Implementation of Simple XSEDE-Like Clusters: Science Enabled and Lessons Learned

Eric Coulter
Indiana University
VPIT/RT/CBRI
2709 E 10th St.
Bloomington, IN
jecoulte@iu.edu

Jeremy Fischer
Indiana University VPIT/RT
2709 E 10th St.
Bloomington, IN
jeremy@iu.edu

Barbara Hallock
Indiana University
VPIT/RT/CBRI
2709 E 10th St.
Bloomington, IN
bahallock@iu.edu

Richard Knepper
Indiana University
VPIT/RT/CBRI
2709 E 10th St.
Bloomington, IN
rknepper@iu.edu

Craig Stewart
Indiana University VPIT/RT
2709 E 10th St.
Bloomington, IN
stewart@iu.edu

ABSTRACT

The Extreme Science and Engineering Discovery Environment (XSEDE) has created a suite of software that is collectively known as the XSEDE-Compatible Basic Cluster (XCBC). It is designed to enable smaller, resource-constrained research groups or universities to quickly and easily implement a computing environment similar to XSEDE computing resources. The XCBC system consists of the Rocks Cluster Manager, developed at the San Diego Supercomputer Center for use on Gordon and Comet, and an XSEDE-specific “Rocks Roll”, containing a selection of libraries, compilers, and scientific software curated by the Campus Bridging (CB) group in the XSEDE project, kept current with those implemented on XSEDE resources. The Campus Bridging team has helped several universities implement the XCBC, and finds the design to be extremely useful for resource-limited (in time, administrator knowledge, or funding) research groups or institutions. Here, we detail our recent experiences in implementing the XCBC design at university campuses across the country. These XCBC implementations were carried out with Campus Bridging staff traveling on-site to the partner institutions to directly assist with the cluster build. In implementing XCBC on campuses, we found that number of the needs described by campus communities as well as the broader cyberinfrastructure community are solved by technical means, although financial issues remain. The remaining issue to be addressed is technical interoperability between systems, and we describe efforts to improve here.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

XSEDE16 July 17-21, 2016, ,

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4755-6/16/07.

DOI: <http://dx.doi.org/10.1145/2949550.2949570>

CCS Concepts

•Social and professional topics → Systems development; Implementation management; Systems planning;

Keywords

XCBC, Cluster Management, Rocks, Campus Bridging, Cluster Build

1. INTRODUCTION

In 2011, a task force of the NSF Advisory Committee on Cyberinfrastructure published a report outlining a number of national challenges and needs related to the general area of campus bridging - connecting and interoperating campus cyberinfrastructure and national cyberinfrastructure [1]. The major themes and common needs identified in that report summed up to the finding that the aggregate US cyberinfrastructure (CI) capacity was inadequate to meet open (non-classified) research needs, and the total capacity that was available at that time was not nearly as efficiently used as it could or should be. Particular needs identified that contributed to this situation were overall inadequacy of funding for campus CI, inadequate ability to move files (between campuses and national facilities, and from campus to campus), obstacles to interoperability of campus CI facilities that ranged from policy obstacles to software not mature enough to enable such interoperability, and a need for nationally accessible file systems.

Of these challenges, policy issues and funding for campus CI are being addressed at least in part by the Department of Energy’s ESNet and the National Science Foundation. The Science DMZ concept [2], and the NSF CC* solicitation in particular, are designed to foster growth of campus cyberinfrastructure facilities and promote their interoperability. The Science DMZ concept focuses on creating network infrastructures without obstacles at the network layer to interoperability of CU facilities. The NSF CC* solicitation [3] provides funding for implementation of Science DMZs also provides some funding for campus cyberinfrastructure. The

funding provided is clearly not enough to meet US aggregate needs for campus CI - but the NSF is making a significant investment that is providing funds while also increasing interest in local investment in CI by universities and colleges throughout the US. For example, today, cloud-based storage facilities as well as the NSF-funded Wrangler system (implemented by the University of Texas at Austin Texas Advanced Computing Center with the Indiana University Pervasive Technology Institute as a partner) largely meet the needs for share file systems identified in 2011 [4].

The XSEDE campus bridging mission is, quite simply, to make science easier for researchers [5]. Campus Bridging largely focuses on two of the main campus bridging problems identified in 2011 - file movement and interoperability of campus CI and national CI facilities. File movement is a problem now largely solved within the physical capabilities of existing networks, at least, by Globus Transfer, the first campus bridging tool rolled out by XSEDE. Implemented and operated by the University of Chicago /Argonne National Labs Computation Institute, Globus Transfer is now tightly integrated with the overall XSEDE infrastructure as well as widely used nationally. For example, during PY5 of XSEDE (July 1 2015 - June 30 2016), 603 individuals on 38 campuses moved 1191 files to or from an XSEDE-supported resource. Total data moved during this year from a campus resource to an XSEDE-supported resource totaled 6,102 PB; data moved from an XSEDE-supported resource to a campus resource totaled 6,270 PB. Globus Transfer uses Periscope [6] to optimize file transfer across existing paths. And the increasing availability of 100 Gbps connections to Internet2, the ability of researchers to move files across wide areas is dramatically better today than it was in 2011 when the ACCI task force completed its report.

Of the challenges identified in 2011, the one that has proved most problematic is interoperability of campus CI facilities at the campus level [7]. There has been some prior work on economics and benefits to sharing of campus resources [8]. However, in some cases, simple policies such as "no one but a member of our university community will use our university's CI resources" makes sharing of resources impossible. The other challenge to interoperability is the software that runs clusters.

The focus of this particular paper is enabling technical interoperability among campus computing clusters and interoperability between campus computing clusters and clusters operated and supported by XSEDE. This was one of the earliest needs identified by XSEDE as a whole via its use case process [7, 9]. There are several aspects to "interoperability": adding software tools that allow technical interoperability between campus CI facilities and XSEDE; interoperability of user skills and software commands between campus PI facilities and national facilities; and interoperability of utility of user training and education materials, so that training materials created by XSEDE can be used - usually with modification - for local campus CI resources. The technical interoperability of campus CI and XSEDE in terms of authentication and support for distributed workflows remains a work in progress.

The campus bridging effort within XSEDE, now part of the XSEDE Community Infrastructure group, has made significant progress in user-perceived interoperability through the creation of the XSEDE-compatible toolkit. This is a suite of software that allows one to create a cluster based

completely on open source tools (including a scheduler) where the open source tools are consistent with the open source tools that underlie the least esoteric of the XSEDE-supported clusters. XCBC is distributed as a Rocks Roll [10]. When building a cluster from scratch, XCBC is an excellent tool that is straightforward to implement. The XSEDE campus bridging team also provides an XSEDE National Integration Toolkit (XNIT) that allows administrators of an existing cluster to download and install tools from a YUM repository to supplement the functioning clusters and add to them software version / command line interoperability with XSEDE clusters, as well as the option of adding additional optional tools from a growing set of collections that includes statistics and bioinformatics software toolkits.

"Straightforward to implement" does not always mean easy to do, especially for cluster management staff at smaller universities and colleges who are often overworked and underfunded. XCBC installations in particular can be complex. In order to campus-based cluster management staff, the XSEDE Campus Bridging team regularly visits campuses to help local administrators or researchers set up and manage their clusters. Demand for these site visits has been steady, at about one visit request per quarter. In following sections we detail the software toolkits provided by XSEDE Campus Bridging. Then, we provide an overview of the sites where visits have been conducted, and describe the site visit protocol. Next, we discuss individual site visit results including usage of the systems after site visits and common issues with implementing Campus Bridging toolkits. Finally, we discuss some of the reflections that site visit leaders provided after implementation and the state of the XSEDE-compatible toolkit as part of XSEDE's overall strategy to be a connector of national cyberinfrastructure services.

2. XCBC AND XNIT TOOLKITS

A standardized and widely used software build can help campuses get improved use out of their existing clusters, quickly and effectively implement new systems, and prepare campus users for environments that are like XSEDE. It's certainly not desirable to completely standardize the software environments of compute clusters for scientific research, using some of the largest computational resources as models for software toolkits is a reasonable approach to aiding campus IT professionals. For practicing researchers in any field, having local resources that model the environments of larger resources only improves the ease with which calculations can be pivoted onto larger resources as the size or complexity of their work grows. To that end, the XCBC was developed to provide local campuses with a simple, well-documented method of implementing their own XSEDE-like computing environment. The XSEDE-compatible toolkit was first deployed as a Rocks Roll called the XSEDE-Compatible Basic Cluster (XCBC). A complete cluster management system based on the Rocks Cluster software produced at SDSC [11]. Rocks packages several useful cluster administration tools in the basic build, and enables the simple installation of other software with a system of "Rolls", which is a Rocks-specific method of packaging software to be installed on a cluster [11]. The XSEDE CB team has produced and curated an "XSEDE roll" that contains open-source tools modeled after the large clusters supported by XSEDE. Building an XCBC consists basically of installing a Rocks cluster with the XSEDE roll included.

Figure 1: Basic schematic of a Rocks cluster, consisting only of a head node with a public interface and internal network for compute nodes. The “optional specialized network” represents something like an IMPI (Intelligent Platform Management Interface) network to remotely power nodes on and off, or the network for a parallel file system. Image sourced from the Rocks User Guide [12].

The hardware standard for the XCBC is straightforward (see Fig. 1), consisting of a single headnode with two network interfaces (public and private), a simple switch for the private network, and compute nodes with a single network interface connected via the switch to the head node. Additional devices such as NAS (Network Attached Storage) or multiple login nodes are optional, but still within the scope of what can be built during a CB site visit. Software-wise, the XCBC is a cluster running Rocks, either the SGE (Son of Grid Engine) or Torque/Maui resource managers, and with the XSEDE roll installed. The Rocks software enables quick installation of the front-end machine and automated building of compute images with scientific software included (so that compute nodes are installed with minimal intervention other than powering them on). The main effort in an XCBC build consists of making the hardware ready to work as a cluster, and configuring the resource manager of choice to work with that hardware.

The one shortcoming of XCBC as a Rocks Roll is that to use it you need to be starting your cluster configuration from scratch - either building a new cluster or willing to completely give up on an existing cluster build and start over. In response to suggestions from the national user community, particularly XSEDE campus champions and participants in the ACRI-REF project [13], we developed the XSEDE National Integration Toolkit (XNIT) as a means to add open source software and create compatibility with XSEDE-supported clusters by adding new software to an existing and functioning campus cluster. The XNIT provides many open-source scientific packages used on XSEDE resources, managed in a similar fashion as they are on XSEDE resources, which simplifies the lives of cluster admins by removing the need for learning a new set of administration tools[14].

Current software suites maintained by XSEDE staff in-

clude statistical and bioinformatics software. Software may be downloaded on a package by package basis, by a cluster administrator, or one can subscribe to one or more packages in the YUM repository. Doing this removes much of the pain involved in maintaining a cluster used for a variety of scientific computing endeavors. The definitive web site that serves as a reference point to all other online information on the XSEDE XCBC and XNIT projects is the XSEDE Campus Bridging home page [5].

3. SITE VISITS

3.1 Partner Institutions

We will discuss the results of Campus Bridging site visits at four different institutions, in chronological order: Marshall University, in Huntington, WV; Southern Illinois University, in Carbondale, IL; Bentley University in Waltham, MA; and the University of Texas at El Paso (UTEP). Of them, one is in an EPSCoR state, and one is an MSI.

CB staff traveled from Indiana University to each of these sites for one week to help build an XCBC. Initial contact from each site to the CB team was generally made via the XSEDE Campus Champions mailing list, or as a result of a message on that list. Given the difficulty in marketing generally observed by projects serving low-resource institutions, the XSEDE Campus Champions program has been a great help as a grassroots marketing effort. XSEDE Campus Bridging also utilizes contacts via the Underrepresented Community Engagement (URCE) efforts in XSEDE, as well as contacts in other cyberinfrastructure organizations such as NSF’s ACI-REF team.

3.2 Site Visit Protocol

A site visit typically consists of a week-long visit to the site by Campus Bridging staff that is preceded by several phone meetings or email exchanges to work out details and prepare for the cluster build. The initial communications are set to establish the scope and extent of the set up tasks, identify any potential roadblocks, and allow all participants to familiarize themselves with the project. So far, one week has proven to be a suitable amount of time for the initial cluster set-up, benchmarking and testing of a set of applications; however, it is not enough time to get local users involved. The week is usually spent working full-time with local IT Staff and researchers to build the cluster. The goal is to make sure that each step is well understood by the campus team before moving on to the next step. After the visit, CB staff are available by phone and email for support and consultations as needed. Ongoing collaboration with site hosts is strongly encouraged.

In terms of their contribution to the visit, hosts at each site were required only to set aside sufficient time for the implementation and to provide a few basics to the Campus Bridging team: parking, local network access, and recommendations for local hotels and restaurants. Otherwise, our assistance was entirely cost-free to the host site. Extensive communication took place before each visit to help CB staff determine the needs of each site, and what materials would best help them maintain the XCBC after the initial visit. While help is always available remotely from the CB group, great effort was made to ensure that the on-site visit would provide a maximum of useful help and instruction. Because each site has varying levels of familiarity with HPC at each

site, this was a non-trivial part of the planning. During this initial planning phase, the main interest lies in determining several things: existing hardware configurations, software needed on the cluster, what the expectations are for after the visit, what sort of science they primarily intend to do, and special queuing needs or scheduler preference. The planning phase of site visits is a continual work-in-progress, with additions made to the checklist after (and during) each one.

A major component of site readiness is hardware availability. In two out of the three cases, the CB team was visiting in order to rebuild an existing cluster. At Marshall University, we converted a cluster formerly using proprietary StackIQ software into using open-source Rocks software. At SIU, we upgraded an existing Rocks cluster with an improved storage node, and integrated GPU nodes that were previously unusable. At Bentley, we turned existing hardware into a cluster that was previously used as virtual server infrastructure, and provided instruction on the use of schedulers in an HPC environment.

The major hardware requirements are that each node be able to PXE (Pre-boot eXecution Environment) boot, have at least one network interface for compute nodes and at least two for the head node, and there must be a DNS entry for the head node in the local network. The major difficulty in a cluster build often comes in some variety of hardware issue. The first day generally involves exploring the hardware and assessing the existing hardware state and starting preparation work for the cluster installation. Most often, the problems involve correct booting of either the head-node or the compute nodes. The most common source of hardware-related work comes from remote storage. Two out of three sites had separate storage machines, which must be first mounted to the head node, and *then* NFS-exported to the compute nodes over the internal network. Specific problems and solutions (or mitigation strategies) will be discussed in the “Lessons Learned” section.

Once the hardware is working as needed, it can be trivial to actually build the XCBC. At Bentley University, we were able to rebuild the basic cluster from scratch (install front-end and compute nodes - not including time to configure the scheduler or any software) in less than an hour and a half, thanks to a remote management system connected to each node.

Configuration of the software environment easily consumes the remaining site visit time. Depending on the familiarity of the host with HPC, this time can also be used for user or admin education. Development of educational materials supporting these activities is ongoing. Current materials include provide online training in the form of a repeating Rocks install webinar, articles in the XSEDE knowledge base, and a video detailing the installation of an XCBC using virtual machines on a laptop. The second critical area is configuration of the scheduler - in most cases, it has been sufficient to set up a “default” queue covering all nodes. For smaller institutions, condo models (in which researchers buy nodes to be added to an existing cluster, similar to buying a condominium) seem to be more rare, and simply getting users to *use* a scheduler is the largest struggle. The technical work scheduler-wise comes with either integration of GPU nodes, or getting proprietary software such as Matlab or Mathematica to work with an open-source scheduler. These will be covered in more detail in Section 4.2.

Figure 2: Ganglia screenshot showing heavy use of Big Green (Marshall University) during the last academic year. Ganglia portals from other sites are not publicly available.

3.3 Marshall University

Prior to the XSEDE CB site visit at Marshall University, there was already a functioning Rocks cluster (“Big Green”) that was running a paid, licensed version of Rocks. The license had expired, which limited the number of working compute nodes to 16 and broke GPU functionality. The goal of the site visit was to restore the cluster to working order with all nodes available, and restore the ability to run calculations using the GPUs. In May of 2015, Campus Bridging staff visited the campus in Huntington, WV, and were able to get Big Green up and running within four days.

The last day of the visit was spent testing installed software, configuring the scheduler, and testing the re-enabled GPU nodes. Currently, Big Green has a wide variety of software for materials research, visualization, and bioinformatics, including COMSOL, Multiphysics, Mathematica and CLC Genomics WorkBench[15]. Big Green has seen heavy use in the last year since XSEDE CB staff visited the campus - see Fig. 2.

3.4 Southern Illinois University

In September of 2015, XSEDE Campus Bridging staff assisted in a rebuild of the BigDog Cisco cluster at Southern Illinois University in Carbondale, IL. BigDog was previously running an older version of the XCBC package, but did not have sufficient hardware for the needs of SIU research computing. The new upgrade included 70TB of storage on the front-end machine, and two GPU nodes with dual NVIDIA Tesla K40 cards. The GPU nodes were finally fully supported with the new build. The final build has 40 compute nodes with 64GB of RAM and 20 cores each, giving the entire cluster a theoretical Rmax of 34.7 TeraFLOPS.

BigDog is used by researchers in materials science, computer science, and biology, as well as other disciplines at SIU. The integration of the GPU nodes allowed professors at SIU to begin teaching courses focused on massively parallel GPU programming. The addition of more local storage was also vital for researchers in biology - recently, the cluster has been used in one student’s thesis research, and enabled undergraduate students to learn and give a workshop on genome assembly and analysis [16]. Students also participated in developing a GUI (Graphical User Interface) for

Site	Nodes	Cores	RPeak	Other
Marshall University	24	264	6 TFLOPs	8 GPU Nodes, 3584 CUDA Cores
Southern Illinois University	42	800	29.4 TFLOPs	2 GPU Nodes, 5760 CUDA Cores
Bentley University	16	128	2.8 TFLOPs	

Table 1: Table showing final specifications of the XCBC at each site, after the CB visit.

some of the bioinformatics tools used on BigDog [16]. The Campus Bridging team is continuing to collaborate with the group at SIU on the maintenance and continual improvement of the BigDog cluster.

3.5 Bentley University

Campus Bridging staff helped build the first HPC resource at Bentley over the course of a week during January 2016, resulting in a 16 node cluster (currently named Rocks-at-Bentley) with 2.8 TFLOPS and 544 GB of RAM. Six more nodes will be added in the future, now that staff there has become familiar with cluster building and administration.

The XCBC build gave researchers and students at Bentley access to a true HPC resource for the first time, and significantly improved calculation times for several researchers. The new cluster has also enabled Bentley staff and researchers to embark on an expanded data science program, with additional plans to create a truly open HPC resource for the local community. Campus Bridging staff were able to integrate *some* proprietary software with the Bentley cluster, and are continuing to collaborate with Bentley staff to make sure the process will be easier for future XCBC implementations.

3.6 UTEP

During mid-May 2016, Campus Bridging staff visited the UTEP campus, and helped begin building the largest XCBC to date. This was the first HPC experience for the local administrators, so a large focus of the trip was on education for future maintenance. As in other cases, and also due to the large number of machines at this site, this education was carried out in tandem with the actual cluster build and configuration. The true volume of available hardware was unknown both to CB staff and local administrators, so plans for the trip had to be adjusted on the fly to incorporate the necessary build time.

At the end of the visit, the new cluster at UTEP (Orion) had 70 working nodes, including 9 GPU nodes, with an additional ~150 nodes to be installed in the future. The final RPeak estimate is around 49 TFLOPS, including the GPU nodes. The CB team is continuing to collaborate with the administrators at UTEP during the effort to integrate all of the hardware into Orion, including several Infiniband switches, which have not yet been integrated into the cluster (due to time constraints).

4. LESSONS LEARNED

Thanks to our work at these four sites, the Campus Bridging team has greatly improved several aspects of the site-visit. A pre-site checklist has been implemented, to ease the initial days of the visit, and prepare the team for potential hardware/software issues. This has also allowed us to prepare any educational material for local sysadmins or users beforehand. We have also developed a recipe for integration of NVIDIA GPUs into the XCBC, expanded our

experience of hardware issues, and improved our scripts for scheduler configuration. These were well tested during the UTEP build, and proved to simplify the addition of GPU hardware quite well. We look forward to encountering a wider variety of network fabrics, but so far all sites have simply been using standard Ethernet or 10Ge, which present no additional challenges to integrate into Rocks.

4.1 Hardware Issues

As far as the pre-site visit procedure, CB staff have learned to carefully check a few aspects of the hardware. The initial interview with campus staff focuses on basic hardware specifications, which covers the minimum requirements for Rocks. We also ask each site to have a DNS entry for the front-end machine prior to the visit, to avoid time spent waiting on a networking group, though this is usually a minor issue. Following that, the main issue is the ability of all nodes to PXE boot. While it's possible to build a Rocks cluster by installing each compute node with a CD, that would eat up a considerable amount of the limited site visit time!

While it is possible to determine the hardware involved in a cluster build beforehand, it is impossible to know exactly all the parts involved, or how they can fail. In some sense, preparing for hardware issues is both a fool's errand, and an absolute necessity. Hardware issues do not generally become apparent until one actually starts trying to build a cluster, so site visits must always come with the caveat that time will be lost due to unforeseen problems. It is even possible that time-constrained administrators will be unaware of the sheer volume of hardware available. In the surprising event that everything goes perfectly, extra time can always be devoted to a running a rough benchmark, user/admin education, or software configuration. The best response in the face of unknown hardware situations is to have a set of general tools available, beyond being willing to spend significant time searching the Internet for answers.

After reflecting on the situations encountered in the last year of site visits, there are a few simple tools that have the potential to mitigate hardware problems frequently encountered in a cluster build. First of all, once the hardware involved is known, having some familiarity with the available vendor support is a must. While many sites use old or donated hardware, most hardware vendors keep firmware updates and drivers available for even very old hardware - knowing where to find these can save hours of searching and troubleshooting. It is also occasionally necessary to be able to boot a machine into some sort of recovery mode to fix or troubleshoot hardware more effectively, so a variety of boot media is also a must-have. It is also useful to have a way to boot into DOS, in order to perform firmware modifications (some network cards, for example, require booting into DOS to change firmware settings, which can prevent/enable PXE booting.) It is also important to be aware that all machines may not boot from USB, or may not have an optical drive. Both optical and USB boot media should be on hand, and an external CD/DVD drive is always useful (while the site host

probably has many of these tools available, it is infinitely better to be prepared and familiar with one's own tools!).

A passing familiarity with EFI (Extensible Firmware Interface) and GRUB (GRand Unified Bootloader) shells can be quite helpful for diagnosing problems in PXE booting or even in installing the head node. Familiarity with disk partitioning tools and filesystem types is also necessary, since the default Rocks partitioning setup is generally too restrictive for most sites. In rare cases, large disks (> 16TB) can cause problems with installation of the head node, if large RAID (Redundant Array of Independent Disks) volumes are not supported by the machine BIOS (Basic Input/Output System).

It is also of utmost importance to record any problems encountered and solved (or remaining unsolved) in the course of a site visit, no matter how much trust one has in one's memory. The CB team is working on a way of sharing these more detailed technical "lessons learned" with past, future, and prospective XCBC hosts. While detailed notes are taken by CB staff and local admins, it would be beneficial to implement an XCBC wiki that would allow all site visit participants to record and share information. This is valuable for internal use, for future site visits, and for helping site hosts continue to maintain the XCBC at their institution.

4.2 Software issues

One of the most important advances the CB team has made with respect to software has been the integration of NVIDIA GPUs with the XCBC scheduler. The Rocks system out of the box does not offer GPU drivers, which is quite understandable given the varied ecosystem of graphics cards. Currently, the CB team has been able to develop scripts for integration of NVIDIA GPUs into a Rocks cluster, though the process is likely similar for other brands. These scripts are not yet publicly available, but will be released once they have been further tested.

The broad steps to correctly implement GPUs into a Rocks cluster are:

1. Create a separate node boot profile for GPU nodes.
2. Ensure that the open-source "nouveau" drivers are disabled in the GPU node boot parameters.
3. Download the NVIDIA CUDA developer toolkit[17].
4. Edit the GPU node profile so that Rocks will include the necessary drivers when the compute image is built.
5. Set the GPU nodes in the scheduler to have a consumable resource with the number of GPUs per node. (This can vary depending on the scheduler used!)

In addition to supporting GPU integration, we will also attempt to help a site host install and integrate any site-specific software needed by their researchers, though Campus Bridging cannot provide support for licensed software. This can prove problematic, as some proprietary software does not integrate well with open source schedulers. Matlab, for example, has a "Rocks roll" produced at SDSC, but proved difficult to install at Bentley University (the developers at SDSC proved to be very responsive, but we were not able to resolve the issues in the time we had left during the site visit). The only open-source scheduler that Mathematica officially supports integration with is SGE. It is very

important to note that SGE requires a library for java support (libdrmaa) that is not included in the default Rocks copy of SGE. Acquiring this library requires either finding an instance of SGE with DRMAA (Distributed Resource Management Application API)[18] support and copying the library file over, or re-building SGE with java support, which is not a trivial task. However, once SGE is set up properly, it is very easy to send parallel jobs to the scheduler from within the Mathematica interface, which lessens the need for user education in scheduler use. Documentation for the initial configuration, however, was found to be rather scant.

In large part, configuration of site-specific software ends up being left to local administrators after the site visit, due to the vast array of scientific software. There is really no way for a team to be prepared to swiftly and painlessly install any and all possible software, but having experienced people willing to *try* can make all the difference in a cluster build. Ongoing collaboration with site hosts is also invaluable in this regard, as the necessary steps for configuration of particular software packages can be saved and propagated to other XCBC hosts. The CB team is currently planning a system by which this collaboration can be eased, which will be open to the public.

5. BENEFITS OF CAMPUS BRIDGING TOOLS

The XCBC and XNIT software confer a number of benefits to campuses which make use of them, not only when coupled with a site visit that provides help in the implementation phase, but also throughout the life of the resource as campuses can benefit from the broader XSEDE ecosystem.

5.1 Benefits to campuses in general

One of the general conclusions we have derived from the site visit program is that these opportunities represent a catalyst for campuses to be able to address research computing needs in a definitive way. At all of the campuses visited, site teams were highly competent individuals, who were able to pursue acquisition and usage of the resources, as well as management of the systems on an ongoing basis. Like many staff at similar institutions, commitments to teaching, research, and administration meant that engaging in a task such as a cluster rebuild was difficult, especially if multiple players were involved to manage hardware, networking, and software. It appears to us that the site visit opportunity afforded these sites a "nudge in the right direction" when it came to providing the necessary expertise.

The prospect of having an engineer visit with a defined scope of action and implementation plan also helped enable a concerted effort to be made to install the clusters. Arranging for a site visit also provided campuses with a direct opportunity to ask questions and get answers (and build a relationship for future questions) in a timely and focused fashion.

When surveyed, some of the site participants had very strong responses about the site visit activities. One site visit partner remarked, "We'd have been lost without [Identifying Reference Deleted]. There were too many issues that we encountered for non-HPC people like us to have made it through on our own. We'll be there soon though, thanks to him." Another partner stated, "we really do greatly appreciate having been part of the CB evolutionary process," noting that the site visit was particularly useful in providing perspective on how their local cluster differed from others, par-

ticularly in software distribution and queue management for GPU use. One site host commented, regarding the XCBC toolkit, that, "The toolkit is very easy to use and it simplified the initial setup of our cluster. It was perfect for me since this was the first time I built a cluster." Estimates of time saved by the CB site visit range from 50 to 120 hours, and at one site, it was estimated that CB staff were able to save \$2000 in hardware costs by improving the configuration of certain hardware.

In addition to the campuses with whom we have worked closely, there are many campuses that have subscribed to the XNIT repository with no assistance from or contact with the XSEDE campus bridging team other than reading our materials online or our published papers. We have worked directly with a total of six campuses over the five years of the XSEDE project. Adding to this the campuses that have subscribed to XNIT without direct help from XSEDE staff, the total number of campuses within the US that are using some or all of the XSEDE-compatible toolkit is 74.

5.2 Benefits via the XSEDE ecosystem

Users of the XCBC and XNIT software are not only helped by the ease of installation and support from the Campus Bridging team, they are able to leverage the XSEDE ecosystem in general. As the packages are built based on a common set of software deployed on many resources in XSEDE, campus users can make use of XSEDE documentation, training, and optimization resources. This approach allows campus IT staff to direct inquiries about the software to the XSEDE knowledge base, for the local cluster user base to benefit from XSEDE online training and the virtual workshop library, and provides a line of support via the XSEDE Help Desk (help@xsede.org).

In order to facilitate experimentation with the XSEDE-compatible toolkit, this software environment will be available as a published and publicly-available VM on the Jetstream system [19].

6. CONCLUSION

Over the course of the last year, we have gained significant insight into the process of helping institutions build XSEDE-like clusters. Garnering attention for XNIT and XCBC is an iterative process, in which we publicize what we perceive the value of these tools to be, work with system administrators and researchers to use these tools, learn from these deployments and improve the tools, and then publicize the value of these tools more. In this regard the work documented here is a critical point in obtaining and disseminating information about the value of these tools.

A large part of the success of the XCBC toolkits has been due to the ability to disseminate news and identify campus staff interested in the toolkits and site visits - much of this happening via XSEDE campus champions and participants in the ACI-REF project. The community infrastructure and outreach initiatives of the XSEDE project have helped campus bridging gather requests for new software, prioritize the delivery of software components, and have provided direction and input on further useful toolkit requirements.

Two of the most frequently noted needs are the need for configuration management tools and the need for identity management tools. The XSEDE Community Infrastructure group, which now encompasses the XSEDE campus bridging effort, is planning to create and deploy ansible playbooks

as a means of providing a simple configuration management capability for campus clusters using the XSEDE compatible cluster suite and the additional optional software packages available from the XNIT YUM Repository. In addition, XSEDE plans to create authentication modules that will allow campus CI providers to offer authentication for clusters based on the current authentication services used by XSEDE.

This paper focused on four sites which had close engagement with XSEDE staff. We will continue to benefit from learning experiences with our colleagues at Marshall, SIU, Bentley, and UTEP. There are many other campuses which are using both XCBC and XNIT, as evidenced by repeated downloads of XNIT packages from the Campus Bridging repository. At the time of this writing, there are a total of nine campuses that have installed XCBC, and 68 installations of XNIT. These installations are now found in 26 states (Illinois and New York have five installations each, California has four installations), and 10 of these XNIT installations are in EPSCoR states.

Current social science understanding of technology adoption suggests that adoption decisions are driven by performance expectancy (perceived value); effort expectancy (perceived ease of use); social influence; and facilitating conditions (including knowledge of a technology and the belief that end users will find it accessible) [20]. With now more than 70 campuses using some form of the XSEDE-compatible toolkit, we believe that the community has spoken and declared these tools to be of value. We believe that the lessons learned in engagement with the four campuses detailed here, along with continued input from the community, will allow the XSEDE-compatible toolkit to be of increasing value to the national research community. We believe that adoption will now accelerate because of the demonstrated value of these tools and the social influence of satisfied adopters of these tools on their peers in the US research community. By expanding the adoption of the XSEDE-compatible toolkit, XSEDE will aid research at highly resource-constrained institutions as well as those not so highly constrained. We believe that we are now making significant progress in addressing one of the major, largely yet unsolved, problems in campus bridging identified in the 2011 NSF ACCI task force report - technical interoperability of campus CI facilities. In so doing, we are aiding XSEDE in achieving its goal to aid US research productivity and innovation by being a connector of US cyberinfrastructure services.

7. ACKNOWLEDGMENTS

The authors would like to thank their site hosts Jack Smith (Marshall), Rustomji Vania (SIU), Chet Langin (SIU), Jason Wells (Bentley), and Patrick Solis for their excellent hospitality during the site visits, and for their continuing collaboration with XSEDE. We thank leaders of ACI-REF for their important contributions to this effort, particularly Jim Bottum. We thank Ms. Therese Miller, who has led project management and budgeting for involvement of the Indiana University Pervasive Technology Institute in XSEDE. We thank several anonymous reviewers and our colleague Dr. Winona Snapp-Childs for their contributions to the quality of this report.

This document was developed with support from National Science Foundation (NSF) grant OCI-1053575. This effort

has also been supported by the Indiana University Pervasive Technology Institute. 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 NSF or of Indiana University.

8. REFERENCES

- [1] NSF Advisory Committee for Cyberinfrastructure Task Force on Campus Bridging. *Final Report*. NSF, March 2011.
- [2] Eli Dart and Joe Metzger. The science dmz. *CERN LHCOPN/LHCONE Workshop*, 2011.
- [3] National Science Foundation. Campus cyberinfrastructure (cc*) program solicitation nsf 16-567, 2016.
- [4] Christopher Jordan, David Walling, Weijia Xu, Stephen A Mock, Niall Gaffney, and Dan Stanzione. Wrangler’s user environment: A software framework for management of data-intensive computing system. In *Big Data (Big Data), 2015 IEEE International Conference on*, pages 2479–2486. IEEE, 2015.
- [5] Xsede campus bridging home page. <http://www.xsede.org/campus-bridging>, 2016. [Online; accessed 27-April-2016; XSEDE].
- [6] Dan Gunter, Raj Kettimuthu, Ezra Kissel, Martin Swamy, Jun Yi, and Jason Zurawski. Exploiting network parallelism for improving data transfer performance. In *Proceedings of the 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, SCC ’12*, pages 1600–1606, Washington, DC, USA, 2012. IEEE Computer Society.
- [7] Craig A. Stewart, Richard Knepper, James Ferguson, Felix Bachmann, Ian Foster, Andrew Grimshaw, Victor Hazlewood, and David Lifka. What is campus bridging and what is xsede doing about it? In *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the Campus and Beyond*, XSEDE ’12, pages 47:1–47:8, New York, NY, USA, 2012. ACM.
- [8] Md Anindya Prodhon and Andrew Grimshaw. Market-based on demand scheduling (mbods) in co-operative grid environment. In *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, page 26. ACM, 2015.
- [9] Craig A. Stewart, Richard Knepper, Andrew Grimshaw, Ian Foster, Felix Bachmann, David Lifka, Morris Riedel, and Steven Tueke. Xsede campus bridging use cases. Technical report, XSEDE, Tech. Rep, 2012.
- [10] Philip M Papadopoulos, Mason J Katz, and Greg Bruno. Npaci rocks: Tools and techniques for easily deploying manageable linux clusters. *Concurrency and Computation: Practice and Experience*, 15(7-8):707–725, 2003.
- [11] Rocks cluster software. <http://www.rocksclusters.org/>, 2016. [Online; accessed 26-April-2016; University of California San Diego].
- [12] Rocks cluster user guide. <http://central6.rocksclusters.org/roll-documentation/base/6.1.1/>, 2016. [Online; accessed 26-April-2016; University of California].
- [13] ACI-REF. Research computing - aci-ref, 6 2016.
- [14] J. Fischer, E. Coulter, R. Knepper, C. Peck, and C. A. Stewart. Xcbc and xnit - tools for cluster implementation and management in research and training. In *2015 IEEE International Conference on Cluster Computing*, pages 857–864, Sept 2015.
- [15] Big green cluster. <http://www.marshall.edu/itresearch/resources-and-tools/biggreen-cluster/>, 2016. [Online; accessed 19-April-2016; Marshall University].
- [16] Bigdog hpcc research news items. <http://oit.siu.edu/rcc/bigdog-folder/news-folder/>, 2016. [Online; accessed 19-April-2016; Southern Illinois University].
- [17] Nvidia cuda toolkit. <https://developer.nvidia.com/cuda-toolkit>, 2016. [Online; accessed 27-April-2016; NVIDIA].
- [18] Drmaa. <https://www.drmaa.org>, 2016. [Online; accessed 27-April-2016; DRMAA Working Group].
- [19] Craig A Stewart, Timothy M Cockerill, Ian Foster, David Hancock, Nirav Merchant, Edwin Skidmore, Daniel Stanzione, James Taylor, Steven Tuecke, George Turner, et al. Jetstream: a self-provisioned, scalable science and engineering cloud environment. In *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, page 29. ACM, 2015.
- [20] Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis. User acceptance of information technology: Toward a unified view. *MIS quarterly*, pages 425–478, 2003.