XCBC and XNIT – tools for cluster implementation and management in research and training

Jeremy Fischer
IU Pervasive Technology Institute
jeremy@iu.edu

Richard Knepper
IU Pervasive Technology Institute
rknepper@iu.edu

Eric Coulter
IU Pervasive Technology Institute
jecoulte@iu.edu

Charles Peck
Cluster Computer Group
Earlham College
charliep@cs.earlham.edu

Craig A. Stewart
IU Pervasive Technology Institute
stewart@iu.edu

ABSTRACT

The Extreme Science and Engineering Discovery Environment has created a suite of software designed to facilitate the local management of computer clusters for scientific research and integration of such clusters with the US open research national cyberinfrastructure. This suite of software is distributed in two ways. One distribution is called the XSEDE-compatible basic cluster (XCBC), a Rocks Roll that does an "all at once, from scratch" installation of core components. The other distribution is called the XSEDE National Integration Toolkit (XNIT), so that specific tools can be downloaded and installed in portions as appropriate on existing clusters. In this paper, we describe the software included in XCBC and XNIT, and examine the use of XCBC installed on the LittleFe cluster design created by the Earlham College Cluster Computing Group as a teaching tool to show the deployment of XCBC from Rocks. In addition, the demonstration of the commercial Limulus HPC200 Deskside Cluster solution is shown as a viable, off-the-shelf cluster that can be adapted to become an XSEDE-like cluster through the use of the XNIT repository. We demonstrate that both approaches to cluster management – use of SCBC to build clusters from scratch and use of XNIT to expand capabilities of existing clusters – aid cluster administrators in administering clusters that are valuable locally and facilitate integration and interoperability of campus clusters with national cyberinfrastructure. We also demonstrate that very economical clusters can be useful tools in education and research.

Categories and Subject Descriptors

Theory of computation - Parallel computing models; Computer systems organization - Grid computing; Computer systems organization - Special purpose systems

Keywords

Cluster Management, computational science education, systems administration education, campus bridging, sysadmin training, cluster training, cluster system software, cluster packaging, open source clusters, XSEDE, Rocks, rolls, cluster, Linux, RedHat, CentOS, rpm, yum, campus, bridging, research, LittleFe

1. INTRODUCTION

An NSF Advisory Committee for Cyberinfrastructure taskforce found in 2011 that [2]

> *The current state of cyberinfrastructure software and current levels of expert support for use of cyberinfrastructure create barriers in use of the many and varied campus and national cyberinfrastructure facilities. These barriers prevent the US open science and engineering research community from using the existing, open US cyberinfrastructure as effectively and efficiently as possible.*

For many researchers in computationally-oriented science, those barriers are due to the state of software running on their local clusters. Since the NSF ACCI taskforce came out in 2011, demand for computational resources has continued to grow, and federal investment in cyberinfrastructure for open research has stagnated. This means that campus-based resources are now even more important to advances in open science research than when the report was finalized in 2011. The scale of the problems identified by the taskforce in 2011 are, barring improvements in the way cyberinfrastructure software is developed, going to be more of a challenge as demand continues to outstrip the pace of federally-funded resources [3].

Clusters are often set up in ways that are just good enough to get the job done, and often not in ways that are optimal for enabling discovery. Such clusters are generally administered by people with too much to do and too little time. Maintenance of clusters is often an ongoing challenge. For practicing scientists, differences in the way clusters are set up is often a source of frustration and time lost reading documentation, simply to figure out how to run a particular application or submit a job on a particular cluster.

The Extreme Science and Engineering Discovery Environment (XSEDE) Campus Bridging group has developed the concept of an "XSEDE-compatible basic cluster" (XCBC) build [4, 5] – a build that enables the creation of a cluster entirely from open-source tools modeled after the clusters supported by XSEDE. XSEDE supports a large group of clusters and supercomputers – more than a dozen clusters and more than 13 PetaFLOPS of computational capacity [6], used by thousands of researchers in the US and supported by more than 250 FTEs (Full Time Equivalents) of professional and academic staff [7]. In the absence of any single model for setting up clusters, using the clusters in this major national cyberinfrastructure (CI) facility as a model can create consistency that helps researchers.

XCBC builds on and currently depends on the very successful Rocks project [8]. Early reaction to XCBC was very positive but there were two very clear community reactions for capabilities not supported in the initial Rocks-based implementation of XCBC. First, many members of the community wanted to be able to add specific software tools and capabilities consistent with XSEDE clusters without starting from scratch. Second, many members of the US research and cyberinfrastructure communities wanted the ability to add into the standard XCBC build tools that were needed by the community but not necessarily a part of the basic software installation on a typical XSEDE-supported cluster. These viewpoints were expressed particularly strongly by two very important groups of community representatives: XSEDE Campus Champions and participants in the ACI-REF (Advanced

Cyberinfrastructure – Research and Education Facilitators) project. The XSEDE Campus Champions serve on their own campuses as local experts in national cyberinfrastructure and within XSEDE as experts on campus computing needs [9]. There are more than 250 individuals at more than 200 institutions of higher education, representing every state in the US and the District of Columbia, Puerto Rico, and the US Virgin Islands. ACI-REF is a national consortium of institutions of higher education “dedicated to forging a nationwide alliance of educators to empower local campus researchers to be more effective users of advanced cyberinfrastructure” [10].

As a result of community input, we created the XSEDE National Integration Toolkit (XNIT). XNIT is based on the Yum repository for installation or updates of RPMs [11, 12]. XNIT includes all of the software included in the standard XCBC build, and more. A cluster administrator may benefit by using the XNIT Yum repository simply as a source for updates of allocations software of interest to a given researcher. XNIT and the Yum repository make it easy for campus cluster administrators to do one-time installations of any particular software capability they want within the suite of the XNIT set, and to subscribe if they wish to automatically be notified of updates to particular packages. XNIT also includes software not included in the basic XCBC build – this will be increased over time in response to community requests.

XCBC and XNIT enable economies of scale in cluster administration and in user support because many of the documents and user training materials prepared for XSEDE can be repurposed and reused to support a campus-based cluster. The commands used to execute open-source applications on any cluster created with XCBC or XNIT are compatible with the way these commands are used on a typical cluster supported by XSEDE. A user’s knowledge of software, system commands, etc., becomes portable from one cluster built with XCBC to another, and to XSEDE-supported clusters generally. This makes it easier for a researcher to move from an XCBC- or XNIT-based campus cluster to an XSEDE-supported resource available to the national open research community, and from one cluster to another. XNIT in particular enables such compatibility to be added to an existing, operating cluster in part or in whole, without changing the pre-existing cluster setup.

In this paper, we describe the current state of the tools included in XCBC and XNIT. We provide an overview of current cluster installations using XCBC and XNIT for research. We also describe at length the use of XCBC in training for systems administrators, using two modestly-priced and luggable clusters: LittleFe and Limulus HPC200. LittleFe (“little iron”) is a popular, inexpensive, build-it-yourself option for building a small (6-node) cluster for training or production. The Limulus HPC200 is a “cluster in a deskside case” available commercially and intended to support individual computational scientists. Finally, we discuss the suitability of LittleFe and Limulus HPC200 as personal deskside clusters.

Our purposes for presenting this material are to: provide cluster administrators with information about the utility of XCBC and XNIT so that they can consider adopting these tools, provide information on tools that ease cluster administration, inform cluster administrators about the XNIT Yum repository as a source of RPMs for dozens of useful software packages, and provide information that will enable better training for current and future cluster administrators. The set of tools we describe here can, if adoption continues and grows, help improve the ability of US researchers to make new discoveries by enabling a more consistent national cyberinfrastructure. Ultimately, our goal in this work is to ease use of campus cyberinfrastructure for practicing scientists and students throughout the US and beyond, and to simplify migration between campus and national cyberinfrastructure, such as the facilities supported by XSEDE.

2. XCBC AND XNIT BUILD CONTENTS

The XCBC and XNIT basic builds include a set of open-source tools sufficient to set up and operate a cluster from scratch. Full listings of the software tools available within XCBC and the XNIT Yum repository are available online [13]. Highlights of the software included in the current XCBC release (0.9) are described in Table 1 and Table 2. Table 2 focuses on applications that are kept consistent with versions in use on XSEDE (using the current Stampede system [14] as the definition of “current best practices” for XSEDE clusters). In particular, libraries are in the same place as on XSEDE clusters, versions are the same, and commands work as they do on XSEDE-supported clusters.

Table 1. Components of current XCBC build Part 1 – General cluster setup (mostly existing Rocks optional rolls)

Category	Specific packages
Basics	Rocks 6.1.1, Centos 6.5, modules, apache-ant, fdepend, gmake, gnu-make scons
Job Management	Torque, SLURM, sge (choose one)
Rocks optional rolls	
area51	Security-related packages for analyzing the integrity of files and the kernel
bio	Bioinformatics utilities
fingerprint	Fingerprint application dependencies
htcondor	HTCondor high-throughput computing workload management system
ganglia	Cluster monitoring system
hpc	Tools for running parallel applications
kvm	Support for building Kernel-Based Virtual Machine (KVM) virtual machines on cluster nodes
perl	Perl RPM, Comprehensive Perl Archive Network (CPAN) support utilities, and various CPAN modules
python	Python 2.7 and Python 3.x
Web-server	Rocks web server roll
Zfs-linux	Zetabyte File System (ZFS) drivers for Linux

Table 2. Components of current XCBC build Part 2 – Components specific to XSEDE cluster “run-alike” compatibility

Category	Specific packages
Compilers, libraries, and programming tools	Charm, compat-gcc-34-g77. gcc, gcc-gfortran, fftw2, fftw, gmp, hdf5, java-1.7.0-openjdk, libRmath, libRmath-devel, mpfr, mpi4py-common, mpi4py-mpich2, mpi4py-openmpi, mpich2, openmpi, PSM API, numactl, librdmacm, libibverbs, papi, python, tcl, R, R-core, R-core-devel, R-devel, R-java, R-java-devel
Scientific Applications	BEDTools, GotoBLAS2, PLAPACK, PnetCDF, SHRiMP, Abyss, arpack, atlas, autodocksuite, boost, bowtie, bwa, darshan-runtime-mpich, darshan-runtime-openmpi, darshan-util, libgfortran, libgomp, elemental, espresso-ab, gatk, glpk, gnuplot, libXpm,

	gd, gnuplot-common, gromacs, gromacs-common, gromacs-libs, hmmer, lammps, lammps-common, libgtextutils, lua, meep, mpiblast, mrbayes, ncbi-blast, ncl, ncl-common, nco, netcdf, numpy, octave, petsc, picard-tools, plplot, libtool-ltdl, saga, libmspack, wxBase3, wxGTK3, Samtools, scalapack-common, shrimp, slepc, sparsehash-devel, sprng, sratoolkit, sundials, trinity, valgrind
Miscellaneous Tools	ant, scone, giflib, libesmtp, libicu, pulseaudio-libs, libasyncns, libsndfile, libvorbis, flac, libogg, libXtst, rhino, jpackage-utils, jline, tzdata-java, wxBase, wxGTK, wxGTK-devel, xorg-x11-fonts-Type1, xorg-x11-fonts-utils,
Scheduler and Resource Manager	maui, torque
XSEDE Tools	Globus Connect Server, Genesis II, GFFS

Software available as part of the XCBC build includes a number of scientific packages and supporting software added since earlier reports about XCBC [5]. There have been two major XSEDE Rocks Rolls released since the 2014 report. Version 0.0.8 saw a major OS release update from Centos 6.3 to 6.5 and 27 scientific and supporting packages have been added, including GenomeAnalysisTK, gromacs, mpiblast, and others [15]. The 0.0.9 release from November 2014 saw 41 additions, including TrinityRNASeq, R, significant Java updates, and other scientific and supporting packages [16]. Software included in XNIT, but not part of the basic XCBC build, continues to evolve in response to community requests.

3. BUILDING XCBC FROM SCRATCH OR ADDING PARTICULAR COMPONENTS TO AN EXISTING CLUSTER VIA XNIT

The Rocks team has worked for a number of years to help enable the creation of easily deployed and managed clusters [17]. Using CentOS as their base operating system, they have created a system for managing computational nodes from a central (frontend) node [8]. This creates a fairly simple way to deploy a basic cluster. Using an internal database, Rocks can manage many compute nodes. This allows an administrator to easily add, remove, and upgrade software across nodes and to maintain a uniform environment.

Using the XSEDE roll during the Rocks cluster install will add the packages necessary for an XSEDE-compatible basic cluster. Once up and running, to maintain the package levels, you can enable the XSEDE Yum repository, then follow the Rocks instructions or use the preferred method and create an update roll to add to your distribution [18]. The negative side of the Rocks upgrade options is that neither method will seem easy to a novice administrator. So while clusters are relatively easy to bring online and expand, upgrading and other more in-depth maintenance may be daunting to less experienced users, which may mean clusters aren't maintained, kept secure, or upgraded with the latest XSEDE-compatible cluster software. These problems aside, Rocks may be the best solution for getting an XSEDE-compatible cluster up for institutions that may have to depend on graduate students, faculty, or shared IT staff for installing and maintaining an XCBC.

Using XNIT to create an XSEDE-compatible cluster is a fairly easy task. An administrator would need to initially set up the repository configuration. There are two ways to do this. The first method is to download and install the XSEDE repo RPM from the XSEDE Yum repository [19]. The second is to install the yum-plugin-priorities package, then create the file /etc/yum.repos.d/xsede.repo with the lines specified in the XSEDE Yum repository README file [13].

As new packages are created, when "yum update" is called, it will find any new packages in the repositories your server is using and will try to resolve any dependencies for those packages. Then it will provide the administrator with a full list of packages to be updated. Yum still requires an administrator to periodically run update checks. Tools are available (or admins can write their own scripts and cron jobs) to either automate Yum updates or notify administrators of package updates.

Updating packages automatically may cause unexpected behavior in a production environment, especially for less-experienced system administrators. Creating a notification script so that packages may be reviewed and tested on non-production nodes or systems might be the more prudent action. There are several tools that do this such as Yum updates developed by Duke and available from CentOS and other distribution packagers.

4. IMPLEMENTATIONS OF XCBC AND XNIT FOR RESEARCH TO DATE

There are now a number of clusters in operation that use XCBC or XNIT as the primary or supplemental source of cluster management and application software. These include clusters at Howard University, Michigan State University, Marshall University, Montana State University, and the University of Hawaii. The first three clusters are built from the ground up with the XCBC Rocks installation media, while those at Montana State University and the University of Hawaii use the package repository.

Two clusters had been in operation and were torn down and rebuilt from scratch with XCBC. The Marshall University cluster, consisting of 264 cores in 22 nodes (2.8TF theoretical), including 8 GPU nodes with 3584 CUDA cores, leveraged the XCBC to replace a prior cluster management system. The cluster at Howard University, which is operated by a professor of chemistry, had also been in service under another management system and was taken down and rebuilt from scratch with XCBC, to the significant satisfaction of the professor responsible for it (Dr. Marcus Alfred). All of these implementations have been done with support from the XSEDE Campus Bridging team. In the case of Marshall University XSEDE campus bridging staff spent a week on site working with the Marshall University IT staff.

XCBC adopters have performed a critical function in hardening the installation and implementation of XCBC, and have provided guidance on which packages should be included in the system. Administrators of the Montana State installation, in particular, have been instrumental in investigating how to implement software from XCBC in environment modules, and integrate it with existing cluster management systems. Colleagues at the University of Hawaii have also been extremely helpful in helping us learn how to successfully integrate particular components of XCBC to supplement an existing commercial cluster management system.

Table 3. Deployed XCBC Clusters that had XSEDE Campus Bridging team involvement.

Site	Nodes	Cores	Rpeak (TFlops)	Other Info
University of Kansas	220	1760	26.0	Will be in production in

				summer 2015
Montana State University	36	576	11.98	300 TB of Luster storage [20]
Marshall University	22	264	6.0	8 GPU Nodes, 3584 CUDA Cores [21]
Pacific Basin Agricultural Research Center (Univ. of Hawaii – Hilo)	16	80	4.3	40TB storage, 60TB scratch
Indiana University	6	12	.54	LittleFe Teaching Cluster
Indiana University	4	16	.79	Limulus HPC 200 Cluster
Total	304	2708	49.61	

Clusters making use of XCBC or XNIT total almost 50 TFLOPS of processing capability. By the end of 2020, nearing the end of the second XSEDE funding, our goal is to have the aggregate processing capacity of the clusters making use of XCBC and XNIT exceed half a PetaFLOPS of processing capacity.

5. XCBC ON LITTLEFE AND XNIT ON LIMULUS HPC200 LUGGABLE CLUSTERS

The need for cluster administrators exceeds supply for a variety of reasons. There is never enough budget, and cluster administration often falls through the cracks in education and training efforts. A minority of computer science departments teaches classes in cluster administration. Also, cluster administration is not generally included in training efforts outside the credit-bearing curriculum. For example, the highly effective Software Carpentry project does not include cluster administration [22].

The LittleFe project began in 2005 under the leadership of Paul Gray (University of Northern Iowa), Dave Joiner (Kean University), Tom Murphy (Contra Costa College), and Charlie Peck (Earlham College). The goal of this project was to engineer a low-cost, easy-to-assemble cluster for the purpose of cluster administration and computer science education. LittleFe is a complete 6-node, Beowulf-style cluster that weighs less than 50 pounds and can be built from easily available components for less than $4,000. It is straightforwardly luggable, if not quite easily portable. The LittleFe project expanded in 2010 with a grant award from Intel to build 25 LittleFe devices and deploy them across the country for computational science education [23].

Another luggable option for teaching cluster administration and computational science is the Basement Supercomputing Limulus series Personal Cluster Workstation. The Limulus HPC200 Personal Cluster Workstation encases one headnode and three compute nodes in a single case [24] weighing 50 pounds. It is built using Scientific Linux, an RPM-based Red Hat Linux variant [25]. The Limulus system is more polished and self-contained with some integration work that gives adequate power and resource management for $5,500 to $8,000.

5.1 A modification to standard LittleFe design enabling use of XCBC and improving numerical performance of LittleFe

We have developed a modification of the standard LittleFe design to enable use of XCBC with LittleFe and improve numerical performance of the LittleFe design. The LittleFe design is similar to a blade chassis. Instead of using boards designed for a specific footprint, backplane, and task, it uses off-the-shelf parts to accomplish the same idea of a (relatively) large number of cores in a small, economical footprint. The goal is to provide an environment suited to teaching computational science with the most computing power in a portable package. The LittleFe design uses atom-based processors that provide x86 instruction set compatibility in a low-power configuration on mini-ITX, small-footprint boards. These system-on-a-board configurations allow for a very small overall design with modest power needs.

As is the nature of the computing world, the choice of components for building a LittleFe have evolved since the last instruction set (LittleFe v4 [26]). This is relevant to assembling a cluster for training purposes and provides the option of creating a LittleFe cluster for research purposes. One could change power supply options and use mini-ITX-based, Haswell-based Celeron CPUs for a more modern, yet cost-effective CPU. This increases power requirements somewhat for significant gains in single-core performance [27]. (These CPU choices also eliminate the option of using hyperthreading, which may be an issue depending on training goals.) Also, because the XCBC-from-scratch installation is based on Rocks, and because Rocks does not support diskless installation, the standard LittleFe components must be expanded to include a hard drive to be used as a training tool specifically with XCBC. This means adding some sort of hard disk drive (HDD) or solid-state drive (SSD) for each node. One could physically mount a 2.5-inch laptop-type drive for each node. An alternative would be to use an internal mini Serial-ATA (mSATA) drive that directly mounts to a compatible motherboard. The advantage is minimizing space in the LittleFe rack while minimizing components that need to be isolated electronically as well as physically secured. The disadvantage is that each added component increases the power needs.

We built an exemplar of a modified LittleFe using Haswell-compatible Gigabyte mini-ITX boards utilizing the LGA-1150 socket [28]. In addition, we added Crucial 128gb internal mSATA drives [29] on each node. We used a hard-wired connection using a dual-homed headnode. All nodes utilize the same motherboard, but only one of the two network interfaces will be used on compute nodes. The differences in power needs for the CPU and disk on each node meant that we had to diverge from the single power supply LittleFe calls for. Instead, we added an individual power supply for each node. This adds complexity to the assembly process but enables more flexibility for future upgrades.

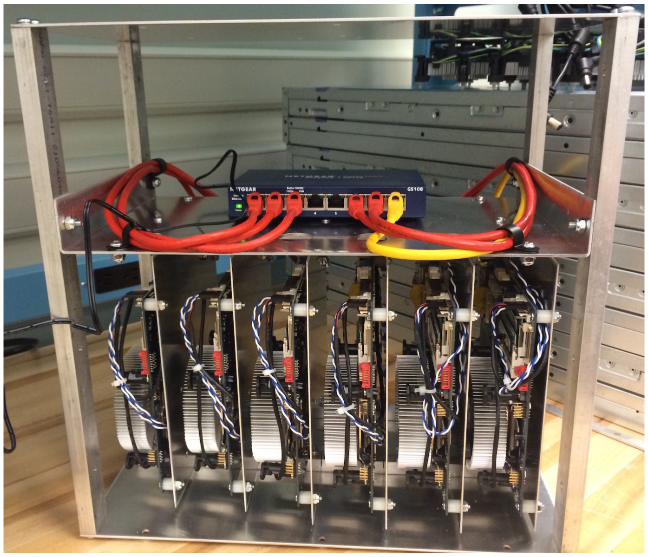

Figure 1. A LittleFe V4 frame showing six nodes exposed in a single, portable chassis, rear view.

The original LittleFe used a heat sink on the CPU and a small add-on fan to blow air over the heat sink fins. Since the power needs of the Haswell CPUs are higher, we had to add a CPU fan for cooling. The Atom (D510) used historically in the LittleFe build uses 10.56 watts versus 43.06 watts for the Celeron G1840 [27]. The fan that comes packaged with the Celeron G1840 processor we used is too large to fit in the space allocated per LittleFe node. You need to use a lower-profile fan assembly. We chose the Rosewill RCX-Z775-LP 80mm Sleeve Low Profile CPU Cooler as it fits well in the allotted space.

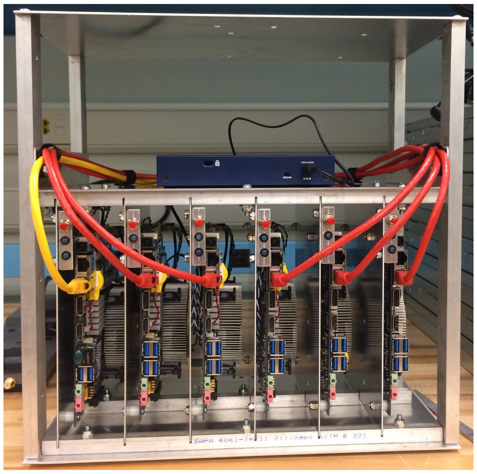

Figure 2. A LittleFe V4 frame showing six nodes exposed in single, portable chassis, front view

Figure 1 and Figure 2 show the LittleFe frame with the slightly modified design. The boards are still mini-ITX form factor, but using Gigabyte GA-Q87TN motherboards that use the LGA-1150 socket for more modern processors [28]. The fan housing is visible in these pictures.

As of the publication of this paper, XCBC will be included among the standard supported options for operating environments within the LittleFe project. Instructions for XCBC on LittleFe clusters and the parts list and building instructions are included in the LittleFe web site and class materials [30].

5.2 LIMULUS HPC200 AND XNIT

The Limulus HPC200 from Basement Supercomputing is a commercial product that puts a cluster in a deskside computer enclosure. It includes fewer compute nodes than the Rocks-based LittleFe but they are diskless in design, so a little less complex. The HPC200 has an 850W power supply, allowing for more powerful CPUs, consistent with its main purpose as a personal cluster The current build uses i7-4770S CPU Haswell (3.10GHz, 8MB cache, 65 watts). Since there are fewer nodes, maximizing the CPU power available for this power footprint is key. The HPC200 currently provides 16 cores of Haswell-generation die CPUs versus the 12 cores in the IU-built LittleFe. [1]. Further, there is power management that turns nodes on and off as needed for maximum power efficiency. This can also be scheduled [31]. The Limulus HPC200 with cover removed is shown in Figure 3.

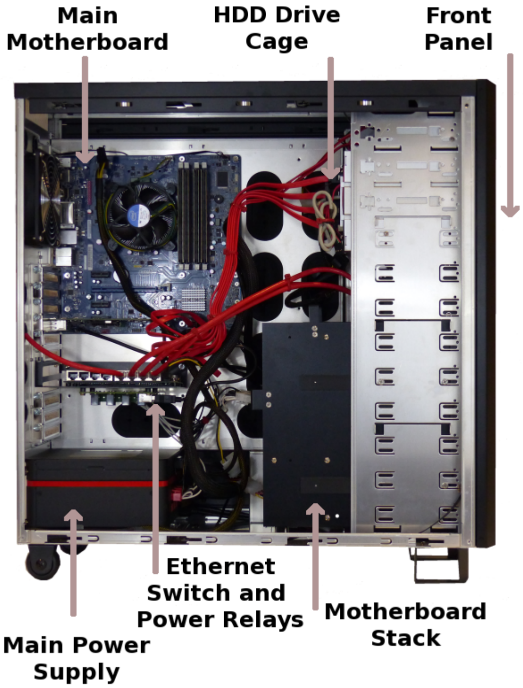

Figure 3. The internals of the Limulus HPC200 Deskside Cluster architecture.

The HPC200 is delivered with software cluster management utilities off the shelf, so one has only to add RPMs from the XSEDE Yum repository to get the desired XCBC capabilities.

6. XCBC, LITTLEFE, XNIT, and LIMULUS HPC200 FOR TRAINING AND EDUCATION

LittleFe was developed to support education in high performance and parallel computing. Since 2005 LittleFe has been featured in workshops and conferences, including Supercomputing (SCxx) and TeraGrid/XSEDE, and at a number of smaller conferences and internship programs [32]. In 10 years of instruction, LittleFe has shown that it meets the needs as an inexpensive and portable means for teaching HPC principles.

We are expanding this rich history by adapting XCBC to the LittleFe model. A curriculum module entitled "Building and administering a Beowulf-style cluster with Little Fe and the XSEDE-compatible Basic Cluster build" is available from the LittleFe web site [30]. Similarly, XSEDE and HPC University offer a variety of online training modules on parallel computing, scientific computing, and available software [33, 34]. The XSEDE campus bridging team will maintain a web page on the XSEDE site [4] that links to training modules that may be used as is with an XCBC build operating on a LittleFe or XNIT and Limulus HPC200 cluster. Together, these resources will enable educators to teach cluster administration and a good variety of important topics in parallel computing.

7. XCBC, LITTLEFE, and LIMULUS HPC200 FOR RESEARCH

The Limulus HPC200 cluster is billed as a turnkey solution for personal scientific computing, targeting workloads requiring fewer than 16 cores [31]. LittleFe was developed as a teaching tool. However, given the CPU modifications of LittleFe presented in this paper, it's worth considering either system as a potential research computing resource for an individual researcher.

Table 4 shows the basic processing components of the Limulus HPC200 and LittleFe cluster as constructed with the components described here. Table 5 shows the peak theoretical processing capability (Rpeak) and maximum achieved processing capability (Rmax) in GFLOPS of the LittleFe described in this paper and the Limulus HPC200. The Rmax shown for Limulus HPC200 is based on actual results of tests conducted by Basement Supercomputing [35], based on the HP Linpack benchmark [36].

Table 4. Basic characteristics of a Limulus HPC200 cluster and a LittleFe cluster

Cluster	Nodes	CPU clock rate	CPUs	Cores
LittleFe	6	2.8 GHz	6	12
Limulus HPC200	4	3.1 GHz	4	16

Table 5. Performance and price/performance for LittleFe and Limulus HPC200.

System	Rpeak	Rmax	Cost	Rpeak $/GFLOPS	Rmax $/GFLOPS
LittleFe	537.6	403.2*	$3600	$7/GFLOP	$9/GFLOPS
Limulus HPC200	793.6	498.3	$5995	$8/GFLOP	$12/GFLOPS

** Rmax for LittleFe is estimated due to a hardware failure prior to Linpack. Estimated at 75% of Rpeak. Testing will be complete prior to conference.*

While LittleFe was originally conceived as an educational tool, the components used in the LittleFe system constructed here provide a very reasonable solution for a deskside cluster. A half-TeraFLOPS deskside cluster for under $4,000 could be attractive to a number of researchers as could a roughly $6,000, three-quarter-TeraFLOPS deskside system with considerable local storage capabilities as a commercial product. Additionally, these prices are an order of magnitude lower than similarly powered systems in a typical server configuration [36] [37]. Use of the XNIT Yum repo helps provide a straightforward way to keep such a cluster updated, which is certain to be a concern for scientists using deskside systems for parallel computing.

8. CONCLUSION

The Rocks project, XCBC Rocks Roll and the XNIT Yum Repository are important tools that enable automating many cluster administration tasks. With the XCBC build and the associated implementations for installation from scratch via Rocks, or addition of specific sets of XCBC components, we have made it possible for cluster administrators to manage a cluster that keeps the software consistent with the current open-source software available via XSEDE-supported clusters. While there is no one gold standard for how best to set up a cluster, consistency with XSEDE offers many advantages to cluster users and administrators.

These advantages include using a tried-and-true cluster management system such as Rocks. Rocks works well for the experienced, novice, or intermediate administrator. Rocks is proven on systems from test clusters to intermediate clusters like Marshall University's, all the way up to national resources such as the San Diego Supercomputing Center Gordon supercomputer [38]. In addition, the common software packages and configurations on XSEDE resources packaged for local clusters should help reduce use barriers for researchers. A common repository for maintaining and upgrading these scientific software packages simplifies keeping the system up to date. Finally, the XSEDE Campus Bridging team is very active and strives to be responsive to XCBC administrators' and users' needs and requests.

Our work with XSEDE Campus Champions, ACI-REF, and other information technology professionals at small colleges and universities confirms that there are inadequate staff resources to administer and support local cyberinfrastructure resources. Our interactions with faculty at such schools confirm that for faculty, one of the most difficult types of time to get is time for curriculum development. Of the existing XCBC and XNIT installations, all but one are at universities that are either Minority Serving Institutions or Institutions in an EPSCoR state. Our initial beliefs about the utility of XCBC and XNIT to researchers and IT professionals at such institutions are borne out by the current pattern of adoption.

Relatively low-cost solutions can be utilized in secondary-education STEM classes as well as post-secondary institutions. While learning the basic principles of computer science may be the key focus in secondary education, preparing students to use HPC resources and understand the fundamentals can only help to create a solid foundation for future education and research. Using free software such as XCBC and XNIT can serve to train secondary students in programming and research principles and possibly in system administration principles as well.

A result of previous work is that faculty at small colleges with limited time for curriculum development can now create and administer a cluster using XCBC or XNIT, and use curriculum tutorials on XSEDE resources. They do not need to re-create new materials specialized for local resources. And for faculty who are their own cluster administrators, that task is easier and less time consuming. With tools such as LittleFe or a Limulus HPC200 cluster, classes can use these clusters without impacting research going on with departmental or university resources. Also, bare-metal installations can be done as part of the curriculum, meaning students experience installing clusters and software and monitoring. This helps educate the next generation of server administrators and admin-savvy researchers.

While there are many ways to achieve the goals of a teaching HPC system and a practical, deskside HPC system, these two methods seem to fit the criteria. Both systems are portable, with the LittleFe weighing under 50 pounds and the Limulus HPC200 weighing in at 50 pounds. They are similar in size, reasonably portable, and have enough power to demonstrate HPC capabilities. Both systems offer advantages. LittleFe is built from off-the-shelf components for a modest budget ($3,000 to $4,000) and can be easily upgraded.

The same result could be accomplished with "scrap" or recycled hardware, but this has several downsides. Generally you'd be using hardware a number of generations old, and often in the academic world, at the end of its lifecycle. Old workstations will have a considerably larger footprint, and will be noisier and often ill-configured to become a modest HPC cluster. Old servers, by contrast, may be better suited for the task. However, they are often very large form factor (multi "u" 19-inch rackmount or comparable), have air handling that may be loud and unsuited to an office or educational environment, and are definitely not portable or practical as deskside or teaching applications.

While options such as the Raspberry Pi are often used for teaching computer science principles, such solutions aren't as practical for teaching real-world parallel languages or HPC applications, or for small problem-solving or experimental clusters because they are not based on the x86 instruction set. LittleFe and the Limulus HPC200 cluster are. Combined with the XCBC and XNIT tools software, they may be used as instructional clusters using the same software set found on a typical XSEDE or campus cluster.

The concept of creating a small cluster from inexpensive hardware is hardly a new one. The Beowulf concept is more than 20 years old. The idea of creating a small-footprint cluster for teaching, outreach, and even prototyping projects, however, is somewhat new. LittleFe and the Limulus HPC200 cluster models embrace this compact design and show that there are many excellent uses and possibilities with small, portable clusters.

Commercial cloud services are often touted as a resource for HPC education, and in certain contexts may be an excellent choice. However, there is a fundamental difference in the cost and payment models of a small cluster vs. use of AWS or other commercial cloud providers. With a small cluster, one-time monies can be pooled to purchase a hardware resource that is as large as appropriate / possible, and maintained over time with very little cost. Cost is fixed at purchase time, which can be very practical for any group operating on a limited budget. The usage of such a cluster is capped by the capabilities of the system purchased. Use of commercial cloud is typically an ongoing service expense rather than a one-time capital expense. It can be surprisingly straightforward for an enterprising student to use more resources (and commit more university funds) than intended, since not all commercial services support proactive capping of usage. These are some of the practical reasons for choosing the approach of a small cluster such as LittleFe or Limulus HPC200 for institutions with limited budgets.

The Limulus HPC200 also gives a different perspective on teaching the mechanics of creating an XCBC. The XSEDE Yum repository allows a user to take an existing cluster and install software that makes it the equivalent of an XSEDE cluster [11]. Most prior discussions of creating an XCBC have focused on using the Rocks installation to start from bare metal. Using the Limulus HPC200, one can take the running cluster, and with XNIT add software, change the schedulers, and easily document the approach to make it reproducible.

In summary, the XCBC build is a useful addition to existing tools for cluster management and cluster systems software. XCBC has already proven useful in aiding research at a small set of institutions with high-quality researchers and overworked IT staff. The integration of XCBC with the LittleFe project will expand the utility of XCBC in training and educating sysadmins. Further adoption of XCBC will facilitate training and education of parallel computing users generally. Finally, we have demonstrated here that the LittleFe modified design we present offers performance comparable to the Limulus HPC200 at a lower price point. Both offer an option for a deskside-computing environment that may be easily maintained by the practicing scientist using XCBC and the XSEDE Yum repository for cluster management and maintenance. We expect wider adoption of XCBC in the future and believe that such adoption will help successfully address some of the challenges in US cyberinfrastructure identified by National Science Foundation task forces. Our focus on XCBC implementation and support has been the US, because this effort is funded to address specific challenges within the US research community. However, this project is entirely open source and the software, repositories, and training materials are readily accessible to anyone to download and use throughout the world.

9. ACKNOWLEDGMENTS

This document was developed with support from National Science Foundation (NSF) grant OCI-1053575. The LittleFe project has been funded in part by a grant from Intel, Inc. to Charlie Peck as well as NSF grants 1258604 and ACI-1347089. This research has also been supported in part by the Indiana University Pervasive Technology Institute, which was established with a major grant from the Lilly Endowment, Inc. The financial support of these organizations is greatly appreciated. The assistance of the XSEDE Campus Champions, the participants and leaders of ACI-REF, and other members of the US scientific community are gratefully acknowledged. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the organizations that have supported this work.

10. PHYSICAL ADDRESSES

IU Pervasive Technology Institute: Indiana University, 2709 E. Tenth Street, Bloomington, IN 47408-2671

Earlham College: 801 National Road West, Richmond, IN 47374

11. REFERENCES

1. Basement Supercomputing. *Limulus HPC 200 Personal Cluster Workstation*. [cited 2015 February 9]; Available from: http://basement-supercomputing.com/index.php/products/hikashop-menu-for-products-listing/product/2-limulus-hpc-200-personal-cluster-workstation.
2. NSF Advisory Committee for Cyberinfrastructure Task Force on Campus Bridging. *Final Report*. [PDF] March 2011; Available from: http://www.nsf.gov/cise/aci/taskforces/TaskForceReport_CampusBridging.pdf.
3. Hackworth, K., *Finding data (Personal communiction + Spreadsheet)*, J.L. Fischer, Editor. 2014.
4. XSEDE. *Campus Bridging Home Page*. March 2015]; Available from: https://www.xsede.org/campus-bridging.
5. Fischer, J., R. Knepper, M. Standish, C. Stewart, R. Alvord, D. Lifka, et al., *Methods For Creating XSEDE Compatible Clusters*, in *XSEDE '14*. 2014, xsede.org: Atlanta, GA.

6. XSEDE. *Resources Overview*. March 2015]; Available from: https://www.xsede.org/resources/overview.
7. XSEDE, *2013 XSEDE Staff Climate Study Report*, Lizanne DeStefano, L. Rivera., and M.J. Culbertson, Editors. 2013.
8. Rocks Clusters. *Home page*. 15 Jan 2015]; Available from: http://www.rocksclusters.org/.
9. XSEDE. *XSEDE Campus Champions*. 20 Feb 2012]; Available from: https://www.xsede.org/campus-champions.
10. ACI-REF. *Advanced CyberInfrastructure - Research and Education Facilitators - Home Page*. Available from: http://www.aciref.org/.
11. Fischer, J.L., Brown, Paul. *What is the XSEDE Yum Repository, and how do I use it?* 2013 [cited 2015 February 9]; Available from: https://kb.iu.edu/d/bdwx.
12. Fischer, J.L.a.B., Paul. *What software is installed on a "bare-bones" XSEDE-compatible Rocks cluster?* 2013 [cited 2015 January 14]; Available from: https://kb.iu.edu/d/bdww.
13. XSEDE. *readme.xsederepo*. March 2015]; Available from: http://cb-repo.iu.xsede.org/xsederepo/readme.xsederepo.
14. XSEDE. *User Portal - Software Listing*. March 2015]; Available from: https://portal.xsede.org/software/.
15. XSEDE. *README.0.0.8*. March 2015]; Available from: http://software.xsede.org/cb/centos6/x86_64/README.0.0.8.
16. XSEDE. *README.0.0.9*. March 2015]; Available from: http://software.xsede.org/cb/centos6/x86_64/README.0.0.9.
17. Rocks Clusters. *About*. March 2015]; Available from: http://www.rocksclusters.org/wordpress/?page_id=57.
18. Rocks Clusters. *The Rocks Group. Base User Guide Rocks 6.1. Chapter 8: Advanced Tasks – 8.10 System Updates*. March 2015]; Available from: http://central6.rocksclusters.org/roll-documentation/base/6.1/update.html.
19. XSEDE Campus Bridging. *XSEDE Yum Repository*. Available from: http://cb-repo.iu.xsede.org/xsederepo/.
20. Montana State University. *Hyalite Research Cluster*. Available from: http://www.montana.edu/researchcomputing/HyaliteCluster.html.
21. Marshall University. *New high-performance computing cluster gives Marshall unprecedented research, teaching and learning tools*. 2011; Available from: http://www.marshall.edu/murc/computing-cluster/.
22. Software Carpentry. *Home page*. Available from: http://software-carpentry.org/.
23. LittleFe. *History*. [cited 2015 February 9]; Available from: https://littlefe.net/history.
24. Basement Supercomputing. *Limulus HPC 100 Personal Cluster Workstation*. Available from: http://basement-supercomputing.com/index.php/products/hikashop-menu-for-products-listing/product/1-limulus-hpc-100-personal-cluster-workstation.
25. Scientific Linux. *Why make Scientific Linux?* [cited 2015 February 3]; Available from: https://www.scientificlinux.org/about/why-make-scientific-linux/.
26. LittleFe. *LittleFe 4d Parts List*. March 2015]; Available from: https://littlefe.net/parts-v4d.
27. CPU Boss. *Intel Celeron G1840 vs Atom D510* [cited 2015 February 9]; Available from: http://cpuboss.com/cpus/Intel-Celeron-G1840-vs-Intel-Atom-D510.
28. Gigabyte. ***GA-Q87TN***. Available from: http://www.gigabyte.com/products/product-page.aspx?pid=4755 - ov.
29. Crucial. *Crucial M550 128GB mSATA Internal SSD*. March 2015]; Available from: http://www.crucial.com/usa/en/ct128m550ssd3.
30. LittleFe. *Curriculum*. Available from: https://littlefe.net/curriculum.
31. Basement Supercomputing. *Limulus Personal Cluster Questions*. Available from: http://basement-supercomputing.com/index.php/faq/personal-cluster-questions.
32. LittleFe, *On the Road*. https://littlefe.net/on-the-road
33. HPC University. *Home Page*. March 2015]; Available from: http://www.hpcuniversity.org/.
34. XSEDE. *XSEDE User Portal Online Training*. March 2015]; Available from: https://portal.xsede.org/online-training.
35. Basement Supercomputing. *i7-4570S-Raw-64G-HPL*. March 2015]; Available from: http://limulus.basement-supercomputing.com/wiki/i7-4570S-Raw-64G-HPL.
36. TOP500.Org. *TOP500 Description*. March 2015]; Available from: http://www.top500.org/project/top500_description/.
37. Dell Computers. *Poweredge VRTX*. 2015; Available from: http://www.dell.com/us/business/p/poweredge-vrtx/fs.
38. San Diego Supercomputer Center. *SDSC User Support: Resources: Gordon*. March 2015]; Available from: http://www.sdsc.edu/us/resources/gordon/.