XSEDE Campus Bridging – Cluster software distribution strategy and tactics

15 April 2013

Version 1.0

Available from: http://hdl.handle.net/2022/15459

Table of Contents

A. Document History

Overall Document Authors:

Victor Hazlewood
National Institute for Computational Sciences, University of Tennessee
Oak Ridge National Laboratory
PO Box 2008, BLDG 5100
Oak Ridge, TN 37831-6173
vhazlewo@utk.edu

Steven Lee
Cornell University
512 Frank H. T. Rhodes Hall
Ithaca, NY 14853
shl1@cornell.edu

JP Navarro
Argonne National Laboratory
9700 S. Cass Avenue
Argonne, IL 60439
navarro@mcs.anl.gov

Richard Knepper
Indiana University
2709 E 10th St
Bloomington IN 47408
rknepper@iu.edu

David Lifka
512 Frank H.T. Rhodes Hall
Cornell University
512 Frank H. T. Rhodes Hall
Ithaca, NY 14853
lifka@cac.cornell.edu

Craig A. Stewart
Indiana University
2709 E 10th St
Bloomington IN 47408
stewart@iu.edu

	Version	Date	Changes	Author
Entire Document	0.1	11/12/2012	Initial draft	Lifka et al
Entire Document	0.2	12/1/2012	Additional drafting	Knepper
Entire Document	0.3	12/21/2012	And yet more editing	Stewart et al
Entire Document	0.4	12/21/2012	Responding to comments from Lifka & Towns	"
Entire Document	0.5	1/7/2013	Integrating feedback from Lifka and campus bridging discussions	"
Entire Document	0.6	1/25/2013	Checking puppet/cobbler/yum statements, response to comments from Scott Lathrop, timetable and communications plan added	Knepper
Entire Document	1.0	4/15/2013	Adding test and validation plans	Knepper, Navarro

B. Document Scope

This document is both a public document and an internal working document intended to define XSEDE strategies related to XSEDE's cluster build software distribution project. This is part a strategy document, part tactical.

This document is one component of a process that generates at least the following documents, some of which are public and some that are, as of now, intended to be internal working documents:

Public Documents:

- Stewart, C.A., R. Knepper, A. Grimshaw, I. Foster, F. Bachmann, D. Lifka, M. Riedel and S. Tuecke. *XSEDE Campus Bridging Use Cases*. 2012. http://hdl.handle.net/2022/14475
- Stewart, C.A., R. Knepper, A. Grimshaw, I. Foster, F. Bachmann, D. Lifka, M. Riedel and S. Tuecke. *Campus Bridging Use Case Quality Attribute Scenarios*. 2012. http://hdl.handle.net/2022/14476
- Craig A. Stewart, Richard Knepper, James Ferguson, Felix Bachmann, Ian Foster, Andrew Grimshaw, Victor Hazlewood, and David Lifka. 2012. 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). ACM, New York, NY, USA, Article 47, 8 pages. DOI=10.1145/2335755.2335844 http://doi.acm.org/10.1145/2335755.2335844

Internal documents:

- Stewart, Craig A. *Campus Bridging Use Case - Initial Prioritization.* 2012. http://hdl.handle.net/2022/15214
- A binary mapping of use cases to Requirements in DOORS (a binary mapping means that for each use case a "yes" or "no" flag indicating whether a particular requirement within the full list of requirements is or is not required to enable a particular use case).
- A set of level 3 decomposition documents, which include:
  - Quality Attributes descriptions
  - Connections diagram in UML

C. Campus Bridging – Background to the cluster software distribution project

The XSEDE campus bridging cluster software distribution project is based on the following use case:

UCCB 2.0	**Enable economies of scale in usability and training for XSEDE and campus resources through dissemination of information and tools**
Description	Make it easier for on-campus users to use XSEDE resources, and make it easier for all to create high quality and reusable training materials, by taking steps that make it possible for campus clusters and other resources to be more like XSEDE resources and thus easier to document, learn, understand, and use.
References	http://creativecommons.org/licenses/by/3.0/ Creative Commons Attribution 3.0 Unported license (CC BY 3.0)
Actors	• XSEDE: Senior Leadership and SP Forum • XSEDE: administrators of all Level 1 and 2 resources • XSEDE: SD&I / A&D • XSEDE: documentation and support teams • XSEDE: campus bridging • Campus: campus systems administration staff • Campus: instructors and learners
Prerequisites (Dependencies) & Assumptions	XSEDE Architecture and Design (A&D), Software Development and Integration (SD&I), Senior Leadership, and Service Provider (SP) Forum have an agreement on basic aspects of system implementation – directory hierarchy, locations of standard software kits and optional locally-installed or user-contributed software – and that compliance with this set of standards is uniform across at least all Level 1 and 2 SPs.

UCCB 2.0	Enable economies of scale in usability and training for XSEDE and campus resources through dissemination of information and tools
Steps	• XSEDE must create a standard way to document system characteristics and software configurations. A document template, in an editable format, must be released with a license that allows re-use and modification, such as the Creative Commons CC BY 3.0 license. • XSEDE must create and disseminate training materials in ways that allow them to be minimally altered and used by on-campus users. All materials must be released with a license that allows reuse and modification, such as the CC BY 3.0 license. • XSEDE Campus Bridging should create a "ROCKS Roll" distribution that allows a campus-based sysadmin to install a cluster that includes the open source elements of a basic XSEDE cluster configuration using ROCKS. • As part of the creation of a ROCKS Roll distribution, documentation should be prepared that defines how a generic, XSEDE-like cluster would be configured. This documentation will enable systems administrators who do not wish to use the ROCKS approach to still configure systems to be as similar as possible to the least specialized of the major XSEDE Level 1 resources. This should be presented as a guide to current practice and NOT a standards document. The former is what we can realistically aspire to; the latter is not obviously beneficial and standards development processes often take so long and are so much work that they are not effective use of time UNLESS one really is describing some sort of fundamental standard (which is not the case here).
Variations (optional)	OSG is moving to an RPM-based distribution mechanism for OSG software. An RPM-based mechanism for distribution should be considered as a supplement to or alternate for a ROCKS-based distribution.

This document also supports portions of Campus Bridging use cases 4.0, 5.0, and 6.0:

UCCB 4.0	**Use of data resources from campus on XSEDE, or from XSEDE at a campus**
Description	Support analysis of data integrated across campus-based and XSEDE-based resources.
Steps	Basic case (A): Movement of data from campus resource to XSEDE, and back to campus • User has data resource(s) on a campus resource they wish to access from or at an XSEDE Level 1 or 2 resource for analysis and/or visualization. Access may be accomplished by either direct remote access or by transferring file to local storage with local access. Examples of data resources include a flat file, tar ball, database to be moved wholesale, an extract from a database, or a file looked up via a metadata database. • User reads data located on a campus resource from an XSEDE resource. • User analyzes and/or visualizes data on XSEDE resource. • User writes/updates/deletes data back to campus resource.
Variations (optional)	Variant (B): User has generated data resource(s) on an XSEDE resource and wishes to transfer them to campus • User analyzes and/or visualizes data on XSEDE resource. • User writes/updates/deletes data back to campus resource. Variant (D): Synchronization of copies of data between campus and XSEDE resource • User identifies a data set that s/he wishes to maintain, in a synchronized fashion, on one campus resource and one or more XSEDE resources. • User makes a change to one version of the file, and the other copies are automatically updated.

UCCB 5.0	Support for distributed workflows spanning XSEDE and campus-based data, computational, and/or visualization resources
Description	Enable distributed workflows – interactively or in batch mode – possibly spanning XSEDE and campus cyberinfrastructure resources, without user intervention after workflow is initiated.
Steps	Variant (A): Interactive management of workflows • User wants to perform an analysis with a distributed workflow and has a workflow (directed acyclic graph) where vertices will execute on different resources at different locations. Vertices consist of jobs that may involve stage in/stage out and direct access allowing create-read-update-delete (CRUD) access to remote (remote to the locus of execution) data resources. • User starts an interactive session with a workflow tool, and it accesses data sources, computational tools, visualization resources, and data transfer tools on a variety of resources. Some of those resources are XSEDE resources, and some are local campus-based resources that the user accesses with local credentials. File I/O is sometimes limited to only a single process accessing a file any given time. Ability for simultaneous I/O by multiple processes to a single file is a (relatively rare) requirement, and in this case the user/program is responsible for file integrity. This implies ability for multiple sources to read/write data simultaneously (user responsible for housekeeping). • Job completes, user initiates new workflow interactively or stops.
Variations (optional)	Variant (B): Distributed workflows in batch mode • As above, but in batch mode, with notification to user when workflow has successfully completed or failed. Variant (C): Support for distributed workflows initiated via Science Gateways • As above, but mediated by a Science Gateway that accesses XSEDE and campus-based resources. Access to campus resources is handled with user credentials (e.g. not as part of a sharing arrangement, as described in UCCB 6.0). XSEDE must document the types of credentials that science gateways must support for campus access. Since the access is mediated through the science gateway, the credentials must support delegation from the user to the science gateway. OAuth provides a standard protocol for delegation (see: www.sciencegatewaysecurity.org).

UCCB 6.0	Shared use of computational facilities mediated or facilitated by XSEDE
Description	XSEDE can provide tools and mediate relationships that enable the US to make better use of its aggregate cyberinfrastructure resources.
Steps	Variant (A): Creation and use of a Shared Virtual Compute Facility (SVCF) – Multiple researchers or groups have campus-based compute resources they are willing to expose to each other (subject to access control), and this group manages the internal economics of the exchanges. • Participants create virtual clusters, virtual high throughput computing facilities (e.g. condor flocks), virtual clouds, and/or other sort of virtual resources based on campus compute resources at one or more campuses. • Participants install on their resources a capability kit that implements InCommon-based authentication in ways that maintain the basic functionality, look, and feel as XSEDE-like authentication, but without reference to XSEDEDB for authorization or accounting. • Participants manage accounting, value exchanges, policy compliance, and security response. • Participants must have the ability to, on their own, create groups and set access control to resources based on groups. • Note: The entity operating a Shared Virtual Compute Facility would not need to be (and may not want to be) an XSEDE Level 3 Service Provider as defined in the Service Provider Forum charter (https://www.xsede.org/documents/10157/281380/SPF_Definition_v10.1_120228.pdf).

UCCB 6.0	Shared use of computational facilities mediated or facilitated by XSEDE
Variations (optional)	Variant (B): An organization (virtual or otherwise) becomes a Level 3 Service Provider and contributes access to campus-based resources via a Shared Virtual Compute Facility (SVCF) in return for in-kind use of XSEDE resources later. • An organization (virtual or otherwise) operates an SVCF and is willing to allow usage of that SVCF by users with XSEDE credentials and allocations (that is, outside the group operating the SVCF) in return for later ability for contributor of resources to obtain cycles via XSEDE in kind. • The organization (virtual or otherwise) providing resources is willing to become an XSEDE Level 3 Service Provider, and has a particular resource, or creates virtual clusters, condor flocks, virtual clouds, and other sort of virtual resources. • XSEDE creates and distributes a capability kit for implementation of InCommon-based authentication in ways that maintain the basic functionality, look, and feel of XSEDE-like authentication, with authorization and accounting done with reference to XSEDEDB. • SVCFs have this capability kit installed and in operation. • XSEDE provides security notification responsibilities in case there is a security breach related to accounts or services that use campus-based authentication mechanisms. • XSEDE has ability to manage exchange rates between campus-contributed resources and resources campuses might AND ability for XSEDE to provide cycles per some Service Level Agreement back to the contributors. • Integrated ticket management – expanded to include local trouble ticket system of campuses that are providing resources.

D. Prioritization and strategy

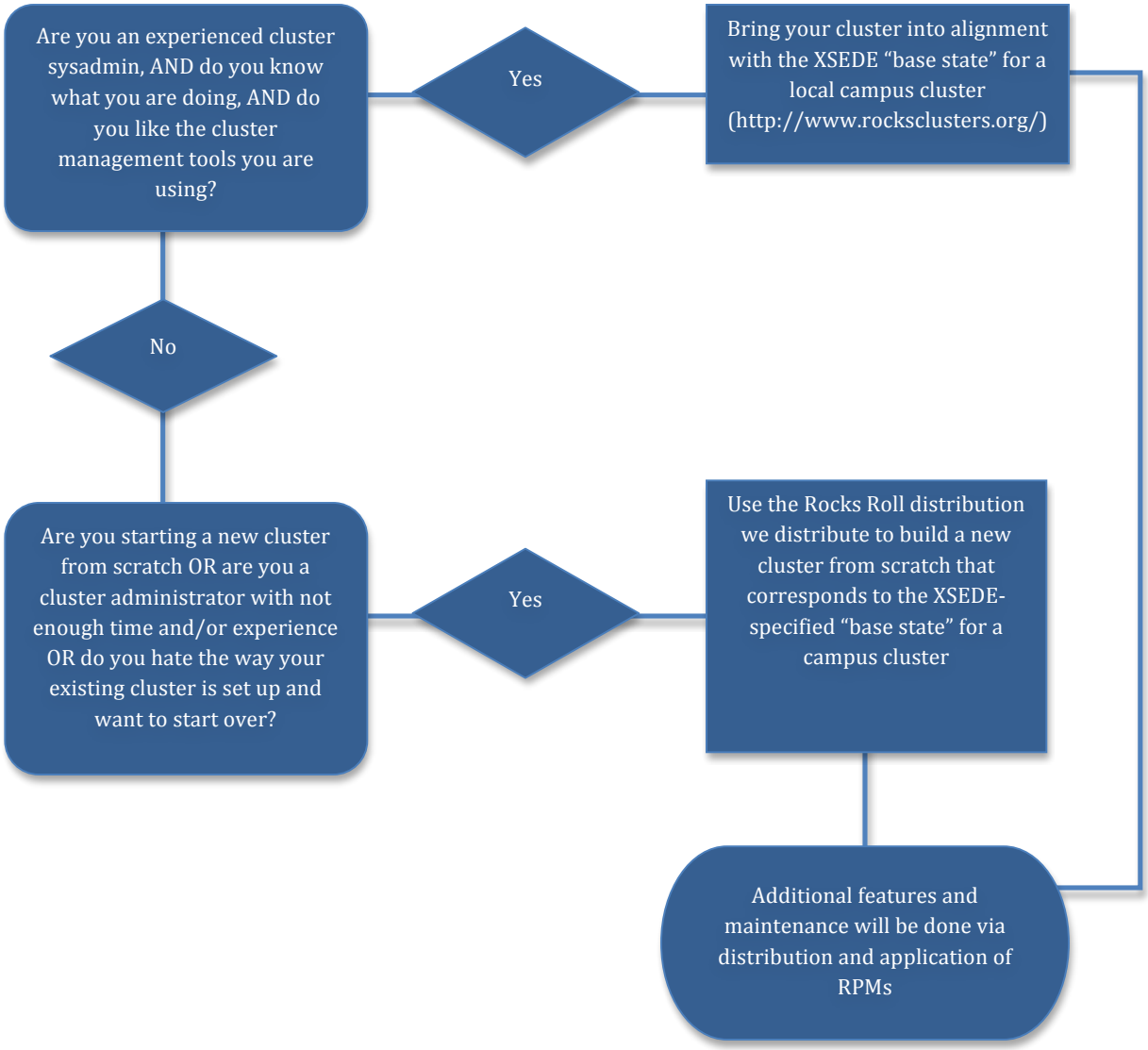
The basic goals of this project can be subdivided into three phases, as follows:

1) **Promulgate XSEDE-like cluster setups on campuses throughout the US.** Distribute tools for creating and maintaining clusters on campuses so that they seem as much like an XSEDE cluster as possible. This achieves the following strategic objectives related to XSEDE's mission:
    a. Facilitates the most effective use of the nation's cyberinfrastructure resources by making it easier for staff at campuses to manage their clusters effectively.
    b. Achieves economies of scale in use of XSEDE training materials and training materials created on campuses. As the nation's population of campus clusters comes to look more and more like XSEDE, the reusability of training materials created within XSEDE and on campuses will increase.

    c. Improves user experiences and user ability to move from campus clusters to XSEDE and back, since the computing environment on campuses and XSEDE will grow more like each other over time if this project is successful.
2) **Enable integration from campuses to XSEDE.** This is very much a strategy based on enabling a variety of tactical objectives. The tools that provide this sort of functionality within XSEDE are still evolving and being developed, but possible functionality and tools that we could support during PY2 and PY3 include the following:
    a. Use tools such as GFFS, EBS, and Globus Online to submit jobs from local clusters to XSEDE resources, and/or move files from local clusters to XSEDE resources
    b. Implement InCommon-based authentication to access XSEDE resources
    c. Synchronize copies of files stored on XSEDE resources
3) **Enable integration from XSEDE to campus clusters.** As with Phase 2, tools in this area are still in the early stages. Tools and functionality that might be supported in Phase 3 – most likely starting in PY3 – include the following:
    a. Running jobs submitted to XSEDE on local campus clusters (not funded or operated by XSEDE)
    b. Making data resources available from campus clusters available as scientific resources from within any XSEDE system

Phases 2 and 3 build on the foundation of clusters on campuses set up in ways similar to XSEDE clusters, and enhance integration of campus clusters with XSEDE by distributing software tools and modules that will aid the XSEDE <-> campus cluster integration via the steps called for in the XSEDE campus bridging use cases and project plans.

A basic strategy of "all you have to do is rebuild all of your clusters with a new tool you don't know" is likely to be successful at schools with very small staffing levels, but not likely to get very far with campuses that have already invested in infrastructure and set up resources. Campus Bridging, with input from Operations, Security, and SD&I, will instead take the approach of identifying and documenting a basic state for setting up a campus cluster, dealing only with the local cluster setup and based on one of the XSEDE clusters (or general XSEDE practice) as a model. We will proceed with this project on the basis of the following very simple algorithm, shown in the diagram below:



All packages/features/capabilities will be available via an XSEDE Red Hat Package Manager (RPM) repository (yum-accessible package repository and web site with documentation). This strategy is based in large part on following the experience and leadership of the Open Science Grid (OSG), which is successfully switching to an RPM-based model for distribution of OSG software. It also recognizes that while ROCKS is a great tool for setting up clusters, use of RPMs and tools such as Cobbler and Puppet offer better mechanisms for provisioning and configuring systems, respectively.

Software and tools that will achieve the data and operational integration of campus clusters and XSEDE will in this strategy be released via RPMs. This strategy should allow much more rapid adoption of the tools we are distributing than one based solely on use of ROCKS.

Providing the building blocks (software packages, software distribution mechanisms, documentation, and web-based training) to enable campus system administrators to deploy an HPC cluster that looks, feels, and behaves, as closely to an XSEDE computing resource as possible is a more difficult challenge than it seems at first blush. There is a lot of variety in XSEDE; not all XSEDE provided software is free or available for all hardware configurations; and XSEDE systems are not general purpose, serving all same users and applications the same way. Furthermore, basic system configurations such as choice of operating system or job management tools had never been standardized or identified in the documentation inherited from TeraGrid. To narrow the scope of this effort accordingly we are targeting systems that closely resemble HPC clusters having the following attributes:

- Intel/AMD based servers
- Linux-based operating systems
- Local disk space that support OS, swap, and scratch
- Standard interconnects (Ethernet)
- Support for InfiniBand (not required)

D.1. Software delivery mechanism

Campus Bridging intends to provide a repository of standard Linux RPMS for all open source or publically available software components, provided by David Lifka's team at Cornell and maintained by Stephen Lee in Lifka's group. In addition we will provide documentation and web-based training on the use of Puppet (http://puppetlabs.com/) for system configuration and maintenance and Cobbler (http://cobbler.github.com/) to install the software components on HPC cluster servers. We believe that Puppet and Cobbler provide a robust and flexible mechanism to help install an HPC cluster and/or modify/update an existing HPC cluster. There are many HPC cluster deployment packages available today and we believe that Puppet and Cobbler should work with most, if not, all of these.

D.2. Integration of campus resources with XSEDE

A critical part of the overall strategy as regards the software distribution project is to use it as a platform for dissemination and adoption of software that promotes integration of campus cyberinfrastructure and XSEDE. The economies of scale in consistency, use and reusability of training materials, and the ease of use created for scientists are all good and important things. The most critical benefits arise from the creation of a network of clusters on campuses that can be integrated reliably, in terms of functionality and modes of operation, with XSEDE. This view is foundational to the understandings between the Campus Bridging team, the A&D team, the SD&I team, and Operations. Operations will not do approval / quality checks on the basic cluster software. If CentOS, gcc, MOAB, and Torque don't work properly, it indicates large problems in the software package. Operations will check and approve the distribution mechanisms, to ensure that they function properly, but not the packages that are included in the basic cluster build.

Operations will review and give final approval to all software that has XSEDE integration as part of its function. The Campus Bridging team's software packagers at Cornell will provide documentation and support for the installation process. As a result of our efforts to ensure that XSEDE provides quality software and documentation, we are, as a group, behind where we expected to be in terms of deployment of GFFS and Unicore clients. This project will create the foundation on which the XSEDE Operations team will be able to deploy Globus Online end points, GFFS, and Unicore clients. These will serve as a foundation for other technical and administrative relationships between campus clusters and XSEDE, and this is critical to the broader XSEDE strategy to serve as an organizing agent in the US national cyberinfrastructure.

D.3. Communications plan

Jim Ferguson from NICS will manage communications about the software packaging project as part of his overall responsibilities for communications about the Campus Bridging team's efforts. Jim's responsibilities include connecting with XSEDE Extended Support for Training, Education, and Outreach (ESTEO); Campus Champions; and SD&I teams in order to discuss Campus Bridging initiatives with those groups, submitting items to the XSEDE Campus Bridging discussion forums, and gathering and posting notes from these meetings and forum responses to the XSEDE staff wiki in order to inform the Campus Bridging team of new information, reactions to the program, and possibilities for partnership.

Barbara Hallock from IU will offer presentations on the software program and Campus Bridging team's efforts via videoconference and teleconference, as well as recording a software project video in the same format as previous videos created by the Campus Bridging team and currently hosted on the XSEDE website.

D.4. Plan for software repository

In discussions with the SD&I team, it has been agreed that the software.xsede.org site can be broadened to hold additional software for the XSEDE management teams as well as the broader community. The proposed schema for the software repository is:

Operations-owned production software	https://software.xsede.org/packages/production/
SD&I-owned development software	https://software.xsede.org/packages/development/
Campus Bridging-owned cluster distribution (Cornell)	https://software.xsede.org/packages/cb/

The respective teams listed above would own files under each directory. Each team could also deploy html documentation under their directory.
We would further agree to use the following sub-directory convention:
<distribution>/<architecture>/
Where <distribution> = centos5, centos6, or sles11
Where <architecture> = x86_64

For example:
https://software.xsede.org/packages/cb/centos5/x86_64/

D.5. Plan for software validation activities

Operations and SD&I have both asserted that there are insufficient resources in each of their areas to test and validate the installation of ROCKS Rolls, both in terms of FTEs available to complete validation and in terms of available ROCKS clusters to test package installs. As a result, SD&I has agreed to assist the Cornell team in creating a test plan with which to validate package installations on hardware available to their team. When the packages are able to pass the test plan, they can be placed at the software.xsede.org website for distribution first to friendly user sites and then to a broader range of testers. (The Cornell team is currently using their own subversions service to create packages.)

D.6. Timeline for software packaging activities

- Phase 1
  - What: Creation of ROCKS Rolls, RPMS, Cobbler & Puppet recipes
  - When: Complete at end of PY2 (July 2013)
  - Who
    - SD&I/Operations/Campus Bridging teams create strategy and finalize list
    - Campus Bridging (Stephen Lee & others at Cornell) package and distribute software
    - Operations tests installation
    - Campus Bridging provides support and training
- Phase 2
  - What: Upward integration (from campuses up to XSEDE)
  - When: Complete at end of Q2 PY3 (December 2013)
  - Who:
    - Campus Bridging advocates
    - Operations and SD&I implement (as well as SPs)
    - Campus Bridging provides support, training, and distribution of software packages
- Phase 3
  - What: Outward integration (from XSEDE to campuses)
  - When: Complete at end of PY3 (July 2014)
  - Who:
    - Operations/SD&I/Campus Bridging determine tools from XSEDE stack
    - Operations tests
    - Campus Bridging provides distribution, support, and training

D.7. Current DRAFT software list

That current draft list of software components to be included in the software distribution for campus clusters follows. (A question mark indicates a software package still under consideration for interoperability and not fully decided upon.) The first several sections are part of the basic software distribution; after that is the first draft of XSEDE integration software suites.

- Operating System, basic management tools, & job management
  - CentOS and/or Scientific Linux
  - modules
  - apache-ant
  - fdepend
  - gmake
  - gnu-make
  - scons
  - Torque / MOAB
- Compilers, libraries, and programming tools
  - charm++
  - compiler-c-gnu
  - compiler-f77-gnu
  - compiler-f90-gnu
  - compiler-f95-gnu
  - cuda
  - cuda_SDK
  - fftw2
  - fftw3
  - gcc
  - gmp
  - gotoblas
  - gotoblas2
  - grvy
  - hdf5
  - hecura
  - hypre
  - java
  - jdk32
  - jdk64
  - kojak
  - libflame
  - mpfr
  - mpich
  - mpi4py
  - mpiP
  - openmpi
  - papi
  - perfexpert
  - petsc
  - python
  - tao

    - tau
    - tcl
- Scientific Applications
    - abyss
    - acml
    - amber
    - arpack
    - atlas
    - autodock
    - bedtools
    - BioPerl
    - blast
    - boost
    - bowtie
    - bwa
    - darshan
    - desmond
    - elemental
    - espresso
    - gamess
    - gatk
    - glpk
    - gnuplot
    - gromacs
    - gulp
    - gxmap
    - hmmer
    - lammps
    - meep
    - mpiblast
    - mrbayes
    - namd
    - nco
    - netcdf
    - nwchem
    - octave
    - pdtoolkit
    - phlawd
    - picard
    - plapack
    - plplot
    - pnetcdf

    - saga
    - samtools
    - scalapack
    - shrimp
    - siesta
    - slepc
    - soap
    - sratoolkit
    - sundials
    - trilinos
    - vasp
    - vmd
    - libgtexutils
    - lua
    - ncl_ncarg
    - numpy
    - phdf5
    - sparsehash
    - sprng
    - valgrind
- XSEDE software – as beginning of Phase 2: Integration of campuses with XSEDE:
    - tginfo
    - (tgwheream, tgwhatami) GSI OpenSSH w/ HPN
    - MyProxy client
    - Common User Environment "CUE"
    - GFFS
    - globus-client (globus-url-copy)
    - irods
    - uberftp
    - condor
    - condof-g
    - mycluster
    - glue2
    - glue2-secure
    - Globus Online server endpoint
    - globus-client (gram5)
    - Unicore client