GPU Age-Aware Scheduling to Improve the Reliability of Leadership Jobs on Titan

Christopher Zimmer, Don Maxwell, Stephen McNally, Scott Atchley, Sudharshan S. Vazhkudai
Oak Ridge Leadership Computing Facility
Oak Ridge National Laboratory
{zimmercj,maxwellde,mcnally,atchleyes,vazhkudaiss}@ornl.gov

***Abstract*—In 2015, OLCF's Titan supercomputer experienced a significant increase in GPU related job failures. The impact on jobs was serious and OLCF decided to replace ~50% of the GPUs. Unfortunately, jobs using more than 20% of the machine (i.e., leadership jobs) continued to encounter higher levels of application failures. These jobs contained significant amounts of both the low-failure rate and high-failure rate GPUs. The impacts of these failures are more adversely felt by leadership jobs due to longer wait times, runtimes, and higher charge rates. In this work, we have designed techniques to increase the use of low-failure GPUs in leadership jobs through targeted resource allocation. We have employed two complementary techniques, updating both the system ordering and the allocation mechanisms. Using simulation, the application of these techniques resulted in a 33% increase in low-failure GPU hours being assigned to leadership jobs. Our GPU Age-Aware Scheduling has been used in production on Titan since July of 2017.**

I. Introduction

The Oak Ridge Leadership Computing Facility's (OLCF) Titan supercomputer is one of the largest, heterogeneous CPU-GPU-based HPC deployments in the world with one GPU per compute node, for a total of 18,688 GPUs. The GPUs have enabled Titan to achieve high performance (27 petaflops (PF) peak and 17.57 petaflops actual, formerly No. 1 and No. 5 on the November 2017 Top500 list) in an energy-efficient fashion. As promising as this technological path is, we are still in the early stages of understanding the reliability of GPUs in extreme scale machines. This is particularly important to OLCF for the following reasons. One, a majority of the leadership jobs on Titan, those that use 20% or more of the compute nodes, have successfully adopted the use of GPUs, and the reliability of the GPUs impacts the reliability of these large-scale jobs. Two, OLCF's next system, the 200 PF Summit system that is currently being deployed, is continuing on the heterogeneous CPU-GPU node architectural path with six GPUs per node, for a total of 27,648 GPUs. Potential issues will only compound if left unaccounted for given the 48% increase of physical GPU components in Summit.

With Titan's operation beginning in 2013, the system exhibited a lower than projected failure rate. One every seven days compared with one every 24 hours. This changed in the second half of 2015 when GPU failures increased, resulting in the loss of several nodes per day. The failure rate accelerated through 2016 and into early 2017. A collaboration between OLCF, Cray, and NVIDIA identified that the failures were influenced by temperature and lifetime. The identification of the defect and the rate of failures resulted in the need to preemptively replace GPUs to mitigate the long-term impact to the system.

By early 2017, roughly 8,500 of Titan's 18,688 GPUs were replaced. Unfortunately, replacing every GPU was impossible as NVIDIA no longer manufactured the part. Due to the inability to replace all of the GPUs, replacements were limited to the devices that were believed to be the most likely to fail. This analysis was provided by NVIDIA and was based on the criteria of number of past failures, age of the device, and physical location of the device in the system. Location, both on the machine floor and within a cabinet, plays a significant role in the temperatures of the devices. Upon the completion of the replacements, the loss of nodes per day stabilized but large jobs continued to see higher failure rates than in the past. This prompted the investigation of additional solutions seeking to mitigate the impact to users.

In this work, we contribute two complementary techniques for using scheduling and resource selection to improve the number of stable GPUs allocated to leadership (large-scale or high-priority) jobs. We use insights from the analysis of the current system to accomplish this without impacting system utilization, a major metric for evaluating the success of the OLCF flagship systems.

The first technique modifies the resource allocation list created by the Application Level Placement Scheduler (ALPS) and places stable GPUs in a scheduling order that makes them more likely to be allocated to leadership applications. In simulation, this technique shows approximately 200,000 additional stable GPU hours per week for large jobs. Our results from production jobs on Titan demonstrate continued positive impacts with several consecutive months where the majority of failures occurred in non-leadership jobs.

The second technique builds upon a strategy developed in related work called *Dual-Ended Scheduling* (DES). With this technique, different classes of jobs are allocated from the opposite "ends" of the resource list with the goal of reducing fragmentation [1]. In this paper, we build upon that strategy to reduce the contention for stable GPUs by mapping jobs needing less stability, e.g. small short-lived jobs, to more suitable resources. Additionally, we investigate scheduling CPU-only jobs with DES moving these jobs to nodes with less stable GPUs without impacting the stability of the job. The result is an increase of availability of stable GPU nodes for leadership jobs. Applying this technique with the reordered

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

resource list we were able to add an additional 100,000 stable hours per week to large GPU jobs.

II. Background and Motivation

A. Titan Specification

The Titan supercomputer is the OLCF's 27 PF Cray XK7 system. It is powered through a combination of AMD Opteron Interlagos CPUs and NVIDIA Tesla K20X GPUs. Most of the 27 PF comes from the GPUs. Each K20X is capable of 1.31 TFs of performance through the use of 2,688 CUDA cores. To feed the huge number of available cores, each GPU contains 6 GB of GDDR5 memory with 250 GB/s of memory bandwidth. A Titan node contains one GPU per node and has 18,688 total compute nodes connected together by a Cray Gemini [2] network, providing 5.2 GB/s of network access to each node. Titan serves as the flagship resource for the OLCF.

B. Leadership Facility

The OLCF is tasked with the goal of providing resources necessary to run large-scale scientific applications. Scientific applications requiring significant resources are prioritized over smaller applications. Providing compute core hours to leadership jobs is a major metric that the OLCF is evaluated on annually. DOE also evaluates OLCF on other metrics such as, system utilization, in which OLCF must ensure that 90% of the compute hours are used.

Access to Titan is handled through a highly competitive peer-reviewed, proposal process where 90% of the hours on the machine are split between two programs, INCITE and ALCC. Both programs support a wide range of science in academia, government, and industry. A key evaluation criterion for proposals is a computational readiness metric that determines if an application can scale to the appropriate leadership class sizes on Titan and can also make use of the computational benefits of the GPU devices. A leadership job is one that uses at least 20% of the 18,688 compute nodes or 3,750 nodes. As a result of this process, the workload on Titan tends to favor leadership class applications using GPUs for acceleration.

C. GPU Reliability

Due to the role of GPUs as the primary driver for flops, it is important to understand GPU reliability. Since Titan was one of the first flagship GPU supercomputers, much has been learned from observing Titan in production. The common types of GPU hardware errors and their impact on jobs are discussed in [3]. Several of these errors are associated with failing hardware and have been uncommon since the initial component burn-in. Other errors, transient soft errors such as off-the-bus or double-bit errors (DBE), occurred infrequently. In an early study [4] using two years of GPU data, the mean time between job failure on Titan was 7 days.

GPU failure rates started increasing in July 2015. Figure 1 shows the increasing trend in GPU failures. However, it was not until the significant set of loses in April 2016, that the magnitude of the problem was apparent. The GPU failures highlighted an increase in the number of GPUs experiencing regular DBEs. Figure 2 shows the rate of failures caused by DBEs increasing in early 2016. The first two months of 2016 resulted in 47 job failures by DBE errors compared with 58 total in the previous year.

The sharp increase in the number of DBEs on a node was later discovered as the indicator of a failing device. A failing device could cause job and data loss in multiple separate jobs prior to being removed from the cluster. Since DBEs are transient and can occur in healthy equipment, the removal procedure was not enacted until a single GPU device received multiple errors before being pulled from the node. Typically a node would be removed from service, the faulting component would be replaced, and the node would be released back into the system. At the peak of the problem, Titan was losing an average of 12 nodes per day.

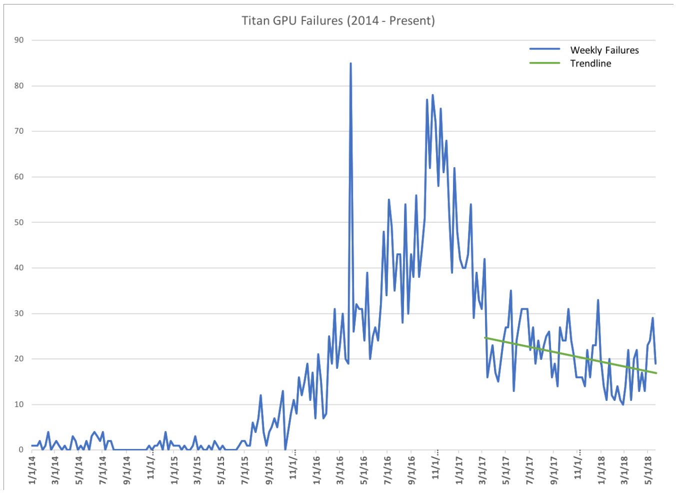


Fig. 1. GPU total failures by week on Titan, show the increasing rate of failures until the replacement of the GPUs. After replacement, the failure rate has stabilized and the trend is continuing to decrease.

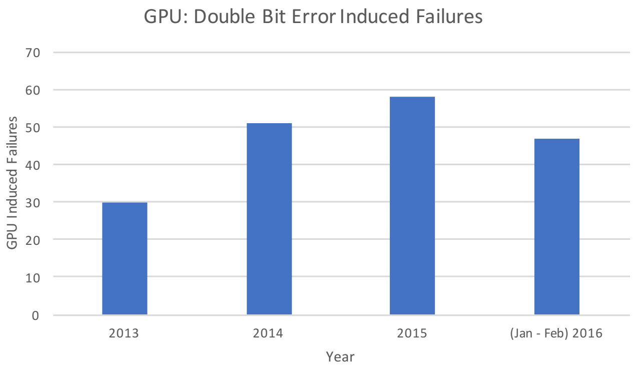


Fig. 2. GPU DBE failures by year on Titan, demonstrate the low failure rate of the system leading into mid 2015 and accelerating through 2016

By late-2016 the underlying cause of the failures was found to be a manufacturing issue on the card, but not on the GPU chip itself. The fix meant replacing failing GPU SXM devices since the component could not be easily repaired. At the peak

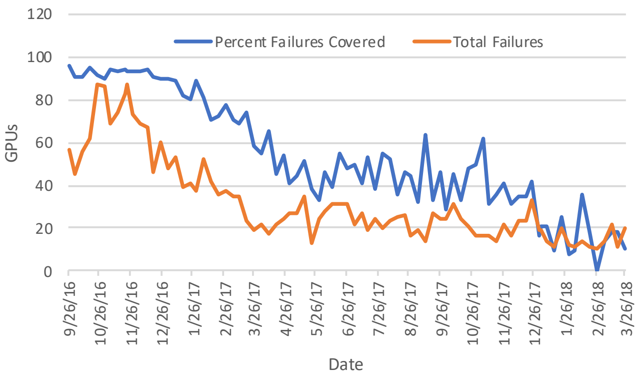


Fig. 3. Failures since September 2016 and the accuracy of NVIDIA's predictive model.

of these failures, Titan had been in service for 3.5 years. NVIDIA no longer manufactured the Tesla K20X and their failure models did not account for the number of parts created by this scenario. This meant that only a portion of Titan's GPUs could be replaced.

Ultimately, a significant number of replacement GPUs were either located or manufactured and as of July 2017, 9,500 GPUs were replaced. The choice of which GPUs to replace was based on predictive models developed by NVIDIA, which took into account the age, location in the room, and heat exposure for individual GPUs with heat identified as the prime contributor to failure. GPUs highest in a rack would fail sooner and more frequently than those lower in a rack. Figure 3 depicts the actual failures since September 2016 and the ability of NVIDIA's model to predict which specific GPUs would fail. When the failures were at their peak, NVIDIA's model was over 90% accurate and provided confidence on which GPUs to replace. As the replacements lowered the failure rate, the model's ability to predict specific failures decreased as well. The predictive model indicated that the components most likely to fail were based on the lifetime of the component and temperature exposure over time. This overlaps with previous studies [5] that identified clusters of failures based on spatial locality, i.e., location of the rack in the data center and the location of GPUs within a rack. While this decision was pragmatic, the impact on scheduling was not well understood.

D. Scheduling

1) MOAB: Titan schedules jobs through Adaptive Computing's MOAB [6] scheduler. Since Titan contains a single node type, there is one primary queue for batch scheduling as node differentiation is unnecessary. The policy MOAB uses is broken into four domains based on the size criterion. This policy is outlined in Table I and as shown, large leadership jobs receive the highest priority and are enabled to run up to 24 hours, the maximum duration allowed. While this significantly reduces the queue wait times experienced by leadership jobs, they still generally wait the most time in the queue, since smaller jobs primarily run in backfill windows [7]. The concept of backfill is a technique for enabling higher system utilization by shuttling smaller, shorter jobs onto the machine through side channel scheduling. Essentially, the concept works by calculating the soonest that the highest priority job can be placed on the machine. This is performed by forcing each application to input the expected time duration of their job. A job exceeding this time will be forcibly terminated. Using this information the scheduler is able to calculate the set of nodes that will be available soonest, if all jobs run up to their wallclock time. This set of nodes becomes a backfill window. As jobs running in this allocation finish up, smaller jobs on the queue can be placed on these nodes, out of order, if their requested wall clock time do not impede the larger job trying to get onto the machine. This significantly reduces the time to placement for smaller applications on Titan, traditionally large jobs can spend several days to weeks in queue prior to being scheduled on the machine. Resource reservations are granted to the two highest priority jobs on the scheduling queue. Once resource reservations have been calculated, it can take up to an additional 24 hours before a job is actually placed on the machine.

Policy Name	Nodes/Job	Maximum Runtime	Aging Boost
Bin60	11,250 - 18,688	24 Hours	15 Days
Bin20	3,750 - 11,249	24 Hours	5 Days
Bin0	125 - 3,749	6 Hours	0 Days
SmallMaxJobs	1 - 124	2 Hours	0 Days

TABLE I
SCHEDULING BINS FROM TITAN MOAB CONFIGURATION SHOWING THE PRIORITIZATION AND TIME DURATION BASED ON REQUESTED JOB SIZE.

2) ALPS: Resources, in this case compute nodes, are scheduled from an ordered list that is generated by Cray's ALPS. The list is constructed at system boot time and integrates knowledge of the network into the resulting list. There are several network properties that are important in the construction of the ALPS list. First, the Gemini 3D Torus is anisotropic, meaning that latency and bandwidth in the X,Y, and Z dimensions are not symmetric. In particular, the Y-dimension of the network alternates between 4.68 GB/s and 9.36 GB/s, while the X and Z dimensions achieve 9.36 GB/s and 15 GB/s respectively. ALPS takes advantage of this knowledge by using basic building blocks to construct the list. A basic building block on Titan is a 4x2x4 block of nodes, shown in Figure 4. The block is 4 wide in the X-dimension, 2 wide in the Y-dimension, and 4 wide in the Z-dimension. From an application MPI placement perspective, this results in a grouping of processes that have 9.36 GB/s of bandwidth in the X and Y dimensions and 15 GB/s in the Z-dimension. Each block contains 4, 2x2x2 Hilbert curves using a clockwise orientation, as depicted in the second half of Figure 4. Basic building blocks are used to populate the system. Blocks are placed starting from the 0,0,0 coordinates filling up the z-dimension first. The y-dimension is populated second, until fully populated, and the x-dimension is populated

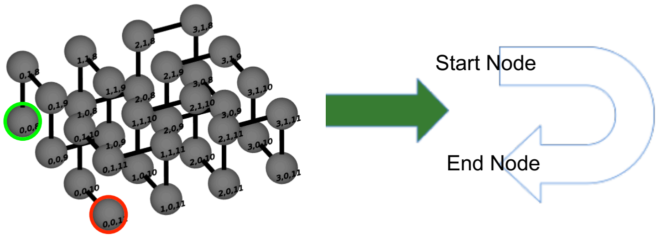
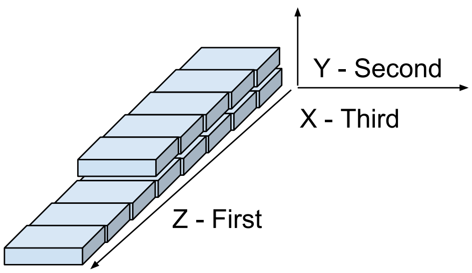


Fig. 4. ALPS 4x2x4 Basic Building Block, used for network aware enumeration of Titans nodes for scheduling and rank placement

Pool A (New GPUs)	Pool B (Old GPUs)	MTBF
0	4,000	11.0
1,000	3,000	13.8
2,000	2,000	18.5
3,000	1,000	27.9
4,000	0	57.0

TABLE II
IMPACT ON MTBF FOR A 4000 NODE JOB ON TITAN USING DIFFERENT RATIOS OF NEW AND OLD GPUS.



NID	Coordinate
92	0 1 1
2	0 0 1
18434	1 0 1
18524	1 1 1
18526	1 1 0
18432	1 0 0
18430	2 0 0
18336	2 1 0
16990	3 1 0
16896	3 0 0
16898	3 0 1
18428	2 0 1

Fig. 5. Building block placement in a 3D torus. Population prioritizes the Z, Y, and then the X dimension

third. Figure 5 shows the population of a Titan like system using 13 basic blocks. The result is a list of nodes, as shown in Figure 5.

Outcome: As GPUs were being replaced the position in the ALPs list was not considered. The results was a wide distribution of stable GPUs across the ALPs list. This meant that even when jobs got large contiguous allocations, the mixture of nodes contained a random mix of stable and unstable GPUs. The high mixture of GPU types in a job reduced the potential impacts intended by the replacements. Leadership jobs continued to accumulate the majority of job failures on Titan.

III. Design

1) Design Space:

a) Motivating Example: To motivate the problem, we present analysis of the impacts and rates of failure based on a simplistic representation of the system. For this effort we consider a single 4,000 node leadership job running for 24 hours. Using the scheduling and resource selection mechanism described in the previous section, we consider the impact to the job's MTBF at different ratios of GPUs. Titan contains two separate pools of GPU resources. Pool A contains 9,500 nodes of stable GPUs with a failure rate of 1 node per 24 hours while Pool B contains 9,188 nodes of unstable GPUs with a system-wide failure rate of 5 nodes per 24 hours. We consider five different ratios of new and old GPUs in Table II. We treat the two pools as a series (i.e., failure of either pool causes system failure) and we compute the MTBF for the combined pool.

The table shows that job stability improves substantially by increasing the Pool A GPUs within the allocation. Biasing GPU-enabled tasks toward nodes with newer GPUs and CPU-only tasks to nodes with older GPUs can benefit non-HPC, work-stealing runtimes as well even though the impact of a node failure may not be as large as in tightly-coupled HPC simulations.

b) Alternatives and Challenges: From a practical perspective, influencing the number of new GPUs in a leadership allocation is not trivial. Several strategies were proposed including enabling more queues for users to specify allocation needs, such as CPU only, to free up GPU resources. Unfortunately, without incentives, it would be difficult to get users to specify such information. Additionally, incentives may encourage unintended consequences such as a misuse of the resource, creating more negative effects. We also considered specifically reserving new GPUs for leadership jobs. Unfortunately, the mechanisms for performing this, such as a separate queue or a new-GPU resource flag, could impact system utilization, create longer wait times on the system, or both. Given that system utilization is one of the most significant DOE metrics used to evaluate the effectiveness of the OLCF, this option was not considered. We considered moving GPUs, but moving old GPUs to higher heat locations would accelerate their failures and moving large numbers of GPUs to make contiguous allocations using the original node ordering was impractical. A fourth option was to prioritize new GPUs into large jobs. The challenge with this approach would be validating that the change was effective and ensuring that the impact on the allocations and the network did not result in significant performance degradation due to increased fragmentation.

c) Basis: A previous study [1] modified Titan's scheduling to improve application performance based on a job size criteria to shape allocations by reordering the ALPS list. Additionally, a secondary technique was developed called *Dual-Ended Scheduling* (DES) that sought to alleviate the impacts to network performance that were caused by allocation fragmentation. The authors analyzed the number of hours that had been historically allocated to each node based on the node's position within the ALPS scheduling list. They found that the first-fit algorithm allocated the most hours to nodes at the front of the list, with the back end of the list getting the fewest hours. This is shown in Figure 6 prior to the

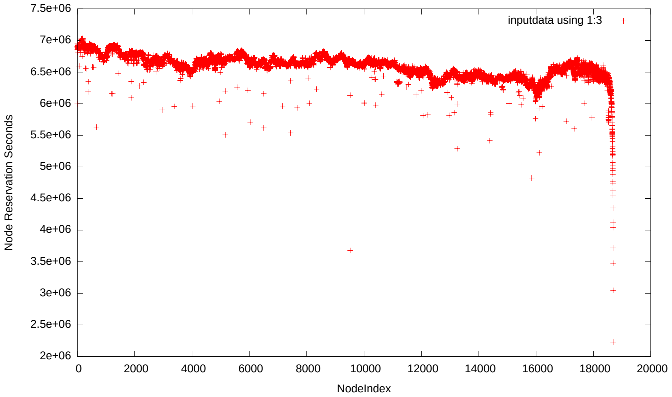


Fig. 6. Allocated hours to each node of Titan based on position in the ALPS list. Nodes in the first 20% of the machine are traditionally allocated first due to a top-down first-fit allocation strategy.

implementation of DES on Titan. With the use of DES, the back end of the list was allocated more small job hours, while the front end of the list was allocated more large job hours.

d) Overview: In this work, we exploit the aforementioned scheduling strategies of ALPS ordering and DES on Titan, and bring them to bear in a novel way to improve the reliability of leadership jobs. These techniques were originally developed to fundamentally improve job performance by reducing fragmentation. However, we see the potential to apply smart scheduling to improve the reliability of the jobs and consequently improve the productivity of the system. Specifically, our contributions are as follows.

- GPU Age-aware ALPS Reordering: In this technique, we create a new node ordering of the scheduling list, one based on the age and stability of the GPUs.
- GPU Focused Dual-Ended Scheduling: A new application of DES seeking to match jobs needing less GPU stability such as small or cpu-only jobs, to the set of resources offering less stability, creating less contention for stable GPUs.
- Multi-parameter Simulation Study: Using scheduling simulation we are able to evaluate a wide range of parameters to the proposed techniques on an actual workload extracted from Titan.
- Large-Scale Test-shots on Titan: Simulation results provide the confidence necessary for live test-shots on the full Titan system. Results from the test-shots allow us to quantify the potential negative impacts to network performance from the proposed strategies.
- Deployment: Our scheduling improvements have been deployed in production for the past year, and has resulted in tangibly improving the reliability of leadership jobs. Finally we continue to perform on-going analysis of our techniques on the production Titan system to gauge their efficacy.

2) GPU Age-aware ALPS Reordering: From the scheduling work mentioned previously, it became clear, that the policies in MOAB have resulted in a preference being given to the nodes at the head of the list. From this, we propose a GPU Age-aware reordering of the scheduling list in order to improve the reliability of leadership jobs. A reorganization of the scheduling list may be utilized to provide stable GPU preference to jobs that are scheduled on the machine through traditional (non-backfill) scheduling mechanisms. Further, when employed with a past approach such as DES, it may be possible to additionally schedule jobs not needing GPUs in a portion of the machine with older GPUs.

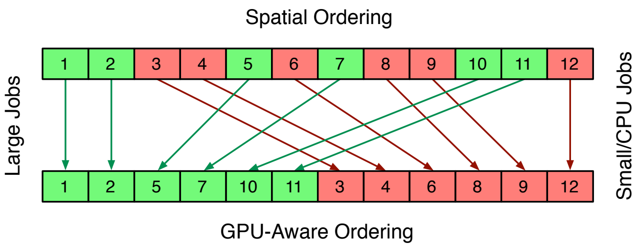


Fig. 7. Reordering to move new GPUs to the start of the list to service large, GPU-enabled jobs

The complexity built into ALPS, discussed in II-D2, is intended to provide better network performance for jobs. However, in practice, many of the impacts get lost. This is due to operationally mixing multiple jobs into the supercomputer. The result is fragmentation, an imperfect allocation that creates several disjoint sets of nodes spread throughout the machine for a job's use. The result of fragmentation is decreased network performance. There have been several studies on fragmentation at different facilities [1] [8], each with solutions meeting a particular facility's operational needs.

GPU Age-aware ALPS Reordering is simple in implementation, yet very powerful. Titan continues to use ALPS to generate the original scheduling list based on the network enumeration of the system. A secondary reorder pass of the list is conducted prior to handing off to MOAB, shifting all known stable GPUs upward in order as shown in Figure 7. The resulting list has the 9,500 new GPUs at the top of the list and the 9,188 older GPUs below them. The ordering of the GPUs in relation to their type is maintained. The list was created with the understanding that it would impact network performance. However, due to natural fragmentation that occurs on the system, it is possible that the resulting allocations under a traditional workload would have marginal impact.

3) GPU Focused Dual-Ended Scheduling: Dual-Ended scheduling is a strategy used on Titan that reduces the impact of fragmentation on large jobs by scheduling smaller jobs from the opposite end of the ALPS list. In the original work, it was observed that the top-down scheduling strategy of jobs of different sizes and wall-clock times would lead to gaps in the scheduling list. This was particularly true at the OLCF with a mixture of very large jobs running upward of 24 hours. Instead, Dual-Ended scheduling combines the use of top-down and bottom-up scheduling for the machine using a demarcation point to select which strategy to use. The current implementation on Titan uses a demarcation point of jobs sized at 16 nodes to trigger bottom up scheduling. The challenge was the selection criteria of the demarcation point for best results.

Similarly, the impact of dual-ended scheduling could be used to move smaller jobs away from larger jobs needing more stable GPUs. Using a similar strategy it may be possible to improve the matching of jobs with less stability requirements to nodes better matching those needs. Again, the question becomes, how to to select a demarcation point to match the needs of the system.

In the past, job-size was a good selection to reduce fragmentation because of the system imposed time-limits on these jobs (24-hours). Other strategies were evaluated, such as selecting jobs with short wall clock requests. It is reasonable to prioritize small jobs to less stable GPUs. Their shorter wall-clock duration makes them less likely to lose significant work in the face of a failure. However, the definition of a small job remains rather nebulous. With the specific goal of seeking to improve stability for leadership class jobs over 3,750 nodes, any job less than 3,750 nodes could potentially constitute a small job. For the course of this study, we compare the current dual-ended 16 node (DE16) with a demarcation point that considers any less than 20% (3,750 nodes) of the system for separate scheduling.

Another potential demarcation point relies on the knowledge that we have projects on the machine that operate using only CPUs. The most common leadership job size on Titan sits within the Bin20 bucket. These jobs, shown in table I, have a large range, but the most common job submissions sit at the lower end of that bucket. This means that the machine is often running simultaneous 3,750-4,096 node jobs. For instance applications from Climate studies tend to use CPUs only for computation over particularly large allocations. If CPU only leadership jobs are prioritized to stable GPUs, that could reduce the stability of a subsequent job that is scheduled slightly later. Instead it may be possible to use the dual-ended techniques to prioritize known CPU only accounts on nodes containing older GPUs. The number of CPU only accounts on Titan is small and the numbers decrease as scientific users become more comfortable with GPUs. For this reason, we do not consider CPU only accounts as a primary demarcation point, and instead couple it with one of the other sized based approaches.

4) Motivation for Simulation and Test-Shots: The study and implementation of scheduling policies on Titan is challenging. The policies used on the production system are well understood as well as the impacts to the scheduling queues and utilization. New scheduling policies could introduce unintended consequences such as increased queue lengths or decreased utilization. Ultimately, pushing new scheduling policies onto Titan requires in depth analysis and demonstrable impacts. A second challenge is understanding the impact to the network performance associated with changing the allocation policy. As mentioned previously, the ALPS ordering integrates knowledge of Titan's network into the list to densely pack processes for better communication performance. While maintaining this order is important, it often becomes the case that jobs are naturally fragmented due to the system being multiplexed. However, impacts to application performance through scheduling should be understood prior to deploying such changes.

We address these challenges in several experimental phases. To study the possible set of changes to the scheduling system, we use scheduling simulations that model the Titan system. These simulations allow us to take actual workloads from Titan and modify the scheduling approaches to observe the effects without impacting our production resource. The results of the simulation can then be analyzed to determine quantifiable impacts to the production workload. Studying the actual impacts to network performance from scheduling requires the use of a test-shot. A test-shot is a targeted study on the actual machine where a representative workload is run using base and modified conditions. Careful measurements are taken during the experiments to serve as a basis for comparison. At this point an argument can be made for pushing the changes to the production machine. Upon successful deployment to production, monitoring is continued to verify the outcome of the changes. In the next section we will discuss our simulation analysis.

IV. Simulation

The workload on Titan includes a wide range of job sizes, arrival times, and run times. Modeling the workload is best done by using historical traces. As mentioned in the previous section, Titan uses the Adaptive MOAB scheduling system. The MOAB scheduler provides a simulation mode that enables historical traces to be replayed using different scheduling conditions. While replaying applications, the simulator is able to use the timings from a JOBEND record to model the request that was originally used on Titan.

Along with other information, a scheduling trace contains the number of nodes, submission time, project, and wall clock time. Each of these plays a factor in the scheduling decision made for the job. For replaying, the traces contain the actual job outcome (completed), completion time, and original dispatch time so that the simulator is aware of how long the job actually ran for compared to the requested wall clock. The result of changing the scheduling decision coupled with the actual run time of the job impacts the set of nodes and backfill windows that are available at each scheduling iteration. The result of a simulation is a change to the scheduling decisions of when jobs are dispatched, what nodes are used, and impacts to average utilization, i.e. how often nodes sit idle due to inability to schedule.

As discussed in the design section, we are interested in determining the impact to scheduling from the modification of two separate but correlated scheduling policies with the goal of increasing large job stability. In our set of simulation experiments, we used a one month trace from the period before the GPU stability issues. Our simulation hardware environment consisted of 8,500 stable GPUs and 10,188 original GPUs. At the time of the simulation experiments, this modeled the configuration on Titan. This trace contained roughly 4,000 jobs and had a common distribution of job sizes that made it a good candidate for studying the impacts across a wide range of leadership job sizes. We chose a trace from before the GPU stability problems to model workloads from when the system was healthier. At the peak of the stability problems the scheduling traces had a higher failure rate and the number of successful jobs dropped. Users were also modifying their usage behavior in an attempt to mitigate the impact of job loss.

We ran our simulation to measure four different cases.

1) DE16 Base: The base system, which was the default ALPS ordering coupled with the DE16 scheduling demarcation point.
2) DE16 Reorg: The modified ALPS ordering with the DE16 demarcation point.
3) Bin20Bin60 Base: The default ALPS ordering with less than Bin20Bin60 used in the demarcation point.
4) Bin20Bin60 Reorg: The modified ALPS ordering with less than Bin20Bin60 use in the demarcation point.

The simulation was run over the course of several weeks as the MOAB simulator has little acceleration capability and playback is at near real time. The results of the simulation informed policy decisions and changes to Titan's scheduling preferences. From these changes we were able to determine the new sets of nodes that jobs were allocated to, allowing us to infer information about the type of GPUs the jobs were given and the layout of the job within the network. We split the leadership sized jobs into 3 buckets representing jobs greater than 20%, 40%, and 60% of the machine (total number of nodes). Due to size constraints it was likely that the most impact would be in the 20% bucket, while the 60% was unlikely to be affected much because these jobs allocate most of the nodes in the machine. A reordering strategy will have no effect on a full machine size job. The first set of analyses broke large jobs into the three buckets and calculated a min, max, and average on the number of "New GPUs" across the set of jobs.

Figure 8 shows the results from the 20% bucket. The jobs in this bucket represent the majority of the representative leadership class jobs in the simulated workload. These results show an interesting impact with the reorganized ALPS list strategies. In these results, the minimum measured job shows that it actually received zero stable GPUs, however the average in both cases saw increases. The largest increase was with the base reordering and the original DE16 demarcation point. In this case the average Bin20 job saw almost 500 more stable GPUs on average. Deeper analysis of the cases showing zero GPUs were shown to occur when two or more leadership jobs were simultaneously allocated. This increase in number of simultaneous leadership class jobs running at a given point means that there are fewer stable GPUs available for subsequent jobs.

The next set of results in Figure 9, show similar impacts. Both reorganized models showed significant improvements to allocated stable GPUs. However, the overall impact was greater than in the Bin20 bucket. These results were promising as they demonstrated that in many cases the allocation mechanism was able to schedule the full set of stable GPUs into these Bin20 jobs. Compared with early allocation strategies which commonly left 50% or more of the GPUs allocated into smaller jobs. This bucket also saw scenarios where the maximization of stable GPUs into one job, left co-scheduled jobs with no stable GPUs.

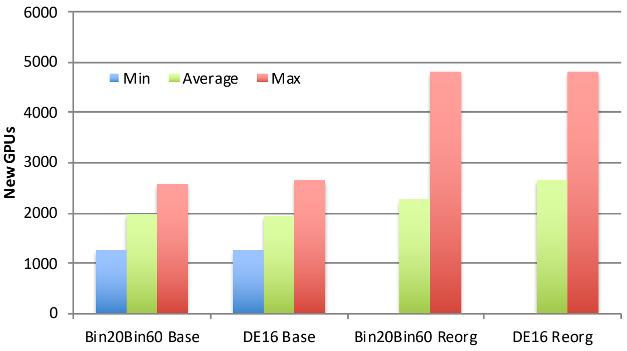


Fig. 8. Simulation results showing the impact to Bin20 jobs using GPU Age-aware ALPS Reordering and varying the DES demarcation Point

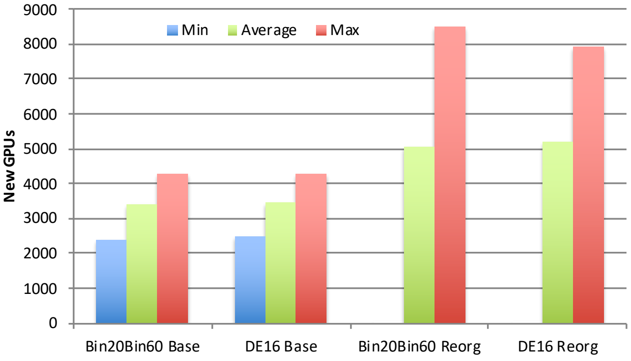


Fig. 9. Simulation results showing the impact to Bin40 jobs using GPU Age-aware ALPS Reordering and varying the DES demarcation Point

The Bin60 bucket analysis shown in Figure 10 has some surprising results. Due to the node coverage of these jobs, it was believed that there would be little impact at this scale. However, the results demonstrate again, an increase in the number of stable GPUs in the average job. Analysis following the simulation on the Bin60 bucket indicates the impacts were experienced by Bin60 jobs sitting within the 11,000 to 14,000 node count ranges. The average increase of over 1,000 nodes into these jobs would provide significant boosts to stability.

From the original set of simulations we determined that the reordered list with a dual-ended 16 node demarcation point provided significant increases in the average number of stable GPUs across all leadership buckets. In our next set of results we add an additional simulated strategy which uses a combined demarcation point for scheduling of DE16 and CPU only accounts. In these results we analyze the set of

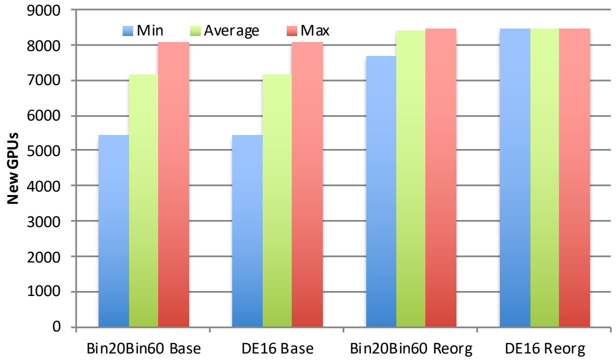


Fig. 10. Simulation results showing the impact to Bin60 jobs using GPU Age-aware ALPS Reordering and varying the DES demarcation Point

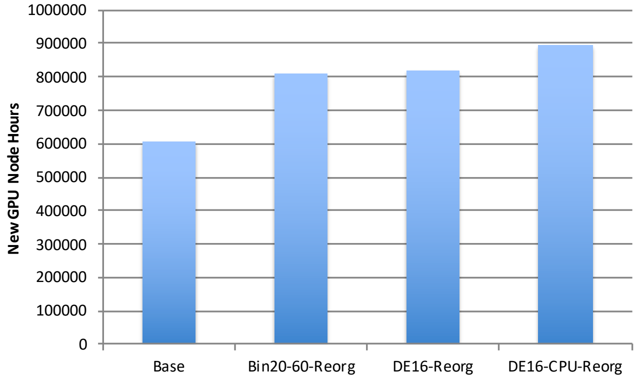


Fig. 11. Simulation results showing the impact across all demarcation points with the CPU only job selection point added. The impact of adding CPU Only jobs to the back end of the list indicates a promising impact in additional GPU hours.

used nodes against the number of hours each of the jobs run. We present these results as the number of new GPU node hours. These results show the aggregate number of new GPU hours across the full set of leadership sized jobs based on the scheduling algorithm used. From these results, we see that the base reorganized strategies presented previously show a significant improvement in the number of hours allocated on stable GPUs. With the addition of CPU-only accounts, we see an increase of almost 100,000 additional stable GPUs hours. Clearly, the combination of reordering the ALPS list with a DE16 + CPU-only account dual-ended scheduling can have positive impacts on large job stability.

In the final set of simulation results provided, we investigate the impact to job fragmentation associated with the DE16+CPU reorganized ALPS scheduling strategy. Job fragmentation serves as a good measure of performance impact. The fragmentation measurement is a calculation of the network hop count distance between every two sets of nodes in an allocation. The measurements are averaged across all nodes within a job. In general, the higher the fragmentation value, the more it is possible for the job's network performance to be low.

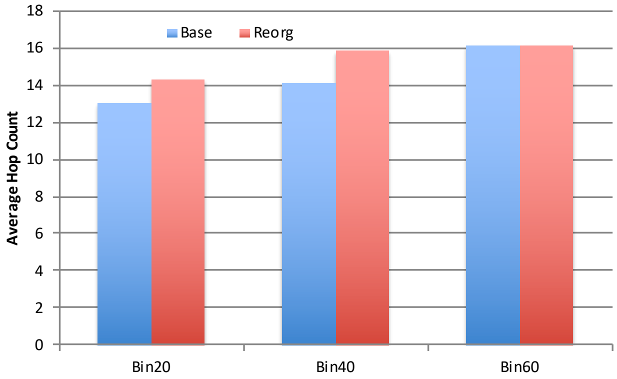


Fig. 12. Simulation results showing the impact to network fragmentation from the DE16+CPU Reordering Strategy

The results from the simulation compare jobs against the base scheduling algorithm using the previously discussed buckets. The results indicate that network fragmentation does increase as expected from stability based scheduling improvements. It will be necessary to measure these impacts to real applications on the system.

A. Network Impact Study

The results from simulation indicate that targeted scheduling can be used to improve reliability outcomes in leadership jobs. It also indicates that using these techniques will possibly introduce additional fragmentation into the system. To understand the impacts to performance, we performed a multi-month study involving the measurements of jobs in both ideal scheduling and production scheduling environments. Measuring ideal scheduling, meaning the jobs are not fragmented due to production based multiplexing, is done using a test-shot. However, ordering based fragmentation will still occur. Production scheduling measurements are performed by injecting the same jobs into the production systems while in use with other production jobs. This subjects jobs to adversarial traffic loads and multiplexing induced fragmentation. In both cases we compare ALPS ordering against the GPU Age-aware Scheduling and measure the impact to application runtimes.

Our experiments use a set of benchmarks and applications from the acceptance harness [9], [10] used by OLCF. In these tests we use

- Ziz: A benchmark from the Chimera application modeling the core collapse of a supernova,
- XGC: A multiphysics simulation with significant GPU use,
- LAMMPS: A molecular dynamics application,
- GTC: A turbulence simulation in plasma code,
- Minisweep: A benchmark modeling the sweep portion of the Denovo radiation transport code,
- S3D: A turbulent combustion code and,
- DCA: A quantum monte-carlo solver for high-temperature superconductivity.

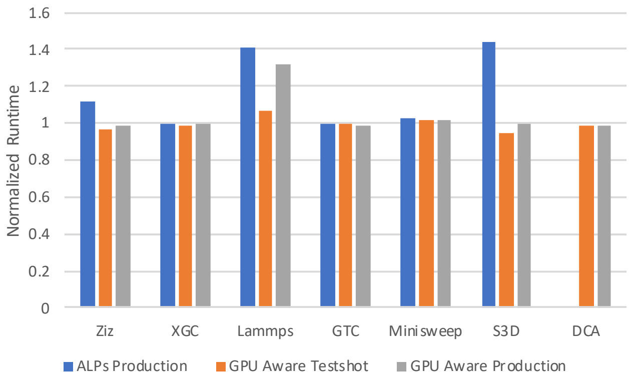


Fig. 13. Network impact study application sizes of 4096 nodes normalized to the ALPS testshot performance. Lower is better.

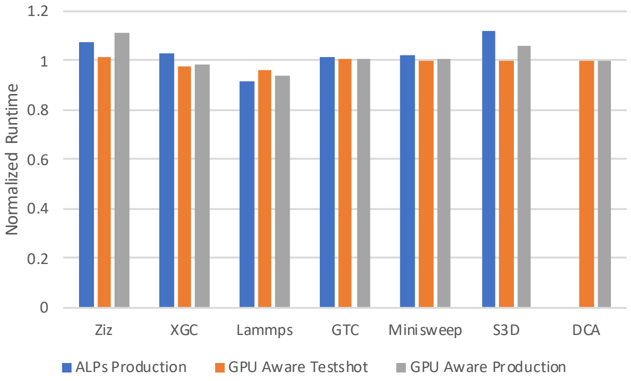


Fig. 14. Network impact study application sizes of 8192 nodes normalized to the ALPS testshot performance. Lower is better.

These applications were originally chosen to be included in the OLCF acceptance harness due to their being representative of traditional workloads. We ran our applications against two common leadership sizes seen on Titan, 4,096 and 8,192 nodes.

The first set of production measurements were taken over the course of May and June 2017. Unfortunately, due to unresolved library issues, we have no results for the ALPS production runs of DCA. In early July of 2017, Titan was taken offline for 8 hours to perform these tests in isolation. The first half of the testshot was used to take baseline measurements against the default ALPS ordering and the second half was used to run applications using the new ALPS list. This list would include the baseline fragmentation that was created during the reordering. To maintain consistency, jobs were deployed in the same order across both testshot tests.

The first set of results are shown in Figure 13. The results are normalized to the ALPS testshot performance. The results broadly indicate that there are no significant negative impacts when running the applications across the separate lists. However, in some cases, such as with S3D, the production ALPS runs are out performed by the GPU Aware runs. These types of results are common in production and are the result of network fragmentation and adversarial network traffic introducing perturbation into the applications. These negative performance impacts are often temporal and vary from job to job. Overall, the results indicate, in both testshot and production models, is that the 4,096 sized jobs do not perform measurably worse than their ALPS production based comparisons. Similar, results can also be seen in the 8,192 node job results shown in Figure 14.

Based on the promising results from these tests, GPU ordering was left on when returning to production so that additional measurements could be taken. These results are labeled GPU Aware Production and shown in both the 4,096 results, Figure 13, and 8,192 results, Figure 14. What these results indicate is that the additional fragmentation created by the list reordering resulted in little appreciable degradation in performance. The only test appearing to perform worse than ALPS production occurs in the single benchmark application Ziz. While in both sets of tests, the testshot results show performance decline as the state of the machine is traditionally fragmented. Based on these results and the potential to reduce leadership class job failures, the decision was made to leave the updated ordering in production and allow base reordering on request for full system jobs when needed.

B. Production Impact

The scheduling changes to the ALPS ordering list have been deployed in production on Titan since July 2017. On-going analysis has shown immediate impact over the last several months. Figure 15, shows the percentage of DBE failures in leadership jobs that occurred between January of 2016 to May of 2018. These results show that historically around 55% of DBE failures would occur in leadership jobs. After the implementation of our first set of changes, the reordering of the ALPS list, the failures dropped immediately. In the months of August 2017 - November 2017, the leadership jobs made up less than 45% of the failures. In January 2018 - March 2018 these same jobs made up only 32% of the failures.

The results do show spikes in three of the months. Analysis of the workload of these months show a large increase in the number of jobs over 10,000 nodes. This could be due to several scenarios. INCITE project allocations are based on the calendar year and users often rush to run more leadership jobs in December to finish out allocations. While this happens annually, the size of the jobs associated with the allocations change frequently depending on the awarded projects. The months of April and May often have a significant boost in the number of very large jobs as this time corresponds with the SC Gordon Bell challenge paper submission season. Many large jobs ran up to the deadline in mid-April and continued beyond, as participants continue to tune applications for the possibility of being Gordon Bell finalists.

The production results demonstrate a change to the trend of job failure ratios starting in July of 2017. This time coincides with the deployment of our scheduling change. Unfortunately, it is very difficult to quantify the precise impact of our

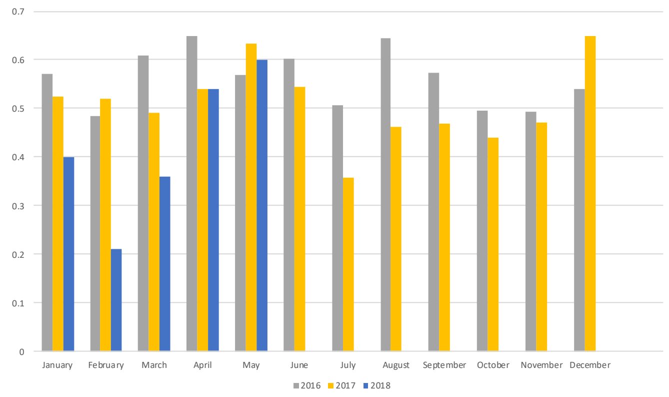


Fig. 15. Job failure data from Titan from the last 25 months. This data shows a measurable change in the percentage of failures occurring in leadership jobs. The GPU-Aware scheduler changes entered production in July 2017.

scheduling modifications in the production environment and separate them from the effects of things such changes in system workload. For this, we added another angle of analysis that investigated the locations of failures under the new scheduling paradigm for the months of August, September, October, and November of 2017. In this analysis, we mapped the location of the failure back to the original position in the ALPs list and determined what job would have been running on the node at that time of failure. Our findings indicate that in all of the months measured, the majority of the failures would have occurred in leadership sized jobs if we were still using the original list. In particular, September 2017 and October 2017 would have had 65% and 69% of the failures in leadership sized jobs compared to 46% and 43%, respectively, using the revised list. We must point out that this analysis is simply a "what if" analysis that has several caveats. GPU failures may not have occurred in the same manner at the same time depending on the changes in workload. Also, there was not always a job running on a node at the time of failure.

The CPU-only job scheduling changes are currently under development for automated integration into the scheduling system that will update the set of accounts labeled CPU-only intelligently. We expect these changes to positively increase the impact to leadership class jobs.

V. Related Work

Moab resource selection schemes are similar to traditional memory allocation schemes such as traditional first fit or next fit techniques [11]. Unfortunately, naive node selection can have significant performance impacts if other system factors are not considered. Thus many HPC schedulers maintain a level of network awareness, this can be easily seen in [12], where the authors study the performance impacts on the BlueGene/L system and determine the importance of geometric allocations.

The scheduling techniques on Titan such as the use of Hilbert Curves [13], originally proposed in [14] are another network aware adaptation for HPC systems finding that the tight placement of processes work well with traditional communication patterns. This work ultimately lead to the work integrated into ALPS [15], [16] which evaluated several single list enumeration techniques for torus networks. The findings indicated that the single list techniques result in high network locality and reduced scheduling costs for systems.

Targeted scheduling is a well studied area. Many HPC systems have limitations associated with power, stability, or performance that may require specialized scheduling techniques. The previous discussed techniques are used for targeting network performance. Examples of power aware resource allocation techniques such as those presented in [17], using scheduling windows and online energy cost mechanisms for delaying known power consuming applications to lower energy demand periods. The techniques trade off utilization but at significantly reduced power costs for the total system.

Another example of resource aware scheduling came in recent changes to Spectrum Scale LSF [18], with the implementation of SSD awareness in scheduling. This change is being integrated as a set of changes from LSF for the CORAL systems which contain node-local SSD devices. LSF is adding a layer of depth for node-local burst buffers to enable data prestaging, post-staging, and wear-aware node selection. In this work, the scheduler is balancing the system workload based on device drive writes per day to ensure that resources in the machine are balanced. Resource scheduling like this in the future could also be applied to accelerator devices to help eliminate the need for scheduling techniques presented in this work.

VI. Conclusion

In this work we have presented a multi-faceted strategy for improving the reliability of leadership jobs in a mixed-failure rate system. We have achieved this by modifying Titan scheduling–reordering the ALPS list to actively target more stable GPU resources and introducing a method for moving CPU only jobs to less stable GPU nodes. We have demonstrated the positive impacts using significant simulation experiments and in production on the Titan system. We were also able to demonstrate that the potential impact on network performance associated with these changes is very low (virtually no impact) in typical operational modes of the system. The success of these strategies has led to the active use of these techniques on Titan to improve the reliability of large-scale jobs.

Acknowledgement

This work was supported by the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is managed by UT Battelle, LLC for the U.S. DOE (under the contract No. DE-AC05-00OR22725).

REFERENCES

[1] C. Zimmer, S. Gupta, S. Atchley, S. S. Vazhkudai, and C. Albing, "A multi-faceted approach to job placement for improved performance on extreme-scale systems," in *SC16: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2016, pp. 1015–1025.

[2] R. Alverson, D. Roweth, and L. Kaplan, "The Gemini System Interconnect," in *High Performance Interconnects (HOTI), 2010 IEEE 18th Annual Symposium on*, Aug. 2010, pp. 83–87.

[3] "Nvidia XID Errors." [Online]. Available: http://docs.nvidia.com/deploy/xid-errors/index.html

[4] D. Tiwari, S. Gupta, G. Gallarno, J. Rogers, and D. Maxwell, "Reliability lessons learned from gpu experience with the titan supercomputer at oak ridge leadership computing facility," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '15. New York, NY, USA: ACM, 2015, pp. 38:1–38:12. [Online]. Available: http://doi.acm.org/10.1145/2807591.2807666

[5] S. Gupta, D. Tiwari, C. Jantzi, J. Rogers, and D. Maxwell, "Understanding and exploiting spatial properties of system failures on extreme-scale hpc systems," in *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, June 2015, pp. 37–44.

[6] "Adaptive Computing Moab HPC Suite." [Online]. Available: http://www.adaptivecomputing.com/products

[7] D. B. Jackson, Q. Snell, and M. J. Clement, "Core algorithms of the maui scheduler," in *Revised Papers from the 7th International Workshop on Job Scheduling Strategies for Parallel Processing*, ser. JSSPP '01. London, UK, UK: Springer-Verlag, 2001, pp. 87–102. [Online]. Available: http://dl.acm.org/citation.cfm?id=646382.689682

[8] J. Enos, R. Bauer, S. Islam, R. Fiedler, M. Steed, and D. Jackson, "Topology-aware job scheduling strategies for torus networks," in *Proceedings of the Cray User Group*, May 2014.

[9] O. B. Messer, E. DAzevedo, J. Hill, W. Joubert, M. Berrill, and C. Zimmer, "Miniapps derived from production hpc applications using multiple programing models," *The International Journal of High Performance Computing Applications*, vol. 0, no. 0, p. 1094342016668241, 2016. [Online]. Available: https://doi.org/10.1177/1094342016668241

[10] V. G. Vergara Larrea, W. Joubert, M. Berrill, S. Boehm, A. Tharrington, W. R. Elwasif, and D. Maxwell, "Experiences evaluating functionality and performance of ibm power8+ systems," pp. 254–274, 10 2017.

[11] C. Bays, "A comparison of next-fit, first-fit, and best-fit," *Communications of the ACM*, vol. 20, no. 3, pp. 191–192, Mar. 1977.

[12] E. Krevat, J. Castaos, and J. Moreira, "Job scheduling for the BlueGene/L system," ser. LNCS. Edinburgh, Scotland: Springer, 2002, pp. 38–54.

[13] "Hilbert curve – from wolfram MathWorld," http://mathworld.wolfram.com/HilbertCurve.html, Mar. 2010.

[14] V. J. Leung, E. M. Arkin, M. A. Bender, D. Bunde, J. Johnston, A. Lal, J. S. Mitchell, C. Phillips, and S. S. Seiden, "Processor allocation on Cplant: achieving general processor locality using one-dimensional allocation strategies," in *Proc. 4th IEEE International Conference on Cluster Computing*, 2002, p. 296304.

[15] C. Albing and M. Baker, "ALPS, topology, and performance: A comparison of linear orderings for application placement in a 3D torus," in *Proceedings of the 2010 Cray User Group International Conference*. Edinburgh, Scotland, UK: Cray User Group, May 2010.

[16] C. Albing, N. Troulier, S. Whalen, R. Olson, J. Glenski, H. Pritchard, and H. Mills, "Scalable node allocation for improved performance in regular and anisotropic 3d torus supercomputers," in *Proceedings of the 18th European MPI Users' Group Meeting*, ser. EuroMPI '11, 2011, pp. 61–70.

[17] Z. Zhou, Z. Lan, W. Tang, and N. L. Desai, "Reducing energy costs for ibm blue gene/p via power-aware job scheduling," in *Job Scheduling Strategies for Parallel Processors Workshop (JSSPP 2013)*, Boston, MA, 2013. [Online]. Available: http://www.mcs.anl.gov/papers/P5019-0913.pdf

[18] S. Oral and G. Shah, "Spectrum scale enhancements for coral," in *GPFS User Group*, 2016.