Best Practices and Lessons from Deploying and Operating a Sustained-Petascale System: The Blue Waters Experience

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

***Abstract*—Building and operating versatile extreme-scale computing systems that work productively for a range of frontier research domains present many challenges and opportunities. Solutions created, experiences acquired, and lessons learned, while rarely published, could drive the development of new methods and practices and raise the bar for all organizations supporting research, scholarship, and education. This paper describes the methods and procedures developed for deploying, supporting, and continuously improving the Blue Waters system and its services during the last five years. Being the first US sustained-petascale computing platform available to the open-science community, the Blue Waters project pioneered various unique practices that we are sharing to be adopted and further improved by the community. We present our support and service methodologies, and the leadership practices employed for ensuring that the system stays highly efficient and productive. We also provide the return on investment summaries related to deploying and operating the system.**

Keywords—best practices; system management; HPC center

I. Introduction

Making extreme-scale systems work efficiently and productively for a range of frontier science research presents many challenges and opportunities that need new methods and practices. Unfortunately, the solutions created, experiences acquired, and lessons learned from deploying and operating extreme-scale systems are rarely reported in the literature, despite the fact that they can be widely applicable, regardless of system size. Thus, we believe that the materials presented in this paper will contribute to enhancing the community's body of experiences and skills in handling such systems.

Since its deployment at the National Center for Supercomputing Applications in 2013, with funding from the National Science Foundation (NSF), Blue Waters has been the leading extreme-scale system available to the open-science community in the US. In what might be our first best practice, the Blue Waters project, NCSA and NSF very intentionally and publicly announced the system would not be listed on the Top500 list because that list does not represent an accurate assessment of sustained performance. The reasons for this are too long to discuss in this paper, and are well documented in [1]. But to give the reader a sense of the extreme scale of Blue Waters, we briefly compare Blue Waters to a similar, but smaller Cray Gemini HSN system that was listed on the Top500 list, the *Titan* system at Oak Ridge National Laboratory [2]. Compared to Titan, Blue Waters is a mixture of all CPU and CPU/GPU nodes. It has a 24x24x24 Torus that has 44% more links, 44% more cabinets, 43.7% more computational nodes, 165% more AMD Opteron processors running at 10% higher clock rate, and has 119% more total memory. Compared to Titan, Blue Waters has over twice the sustained performance measured by the Sustained Petascale Performance (SPP) tests [3] [4], and has more storage and a much higher I/O bandwidth. Blue Waters is, by far, the largest and most complex system ever built by Cray Inc. Full descriptions of Blue Waters can be found at [5] and [6].

Being the first system able to provide sustained-petascale performance to a broad range of science and engineering applications, Blue Waters featured several innovative design decisions that had to be carefully considered and implemented. For example, in order to support a broad range of science applications and accelerate numerically-intensive workloads, it was decided to instrument a fraction of compute nodes with GPUs. Today, both CPU and GPU-accelerated nodes share the same interconnect and access the same filesystems, improving users' productivity and encouraging them to migrate their compute-bound CPU applications to GPUs. At the time of deployment, however, the challenge was to place those GPU-enabled nodes in a way that minimizes the interference [7] between the jobs running on different types of nodes. As we report, finding the best configuration required a series of modifications to the system.

Blue Waters' unique feature is its balance of excellent computational performance, petascale storage system, scalable visualization support, and high-speed network that connects it to the rest of the world. While such a balance is critical to providing an optimal environment for many science projects, it presents additional facets that must be managed and jointly optimized, so that the full potential of the system can be achieved by applications.

More than a single machine, Blue Waters is a project that contains multiple inter-related elements, requiring a coordinated management effort focused on the success of science teams running on the system. To overcome the challenges of deploying such a complex project, NCSA adopted a comprehensive project management structure that extended from pre-deployment to operation of the system. That structure also included a significant component of user

Blue Waters is funded by NSF (awards OCI-0725070 and ACI-1238993) and by the state of Illinois.

SC18, November 11-16, 2018, Dallas, Texas, USA
978-1-5386-8384-2/18/$31.00 ©2018 IEEE

support that has proven to be critical for application owners to achieve their scientific goals by using the system. The number of new discoveries and advances enabled by the five years of Blue Waters operations shows that the investment on the project has truly achieved its original goals of enabling new advances in science.

This paper describes our overall support and service methodology and lists many of the leadership best practices that we employed, in a variety of system and support areas, to ensure that Blue Waters is highly efficient and productive throughout its lifetime. We also list returns on investment summaries related to deploying and efficiently operating the system and helping science teams be the most productive.

The remainder of this paper is organized as follows. Section II presents our methodology for managing most aspects of the project. Section III describes, in chronological order, different configurations of the system since its original installation. Section IV explains the infrastructure that was deployed to extensively monitor the system and the facilities in place to analyze the captured data. Section V covers the existing visualization support for users that require graphical capabilities, while Section VI shows the improvements in performance for key science applications in the system. Section VII lists our education and outreach activities, which focus on training and improving the qualification of the community of users. Section VIII summarizes the returns on investments made on Blue Waters acquisition and operations, and Section IX concludes the paper by stressing its most relevant points.

II. Methodology for Project Management

To manage the various phases and efforts employed in the Blue Waters project, NCSA adopted an extensive structure of methods and procedures that was followed by all project participants and constantly monitored by the project leaders. Project personnel were partitioned into several *teams*. A central, coordinating team was the *Project-Office*, which included top leadership and executive managers or assistants overseeing the entire project. Other teams covered specific areas, such as system administration, application support, storage, networking, security, facilities, education and outreach, industrial components, human resources and public affairs. The number of persons varied greatly across teams but was always kept at a level such that each team could deliver its share of the project responsibilities.

Each of the teams above has a weekly meeting, to review existing issues and discuss potential solutions. There is also a weekly update meeting for the entire Blue Waters staff, where general subjects are reviewed, and cross-team aspects are debated. One important practice that is observed is *a remote participation component in every scheduled meeting*: the meetings always have a dial-in number, such that project members who cannot be physically present at NCSA can still participate and contribute to the discussions.

Since the start of the project, a critical component was an internal Wiki, accessible by all project members and with appropriate/individualized permission levels, containing notes and documents in all areas of the project. This structure was essential to avoid the need of sending documents (many with confidential material) by email, and to create a chronological repository of actions conducted by each team. It also hosted several documents resulting from many of the project's activities. These activities, listed according to their corresponding best practices adopted, include the following:

A. JIRA-Based Ticketing System as Information Nexus

The Blue Waters project uses an *Atlassian Jira-based ticketing system* [8] for fusion of basic and advanced user support, with participation of project members from all teams. This includes monitoring tickets, even if they do not directly concern staff work or supported users. This leads to an awareness of system/project state, including new features, problems, solutions (or at least whom to ask for a solution), problem-solving/debugging techniques and tricks, project policies, etc.

B. General Approach for New Component Acceptance

For installations and upgrades of any component on Blue Waters, our approach is to *implement detailed acceptance planning and employ multiple levels of coordination and approval.* This maximizes the likelihood that new components will behave according to specifications, and that upgrades will not degrade observed quality of service for the system. It also minimizes the chance for users to be affected by unexpected bugs in new products, as those bugs would be proactively detected by NCSA staff in early tests.

C. Installation of Early-Science System

Prior to Blue Waters' full installation, NCSA deployed a smaller Early-Science System (ESS), comprising 48 cabinets populated with the same type of XE nodes that would be in the final machine. A few selected application groups were given access to ESS, in a user-friendly mode. The use of ESS was very important, both for application teams to become familiar with the Blue Waters environment, and for NCSA staff to start exercising some acceptance test procedures.

D. Strong Interaction with Vendors

Throughout the project, we maintain a strong interaction with the vendors and their teams that support the products used on Blue Waters. By forcing the vendor to actively work on a given problem and provide a timely fix, either through a new version of the product or through a patch to an existing version, we ensure that problems observed are solved quickly and effectively, minimizing any possible disruptions or inconvenience to users of Blue Waters.

As an example of such interaction, we hold a bi-weekly Cray/NCSA bug meeting, where major bugs are reviewed and progress towards resolution is analyzed. Figure 1 shows the evolution in the number of Critical or Urgent bugs that were detected during the pre-deployment of Blue Waters. According to informal discussions with Cray personnel, NCSA was their customer with the largest number of bugs reported during a system acceptance period. This probably reflects both the complexity of the system and the thoroughness of our testing approach. We continue to track bugs, in a similar fashion, throughout the system lifetime.

Similar review meetings are held regularly with other vendors, such as the storage OEM provider and the job scheduler creator. Whenever there is unsatisfactory progress on the resolution of a given bug, the issue is escalated to project management, where it is handled directly with the vendor via higher-level channels. This direct communication between higher ranks has proven to be necessary for appropriate resolution of some bugs discovered so far.

E. Milestone Tracking and Certification Processes

Keeping track of an activity's progress – including *certifying and making recommendations on the milestone reports* – throughout the duration of the activity ensures that the acting team can get past any issues that might cause their effort to stagnate. Certification of completion is done by a separate staff member with knowledge in the matter. Accountability is essential to ensure that groups continue making progress toward their goals, but having a certification process also gives Blue Waters staff a chance to check in on a team's progress in a structured manner and offer any tips or insights they might have.

F. Management of Project Risks

NCSA managed project risks through *an internal risk register tool*. This is essentially a database with a graphical user interface and contains a record for each factor that is viewed as a risk for the project. That record stores a description of the risk, its current status, dates when it is expected to be monitored or triggered, and mitigation actions that should be taken in case the risk is triggered. Since it is a home-grown tool, it can be easily adapted or adjusted to the project's needs. This open-source tool has been made available to other centers as well [9].

Throughout the entire project, NCSA closely monitors the risk register, and takes action whenever a certain factor would trigger some of the planned mitigation alternatives. Each risk is classified according to its probability and its impact on operation of the system. Both probability and impact are assigned a level of low/medium/high, and key risks (i.e. risks with a high probability or impact) are clearly marked in the risk register. Each risk also has a staff member who is the risk owner, in charge of making monthly updates to the register regarding the status of that risk.

G. Protocol for Project-Level Change Control

Blue Waters implemented a project-level Change Control method that is intended to address changes in statements of work or other project design documents. It is designed to reduce the impact of modification to the project by managing changes that result in modifications to the project's scope/deliverables, schedule and cost. Furthermore, the main goal of the process is to prevent unauthorized or uncontrolled physical hardware changes to equipment, changes to controlled software and changes to controlled documents.

The Change Control process is applicable to all work performed as part of the Blue Waters project. The scope of this program encompasses the lifecycle of the project. Each change procedure consists of (1) a Change Request, (2) a review, and (3) an approval or rejection. Changes are classified into one of three levels, according to their cost and schedule implications. There is a Change-Control Board, comprising project leaders, that analyzes those requests and decides if they should be approved.

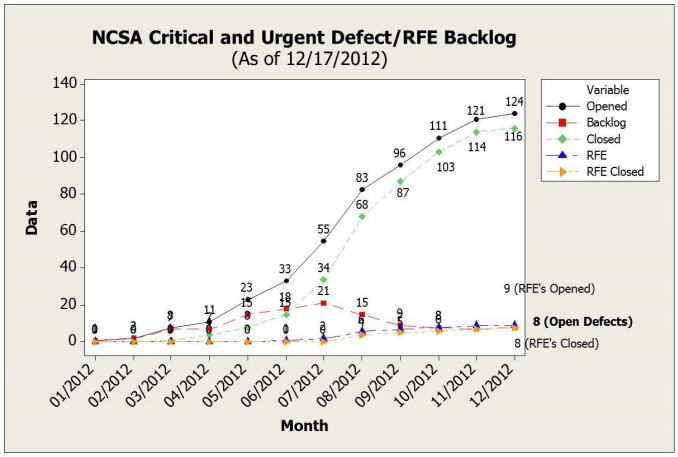

Fig. 1. Critical and Urgent bugs detected during system pre-deployment.

H. Documentation Provisioning and Sharing

In addition to provisioning documentation about system usage via traditional web-based channels (e.g. Blue Waters Portal [10]), sharing or exchanging documentation between centers operating extreme-scale systems is also very important. This may include both installation and support materials. Quality documentation improves overall experience and reduces support level of effort. Other centers can do some things better, and by sharing their good practices we raise the level of quality in our service to the community. This practice reduces costs (work, time, etc) while improving user experience and effectiveness.

I. Point-of-Contact User Support Model

Each major science team is *assigned a Point-of-Contact (POC) for the duration of their Blue Waters allocation*. With a dedicated POC, the science team can write in with questions to someone who is already familiar with that team's software and science approach. The POC is typically a member of NCSA's Science and Engineering Application Support (SEAS) team. An informed response can be given much more quickly than the typical approach where the support staff has little knowledge of science teams and service requests pass through a traditional hierarchical support structure.

The level of service provided by the POC approach can be substantially *higher* than that provided by the standard HPC center support structure [11] [12], but is similar to DOE support structures such as ORNL liaisons [13]. The higher level of service enables greater productivity from the science teams.

J. Public Relations Efforts

To bring together, in a single document, the broad range of scientific and computing collaborations and science successes and achievements made possible by Blue Waters, *NCSA's public relations team has annually published a Yearbook* [14].

Every project with a Blue Waters allocation is represented in the yearbook, either with a full report or a project listing. The yearbook is distributed to Blue Waters stakeholders and members of the general public interested in science and technology. It is available in a printed paper format as well as online as a PDF file on the Blue Waters Portal [10].

III. Configuration Evolution

Blue Waters today is not the machine it was when it arrived. The hardware and software configurations have both been upgraded several times to keep the system relevant, improved, and meeting user demands, with careful attention to cost-benefit analysis of each adjustment. When possible, changes are first tested on the single cabinet Test and Development System (TDS). Depending on the magnitude and type of change, full system scale-test time is allotted adjacent to a maintenance window for implementation as well.

After Blue Waters was initially operational at 276 cabinets, it was deemed appropriate to add 12 additional cabinets of compute nodes [15]. This change also included a physical redistribution of I/O nodes within the machine, bringing additional complexity on both software and hardware fronts, with as much work as possible being performed with the system in operation. That maintenance was meticulously planned, reviewed, and rehearsed months prior to the event, with all actions approved through change control. The unavailable system time was also minimized by consolidating changes, as much as possible, to a single extended maintenance window, which also necessitated staff rotation planning within the maintenance period. While this event represented one of the more complex maintenance scenarios, the practice of applying project management style organization and planning to every maintenance task maximizes efficiency and predictability and minimizes error.

Early in the life of Blue Waters, a limitation was realized with the XK7 CPU/GPU nodes when compared to some traditional GPU clusters. Aside from computation acceleration, GPUs are sometimes used in HPC for accelerating graphics rendering, which is what GPUs do best. This required some clever software enablement technique that allowed the XK7 operating system to *pretend* like it had a traditional GPU installed with an external graphics port [16]. Initially, doing this meant leaving a supported mode where the vendor was concerned. However, after successfully demonstrating that this can be extremely valuable to several HPC applications for post processing, visualization, and in some cases completely eliminating a very time-consuming data movement to a remote resource, the vendor became sufficiently convinced to integrate the solution into their supported product. We consider this an example of a general best practice: occasionally taking a calculated risk and departing the confines of vendor support in order to lead and demonstrate a valuable innovation for eventual consumption by the community and its vendors.

Blue Waters employs a high-performance network with a 3D Torus topology interconnecting compute nodes. An extensive investigation into inconsistent performance that applications were experiencing yielded a conclusion that placement on the Torus network could greatly impact performance. A topology-aware scheduling (TAS) extension to the workload management system was developed to take the network location into account when scheduling jobs. While it was well known that TAS significantly improved performance in many scenarios [7], it also represented a reduction to overall node occupancy. Six-month trial periods were compared between each scheduling mode to identify which mode ultimately produced more science, which could only be effectively estimated by leveraging performance metrics collected over time. TAS ultimately showed a clear advantage over the traditional scheduler. The TAS project encompasses multiple best practices, ranging from continually monitoring application performance, fully investigating anomalies, investing in innovation to measure and maximize real system utilization (as opposed to node occupancy), and finally putting TAS into production use.

The Lustre storage subsystem on Blue Waters was already maturely utilized in full production when a significant technology evolution, declustered arrays, became available as an upgrade. The catch was that it required a full reformat of a filesystem to apply. With much to gain in Mean Time To Data Loss (MTTDL) and single-stream performance specifications, the Blue Waters team worked with vendors and formulated a plan to accomplish the upgrade with minimal downtime. The solution required in-house development of custom data movement and synchronization tools designed to purpose [17], and a complex rotation of filesystems into free space, as transparent as possible to ongoing production use of the system. A phased approach was taken, ultimately completing a full upgrade in an operation spanning months, but mostly transparent and seamless to Blue Waters use, except for a few relatively brief maintenance interruptions to adjust mount points. Best practices employed for this upgrade project include cost-benefit analysis, application of project management techniques to maintenance planning, checking and double-checking data integrity, and keeping the system relevant with modern technological evolutions.

Large per-node memory capacity is well-known as a key strength of Blue Waters' design, but some applications still demand more on occasion. In several instances, an application just needed more memory per node on a single node within a job responsible for coordination, or on just a few nodes in a smaller post-processing job. The project decided, with input from its volunteer advisory committee, to purchase enough memory to double capacity in two cabinets of Blue Waters, representing 96 XK (CPU/GPU) nodes and 96 XE (CPU) nodes. The nodes to be upgraded were carefully selected to minimize disruption to TAS jobs that do not have an extra memory requirement, but to still permit all the upgraded nodes to be used in a TAS mode if necessary (topologically consolidated on the network). Key practices represented by this change include observation of evolving application demands and involving the project's advisory committee in decisions about how to best improve service.

IV. Holistic System Monitoring and Analysis

System monitoring at some level is performed at almost all HPC centers. However, the system monitoring at most sites is limited in scope and results in disjointed sets of information.

For example, most sites collect logs on batch system usage and centralize syslog streams from all hosts, and many collect at least some performance statistics from the nodes. Unfortunately, these streams of information are rarely correlated in any automated fashion, such that attempting to track a problem to a specific job on a specific set of nodes is often a cumbersome manual process.

In contrast, Blue Waters is quite likely the most instrumented HPC system in the world. The general philosophy is that if something can be measured without impacting application performance, then it is measured and recorded. In addition, Blue Waters makes use of not only passive monitoring approaches, but also active monitoring to probe performance and identify issues before science teams report them. Blue Waters makes use of both off-the-shelf and custom software to centralize all monitoring information, and tools to allow efficient traversal of the data as well as automated alerting. This infrastructure has proven extremely useful for rapidly diagnosing active system issues. One of the more common is an application with a bad I/O strategy causing filesystem slowdowns that affect all users. The monitoring infrastructure can be used to determine, or at least narrow down, the offender, so that remedial action can be taken to restore the system for other users.

A. *Passive System Monitoring*

Traditional system monitoring begins with centralizing text logs from the system to a central point. That is a good first step, but even on small systems the volume of logs quickly becomes overwhelming, requiring automation in order to be useful. On Blue Waters, the Simple Event Correlator (SEC) software [18] is used to provide alerts based on log messages. SEC uses simple regular expressions with logic to roll-up events together to prevent alert floods. The logs are then further parsed and loaded into a database where all other types of monitoring data also flow, allowing cross correlation of events, batch jobs and performance data.

Performance data is also a challenge on many systems. Ideally, high-resolution data for the life of the system would allow detailed examination of the system's behavior at any point in the past. The size of that data poses challenges both in the data collection and in storage. On Blue Waters, the data collection challenge is solved through the use of the OVIS software [19]. OVIS makes use of the hardware features in the Cray to allow synchronized collection of over 150 metrics per node across the full system with no significant impact on application performance. Figure 2 demonstrates the value of high fidelity in the performance data as it compares the one-minute samples used on Blue Waters with five-minute averages. The higher frequency data clearly show a behavior that is completely missing in the five-minute averages. The combination of logs and performance data results in over 20 billion datums collected per day, posing a big data challenge! In fact, the performance metric data are too large for storage in high-speed flash storage for more than about a week. Beyond that, all data and logs are stored on spinning disk to allow for slower post processing. Storing the data over the long term allows detailed post-mortem analysis of system usage and system issues. As an example, a detailed study of the workload on Blue Waters [20] was produced looking at various aspects of application usage of the system, including memory and GPU usage, network usage and I/O characteristics.

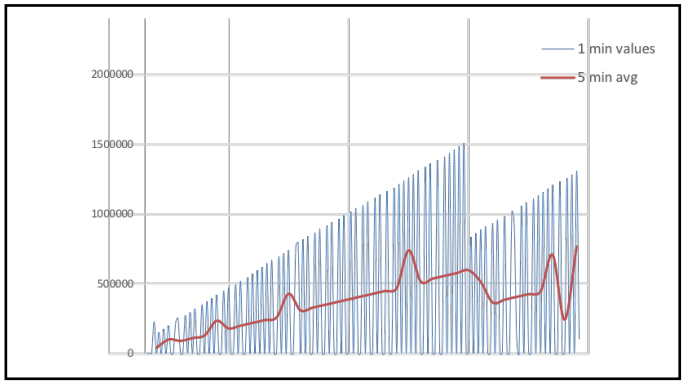

Fig. 2. The value of high-frequency monitoring.

B. *Active System Monitoring*

Active monitoring is rarely conducted on HPC systems, in part because many administrators are worried that the act of probing the system will impact application performance. However, active monitoring is an excellent practice both to identify issues before science teams report them and to provide the ability to perform post-mortem analysis in order to determine when an issue began. The goal of the testing is to measure the user experience at that point in time. Blue Waters utilizes multiple tools to perform active testing, including a Jenkins server [21], to drive many tests, and custom tools to measure items such as the file system responsiveness across the range of node types. The tests include functionality tests, application performance tests and general performance tests. Functionality tests include testing *ssh* connectivity to the login nodes, compiling an application, and other items. Many are run routinely and others only occasionally to verify functionality after a system change. Similarly, short performance tests, including application tests, are run routinely, while others, representing the full acceptance-test suite, are only run on demand. Figure 3 illustrates one such test, which performs a simple file listing of the *home* and *scratch* filesystems, on the login node and on a batch-system mom node, at regular intervals. Spikes indicate times when the performance dropped significantly and might be noticeable to science teams.

Finally, all failures and interrupts are logged and classified, including all out-of-service time. This data allows identifying both immediate issues, such as a spike in failures due to a specific component after a software change, as well as long-term trends. The data also directly support the determination of required service metrics. Long-term trends are important to determine if component failure rates are (or are likely to become) a serious issue. For example, is the rate of disk failures likely to increase to the point where a multiple disk failure could lead to data loss? These data, along with the other log data, are also very useful for research teams seeking to develop ways to predict failures in time to prevent an application failure.

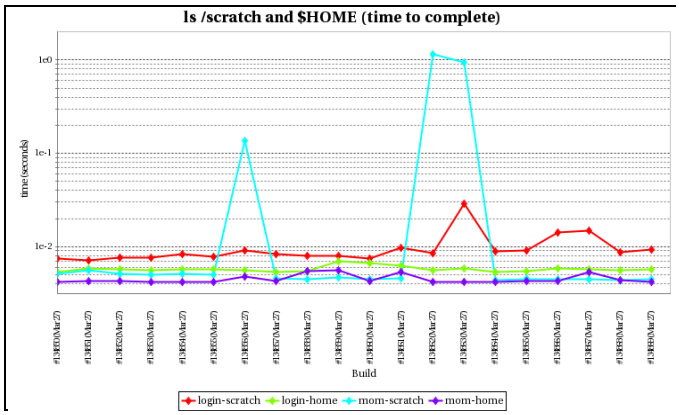

Fig. 3. Time to "ls" the *scratch* and *home* filesystems.

V. Visualization Support

In this section, we describe the considerations for providing visualization resources in an HPC environment. The large-scale science initiatives driving the deployment of supercomputing resources dictate the primary priority of HPC visualization staff: integrating visualization into domain-specific, at-scale scientific workflows. As such, visualization support centers around physically structured, scientific simulation data; efforts are therefore routinely in the application of scientific visualization methods. The sheer data sizes necessitate data-parallel techniques and have resulted in a specialized sub-community of scientific visualization researchers. Figure 4 shows various views of data generated early in Blue Waters' deployment that illustrate the inherent difficulty in large-scale visualization. In this case, the area of interest represents a significant data reduction, yet is still of a size that is not trivially viewable at its native resolution. The leftmost image in that figure is the full mesh of a single time step of an Adaptive Mesh Refinement dataset [22] [23]. The middle is a full rendering of the region of interest, with the far right a small sample of this region that is still adequate for high resolution imagery.

In the following subsections, we will first outline the practical requirements for deploying visualization support, and provide a representative example of what we consider a particularly successful collaboration. Finally, we will conclude with a discussion of some of the many ways the HPC environment offers additional opportunities for innovative applications of visualization.

A. Enabling HPC Visualization

Hardware: Bethel et al. describe the necessities and expectations of visualization resources at HPC centers [24]. In addition to staff, an HPC center must provide appropriate hardware for data analysis and visualization tasks. While much discussion was focused on the considerations of standing up such a resource, the authors acknowledged that data sizes would eventually make this model unsustainable. Without clear evidence that a smaller visualization cluster would be sufficient, the Blue Waters supercomputer was officially deployed for use as its own post-processing resource. To our knowledge, this differentiates Blue Waters from other machines of this era, and our experience points to the viability of the approach for future deployments. The benefits are obvious: the reduction of necessary data movement and the guarantee of the availability of appropriate resources. We believe adoption of this model is slow due to the perception of convenience regarding additional visualization resources. It is our experience, however, that visualization suites are tailor made for such an environment and differences in convenience are negligible.

Software: The integration of visualization in general relies heavily on supported software. The visualization software staples at HPC centers are those which are open-source, actively developed, and especially scalable. VisIt [25] and ParaView [26] are the standards for HPC environments, as in addition to the attributes listed, they offer client/server execution models, *in situ* or concurrent with the simulation libraries, and a complete interface for scripting batch operation. While such tools cover a great deal of the overall functionality critical to support, there are other tools, for example yt [27], that while less generalized, may offer significant benefit to researchers in a subset of science domains. Such software suites are unsurprising standards but are often slightly more difficult to maintain and significantly more difficult to install on a supercomputer over the typical separate visualization cluster. We therefore recommend coordinating installation and maintenance carefully among system engineers, the software developers, and visualization experts at other centers. Being aware of interdependencies among these groups is crucial for providing a roadmap of future functionality, as new releases of supported tools may be incompatible with current system software.

In accommodating the maximum amount of science workflows, it is necessary to further support legacy software, such as IDL [28], or serial utilities, such as ImageMagick [29]. We have undertaken the support of software requiring more onerous efforts as well. For typical data-parallel, large-scale data analyses, interactivity is rarely expected or pursued. In fact, in early investigations of allowing interactive visualizations directly on Blue Waters, we found this was not possible without changes to default system software [16]. There is nonetheless a contingent of scientists for whom typical visualizations are based on geometry and, therefore, it is possible to generate renderings at a speed conducive to interactivity. One such workflow utilizes VMD [30] to analyze molecular dynamics simulation data. In that case, not only is interactivity prioritized, but typical avenues, such as the available VNC, are not performant enough. We therefore invested significant effort in the support of a tool with excellent performance, NICE DCV [31], which was not developed for deployment on an HPC resource. We believe interactivity will remain important for certain workflows for the foreseeable future; the best practice is to incorporate enabling technologies as standards for deployment and support.

B. Case Study: Visualization of Climate Simulation Data

As stated above, our primary initiative is enabling integration of visualization into domain-specific workflows. Here we will summarize a collaboration that exemplifies multiple ways in which this may be accomplished. The

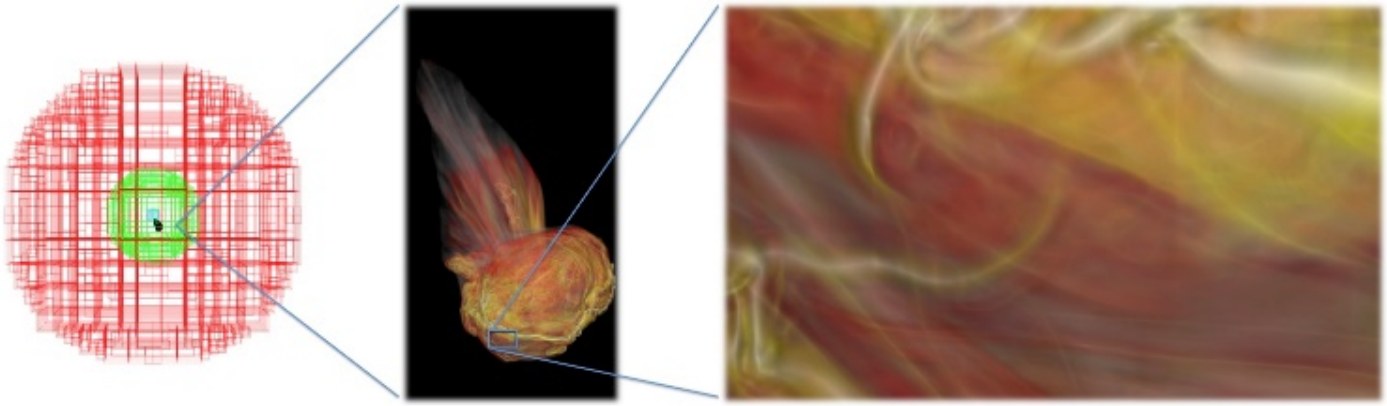

Fig. 4. An illustration of the disparity between HD image resolution and available data resolution. The extremely small highlighted portion in the middle image contains sufficient data to create an HD-quality image.

collaboration was initiated by a scientist needing help visualizing climate simulation data. Having put significant effort into developing a framework using temporal buffering to address I/O performance issues, the resulting dataset was no longer recognizable to our visualization tools. After initial efforts to create a custom reader plugin in VisIt, we collaborated on the creation of scientific visualization showcase outreach materials [32]. We direct the interested reader to those showcase materials, which include an in-depth description of the collaboration summarized above. While this represented a successful integration of visualization into a scientific workflow, we were responsible for much of the effort in generating the visualizations. However, through sustained support and consultation, the incorporation of visualization was natural enough that we were not involved at all in the creation of subsequent showcase materials [33]. For this type of integration of visualization, we believe this represents an optimal outcome; this is also the reason we selected this collaboration as our case study.

Collaborations in general do not adhere to any model; they range from single tasks for an afternoon to multi-year efforts. It is therefore helpful to categorize efforts and essential to review those ongoing. We, for example, track high-priorities as Jira issues and use tasks to track ongoing efforts, be they low-priority or in-depth collaborations. The visualization team routinely evaluates current tasks as does, on the order of monthly, the Blue Waters Project Office.

C. *Support Through Innovation*

Beyond supporting the various scientific domains utilizing a supercomputer, the collection of varied operations across the management of an HPC center offers further opportunities for innovation in visualization [34]. Such tasks may still appropriately be labeled as support but are of indirect benefit to those utilizing the supercomputer. For instance, we are far from exhausting the potential of the comprehensive diagnostic data generated and collected on Blue Waters. Our first effort was the creation of a new visual representation of job/utilization data that is routinely used in system monitoring [35]. Similarly, we have researched visual representations of I/O benchmark data to better document recommendations and best practices to the end user [36]. In both cases, the methods we developed were possible as a result of routine interdisciplinary staff meetings and a project-wide mentality of leveraging available expertise.

There are also many open problems general to HPC visualization for which progress benefits the wider community. For example, *in situ* processing is recognized as enabling technology, particularly for future extreme-scale architectures. In research and development of *in situ* middleware, we have shown it is possible, in certain circumstances, to generate visualizations for free *in situ* [37]. We have also researched a visualization algorithm with notorious load balancing issues, particle advection/streamline generation, and found that common default parameters lead to severely poor performance [38]. Such a result has clear implications in cost-effective and efficient resource usage. We encourage such work as a best practice and believe it reflects back on the project by exhibiting a desire to be good stewards of the HPC community.

VI. Petascale Application Improvement Discovery

Blue Waters has always recognized the need for a comprehensive view of high-performance computing. The rate of change of hardware technology has made it difficult for science domain specialists to keep their applications performant on today's systems. Multicore processors and many-core accelerators, the memory wall, network topologies, and parallel filesystems are some of the areas where experts in programming models, mapping and load balancing, scalable I/O, and data movement can come to the aid of domain specialists. The Blue Waters project implemented the Petascale Application Improvement and Discovery (PAID) program as a best practice to help domain-scientists partner with technology experts in a managed, result-guided way. In this section, we describe why programs like PAID are a good idea for the HPC community compared to more common, self-directed collaborative partnerships.

The first NSF-awarded Petascale Computing Resource Allocation (PRAC) teams on Blue Waters, selected on a bi-annual basis by an NSF review process, were eligible to participate in a self-directed program designed to provide one year of financial support for a graduate student or post-doc, to be used towards improvement of their codes. The results of the program were mixed. Teams with experienced

programmers in-house were able to benefit from the enhanced support, while teams with more limited expertise were less able to. The PAID program was developed with the lessons learned from this first program.

The PAID program formed teams of experts in areas determined by reviewing the results of the earlier program; such teams are referred to as Improvement Method Enabler (IME) teams. The five areas selected from that first-year opportunity were: task mapping and load balancing, scalable I/O and hierarchical data formats (HDF), Fourier transforms (FFTs), programming model best practices, and exploitation of graphics processing unit (GPU) accelerators. Eight IME teams from five different institutions were recruited.

A. PAID Program Management

Following project management best practices, each IME team developed a statement of work (SOW) that documented the expected number of PRAC teams they would work with, participation in workshops and other training events, and any resulting products from the program such as libraries or best practices. Each IME was assigned a POC within the project to facilitate tracking progress, reviewing milestones and deliverables, and addressing any support-related issues. The POC model as a best practice was important in ensuring good communication with the IME teams.

Participating PRAC teams were required to have an award end date that would allow for adequate time to benefit from improvements while still on Blue Waters. The teams were also required to have personnel available, such as graduate students or post-docs, to work with the IME team members. This requirement facilitated the transfer of knowledge from the IME team to the application team. In total, 15 PRAC teams from seven fields of science engaged with the eight IME teams; some application teams worked with up to three different IME teams.

Each PRAC–IME team paring developed a work plan and SOW with the assistance of a POC (usually the PRAC team POC from the SEAS team). SOWs were written to be flexible, and accommodate changes in the work plan, which resulted in less management effort later in the program. The POC for each science team was responsible for tracking progress, reviewing milestones and deliverables, and addressing any support-related issues. All SOWs had a required baseline performance as a first deliverable. This baseline would then be used to ascertain and quantify improvements in subsequent progress milestones and associated deliverables (usually reports). The best practice requirement of a performance baseline is essential for the ability to compute a Return-on-Investment for the program.

The PRAC and IME teams were required to report at monthly progress meetings. The project office and POCs would review progress, check due dates for milestones and deliverables, and adjust schedules as warranted. Tracking was managed via an integrated Wiki and ticket system that allowed for automated notification and collection of due milestones and deliverables, along with their eventual review and certification. Effective project management best practices were needed to handle the multitude of meetings, presentations, reports, reviews and certifications.

The result of each pairing is a measured level of performance improvement that clearly translates to improved science throughput. In some cases, baseline performance indicates a code that is not highly tuned or has not been ported to a new programming model, allowing for potentially substantial gains in performance. In some situations, such as for established community codes, baseline performance indicates that an application is already sufficiently optimized or tuned, and percentage performance improvements should be expected to be lower compared to the upside potential for a code being ported for the first time. Enabling new functionality should be considered as important as performance for the teams benefiting from the new methods but the concept of baseline performance is not well defined.

B. Topics and Results

Accelerator IME teams worked with six applications, porting them to OpenACC or CUDA, or tuning already existing implementations to make better use of the GPUs. As indicated earlier, a lower overall improvement is expected for codes already ported to the accelerator. Three CUDA-enabled codes, with multi-node baselines, showed a range of speedups after working with the teams, which included support from NVIDIA, who provided access to a dedicated engineer. Overall, speedups ranged from 1.01x to 1.6x. The range in speedups was due to differences in code coverage and un-ported portions of applications. IME teams enabled OpenACC in two applications, working with single-node versions of the codes. Overall, speedups ranged from 2.8x to 3.9x for single GPU node performance compared to single XE node. In one case, after consultation with IME teams, an application scientist decided to rewrite the code, resulting in a 2x overall speedup over its earlier, CUDA-accelerated version, using a single-node baseline. Multi-node functionality is expected for these applications with continued development.

The Blue Waters high-speed network has a 3D Torus topology, providing a scalable communication subsystem; however, effort is needed to avoid network congestion. The *libtopomapping* library [39] was developed by an IME team to automatically provide improved MPI task placements. The library collects the necessary information during a shorter instrumented run and then produces an MPI rank reorder file that works with the Cray MPI library to place the tasks optimally for on-node communication and for reduced off-node congestion. Speedups ranged from 1.2x to 2.2x for 65K MPI ranks when compared to default placement. Results are sensitive to the node geometry and typical geometries from production runs were used for the analysis. The realizable benefit from improved communication performance due to rank mapping is possible only with the investment made in topology-aware scheduling implemented on Blue Waters.

File I/O is often the last aspect of application performance to be addressed and it typically is not an issue until the scale of the code is increased. Two IMEs partnered with several teams to improve application I/O. An IME team developed the *meshio* scalable I/O library [40] to address I/O performance.

The *meshio* library improved I/O performance up to a factor of 20x, making I/O a minor contributor to runtime for the applications. The HDF-Group IME team addressed I/O issues in several applications, reducing the I/O time step by 9x at 32K MPI ranks. The team improved the scalability of I/O for another application from 2,048 ranks to 16,384 ranks with a 6x reduction in I/O time. Best practices for I/O are accessible to the community from improvements provided by the *meshio* library and improvements to HDF5.

In at least one case, unexpected issues arose that impacted the success of the program. The FFT IME team (SpiralGen) implemented an improved recursive k-way Alltoall algorithm for 3D FFTs in their ACCFFT library, with speedups of 2x to 4x, depending on node count. Unfortunately, multi-language support issues prevented use by the interested application teams. The library is available at the SpiralGen website [41].

The PAID program enabled new or additional science to be achieved by the improvements to the applications discussed above. Many of the application and IME teams have stated their support for this type of program, affirmed the success of the PAID program and recommend similar programs be available in the future [14]. The return-on-investment can be estimated from both the additional science made possible by improved code performance and the cost of the saved node-hours, which could be used by other projects running on the system. The application improvements should be viewed as having an impact on science on Blue Waters as well as having an impact on science on other HPC systems.

VII. Education and Outreach Initiatives

The annual Blue Waters Symposium is offered to build an extreme-scale community of practice among researchers, developers, educators, and practitioners. The unique value of this event is that more than 1/3 of the attendees are science team PIs and there is strong participation by graduate students and post-docs. We learned that active PI participation is best achieved with strong encouragement from NSF to attend. The Symposia provide an opportunity to define petascale and extreme scale requirements, identify improvements to resources and services, foster the exchange of challenges, opportunities and solutions across diverse fields of research, and recommend future directions. Participants reported that the individual sessions were very valuable and provided highly useful resources and information. They also commented that the strength of the symposium was the diversity of the topics and interaction opportunities [42].

A. Education Allocations

Education allocations are a valuable resource for programs engaging participants in learning about computational and data-enabled tools, methods, resources and applications to advance discovery. We receive requests for undergraduate and graduate courses, research experiences for undergraduates, internships, fellowships, training sessions, workshops, and summer schools. We have improved the allocation request procedures, for example, by streamlining the process for setting up accounts and developing secure procedures to offer training accounts to remote participants.

B. Education and Training Events

We offer a variety of training and education events, including webinars, workshops, tutorials, hackathons, and courses, to address diverse backgrounds, learning styles and needs. To serve a national constituency, we strive to make most events accessible via webcast and/or video conferencing to engage people from their home institution. Effective sessions provide a heavy emphasis on hands-on activities, assistance to participants during the hands-on sessions, Q&A among participants and the presenters, and recordings of sessions for subsequent access.

The Blue Waters Virtual School of Computational Science and Engineering pursued unique strategies to deliver HPC credit courses to campuses that would otherwise not be able to offer these topics to their students [43]. There is strong demand for these credit courses, and opportunities for scaling-up and sustaining these efforts.

C. Broadening Participation

Blue Waters supports undergraduate and graduate student engagement programs to prepare future generations of innovators and leaders. Mentors state that the intensive two-week parallel programming institute for undergraduates better prepares them for a research experience. Immersing undergraduates in a year-long research experience allows the students to go deeper into the research, which improves the research outcomes for the mentors. Graduate fellowships allow students to accelerate their research projects. Providing a Blue Waters staff member as a point of contact for each fellow has proven instrumental in advancing the research by quickly resolving technical challenges. The POCs noted that their presence would not only provide additional resources, but would be adding a face to Blue Waters. The virtual resources became personalized to the fellows because of the POCs. The faculty advisors reported that this program opens up new directions in their labs, provides new types of resources in their fields, and provided creative ideas regarding the ways in which to use massive computing resources in their fields [44].

To prepare a larger and more diverse workforce, emphasis is placed on pro-actively recruiting underrepresented individuals. The Broadening Participation program is engaging more diverse populations in petascale research through an allocations process focused on supporting underrepresented communities.

To address the preparation of the national HPC workforce, partnerships are essential. We are working with minority serving organizations (e.g. AIHEC, HACU, and NAFEO), XSEDE Campus Champions, women's groups (e.g. Women in HPC), Software Carpentry, Data Carpentry, ACM SIGHPC Education Chapter, the International HPC Training Consortium, XSEDE HPC partner organizations, and DOE HPC centers, to name but a few. These collaborations empower the organizations to share resources and materials, improve practices, and disseminate resources and opportunities for engagement to a larger community.

VIII. Return on Investment

While we do not yet have a thorough quantified financial analysis of the benefits for science and engineering that the Blue Waters project has enabled, we can provide concrete numbers for specific parts of the project.

As an example, the PAID program, described in Section VI, cost approximately 5.5 million dollars, mostly to support personnel in the Improvement Method Enabler (IME) teams. The improvements in application performance resulting from work by the IMEs can be quantified according to the reduced running time of the optimized applications. Based on the historical node-hour usage patterns of those applications, their baseline performance level prior to PAID, and their improved performance achieved after PAID, we can estimate the total savings in node-hours. Taking the node-hour cost charged to industrial users of Blue Waters, this corresponds to a total savings of 9.7 million dollars. Thus, those savings are nearly twice the cost of the investment in PAID.

The Topology-Aware Scheduler (TAS) project, described in Section III, has an enormous impact on the quantity of science Blue Waters is able to produce. Noting that the high-speed network represents the performance limiting factor for many parallel applications using Blue Waters, we used collected network metrics to estimate the impact on network-constrained applications. The results of the two six-month periods that were compared running each scheduling mode were striking. The TAS period workload, utilizing 14% fewer nodes on average, still transmitted 42% more bytes on average than workload placed by the traditional scheduler did. In addition, since that evaluation, TAS has been further feature-improved to utilize more nodes without compromising the performance gains resulting from original optimized placement. Assuming the 42% gain resembles increased science throughput of the machine, we can apply that to 4 of the 6 years of Blue Waters projected operation as a return on the whole of the project investment. Assuming a ~$400M total project investment, this would translate to $112M of value in additional science.

Another example is the economic impact of the Blue Waters project upon the state of Illinois' economy. According to a recently published study [45], Blue Waters has a projected $1.08 billion direct economic impact on Illinois' economy and will have created 5,772 full-time equivalent employment (FTE) over the project's lifespan (October 2007 - June 2019). That impact includes $487,143,813 in labor income from 5,772 FTEs, $56,477,093 in state and local taxes, $122,813,903 in federal taxes, and a $227,300,000 impact resulting from research grants awarded from granting agencies to Illinois researchers, faculty, and students because they had access to conduct research on Blue Waters.

That study does not include additional economic and societal benefits coming from the significant amount of computer time provided to Illinois researchers, strategic projects, and industry, nor does it account for the workforce development activities of the Blue Waters project, or the impacts of the science, engineering and research results that can only be accomplished on Blue Waters.

IX. Conclusion

Deploying extreme-scale systems for advancing science and technology requires a significant investment in hardware, software and professional staff. The recent National Academies Report [46] recommends achieving four broad goals: (1) positioning the United States for continued leadership in science and engineering, (2) ensuring that resources meet community needs, (3) aiding the scientific community in keeping up with the revolution in computing, and (4) sustaining the infrastructure for advanced computing.

These advanced-capability HPC systems are crucial for the US to remain competitive in a global scenario. Since acquisition costs for these systems are typically non-negligible, one must ensure that the resources and services they provide are highly productive and that their users are empowered to fully achieve their research goals.

As the first machine to offer sustained-petascale performance to open science in the US, Blue Waters presented an environment with many pioneering features that had to be managed and improved. Its combination of petascale compute capability, petascale storage and high-speed external network links presents a well-balanced system that serves applications from a diverse mix of domains. At the same time, this complex environment presents considerable challenges in terms of management and coordinated, reliable operation.

To face these challenges, NCSA employs best practices, as described in this paper, such that operation of Blue Waters proceeds smoothly and has garnered strong positive reviews from NSF panels. The paper includes lessons learned from five years of machine operation, describing our actions to ensure that the system remains productive for users.

In practical terms, the legacy of an extreme-scale system can be evaluated by the amount of new science and engineering that it has enabled. The numerous documented discoveries achieved by scientists using Blue Waters [14] show that the system delivered on its promise of providing a productive computational engine for the national community.

While Blue Waters is pioneering petascale-computing best practices, the positive community feedback from our presentations shows these practices are applicable to a wide range of HPC centers. Given the scarcity of literature on how to effectively manage these systems, we expect that our experiences, presented in this paper and in a complementary paper more specific to large Cray machines [47], will help guide the deployment and operation of HPC centers well into the future, and raise awareness in the community to the importance of a thorough approach to system management and customer service, as well as sharing of lessons learned.

Acknowledgment

We thank all members of the Blue Waters project for their efforts and dedication. This work is part of the Blue Waters sustained-petascale computing project, which is sup-ported 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] W. Kramer, "Top500 versus sustained serformance – or the top ten problems with the Top500 list – and what to do about them," in *21ST International Conference On Parallel Architectures And Compilation Techniques (PACT12)*, Minneapolis, 2012.

[2] ORNL, "Titan Specification," [Online]. Available: https://www.olcf.ornl.gov/olcf-resources/compute-systems/titan/. [Accessed 14 August 2018].

[3] G. H. Bauer, T. Hoefler, W. T. Kramer and R. A. Fiedler, "Analyses and modeling of applications used to demonstrate sustained petascale performance on Blue Waters," in *Proceedings of Cray User Group Meeting (CUG-2012)*, Stuttgart, 2012.

[4] G. Bauer, V. Anisimov, G. Arnold, B. Bode, R. Brunner, T. Cortese, R. Haas, A. Kot, W. Kramer, J. Kwack, J. Li, C. Mendes, R. Mokos and C. Steffen, "Updating the SPP benchmark suite for extreme-scale systems," in *Proceedings of Cry User Group Meeting (CUG-2017)*, Redmond, WA, 2017.

[5] 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*, J. Vetter, Ed., Sitka Publications, 2012.

[6] W. Kramer, M. Butler, G. H. Bauer, K. Chadalavada and C. L. Mendes, "National Center for Supercomputing Applications," in *High Performance Parallel I/O*, Prabhat and Q. Koziol, Eds., Boca Raton, Florida: CRC Publications, Taylor and Francis Group, 2015.

[7] 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 Cray User Group Meeting (CUG-2014)*, Lugano, Switzerland, 2014.

[8] "Atlassian Jira," Atlassian, 2018. [Online]. Available: https://www.atlassian.com/software/jira. [Accessed 14 August 2018].

[9] NCSA, "NCSA Risk Register," 2015. [Online]. Available: https://wiki.ncsa.illinois.edu/display/ITS/NCSA+Risk+Register. [Accessed 14 August 2018].

[10] "Blue Waters Portal," NCSA/University of Illinois, 2018. [Online]. Available: https://bluewaters.ncsa.illinois.edu/. [Accessed 14 August 2018].

[11] L. DeStefano and J. S. Sung, "Blue Waters Fellows Program Focus Group Report," Champaign, IL, 2015.

[12] L. DeStefano and J. S. Sung, "Blue Waters Fellows Program Focus Group Report," Champaign, IL, 2016.

[13] F. Foertter, "Overview of the OLCF," 2013. [Online]. Available: https://www.olcf.ornl.gov/wp-content/uploads/2013/02/OLCF_Overview_lowres-FF.pdf. [Accessed 14 August 2018].

[14] W. Kramer, "Sustained Petascale In Action: Enabling Transformative Research - 2017 Annual Report," 2017. [Online]. Available: https://bluewaters.ncsa.illinois.edu/liferay-content/document-library/BW_AR_2017.pdf. [Accessed 14 August 2018].

[15] G. Bauer, C. Mendes, W. Kramer and R. Fiedler, "Expanding Blue Waters with improved acceleration capability," in *Proceedings of Cray User Group Meeting (CUG-2014)*, Lugano, Switzerland, 2014.

[16] M. D. Klein and J. E. Stone, "Unlocking the full potential of the Cray XK7 accelerator," in *Proceedings of Cray User Group Meeting (CUG-2014)*, Lugano, Switzerland, 2014.

[17] A. Loftus, "Psync - parallel synchronization of multi-pebibyte file systems," in *Proceedings of Cray User Group Meeting (CUG-2016)*, London, England, 2016.

[18] R. Vaarandi, "SEC - Simple Event Correlator," 2017. [Online]. Available: https://simple-evcorr.github.io. [Accessed 14 August 2018].

[19] 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.

[20] M. Jones, J. White, M. Innus, M. DeLeon, A. Simakov, P. J. S. Gallo, T. Furlani, M. Showerman, R. Brunner, A. Kot, G. Bauer, B. Bode, J. Enos and W. Kramer, "Final Report Workload Analysis of Blue Waters (ACI 1650758)," 2017.

[21] "Jenkins Automation Server," [Online]. Available: https://jenkins.io/. [Accessed 14 August 2018].

[22] A. S. Almgren, V. E. Beckner, J. B. Bell, M. S. Day, L. H. Howell, M. J. Joggerst, A. Nonaka, M. Singer and M. Zingale, "CASTRO: A new compressible astrophysical solver. I. Hydrodynamics and self-gravity," *The Astrophysical Journal,* vol. 715, no. 2, p. 1221, 2010.

[23] A. Nonaka, A. S. Almgren, J. B. Bell, M. J. Lijewski, C. M. Malone and M. Zingale, "MAESTRO: An adaptive low Mach number hydrodynamics algorithm for stellar flows," *The Astrophysical Journal Supplement Series,* vol. 188, no. 2, p. 358, 2010.

[24] E. W. Bethel, J. Van Rosendale, D. Southard, K. Gaither, H. Childs, E. Brugger and S. Ahern, "Visualization at supercomputing centers: the tale of little big iron and the three skinny guys," *IEEE Computer Graphics and Applications,* vol. 31, no. 1, pp. 90-95, 2011.

[25] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pumgire, K. Biagas, M. Miller, C. Harrison, P. Navrátil, G. W. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E. W. Bethel, O. Ruebel, M. Durant, J. M. Favre and O. Rübel, "VisIt: an end-user tool for visualizing and analyzing very large data," in *High Performance Visualization--Enabling Extreme-Scale Scientific Insight*, 2012, pp. 357-372.

[26] J. Ahrens, B. Geveci and C. Law, "ParaView: an end-user tool for large-data visualization," in *Visualization Handbook*, C. D. Hansen and C. R. Johnson, Eds., Burlington, Butterworth-Heinemann, 2005, pp. 717 - 731.

[27] M. J. Turk, B. D. Smith, J. S. Oishi, S. Skory, S. W. Skillman, T. Abel and M. L. Norman, "yt: A multi-code analysis toolkit for astrophysical simulation data," *The Astrophysical Journal Supplement,* vol. 192, no. 1, p. 9, 2011.

[28] "IDL," Harris Geospatial Solutions, [Online]. Available: http://www.harrisgeospatial.com/SoftwareTechnology/IDL.aspx. [Accessed 14 August 2018].

[29] "ImageMagick," [Online]. Available: https://www.imagemagick.org. [Accessed 14 August 2018].

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

[31] "NICE DCV," NICE, [Online]. Available: https://www.nice-software.com/products/dcv. [Accessed 14 August 2018].

[32] R. Sisneros, L. Orf and G. Bryan, "Ultra-high resolution simulation of a downburst-producing thunderstorm," in *Proceedings of Supercomputing*, Denver, 2013.

[33] L. Orf, R. Wilhelmson and L. Wicker, " Visualization of a simulated long-track EF5 tornado embedded within a supercell thunderstorm," *Parallel Computing,* vol. 55, pp. 28-34, 2016.

[34] R. Sisneros, "Visualizing the big (and large) data from an HPC resource," *In Numerical Modeling of Space Plasma Flows ASTRONUM-2014,* vol. 498, p. 240, 2015.

[35] R. Sisneros, J. Fullop, B. D. Semeraro and G. Bauer, "Ribbons: enabling the effective use of HPC utilization data for system support staff," in *EuroVis Workshop on Visual Analytics*, Swansea, Wales, 2014.

[36] R. Sisneros, "A classification of parallel I/O toward demystifying HPC I/O best practices," in *Proceedings of Cray User Group Meeting (CUG-2016)*, London, England, 2016.

[37] M. Dorier, R. Sisneros, T. Peterka, G. Antoniu and D. Semeraro, "Damaris/viz: a nonintrusive, adaptable and user-friendly in situ visualization framework," *Large-Scale Data Analysis and Visualization (LDAV), 2013 IEEE Symposium on,* pp. 67-75, 2013.

[38] R. Sisneros and D. Pugmire, "Tuned to terrible: a study of parallel particle advection state of the practice," in *IEEE International Parallel and Distributed Processing Symposium Workshops*, Chicago, 2016.

[39] J. J. Galvez, N. Jain and L. V. Kale, "Automatic topology mapping of diverse large-scale parallel applications," in *Proceedings of the International Conference on Supercomputing*, Chicago, 2017.

[40] E. Karrels, "Mesh_IO - parallel IO for mesh-structured data," [Online]. Available: https://github.com/oshkosher/meshio. [Accessed 14 August 2018].

[41] F. Franchetti, "SPIRAL - Software/Hardware Generation for DSP Algorithms," [Online]. Available: http://www.spiral.net/. [Accessed 14 August 2018].

[42] L. DeStefano and J. Sung, "Blue Waters Symposium for Petascale Computing and Beyond Report," in *Blue Waters Symposium*, Sunriver, OR, 2017.

[43] K. Cahill, S. Lathrop and S. I. Gordon, "Building a community of practice to prepare the HPC workforce," in *International Conference on Computational Science*, Zurich, 2017.

[44] L. DeStefano and J. Sung, "Blue Waters Fellows Program: Third Cadre Report," Champaign, IL, 2017.

[45] Z. Chen, "The impact of Blue Waters on the economy of Illinois," 2017. [Online]. Available: http://www.ncsa.illinois.edu/assets/pdf/about/bw_economic_impact.pdf. [Accessed 14 August 2018].

[46] W. Gropp and R. Harrison, "Future Directions for NSF Advanced Computing Infrastructure to Support U.S. Science and Engineering in 2017-2020," The National Academies Press, 2016.

[47] S. Lathrop, C. Mendes, J. Enos, B. Bode, G. Bauer, R. Sisneros and W. Kramer, "Best practices for management and operation of large HPC installations," in *Proceedings of Cray User Group Meeting (CUG-2018)*, Stockholm, Sweeden, 2018.