Extending XSEDE Innovations to Campus Cyberinfrastructure - The XSEDE National Integration Toolkit

Eric Coulter
Indiana University
Bloomington, Indiana
jecoulte@iu.edu

Jodie Sprouse
Cornell University Center for Advanced Computing
Ithaca, NY
jhs43@cornell.edu

Resa Reynolds
Cornell University Center for Advanced Computing
Ithaca, NY
resa.reynolds@cornell.edu

Richard Knepper
Cornell University Center for Advanced Computing
Ithaca, NY
rich.knepper@cornell.edu

ABSTRACT

XSEDE Service Providers (SPs) and resources have the benefit of years of testing and implementation, tuning and configuration, and the development of specific tools to help users and systems make the best use of these resources. Cyberinfrastructure professionals at the campus level are often charged with building computer resources which are compared to these national-level resources. While organizations and companies exist that guide cyberinfrastructure configuration choices down certain paths, there is no easy way to distribute the long-term knowledge of the XSEDE project to campus CI professionals. The XSEDE Cyberinfrastructure Resource Integration team has created a variety of toolkits to enable easy knowledge and best-practice transfer from XESDE SPs to campus CI professionals.

The XSEDE National Integration Toolkit (XNIT) provides the software used on most XSEDE systems in an effort to propagate the best practices and knowledge of XSEDE resources. XNIT includes basic tools and configuration that make it simpler for a campus cluster to have the same software set and many of the advantages and XSEDE SP resource affords. In this paper, we will detail the steps taken to build such a library of software and discuss the challenges involved in disseminating awareness of toolkits among cyberinfrastructure professionals. We will also describe our experiences in updating the XNIT to be compatible with the OpenHPC project, which forms the basis of many new HPC systems, and appears situated to become the de-facto choice of management software provider for many HPC centers.

KEYWORDS

Clusters, Scientific Computing, System Administration, OpenHPC, XSEDE, XCRI

https://doi.org/10.22369/issn.2153-4136/10/1/3

1 INTRODUCTION

1.1 Overview of XCRI

The XSEDE Cyberinfrastructure Resource Integration (XCRI) team is charged with helping to provide resources that extend the impact of experienced campus support and computational capacity for support of research activities. XCRI is a sub-group of the XSEDE Cyberinfrastructure Integration (XCI) team within XSEDE. While many campuses provide substantial resources to their users, IT staff are performing a common set of tasks locally that are duplicated at each site. Furthermore, when users move from these local environments to XSEDE SP resources, the change can be disconcerting as environments and software implementations are different, despite being billed as similar systems. By providing a set of consistent instructions and software layouts, XCRI strives to improve the aggregate use of national infrastructure, save local IT staff time and frustration, and support environments that are more similar to those found in the national scale cyberinfrastructure, easing transitions for scientists moving from campus to national computing.

1.2 XCRI Toolkits

One of the core pieces of the XCRI mission is providing software toolkits to enable cyberinfrastructure providers to easily emulate eminent SPs within XSEDE. Most of the toolkits curated by XCRI involve helping ease some type of pain common to a provider of research computing services, such as building an HPC system (The XSEDE Compatible Basic Cluster - XCBC[4, 5], and the Jetstream Virtual Cluster), assisting users with data movement (Ansible scripts to build a local Globus Connect Server), or providing users with scientific software. The last case is the primary driver of XNIT.

1.3 Toolkit Development Process

The XCRI toolkits are not produced in a vacuum, of course. By interfacing with the Requirements Analysis and Capability Delivery (RACD) team in the XSEDE Cyberinfrastructure Integration (XCI) group, we are able to gain access to the needs and preferences of the large XSEDE SPs. RACD is concerned with identifying software or services that will help the XSEDE SPs operate in an integrated way, while XCRI aims to take the lessons learned from XSEDE SPs on a large scale, and make them available to smaller scale institutions. The XCRI toolkit plan is based on the initial set of use cases developed in collaboration between the XSEDE Campus Bridging

team (now known as XCRI) with the XSEDE Architecture team during the first iteration of the XSEDE project [14]. Campus Bridging use cases were largely based on the output of the NSF Advisory Committee for Cyberinfrastructure Task Force Report on Campus Bridging[6], which examined ways to best enable researchers at institutions on all levels of the funding spectrum to use advanced research computing infrastructure with minimal friction. Campus Bridging use cases were later augmented by a set of project-wide "canonical use cases" that cover activities for multiple facets of XSEDE[15].

In creating use cases, XCRI goes through a drafting and review process to ensure that the essential parts of the Use Case will serve its constituents without requiring a particular solution. The Use Case process involves review by multiple parties in XCI, who are familiar with the XSEDE SP network and aware of the many difficulties involved in sharing knowledge between educational institutions. While the Use Case describes an outcome, it does not *prescribe* the solution. At this point in the process, interaction with the larger XSEDE user community is necessary, to determine what technologies are commonly used or coming into use. XCRI also often undertakes surveys with community collaborators to determine their interest level in new toolkits, based on the Use Cases in question. Once a plan is in place, XCRI team members will divide the project, and prepare a delivery mechanism for interested parties to use the selected software. If necessary, "glue code" will be created to ease the installation process, and friendly users will be sought out to ensure the toolkit is easily used by an outside party. At this point, the toolkit is considered completed.

Once a toolkit is completed, the release and distribution process begins. This depends quite heavily on the form the toolkit takes. XCRI toolkits are always linked in the XSEDE Community Software Repository. Toolkits have taken the form of everything from iso files (the original XCBC toolkit, based on Rocks), to Ansible playbooks delivered via a git repository (the current XCBC toolkit). For more frequently updated toolkits, delivery via GitHub offers an easy form of distribution, which invites community feedback. In the case of XNIT, of course, updated packages are simply copied into the yum repository from build servers.

Many of the XCRI toolkits fall under use case CB-02, "Share the XSEDE Environment with Campus Uses"[14] as this is an area with significant challenges for resource-constrained institutions, but still remains amenable to primarily technical solutions without the need for massive institutional buy-in, as is necessary for say CB-06, "Sharing computational resources between campuses". Many problems faced by campuses attempting to expand research computing are in fact political and cross-departmental, which are not well addressed by the provision of toolkits. Something akin to the ACI-REF[1] model of research computing facilitator, or the XSEDE Campus Champion, is more appropriate in tackling such issues (for example, convincing the administration that sharing scientific data with the broader community will be severely hindered if all access to institutional data must be done through local affiliate accounts rather than utilizing the InCommon Federation tools[16].

2 DEVELOPMENT OF XNIT

The idea behind the creation of XNIT was to provide campus cyberinfrastructure adminstrators a way to include software that is commonly found on XSEDE resources, without having to do a complete re-installation of operating system that the XCBC cluster distribution would require. Campus cyberinfrastructure may be set up for any number of purposes and local administrators may have quite salient reasons for configuration choices within the local environment, based on the needs of campus faculty. To satisfy the needs of a wide range of users, XNIT had to provide a set of scientific software that can be installed without overly disturbing local environmental configurations. The XNIT was created as a straightforward yum repository, so that administrators could install scientific software as easily as installing standard linux tools, without forcing anyone to learn to use yet another piece of software.

The XNIT was intended to provide access to rpms of commonly used scientific software that future users of XSEDE software could expect to see on XSEDE resources. In order to facilitate free distribution and extension, packages were all based on open-source software. The earliest list of software included in XNIT was elicited from the XSEDE Campus Champions list, with some discussion and modification by the packaging team led by XSEDE staff at Cornell University. Package maintainers regularly update package files based on new versions as they are released by the developers (which can be few and far between), and add new packages requested by campus CI providers, XSEDE Campus Champions, or other members of the community. Initial testing of XNIT was undertaken on systems at Cornell and Indiana University, with some "friendly-user" testing courtesy of Notre Dame's Center for Research Computing.

3 SPREADING THE WORD

Developing a user community for XCRI offerings represents a unique set of challenges. While there are a number of venues for discussing research computing on campuses, many of these activities focus on the concerns of established centers, and there are few signposts for those looking to start and support a research computing effort, implement the first cluster on campus, or those who are directed to start supporting a new research effort. Campus CI providers that are already engaged with XSEDE are usually (though not always!) able to get information about XSEDE software and practices to incorporate into their own local offerings. Larger conferences such as Supercomputing (SC) are mostly directed at those who are already fairly established in providing HPC offerings to their user base. Previously one venue where the XCRI team has been able to find interested CI providers has been the Educause Regional conference series, although the stream of interactions which lead to XCRI engagements has slowed over time. While it is difficult to find people and institutions seeking the type of help that XCRI can provide, the rewards gained in terms of increasing engagement with national CI, and enabling students and researchers at resource-constrained institutions to work beyond their funding capacity, are incalculable.

It is surprisingly difficult to spread word about free resources in the research computing community, particularly resources aimed at helping people who are not already well-established members.

Site	Date	TeraFlops
Marshall University	April 2015	9
Souther Illinois University	September 2015	35
Bentley University	January 2016	2
University of Texas at El Paso	May 2016	43
Brandeis University	March 2017	200
South Dakota State University	June 2017	10
Slippery Rock University	October 2017	10

Table 1: Table of site visits in which XCRI Staff helped implement or upgrade a physical cluster using the XCBC toolkit. Remote consultations are not listed, but include a variety of activities, including support for existing clusters, XNIT implementation, Virtual Cluster builds, and pointing institutions to other resources if necessary.

While XCRI has access to a vast array of XSEDE resources, it is rare that an institution which is already aware of XSEDE or already providing on-campus computational resources is in need of aid from XCRI. It follows in the same vein that institutions who are sending staff to large HPC conferences such as Supercomputing are rarely (but not never!) looking for help implementing or improving their current resources. The XSEDE Campus Champions community has proven to be a most valuable resource when attempting to spread the word that institutions looking for help can come to XCRI. Having XCRI members involved in the mailing list and providing occasional highlights in the monthly Campus Champion calls are outreach strategies that have resulted in several useful interactions for XCRI. Connections for six of the site visits XCRI has conducted have been made through the Champions' mailing lists. These interactions have also included support of campuses applying for hardware awards from the NSF and other institutions, as well as institutions building or rebuilding hardware they already own. To date, XCRI has written letters of support for five different proposals for cluster acquisitions, pledging to provide an integration site visit for winning campuses. Working with Campus CI implementers to build a community of practice around XNIT and other XCRI toolkits takes time, useful sets of tools, and a critical mass of users. Regular showings at conferences is a primary method of fostering awareness of XCRI offerings in the research community. While conference papers are quite useful for disseminating technical information, we've found that running tutorials aimed at new-to-HPC CI professionals is an excellent way to both support the community and spread awareness of XCRI toolkits. In general, XCRI tutorials and informational webinars have generated the highest attendance, compared to more traditional presentations. Beyond conference gatherings, BoFs, and panels, XCRI has worked to provide permanent venues for discussion and capture of knowledge so that XNIT users can record and exchange information amongst each other. Uptake of these tools has been variable and mostly transient at best. CI providers are most engaged with tools that have low barriers to entry and a bit of current interest, notably Slack, in conversing with each other and sharing their experiences with other providers. In addition to an **#xcri** Slack channel, XCRI has provided forum access on the XSEDE Community Software Repository, an **xcbc-hosts** mailing list, and a wiki area hosted by XSEDE. Thus far the clear winner in terms of user engagement has been the Slack channel, whereas the wiki area, which requires more planning and regular engagement by users and XSEDE staff in order to be useful, has seen little-to-no engagement. The mailing list proved to be a difficult effort in light of the small population involved. Enabling a community of practice takes time and a population that is encouraged to engage by common goals, identity, and participatory spirit[2]. Given the relatively limited population of adopters of XCRI toolkits, considerable work could be put into creating frameworks for collaboration that still fail to reach critical mass. Adopters may feel that their local issues would not be interesting to CI providers at other campuses. XSEDE resources for XCRI are thinly-spread at best, some work to improve the offerings for community collaboration might be provided by a future Campus Champions fellow or other similiar effort.

3.1 Tracking Usage

During the early stages of XNIT, XCRI staff worked closely with two early-adopters at the University of Hawaii and Montana State University. In both cases, the campuses had existing HPC infrastructure which didn't require a full rebuild à la the XCBC, but needed the additional support provided by an easily-installed repository of scientific software. Beyond direct communication with end-users, tracking the uptake of XNIT is very difficult. The most information it's really possible to see is how often, and from where, rpms are downloaded. Over the lifetime of the project, XNIT has attracted roughly 90 regular users across the US, based on simple download numbers.

4 EVOLUTION OF XNIT

In the initial conception of XNIT, the plan was to provide a means by which local campus administrators could easily install the same open-source scientific software as that found on XSEDE resources, concentrated around packages specifically called out by the XSEDE community. In order to provide a widely applicable toolkit, which could work both for established and new systems, the packages were built to be fully relocatable, by providing a "prefix" option in the rpm configuration. While initially this seemed a promising method of satisfying the needs of existing systems, in practice it is actually fairly non-trivial to successfully install an rpm to a non-standard location while also providing users with an easy means of access, without conflicting with existing standard package installations. In addition to the challenges involved in providing relocatable rpms, which often involves tracking down source code and creating ".spec" files from scratch, the adoption of the OpenHPC[13] project for the foundation of the XCBC toolkit led to a fundamental incompatibility between the two toolkits.

Several factors in the ongoing development of the HPC community made it more apparent that XNIT needed to change. After a few site visits involving already established clusters, the team learned that many admins have a need to maintain multiple versions of scientific software, and that the dependency enforcement built into yum, in which only a single version of a package can be available, makes XNIT untenable in that situation. Many site administrators also seem to accept that the burden of building packages will fall on them, and simply use an array of tools such as EasyBuild[8],

Helmod[3], or Spack[7], along with a module environment[8] to manage their scientific software.

With the rise in popularity of container solutions for platform-independent application deployment, and the arrival of HPC-friendly container runtimes such as Singularity[10] and CharlieCloud[12], it became clear that there are now much more powerful means of providing general scientific software. While an immediate switch to containerized software is not in the works, the XNIT project is in the process of pivoting to providing a more focused set of OpenHPC-compatible packages alongside the current relocatable options. By offering packages built with the open-source compilers and MPI implementations available in OpenHPC, XNIT will continue to support existing systems while staying in step with an open-source cluster management system with active development that is often the choice for new deployments based on ease of use and management. In the near future, the XCRI team will also begin to offer containerized versions of some packages, selected based on usage numbers gleaned from the XSEDE-wide XDMoD[11] usage tracking site and other data sources from XSEDE, which will work within the Singularity container runtime. There is also collaboration in the works with one of the lead SingularityHub developers, to offer a container registry tool that will allow access to both local and non-local containers, and easy interoperability with the Slurm[9] resource manager, used in the XCBC and a large segment of HPC systems.

5 LESSONS LEARNED

Since the initial release of XNIT toolkit in 2012, the team has taken several lessons from working with the community. First and foremost, it is extremely hard to gain live feedback from a community without a dedicated engagement team. XSEDE External Relations and Broadening Participation teams have been vital in helping produce materials and identify venues for outreach in the areas where new offerings for campus cyberinfrastructure will be well-received. The Indiana University Center for Engagement and Support has been invaluable to the XCRI team in managing and collecting survey data from the relevant groups, despite the challenges in surveying such small numbers. It is equally hard to observe usage of a service when the main interface is a passive yum repository. While this is excellent for ease-of-use, it provides little visibility past a list of IP addresses accessing the repository.

Second, it is not always the case that soliciting user requests will result in a tool with long-term uptake. Even after designing the toolkit based on community input, it still required a great deal of outreach in order to keep the toolkit visible to the community. Requirements alone are not sufficient for generating a toolkit, as a number of potential users are interested and want to move into cyberinfrastructure projects, but have little to guide them as to what will support those activities. It is important for a potential toolkit developer to take requirements, direction of technological initiatives, and goals and needs of users into account. In light of this, taking advantage of tools like XDMoD[11], which gives detailed job information for the whole of the XSEDE SPs, is essential. Detailed reporting like this makes it possible to see what the community is *really* using without the necessity of collecting user responses in a survey format.

Third, it is absolutely necessary for software toolkits to be built in such a way that it is possible to evolve the delivery mechanism based on user needs and preferences. While the demand for scientific software remains, the desire for such software delivered in rpm form has greatly dwindled. Due to the relative inflexibility of yum, and lacking a module system such as is available in OpenHPC, actually using XNIT is much more cumbersome than initially planned. In an age where software build solutions run rampant (such as Spack[7],EasyBuild[8], and Helmod[3]), cyberinfrastructure administrators have a wide array of flexible options at their fingertips.

Moving to a more flexible, extensible solution for scientific software delivery is a must for XNIT to continue providing a useful solution to easing the pain of administrating a general HPC-style resource at the campus level.

6 CONCLUSION

THE XCRI team occupies a unique space in the national cyberinfrastructure environment. Neither SP nor software developers, we stand at the edges of the research computing world, trying to help new or under-resourced institutions find their way towards providing high-quality resources to their scientists. With a mandate to primarily provide software toolkits, it can be challenging to keep up efforts to make newcomers to the research computing community aware of our presence. These same challenges appear through the HPC education community. Through a combination of white papers, tutorials, and engagement with online communities, however, it *is* possible to reach the relevant population.

With the ever changing nature of computing, it is also necessary to continuously update the offerings that XCRI makes available. In the case of XNIT, it has been not only necessary to change the software we offer (by keeping the repository up-to-date and being open to adding new software as requested), but to completely change the model of *how* that software is delivered. By keeping extant packages viable, offering an installation method that stays in step with a modern HPC solution, and beginning to offer truly system-independent container software, the XSEDE National Integration Toolkit will continue to evolve, and broaden the reach of resources available to all institutions taking part in the national research computing arena.

REFERENCES

[1] ACI-REF. 2016. Research Computing - ACI-REF. https://rc.fas.harvard.edu/partnerships/aci-ref/

[2] Pete Bradshaw, Stephen Powell, and Ian Terrell. 2004. Building a community of practice: Technological and social implications for a distributed team. In *Knowledge networks: Innovation through communities of practice*. IGI Global, 184–201.

[3] Harvard Research Computing. 2018. HeLmod Package Manager. https://www.rc.fas.harvard.edu/helmod/

[4] J. Fischer, E. Coulter, R. Knepper, C. Peck, and C. A. Stewart. 2015. XCBC and XNIT - Tools for Cluster Implementation and Management in Research and Training. In *2015 IEEE International Conference on Cluster Computing*. 857–864. https://doi.org/10.1109/CLUSTER.2015.143

[5] Jeremy Fischer, Richard Knepper, Matthew Standish, Craig A. Stewart, Resa Alvord, David Lifka, Barbara Hallock, and Victor Hazlewood. 2014. Methods For Creating XSEDE Compatible Clusters. In *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment (XSEDE '14)*. ACM, New York, NY, USA, Article 74, 5 pages. https://doi.org/10.1145/2616498.2616578

[6] NSF Advisory Committee for Cyberinfrastructure Task Force on Campus Bridging. 2011. *Final Report*. NSF. http://www.nsf.gov/od/oci/taskforces/

TaskForceReport_CampusBridging.pdf

[7] Todd Gamblin, Matthew LeGendre, Michael R. Collette, Gregory L. Lee, Adam Moody, Bronis R. de Supinski, and Scott Futral. 2015. The Spack Package Manager: Bringing Order to HPC Software Chaos. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '15)*. ACM, New York, NY, USA, Article 40, 12 pages. https://doi.org/10.1145/2807591.2807623

[8] Markus Geimer, Kenneth Hoste, and Robert McLay. 2014. Modern Scientific Software Management Using EasyBuild and Lmod. In *Proceedings of the First International Workshop on HPC User Support Tools (HUST '14)*. IEEE Press, Piscataway, NJ, USA, 41–51. https://doi.org/10.1109/HUST.2014.8

[9] Morris A. Jette, Andy B. Yoo, and Mark Grondona. 2002. SLURM: Simple Linux Utility for Resource Management. In *In Lecture Notes in Computer Science: Proceedings of Job Scheduling Strategies for Parallel Processing (JSSPP) 2003*. Springer-Verlag, 44–60.

[10] Gregory M. Kurtzer, Vanessa Sochat, and Michael W. Bauer. 2017. Singularity: Scientific containers for mobility of compute. *PLOS ONE* 12, 5 (05 2017), 1–20. https://doi.org/10.1371/journal.pone.0177459

[11] Jeffrey T. Palmer, Steven M. Gallo, Thomas R. Furlani, Matthew D. Jones, Robert L. DeLeon, Joseph P. White, Nikolay Simakov, Abani K. Patra, Jeanette Sperhac, Thomas Yearke, Ryan Rathsam, Martins Innus, Cynthia D. Cornelius, James C. Browne, William L. Barth, and Richard T. Evans. 2015. Open XDMoD: A Tool for the Comprehensive Management of High-Performance Computing Resources. *Computing in Science & Engineering* 17, 4 (2015), 52–62. https://doi.org/10.1109/MCSE.2015.68 arXiv:https://aip.scitation.org/doi/pdf/10.1109/MCSE.2015.68

[12] Reid Priedhorsky and Tim Randles. 2017. Charliecloud: Unprivileged Containers for User-defined Software Stacks in HPC. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '17)*. ACM, New York, NY, USA, Article 36, 10 pages. https://doi.org/10.1145/3126908.3126925

[13] Karl W. Schulz, C. Reese Baird, David Brayford, Yiannis Georgiou, Gregory M. Kurtzer, Derek Simmel, Thomas Sterling, Nirmala Sundararajan, and Eric Van Hensbergen. 2016. Cluster Computing with OpenHPC. http://hdl.handle.net/2022/21082.

[14] Craig A. Stewart, Richard Knepper, Andrew Grimshaw, Ian Foster, Felix Bachmann, David Lifka, Morris Riedel, and Steven Tueke. 2012. *Xsede campus bridging use cases*. Technical Report. XSEDE, Tech. Rep.

[15] XCUC 2013. XSEDE Canonical Use Cases. https://www.ideals.illinois.edu/handle/2142/43877. Online; accessed 19-April-2018; University of Illinois Urbana-Champaign.

[16] XSEDE InCommon Page 2018. XSEDE InCommon Identity Provider. https://www.xsede.org/ecosystem/operations/security/incommon. [Online; accessed 24-March-2018; XSEDE].