Yellowstone User Environment Survey Results

To help CISL prepare and configure the user environment for the forthcoming Yellowstone system, the User Services Section (USS) solicited input from users in a survey in December 2011 and January 2012. We received responses from 163 individuals: 88 users from 24 universities; 72 NCAR users; and nine other users.

First and foremost, we want to thank the respondents for their thoughtful comments and suggestions. We also want to let the user community know how we are following up after the survey, and to provide more details about the Yellowstone plans.

Here we focus on summarizing responses to each survey question, describing Yellowstone plans in light of our respondents’ suggestions, and addressing some of the questions they raised.

Executive summary

The survey was useful and informative in a number of ways. First, it gave us a picture of our user community’s overall level of satisfaction and of users’ satisfaction with individual systems and services. It confirmed our belief that users generally are highly satisfied but there is room to improve in several areas (survey Question 3). We took this snapshot to identify where we have room to improve and to set satisfaction targets for the Yellowstone environment.

<table>
<thead>
<tr>
<th>Service</th>
<th>Average rating</th>
<th>% Not used</th>
</tr>
</thead>
<tbody>
<tr>
<td>CISL overall</td>
<td>4.3</td>
<td>1%</td>
</tr>
<tr>
<td>Mirage/Storm</td>
<td>4.3</td>
<td>49%</td>
</tr>
<tr>
<td>Support</td>
<td>4.2</td>
<td>9%</td>
</tr>
<tr>
<td>Bluefire</td>
<td>4.2</td>
<td>2%</td>
</tr>
<tr>
<td>GLADE</td>
<td>4.0</td>
<td>23%</td>
</tr>
<tr>
<td>HPSS</td>
<td>4.0</td>
<td>24%</td>
</tr>
<tr>
<td>Training</td>
<td>3.9</td>
<td>57%</td>
</tr>
<tr>
<td>RDA</td>
<td>3.9</td>
<td>69%</td>
</tr>
<tr>
<td>Allocations</td>
<td>3.7</td>
<td>15%</td>
</tr>
<tr>
<td>Documentation</td>
<td>3.7</td>
<td>3%</td>
</tr>
</tbody>
</table>

Training and the Research Data Archive (RDA) both averaged ratings of 3.9, but a majority of respondents had not used these services; we will work on making them more widely known and useful to the community. Allocations and documentation both averaged 3.7, and we will continue to focus more attention on improving in these areas as described below.

Further, the results have helped us make important decisions regarding the Yellowstone environment. In many cases, users confirmed that plans we have in progress are on target and consistent with their needs. Respondents also made valuable suggestions, many of which are being adopted.

Some notable findings and CISL actions that have been taken or are planned in response include:

- **Workflow efficiency can and should be improved.** **Action:** As has been a goal in CISL planning, the new system will support batch workflows that span the HPC, data analysis, and visualization clusters to improve efficiency. The use of system-wide login nodes will further increase efficiency.

- **The user community expressed a desire for longer wall-clock limits.** **Action:** In most cases, the previous six-hour limit is being changed to 12 hours.
Yellowstone User Environment Survey Results, continued

- Users found Bluefire’s queue structure to be too complex, and scheduling / fair share policies difficult to understand. **Action:** The new queue structures will be simplified significantly, and policies will be documented more fully.

Questions 1 and 2 simply asked respondents to categorize themselves and identify their institutions. The other questions are addressed below in these categories:

**Scheduling/Fair share**

**User Environment**

**Software**

**Support/Documentation**

Survey results and response

**Scheduling/Fair share**

**Q4. HPC wall-clock limits**

Most users expressed a desire for longer wall-clock limits for the batch queues on Yellowstone. Among the comments on Q4 (and also on Q7, regarding fair share approaches) were a number of suggestions for structuring the queues.

A common theme was that somewhat longer limits would be good, but they should not be so long that wait times increase for everyone. Some users suggested strategies for managing longer jobs—restricting the longest jobs to fewer nodes, to separate queues, or to off-peak hours. A significant minority wanted very long wall-clock limits—two days to 10 days or even no limits at all. Almost as many users asked for a short-duration, high-priority queue for fast turnaround during the workday.

**Wall-clock limit responses**

![Wall-clock limit responses chart]

For the most part, these suggestions were in line with CISL’s thinking for setting up the Yellowstone queues, which had already been informed by comparisons with other centers, including NERSC and NICS (as was suggested by one survey respondent).
Our objectives for the queue structure included:

- Remove the 6-hour wall-clock limit.
- Simplify the queue structure in comparison to Bluefire’s ~30 queues.
- Support batch workflows that span the HPC, data analysis, and visualization clusters.
- Enable fast turnaround for development and testing work during daytime hours.
- Allow large-scale jobs to run without special request but with reasonable constraints.
- Permit CISL-settable reservations initially and explore user-settable reservations.
- Consider scheduling opportunities for very long jobs or on-demand/preemptive jobs.

For Yellowstone, we are establishing a 12-hour wall-clock limit for the main production queues. Those queues will include a “capability” queue for very large jobs that will run only on weekends; regular, premium, and economy queues; a “small” queue for short, small jobs (with higher priority during daytime hours); and a “standby” queue. These queues align well with the survey responses and suggestions and we welcome further comments. (More information about the new queue structure and the transition to Yellowstone is available here: [http://www2.cisl.ucar.edu/resources/yellowstone/transition](http://www2.cisl.ucar.edu/resources/yellowstone/transition).)

Users have asked (in the survey and elsewhere) why we intend to place the Geyser and Caldera data analysis and visualization clusters within the scheduler. There are three main reasons.

- First, we want to enable users to incorporate pre- and post-processing into their increasingly data-intensive workflows while using the most appropriate hardware for those tasks.
- Second, we want to permit users to take advantage of these clusters for batch-style GPGPU- or memory-intensive computations—during nights and weekends, for example.
- Finally, we want to leverage the scheduler information to better understand how these systems are being used, and by whom, and to inform future procurement decisions.

Q7. Fair share and scheduling

The fair share policy is an important complement to the queuing policies. Without it, the scheduler would fit jobs on a first-come, first-fit policy, modified only by the queue priority. Fair share modifies a job’s priority based on factors other than a job’s submission time in order to prevent one user from monopolizing a system.

For Yellowstone, we intend to carry over the Bluefire practices (which were the subject of a number of positive comments). Essentially, the fair share policy takes into account a user’s currently running jobs and recently completed jobs to adjust scheduling priorities. Yellowstone fair share priorities also will be managed to ensure fair access to the system by all of these users and groups: the university community, the NCAR community, the Climate Simulation Laboratory, and the Wyoming-NCAR Alliance community.

As several respondents requested, we will make sure the fair share and queuing policies are well-documented. In addition:

- We will carefully monitor fair-share scheduling management to ensure that “capability” queue users with large-scale allocations are not inadvertently penalized and are able to complete their work.
- We will consider limiting the number of jobs that a single user can schedule, as was suggested in a survey response and is common practice at other sites. That is, a user may
submit any number of jobs, but the scheduler will consider only the first X jobs from that user in planning its schedule.

We will not be able to accommodate some comments directly, such as keeping small-scale users off of the large-scale Yellowstone system. By making shared nodes available only on the Geyser and Caldera clusters, however, we will encourage small-scale users to choose them as more efficient and economical for their jobs.

One respondent also suggested that we limit backfilling and other ways of “gaming the system.” However, if users can adapt their runs to fit existing backfill windows, everyone wins. The backfill job sees a faster turnaround and other users see less competition for their future jobs. We want to encourage users to take advantage of backfill opportunities and we will provide documentation on how to do so. (The simplest advice: Provide accurate estimates of job run time in your submission scripts.) The alternative to backfill is letting nodes sit idle.

Other scheduling comments

Two comments submitted in response to Question 12 are worth noting here, under the topic of scheduling.

- Users should be able to submit jobs to both the Yellowstone, Geyser, and Caldera clusters from the same LSF instance. This is precisely the capability we are planning to provide.

- Users should have data processing tools on nodes that can directly access the file systems for output from the simulations. This has been CISL’s goal in deploying the central GLADE resources for Yellowstone, Geyser, and Caldera. (In fact, it is available now for Bluefire, Mirage, and Storm.)

By structuring the environment in this way, CISL is working to enable new workflow efficiencies and automation for users to generate, post-process, and analyze large amounts of data. For example, a user will be able to prepare and submit a set of dependent, chained jobs to read and preprocess an RDA data set on Geyser; run a large-scale simulation using the input data on Yellowstone; post-process the results on Geyser; then generate a set of visualizations on Caldera.
Q10. Allocations and accounting

About 30% of respondents estimated their Yellowstone computing needs as 1 million core-hours or more, but nearly 40% weren’t sure. Most comments favored the changes being implemented for allocations and accounting—moving from GAUs to core-hours and making separate allocations for different resources.

On the issue of analysis and visualization cluster accounting, which was raised by two respondents, we recognize this is new for users. As we make the transition, our priority will be to ensure that science gets done and that users are not overburdened with red tape.

Allocations

Other common questions were about defining or estimating core-hours, and how to convert Bluefire GAUs to Yellowstone core-hours. To calculate core-hours for a given HPC job, you multiply the number of processor cores used by the duration of the job in hours. (GAUs on Bluefire are calculated by multiplying core-hours by a constant “machine factor” of 1.4.)

On Yellowstone, 16-core batch nodes will be used exclusively by a single job at a time, so this is equivalent to:

\[
\text{# nodes} \times 16 \, \text{cores/node} \times \text{# hours} = \# \text{core-hours}
\]

Jobs that do not use all 16 cores per batch node—for example, to allow each process on a node to access more memory—still will be charged for use of all 16 cores.

And to answer one respondent’s question: Yes, charges for jobs will be core-hours adjusted by the queue priority factor.

To convert from Bluefire GAUs to Yellowstone core-hours, we have been providing the following guidance with all recent allocation instructions:

To estimate Yellowstone core-hours, assume that 1 GAU is equivalent to 0.47 core-hours on Yellowstone. That is, GAUs \(* 0.47 = 0.47\) = Yellowstone core-hours needed. Equivalently, 1 core-hour on Bluefire is equal to 0.65 core-hours on Yellowstone.

This conversion will be accepted by all review panels at least until more accurate comparisons can be made with the actual Yellowstone system.
Respondents expressed a desire for better transparency in the allocation process. We can’t post reviewer comments directly, as was suggested, because we must preserve confidentiality, but we will work to provide more information about university and other allocation awards.

Several comments and questions also related to small allocations. Our responses:

- Yes, small university allocations on Yellowstone will be larger. The limit will be 200,000 core-hours per NSF award. (That’s equivalent to more than 400,000 GAUs!)

- We will continue to make large university allocations twice per year. Still, researchers with new NSF awards can request small allocations at any time to bridge the gap until the next large allocation opportunity.

- We will continue to offer several limited opportunities for allocations not supported by NSF awards—including allocations for graduate students, post-docs, and new faculty, as well as for instructional purposes.

We also will be looking into a number of additional requests and questions.

- *Will Yellowstone resources be available for “purchase”?* It has been possible to pay for use of CISL resources in the past. We have not made any decisions on this in regard to Yellowstone; it depends in part on the level of utilization by our targeted allocation communities.

- *Will we be able to check on the usage of just-completed jobs (that is, sooner than the daily accounting permits)?* We agree this would be a useful feature and are investigating how we might provide it.

- *Will there be a web page with timings of standard CESM configurations?* The CESM team maintains such a page (see [http://www.cesm.ucar.edu/models/cesm1.0/timing/](http://www.cesm.ucar.edu/models/cesm1.0/timing/)), and we will work with the CESM team to get timings for Yellowstone. (Note that this timing table is in “pe-hrs/year”—processing-element-hours, or core-hours, per year.)
User environment

Q5. UNIX shell preference

The results here were straightforward. Nearly two-thirds of the respondents expressed a preference for tcsh/csh; bash was a distant second. Given such a clear “winner,” we will set up user accounts with tcsh by default; users will still be able to change their default shell. We also will ensure that other necessary shells are available to support CESM and job-scripting.

One respondent also indicated a need for a UNIX shell and full operating system on the Yellowstone batch nodes. These will be provided.

Q8. GLADE disk quotas and usage policies

Yellowstone, Geyser, and Caldera all will mount an 11 PB central disk resource. By default, individual users will have home and work spaces and access to scratch temporary storage space. Additional project space will be available through the allocations processes, as well.

GLADE also will have community collections space for storage of CSDL-managed and CSDL-curated data collections from the Research Data Archive, Earth System Grid, and elsewhere.

A number of survey respondents said they were satisfied with current GLADE policies, but about twice as many expressed a desire for more disk for longer periods. While we would like to give everyone all the disk space they need forever, it’s just not feasible in a shared environment like Yellowstone. However, we are considering modifications to the GLADE scratch policies for Yellowstone that may address this issue to some degree.

Some respondents commented on the need for better documentation on how to check disk usage, on the diverse purge/deletion policies for various file spaces, and on confusion over the purpose of the different spaces. To address those situations, we have updated the documentation and are reconfiguring these spaces as described on our new “Transition from Bluefire” web page—
http://www2.cisl.ucar.edu/resources/yellowstone/transition
Another repeated comment was that the current GLADE is slow. The GLADE system on Yellowstone is designed to address that issue, with an expected I/O performance of 92 GBps, more than 15 times faster.

Q9. Data transfer tools

We will support several mechanisms for data transfer in the Yellowstone environment, including SCP and SFTP, Globus Online, BBCP, and GridFTP. (We recommend using Globus Online—see http://www2.cisl.ucar.edu/docs/transfer/globusonline—because it delivers much better performance.) Some respondents were OK with the proposed set or said they used SCP primarily. Others identified additional tools we will be examining, including rsync, wget, and SCP with HPN patches.

Other comments described what users would like to see in data transfer capabilities:

- **Better documentation of the options.** We recently published new and more extensive documentation on “Transferring files” at https://www2.cisl.ucar.edu/docs/transfer and will continue to improve and update these pages.
- **Script-based, unattended file transfer.** We will continue to support this option. See the SCP page in the “Transferring files” documentation.
- **Docs for getting through security at both ends of a transfer.** We recently made Globus Online a production option for our users to address this. It greatly simplifies data transfers between remote sites and to or from a user’s desktop.
- **HPSS transfers to remote hosts or remote storage systems.** We are investigating this capability.
- **Remote mounting of GLADE file systems.** While this capability won’t be possible at the outset, CISL is interested in exploring it.
- **Data sharing via anonymous FTP-like service.** While this is possible for NCAR staff, it is not presently available to external users. We will be looking into further options for data sharing; users with such needs should contact us at cisihelp@ucar.edu.

A reminder: It will not be necessary to transfer data between Yellowstone and the Geyser or Caldera clusters because all nodes, including the login nodes, will mount the same GLADE file spaces. That is, the Geyser and Caldera nodes see the same files, in the same locations, as the Yellowstone nodes, so there’s no need to move data around.
Software

Q6. Modules and development software

The Consulting Services Group (CSG) will use **environment modules** to make the Yellowstone software environment as usable and useful as possible. For example, modules will play an important role in helping users take advantage of the different nodes on the analysis and visualization clusters, and in managing their compiler selection and use.

About 60% of respondents said they have used modules before. A common theme among the survey comments was that CISL should provide good documentation for modules, manage and test them well, and notify users of changes to modules, all of which we are planning to do. Some documentation for modules has already been published ([http://www2.cisl.ucar.edu/resources/software/modules](http://www2.cisl.ucar.edu/resources/software/modules)—modules are already used on Bluefire—and it will be updated and expanded for the more complex Yellowstone environment. CSG also will offer training that will cover the Yellowstone user software environment and use of the module command.

The survey question included a partial list of Yellowstone compilers and related development software. Modules will be used to manage those and other software packages such as NCO, CDO, IDL, and many more, as shown on this software list on the CISL website:

[https://www2.cisl.ucar.edu/resources/yellowstone/software](https://www2.cisl.ucar.edu/resources/yellowstone/software)

Q11: Yellowstone software and libraries

This question presented a **partial** list of other software and libraries to be installed on Yellowstone. Respondents identified more than 44 additional packages and libraries, some of which will be included and others that are being considered. ([See the software list on the CISL website.](https://www2.cisl.ucar.edu/resources/yellowstone/software))

Those identified by more than one respondent were:

<table>
<thead>
<tr>
<th>Respondents</th>
<th>Package/library</th>
<th>To be included?</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Python: numpy, scipy, scipy.io, matplotlib, sage</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>xdiff (and/or kdiff3, meld)</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>IDL (on YS)</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>LAPACK</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>Perl</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Bazaar, Mercurial (CVS)</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>BLAS library</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>DDT</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>emacs (w/CEDET and ECB)</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>GNU Screen</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>IMSL or NAG</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>NCAR Graphics</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Vampir</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>wgrib, wgrib2, GRIB, GRIB2</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Q12: Geyser and Caldera software and libraries

This question included a partial list of the software and libraries that CSG is planning to install for use on the data analysis and visualization clusters. Respondents identified 20 additional packages and libraries, some of which will be included and others that are being considered. (See the software list on the CISL website.)

Those identified by more than one respondent were:

<table>
<thead>
<tr>
<th>Respondents</th>
<th>Package/library</th>
<th>To be included?</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Python: scipy, matplotlib, etc</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>GRADS</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>ncview</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>xxdiff</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>NetCDF, operators</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>Tecplot</td>
<td>No.</td>
</tr>
<tr>
<td>2</td>
<td>VisIt</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Support/Documentation

Q13. Documentation

The User Services Section has recognized that our documentation of CISL resources has not been up to the standards we would like to meet, and this was reflected in the survey responses. We brought on a documentation specialist in August 2011 and already have made significant improvements.

We hope that at least some of the comments (over broken links, timeliness, and organization, for example) reflect experiences with earlier versions of the documentation. We encourage users to let us know where they have suggestions for improvement, and we recently posted a feedback form to facilitate that: https://www2.cisl.ucar.edu/docs/feedback.

Among the updates made so far, some in response to survey comments:

- **New user orientation** documentation was published in March 2012—http://www2.cisl.ucar.edu/docs/newusers
- “**Best practices**” information was developed and published in March 2012—https://www2.cisl.ucar.edu/docs/best_practices
- **File transfer.** As mentioned earlier, in January 2012 we published new documentation on “Transferring files”—https://www2.cisl.ucar.edu/docs/transfer
- **HPSS.** The HPSS documentation was overhauled in October 2011. If you have suggestions for further improvements, please let us know.
- **Broken links and outdated pages.** We have made a concerted effort to fix broken links, automatically check them on an ongoing basis, and to remove outdated pages. We will appreciate hearing about any we may have missed.
- **Tabbed pages.** Some page layouts make it difficult to use a browser’s Find command and to navigate some documentation areas. More recently developed pages avoid that problem and we are working to replace others.
- **Link to user guides on main page.** The CISL home page features a large, yellow “User Services” button (see https://www2.cisl.ucar.edu) linked directly to our main documentation menu.
- **A more focused search.** The search utility has been enhanced to address this.

Comments we will be trying to address soon:

- **Example scripts, makefiles, dot files.** We will definitely provide examples for the Yellowstone environment.
- **Standard setups and practices for compiling, running CESM and other community models.** Model-specific documentation traditionally has been provided by the individual model’s development group at NCAR. We will consider how USS might provide helpful “getting started” information to users to complement that documentation.
- **Documentation for TotalView, modules, and other software.** We have begun developing documentation for the Yellowstone environment.
- **System status.** We are working to display near-real-time system status on the CISL web site.
Q14. User support

CISL and the User Services Section strive to provide high-quality user support. Based on the survey, we do pretty well. A few comments indicate that some help requests have lingered, as have some negative opinions of the ExtraView ticket system.

Both emailing cislhelp@ucar.edu and calling 303-497-2400 are good alternatives to submitting ExtraView tickets. Recognizing that some emails coming out of ExtraView have not been easy to decipher, we have recently taken steps to simplify and restructure them. We also plan to look at making it easier for users to find and see a list of their open tickets.

One additional note is worth mentioning: CISL’s in-person user support—and, in fact, all CISL user support staff—will remain in Boulder at NCAR’s Mesa Laboratory. Users won’t need to visit Cheyenne to see one of the consultants or Help Desk staff face to face.

Q15. Training

CSG offers two weeklong training workshops each year and supports workshops offered by other NCAR labs. The primary challenge for training remains the large number of users who have not taken advantage of these offerings or who were not aware of them.

Respondent comments indicated support both for our plans to increase the number of courses that permit remote participation, via webinar or other mechanism, and to offer shorter, single-topic sessions. We intend to move forward and offer more single-topic sessions for remote users while still offering intensive workshops each year. We already provide the presentations and video of many lecture sessions from our workshops on the CISL web site. (See https://www2.cisl.ucar.edu/training/library.)

A number of respondents suggested topics that would be of interest to them: compilers and compiler options; debugging and parallel debuggers; profilers; NCAR system architecture and best practices; software carpentry; data management, data utilization, and workflows; parallel computing options; full-day on Fortran; full-day MPI/OpenMP training. We will consider these topics as we expand our remote-participation offerings.