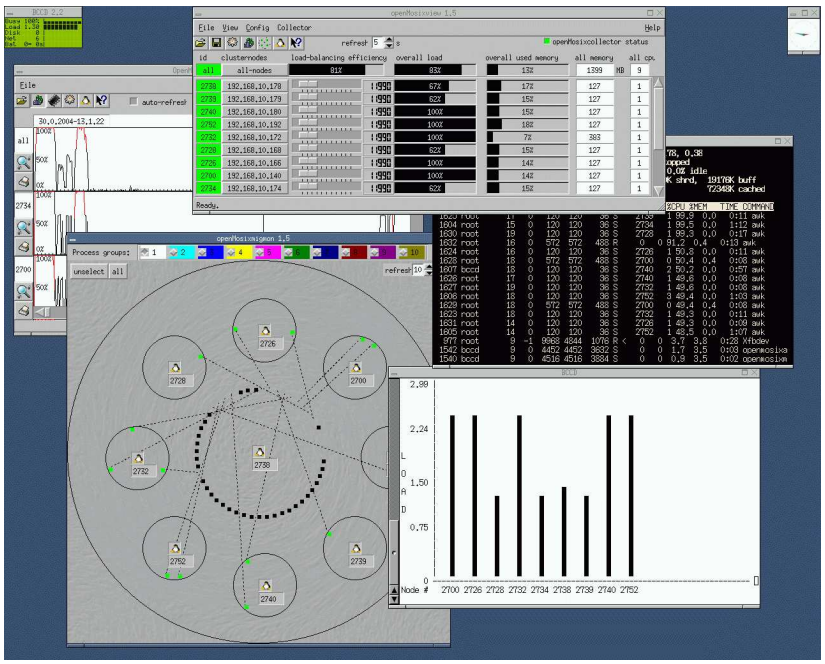

Fig. 3. Kernel-level distribution and clustering courtesy of openMosix and corresponding visualization and monitoring tools are provided on the BCCD. The figure above shows the graphical applications openmosixview, openmosixanalyzer, and openmosixmigmon (migration monitor). Also shown are the text-based utilities `mosmon` and `mtop`.

eration and distribution), ease of configuration (all of the software is preconfigured for the runtime system) and customization (being built from on-demand web-fetched sources). For institutions that lack dedicated hardware or administrative resources to support a dedicated clustering environment, the BCCD provides a compromise to dual-booting or maintaining NFS and tftp servers for supporting PXE booted kernels.

As it stands, the BCCD currently does not offer any scheduling service, which would greatly improve the breadth of educational topics. OpenPBS and the Sun Grid Engine (SGE) are currently being examined for licensing and distribution terms that would be suitable to include on the BCCD image. Another aspect which the BCCD project is working toward is a Grid Services image, which would include all necessary components to set up and configure a grid service, in the context of a classroom atmosphere.

References

1. BAR, M. Linux clusters state of the art. Online document available at http://openmosix.sourceforge.net, 2002.
2. BLOMGREN, M., AND JR., M. A. M. openMosixLoaf. Information available at the project web site http://openmosixloaf.sourceforge.net, 2003.
3. BURNS, G., DAOUD, R., AND VAIGL, J. LAM: An open cluster environment for MPI. In *Supercomputing Symposium '94* (Toronto, Canada, June 1994). Source available at http://www.lam-mpi.org.
4. CARL CHANG ET.AL. Computing curricula 2001; computer science, final report. Curricular Standards available for download at http://www.sigcse.org/cc2001/, 2001.

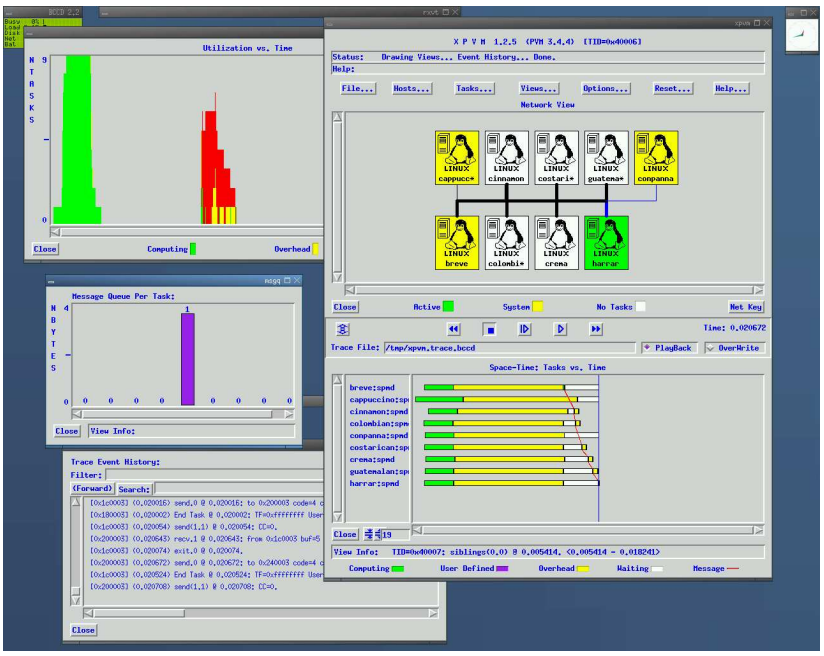

Fig. 4. A full PVM environment, including `aimk` and `xpvm`, is integrated into the BCCD

5. DES LIGNERIS, B., SCOTT, S., NAUGHTON, T., AND GORSUCH, N. Open source cluster application resources (OSCAR): Design, implementation and interest for the [computer] scientific community. In *Proceedings of the First OSCAR Symposium* (Sherbrooke, May 2003).
6. EMC CORPORATION. Vmware. Information available at the project web site http://www.vmware.com, 2004.
7. GEIST, A., MUGLER, J., NAUGHTON, T., AND SCOTT, S. Cluster command and control (c3) tools. Information available at the project web site http://www.csm.ornl.gov/torc/C3/, 2003.
8. GEIST, G. A., AND SUNDERAM, V. S. The PVM system: Supercomputer level concurrent computation on a heterogeneous network of workstations. In *Proceedings of the Sixth Distributed Memory Computing Conference* (1991), IEEE, pp. 258–261.
9. GRAY, P. The bootable cluster cd. Information available at the project web site http://bccd.cs.uni.edu, 2004.
10. GROPP, W., LUSK, E., AND SKJELLUM, A. A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing 22*, 6 (1996), 789–828. Source available at http://www.mcs.anl.gov/mpi/mpich.
11. KNOPPER, K. Knoppix. Information available at the project web site http://knoppix.org, 2003.
12. KURTZER, G. How warewulf works (a look into the tools). Information available at the project web site http://warewulf-cluster.org, 2003.
13. LATTER, I. Running clusterknoppix as a head node to a CHAOS drone army. Tech. rep., Macquarie University, AU, Dec. 2003. ITSecurity Group publication.
14. NICK MOFFIT ET.AL. The lnx-bbc project. Information available at the project web site http://lnx-bbc.org, 2004.
15. RECHENBURG, M. openmosixView. Online document available at http://www.openmosixview.com, 2003.
16. SDSC (UCSD) AND MILLENNIUM GROUP (BERKELY). Npaci rocks. Information available at the project web site http://www.rocksclusters.org/Rocks/, 2003.
17. SUSE LINUX. Suse linux for i386-Live. Image available for download at ftp://ftp.suse.com/pub/suse/i386/, 2003.
18. VANDERSMISSEN, W. Clusterknoppix. Information available at the project web site http://bofh.be/clusterknoppix, 2004.