Best Practices for Management and Operation of Large HPC Installations

Scott Lathrop, Celso Mendes, Jeremy Enos, Brett Bode, Gregory Bauer, Roberto Sisneros, and William Kramer
National Center for Supercomputing Applications
University of Illinois
Urbana, Illinois 61801, USA
{lathrop, cmendes, jenos, brett, gbauer, sisneros, wtkramer}@illinois.edu

***Abstract*—To achieve their mission and goals, HPC centers continually strive to improve the effectiveness of their resources and services to best serve their constituencies. Collectively, the community has learned a great deal about how to manage and operate HPC centers, provide robust and effective services, develop new communities, and other important aspects. Yet, cataloguing best practices to help inform and guide the broader HPC community is not often done. To improve the situation, the Blue Waters project has documented sets of best practices that have been adopted for the deployment and operation over the past five years of the Blue Waters leadership system, a large Cray XE6/XK7 supercomputer at NCSA. Those practices, described in this paper, cover aspects of managing and operating the system and its resources, supporting its users, and expanding the diversity of applications and communities. Although the technical practices are sometimes discussed relative to Cray systems, and leadership-scale systems, we believe that they would benefit the deployment and operation of other large HPC installations as well.**

Keywords-best practices; system management.

I. INTRODUCTION*

To achieve their mission and goals, HPC centers continually strive to improve their resources and services to best serve their constituencies. Collectively, the community has learned a great deal about how to manage and operate HPC centers, provide robust and effective services, develop new communities, and other important aspects. Yet, cataloguing best practices to help inform and guide the broader HPC community is not often done, particularly when it spans sites and different technology providers. Moreover, it is hard to find such guidance in the current literature.

To improve the situation for the HPC community, the Blue Waters project set about an internal effort to identify and document best practices adopted for the deployment and full service operation. Installed at NCSA in April 2013, with funding from the US National Science Foundation, the state of Illinois and the University of Illinois, Blue Waters is the largest XE6/XK7 system ever produced by Cray [1] [2], both in number of nodes as in number of cabinets, 45% larger than the next system (Titan). It was also the first leadership-scale system devoted exclusively to the entire breadth of the open-science community providing sustained-petascale performance on a wide range of scientific applications [3]. Thus, the effective management of such a complex system posed technical challenges that were addressed by NCSA with the best practices described in this paper.

Besides enabling system deployment and testing under a very tight schedule, as previously reported [4], these best practices have largely contributed to a highly productive operation and use of the system, as proved by the minimal outages observed so far and the numerous scientific discoveries that have been reported by the scientific community of Blue Waters users [5].

Some of our best practices are clearly visible to users, such as the one that populates the Blue Waters Portal with instructions on how to build and run traditional community codes on Blue Waters. However, many other best practices are not so exposed to users. One example is the practice to constantly analyze the system workload and make appropriate scheduling adjustments. While the final effect of this is certainly noticeable to users (i.e. their jobs might take less time to start running), the factors and approach taken by NCSA personnel leading to that effect are not directly observable by the users.

Meanwhile, there are best practices that are completely (and intentionally) not visible to users, and yet contribute very positively to the overall impact of the system. As an example, the practice of maintaining a strong, positive engagement with vendors enables quick handling of any issues that cannot be resolved internally by NCSA staff. It also ensures the availability of direct channels between NCSA and various teams of a certain vendor, implying that NCSA can actively participate in efforts about a given problem or about an upgrade in a certain area of the system. Over the years, this kind of relationship was shown to be critical to provide minimal interruptions in system availability to users.

These best practices are the result of NCSA's experience with Blue Waters and of our observations about other large systems. Given Blue Waters' unique capabilities in (i) computational power, (ii) storage volumes, and (iii) external network connectivity, we believe that these best practices are particularly relevant for large Cray systems. Nevertheless, many of these best practices can benefit the operation of other HPC installations, both large and small, as well.

The remainder of this paper is organized as follows. Section II presents the major areas covered by our best practices, and, for each area, we provide a few instances of

* This paper has been accepted and will be published as an article in a special issue of Concurrency and Computation Practice and Experience on the Cray User Group 2018.

the corresponding best practices. This section also shows the criteria and format defined to document each best practice. Section III describes, in some detail, many of the best practices that we adopted. Section IV contains our plans to disseminate Blue Waters results including these practices across the HPC community and, collectively, build a richer set of best practices that also include experiences from other centers. Finally, Section V concludes the paper by stressing aspects that we consider to be the most relevant ones to be shared with the community.

II. Areas of Best Practices

The best practices from the Blue Waters project cover a variety of areas. In this section, we list those areas and present major representative aspects for each. We also show the "light weight" method we adopted to properly document each best practice, which includes a description of the practice and the criteria for its selection.

A. How to get busy systems and application support staff to document the best practices:

To better organize and document our set of best practices without placing insummitable burdens of time and effort on any already busy stff, we defined a format to concentrate information about each best practice. This format is based on two "sheets". The first sheet follows what we call the Q*uadrant format*: it contains four fields, listing: (i) what that best practice is, (ii) why it is needed, (iii) who is impacted, and (iv) why this is a best practice.

The second sheet, includes an explanation of how this best practice is different from the common practice adopted in other centers, an example of use when appropriate, and any other information that is relevant to characterize the best practice. These sheets were populated by NCSA staff, and their collection represents an inventory of best practices that have been accumulated across the duration of the Blue Waters project.

A subset of these practices that seem particularly relevant to share are documented in more detail, such as in this paper.

B. Major Best Practice Areas:

For each area covered by our set of best practices, we now present a brief description of the area and list some of the factors related to that area. In subsequent sections, we will provide more details about the various best practices.

- **Project Management:** This area corresponds to overall organization of the project and its regular activities. Relevant topics include: team structure, organization and communications; community engagement (academia, industry, etc); management of allocations on Blue Waters; risk management; and project evaluation (surveys, focus groups, external reviews)
- **Deployment, Operations and System Management:** These are practices directly related to the system deployment and its operation. The area includes: acceptance testing; external and internal documentation of technologies, practices and methods; project team communications; change control processes (i.e. tracking of every change in the configuration of Blue Waters); system quality assessment; systems and resource management; storage management and cyber-protection.
- **Models of Support:** This area contains those practices related to providing a superior level of support to the users of Blue Waters. It includes our service request architecture, and the point of contact (PoC) model.
- **Communicating Success:** The practices in this area go much beyond the system, and are designed to ensure that the results from Blue Waters usage are properly communicated to society. The area includes: public relations; tours of the Blue Waters facilities; science stories describing advances enabled by the system; Blue Waters annual report; working with science teams and the annual edition of the Blue Waters symposium.
- **Expanding current communities and incorporating new communities and workforces (aka Education, Outreach and Training):** This area is related to activities aimed at extending the benefits of Blue Waters to a wider community and improving the support to that community. It includes practices related to: education and broadening participation allocations; training and education offerings; student programs; and repository of materials

As the list above indicates, Blue Waters is much more than a machine. The Blue Waters project includes a wide range of activities that are centered on the leadership system and its attached sub-systems, but the project extends the machine's scope and impact across several directions. This diversity of activities was designed to maximize the system's impact to the rate of new discoveries by the science, engineering and research communities.

III. Relevant Examples of Best Practices

In this section, we present concrete examples of our best practices, providing enough detail about them such that our practices could be guidance and implementation recipes to other centers. We align our presentation with those practices that are particularly useful on Cray systems, but many can be generalized in a straightforward fashion to other technology solutions.

Furthermore, the best practices discussed below have been called out by panels of external experts (Blue Waters has been formally reviewed almost twenty times) as well as external evaluators.

User and Application Support

The following four user and application support functions are best practices: support of community codes, support of Python, the point of contact (PoC) model, and user impersonation. These best practices have improved quality of service and reduced support overhead on Blue Waters.

The support for community codes best practice runs counter to the traditional approach implemented by most HPC centers. It was decided, from the beginning of the project, to avoid investment in providing access to prebuilt

applications typically used on HPC systems, based on the unique allocation processes for the National Science Foundation (NSF) PRAC awards, which comprise the largest allocation pool on Blue Waters. The PRAC allocations are awarded to a relatively small number of teams having a relatively small number of members. Without advanced knowledge of which codes will be used, it was decided to document on the user portal [6] the porting and running of the applications on Blue Waters rather than maintaining centralized binaries that need updating, e.g. after software upgrades, etc. This allows NCSA staff to focus on support activities rather than third party software maintenance.

The provisioning of Python on Blue Waters as a best practice takes the approach of investing the effort to provide as many Python modules in the Python instance with the goal of reducing the support load for assisting users with porting the wide range of modules and packages. This is again different than what is typically found on other HPC systems where a base Python is provided with a handful of the core modules such as numpy, scipy, and matplotlib. As reported previously in [7] and [8], Python on Blue Waters provides a unique way to allow simple access to several Python versions with very complete builds that are also easy to maintain and update. The benefit from reducing the redundant effort of repeatedly assisting users with porting and building the same set of modules outweighs the effort put into providing the more complete Python builds. It also allows the center to provide properly configured builds that use the performant numerical libraries. More recently, distributions like Anaconda are providing support for Cray Linux (CLE) with a broad range of python packages [9].

The Point of Contact (PoC) support model best practice is used across a wide range of activities. Its primary function is to provide the NSF allocated PRAC teams with a primary liaison who is responsible for each assigned teams' experience on the system. The PoC responds to service requests from the teams and follows up on issues to ensure proper treatment and resolution of issues. The PoC is responsible for knowing the needs of the teams and present them to center management. The PoCs also provide targeted support to other strategic projects on Blue Waters such as Blue Waters professors and the Graduate Fellows, where the individual support has proven effective [10]. It is important to understand that the PoC model augments rather than replaces traditional consulting and user support practices, and science and engineering teams are free to request assistance not only from the PoCs but from the entire organization.

The final support best practice is the implementation of end user "impersonation" on the system by application support staff in order to address issues reported to the Blue Waters service request system. The traditional practice is for center staff with elevated privileges (typically **`root`**) to investigate issues (sometimes changing file permissions to allow staff access to files) as the root user or to become the user via **`su`** without taking care to avoid polluting the user's shell command history. The best-practice developed for Blue Waters allows center staff to **`sudo`** a login shell as the user, record a replay log and keep a history file of the session for record keeping without leaving commands in the user's history file and allowing for accountability in case of accidents. The impersonate utility has the ability to limit which groups can be impersonated and to limit which staff can do the impersonation, limiting risk.

Topology Aware Scheduling (TAS)

The Blue Waters project guided the development of a scheduler modification that permitted job placement on nodes with consideration for physical network proximity and routing patterns. [11] A graphical representation of the 3D torus network before and after TAS was placing jobs into convex cuboids is shown in Figure 1 and Figure 2, respectively. Smaller jobs are excluded from display in Figure 2 so that the large job placement is not obscured from view. The effort required was significant, including requirements gathering, testing, and measurement of benefit, but ultimately effected an increased science throughput from Blue Waters worth millions over its lifetime.

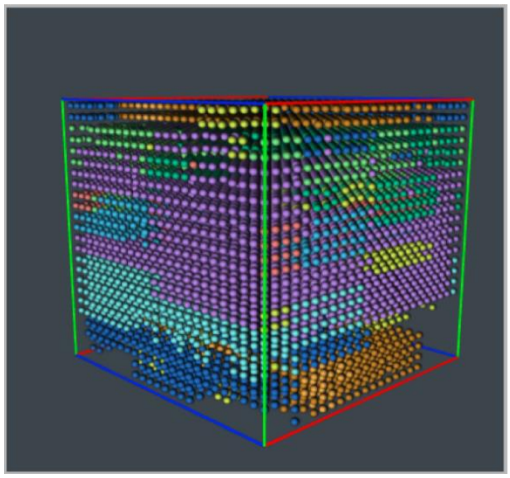

Figure 1 – Job placement on Blue Waters' torus *prior to* topology-aware scheduler installation.

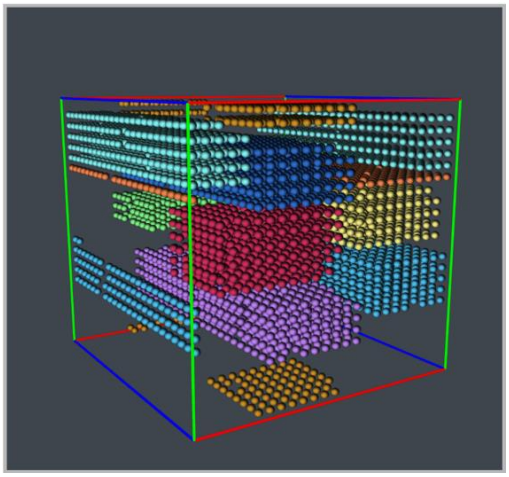

Figure 2 – Job placement on Blue Waters' torus *after* topology-aware scheduler installation.

It would have been easy to consider the status quo of random node assignment at the time to simply be a limiting factor, and the initial prediction of the increased science throughput before undertaking the effort was theoretical. Several best practices are encompassed by this effort: occasionally, take a calculated risk for great reward; maintain pressure to continually improve a resource's efficiency throughout its lifetime; and, finally, self-measurement to ensure benefit predictions are matched or exceeded.

After TAS was in production, with its value proven, these practices were re-applied and further feature enhancements were created to accommodate changing workflows, building further on the already beneficial scheduling platform.

The concerns of lower utilization have been clearly offset with applications being much more efficient (more work done over a given run), greatly improved runtime consistency, significant increases in job work factors (e.g. a 42% increase in HSN injection rates for similar jobs before and after TAS) and increase in user satisfaction. Since TAS was originally implemented, further improvements in resource scheduling have resulted in utilization being typically in the mid 90%, while still running very large applications with little delay.

Security Model

Blue Waters employs several processes and methods to achieve an operational best practice in this area. First, all traffic is monitored, both at the border routers and at zone entry points routing to the Blue Waters systems. Passive optical taps split a portion of the signal to a copy for analysis, so as not to introduce any reliability or performance dependency on the Bro Intrusion Detection System (IDS) [12]. Bro provides deep packet inspection, as well as the ability to parse and understand the semantics of network traffic, and can automatically trigger routing table updates to block detected threats. On the system, ssh daemons are instrumented at all access points for full keystroke logging. Two-factor authentication is also required at all access points, which is an extremely effective mitigation to whole categories of threat vectors related to use of fraudulent credentials.

Within the system, security policy is handled with the assumption of a compromised account. For this reason, configuration is laid out such that privilege cannot be escalated, even by administrators, when accessing the system through user access points [13]. Administrative access can only be performed when logins traverse a set of limited-access bastion hosts managed by the security team. Privilege propagation is unidirectional through multiple subsystem server-client hierarchies, so even if privilege were gained through compromise of a user access point, it cannot propagate and is isolated. The configuration is aimed not to rely on secrecy of passwords or key data to remain secure. Password use is mostly eliminated as well, and those that remain are periodically rotated.

User access to the system is managed by membership in LDAP groups, which are restricted by PAM access control. While this was straightforward for ssh configuration on login nodes, a PAM access library had to be developed to function with gridftp daemons serving the data transfer nodes (DTNs) for Globus Online access control. Additionally, grid-mapfile function has been substituted by an extension to the LDAP schema so it can be centrally managed. Access lists, which can vary per subsystem (login, near-line, or import/export), are managed centrally, and generated by a web-based administrative console. The web frontend directly queries LDAP, making manipulation of the access list convenient and less prone to human error. From there, all access points (both logins and DTNs) automatically synchronize at a high frequency so that access configuration changes are effective everywhere soon after the configuration is saved.

Sharing of Datasets

While a fully featured Data Sharing Service (DSS) did not see adequate adoption during a one-year trial to warrant continued support, and the corresponding costs would not justify extending that trial period, we have nevertheless found a data sharing utility crucial to certain workflows. The common thread during the DSS trial was the Blue Waters visualization group. After the one-year trial, Blue Waters partners were able to share data using Globus Online by configuring share permissions on the "share" directory in the project's space on the `/project` filesystem. Such an ability was invaluable during visualization collaboration tasks ranging from image generation to data conversion. The sustained successful use of these resources by the visualization group, when working with Blue Waters partners, leads us to designate such capability a best practice.

Heterogeneity

Heterogeneity is present in many places on the Blue Waters system: by node type with CPU-only nodes (XE) and CPU+GPU nodes (XK) on the same high-speed network (HSN), within the XK node with one CPU and one GPU, with differing amounts of memory per node (large memory nodes), and inter-node bandwidth differences in the x-,y-, and z-directions on the HSN. Prior to Blue Waters it was common for GPU clusters to be provided as a complementary system to a CPU based cluster [14] with little common infrastructure. The best-practice of exposing users and their applications to a diverse set of resources is an essential part of the Blue Waters mission.

The design of the Blue Waters system required the determination of the ratio of CPU-only nodes to CPU+GPU nodes. The goal was to provide a balanced system that would allow research teams to execute production workloads while allowing room for development and testing on new architectures. The best practice is the use of a quantitative approach based on a well-defined metric that represents an average, sustained measure of work. The Sustained Petascale Performance (SPP) application benchmark suite was used to evaluate ratios based on the SPP metric [3], where the composition of the benchmark suite was designed to be representative of the actual workload. A number of the SPP codes had GPU-enabled versions suitable for production

runs, allowing for a system SPP to be computed based on node counts and workload make-up. The availability of both node types allowed research teams to flexibly schedule their workloads on either resource, depending on factors like relative performance and queue wait times. To encourage experimentation with acceleration, the charge factor for the GPU node was set equal to a CPU node, despite the potential performance differences. The percentage of GPU nodes was increased [15] in part to accommodate changing workloads.

The Blue Waters large memory nodes have twice the memory of the standard XE and XK nodes. There are 96 nodes with large memory, for each node type. Jobs may target large memory nodes by specifying a node resource at job submission time, otherwise normal workloads are eligible to schedule the large memory resources. Jobs requesting large memory nodes are given a priority boost to avoid competing with large node count jobs that might get some large memory nodes included. Besides the obvious use case of smaller jobs needing more memory, such as codes with limited scalability, another use case is the use of one large memory node and N regular nodes; it is possible to use scheduler syntax to place rank 0 on the large memory node and the remaining ranks on the N regular nodes. This allows codes with a non-parallel phase such as a meshing step to use a modest amount of memory to form the mesh that is then distributed to the remaining ranks.

Approach to System Updates

Blue Waters has taken an aggressive approach to deploying software updates, both for functionality and security. Careful functionality and performance testing is performed on a Test and Development System (TDS) before the update is scheduled for the full system, and testing is conducted again on the full system to test for issues at scale. Blue Waters does not have a regularly scheduled 'preventive" maintenance outage. Almost all hardware maintenance is done with the system in service, because of the redundancy designed into the system and the dynamic routing in the HSN. At the point of this writing, Blue Waters enjoys roughly a 6 month mean time to system interrupt, including scheduled interruptions.

Instead, outages are scheduled based on need, with a goal of no more than one outage per month. For software functionality updates, the severity of the issues corrected is taken into account, and, usually, multiple updates are scheduled at once and typically no more than once every two to three months.

Blue Waters operates in a particularly open network environment, mitigated by the use of two-factor authentication, but, being a system with a large and varied user community, security patching is taken very seriously. In particular, critical security updates, like those representing a potential privilege escalation, are deployed as quickly as possible. Often, that means applying the patch and rebooting the system within 24 hours of receiving the critical vulnerability patch from our vendor. This dramatically limits the exposure of Blue Waters to a system compromise that would require a major effort to cleanup.

Notifications to Users

Keeping users up to date on activities and upcoming system changes and outages is important, but difficult to do in a way that satisfies all users. Some users want detailed information on changes, while others want no information at all. Recently, Blue Waters has deployed a notification system that helps satisfy these needs by allowing users to opt-out of different types of notifications. The notification categories include planned and unplanned outages, system notices, policy changes, upcoming training events, software updates and more general public-affairs messages. For system outages and other system changes, there are also metrics for user notification. Seven-day advance notice is required for all outages involving a system change, and one day advance notice of all outages that do not involve a system change. Critical security updates are an exception, as they do not require the usual advance notice.

Storage Management

Blue Waters best practices in storage management begin with setting up, documenting and consistently enforcing storage policies. The policies include block and inode quotas for each file system, backup policy for the `/home` and `/project` file systems, and the purge policy for the `/scratch` file system. Each of these policies is defined in the staff documentation and in the public-facing user documentation. The documentation defines the default policy, such as a 500TB group-based quota on `/scratch`; exceptions to the policy can be requested. Requests for exceptions (quota increases, purge exemptions) are submitted via the ticket system, and reviewed by project management and the storage team, based on fairness and impact of the request on other science teams and potential impact to the system as a whole.

A key part of this best practice is consistent enforcement of the policy. For example, many sites define a purge policy, but often that policy is not implemented unless the storage system is low on free space. Then a large purge is performed, surprising users who are not used to an automated purge of their data. On Blue Waters, the purge policy is defined as: files last accessed over 30 days ago are subject to purge. This policy is implemented by purging essentially every day, and automated "touching" of files is discouraged. This sets the user expectation that purges are continuous, avoids massive purges, and keeps the level of file system usage continuously below the level at which performance begins to suffer.

The second storage best practice is tracking storage usage with Robinhood. Lustre, like other file systems, provides user and group quota capabilities. However, other information on storage usage often requires an expensive walk of the file system. Robinhood avoids the expense by tracking file system changes in an external database. Queries are then faster and do not impact the file system. In addition, the database is used to generate the purge candidate list such that a file system purge pass touches only the files slated for removal, rather than requiring full traversal of the file system.

Software Management

Blue Waters, as many HPC systems, utilizes an enterprise Linux distribution for the base operating system. Enterprise distributions focus on stability and reliability rather than having the latest versions of software. This means that the distribution starts out somewhat dated compared to desktop distributions and is quite a bit behind after a couple years of service. It is common for HPC sites to need to install more updated versions of packages to meet the needs of users. Traditionally, this was done largely by hand, but there are now multiple community-supported software management tools that largely automate the installation work. Blue Waters makes use of the EasyBuild framework, in part due to its integration with the Cray-provided programming environment [16]. EasyBuild provides automatic builds of a wide variety of software, from tools such as Git and Curl to development tools such as compilers and AutoTools, and full science applications such as WRF, with the results automatically versioned and added to the modules system.

Another approach to providing updated software on HPC system is the use of container technologies. While Docker is the most widely used container tool, it poses some serious security problems in an HPC environment. Shifter [17] was the first alternative to Docker that addressed the needs of the HPC environment while still utilizing widely available Docker images. Those images can provide a completely different and much newer Linux user space over the top of the much older base OS. The images also provide benefits for metadata performance when dealing with large file counts, such as in some science experiments. Blue Waters has Shifter in production use by multiple science teams [18].

Use of Monitoring

Blue Waters is one of the most monitored HPC systems in the world, with over 20 billion performance, activity and reliability data points collected every day. Monitoring HPC systems is a complicated problem that is made much easier by collaborating with other sites. Blue Waters staff have collaborated with multiple other sites in the development of tools now in production use on Blue Waters. This effort is represented in a joint paper prepared by multiple collaborating sites at CUG [19]. The Blue Waters staff makes use of this data to diagnose many types of system issues, often related to poor application behavior. An example is given in Figure 3. This figure illustrates combining data from multiple sources to solve a potential performance problem. In this case, this includes load data from the Sonexion storage servers and IO operation compute node data from OVIS [20] to understand why there were short periods of very high load on individual OSTs. The problem turned out to be a ~500 node job performing a large amount of opens and seeks within a non-striped, modest-sized file. Once discovered, a simple solution to improve performance and lessen the impact on other users was to have the user stripe the file to spread the IO out across multiple OSTs.

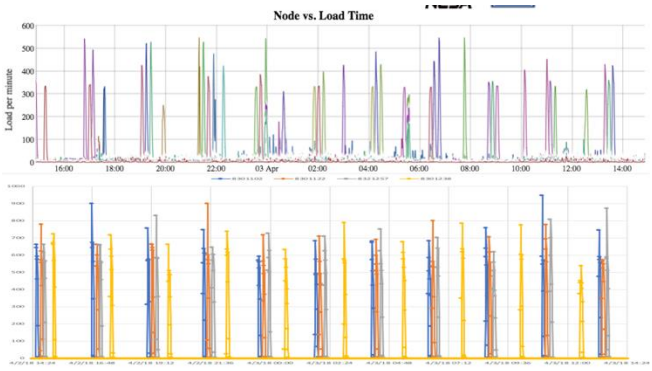

Figure 3 - The top plot shows high load spikes on individual OSTs while the bottom plot shows open and seek operation spikes on compute nodes.

Compute Node Graphics

Most analysis and visualization workflows at HPC centers leverage a data-parallel pipeline. As such, performance requirements are centered on capability for extremely large datasets and interactivity is not among the highest priorities. Furthermore, expectations of developers for large-scale visualization suites have historically offered scant support for GPU-enabled or interactive methods. Such methods, until only just recently, required access to a running X-server, and typical lack of support for these tools was evident early in Blue Waters' deployment, as this capability was not enabled by default on Cray's GPU nodes (XK) [21]. There are, however, a few scientific workflows for which analysis and visualization focuses on data that is readily representable with geometry, viewable with methods typically having lighter computational burdens than full, topologically 3D visualizations (volume rendering).

An example of this is a Blue Waters science team using VMD [22] to analyze molecular dynamics simulation data. For that, and similar groups, speeds and latencies conducive to interactivity are required. In our early support, we diagnosed issues and proposed solutions for enabling compute node graphics [21]. Still, usual software for remote interactive graphics is far from ideal, which has led us to augment the usual environment through additional third-party software. There is anecdotal evidence supporting TurboVNC [23] as a viable solution and others, including science teams on Blue Waters, have seen excellent performance from NICE DCV [24]. We therefore decided to support the latter, but this software was not developed for HPC environments and consequently required significant effort to work in even a prototypical sense. Nevertheless, we consider supporting compute node interactive graphics as a best practice in that, for the workflows for which this is important, it will remain so for the foreseeable future. This approach has the benefit of greatly decreasing science team time to insight (e.g. increasing their productivity) because it eliminates the need of time-consuming data transfers to other specialized systems.

Resource Management

Resource management is one of the key areas of interaction with users and a key component of user satisfaction. Scheduling jobs on the system to maximize throughput, meet any specific time requirements and ensure fairness between teams is an art and one that is constantly changing as workloads change and users attempt to game the system. To maintain high user satisfaction a portion of the Blue Waters systems', storage administrators, user support and project management staff meet once per week. In this weekly meeting, data from a local XDMod [25] and Integrated System Console (ISC) [26] instances and our systems and storage monitoring systems are reviewed to identify issues. Standard questions include: How is system utilization? Is the job turnaround time in each queue class reasonable? Is storage utilization reasonable? Were there any problems causing file system slowdowns? Once the base usage is analyzed, user requests for special service are reviewed. These requests include quota increases, exemption from purge, special compute-node reservations to enable faster job turn around, e.g. for a workshop or for an underserviced workload.

Scheduler reservations are a best practice in meeting several important needs. Time-based reservations allow for quick job turnaround for a set of users at a known time and are particularly useful for supporting workshops and other training events with a very limited window of use, as well some debugging challenges. Reservations are also used to provide teams with long time-evolution simulations with efficient back-to-back execution. Reservation requests are reviewed weekly based on need, fairness and impact to the overall system, with a goal of ensuring that all teams have a fair chance at utilizing their allocation, without rewarding bad behavior by allowing a team to monopolize the system at the very end of their allocation period.

When overall utilization dips, Blue Waters makes use of charging discounts to encourage increased job submissions. In the past, general discounts have been offered for specific time periods, such as over holidays when many are on vacation, but more often the discounts have been focused on encouraging specific types of submissions that will improve overall utilization. These have included discounts for jobs that are "system friendly" such as backfill jobs, jobs that were accurate with their wall time request or that use flexible wall clock time limits. Using the combination of improved system scheduling methods (e.g. TAS), incentivizing teams to submit system friendly jobs, and constant oversight and adjustment to changing workloads, Blue Waters typically runs with over 90% node occupancy (aka utilization) despite focusing on providing the shortest turnaround for extremely large jobs (320,000 cores and above).

Documentation

Writing system and user documentation is something that few people enjoy doing, so it is often a task that is put off or not done at all. However, good system documentation is very important to being able to reproduce an existing setup and to training staff not involved in the initial setup. Blue Waters staff created documentation as the system was assembled and then made a concerted effort, after system acceptance, to complete the documentation for all aspects of the system and its operation. The Blue Waters documentation is stored in a Wiki and is divided up by subsystem and service. It includes both details on system configuration and setup as well as operational documentation for supporting the system and user requests. The documentation process first divided up the documentation responsibility among staff, with the person most knowledgeable about an area generally taking the lead. Each documentation section was also assigned a reviewer from a different project area. The reviewer's job was to ensure that the documentation was clear and could be followed by someone not versed in the specific subject of the documentation page. As the system has evolved since that time, one component of a system change is updating the documentation for that change.

Managing System Changes

Managing system change is important for all systems, particularly large systems with a large support staff such as Blue Waters. Once the system is accepted, it is very important that all changes be known, and appropriate review be conducted. Without such review, it is too easy for a change to be made that inadvertently causes a regression in system performance, functionality or security that may not be discovered immediately, and may cause a significant effort to track back to the original change. On Blue Waters, all system changes go through an explicit change control process. That process starts with the creation of a ticket in the change control JIRA queue. That ticket must document the change, including:

1. A general description of the change, including what files or packages are being changed.
2. Why is the change needed?
3. The anticipated impact of the change, with explanations if the answer to any of the following is positive:
    a. Will the change be visible to applications?
    b. Will the change affect system performance?
    c. Does the change require a system outage?
    d. Are there any known potential issues associated with the change?
4. Who will make the change?
5. Has the change been implemented, tested and verified on the Test and Development System (TDS)?
6. Does the change impact system documentation?

To the extent possible, all changes are to be first verified on the small-scale test and development system. When a change-request ticket is created, a representative of each functional area (security, system, storage and networking administration, user support and project management) for both NCSA and vendor staff is notified to begin review. Change requests are discussed at multiple weekly meetings, during which the request can be approved, denied, or more information or testing can be requested. Once approved, the change is assigned to a specific staff member to implement,

and a specific time frame is chosen to implement the change. Once the change is implemented, appropriate testing is performed to verify the change and it is documented in the ticket. If necessary, the system documentation is updated before the ticket is closed. In the case of an emergency change to address an issue causing a system outage, the change control request can be submitted after the fact to document and fully review the already implemented change.

In addition to the change control process, Blue Waters also makes use of a logging process on each administration host. The logging process utilizes rcsvi to make entries in a system log book documenting all changes, including who made the change and the commands executed. That log is kept over the long term and has a daily process to email any new entries out to the entire administration team to increase awareness of all changes.

Bug Tracking

All systems have bugs and most track those bugs. On Blue Waters, bugs are recorded in the JIRA system used for all system problem reports. Indeed, a system problem report from a user may turn into a bug report once the cause of the initial problem is determined. If the bug is in a component provided by Cray, then a crayport case is opened. To help track the vendor bug, the crayport case number and subsequent bug number (CAST #) is entered into fields in the Blue Waters JIRA ticket. A custom email parsing script then provides an automated service to import changes in the crayport bug into the Blue Waters JIRA ticket. This allows Blue Waters staff to see the latest state of the bug in the vendor system directly from the JIRA interface. Bugs reported to vendors are kept open and in a waiting state, indicating the bug is awaiting a vendor response to correct until that vendor correction is received, installed and tested to verify the problem has been resolved.

Ensuring submitted bugs are getting attention requires regular review. Blue Waters conducts periodic bug reviews both with Cray and separately for all problems in the Blue Waters JIRA system. These reviews ensure that bugs do not stall waiting on input or activity. The reviews also allow Blue Waters staff to prioritize the bugs to help Cray address the most impactful issues quickly. Blue Waters also tracks and reports metrics on the responsiveness and turnaround time for both Cray and NCSA bugs. The metrics include time to a human response for a request (95% receive a response in less than 4 business hours) and time to solution (80% requests resolved within three business days).

Project Management

The Blue Waters Project Office has worked from the outset to ensure that the project is well managed. There are weekly meetings of the Blue Waters project managers, and weekly meetings with all Blue Waters staff to ensure regular communications and coordination among all aspects of the project. All expenditures are reviewed to ensure that the NSF funds are being appropriately expended to directly advance the project goals, and regular audits are conducted in coordination with the University of Illinois auditors. The security team is continuously monitoring the computing resources, the software and data of the users, compliance by all users with policies and procedures for appropriate use, and the safety and security of all people involved.

The project office tracks a set of risks related to all aspects of the project using a custom Risk Register Tool [27] available as open source. As summarized in Figure 4, the current risks are tagged relative to the impact on the project and the probability of occurrence. Red represents a high risk, and green represents a low risk. During the early deployment phase of the project, there were a number of high risks. Each risk includes management approaches that are intended to decrease the probability of the risk occurring and/or decreases the impact if it does occur. Furthermore, each risk has documented mitigation strategies to be pursued if the risk is triggered. The risks are reviewed on a monthly basis to determine if any need to be adjusted, mitigated or retired. As the figure displays, there are no longer any high risks, but we recognize there will always be some risks with any project.

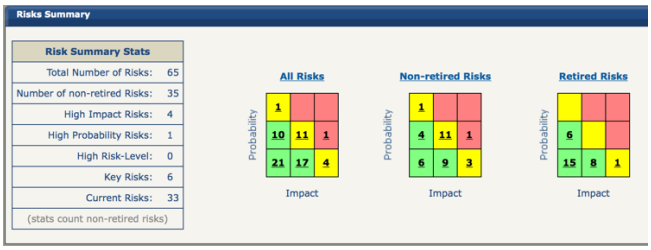

Figure 4 – Screenshot of Blue Waters' risk register tool

Blue Waters, in coordination with NSF, conducts regular project reviews to track overall project progress and to identify tasks needed to adjust plans, or to address any deficiencies. Early in the deployment phase, NSF conducted frequent panel reviews, which were very effective. Once the project entered the operations phase, it was determined that an annual NSF review was sufficient to track progress and recommend corrective actions. External evaluators were engaged from the outset to provide constructive feedback, allowing the project to improve the resources and services. The evaluators conduct surveys, focus groups, and interact with PIs, students, users, and Blue Waters staff to provide a comprehensive report that addresses the impact of the project, including recommendations for improvements. Furthermore, Blue Waters has several advisory committees that provide expert guidance. During development and deploy, there were three advisory committees: systems and architecture, applications and engagement with science teams, and data and visualization. Since operations started, the Science and Technology Advisory Committee meets with the project quarterly.

To gain a deeper appreciation for the impact of Blue Waters on discovery, Blue Waters contracted with IDC (now Hyperion Inc.) to conduct a study investigating the scientific returns from research projects conducted on the Blue Waters supercomputer system at NCSA. "In IDC's opinion, confirmed that Blue Waters has proven to be an exceptionally—and in some case uniquely—competent

platform for accelerating scientific innovation. The Blue Waters-enabled innovations described and ranked in this study will produce strong benefits for the scientific disciplines they belong to. They also have great potential for benefiting U.S. industry and American society as a whole over time."

Procedure for Acceptance Testing

The installation of any new component in Blue Waters follows a protocol that includes careful planning, extensive testing and formal certifications for acceptance. Starting with the original deployment of the system, NCSA established a comprehensive procedure for acceptance testing [4], which included (i) a phase of designing tests, (ii) a period of a where the tests were applied, and (iii) a certification phase that analyzed and approved the results from those tests.

To organize the test-design phase, NCSA created an internal tool to manage every test that was designed. This tool was basically a database with a graphical interface, and each register associated to a given test included information such as how to execute the test, the criteria for the test to be successful, etc. Meanwhile, the reports about test execution, including the obtained results, were stored in an internal Wiki accessible to the entire Blue Waters team. Certain sections of this Wiki could also be viewed by vendors, such that those vendors notified about problems detected in the tests.

This quality assurance procedure ensured that system acceptance was conducted with several levels of control and coordination: while an NCSA staff member with expertise on a certain area was responsible for designing and applying a given test, a different member was tasked with analyzing and eventually approving the test results reported in the Wiki. A third member, with management responsibility, would then certify the complete process. Thus, every test was subject to the screening of several staff members, ensuring a thorough testing process.

A similar procedure, at the proper scale, was adopted for accepting various other additions to Blue Waters, such as the installation, in the summer of 2013, of twelve extra cabinets containing GPUs [15], to encourage migration of applications to accelerated versions. By the same mechanism, a rigorous testing was conducted for accepting the new job scheduler that included awareness about the topology of the Blue Waters interconnection network [28].

IV. PLANS FOR DISSEMINATION

A Blue Waters project goal is to share lessons learned and best practices to help other organizations learn from our experiences to better serve their constituencies. And in return, we gain lessons learned from other organizations to help us improve our resources and services. We are exploring multiple opportunities for sharing, by writing papers (such as this paper to the CUG audience), by conducting workshops and tutorials, such as at the upcoming PEARC18 Conference, and by presenting information through the Blue Waters webinar series. We will also be exploring publishing the best practices and lessons learned in one or more journals to ensure broad dissemination.

The Blue Waters project actively encourages staff members to participate in HPC community workshops and conferences to share techniques and tools for managing HPC systems and learn about future changes to key technologies. These venues include vendor or product specific meetings including the Cray User Group, Lustre User Group, HPSS User Group, NVIDIA's GPU Technology Conference and more. The project also participates in general HPC forums including the SC conference series, HPC focused system administration workshops and training events and the Joint Laboratory for Extreme Scale Computing (JLESC). JLESC consists of an international group of leading HPC centers and focuses on research and development of tools and techniques needed for future extreme-scale computing systems.

The Blue Waters project conducts an annual Symposium in the spring each year with a goal of building an extreme-scale community among researchers, developers, educators, and practitioners. This forum is unique in that it brings together a cross-disciplinary group of over 50 NSF PRAC PIs and an audience of 150-200 researchers to identify petascale and extreme scale requirements, recommend future directions, and to identify improvements to resources and services. The Symposium has been very effective in fostering the exchange of challenges, opportunities and solutions across diverse fields of research as well as helping advocate for the benefits of HPC and leadership computing.

As an open science system, all projects on Blue Waters are public information. However, Blue Waters goes further than most other HPC centers in making that information easily available to the general public through the web portal. Title, principle investigator and a summary description of every Blue Waters project, past and present, are available on the web portal. In addition, the projects usage over time is also available. This allows the general public to easily see the quality and diversity of computational science as well as providing information on how responsibly each team has made use of their allocation. Also, attached are lists of publications and in many cases video presentations of results captured at the symposium.

Finally, all projects contribute a two-page summary of their work on Blue Waters which are combined with overall Blue Waters information into an annual report. The summaries are written at a level to be understandable to the general public, enabling the book to be used as a way of broadly disseminating the quality of science performed on Blue Waters, helping to justify continued funding. The book is available in a glossy printed form that is sent to key stakeholders at NSF and congress, as well as in pdf form that is freely downloadable [5] from the Blue Waters web portal.

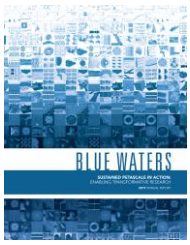

Figure 5 – Cover of Blue Waters book

V. CONCLUSION

Over the last five years, NCSA has been executing the operational phase of the Blue Waters project, which includes not only operating Blue Waters, a large Cray XE6/XK7 system, but also conducting a series of activities aimed at ensuring that Blue Waters remain a valuable and efficient asset for the US science and engineering community. The project thus includes a variety of areas, such as user support, interactions with vendors and with the community, dissemination of results, and others.

As the first system available to open science providing sustained-petascale capability, Blue Waters presented many new challenges, which had to be managed and improved over time. To address these challenges, NCSA developed and employed a set of best practices, to enable maximal productivity to the users and their applications running on the system. According to the scientific results obtained from Blue Waters usage [5], we can confirm that the system has been productive and that those best practices have proven useful in providing an excellent environment to users.

This paper presented many of the best practices that we adopted, over the years, for Blue Waters deployment and operation. We presented a list of the areas covered by our best practices, and for many of those practices, we provided detailed descriptions of their major aspects, including, in some cases, the motivation and historical evolution based on our experience with the system. We also presented our current plans to disseminate widely these best practices, including our desire to complement our practices with those shared by other institutions, thus forming a richer set of best practices that would become a valuable resource to the HPC community.

Because the experiences acquired from running large systems are rarely reported in the literature, we believe that this paper contributes with useful guidance for any center that needs to deploy large systems in the future. Although our experience with Blue Waters would be directly applicable to many large Cray machines, the multitude of activities in our project and our adoption of best practices in many of those activities mean that these experiences can benefit large systems from other vendors as well.

ACKNOWLEDGMENTS

This work is part of the Blue Waters sustained-petascale computing project, which is supported by the US National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

REFERENCES

[1] B. Bode, M. Butler, T. Dunning, W. Gropp, T. Hoefler, W.-m. Hwu and W. Kramer, "The Blue Waters Super-System for Super-Science," in *Contemporary HPC Architectures*, vol. Chapman and Hall/CRC 2013, J. Vetter, Ed., Sitka Publications, November 2012.

[2] W. Kramer, M. Bulter, G. Bauer, K. Chadalavada and C. Mendes, "Blue Waters Parallel I/O Storage Sub-system," in *High Performance Parallel I/O*, Prabhat and Q. Koziol, Eds., Boca Raton, FL: CRC Publications, Taylor and Francis Group, , 2015.

[3] C. L. Mendes, B. Bode, G. H. Bauer, J. Enos, C. Beldica and W. T. Kramer, "Deployment and Testing of the Sustained Petascale Blue Waters System," *Journal of Computational Science*, vol. 10, pp. 327 -- 337, 2015.

[4] C. L. Mendes, B. Bode, G. H. Bauer, J. R. Moggli, C. Beldica and W. T. Kramer, "Blue Waters Acceptance: Challenges and Accomplishments," in *Proceedings of CUG-2013*, Napa, CA, 2013.

[5] University of Illinois, "Blue Waters Annual Report," 2018. [Online]. Available: https://bluewaters.ncsa.illinois.edu/annual-report.

[6] NCSA, "Blue Waters Community Codes," [Online]. Available: https://bluewaters.ncsa.illinois.edu/community-codes.

[7] C. Maclean, "Maintaining Large Software Stacks in a Cray Ecosystem with Gentoo Portage," in *Cray User Group*, 2016.

[8] C. Maclean, "Python Usage Metrics on Blue Waters," in *Cray User Group*, 2017.

[9] NERSC, "Anaconda Python," NERSC, 2017. [Online]. Available: http://www.nersc.gov/users/data-analytics/data-analytics-2/python/anaconda-python/.

[10] L. DeStefano and J. S. Sung, "2017 Blue Waters User Survey Report," Champaign, 2017.

[11] J. Enos, G. Bauer, S. Islam, M. Steed, D. Jackson and R. Feidler, "Topology-Aware Job Scheduling Strategies for Torus Networks," in *Cray User Group*, Lugano, Switzerland, 2014.

[12] "The Bro Security Monitor," The Bro Project, 2014. [Online]. Available: https://www.bro.org/.

[13] B. Bode, T. Bouvet, J. Enos and S. Islam, "Account Management on a Large-Scale HPC Resource," in *HPC Systems Professional Workshop Supercomputing 2016*, Salt Lake City, UT, 2016.

[14] V. V. Kindratenko, J. J. Enos, G. Shi, M. T. Showerman, G. W. Arnold, J. E. Stone, J. C. Phillips and W.-m. Hwu, "GPU clusters for high-performance computing," in *Cluster Computing and Workshops*, New Orleans, LA, USA, 2009.

[15] C. L. Mendes, G. H. Bauer, W. T. Kramer and R. A. Fiedler, "Expanding Blue Waters with Improved Acceleration Capability," in *Proceedings ofCUG-2014*, Lugano, Switzerland, 2014.

[16] P. Forai, G. Peretti-Pezzi, B. Bode and K. Hoste, "Making Scientific Software Installation Reproducible on Cray Systems Using EasyBuild," in *Proceedings of the Cray Users Group Meeting (CUG2016)*, London, 2016.

[17] D. Jacobsen and S. Canon, "Contain This, Unleashing Docker for HPC," in *Proceedings of the Cray User Group (CUG2015)*, Chicago, 2015.

[18] H. Leong, T. Bouvet, B. Bode, J. Enos, D. King and M. Showerman, "Installation, Configuration and Performance Tuning of Latest Shifter Release on Blue Waters," in *Proceedings of the Cray User Group (CUG2018)*, Stockholm, 2018.

[19] V. Ahlgren, S. Andersson, J. Brandt, N. P. Cardo, S. Chunduri, J. Enos, P. Fields, A. Gentile, R. Gerber, J. Greenseid, A. Greiner, B. Hadri, Y. (. He, D. Hoppe, K. Urpo, K. Kelly, M. Klein, A. Kristiansen, S. Leak, M. Mason, K. Pedretti, J.-G. Piccinali, J. Repik, J. Rogers, S. Salminen, M. Showerman, C. Whitney and J. Williams, "Cray System Monitoring: Successes, Requirements, and Priorities," in *Cray User Group*, Stockholm, Sweden, 2018.

[20] A. Agelastos, B. Allan, J. Brandt, P. Cassella, J. Enos, J. Fullop, A. Gentile, S. Monk, N. Naksinehaboon, J. Ogden, M. Rajan, M. Showerman, J. Stevenson, N. Taerat and T. Tucker, "Lightweight Distributed Metric Service: A Scalable Infrastructure for Continuous Monitoring of Large Scale Computing Systems and Applications," in *Proc. IEEE/ACM International Conference for High Performance Storage, Networking, and Analysis (SC14)*, New Orleans, 2014.

[21] M. D. Klein and J. E. Stone, "Unlocking the full potential of the Cray XK7 accelerator," in *Cray User Group*, 2014.

[22] W. Humphrey, A. Drake and K. Schulten, " VMD: visual molecular dynamics," *Journal of molecular graphics*, vol. 14, no. 1, pp. 33-38, 1996.

[23] "TurboVNC," [Online]. Available: https://www.turbovnc.org/.

[24] "NICE DCV," [Online]. Available: https://www.nice-software.com/products/dcv.

[25] J. T. Palmer and et al, "Open XDMoD: A Tool for the Comprehensive Management of High-Performance Computing Resource," *Computing in Science & Engineering*, vol. 17, no. 4, pp. 52-62, 2015.

[26] J. Fullop, A. Gainaru and J. Plutchat, "Real Time Analysis and Event Prediction Engine," in *Proceedings of CUG-2012*, Stuttgart, 2012.

[27] NCSA, "NCSA Risk Register Tool," 2015. [Online]. Available: https://wiki.ncsa.illinois.edu/display/ITS/NCSA+Risk+Register.

[28] J. Enos, G. Bauer, R. Brunner, S. Islam, R. A. Fiedler, M. Steed and D. Jackson, "Topology-Aware Job Scheduling Strategies for Torus Networks," in *Proceedings of CUG-2014*, Lugano, Switzerland, 2014.