XSEDE Campus Bridging Pilot Case Study

Barbara Hallock[i]	Richard Knepper[i]	James Ferguson[ii]	Craig Stewart[i]
Indiana University	Indiana University	National Institute for Computational Sciences	Indiana University
bahalloc@iu.edu	rknepper@iu.edu	jwf@utk.edu	stewart@iu.edu

ABSTRACT

The major goals of the XSEDE Campus Bridging pilot were to simplify the transition between resources local to the researcher and those at the national scale, as well as those resources intermediary to them; to put in place software and other resources that facilitate diverse researcher workflows; and to begin resolving programming and usability issues with the software selected for these purposes. In this paper, we situate the pilot within the domain of existing research cyberinfrastructure (and in the context of campus bridging) and examine the process by which the pilot program was completed and evaluated. We then present a status update for the selected software packages and explore further advancements to be made in this realm.

Categories and Subject Descriptors

D.2 SOFTWARE ENGINEERING

General Terms

Documentation, Performance, Design, Standardization.

Keywords

Campus Bridging, Genesis II, UNICORE, software, usability

1. INTRODUCTION

Changes in the research and information technology fields have provided an ever-growing list of resources to achieve research goals; however, these resources have not traditionally been consistently configured or easy to use without a degree of in-depth technical support that would not have traditionally been available to researchers. As the variety of machines available at the national tier has increased, so have the cognitive costs for users of new (to them) systems. Furthermore, as new resources have been stood up, comparatively little attention has been paid to creating useful and usable documentation for the actual intended user base.

The Campus Bridging pilot began as an attempt to increase eXtreme Science and Engineering Discovery Environment (XSEDE) services that both streamlined and simplified the process of transitioning between machines for the researcher. The pilot program began in January of 2012 and continued through fall of 2013. Six campuses were selected as pilot sites, a mix of

mostly sites with no XSEDE experience and a couple with relatively extensive experience. UNICORE 6 and Genesis II were the applications on which the pilot program focused, with the explicit intention to improve the experience of sharing data and creating workflows on single or multiple XSEDE resources.

A set of sample evaluation questions were given to pilot participants before they began using the software so that they would have an idea of what to focus on while installing, configuring, and utilizing the software, collectively referred to as "squishy" metrics. At the close of the pilot, XSEDE evaluation staff and the pilot participants completed in-person interviews and online surveys. We present here an in-depth examination of the pilot, including insights for future XSEDE pilot programs and an overview of improvements to be made to the pilot software.

2. CAMPUS BRIDGING

The National Science Foundation Advisory Committee for Cyberinfrastructure (CI) task force convened in 2011 to make recommendations for a greater focus on campus bridging within the context of national CI [1]. From that document came the formal 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].

To further understand the idea of campus bridging, consider the following scenario: a researcher needs High-Performance Computing (HPC) resources to complete an experiment. He or she is faced, more often than not, with a marked lack of sufficient CI, and it can be difficult to find the resources. However, even when the researcher is able to acquire the resources, there is little to no guidance available as to how the researcher should go about conducting his or her experiment. Put metaphorically, the researcher is standing at the edge of a cliff and must get across. Campus bridging attempts to provide a "bridge" for the researcher over some common challenges in the process of achieving discovery (the metaphorical other side of the chasm). In more practical terms, the idea is that the scientist should be able to use the wealth of resources available locally all the way up through nationally in a fashion that feels to them as seamless as if they had simply plugged in a peripheral device [2].

Some of the common challenges the XSEDE Campus Bridging staff are working to overcome: a lack of documentation aimed at new users with low technical knowledge; a lack of standardized user interface across machines even within XSEDE; and the lack

of a simple method for secure data sharing between users and groups of users.

3. PILOT PROGRAM

The Campus Bridging pilot program began in late 2011 with a request for proposals sent out to a number of "friendly institutions" throughout the United States, requesting that Primary Investigators (PIs) at those universities with needs that aligned to the challenges listed above submit projects on which the software (described below) would be useful. This provided XSEDE staff with a number of knowledgeable users who had real-world data needs that were being held up by the very challenges the pilot aimed to reduce [2].

3.1 Goals of the Pilot

A set of use cases has been developed by XSEDE to describe particular areas of interest within the environment. The XSEDE Campus Bridging pilot attempted mainly to address two high-priority use cases:

"UCCB 4.0. Use of data resources from campus on XSEDE, or from XSEDE at a campus. Support for data analysis integrated across campus-based and XSEDE-based resources….

UCCB 6.0. Shared use of computational facilities mediated or facilitated by XSEDE" [4].

More specifically, the goals of the pilot were to: provide a unified user interface for scientists regardless of what resource they were using and regardless of personal operating system preference; provide a unified workflow that behaved the same regardless of if the researcher required jobs on one resource or many; to reduce the barriers for entry to XSEDE resources; to determine the feasibility of the proposed software; and to locate and fix multiple kinds of "bugs" existing within one of the two software packages.

3.2 Pilot Sites

The pilot plan allowed for the addressing of these use cases via a mix of two distinct software packages, described in detail later in this paper – Unicore 6 and Genesis II. A grand total of 17 sites submitted proposals; those selected, and their representatives are: Texas A&M University (TAMU), Guy Almes; City University of New York (CUNY), Nikolaos Trikoupis; University of Kansas (UK), Thorbjorn Axelsson; University of Miami (UM), Warner Baringer; Louisiana Tech University/LONI (LONI), Tom Bishop; and Indiana University (IU), Richard Knepper. This included four institutions with relatively little experience with XSEDE and a further two with a relatively high degree of expertise. This breakdown allowed the conditions of the pilot to accurately simulate the range of user needs and behaviors across the spectrum.

The universities whose projects aligned with UCCB 4.0 were Texas A&M, CUNY, UK, LONI, and IU; those whose projects aligned with UCCB 6.0 were CUNY, UK, UM, and LONI. Additionally, LONI did some work with the pilot software that spoke to UCCB 5.0, Support for distributed workflows spanning XSEDE and campus-based data, computational, and/or visualization resources. Staff at the University of Virginia (UVA) provided development, support, and documentation for the Genesis II product.

3.3 Pilot Software

3.3.1 UNICORE

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, 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].

3.3.2 Genesis II

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).'[3]

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.

Another, albeit limited, feature of Genesis II is the Filesystem in USErspace (FUSE) driver. Its availability is limited to Windows at present; with this functionality enabled, the user is able to interact with remote resources as if they were a local part of the Windows file system.

3.4 XSEDE Services Involved with the Pilot

Three XSEDE resources were involved with the pilot: the National Institute for Computational Sciences (NICS)' Kraken and grid services node, Pittsburgh Supercomputing Center (PSC)'s Blacklight, and Texas Advanced Computing Center (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. XSEDE staff from Indiana University and NICS coordinated meetings and worked to facilitate the resolution of issues during the pilot, and

other XSEDE staff from UIUC assisted with the preparation, collection, and analysis of evaluation materials at the end of the pilot period.

4. EVALUATING THE PILOT PROGRAM

4.1 Evaluation Methods and Metrics

Pilot participants were asked at the beginning of the program to evaluate the software particularly for a set of "squishy" metrics focused on user experience and other qualitative factors of using the software. A sample questionnaire was provided on the XSEDE Campus Bridging Wiki [5] to enable the pilot participants to keep the sorts of questions evaluators would be looking for answers to in their minds as they interacted with the software.

During the evaluation period at the end of the pilot, the evaluation team gathered data via two methods. The first method consisted of interviews with key participants at CUNY, UM, and KU, each lasting approximately 30-40 minutes, focusing on local utilization of XSEDE resources, goals for the pilot, GFFS functionality benefits, training, and thoughts on the future development of the software product. The second method consisted of an online survey distributed to TAMU and LONI due to time restrictions. The survey asked the same questions as the interview did. For a complete description of interview protocol and a list of the survey questions, see [6].

Pilot participants were asked at the beginning of the program to evaluate the software particularly for a set of "squishy" metrics focused on user experience and other qualitative factors of using the software.

4.2 Challenges of the Pilot Program

4.2.1 Working in a distributed organization

Beginning the pilot, it was anticipated that a major challenge to the efficiency of the process would be the distributed nature of the XSEDE organization and the pilot sites. It can be difficult to keep up to date on what another group is doing when one is unable to simply stop by a colleague's office and ask for updates in person. As anticipated, this presented a major challenge for pilot staff in facilitating the work that needed to be done in order to successfully provide services to the pilot site. There were a number of communication channels opened up to mitigate this challenge, described in section 4.3.1.

4.2.2 Pioneering the XSEDE operational readiness review process

At the onset of the pilot program, XSEDE was still a relatively new organization, and there was not a clear policy in place for ensuring that new software to be distributed to the entire network was secure enough that it was not a risk. This lack of a clearly defined policy was an unanticipated challenge, but an important one to overcome nonetheless, as without a carefully considered set of standards for what is and is not permissible on XSEDE resources, there could be any number of negative effects not only on the network itself but on the scientists utilizing it for their research. The development of these policies and procedures for reasonably low-risk adoption and implementation was critical to the success of XSEDE as an organization, but in the end, this posed a major challenge to the timely completion of the pilot.

4.2.3 Timeline clarity

Another challenge to the timely completion of the project came from the open-ended nature of the pilot design. A number of the pilot sites, while understanding of the fact that the pilot was intended to be innovative and would thus require work with "unfinished" software, had difficulties accomplishing their objectives because of the length of time it took to produce a usable version of the software [6]. The lack of communication during the RFP that users could expect to see significant delays due to the development process was problematic, and it was recommended that any future pilot projects such as this one should include clear communication about any potential challenges or delays that may come about due to the development and adoption process.

4.2.4 Lack of experienced users and administrators

Given the relative newness of the Genesis II software, it was anticipated at the inception of the pilot that there would be relatively few or almost no experienced users or system administrators prepared to begin implementing the software. While UNICORE has been around for a while, Genesis II had previously seen only limited use in the UVA Cross-Campus Grid (XCG) environment. This meant that the pilot group would need to be provided with materials to help them navigate the installation and use of the software and an experienced support person to help them overcome challenges to successful installation and utilization.

4.2.5 Lack of consistent documentation

The documentation available were simultaneously too detailed and not detailed enough; versions were inconsistent, each covering different parts of the install process, and it was not unusual for the pilot sites to find out upon encountering difficulty with Genesis II that it was because they had been following the wrong set of instructions. High-level terminology and acronyms were frequently presented without any sort of definition for the user, which was frequently cited as a source of frustration. At the same time, many of the different sources of documentation lacked critical steps in the setup process, which in turn caused more issues with the install process. It was also unclear to what level of user (i.e. novice, intermediate, expert) each particular piece of documentation was aimed at, and one pilot participant cited a critical step in the install process that was described only in the omnibus document in vague, unhelpful terms.

4.2.6 Software and authentication issues

One of the challenges inherent in a pilot program that involves a software package still in an active beta development process is that delays are to be expected on a fairly regular basis. Genesis II was no exception; there were a number of issues with the software that needed to be corrected. Error messages did not provide any information about what caused the error, which meant that technical support was required to solve the user's problem, which led to some delays. Users also cited an initial difficulty in grasping the difference between the Genesis II namespace and the local machine namespace, which made certain features of the software less useful [6].

4.3 Successes of the Pilot Program

4.3.1 Communication among sites

In order to minimize the negative effects of the group's geographical disparity, a number of regular meetings were scheduled: an operations teleconference for the software developers, the organizers, and the pilot sites with a high degree of XSEDE expertise; and a more general pilot teleconference, which did not include developers but did include organizers and the entire group of pilot sites; dedicated discussion of the pilot updates in regularly scheduled meetings with the XSEDE Training, Education, and Outreach Services (TEOS) staff, which also included staff members from XSEDE Operations and

Software Development and Integration (SD&I). Daily communication was accomplished asynchronously via e-mail list and a wiki and forum set up on the XSEDE Web site. Finally, when conference travel permitted, members of the pilot group had face-to-face meetings to work through issues.

4.3.2 XSEDE Operational Readiness Review
While the lack of a clearly defined Operational Readiness Review (ORR) process was listed with the challenges, this can really be seen as one of the most important successes of the pilot program in terms of future innovation. A thoughtfully constructed set of policies and procedures that allows XSEDE staff to vet applications before deployment across the network has been developed, and future pilot programs will benefit from the clearly defined set of guidelines for a number of reasons, most importantly because it will allow future teams to more appropriately set timeline expectations among participants during the pilot design phase before any proposals are submitted. Finally, researchers using XSEDE resources for experiments may continue to run their jobs with the knowledge that any applications available for use on the XSEDE network are not going to expose their data unless they specifically set permissions for those items to be viewable by others.

4.3.3 Training
Staff at UVA created training materials for the pilot users. There were a number of training sessions, facilitated via teleconference and screen sharing software provided by XSEDE. Users felt these training sessions were mostly useful, though they had room for improvement. Given the novelty of the Genesis II software, it is expected that the attention to a need for training for new users will support an easier transition to adoption across the network. Existing versions of the training material will need to be updated for current versions and edited for clarity and the expected proficiency level of the audience, but they present a strong foundation for future Education, Outreach, and Training (EOT) efforts, and thus, can be considered a largely successful aspect of the pilot project.

4.3.4 Documentation and Support
While initially the documentation available were varied in their level of detail to the point of being almost unusable, the edits made by UVA staff during the process of the pilot cleared up a number of the issues cited by users as problematic. Additionally, we would be remiss to discuss the successes of the pilot program without specifically pointing to the efforts of Vana Venkataswamy at UVA, who is involved in the development of Genesis II and provided much-needed expert support to the pilot sites.

In addition to her efforts in improving the quality of the available documentation, Venkataswamy held regular "office hours" twice a week wherein she was available by telephone, Skype, or e-mail to answer questions. She used the Skype screen-sharing feature to great effect in her troubleshooting and technical support efforts; many pilot users specifically mentioned these office hours as a strong positive of the pilot. Having a knowledgeable user with extensive experience guiding the installation and configuration of Genesis II sped up the install process considerably, and was integral to the success of the pilot.

4.3.5 Implementation of UNICORE 6 and Genesis II
Perhaps the biggest notable success of the campus bridging pilot program is that both UNICORE 6 and Genesis II have both been approved for use across the XSEDE network. As of this writing, UNICORE 6 has been installed and is running on every XSEDE

service provider, and Operations reports that Genesis II should be available soon, and certainly well in advance of XSEDE '14. As Genesis II's client components are OS independent, this will allow users to more easily interact seamlessly with XSEDE resources from a single, unified interface.

4.3.6 Increased exposure for XSEDE
Four of the six sites that participated in the pilot program were relatively inexperienced with XSEDE. One of those institutions is classified as a Minority-Serving Institution (MSI). Another entered the pilot with a use case about the transmission and processing of data from polar ice flights for the Center for Remote Sensing of Ice Sheets (CReSIS), whose work is shaping and informing our scientific understanding of climate change over time. As a direct result of the pilot, XSEDE will be able to facilitate more important scientific discovery than it has previously (though it is difficult to quantify to what degree).

4.3.7 Diversity of site expertise levels
While the different levels of XSEDE expertise presented by the pilot sites was a challenge, it ultimately can also be seen as a strength and a success of the pilot program. The ratio of 2:1 reasonably inexperienced sites to reasonably experienced sites meant that there was a variety of "difficulties," technically speaking, in the project. This also allowed the pilot to accurately model the effects of the software on new user adoption and presented a range of technical needs to illustrate how the software handled different complexities of the problem. Additionally, while the "advanced" sites strained the capabilities of the software somewhat, they illustrated where improvements needed to be made and thus presented a valuable insight to the evaluation of the software products.

5. FURTHER WORK TO BE DONE
As mentioned earlier in this paper, XSEDE staff are still working to complete the installs of the evaluated version of Genesis II on XSEDE SPs. The development team at UVA continues to improve upon the existing software package, incorporating a number of suggestions from the pilot sites and resolving the issues uncovered by the pilot participants. One of the major issues that still needs to be resolved is the creation of a unified namespace in order to simplify the complexity of understanding the differences between a local account on a resource and the credentials used to access with Genesis II, likely with InCommon authentication. Documentation and training need to be updated and improved as well. Additionally, while Genesis II performed adequately for small-sized files, the pilot participants with particularly large data needs cited a need for the software to adopt more High Performance behaviors. The Campus Bridging unit of XSEDE also continues to develop other resources aimed to simplify the experience of using XSEDE.

6. CONCLUSION
The XSEDE Campus Bridging pilot program was an effort to address the difficulties in transition between a resource local to the researcher and one that is at an institutional or national tier. The pilot evaluated two pieces of software: UNICORE 6 and Genesis II, focusing primarily on the development and improvement of the latter.

Pilot design anticipated and provided for a number of challenges, and while there were some difficulties in the completion of the pilot, it has resulted in the successful adoption by XSEDE of both the UNICORE 6 and Genesis II software for use on all Service Providers. Of particular note among these strengths were the efforts made to minimize the negative effects of the distributed

nature of the pilot program; regularly scheduled teleconferences and frequent asynchronous communication via mailing list were particularly effective at helping to ensure progress was being made.

An additional benefit of the pilot program has been the creation of a clearly articulated, transparent process for the adoption of new software. This will result in the ability for future pilot organizers to plan for the involvement of Operations and SD&I in the process and ensure that potential participant expectations of the kind of resources and time the pilot will require are in line with reality, allowing them to make a more informed decision about participating. Further, it will streamline the work required by those departments in order to facilitate the onboarding process for new software.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1]: ACCI Task Force On Campus Bridging. *National Science Foundation Advisory Committee for Cyberinfrastructure Task Force on Campus Bridging Final Report.* Apr. 2011. doi=http://hdl.handle.net/2022/13210

[2]: Knepper, R. *Campus Bridging Final Report.* Sept. 2013. doi=http://hdl.handle.net/2022/17378

[3]: *Genesis II Wiki.* http://genesis2.virginia.edu

[4]: *What are some examples of Campus Bridging use cases related to XSEDE?* https://portal.xsede.org/knowledge-base/-/kb/document/bbsv

[5]: *XSEDE Campus Bridging Pilot Wiki.* https://portal.xsede.org/web/campus-bridging-pilot-projects/wiki

[6]: DeStefano, L. and Rivera, L. *2013 Campus Bridging Pilot Evaluation Report.* Sept. 2013. Available as part of doi=http://hdl.handle.net/2022/17378

Author Addresses:
[i] B. Hallock, R. Knepper, C. Stewart
2709 E. 10th Street,
Bloomington, IN 47408
[ii] J. Ferguson
National Institute for Computational Sciences
1 Bethel Valley Road
PO Box 2008, Bldg 5100
Oak Ridge, TN 37831-6173