

Campus Bridging Pilot Final Report

Rich Knepper
Indiana University

30 September 2013

[Version 0.4]

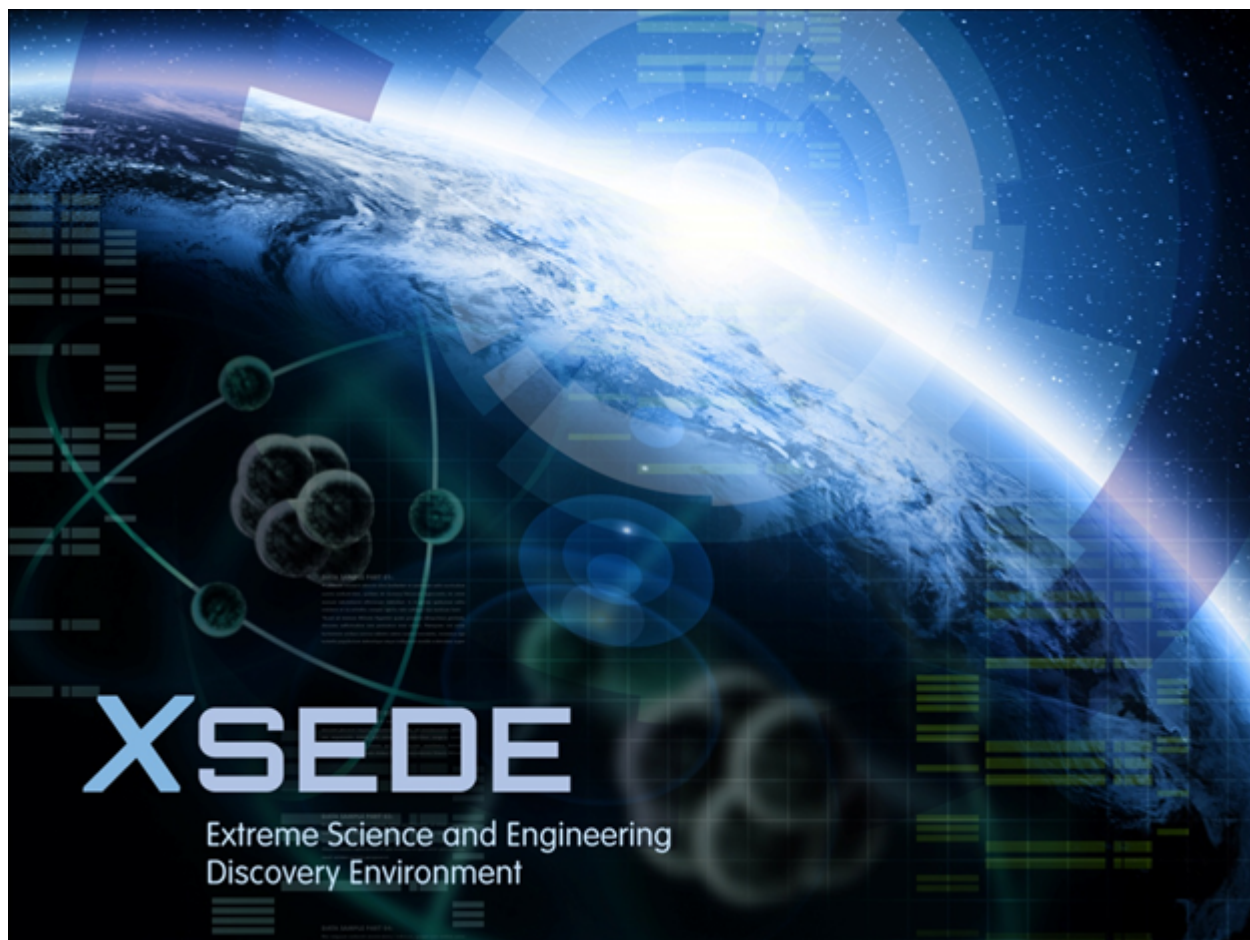


Table of Contents

A.	Document History	3
B.	Document Scope.....	3
C.	Introduction	4
C.1.	Use Cases.....	5
C.1.1.	UCCB 4.0: Use of Campus Data Resources on XSEDE or from XSEDE at a Campus	5
C.1.2.	UCCB 6.0 Shared Use of Computational Facilities Mediated or Facilitated by XSEDE	7
C.2.	Software Environment.....	9
C.2.1.	Overview of Genesis II Software Components	9
C.2.2.	Overview of UNICORE Software Components.....	9
C.3.	Campus Bridging Pilot Results	10
C.3.1.	System Administrator Results	10
C.3.2.	Pilot Participant Results.....	11
D.	2013 Campus Bridging Pilot Evaluation Report.....	13
D.1.	Background and Context	14
D.2.	Data Sources.....	14
D.2.1.	Interviews	14
D.2.2.	Online Survey.....	14
D.3.	Findings.....	15
D.3.1.	Strengths of Campus Bridging Pilot.....	15
D.3.2.	Benefits of Campus Bridging Technology	15
D.3.3.	User-Friendliness.....	16
D.3.4.	Suggestions for Improvement	17
D.3.5.	Future Applications	18
D.4.	Recommendations.....	19
D.5.	Appendix A: Campus Bridging Pilot Interview: Protocol	20
D.6.	Appendix B: Campus Bridging Pilot Interview: Online Survey.....	21

A. Document History

Relevant Sections	Version	Date	Changes	Author
Entire Document	0.1	08/05/2013	Baseline	Richard Knepper, Andrew Grimshaw, Victor Hazlewood, Craig Stewart, Jim Ferguson, Barbara Hallock, Lorna Rivera, Yashema Mack, Guy Almes, Thorbjorn Axelsson, Tom Bishop, Nikolaos Trikoupis, Warner Baringer
Multiple sections	0.2	8/19/2013	Changed title, made changes to section C.1 to include UNICORE and rewrote section C.2	Victor Hazlewood
throughout		9/11/2013	Minor edits, mostly comments	Bishop, Tom
User evaluations		9/23/2013	Added in text on participant evaluations	Stewart, working from input from Lizanne DeStefano & Lorna Rivera

B. Document Scope

This document describes the XSEDE campus bridging pilot that was conducted from January 2012 to July 2013, the relevant XSEDE use cases for the campus bridging pilot, and the experiences of the system administrators and users of the software deployed for the campus bridging pilot. This document is intended to inform the XSEDE engineering process about the implementation and use of the campus bridging software and to identify any additional documentation that XSEDE users and resource providers might need in using the campus bridging software.

C. Introduction

In early 2009 the National Science Foundation's (NSF) Advisory Committee for Cyberinfrastructure (ACCI) charged six different task forces with making strategic recommendations to the NSF in strategic areas of cyberinfrastructure. One of those task forces was Campus Bridging, which led to the following general definition of campus bridging:

Campus bridging is the seamlessly integrated use of cyberinfrastructure operated by a scientist or engineer with other cyberinfrastructure on the scientist's campus, at other campuses, and at the regional, national, and international levels as if they were proximate to the scientist, and when working within the context of a Virtual Organization (VO) make the 'virtual' aspect of the organization irrelevant (or helpful) to the work of the VO. ¹

In more general terms, the goal of campus bridging is to create a sense of “virtual proximity” — so that any resource a researcher or student is using feels as if it were a peripheral plugged into their laptop. A set of use cases has been identified as priorities for XSEDE, and includes these high-priority use cases:

- UCCB 4.0. Use of data resources from campus on XSEDE, or from XSEDE at a campus. Description: Support for data analysis integrated across campus-based and XSEDE-based resources.
- UCCB 5.0. Support for distributed workflows spanning XSEDE and campus-based data, computational, and/or visualization resources

From the outset, the plan was to address these two use cases with the UNICORE client and server software for execution management services, and an adaptation of the Genesis II software for execution management services and for the Global Federated File System. Execution Management Service (EMS) is the name XSEDE uses for job instantiation and management for work activities and workflows (following the Open Grid Forum specifications for EMS). Global Federated File System (GFFS) is the name XSEDE uses for the set of file services based on the University of Virginia Genesis II software tools, implemented as an XSEDE service.

The campus bridging pilot was developed to evaluate and test the software implementations of the EMS and GFFS capabilities XSEDE is developing, and provide feedback to the XSEDE engineering process for the software implementations EMS and GFFS and deployed it on small set of XSEDE resources. A set of friendly users tried this deployment and identified a number of use cases created by users and researchers at institutions with varying levels of experience with XSEDE. To carry out the pilot, XSEDE staff solicited proposals from a broad range of institutions. From the 17 they received, they selected four from participants who had relatively minimal experience with XSEDE resources, and two from participants with extensive experience.

¹ NSF Advisory Committee for Cyberinfrastructure Task Force on Campus Bridging, *Final Report*. March 2011. Available from: https://www.nsf.gov/cise/aci/taskforces/TaskForceReport_CampusBridging.pdf

The institutions and their representatives were:

- Texas A&M, Guy Almes; http://academy.tamu.edu/Guy_T_Almes.html
- City University of NY, Nikolaos Trikoupi; <http://www.csi.cuny.edu/cunyhpc/>
- University of Kansas, Thorbjorn Axelsson; <http://technology.ku.edu/>
- University of Miami, Warner Baringer; <http://ccs.miami.edu>
- Louisiana Tech University/LONI, Thomas Bishop; <http://www.latech.edu/~bishop/>
- Indiana University, Richard Knepper; <http://rt.uits.iu.edu/systems/cbri/>

Participants installed the EMS and/or GFFS software appropriate for their use case(s). The software included the UNICORE 6.5.1 client and server software for the EMS components, and the Genesis II version 2.5 client and container software locally as a way to gain some early user experience lessons with the software, our training materials, and training processes. The XSEDE resource used for the campus bridging pilot included NICS' Kraken and grid services node, PSC's Blacklight, and TACC's Stampede. NICS, TACC, and PSC installed UNICORE 6.5.1 with Genesis II jar file components for access to batch resources, and Genesis II non-root containers for Genesis II basic execution service integration and GFFS services.

C.1. Use Cases

The two use cases to which we planned to apply GFFS software are XSEDE Use Case – Campus Bridging numbers 4 and 6 (UCCB 4.0 and UCCB 6.0). Following are extracts from the use case quality scenarios that describe the use cases and their quality attributes (from <http://hdl.handle.net/2022/14475>):

C.1.1. UCCB 4.0: Use of Campus Data Resources on XSEDE or from XSEDE at a Campus

<i>Description</i>	Support analysis of data integrated across campus-based and XSEDE-based resources
<i>Steps</i>	Basic case (A): Move data from campus resource to XSEDE, and back to campus <ul style="list-style-type: none">• 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 direct remote access, or by transferring file to local storage with• local access. NB: A data resource could be a flat file, tar ball, database to be moved wholesale, and extracted from a database, or a file looked up via a metadata database.• User reads data 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.
<i>Variations (optional)</i>	Variant (B): User has generated data resource(s) on an XSEDE resource and wishes to transfer them to campus. <ul style="list-style-type: none">• 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.

Variant (C): XSEDE-maintained community and reference collections. For efficiency, and to support VOs and XSEDE users, XSEDE maintains copies of community and reference data collections.

- XSEDE determines a list of community and reference data collections it will maintain
- XSEDE creates a GUI for interactive selection of data to be moved from XSEDE to campus resource.
- Access is initiated and completed successfully.

Variant (D): Synchronizing copies of data between campus and XSEDE resource

- User identifies a data set to be maintained, in a synchronized fashion, on one campus resource and one or more XSEDE resources.
- User changes one version of the file, and the other copies are automatically updated.

Variant (E): XSEDE-managed archival storage service

- User provides metadata associated with data to be stored.
- Possible requirement for users: User provides data under a reasonable public-domain license such as Open Data Commons Public Domain Dedication and License, perhaps with a time delay (at latest one year; data become public domain on submitter's death).
 - o NB: NSF policies may obviate this use case

Quality Attributes

- For transient failures, the system should be able to restart transfers and notify users when transfer is complete.
- Stimulus: A properly authenticated user accesses file or directory.
- Environment: An error condition occurs that prevents immediate access.
- Response: System recovers.
- Availability requirement: 98% successful.
- Intuitively usable GUI for interactively initiating and managing file transfer (at least as good as the GUI in Box.net or the Dilbert file transfer tool)
- Stimulus: User installs or requests installation of an access layer interface.
- Environment: User has an XSEDE account and allocation, and a campus account with an associated storage resource. User knows to see the XSEDE portal/web site for help. Necessary software is installed and operational on the XSEDE Level 1 & 2 Service Providers resources.
- Response: Access layer interface is successfully installed.
- Usability: Takes < 1 working day for authenticated user to install the software on the user side.
- Stimulus: User accesses file or directory via an access layer interface.
- Environment: User has an XSEDE account and allocation, and a campus account with an associated storage resource. User knows to see XSEDE portal/web site for help.. User has basic understanding of files and directories. Needed software is installed and operational on the XSEDE Level 1 & 2 Service Providers resources.
- Response: User is able to access file or directory via an access layer interface.
- Usability: A person can copy a file <=15 minutes without support or documentation the first time, and <5 minutes for all subsequent copies.

- Transfer efficiency combined with failures / restarts provides efficiency at least as good as 50% of peak theoretically possible given throughput of optimal network path and storage systems.
- Stimulus: User accesses files,
- Environment: Total size of all files must be >1 GB and average file size > 1 MB. Disk performance on both ends of the copy must have adequate performance specifications. Files accessed within continental US. Achievable performance is measured on an idle network with storage systems on each end.
- Performance: 50% of end-to-end theoretical peak throughput of optimal network path and storage system performance.

C.1.2. UCCB 6.0 Shared Use of Computational Facilities Mediated or Facilitated by XSEDE

<i>Description</i>	XSEDE can provide tools and mediate relationships that enable the US to make better use of its aggregate cyberinfrastructure resources and accelerate research.
<i>Steps</i>	<p>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" (subject to access control) to each other, and this group manages the internal economics of the exchanges.</p> <ul style="list-style-type: none"> • Participants create virtual clusters, virtual high-throughput computing facilities (e.g. condor flocks), virtual clouds, and/or other sorts of virtual resources based on campus compute resources at one or more campuses. • Participants install on their resources the "capability kit" described above that implements InCommon/SAML-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. • NB: In this case, 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
<i>Variations (optional)</i>	<p>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.</p> <ul style="list-style-type: none"> • An organization (virtual or otherwise) operates a Shared Virtual Compute Facility (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 the resource contributor's later ability to obtain in-kind cycles via XSEDE. • 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 sorts of virtual resources.
- XSEDE creates and distributes a “capability kit” for implementing InCommon/SAML-based authentication in ways that maintain the basic functionality, look, and feel of “XSEDE-like” authentication. Authorization and accounting are done with reference to XSEDEDB.
- SVCFs have the “capability kit” installed and in operation.
- XSEDE provides security notification responsibilities in the event of a security breach related to accounts or services that use campus-based authentication mechanisms
- XSEDE has the ability to manage exchange rates between campus-contributed resources and resources campuses might use on XSEDE, and the ability for XSEDE to provide cycles per a Service Level Agreement back to the contributors.
- Integrated ticket management – expanded to include local trouble ticket system of campuses that are providing resources.

Quality Attributes

Variant (A): Shared Virtual Compute Facility (SVCF)

- Ability to set up a Private Gated VO facility in no more than one calendar day.
- Ability to install authentication / authorization capability kit in no more than 2 days (one day to do the work; one day for propagation of attributes.)

Variant (B): Organization (virtual or otherwise) becomes a Level 3 Service Provider

- Ability to install authentication / authorization capability kit in no more than 2 days (one day to do the work, one day for propagation of attributes.)

Both:

- Tickets passed to local trouble ticket system within 1 business day of submission to XSEDE help system

The Table below maps the participating institutions to the use case and use case variants they tested:

Institution	Use Cases Targeted	Description
Texas A&M	UCCB 4	Data read/write performance across multiple sites
City University of New York	UCCB 4,6	Sharing data at CUNY to XSEDE resources, submitting jobs to CUNY system from UNICORE client and Genesis II
University of Kansas	UCCB 4,6	Transfer of polar data files between institutions, job submission on Blacklight through Genesis II
University of Miami	UCCB 6	Submitting jobs to Campus system via Genesis II EMS
Louisiana Tech University / LONI	UCCB 4,6	Data movement to Kraken, submitting jobs to Kraken via Unicore 6 EMS
Indiana University	UCCB 4	Data movement between campus and datacenter, access to Lustre resources at IU via GFFS

C.2. Software Environment

The following describes the software environment employed for the campus bridging pilot that was installed on the campus bridging pilot resources described above.

C.2.1. Overview of Genesis II Software Components

The Genesis II Client software permits users to manage their data and the permissions of the folders on the data, as well as transfer data among directories in the GFFS. From the Genesis II website: "Through GFFS, user applications running on campus and research group machines can directly access (CRUD) files and other resources at NSF-funded service provider (SP) sites and collaborator sites as if they were located at the center. Existing applications, whether they are statically linked binaries, dynamically linked binaries, or scripts (shell, PERL, Python), can access resources anywhere in the GFFS without modification (subject to access control)."²

From the user's standpoint the Genesis II software provides two different facilities for working with GFFS. The Genesis II Client software allows users to view, read, and modify files in the GFFS and manage permissions in the GFFS. The Genesis II Container software allows users or service providers to export directories to the GFFS, where they become part of the Global Filesystem tree. The Global Filesystem tree requires a root server, which in this instance is managed by XSEDE Operations, and provides the basis for containers to connect to create the Globally Federated Filesystem. The root container provides the root of the filesystem, and each of the containers exported connects to create the federated filesystem.

C.2.2. Overview of UNICORE Software Components

UNICORE offers a ready-to-run Grid system including client and server software. UNICORE makes distributed computing and data resources available in a seamless and secure way. UNICORE is made up of a client layer, a service layer, and a system layer. The UNICORE command-line client (UCC) is a versatile command-line tool that allows users to access all features of the UNICORE service layer in a shell or scripting environment. Users can run jobs, monitor their status, and retrieve generated output, in single job mode or in a powerful and flexible batch mode for multiple jobs. The Eclipse-based UNICORE Rich Client (URC) offers users the full set of functionalities like the UCC in a graphical representation. The service layer comprises all services and components of the UNICORE Service-Oriented Architecture (SOA) based on WS-RF 1.2, SOAP, and WS-I standards, including a Gateway, Registry, and the UNICORE/X, which is a WSRF-compliant web service that provides the interface to storage resources, file transfer services, and job submission and management services. The services layer also offers a standardized set of interfaces based on OGSA-* standards is available in UNICORE 6 in addition to the UAS. Currently implemented standards are OGSA-BES and HPC-P, used to create, monitor, and control jobs. The system layer comprises the Target System Interface (TSI) component, which is the interface between UNICORE and the individual resource management/batch system and operating system of the Grid resource(s).

² <http://genesis2.virginia.edu>

C.3. Campus Bridging Pilot Results

C.3.1. System Administrator Results

The following issues were reported by the NICS, PSC, and TACC system administrators related to the campus bridging pilot regarding UNICORE and Genesis II. Some of these issues have already been addressed in the Genesis II software version 2.7.

- State files end up in `~/genesisII-2.0` directory unless `GENII_USER_DIR` environment variable gets set. This is not prominently documented in any of the G2 installation documents that were distributed. It is listed in the omnibus reference manual but it should not be expected that an administrator should have to read the omnibus reference cover to cover to install and configure this software. The important configuration considerations, such as `GENII_USER_DIR` environment variable, should be part of the installation and configuration documentation and a prominent part of the installation (a question on the installer, for example). The side effect of this is if one is not careful with installation then the non-root container state directory will be located in a non-desirable location (non-root container process owner's home directory which could be NFS, for example) and will be used as a G2 name space endpoint to store files.
- The non-root container runs as an unprivileged user and files that are already owned by a user have to have POSIX ACLs set on them to allow the user to have access to files he already owns in the file system. Having to have POSIX ACLs on the file systems to link into the name space seems problematic. Sites may not want to have this file system overhead, or the administrative overhead to set all the ACLs for an existing file system (home directory, scratch space or temp space). We suggest that a container process run as root and switches to the user be considered. There are cases for using the non-privileged non-root container the way it is currently (for example, sharing files with a user that does not have a local account and there are cases for a privileged non-root container (for example, publishing an existing scratch file system with 1000 users and millions of files that already exist).
- The non-root container was linked twice accidentally and there was no warning or error message to indicate that this was already mounted or that this was a problem.
- The omnibus reference documentation in the "linking a UNICORE BES" section does not describe a key item that whatever CA issues the UNICORE host/service certificate that is used for the SSL must have its public CA cert in the trusted `.pfx` file. Without this CA in the truststore, trying to connect from the Genesis II client will never work and give an error. (I believe Chris has already started working on updating this.)
- Several of the configuration tasks that need to be performed are described in the omnibus reference manual in very generic terms. There is no context for understanding what many of the references to files and URLs should be. The omnibus reference document should contain more examples of real commands that could be performed (in the context of some example configuration that is being described). There are too many command options requiring optional items that are never described completely enough for someone who is not already familiar with Genesis II.

- Problem with using 64 bit java on CentOS 6.3. Getting error " Could not reserve enough space for object heap" I tried using the -Xmx1024m option, but certain apps would still fail. Loading 32 bit java version seems to fix the issue.
- When doing a “grid run –async-name={filename}” to launch a job using BES the '{filename}' file is not readable. What is this file? What does it do? This is not documented what is in this file and how it can be accessed for status.
 - Example:
 - victor@acai:/usr/local/GenesisII> ./grid run --async-name=/home/victorh/jobs/helloworld3 --jsdl=local:/nics/d/home/victor/GenesisII/jobs/test.jsdl /bes-containers/nics-kraken
 - victor@acai:/usr/local/GenesisII> ./grid cp /home/victorh/jobs/helloworld3 local:/tmp/junk
 - Source path grid:/home/victorh/jobs/helloworld3 is not a file!

C.3.2. Pilot Participant Results

C.3.2.1. All sites:

Clients referred to documentation from the UVA site as well as documentation developed by XSEDE Operations and Campus bridging in order to install GFFS clients on their campuses and create file system exports to be attached to the XSEDE GFFS via GFFS containers. Andrew Grimshaw’s team provided an online training session early in the pilot to familiarize the pilot site leaders with GFFS installation and concepts.

Pilot sites installed client and container software and were able to export to the XSEDE Beta Root container. Pilot sites were also able to modify permissions within their exports to allow Grimshaw’s team and other sites to share data.

C.3.2.2. Texas A&M

Guy Almes’ team at Texas A&M focused on establishing performance information on cross-site data access, to establish the performance parameters for accessing local data from XSEDE resources and vice-versa, as well as data access across multiple campus sites.

C.3.2.3. Louisiana Tech University

Tom Bishop’s team focused on submitting jobs on Kraken, taking advantage of job submission through the Unicore 6 environment on Kraken and GFFS in order to access data. Tom made considerable use of UVA support resources provided by Vana Venkataswamy and eventually succeeded in executing a workflow that pulled inputs from Bishop’s lab at Louisiana Tech via GFFS, ran a 240-core mpi-based job (NAMD) on Kraken that wrote to local scratch, and then pushed results back to Bishop’s lab at Tech.

C.3.2.4. University of Kansas

Thorbjorn Axelsson at the University of Kansas initially had planned to work with a local researcher to test accessing data located at KU from the Blacklight system at PSC. This researcher left before the root container service was ready. KU worked with IU to provide access to data stored on IU's Lustre storage system to researchers at KU, via a gateway system that could mount Lustre and export directories into GFFS.

C.3.2.5. University of Miami

University of Miami completed installation of clients and container software on their local site systems, despite experiencing some delays due to local hardware installation. The Miami team volunteered to join a resource to the GFFS via the Execution Management System to share jobs between XSEDE and Miami resources, but XSEDE security policies kept this from being implemented. These security issues are superseded in the production version of GFFS, which will be installed on XSEDE resources this fall.

C.3.2.6. City University of New York

Nikolaos Trikoupis and the City University of New York group also completed client and container installation and joined resources to the XSEDE GFFS root. Similar to the University of Miami, CUNY also had timing issues during the pilot resulting from local hardware installation. CUNY also volunteered to connect their local compute resources in order to share jobs from XSEDE. XSEDE Operations staff worked with Nikos and the CUNY staff to allow jobs to be submitted on CUNY's cluster via BES.

C.3.2.7. Indiana University

Indiana University completed client and container installation, and worked with KU to share data on Lustre resources to researchers at KU. IU also used its GFFS export as repository for project documents.

D. 2013 Campus Bridging Pilot Evaluation Report

September 2013

Prepared for: Craig Stewart, Campus Bridging Level 3 Manager
Scott Lathrop, TEOS Level 2 Manager
Richard Knepper
Barbara Hallock
Jim Ferguson
Andrew Grimshaw

Prepared by: Dr. Lizanne DeStefano, Director
Lorna Rivera, Research Specialist
I-STEM Education Initiative
University of Illinois at Urbana-Champaign



ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

D.1. Background and Context

In July 2011, XSEDE received an NSF award from the Office of Cyberinfrastructure to enact their proposed plans for addressing the Cyberinfrastructure Framework for the 21st-century Science and Engineering (CF21) and the nation's need for integrated cyberinfrastructure and services. Led by the National Center for Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign (UIUC), XSEDE is a partnership among 17 institutions working to create and maintain open access to their collection of integrated advanced digital resources and services. The goal of XSEDE is to enable scientific discovery through enhanced researcher productivity. The TEOS branch of XSEDE aims to address the critical need to advance computational science and engineering through recruiting, preparing, and supporting a large and diverse scientific, academic, and industrial workforce equipped to utilize the XSEDE cyberinfrastructure ecosystem. TEOS partners include Illinois/NCSA, University of Tennessee/NICS, Carnegie Mellon University and the University of Pittsburgh/PSC, University of Texas/TACC, University of California San Diego/SDSC, University of California Berkeley, and Rice University.

D.2. Data Sources

D.2.1. Interview

Key participants from each site in the Campus Bridging (CB) Pilot were asked to participate in interviews. Interviewees represented most of the participating institutions, namely City University of New York (CUNY), University of Miami (UM), and Kansas University (KU). Interviews were approximately 30–40 minutes, and participants were asked about their experiences with the CB pilot thus far. Interview questions focused on local XSEDE resource utilization, goals for GFFS implementation, changes to proposed implementation plans, GFFS functionality benefits, training, FUSE driver experience, and thoughts on the future of the GFFS implementation in XSEDE. See Appendix A for the interview protocol.

D.2.2. Online Survey

Due to time constrictions, representatives from Texas A&M University (TAMU) and Louisiana Tech University (LATEch) were asked to complete an online survey on their experience in the CB pilot thus far. The survey included all items in the interview protocol used with the aforementioned pilot sites. It sought information on local XSEDE resource utilization, goals for implementation, changes to proposed implementation plans, functionality benefits, training, support, FUSE driver experience, and thoughts on the future of the Campus Bridging technology implementation in XSEDE. See Appendix B for the Campus Bridging Pilot online survey.

D.3. Findings

During data analysis the following findings emerged. Findings are grouped by theme.

D.3.1. Strengths of Campus Bridging Pilot

Overall, pilot participants were impressed with the level of support offered by the campus bridging team in executing their projects. Some of the support identified included regular office hours with University of Virginia staff, an email address for help, and an online training tutorial.

Pilot participants were highly satisfied with the level of support offered by the campus bridging team. Many regularly took advantage of the office hours held by University of Virginia staff and cited this aspect of support most often. Participants also found the online tutorial helpful for following along during implementation. The email help address was used for less pressing issues such as account setup. Some comments included:

- *“The staff at Virginia was always very expert and very helpful.”*
- *“Vana... she had hours that were set aside but she actually replied to every problem that I had immediately and the support was excellent.”*
- *“...the video, the step by step through... I guess it was a recorded tutorial... I thought that that was very useful... I followed the video when I did it...”*
- *“So I went to a workshop during the XSEDE conference and there was one training session, after that we did online... I guess a remote training... Yup it was good... I think that it worked fine in remote”*
- *“I did send an email to a help address before the office hours were set up. I don’t recall but I think that it was part of the genesis team... Yeah but I emailed him, Andrew, and he suggested that I send an email to some help address email... Yes absolutely [helpful]. Help to set up my account I think...”*

D.3.2. Benefits of Campus Bridging Technology

Participants considered the ability to set permissions on directories and files within the GFFS a highly beneficial aspect of the GFFS technology. When applicable, utilization of the Filesystem in Userspace (FUSE) driver was also deemed highly beneficial.

Pilot participants were most satisfied with the level of access control available through GFFS. They also mentioned browsing data, exporting files, and sharing data. All were seen as “essential” and “non-separable” functionalities for a file system interface; however, they were not mentioned with as much enthusiasm. Some comments included:

- *“I view 1 [browsing data], 2 [exporting files], 3 [sharing data between sites], and 4 [setting permissions on directories] as essential and non-separable functionalities that a file system interface must provide.”*

- *“...which was very nice, create files and have one user create files and then having that user the ability to grant access to others and share them from the desktop and from one of the containers. I think that is the potential for share the access control and the data kind of have it...”*
- *“...and the ...functionality where you are able to expose your data easily to the... (sic) without needing somebody else to intervene and authorize, I think that is also important...”*

The FUSE driver was a popular and desirable feature of the pilot. Participants were highly satisfied with its functionality and disappointed in the lack of an equivalent for Mac OS and Linux. Respondents preferred using the FUSE driver over entering the GUI. Some comments included:

- *“What we did like was the FUSE functionality...the FUSE functionality ... gives you manuscripts and tools. You can use the file manager and your work is with files from GFFS and I think that that is very useful functionality one wants to have as an XSEDE user. We really like that. Of course this is going to be useful for interacting with small files”*
- *“...disappointed in the lack of the FUSE equivalent (for Mac and Linux)... it is a popular one and FUSE file system was also a very popular format to making it easier to integrate without having to go into the GUI”*
- *“I think one of the problems in the end was that there were no Mac clients.”*

D.3.3. User-Friendliness

Pilot participants recommended ways to increase the user-friendliness of the software and pilot program. Recommendations centered on authentication, training documentation, the FUSE driver, Graphical User Interface (GUI), and possibly offering a mobile version of the GFFS client.

Although respondents reported the FUSE driver performed very well, they still had questions about authentication. Nonetheless, participants preferred working with the FUSE driver over the GUI. Participants also expressed a need for more concise and clear training documentation. They noted that several versions of extensive documentation were provided, which led to confusion on application to their situation. Participants are also concerned with the amount of time required to utilize the GFFS client. In the future, offering a mobile version of the GFFS client may be a way of increasing convenience, decreasing time commitments, and ultimately increasing user-friendliness. Some comments included:

- *“Yes. It worked... some questions about authentication had to be clarified...”*
- *“It performed very well. I noticed two key problems for GFFS as a wide-area file system. First, the user must deal with two different userid spaces -- the one on the local system and the one in the GFFS world. This makes all aspects of posix permissions, users, groups, etc., less useful, and can cause confusion for the user.”*
- *“It (training documentation) was too detailed and too extensive. It was hard to drill down to the parts that really mattered to me. There was also significant technical jargon/abbreviations that were never defined. In short the distinction in documentation*

between documents for experts and novice or maybe users, developers, and administrators was not clear to me. To be fair, as a pilot, my role is all of the above.”

- *“It was a lot of documentation ... In the sense that there were several versions of the documentation available. There was the documentation the wiki, on the XSEDE portal, there was documentation from the University of Virginia that appears for some parts we would need to take a look into. So although it was a lot of documentation it ended up being a little bit confusing on what is the correct information that we ought to be looking for, for what we are trying to do ... and yeah that was not a big problem but it kind of got messy at some point.”*
- *“I see a lot of benefits in providing a mobile version because more and more people rely on their smart phones these days to keep more and more of their work. So if they were able to have the GFFS client on their smart phone I know that they would be very very happy. We got people checking their jobs every 2-3-5 minutes. So if it was a tool that would let them do that from their cell phone I think that that would be very good and make them very happy... easier workload from easy manager script to JSBL that is what I would like to see”*

D.3.4. Suggestions for Improvement

Additional suggestions for improvement included increasing the performance capability of the GFFS software and offering explicit project timelines in order for sites to schedule implementation.

When asked about their experience with the FUSE driver, a few participants indicated it lacked “very high performance” and were unaware of plans to increase. They felt a performance increase was needed in order to improve the functionality. Additional suggestions centered on the program’s timelines. Although participants indicated “a greater respect for the difficulties” in conducting such a task, many initially involved themselves in the pilot program with the impression that more resources would be available for use in a timely manner. Lessons learned from this pilot should be shared with similar future programs to ensure realistic expectations are established and maintained. Some comments included:

- *“...the current GFFS implementation is not very high performance. I am not aware of any design decisions that would prevent its performance from being radically improved. But the use of XSEDE often involves very large files, and moving them in a low-friction way requires high performance.”*
- *“Data transfers are a real challenge and issue and I hoped to learn more about it. If GFFS could facilitate data transfers...”*
- *“...it was obvious when we got the product and the software was not really there yet it is still under development... when we were writing the proposal we somehow suspected that it would already be available in the XSEDE resources, which of course not the case but because those XSEDE resources are also a part of the pilot and one of the pilot goals is to see how basic technologies can be put into production or it can (...) as well. So I think that we were thinking a little bit ahead of what is possible, so we had to adjust our expectations and our goals.”*
- *“Well no we had ... other research groups that were interested but the first one had a GRA that was going to work on it but he managed to graduate before we could assign him to it, so it was*

operational because of the delays and [another] decided half way through he say that it wasn't meaningful for him to participate."

— *"I have a greater respect for the difficulties in accomplishing a wide-area file system."*

D.3.5. Future Applications

In the future, pilot sites hope to see the Campus Bridging technology applied to the integration of local pilot systems with external resources to create a seamless environment for researchers. Benefits identified include flexibly unifying access between systems across multi-site institutions as well as local and XSEDE systems.

Many sites initially became interested with the CB pilot program since their own local systems and users were facing similar challenges as XSEDE users; namely, sharing data between sites, setting permissions, and unifying access among local users. Sites shared their visions for unifying access locally and between XSEDE systems and the benefits to their local users. Some comments include:

— *"I would like to see if you look just at GFFS as a standpoint alone and separate from XSEDE I think that it can have a place in that fashion [service provider for a campus] also. So there will have to be a lot more documentation about how to integrate it into your own environment. Whereas when it is launched from the XSEDE environment it is much more straight forward with the authentication and all of that. The certificates and all of the necessary stuff is set up for you and the installation."*

— *"I was very pleased to be able to participate and I think that it is a good concept and it definitely addresses without question some of the biggest problems that we have been facing and others also. In terms of sharing data, controlling access to data and unifying (sic) of systems. So in some fashions of all it address all of these issues, you know it has its drawbacks but it is a good idea."*

— *"...we wanted to see how GFFS had approached the problem and implemented it and see how it worked for us ...not only with XSEDE resources but ... using it as something to unify access even among local users here."*

— *"...Integrate our systems in a provider picture of resources that our researchers might be using in the future. I/we could envision the researcher that has access to both of our systems and XSEDE resources to work more in sync in terms of job submission and data sharing and data transfer and data (...) transferring." "... our goals were at first to have a researcher as a user of our systems try the software and the same researcher making (...) on a XSEDE resource."*

D.4. Recommendations

1. **Provide a FUSE driver equivalent for Mac OS and Linux.** Participants were highly satisfied with the FUSE driver functionality and requested an equivalent for Mac OS and Linux.
2. **Increase user-friendliness by minimizing repetitive authentication.** Pilot sites were somewhat confused by the multiple user identification spaces. A recommendation was made concerning the adoption of InCommon® as a solution for repetitive authentication.
3. **Clearly label training documentation by identifying its purpose and minimizing technical jargon.** Although extensive training documentation was available on the wiki, participants were unable to determine what was relevant to their situation. Organize and refine online training documentation by removing previous versions of documents, creating technical application categories, and minimizing the use of abbreviations and technical jargon.
4. **Provide realistic timelines for pilot site involvement.** Due to the innovative nature of the project, pilots understood delays in operation, but were not always able to cope with the implications. In future releases or tests of campus bridging software, indicate the potential for significant delays in project plans due to internal processes so that pilots can appropriately prepare.

D.5. Appendix A. Campus Bridging Pilot Interview: Protocol

1. Demographics
 - a. Institution, department, job title/academic status, role in GFFS pilot
2. What other XSEDE resources are being utilized on your campus?

XSEDE Resources		
High Performance Computing	Software	Training
High Throughput Computing	User Guides	Education Programs
Visualization	Metascheduling	Campus Champions Program
Storage	SU Converter	
Networking	Student Programs	

3. What were your goals for implementing the GFFS pilot on your campus? Have you been able to meet those goals?
4. How have your originally proposed plans changed since you became a GFFS pilot site?
5. What aspects of the GFFS did you find to be the most beneficial?
 - a. What were the strengths of the GFFS functionalities:
 - Browse data in GFFS with client
 - Export file system to GFFS with container
 - Share data between sites (Pilot-Pilot, Pilot-SP)
 - Set permissions on directories and files within the GFFS
 - Submit jobs to UNICORE EMS via GFFS (on SP or campus resource)
6. Do you believe the training your site received was sufficient?
 - Was the documentation provided useful?
 - When your team faced challenges, what sources did you turn to for help? Were they adequate and useful?
7. Did you use the FUSE (Filesystem in Userspace) driver? How was your experience?
8. What would you like to see for the future of the GFFS implementation in XSEDE?

D.6. Appendix B. Campus Bridging Pilot Interview: Online Survey

1. To the best of your knowledge, indicate what XSEDE resources are being used on your campus:
 - High-throughput computing
 - High-performance computing
 - Visualization
 - Storage
 - Networking
 - Software
 - User guides
 - Metascheduling
 - SU Converter
 - Student programs
 - Training
 - Education programs
 - Campus Champions Program
 - Other (please specify)
2. Have you been able to meet your originally proposed goals for the GFFS pilot?
3. How have your originally proposed plans changed since you became a GFFS pilot site?
4. What aspects of the GFFS did you find to be the most beneficial
5. What were the strengths of the GFFS functionalities:
 - a. Browse data in GFFS with client
 - b. Export file system to GFFS with container
 - c. Share data between sites (Pilot-Pilot, Pilot-SP)
 - d. Set permissions on directories and files within the GFFS
 - e. Submit jobs to UNICORE EMS via GFFS (on SP or campus resource)
6. Was the training documentation provided useful? Please expand.
7. When your team faced challenges, what sources did you turn to for help? Were they adequate and useful?
8. Did you use the FUSE (Filesystem in Userspace) driver? How was your experience?
9. What would you like to see for the future of the GFFS implementation in XSEDE?

