

# Building of a GNU/Linux-based Bootable Cluster CD\*

Paul Gray<sup>1</sup>, Jeff Chapin<sup>1</sup>, and Tobias McNulty<sup>2</sup>

<sup>1</sup> University of Northern Iowa gray@cs.uni.edu, chapinj@uni.edu

<sup>2</sup> Earlham College tmcnulty@ppckernel.org

**Abstract.** The Bootable Cluster CD (BCCD) is an established, well maintained, cluster toolkit used nationally and internationally within several levels of the academic system. During the Education Programs of Supercomputing conferences 2002, 2003, and 2004, the BCCD image was used to support instruction of issues related to parallel computing education. It has been used in the undergraduate curriculum to illustrate principles of parallelism and distributed computing and widely used to facilitate graduate research in parallel environments. The standard BCCD image is packaged in the 3", mini-CD format, easily fitting inside most wallets and purses. Variations include PXE-bootable (network-bootable) and USB-stick bootable images. All software components are pre-configured to work together making the time required to go from boot-up to functional cluster less than five minutes. A typical Windows or Macintosh lab can be temporarily converted into a working GNU/Linux-based computational cluster without modification to original disk or operating system. Students can immediately use this computational cluster framework to run a variety of real scientific models conveniently located on the BCCD and downloadable into any running BCCD environment. This paper discusses building, configuring, modifying, and deploying aspects of the Bootable Cluster CD.

## 1 A Brief History of the Bootable Cluster CD

The original impetus for a self-contained, pre-configured, cluster image that could leverage an ever-increasing number of networked computer laboratories in support of high performance computing education began with the 2001 Supercomputing conference Education Program in Dallas, Texas. The amount of effort involved in coordinating the installation and configuration of software and services on hardware that was to be provided sight-unseen was enormous. Earlier in the year 2000, a company called LinuxCare put out a small, business-card-sized recovery CD that provided a very resilient and powerful bootable GNU/Linux operating system in a small 50MB image. Following Supercomputing 2001, the Bootable Cluster CD[9] project began as Dr. Gray joined with the LinuxCare developers that had forked off the "LNX-BBC<sup>1</sup>" [10] project and the Bootable Cluster CD project began.

By the summer of 2002 a fully-functional BCCD image was available. At Supercomputing 2002 in Baltimore Maryland, the Bootable Cluster CD was mature enough to host a drop-in clustering environment and to support the parallel computing education sessions held during the SC02 Education Program. Since that time, the Bootable Cluster CD has hosted numerous workshops, ranging from National Computational Science Institute workshops on Clusters and Parallel Programming to supporting workshops hosted by the National Center of Excellence for HPC Technology (NCEHPCT).

The educational impact of the Bootable Cluster CD is far from trite. It is reflected in the ability to take a completely unconfigured lab of networked workstations and, within

---

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

<sup>1</sup> As of November, 2005, Dr. Gray has taken the role of project leader for the LNX-BBC project.

a few minutes, create a fully-functioning computational cluster complete with application profiling support through perfctr[17] and PAPI[6], MPI support with MPICH[11] or LAM-MPI[3], PVFS2[16] filesystem support, and applications like Gromacs[13] and gmxbench[12] at the ready. The advantages of the BCCD approach include:

- There is no required formatting of the hard disk
- There is no required operating system installation.
- The system can run completely in RAM and the CDROM image. No portion of the computer's disk is modified.
- Students have the same environment in and outside of class, as well as in a production environment.
- Educators can focus on curricular issues rather than cluster administration issues.
- Production servers with minimized administration and maintenance can be created, reducing cost, and allowing seamless migration of skills from an educational environment to a real world one.

## 2 Description of the Bootable Cluster CD

The goal of the Bootable Cluster CD is to lend support to students, educators and researchers as they gain insight into configuration, utilization, troubleshooting, debugging, and administration issues uniquely associated with parallel computing. As the name implies, the BCCD provides a full, cohesive clustering environment running GNU/Linux when booted from the CDROM drives of networked workstations. The BCCD is unique among bootable clustering approaches in its ability to provide a complete clustering environment with pre-configured clustering applications and examples from a full repertoire of development tools.

An open lab of networked workstations, which also includes environments using 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 three years, NCSI has leveraged the BCCD as a means to support HPCE workshops at Washington University, St. Louis (June 2003), the OU Supercomputing Center for Education and Research (OSCER) (Sept. 23-26, 2003, Aug. 8-15, 2004, and July 1-Aug.6, 2005), at Contra Costa College (June 7-9, 2004), at Bethune-Cookman College, and Wofford College (Nov. 2004). At the Contra Costa workshop, the BCCD was used to bootstrap and sustain computational simulations across a 60-node BCCD cluster.

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.

The wide breadth of pre-configured clustering applications available on the BCCD includes openMosix and openmosixview; PVFS2; PVM[8]; XPVM; MPICH; LAM-MPI; C3-tools[7]; GNU compiler suite; torque[4]; and Gromacs.

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. This approach also allows the BCCD to be used in an educational setting where the cost of a traditional cluster would prove prohibitive due to administrative overhead.

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.

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.

### 3 Building the Bootable Cluster CD

The approach taken to build the Bootable Cluster CD provides it with a unique customizable framework for the end user. This flexibility is being leveraged to expand and customize the features of the standard BCCD in support of cutting-edge topics in distributed computing which will be discussed in the following section.

The current version of BCCD has the following characteristics:

- The BCCD uses the LNX-BBC mechanism to build the operating system into a compressed CD image. The BCCD contains over 500 megabytes of operating system and applications compressed to a 180 megabyte mini-CD.
- The ability to boot from virtually any x86 (PC) and the ability to boot from several PowerPC (Macintosh) versions.
- Ethernet drivers are provided for many network interface cards (NICs) including the latest Gigabit Ethernet drivers.
- The ability to build specialized ISO images, the file representation from which a CD image can be pressed (with some human interaction).
- The BCCD automatically configures networking properties, populates forward- and reverse-dns lookup tables, and dynamically sets up distributed ssh key-based authentication required for the MPI and PVM parallel environments.

- It is by default completely non invasive, i.e. the hard drive is never accessed; only computer memory is used, thus no lasting effects after rebooting, however accessing the hard drive for storage and other purposes is currently available, while an installation system is in the works.
- A BCCD system boots in minutes using only 9 presses of the enter key in most cases.
- On systems with a reasonable amount of system RAM, the image can be loaded from the CD image completely into system RAM. This allows one CD to be used to bootstrap up an entire lab of workstations into the BCCD environment.
- Support for PXE- and USB pen-drive-based booting.

The LNX-BBC mechanism from which the BCCD build process is derived uses a custom creation system called GAR<sup>2</sup>. GAR is used to help automate the creation of bootable images for distribution and use, and also aides the construction of bootable images in a variety of formats. Sharing some similarity with BSD Ports and Gentoo’s emerge package building mechanisms, GAR allows for distributed storage of package sources and automates the build process, fetching the sources needed for a package. GAR cross-builds dependencies and cross-compiles the packages for the target BCCD host image. Since everything is custom compiled, the same repositories can be used regardless of the architecture you are targeting; presently x86 or PowerPC.

Having a standard, self-contained CD image and the non-invasive nature of the project result in several benefits such as the ability to provide an identical environment on each machine, which allows students to work virtually anywhere, to support deep explorations of parallel environments, and to support the use of machines where a traditional clustering environment would not be feasible, turning idle cycles and minds into useful ones.

The build process of the BCCD separates it from other approaches taken by cluster deployment and bootable GNU/Linux CD images. These attributes of the build process are revisited in more detail in the next section which contrasts the BCCD with other projects that share limited similarities with the BCCD.

## 4 Comparison with other Bootable Images and Cluster Imaging Solutions

There are many other “self contained” clustering solution environments available, ranging from OSCAR[5], Rocks[18], and Warewulf[15] to Loaf[2], Linux on a floppy. What makes the BCCD unique amongst all alternatives is it’s three pronged focus on education, customization and non invasiveness.

There are also many bootable GNU/Linux CD images available today. These include a growing multitude of “Live-CD” variations of popular GNU/Linux distributions. Other bootable images include variations of the popular Knoppix[14] bootable CD image or custom-built images. Adding software or features to these images typically requires one to rip the fundamental components of the bootable image apart and introduce compatible binaries to the image by hand, which requires super-user permissions; rolling all components together afterwards, back into a workable bootable CD format. ClusterKnoppix[19] is an example of a Knoppix extension that adds openMosix[1] functionality and management tools to the base Knoppix image.

<sup>2</sup> A common question is what “GAR” stands for. However, “GAR” is not an acronym, but an expression derived from the frustration one encounters when software packages fail to build.

In contrast, the GAR system used by the BCCD allows more flexibility to the end user for customization of the runtime system. An end user, with non-root privileges, can build a complete BCCD image from the CVS code base without additional privileges. A complete CD image can be built via web-fetched sources without the need to reverse engineer an existing CD image. GAR pulls together an approach that is built upon a mixture of BSD's ports, Linux from scratch, Gentoo's emerge, and user mode Linux to support the creation of a dynamic and customized bootable image built from web fetched sources with only user permissions.

The process of building the BCCD image involves three distinct components:

1. The host system's build environment
2. The target's runtime environment
3. The toolchain environment which forms the bridge between the above components.

The utilization of a cross compiler toolchain allows for the greatest breadth of platform support when building the Bootable Cluster CD image. In general, the formal BCCD image is built with the largest set of features and compatible with the largest set of target hosts. For example, by default all binaries compiled for a target x86-based environment are i386-compatible and contain uni-processor kernels. This allows the BCCD image to run on a large set of legacy hardware as well as on the latest x86\_64 platforms. If one desires a specific extension to the default paradigm, which would otherwise break the universality of the BCCD image, the BCCD build tree can be checked out from the CVS archive and these extensions can be integrated manually or the web-based BCCD build portal, discussed later in this document, can be used to automate customized images.

For a software component, MPICH for example, to run on the BCCD's target runtime environment, the following steps are taken by the GAR build system:

1. a cross compiling toolchain is built by fetching the requisite binutils, libc, gcc, and other components for the task of compiling binaries for the target environment. For example, an i686-PPC toolchain or even an i686-i386 toolchain is built from scratch to begin the process.
2. the source code for the package is obtained from an established repository location. Typically these repositories are the official package download locations.
3. host- and cross-tools needed to build the package for the target environment are obtained in source-code form and compiled similarly.
4. any runtime dependencies of the package (a graphics or compression library such as libz, for example) are obtained in source-code form and compiled in the appropriate manner.
5. any necessary patches are applied and the package is built using the appropriate toolchain.

End users that wish to modify the BCCD can easily add more packages to the BCCD runtime image, customize their own services during the build process or even strip away components. For end users that aren't willing or able to build their own customization, a web portal has been developed that offers users a select-and-build BCCD configuration that automates the building of a BCCD image with selected packages and custom configuration files, with guidance as to sorting out dependencies.

## 5 BCCD as a Small Scale Production Environment

One of the limitations on educating students in HPC is access to a cluster, particularly one with short queue times. The principal reason that many institutions are unable to provide a cluster is mainly financial, namely that both the hardware and the administration required of a cluster are often prohibitively expensive. The BCCD provides a solution to both of these issues, through a development called "Liberation". Liberation allows for the installation of the BCCD onto a more permanent cluster. One of the benefits of this approach is the reduction in time invested in maintenance. Once an BCCD image is created, a cluster administrator merely needs to reliberate the software, allowing more time to be devoted to educational issues rather than administrative, letting the BCCD project maintainers manage all the software compatibility and management issues. Additionally the natural versatility of the BCCD allows for a wide variety of cheap commodity hardware to be used as the infrastructure to install upon.

## 6 Grid Education and the Future Directions of the BCCD

Previous sections emphasized the process used by the BCCD to build images which allows a great deal of customizability and does not require special user privileges. This section highlights the future direction of the BCCD project as it leverages this flexibility going forward to support the larger efforts of Grid-based education.

Installation, configuration and ongoing maintenance of Grid services is an arduous task that often precludes our ability as instructors to bring Grid computing topics into the classroom or for hosting Grid-content driven workshops. A natural extension to the traditional BCCD image is one that focuses on aspects of Grid education, namely a Bootable Grid CD (BGCD). Elevating the BCCD paradigm to one that is able to prominently feature the capabilities of a Grid system requires a significant and fundamental integration of user authorization and credential verification. This is where the build process of the BCCD can be uniquely leveraged to offer *customized bootable images* that have been *created uniquely for a specific user's credentials*. Work is under way to provide the community with a web portal for automating the task of creating customized BCCD images that can integrate one's personal Grid credentials into the final image. These capabilities have the potential to significantly impact our ability to support Grid-based educational workshops and Grid-centric curriculum. The goal of these efforts is to establish a paradigm for a classroom or workshop of participants to be given an individualized Bootable Grid CD which would allow them to authenticate and participate in Grid-based computations from any networked workstation capable of booting the BGCD image.

## 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. CLUSTER RESOURCES, INC. The torque resource manager. Information and software available at the product web site <http://www.clusterresources.com/pages/products/torque-resource-manager.php>, 2005.
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. DONGARRA, J., LONDON, K., MOORE, S., MUCCI, P., AND TERPTRA, D. Using PAPI for hardware performance monitoring on linux systems. In *Conference on Linux Clusters: The HPC Revolution* (Linux Clusters Institute, Urbana, Illinois, June 2001). Web site: <http://icl.cs.utk.edu/papi/>.
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. GRAY, P., MOFFIT, N., RIFFE, N., AND SCHOEN, S. The lnx-bbc project. Information available at the project web site <http://lnx-bbc.org>, 2004.
11. GROPP, W., LUSK, E., AND SKIHELLUM, 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>.
12. HESS, B., LINDAHL, E., AND VAN DER SPOEL, D. Gmx benchmarks: The gromacs benchmarking suite. Information available at the project web site <http://www.gromacs.org/benchmarks/index.php>, 2005.
13. HESS, B., LINDAHL, E., AND VAN DER SPOEL, D. Gromacs: A package for molecular simulation and trajectory analysis. Information available at the project web site <http://www.gromacs.org>, 2005.
14. KNOPPER, K. Knoppix. Information available at the project web site <http://knoppix.org>, 2003.
15. KURTZER, G. How warewulf works (a look into the tools). Information available at the project web site <http://warewulf-cluster.org>, 2003.
16. LATHAM, R., MILLER, N., ROSS, R., AND CARNS, P. A next-generation parallel file system for linux clusters. *LinuxWorld Magazine* (Jan. 2004), 56–59.
17. PETERSSON, M. Performance counter support, “perfctr”. Information available at the project web site <http://user.it.uu.se/~mikpe/linux/perfctr/>, 2005.
18. SDSC (UCSD) AND MILLENNIUM GROUP (BERKELY). Npaci rocks. Information available at the project web site <http://www.rocksclusters.org/Rocks/>, 2003.
19. VANDERSMISSEN, W. Clusterknoppix. Information available at the project web site <http://bofh.be/clusterknoppix>, 2004.