High Performance Computing Environments Without the Fuss: The Bootable Cluster CD[*]

Sarah M. Diesburg and Paul A. Gray
Department of Computer Science
University of Northern Iowa
Cedar Falls, IA 50614
e-mail: sarahm@bccd.cs.uni.edu, gray@cs.uni.edu

David Joiner
Department of Science and Technology Education
Kean University
Union, NJ 07083
e-mail: djoiner@kean.edu

Abstract

This paper confronts the issue of bringing high performance computing (HPC) education to those who do not have access to a dedicated clustering environments in an easy, fully-functional, inexpensive manner through the use of the "Bootable Cluster CD" (BCCD). As an example, many primarily undergraduate institutions (PUI's) do not have the facilities, time, or money to purchase hardware, maintain user accounts, configure software components, and keep ahead of the latest security advisories for a dedicated clustering environment. The BCCD project's primary goal is to support an instantaneous, drop-in distributed computing environment. A consequence of providing such an environment is the ability to promote the education of high performance computing issues at the undergraduate level through the ability to turn an ordinary lab of networked workstations temporarily into a non-invasive, fully-functional clustering classroom.

The BCCD itself is a self-contained clustering environment in a bootable CD format. Using the BCCD, students, educators and researchers are able to gain insight into configuration, utilization, troubleshooting, debugging, and administration issues uniquely associated with parallel computing in a live, easy to use "drop-in" clustering environment. As the name implies, the BCCD provides a full, cohesive clustering environment running GNU/Linux when booted from the CDROM drives of networked workstations.

Keywords: Practical experiences with parallel and distributed systems, clustering environments, diskless clusters, PXE-boot clusters, high performance computing environments, high performance computing education

1. Introduction

The Bootable Cluster CD (BCCD, [9]) is a bootable CD image that boots up into a pre-configured distributed computing environment. It is unique among bootable clustering approaches in its ability to provide a complete clustering environment with pre-configured clustering applications and examples a full repertoire of development tools. This provides those who do not have access to a dedicated cluster, such as undergraduate students and independent researchers, with the tools and building blocks necessary to teach themselves the core concepts of high performance computing (HPC) through easy, hands-on experimentation.

In many places, such as primarily undergraduate institutions (PUI's), generic computer labs are becoming universally available for class-time use. The BCCD can be used to temporarily turn these computer labs into a convenient, non-destructive (no installation required), pre-configured clustering environment, providing a full array of clustering tools and environments. In the same way, the BCCD can be used to turn a couple of household computers into a personal, home clustering environment.

The BCCD system runs entirely from the host system's RAM and does not require nor use the local system disc

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

by default. Through this "overlay" approach, the BCCD's main asset is being able to convert multiple workstations running various operating systems into a coherent and powerful clustering environment. This means that the BCCD 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 for the teaching and learning of HPC.

2. What Makes the BCCD Different?

There are several clustering implementations to choose from when deciding on a parallel computing environment for HPC education. Clustering distributions, such as OSCAR[4] and NPACI-Rocks[15], add on or integrate with standard Linux distributions. While this approach benefits from pre-configured packages and cluster administration tools, it also requires dedicated resources. Dynamic (non-invasive) environments, which include ClusterKnoppix[16], CHAOS[13], PlumpOS, and LOAF[2] (Linux On A Floppy), do not require dedicated resources in that these drop-in clustering environments install nothing to the hard drive. In many instances, these environments only support a limited number of clustering paradigms (such as openMosix, but not PVM), and one cannot introduce new software or clustering applications to the environment (as in ClusterKnoppix, CHAOS, and Loaf).

The BCCD, on the other hand, combines the best of both worlds in that it requires no dedicated resources (no installation to the hard drive), no pre-configuration of packages, no administration, and includes a large customizable amount of various clustering and development environments.

3. How the BCCD Works

The BCCD is inherently customizable by design. The entire system is dynamically built from web-fetched sources in a cross-compiling process referred to as "gar". The build process brings together source code and local customizations in order to produce the final raw CDROM image. This allows special, customized images of the BCCD to be easily built, such as images that contain extra applications or images that require a personal private RSA key to unlock.

A group of workstations can be formed into a powerful clustering environment with the BCCD through a relatively simple process. Each BCCD-booted workstation utilizes the existing TCP/IP network to automatically configure the network environment. If a DHCP or DNS server is not present, another BCCD-booted workstation can be easily configured to offer those services.

The pkbcast process then begins to broadcast the public keys of each BCCD user/node. If a remote BCCD user accepts these keys in an appropriate way, the local BCCD user can leverage the corresponding private key (which is not disclosed) to execute commands and gain access to the remote machine. Many clustering tools require the ability to execute certain commands on remote machines without requiring the user to input his or her password. Using a public-key broadcasting approach with pkbcast, as well as other custom scripts provided with the BCCD to ease network configuration, formation of cluster "groups", and specialized bootup modes, hosts booted with the BCCD can be easily brought together into a single computational environment.

4. Contents of the BCCD

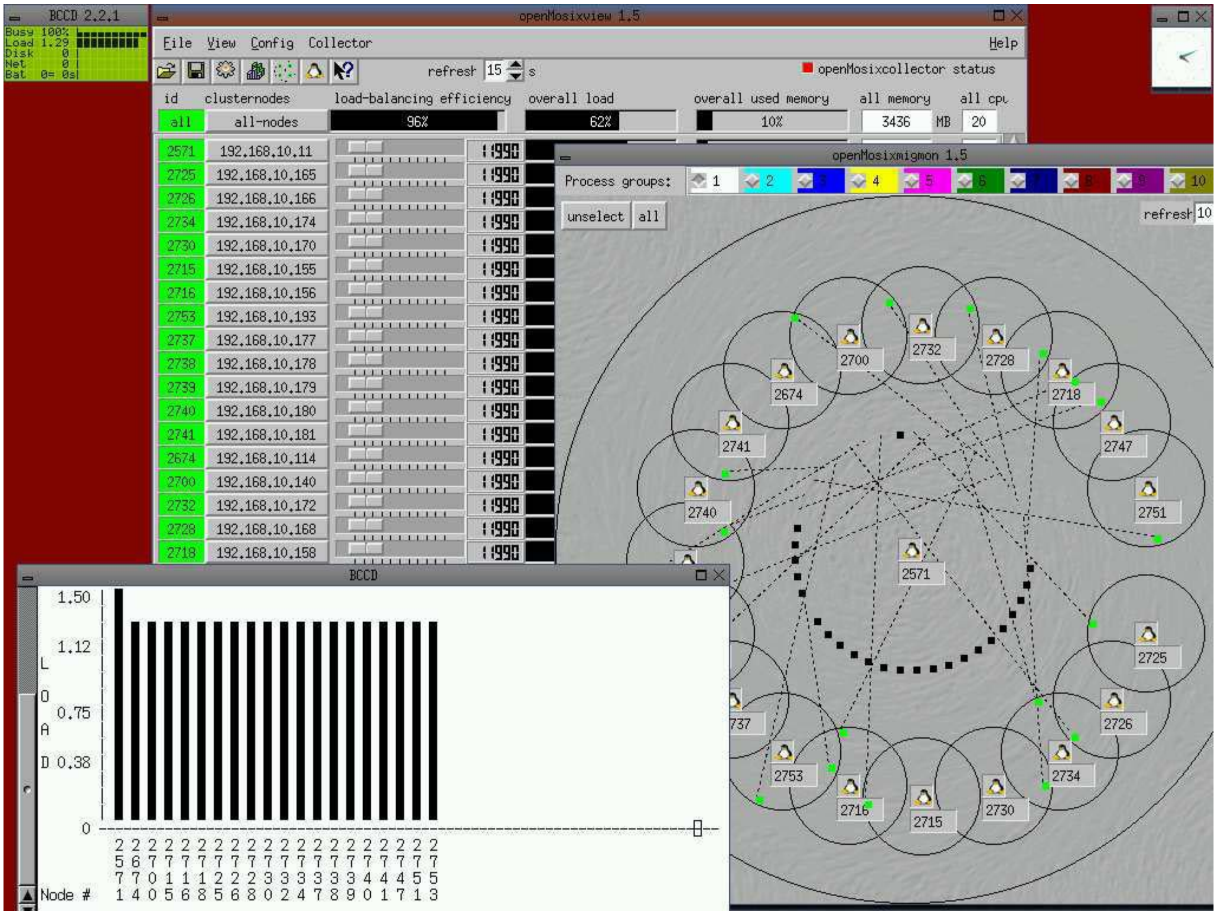
A wide breadth of pre-configured clustering applications are already included on the BCCD, such as openMosix[1], MPICH[10], LAM-MPI[3], PVM[7], and PVFS2[12].

OpenMosix support (as seen in Figure 1 below) provides kernel-level process migration and clustering capabilities across a BCCD-based cluster. 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. The version of openMosix running on the BCCD matches the running kernel; the openMosix-tools version is 0.3.5; the openMosixView distribution on the BCCD is version 1.5.

MPICH is the default MPI environment on the BCCD. Debugging tools are included, such as the MPE tools mpilog, mpitrace, and mpianim. A graphical visualization tools also exits called upshot. Multiple pre-configured example programs for MPICH already exist on the BCCD linked conveniently off the default home directory.

With a simple change to the "live" runtime system, or by creating a custom image, the MPI runtime environment can easily default to LAM-MPI support. LAM-MPI is an alternate MPI environment on the BCCD. Also included is XMPI, a graphical debugger that allows you to start MPI applications compiled with LAM-MPI and view message passing events during execution. And just like with MPICH, the BCCD contains multiple pre-configured example programs to be easily run with the LAM-MPI environment.

Support is also available for creating, compiling, running, and debugging programs under the PVM environment. Users can run PVM programs from the command line, from the pvm console, or from the graphical tool xpvm, which serves as a real time performance monitor for PVM tasks. Again, just like with MPICH and LAM-MPI, preconfigured PVM example programs are preloaded on the BCCD for educational convenience.



```
id     clusternodes       load-balancing efficiency   overall load   overall used memory   all memory   all cpu
all    all-nodes          96%                         62%            10%                   3436 MB      20
2571   192.168.10.11      1990
2725   192.168.10.165     1990
2726   192.168.10.166     1990
2734   192.168.10.174     1990
2730   192.168.10.170     1990
2715   192.168.10.155     1990
2716   192.168.10.156     1990
2753   192.168.10.193     1990
2737   192.168.10.177     1990
2738   192.168.10.178     1990
2739   192.168.10.179     1990
2740   192.168.10.180     1990
2741   192.168.10.181     1990
2674   192.168.10.114     1990
2700   192.168.10.140     1990
2732   192.168.10.172     1990
2728   192.168.10.168     1990
2718   192.168.10.158     1990
```

```
        2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
        5 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
        7 7 0 1 1 1 2 2 2 3 3 3 3 3 3 4 4 4 5 5
Node #  1 4 0 5 6 8 5 6 8 0 2 4 7 8 9 0 1 7 1 3
```

Figure 1. Process migration using openMosix on the BCCD. The openMosix paradigm is a task-based parallelization model. The above figure illustrates the seamless distribution of twenty tasks through the openMosix kernel scheduler. Distributed monitoring is shown through the `openmosixview` application (back) and in a terminal window using `mosmon` (front left). Using the `openmosixmigmon` interface (middle), users can drag-and-drop icons representing openMosix tasks from one resource to another.

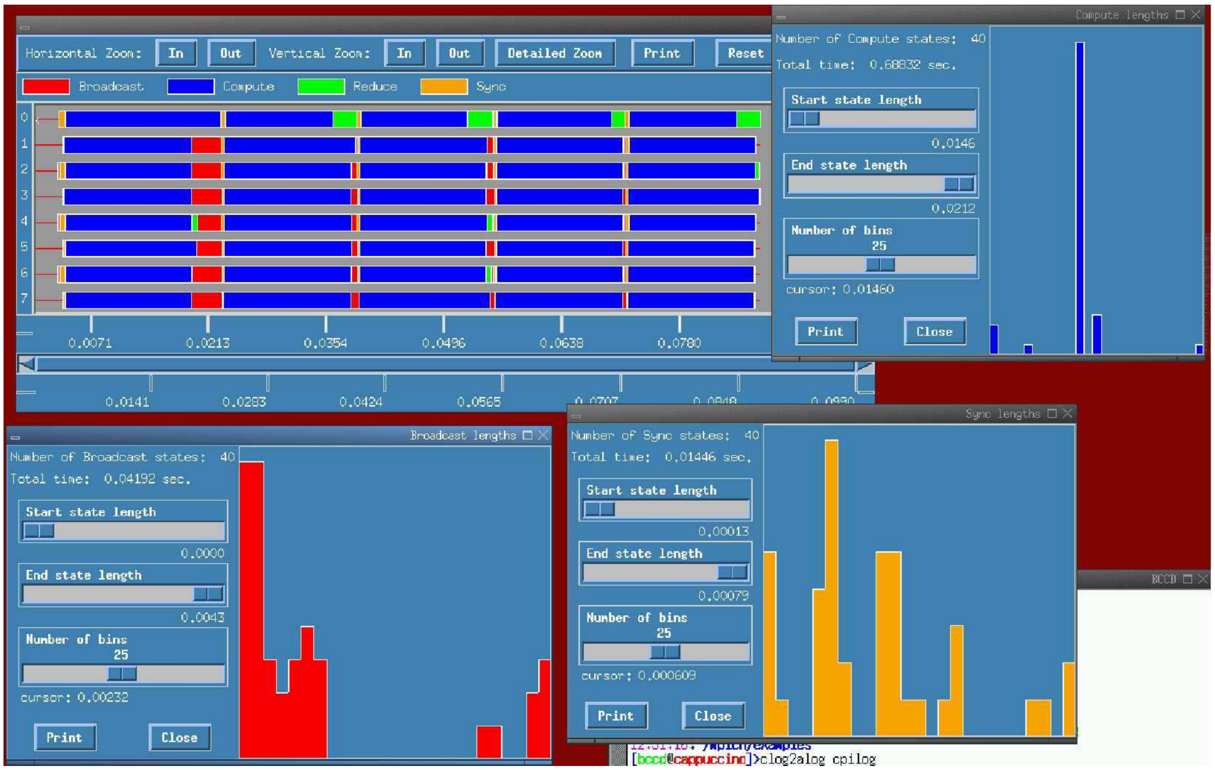
PVFS2 is an open-source, scalable parallel file system designed to scale to very large numbers of clients and servers. PVFS2 support can be used on the BCCD to support teaching aspects of filesystems related to parallel environments, access existing PVFS2 volumes served on another cluster, or to support filesystem distribution over a drop-in cluster environment. The BCCD currently contains PVFS2 version 0.8[1], integrated into the 2.4.25 Linux kernel.

The BCCD also includes C3 (Cluster Command Control, [6]) tools, GNU compiler suite, and openPBS. C3 tools were developed by the Oak Ridge National Laboratory and include tools for cluster-wide command execution, file distribution and gathering, process termination, remote shutdown and restart, and system image updates. A specialized BCCD bootup mode (discussed later) facilitates the use of C3 tools. OpenPBS is a flexible batch queuing system.

These applications are available without requiring configuration, installation, or administration by the end user(s). They have been provided on the BCCD image so that the focus can be on how to "use" a cluster instead of how to setup, administrate, and configure the clustering environment.

A full suite of development tools is available for supporting writing, debugging, and profiling distributed programs. Applications include a wide range of compilers, debugging libraries, visualization and debugging programs for distributed applications, linear algebra programs and libraries, and over 1400 additional applications.

[1] PVFS2 version 1.0.1 is integrated into the BCCD CVS tree



Number of Compute states: 40
Total time: 0.68832 sec.

Number of Broadcast states: 40
Total time: 0.04192 sec.

Number of Sync states: 40
Total time: 0.01446 sec.

```
12:51:10: /mpich/examples
[bccd@cappuccino]>clog2alog cpilog
```

Figure 2. `upshot` **being used on the BCCD to show a parallelized ring-based communication example.** `upshot` **shows computation, communication, and system overhead of the distributed MPICH process (top). Users can also view specific details about the properties of specific MPI commands such as broadcasts, scatter, gather, etc. through the upshot interface.**

Support to accommodate a multitude of filesystems, for storing and accessing files and filesystems is also available on the BCCD. Even though the BCCD environment itself is non-volatile, mechanisms exist to save and access work and data through a variety of means. These include floppy disks (direct mounting and accessible through the `mtools` package), USB drives, zip drives, local disk access all the way through support for remote mounting (and serving) of PVFS2, NFS and Samba shares. Of course, `scp`, `rsync`, `ftp`, `cvs` and other network-supported file copying tools are also available on the live BCCD image.

5. Features unique to the BCCD environment

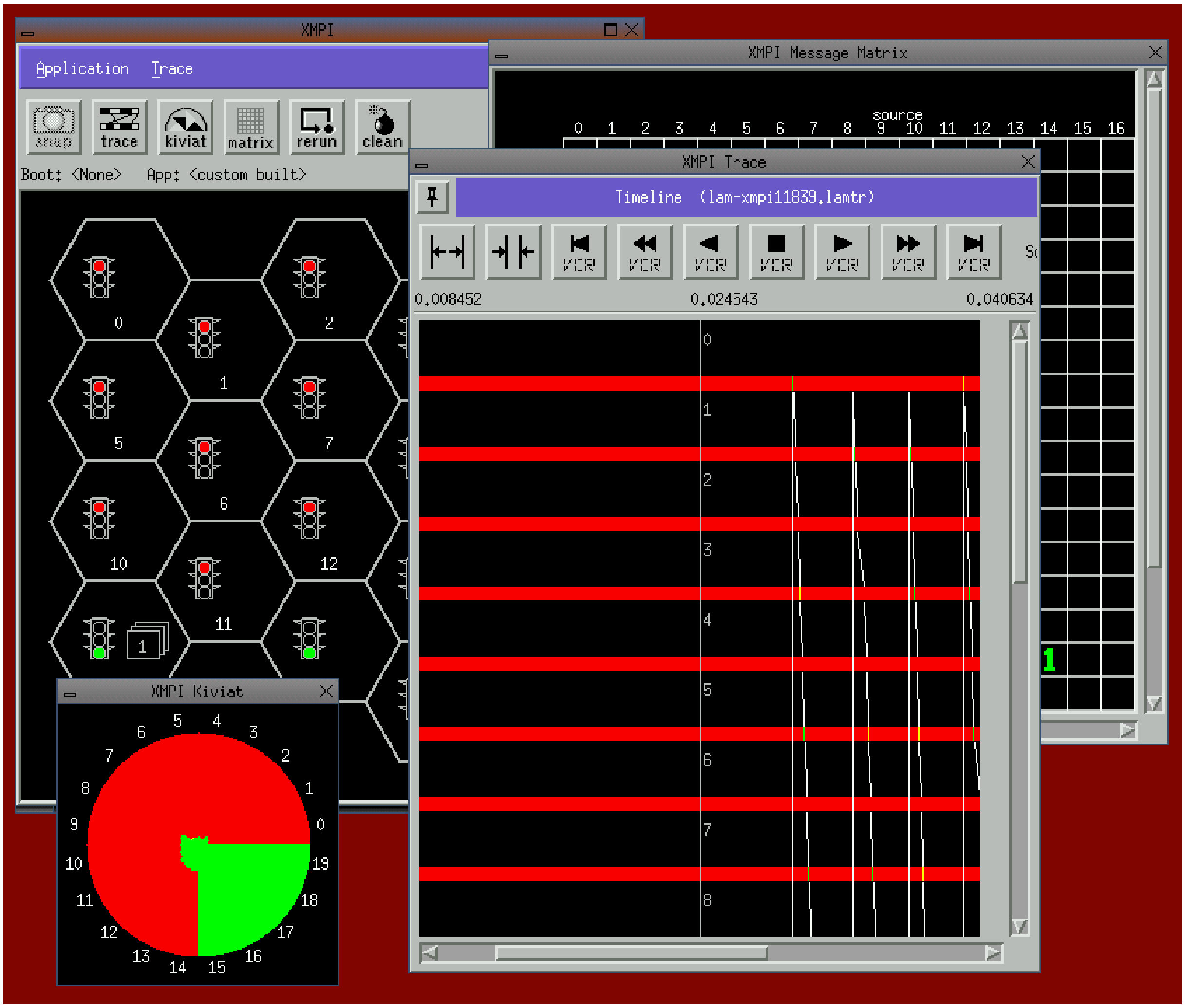
Support for hot-loadable software packages allows the BCCD to introduce new software and capabilities that were not included when the software was burned to the CD. Through hot-loadable software packages, users can dynamically add features to their runtime systems (e.g. Maui support or Ganglia monitoring) and tailor the runtime system to their local environments. This also affords instructors the flexibility to add, on-demand, additional capabilities to their classroom computing environment without the need to rebuild the BCCD image or burn additional CD media.

Extra bootup modes have been included in the BCCD for convenience and to ensure peak performance in environments with hardware incompatibilities and/or networking deficiencies.

5.1. Hardware Compatibility Modes

Hardware compatibility modes include:

- intelfb mode (for specific Intel framebuffers)
- i810fb mode (for the 810/815 framebuffer)
- nohotplug mode (disables the use of hotplug)
- and madwifi mode (pre-loads Atheros wireless drivers).



Boot: <None>   App: <custom built>

Figure 3. A LAM-MPI process as visualized through `xmpi`. `xmpi` provides graphical inspection tool to replay a parallel process. A user can visualize the sending and receiving of messages, determine the various states of the processes involved, and inspect the overall process collective's efficiency through the `Kiviat` tool.

5.2. Extra Networking Modes

Additional networking modes come into play depending upon the features available or lacking in the host environment. These modes include

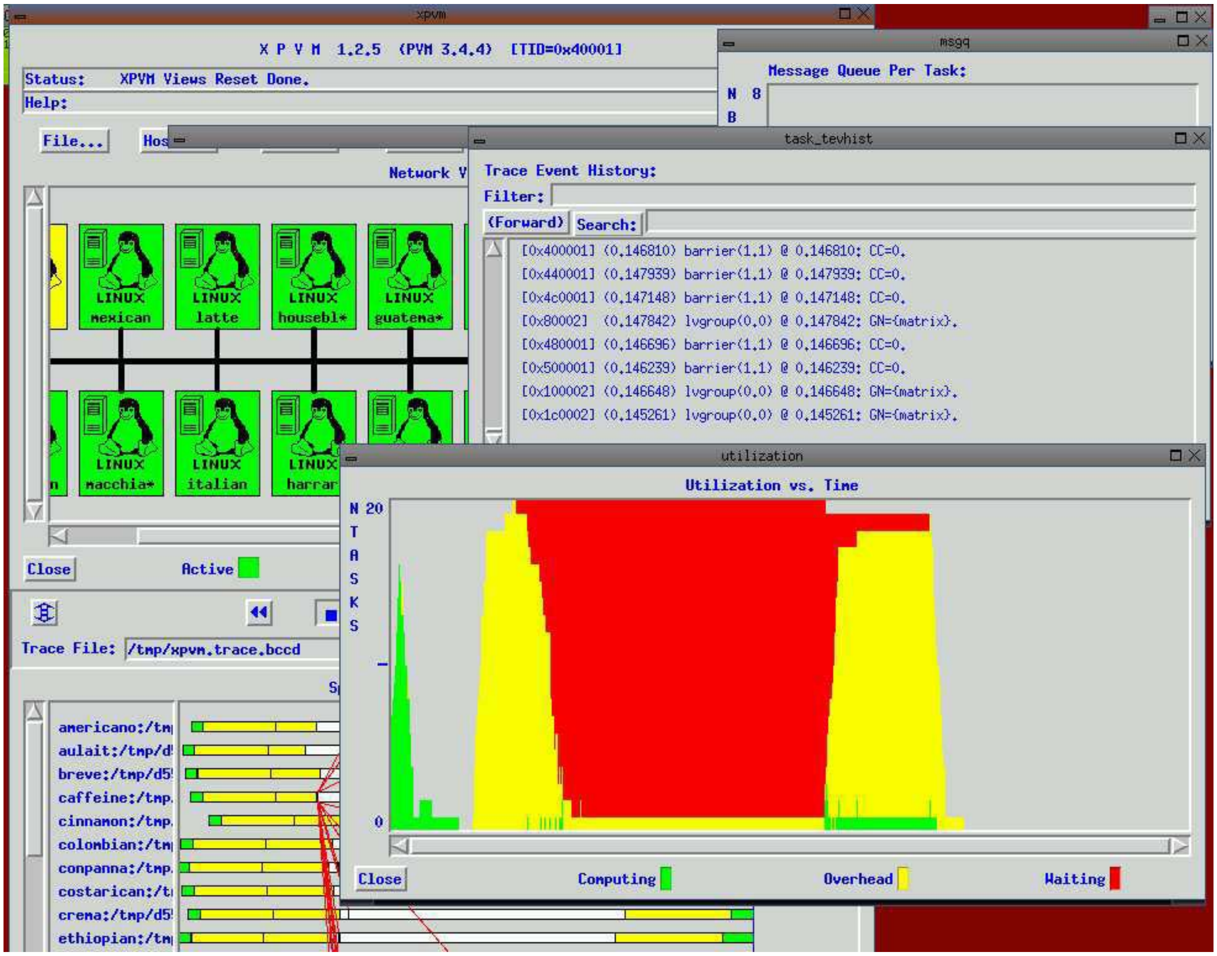
- nodemode
- startdhcp
- pxeserver

These modes support, respectively, environments where the BCCD environment serves as a computational drone, an environment that lacks a suitable DHCP server, and an environment where the BCCD image is to be served up through tftp and PXE-booting.

5.3. Convenience Modes

Convenience modes include:

- automode
- c3mode
- quickboot
- and runinram.



X P V M 1.2.5 (PVM 3.4.4) [TID=0x40001]

Status: XPVM Views Reset Done.

Help:

Trace File: /tmp/xpvm.trace.bccd

Message Queue Per Task:

Trace Event History:

```
[0x400001] (0.146810) barrier(1,1) @ 0.146810: CC=0.
[0x440001] (0.147939) barrier(1,1) @ 0.147939: CC=0.
[0x4c0001] (0.147148) barrier(1,1) @ 0.147148: CC=0.
[0x80002]  (0.147842) lvgroup(0,0) @ 0.147842: GN={matrix}.
[0x480001] (0.146696) barrier(1,1) @ 0.146696: CC=0.
[0x500001] (0.146239) barrier(1,1) @ 0.146239: CC=0.
[0x100002] (0.146648) lvgroup(0,0) @ 0.146648: GN={matrix}.
[0x1c0002] (0.145261) lvgroup(0,0) @ 0.145261: GN={matrix}.
```

Figure 4. `xpvm` **being used on the BCCD to join together four BCCD-based nodes in an illustration of PVM's group-based communication model.** `xpvm` **provides a graphical tool for manipulating the PVM environment. Using the** `xpvm` **interface, users can add and delete hosts from the PVM system, spawn tasks, explore message queues, examine output and debugging messages, and observe many other aspects of the parallel computation.**

Automode was designed to be used in the situation where a large cluster of many nodes is desired for a single user. c3mode allows the use of c3tools (mentioned previously). Quickboot mode boots up the BCCD without attempting to configure the network. This mode can also be leveraged for system rescue. Pxeserver mode requires a specially built image to host a pxeserver which enables the BCCD to network boot. Finally, runinram mode allows the BCCD to run completely off the host system's RAM, freeing up the CDROM for other uses.

Group scripts, such as `bccd-joingroup`, `bccd-leavegroup`, and `bccd-removegroup`, are another new feature that allows groups of users to form their own "subclusters" within the larger, networked BCCD environment. In this way, certain nodes can be picked and chosen to participate in a mini cluster while other nodes are left free. This has been especially useful in classroom and workshop settings in which small groups of users want to participate in their own experiments.

6. Building of the BCCD

The BCCD is built from web-fetch sources in a manner similar to the BSD Ports system[5], or Gentoo's Portage system[8]. The actual build environment of the BCCD is known as "gar[2]." Gar itself is not an acronym, but rather

[2] "gar" was developed in support of the LNX-BBC[14] project.

an expression of angst. The "gar" build system also sets the BCCD apart from other projects. Unlike Ports or Portage, "gar" does not build applications for the native (host) system, but rather cross-compiles all applications, with the live CDROM image as the target image. Fundamentally, one could build the i386 version of the BCCD on a PowerPC platform, and vice versa, using the cross-compilation functionality of the gar build platform.

7. BCCD-supported Workshops in High Performance Computing Education

The BCCD project is a production-quality tool for supporting HPC investigations. Since the first release image two years ago, the BCCD project has matured and evolved accordingly, so as to support a multitude of newer hardware, new releases of programming environments[3], and new projects and tools, such as PVFS2, in support of distributed computing investigations.

An open lab of networked workstations, laptops connected over a wireless network (as was used at Supercomputing 03's Education program), or practically any situation where networked workstations are available serves as a suitable environment for explorations in clustering environments. Two areas where the BCCD has excelled in HPC education include training students in clustering administration (instead of opening up the production systems) and for supporting educator training workshops sponsored by the National Computational Science Institute (NCSI).

Over the past two years, NCSI has leveraged the BCCD as a means to support HPCE workshops during Supercomputing 03's Education Program (Nov. 2003), at the Oklahoma Supercomputing Center for Education and Research (OSCER) (Sept. 23-26, 2003 and Aug. 8-15, 2004), at Contra Costa College (June 7-9, 2004), at Bethune-Cookman College, and was also used during Supercomputing 2004's Education Program. As an example of the ability to scale, the BCCD was used to bootstrap a 60-node cluster at Contra Costa College during the Sigma Xi workshop on High Performance Computing in the undergraduate curriculum, June 2004.

During these workshops educators from research institutions, primarily-undergraduate institutions and community colleges gather to learn, discuss, and develop curricular topics drawn from HPC and parallel computing. Workshop session topics have included:

- traditional message-passing-based programming environments (MPICH, LAM-MPI and PVM)
- pragma-based and evolving environments such as OpenMP, threads, openMosix and JavaSpaces
- pedagogical issues
- classroom activities and curricular issues relating to HPC
- debugging, tracing, and visualization of distributed programs
- building clusters from scratch, from distribution-based solutions (NPACI ROCKS and OSCAR) and through non-destructive approaches such as the BCCD
- setup, configuration, and maintenance of clustering environments and more.

Presentations and workshops such as those described above have also spurred efforts toward providing a similar environment for *Grid Computing* education. Grid computing poses an even more ominous resource, software, and environmental demands than traditional distributed computing paradigms. Efforts are underway to support a self-contained Shibboleth [11] portal along with fundamentals for building grid services which would include support for Tomcat and an appropriately pre-configured Apache server.

8. Conclusions

In summary, the goal of the Bootable Cluster CD (BCCD) project is to encourage and support High Performance Computing Education at all academic levels. The BCCD is self-described as a "Computational Science wet lab for distributed computing," providing a powerful research, experimentation, and debugging environment for investigations in HPC.

The development of the BCCD has led to many refinements over the years, so as to support ease of use, convenience, and most importantly, the education of High Performance Computing issues in the undergraduate curriculum. The BCCD provides students and instructors with a working high performance computing development, debugging and testing environment with minimal infrastructure and administrative demands. Extensive documentation, curricular units, project sources and downloadable BCCD images are available from http://bccd.cs.uni.edu.

References

[1] M. Bar. Linux clusters state of the art. Online document available at http://openmosix.sourceforge.net, 2002.

[2] M. Blomgren and M. A. M. Jr. openMosixLoaf. Information available at the project web site http://openmosixloaf.sourceforge.net, 2003.

[3] G. Burns, R. Daoud, and J. Vaigl. LAM: An open cluster environment for MPI. In *Supercomputing Symposium '94*, Toronto, Canada, June 1994. Source available at http://www.lam-mpi.org.

[3] For example, MPICH-2.0 is optional in the latest BCCD builds.

[4] B. des Ligneris, S. Scott, T. Naughton, and N. Gorsuch. 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.

[5] FreeBSD. The FreeBSD ports system. Information available at the project web site http://www.freebsd.org/ports, 2005.

[6] A. Geist, J. Mugler, T. Naughton, and S. Scott. Cluster command and control (c3) tools. Information available at the project web site http://www.csm.ornl.gov/torc/C3/, 2003.

[7] G. A. Geist and V. S. Sunderam. The PVM system: Supercomputer level concurrent computation on a heterogeneous network of workstations. In *Proceedings of the Sixth Distributed Memory Computing Conference*, pages 258–261. IEEE, 1991.

[8] Gentoo Linux. Gentoo linux portage development. Information available at the project web site http://www.gentoo.org/proj/en/portage/index.xml, 2005.

[9] P. Gray. The bootable cluster cd. Information available at the project web site http://bccd.cs.uni.edu, 2004.

[10] W. Gropp, E. Lusk, and A. Skjellum. A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing*, 22(6):789–828, 1996. Source available at http://www.mcs.anl.gov/mpi/mpich.

[11] Internet2. Shibboleth architecture protocols and profiles. Technical report, The Ohio State University, Nov. 2004. Document available at http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-protocols-05.pdf.

[12] R. Latham, N. Miller, R. Ross, and P. Carns. A next-generation parallel file system for linux clusters. *LinuxWorld Magazine*, pages 56–57, Jan. 2004. ITSecurity Group publication.

[13] I. Latter. Running clusterknoppix as a head node to a CHAOS drone army. Technical report, Macquarie University, AU, Dec. 2003. ITSecurity Group publication.

[14] Moffit, N. et.al. The lnx-bbc project. Information available at the project web site http://lnx-bbc.org, 2004.

[15] SDSC (UCSD) and Millennium Group (Berkely). Npaci rocks. Information available at the project web site http://www.rocksclusters.org/Rocks/, 2003.

[16] W. Vandersmissen. Clusterknoppix. Information available at the project web site http://bofh.be/clusterknoppix, 2004.