Moab and Torque Achieve High Utilization on Flagship NERSC XT4 System

Michael A. Jackson, *Cluster Resources, Inc.*

ABSTRACT: *Moab and TORQUE keep NERSC's Cray XT4 "Franklin" humming with ultra high utilization, high availability and rich policy controls. This document will use the NERSC leadership class deployment as a case study to highlight the features and capabilities implemented by NERSC to achieve consistent utilization in the high 90s and improve manageability and usability of the Cray XT4 system leveraging ALPS on compute node Linux (CNL). Moab integrates Cray's monitoring and management toolset with its scheduling and reservation engine for a holistically optimized solution.*

KEYWORDS: Moab, Torque, XT4, batch, scheduling, workload management, Cluster Resources, High Utilization

1. Introduction

This paper discusses the advantages NERSC achieved in using a Moab/Torque solution for the batch system, and workload management on their Flagship Cray XT4 system. The motivation for such a solution is discussed, along with design principles and examples of policy choices and the benefits of these choices. Ultimately NERSC applied Moab and TORQUE to achieve the highest levels of system efficiency and maximize their flexibility over management, while reducing administrative burdens and improving user simplicity.

NERSC[1]

The National Energy Research Scientific Computing Center (NERSC) is the flagship scientific computing facility for the Office of Science in the U.S. Department of Energy. NERSC is a world leader in accelerating scientific discovery through computation.

More than 2000 computational scientists use NERSC to perform basic scientific research across a wide range of disciplines, including climate modeling, research into new materials, simulations of the early universe, analysis of data from high energy physics experiments, investigations of protein structure, and a host of other scientific endeavors.

While NERSC provides some of the largest computing and storage systems available anywhere, what distinguishes NERSC is its success in creating an environment that makes these resources effective for scientific research. NERSC systems are reliable, secure, and provide a state-of-the-art scientific development environment with the tools needed by the diverse community of NERSC users.

Cluster Resources, Inc.

Cluster Resources, Inc. is a leading provider of workload and resource management software and services for cluster, grid, and utility-based computing environments. As the developers of the popular Maui Scheduler and the next generation Moab Cluster Suite®, Moab Grid Suite®, and other associated products, Cluster Resources is recognized as a leader in innovation and return on investment. With well over 5,000 clients worldwide, and drawing upon over a decade of industry experience, Cluster Resources delivers the software products and services that enable an organization to understand, control, and fully optimize compute resources.

[1] The text describing NERSC is an edited version of NERSC's about description found at http://www.nersc.gov/about/

Moab

Moab Cluster Suite is a professional cluster management solution that integrates scheduling, managing, monitoring and reporting of cluster workloads. Moab simplifies and unifies management across one or multiple hardware, operating system, storage, network, license and resource manager environments to increase cluster invstment ROI. Its task-oriented graphical management and flexible policy capabilities provide an intelligent management layer that guarantees service levels, speeds job processing, and easily accommodates additional resources.

TORQUE Resource Manager

TORQUE[2] is an open source resource manager providing control over batch jobs and distributed compute nodes. It is a community effort based on the original PBS project, and with more than 1,200 patches, it has incorporated significant advances in the areas of scalability, fault tolerance, and feature extensions contributed by NCSA, OSC, USC , the U.S. Dept of Energy, Sandia, PNNL, U of Buffalo, TeraGrid, and many other leading edge HPC organizations. This version may be freely modified and redistributed subject to the constraints of the included license.

2. The Motivation

When an organization invests in a Cray XT4 system, it represents a serious and carefully considered investment. Cray XT4 systems attract sophisticated scientific High Performance Computing (HPC) customers with complex technical requirements. NERSC's approach was no exception to this carefully considered investment. As a result, NERSC has realized much success in their effort to achieve ultra high utilization rates and high levels or management flexibility while also reducing administrative burdens and improving simplicity for their users.

The following list illustrates examples of capability sets that were important to NERSC from a workload management perspective:

Interoperability with Cray's CNL and ALPS technologies so that NERSC could gain the benefits of improved OS and workload execution intelligence and flexibility.

Ultra High Efficiency Scheduling and Policies to help obtain 90 to 99% utilization, with NERSC now commonly achieving 97+% utilization, despite having many extra-large jobs included in their workload mix.

Dynamically Modifiable Limits, Policies and Management of virtually every parameter to allow flexible and easy-to-administer control over resource usage.

Intelligent and Hierarchically Inheritable Policies that reduce management burdens by allowing changes in one location to apply to multiple areas, as well as reduce the knowledge requirement for end-users to efficiently and effectively use the resources.

Advance Reservations that help guarantee resources for critical uses, while intelligently relinquishing resources when not needed.

TORQUE Resource Manager High Availability to ensure added reliability of computing services to end-users.

Holistic Reporting and Integration with Custom and Open Source Accounting Tools to ensure proper accounting and fair allocations of resources.

Support for Heterogeneous Resources so that old and new resources can be combined as well as to prepare for mixed architectures and customization of resources to optimize specific applications.

Grid Support so that all resources can be grid ready for improved sharing and utilization purposes.

Rich Configuration Options so that the environment can behave as desired (e.g. preemption, broad credential choices [users, groups, classes, accounts, QoSs, etc.], fairshare, backfill, etc.).

[2] This product includes software developed by NASA Ames Research Center, Lawrence Livermore National Laboratory, and Veridian Information Solutions, Inc. Visit www.OpenPBS.org for OpenPBS software support, products, and information. TORQUE is neither endorsed by nor affiliated with Altair Grid Solutions, Inc.

3. The Design Principles

From the start it became quite clear that NERSC was designing a system that was highly flexible, easy to manage once it was in place and would result in the users needing even less knowledge to be efficient and effective in their use of the system.

Moab has a broad set of building blocks that NERSC was able to put in place and effectively use to get the desired result. Moab uses five credential sets: users, groups, classes/queues, accounts and QoSs; Moab also uses a highly robust advance reservation capability.

NERSC set up four classes of users: (1) standard user, (2) special user, (3) staff members, and (4) administrators.

In TORQUE Resource Manager, NERSC set up **queues** for regular workload, low priority workload, high priority workload, special workload, interactive workload, debug workload, administrative workload and data transfer workload. It is important to understand why they were set up as routing queues as this is a key to gaining the flexibility NERSC desired while maintaining the user simplicity.

For example the regular workload routing queue was then divided into **multiple sub queues** within Moab, where **based on the size of the workload** it was routed to the size-based sub queues (such as regular_small, regular_large, regular_extralarge, etc). Within Moab, requiring no added work by the user, the jobs were assigned to their proper sub queue, where additional policies were applied that matched the desired priority level and run time for that size of workload. In general, **larger jobs were favored** so that smaller jobs did not continually push the larger workload out of the way. It is notable that for NERSC a small job may actually run on more than 2000 cores, and due to the backfill policy they put in place small jobs will get backfilled around the higher priority large jobs, which helps further balance the workload.

This same organization can be applied to any other workload type to allow end-users to simply identify the workload as regular or special, etc. and then for Moab to apply the intelligence that associates the additional rules for the specific use case.

When NERSC created their **Transfer Queue**, they created a queue that did not allow any tasks, had a very high priority level, and a fairly low maximum duration. The zero task attribute ensured that while users could log in with this queue and transfer data from one point to another, the zero tasks disallowed computational work, thus preventing abuse of the ultra high priority setting. The very high priority setting helped ensure that data movement requests could be satisfied very quickly and the shorter duration period also was designed to be appropriate for the transfer of even very large files. By using this model, accounting is automatically applied by Moab to their use of the service, thus helping NERSC monitor the situation and have the details combined with their regular workload details from an accounting and reporting perspective.

Another important design element is the use of **Quality of Service (QoS) containers that were able to easily apply inheritable rules** to multiple other credentials. For example, all regular workload for each of the queues, regardless of size could now inherit a modification in priority or run time rule, or many other policies by simply changing the attribute found in the QoS that applied to all of the specific queues. NERSC described their need as hierarchical queue complexes and then fulfilled it with a mix of routing queues and sub queues that auto applied workload into the right category and rule set in conjunction with the QoSs that allowed quick modification of groups of use cases rather than having to apply a separate definition for each scenario.

NERSC's selected structure helped apply a highly flexible and easily modifiable policy set that was a far better design than applying the alternative of what could have expanded into hundreds of complex rule sets. Some other resource manager solutions would not have been able to even provide the options in the first place.

When applying **prioritization** weights, NERSC **favored simplicity of communication** to its end-users. In Moab, a prioritization value of 1 means that it has a one minute advantage over workload with priority 0, thus a priority of 60 will have a one hour advantage and 1440 (60 * 24) has a one day advantage. NERSC applied this 1440 value to multiple credentials as their default and then applied multipliers at the queues, QoSs, and other layers. Therefore users that saw the multipliers were able to quickly tell that the multiplier meant the number of days of added advantage to getting their work started.

While the following is not NERSC's specific settings, the example illustrates what they put in place and how simple their model is. If a low priority queue was set at 0, a regular priority queue at 2, and a high priority at 5, this would mean if a low priority job was put in place yesterday, that a regular job would still have an added day of priority over the low priority workload. Similarly a high priority job would run in front of a regular job that had been in the queue for less than three days (5 – 2 = 3) and a low priority job that had been in the queue for less than five days (5 – 0 = 5).

Accounts and Groups were also set up to help track usage per project and organization as accountability is an important aspect of their computing environment. At this point, no critical modifiers are put in place on these credentials, but it is configured to allow quick changes to these areas with minimal effort.

NERSC applied the Moab **Periodic Standing Reservations** capability to provide guarantees of service and to optimize use of the system. They knew that interactive sessions, data transfer requests, and debugging activities needed to have guaranteed quick responsiveness, they needed to make sure that other jobs would not fill up the system and block these activities, and they knew that these activities would be initiated during regular business hours.

Periodic Standing Reservations were a perfect match. Moab's Periodic Standing Reservations allowed NERSC to have a reservation that guaranteed that a minimum number of resources would be preserved for these activities during regular business hours and that after business hours the guarantee terminated and the resources could be effectively used by other workload. After the reservation is created, which specifies the Monday to Friday period and the hour range, the administrator simply needs to associate which queues are able to submit to the reservation.

The **reservation self cancels** during the evenings and weekends and then re-initiates during the next business day. End-users don't have to do anything to take advantage of this guarantee; the benefit is simply inherited as part of the logic put in place, once again extending benefits while maintaining end-user simplicity. The site gains the desired guarantee for the targeted service and makes optimal usage of resources during non-business hours. Moab, by design is filled with intelligent policies that help the system behave in a way that matches the organization's objectives rather than forcing the organization to change how it operates to meet the limitations of the underlying resource manager.

4. Required Attributes

NERSC has had experience with multiple clustering and supercomputing platforms. As a result of this experience, NERSC applied multiple required attributes to their purchase of their new leadership class Cray system. Some of these required attributes were the ability to demonstrate very high utilization rates, high availability, excellent manageability and flexibility of control, grid capabilities, and the ability to help deliver on ease of use requirements to their users.

Ultra High Node Utilization

Any purchase of a cluster or supercomputer requires focus on getting the results for which the system was commissioned. A strong indicator of effectively getting results out of the investment is how highly utilized the system is and that the usage is targeted to the projects, users, and organizations that are of most importance. Statistics and reports are also important to provide evidence of delivered performance and utilization.

NERSC has a **mix of workload** that is small, large, extra large, and extra extra large, with NERSC requiring the **ability to ultimately submit one job that runs across all 19,000+ cores**. Some **resource managers** would ultimately not be able to deliver on the ability to have single jobs run across the entire system as they typically **break the system into pieces with their queue models, creating silos**, where Moab is able to apply the benefits of queues and other such organizational containers while maintaining the option of submitting jobs that fully utilize all resources.

Limitations of **resource manager centric queue only based systems also create inefficiencies with each new queue that is added.** In essence it makes the resource become increasingly fractured. Also as resource managers do not commonly have periodic standing reservations they also cause inefficiency as they try to resolve a guarantee with a queue that is not aware of time of day or time of week commitment limitations. As you enforce a guarantee with a queue that does not apply time of day and week into consideration, that leaves the queue blocking resources for the low use periods, thus adding significant inefficiency.

Another added value that NERSC applied was the use of **Maintenance Reservations**. If an organization uses the traditional resource manager model of draining a system for maintenance they are apt to find that there are long periods—up to a day or more—that no new workload is being applied to the system, even though the work could have started and finished within the window of time that the system was being drained. NERSC, however, is able to apply Moab reservations to their maintenance periods so that the system is used by all workload that can be accomplished prior to the actual maintenance period. This **also allows maintenance to be applied to a subsection of the environment** with proper effect.

When any site applies a maintenance reservation, they **don't need to ask users to stop submitting workload and to let the system drain**. In fact, Moab will try to pack in as much workload as can be accomplished up to the reserved period, and if the reservation is only on part of the system that needs to be upgraded or serviced, Moab will apply workload properly to all other areas and only make the reserved resources unavailable for workload. Also, rather than sending an email and asking users to begin working again, organizations can simply end the reservation when the service work is complete and then Moab will automatically start to apply workload to the newly available resources. This enables NERSC to send fewer maintenance centric emails to users, thus keeping their experience more simplified, and as previously stated, it helps achieve higher utilization due to less disruptive maintenance experiences.

NERSC also required a few other capabilities that exert even higher efficiency. For one, they required **checkpoint restart** capabilities. NERSC has not completed implementation of this function as of the time of this paper, but expects to do so upon availability. This functionality will allow NERSC to run all jobs right up the edge of maintenance reservations and simply checkpoint them and continue them when the reservation is over, rather than just allowing jobs that are able to finish prior to the maintenance period. Also, very long jobs are easier to manage for end-users as they can run for the maximum duration allowed by administrators, checkpoint the job at the end, and then resume the job when they are allotted additional time. This **protects very long jobs** from being lost due to unexpected interruptions of their jobs and allows such users to not have to figure out how to break their jobs up into smaller pieces to fit within administrative constraints.

NERSC also has implemented Moab's **dynamic backfill** capabilities. While some resource managers have backfill, they commonly have much more static backfill that makes a **static** backfill decision and then enforces it, even if over time their are more optimal options to slide the work even further forward because a job finishes early or another job is cancelled, etc. Moab re-optimizes its backfill dynamically each iteration, thus taking advantage of each new available resource to get work accomplished faster. In particular, NERSC has applied Moab's **first fit** backfill policy. This backfill policy applies fairness and prioritization more heavily than other backfill policies by taking the first job in the priority list that fits the available space.

Moab also provides a backfill policy known as **best fit** that considers all jobs that can run on the available resources and selects the largest single job that can fit within the available window. Finally, Moab provides a backfill policy known as **greedy** that looks at all possible combinations of single or multiple jobs that can run on the available resources and selects the combination that will most fully utilize available resources. Therefore organizations can achieve even higher utilization on this factor than NERSC, though it would be at the expense of enforcement of prioritization and fairness. This is simply a judgment call on a per organizational basis as to what is most important.

Effective use of **dynamic backfill is increasingly important the larger your jobs are** in terms of number of resources used. In NERSC's case, the inefficiency shadow of large, extra large and extra extra large jobs could be very impactful to utilization, but is being very well handled with their implementation of dynamic backfill.

While additional node utilization enhancing policies are available and NERSC has implemented a number of them, it is clear that NERSC has considered utilization as a critical factor and has implemented policies that optimize their investment. As a result, NERSC has described their node utilization as being sustained in the high ninetieth percentile range, achieving above ninety-nine percent at times and **commonly maintaining ninety-seven percent or higher**.

NERSC also uses Moab to effectively monitor and report on node utilization, usage, and other factors. This aspect will be covered more in the accounting and reporting section.

High Availability

Availability and reliability are important to any organization, though the larger your investment the more impactful such attributes can be. Moab has had high availability for multiple years and is able to quickly meet this required attribute. However at the beginning of the engagement, TORQUE Resource Manager only had workload high availability and not server level high availability. TORQUE is a free PBS-based resource manager that is maintained by Cluster Resources and used by many thousands of sites worldwide. Cluster Resources committed to and delivered the server level high availability within TORQUE during the acceptance period and was able to deliver **Active-Active-Active High Availability** as opposed to high availability that was limited to just one stand by node. Ultimately a configurable number of nodes can be added into the high availability pool helping to deliver on NERSC's high availability requirements.

While described in the utilization section of this document, **checkpoint restart** is another important availability/reliability feature to protect long running workload.

Moab also provides **intelligent resource allocation** so that it applies workload away from fully or partially failed resources or environments. In previous papers Cluster Resources was able to describe working around Lustre clean up cycles, avoiding failed nodes, etc. Incidentally, a new Moab feature is the ability to avoid overheating nodes.

Other Attributes

As mentioned in section two of this document, NERSC also required **highly flexible management** that can be dynamically changed. While not all attributes of Moab can be dynamically changed, it is far more versatile and dynamic than any comparable solution. Important aspects of the required flexibility are provided in the product itself with the compliment being NERSC's implemented design that allows them to apply changes easily and quickly. Due to the inheritable aspects of the design, single configurations can cause the desired effect across large portions of the environment without repeated duplication of the change.

Ease of use and administration were achieved through the flexibility and inheritable policies mentioned earlier, as well as the routing and other intelligence that gave the benefits with no additional user intervention or knowledge required. Beyond those items already mentioned, Moab's workload management provides organizations with a **web-based job submission portal** and a **graphical administrative/help desk and reporting tool**. These ease of use tools allow users and administrators to focus on what they want to accomplish rather than trying to learn and access the system through a command line interface only.

Organizations will also find **workload templates** allow common inputs to be automated and shared from a job submission perspective, and other templates allow administrators to add defaults to many use case scenarios to help apply best practices in a more automated manner.

Grid support was important to NERSC and NERSC is currently reviewing unique Moab grid capabilities to ensure their environment continues to be grid ready. NERSC is reviewing Moab as part of a Globus based grid scenario, though ultimately NERSC is receiving a grid capable product that can match together mixed resource types, mixed OSs, mixed resource manager types, and can be an effective tool to mitigate political issues, helping organizations to share yet retain sovereign control over their resources.

Preemption is another Moab feature set that NERSC is evaluating. Preemption allows more time critical workload to push lower priority workload out of the way temporarily. For example an organization could respond to an emergency to evaluate a natural or national disaster (for example estimate the impact of a tsunami or earthquake) and lower priority workload that is specified as preemptible can be automatically checkpointed, then the higher priority workload runs, and upon completion the lower priority workload is resumed.

While not on NERSC's original requirements list, Moab is a valuable investment towards applying Cray's **Adaptive Supercomputing** vision. Moab already has the ability to support heterogeneous resources and to fully understand dependencies, best fit uses, affinities, adaptation of environments, and many other factors that allow all compute environments to be virtualized into a shared pool that is intelligently applied to what is of most importance to an organization.

Accounting and Reporting

NERSC, as a shared resource requires a high level of tracking, accounting, and reporting. NERSC maintains an accounting and allocation manager of their own design called NERSC Information Manager (NIM). **NIM** leverages TORQUE Resource Manager's accounting capabilities to ensure the proper tracking and enforcement of usage of their system is applied. TORQUE tracks and reports what is used and NIM maintains what the user or project has left to use.

Each day the system evaluates what was originally allocated to users or projects and then debits against that account what they have used to date. It then determines if the user or project is able to continue to submit workload or if the entity has used up all of its available resources and must wait for the next accounting period.

As a shared resource servicing multiple groups, such accounting becomes quite important. For organizations that do not yet have their own allocation and accounting system they may want to **consider Gold Allocation Manager** (See: http://www.clusterresources.com/pages/products/gold-allocation-manager.php). Gold Allocation Manager is an open source accounting and allocation manager developed by PNNL as part of the Department of Energy – Scalable System Software Initiative. It tracks resource usage on high performance computers and acts much like a bank, establishing accounts in order to pre-allocate user and project resource usage over specific nodes and timeframe. Gold provides balance and usage feedback to users, managers, and system administrators. Gold also enforces the allocations in real-time so that users, projects, and accounts use specifically what was granted. Moab can also apply blocking policies once allocations are used up, or it can simply reduce their priority by forcing all of their remaining workload that is beyond their initial allocation into a queue with a lower quality of service.

6. Conclusion

NERSC has effectively applied advanced scheduling and workload management capabilities to achieve a highly efficient, flexible, and usable system. While important aspects of this achievement came from capabilities unique to Moab and Torque, others came from effective design implementation. NERSC demonstrated their knowledge in their requirements documents early on in the acquisition and have effectively worked with Cray and Cluster Resources to achieve an admirable result in this area of management.

Acknowledgments

The author wishes to acknowledge the contributions from Scott Jackson, Cluster Resources' principal developer for the Cray platform for his edits and content contribution; Nicholas P. Cardo, the XT4 System Lead at the National Energy Research Scientific Computing Center (NERSC) at Lawrence Berkeley National Laboratory for implementing and describing the NERSC design principles, general guidelines and review; and Craig Laurence for document editing.

About the Author

Michael Jackson is President of Cluster Resources, Inc. and manages strategic customer and partner relations with leading sites and partners such as NERSC, ORNL, Tri-Labs and Cray. He is a co-founder of Cluster Resources, Inc., which formed in 2001. Jackson can be reached at michael@clusterresources.com or +1 (801) 717-3722.