HPC Systems Acceptance: Controlled Chaos

Paul Peltz Jr., Parks Fields
High Performance Computing Division
Los Alamos National Laboratory
Los Alamos, NM, USA
{peltz,parks}@lanl.gov

ABSTRACT

Over the last six decades, Los Alamos National Laboratory (LANL) has acquired, accepted, and integrated over 100 new HPC systems, from MANIAC in 1952 to Trinity in 2016. These systems range from small clusters to large supercomputers. Each type of system has its own challenges and having a well established and proven test, acceptance, and integration plan is valuable to the site and vendor to expedite the process. The topic of systems acceptance itself is quite broad, and for the purposes of this paper, it will be mostly focused on the system's software and hardware components. Some discussion will be given to performance testing as well, but the purpose of this paper is to help HPC System Administrators with the acceptance process.

CCS Concepts

•Social and professional topics → **System management; •Software and its engineering** → **System administration; •Security and privacy** → Systems security;

Keywords

Systems Integration; Systems Testing; Systems Acceptance; Lessons Learned

This work has been authored by an employee of Los Alamos National Security, LLC, operator of the Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting this work for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce this work, or allow others to do so for United States Government purposes. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; however, the Laboratory as an institution does not endorse the viewpoint of a publication or guarantee its technical correctness.
This paper is published under LA-UR-16-26711.

SC - HPC System Professionals Workshop '16 Salt Lake City, UT, USA

1. INTRODUCTION

At some point in an HPC Systems Administrator's career a new system will be acquired by their site. Hopefully they will have some input into the Request for Proposal (RFP) and acceptance process since they will be the one responsible for making it operational and integrate it into their infrastructure. Thorough testing and vetting of the system hardware and software is important to make the integration process successful and for ensuring the ease of long term support of the system. The purpose of this paper is to go through some of the procedures that LANL has been going through over the years when acquiring new systems and share the lessons we have learned from this process. The acceptance process has evolved over time and through the work other sites have done as well[1][2][3]. This is in no way an exhaustive list of tests or items that must be tested, but hopefully it is a good starting point and can be expanded upon.

The acceptance process itself can be a long brutal fight with a vendor, or a quick and painless one. This usually depends on the complexity of the system, but not always. There have been times when even a commodity cluster solution has had systemic problems that has required a full system hardware change to correct. For most system acquisitions this is not the case, but the lesson learned so far with acceptance is that it can be an unpredictable and stressful process. Having deadlines to try and meet from site management and pressure from users to get this system operational only adds to the stress, but it is also equally important to not accept a machine that does not fulfill the mission of the site. There is a fine line there that must be trod, but ultimately the machine must be stable enough so that the system administrator is not getting paged all throughout the day and night. There will always be some problems, but no one wants to support a "lemon".

2. PROCUREMENT PROCESS

At LANL, the HPC system integrators are part of the RFP procurement committee. The system integration team is responsible for the system software and operating system aspects of the RFP. This is not always the case at other sites, but LANL finds it valuable to have this team input into the decision that is being made. After the RFP has been agreed upon by the procurement committee, it is then released so that vendors can propose a solution to the RFP solicitation. For large acquisitions, the RFP is released in draft form to allow vendors to give feedback. This feedback tends to be what hardware and software they are capable of delivering within the timeframe of the system acquisition. This also allows the vendor a head start on writing a response. The site will then review the vendor's proposed solution to the RFP. The procurement committee will then select one or two vendors that best meet the requirements and decide which is the best solution. This is where the lead HPC systems administrator should at the very least have some input into the decision. A prior history with the vendor or a lack of should be a consideration for the selection committee. Vendors will typically propose a full system software solution, and sometimes this is necessary, but not always. It is important to give input into these type of decisions because this is eventually the system that you will have to support. The procurement process for a HPC system typically looks like what is shown below in Figure 1.

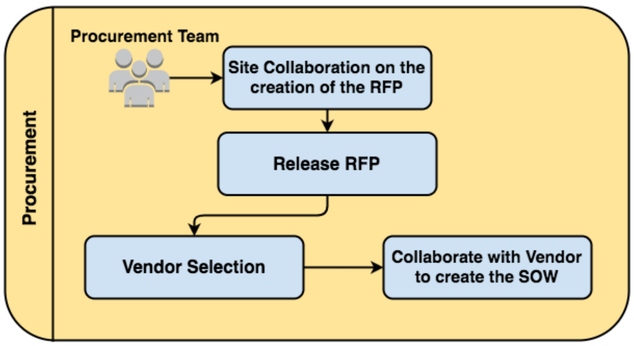


Figure 1: Procurement Workflow.

2.1 Statement of Work

A statement of work (SOW) is a contract between the vendor and the client, and its purpose is to make sure the client's mission is fulfilled by the vendor. This is typical in most business transactions, but there may not be a formal one in place for every purchase. If the site is not using a SOW, it would be advisable to work with a sister institution to see if they have one from a similar HPC system configuration and modify it for your site's needs. The SOW is the site's way to formalize with the vendor, through a negotiated contract, system and institutional expectations for the system to behave operationally and also that performance is at expected levels. The SOW is normally a very comprehensive contract that describes the system architecture, application performance requirements, and many other topics. It allows the site to contractually obligate the vendor to the RFP response that the vendor provided which will ensure that the vendor will deliver what was proposed in the RFP. The creation of the SOW is usually out of the purview of the system administrator, but hopefully they have some input into the inclusion of some important items.

One item to consider should be the homogeneity of hardware components in the system. This seems obvious, but sometimes vendors will change one part to a supposedly identical part such as a DIMM or power supply when the system is being built. This can lead to the system experiencing variable performance or erratic behavior. Different parts have different failure rates which can lead to higher interrupt rates if the new part is lower quality. The same principle applies to DIMMs as well, lower quality parts can exhibit higher parity failures which will affect performance due to the error corrections that would need to be done to fix those errors. Identical part supplies for RMAs should also be guaranteed for the warranty period of the machine. It is also important for the SOW to describe the expected performance/capability of each component of the system. For instance, the DDR memory speed, interconnect bisection bandwidth, and physical hardware description of every node. These details are important to have specified so that they can be tested and measured once the system is in acceptance. Another item of importance is the facilities requirements. There should be a description of the cooling (air and/or water) and power requirements for the system. There should also be a list of software provided with the system. This needs to include, if applicable, work load manager, cluster management software, compilers, debugging tools, etc. Security should also be listed as a requirement. The system software should comply with the site's security rules and those rules should be provided to the vendor.

One additional area that is extremely important is to have the various interrupt rates agreed upon by the site and vendor. These include Mean time between failures (MTBF), System Mean Time Between Interrupt (SMTBI), and Job Mean Time to Interrupt (JMTTI). The MTBF is necessary for the hardware components of the system. It is necessary for the site to know how often the vendor expects a given part to operate before it fails and having a parts cache appropriately sized to handle these failures. The SMTBI is the minimum expected uptime for the system before some event prevents the full system from being able to continue to run jobs. This can include network failures, parallel file system interrupt, or even if a significant number of the compute nodes become unavailable. The JMTTI is the minimum time that between a job failure. This is normally due to a hardware or vendor supplied software event. As the size of the system scales up having this information becomes more important because the likelihood of a failure becomes statistically more probable.

Some other items to consider are the following:

- Warranty/Support
  - Response Time
  - Length of Warranty
  - Part Replacement Inventory
  - Part Replacement ETA
- Monitoring and Logging
- Documentation
- HW/SW Upgradeability of the System
- Testing and Acceptance Plan
- High Availability/Failover

The original RFP for the system may have a lot of this same language in it, but try and be as comprehensive and specific as possible for the site's requirements in the SOW. Having this in place is important for ensuring the long term reliability and support for the system.

3. PERFORMANCE AND RELIABILITY TESTING

3.1 Performance Testing

Performance testing is a very common way to test the capabilities of a system and verify that the system can run at peak performance. These performance tests are usually done through synthetic benchmarks such as HPL, HPCG, STREAM, etc. These benchmarks are useful for testing the performance of a system as the system scales up in size. However, there is also value in running HPL on each individual node of the system. This is especially useful in the early life of the machine. LANL has found that running it on a single node basis on the entire system has value in many different ways. The first is that it tends to find the bad hardware in the system. Vendors normally run a burn in type of tests as well to find any bad hardware, but there is inevitably additional faulty hardware that is discovered by running single node HPL. Another value is that it can also discover underperforming hardware that is not necessarily faulty, but does not perform up to the same level as the other identical hardware components do. This is typically seen in the processor where there is normally some variability in performance, but LANL has seen greater than 10% performance variability out of some processors in the system as well. There are not always error messages that correlate to these out of spec parts either. This variability can be due to the manufacturing process or some other defect in the system. Either way, those parts are usually tagged for follow up with the vendor and replaced if necessary. Figure 2 below shows the distribution of HPL runs on a segment of the Trinity Haswell system in 2015. The far left of the graph shows a low point of approximately 750 GFLOP/s and the far right is the fastest node at approximately 820 GFLOP/s. The performance variation shown below can lead to performance variability in applications depending on what nodes they get allocated by the scheduler. Ideally, this figure would show an almost flat graph, and only a few outliers that are at the very left/right of the bulk of the nodes. Instead, this graph shows a slope as the performance increases across different nodes in the system. This issue was brought up and addressed during testing to better standardize performance across the system.

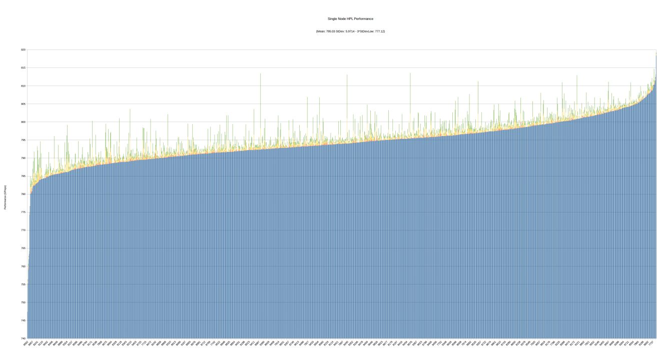


Figure 2: HPL Variation on Trinity Haswell Processors. X-Axis is a node number and Y-Axis is the GFLOP/s reported by HPL.

HPL will also detect other faults in the processor as well. LANL has seen instances of the HPL residual consistently being over what is allowed for the HPL result to be valid. Another useful benefit of the single node HPL run is that it is generally a good thermal test for the system. During this time the system is also monitored for thermal throttling events and if components are exceeding the manufacturers recommended operating temperatures. This is helpful to find the hotspots in the system and if there are components exceeding the operating temperature range then those are components that get tagged for follow up with the vendor as well. There are a variety of reasons this can happen, but LANL has seen instances of thermal paste not applied correctly on a swapped out part, fans not running at the correct speeds, or fans blowing the wrong direction and exhausting into the cold aisle. HPL is also good at testing the whether the system has successfully integrated into its facility. This test will demonstrate that the high-end power draw and cooling capacity of the system do not trip any breakers or overheat the facility. It is important for the system to stay within vendor and site recommended operating temperatures for stability and long term endurance of the system and its components.

LANL also does this type of testing on the memory subsystem as well with STREAM[4]. STREAM can help identify faulty DIMMs as well. STRIDE[5] is another useful memory benchmark as well. The OSU[6]/IMB[7] network test suites to test the high speed network. The network tests also help discover any bad hardware such as network cables, network cards, and switch ports. These tests also help discover any routing issues or switch misconfigurations that can lead to over subscription of a network switch. System Confidence[8] is also an effective tool at analyzing network latency across the system to find any potential congestion points in the network topology.

While synthetic benchmarks are useful for benchmarking the system, they rarely represent the typical workload for the system when it is in production. LANL typically uses representative applications to simulate the applications that are run when the system is in production. This is not always the case however and LANL usually has a select set of applications that are typical of the workload used on the system. These applications tend to put strain on various aspects of the system such as the network, I/O to disk, processor, and memory subsystem. The purpose is to find any potential bottlenecks in the system and address them with the vendor. This is vital to the testing process to verify those applications behave and scale as expected.

3.2 Reliability Testing

Reliability testing is also a key part during system acceptance. If the system is not stable then it does not matter how well it performs. It is important to have a test plan to verify that the system does not have any inherent weaknesses or common failure points. Hardware will fail over time and there should be some expectation that is defined in the SOW between the vendor and the client on how often this occurs. This is typically referred to as the mean time to interrupt (MTTI), and it is important to track the hardware failures during acceptance. There is typically an allowed time between job failures (the JMTTI) over the course of the reliability runs, but there should not be any system wide failures that happen in the SMTBI timespan. The duration of these tests and timeframes are usually negotiated with the vendor and it is important to make sure that the system is capable of achieving these metrics while under a heavy workload. The tracking of any failures during the reliability testing can help discover defects in the system that may need to be addressed. LANL typically asks the vendor to do a root cause analysis on these failures so that there is an explanation for each of them. Here are some things to be mindful of: thermal hotspots, uncorrectable memory errors, network congestion, component failures, and correctness of results. These are a few of the big areas that can suggest a trend of possible bad hardware or a misconfiguration of the system.

Another important thing to do is test for single points of failure within the system. This can be critical components such as the cluster "master" management box and high speed network switch. How well does the system recover from one of these component failures? If there are high availability/failover systems those should be tested as well to verify the system will continue to run jobs while one of the components fail. High availability and failover software is especially important because if it is not working properly, it can make returning the system to a healthy state even more of a challenge.

4. ACCEPTANCE PHASES

While going through the acceptance phases, it will most likely be necessary to repeat the tests at some point in the process to validate that system changes to address issues have not regressed application performance. LANL uses a test harness named Pavilion[9] in order to do this. Pavilion is not a test suite in itself, but has the ability to run any benchmarks or applications within its framework. The framework design makes pavilion quite extensible which allows anyone build and install the tests that the site wants to run and integrate them into Pavilion. This test harness also allows LANL acceptance team members to run and store results over time and reproduce the application runs within the test harness to verify system functionality and performance. It allows for a simple pass/fail metric checking as well as more complex output checking. It also has the ability to fill the job scheduler queue with jobs continuously so that the team members can launch the test harness and then just triage results afterwards.

The acceptance process is a challenge and a team is necessary to accomplish this. It takes a team of people dedicated to working with the vendor and the site's management to determine if the system meets the contractual agreement in the SOW. It requires a large coordination of effort to manage this process and should have the necessary people who have an attention to detail and are subject matter experts in their areas of responsibility. The team members or team leader depending on the size of the team should also have a good relationship with the vendor and have open communication channels with them as well. Communication is vital through this process between the site and vendor in order to make the acceptance process as smooth as possible and to quickly resolve issues as they arise.

4.1 Factory Trial Testing

For LANL, the purpose of the factory trial testing is to verify there are no systemic hardware issues. LANL has a basic testing harness with applications like STREAM, HPL, STRIDE, System Confidence, and OSU/IMB benchmarks to verify there are no hardware problems on the system. This allows LANL to verify the hardware is acceptable to ship. This step will hopefully avoid any type of "forklift" replacement that would be necessary if there is a critical issue discovered. This type of replacement could involve cabinet swap outs, node replacements, etc. These type of replacements are time consuming and extremely costly to the vendor. It is much cheaper to fix systemic hardware problems while still at the factory rather than at the customer's site.

4.2 Post Shipment Testing

The purpose of this testing period is to verify successful site integration, therefore after the entire system is shipped and installed on site, a rerun of the LANL testing harness is run at full scale on the system to verify that there was no damage during shipment or during installation at the site and also verifies the system is configured as it was in the factory. This also verifies facilities integration has been successful and the facility can handle the power and cooling load of the system. This process typically discovers and resolves any type of physical system problems before beginning the acceptance testing process.

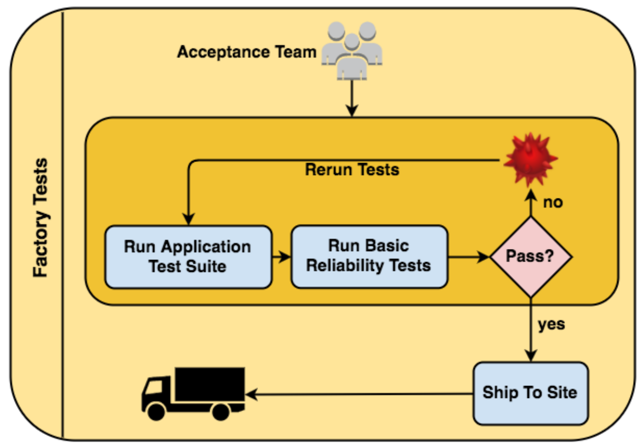


Figure 3: Factory Test Workflow.

4.3 System Acceptance Testing

System acceptance testing itself can be a long process depending on the complexity of the system. This process typically verifies that the SOW contract has been fulfilled and that the system is operational and performs as expected. In order to do this, a number of tests need to be completed on the system to verify that it meets that criteria. Full scale testing of important applications, such as the representative applications mentioned previously, are generally done during this time. LANL usually specifies metrics that a predefined set of applications must meet. The applications are graded on a capability improvement metric, which is defined as `(problem-size-increase x run-time-speedup)`[10]. LANL does not hold every system that goes through acceptance to these metrics, for example the commodity clusters are not, but the advanced technology systems do require this.

Full scale reliability testing is another area that needs to be tested again at full scale on the system. The reliability tests mentioned previously should be done at this time to verify the system will continue to operate without exceeding the SMTBI and JMTTI values. This value depends on the size of the system and what defines an interrupt however. For LANL's larger systems there is typically a two week reliability run in which a suite of applications are run on the system to verify it will handle a typical workload without an interrupt. These interrupts could be a switch failure, storage system failure, or a system software failure such as nodes not returning to an available state after a job is completed. Node failures are not considered a system failure, but a preponderance of node failures could constitute a system failure. The goal of this testing is to make sure the system will be available for its intended purpose for a period of time. If a system can not stay up through its normal operational window between maintenance times, then it puts a considerable strain on the users and administrative staff supporting the system.

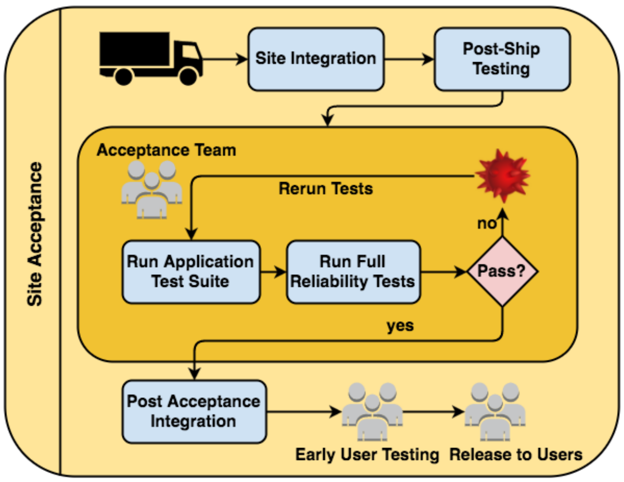


Figure 4: Acceptance Testing Workflow.

4.4 Regression testing with the Test Harness

The initial testing of the system with the testing harness also has additional value throughout the lifecycle of the machine. It is valuable to have a performance baseline to compare against when doing system upgrades or tuning. Continuous use of the test harness will allow the site to track the delta in performance over time. This can lead to discoveries such as the hardware does not perform as well near the end of its lifetime, operating system upgrades impose additional noise to the system which can lead to job performance variability, or kernel/driver upgrades can impose performance degradation/improvements due to changes in the way it handles memory or network traffic. LANL tries to do this type of regression testing throughout the lifecycle of some specific systems. The feasibility of doing this on every system becomes quite burdensome, but for the systems where high performance is expected it is tracked closely, especially throughout the acceptance process.

5. SYSTEM INTEGRATION

Hopefully by the system integration point of the acquisition process the system has been accepted, but that is not always the case. Either way the system needs to be integrated into the site's infrastructure to support users and application developers. The system is most likely still in a state of flux because the vendor and site are working together to get the machine operational, patched, and tuned. Capturing all of these changes can be a challenge to make sure the system is reproducible in case of a catastrophic failure. Ideally this should be done during acceptance as well, but it does not always work out that way. The sys admin should at least try and capture all of the files that are being modified and configured for the site in some way. LANL uses a combination of a revision control system and a configuration management tool to handle this. The revision control system will track changes over time. This is especially useful if there is a performance or behavioral change in the system because it allows the administrator to go back and see if there was a configuration change that could be the explanation of this behavior. The configuration management system will help keep the configuration in place to keep the configuration changes in a manageable state and to be able to restore the system in case of a failure.

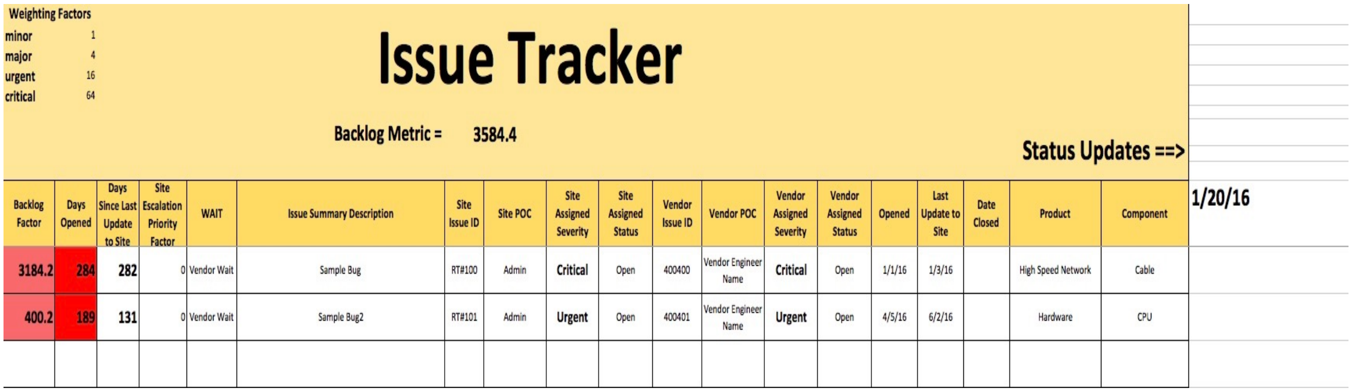
5.1 Vendor Software Testing

Vendors typically provide their own software stack for managing the system, and it is not always used by the site deploying the system. This is the case when the site has some expertise in deploying clusters using their own software stack. LANL uses its own system for managing the commodity clusters it deploys so there typically is not any vendor software to test. However, certain vendors, such as Cray, require their own software stack to be used on their XC product line.

In the case where the site is going to use the vendor provided software there are some areas of the software provided that should be verified and tested. One area that is important to verify is how well the system can be integrated into the site's infrastructure. This is a broad topic, but some examples that are important are system reproducibility, monitoring and security. System reproducibility is important because if there is a catastrophic failure of the system management node, the state of the system before the failure needs to be reproducible. The vendor software needs to be manageable and configureable so that it can be integrate into the infrastructure of the site. Some vendors like to just give one off scripts to address issues or to fix a problem on the system. This is generally undesirable and LANL requests that software be delivered packaged somehow, but preferably as an RPM. Monitoring is another piece of functionality that should be verified. Can the log data be pushed out of the system into the monitoring infrastructure of the site? Do all of the PDUs, switches, and other ancillary management devices log data out to a central logging destination? Security is also an important to verify with vendor software. The implementation of the vendor software may not meet the site's requirements for secure operations. This is difficult for vendors to implement so that every customer meets their compliances. The vendor software should still be functional even with security hardening in place. LANL has found it useful to audit the vendor's software and verify that it was implemented securely and that the implementation is compliant to the site and DOE standards. This is not always the case and should be addressed by the vendor when these issues are discovered.

5.2 Site Provided Software

The use of site system management software is much more prevalent in cluster purchases. Most of the commodity cluster solutions are capable of running just about any of the open source solutions available such as Warewulf[11] or nf-



Weighting Factors

minor	1
major	4
urgent	16
critical	64

Issue Tracker

Backlog Metric = 3584.4

Status Updates ==>

Backlog Factor	Days Opened	Days Since Last Update to Site	Site Escalation Priority Factor	WAIT	Issue Summary Description	Site Issue ID	Site POC	Site Assigned Severity	Site Assigned Status	Vendor Issue ID	Vendor POC	Vendor Assigned Severity	Vendor Assigned Status	Opened	Last Update to Site	Date Closed	Product	Component	1/20/16
3184.2	284	282	0	Vendor Wait	Sample Bug	RT#100	Admin	Critical	Open	400400	Vendor Engineer Name	Critical	Open	1/1/16	1/3/16		High Speed Network	Cable	
400.2	189	131	0	Vendor Wait	Sample Bug2	RT#101	Admin	Urgent	Open	400401	Vendor Engineer Name	Urgent	Open	4/5/16	6/2/16		Hardware	CPU	

Figure 5: A screenshot of the Issue Tracker.

sroot[12]/netroot[13]. There can be problems with sites deploying their own software, but this tends to be more of a vendor support issue. The vendor does not usually provide support for the site's software and there can be some disagreements on whether issues with scaling and performance are due to the hardware or software. This can be a sticking point at times and puts the responsibility on the site to justify that the issue really is a hardware problem and not software.

When LANL uses its own system management software it is generally a much easier acceptance process. Bugs and issues tend to just show up in the hardware and most of the time these are due to "infant mortality" of the components in the system. LANL's stress test harness usually detects most of these early on and helps prevent as many node failures when the system reliability tests are being performed.

6. BUG AND ISSUE TRACKING

LANL has found that having a bug tracking system in place is important for tracking issues and providing a historical overview of how successful the vendor was at resolving issues and what issues were never resolved over the lifespan of a machine. This is valuable information to have, especially when it comes to future purchasing decisions. The records kept will provide definitive metrics to use when evaluating a vendor's ability to not only deliver a system, but also supporting that system throughout its lifecycle. There are various mechanisms to do issue tracking, including using a sites existing ticketing system or bug tracking system. The method LANL is currently using, which has been newly developed, is to use a spreadsheet. The use of spreadsheets is nothing novel, but the spreadsheet that LANL uses is specialized to track specific data including how long bugs have been open, how long it has been since the vendor has communicated to the site, and how critical the issue is. This also tracks hardware, software, and product enhancement/feature requests. Separate tabs are also employed to track bugs gating certain milestones in the acceptance process. This helps communication between the vendor and the site so that expectations and issues are clearly presented and each side knows the state of the issues blocking acceptance. LANL finds it extremely valuable to utilize this to identify bugs that need to be fixed to pass acceptance. Not all bugs generated with the vendor will get fixed, but the ones that are important should be identified and isolated. LANL also sets up weekly meetings with the vendor to do a bug review of the top critical items. These meetings have been useful for making sure that issues are not stalled and waiting on the site or vendor to collect more information. See Figure 5 for a screenshot of the Issue Tracker.

7. CONCLUSIONS AND LESSONS LEARNED

The acceptance is a difficult and stressful process, but having a plan in place will help. Ideally, the acceptance process will be smooth and will not require additional negotiation with the vendor in order to come to an agreement on acceptance. When the acceptance negotiations stall due to an issue that the site encounters is where the plan, having a SOW, and having an issue tracker in place is vital. This is where negotiations must happen between the site and vendor to resolve the remaining issues. The goal for the site is to purchase a system that is productive and reliable so that the mission of the site is fulfilled by the use of the system over its lifecycle. A long and drawn out acceptance process is detrimental to both sides. It causes a continuous draw on the site's resources to continue to advance the acceptance process and also drain the vendor's resources that it could put towards advancing their product line. Therefore, it is beneficial to both the site and vendor to accept the machine, but only if the system meets the contractual requirements. When this does happen vendors sometimes will offer concessions to the site because of contractual obligations that were not fulfilled. Sometimes this can be a small development testbed, but this is not always desirable from the system administrator's point of view. This tends to be just another machine to have to set up, test, integrate, and support!

There are also other issues to consider which can help relieve the stress of acceptance and also ensure that the acceptance process is not rushed. LANL uses milestones with the vendor to make payments for the vendor's completion of these milestones. This could also be very beneficial for smaller sites as well so that the site can pay the vendor for portions of the system that are functioning as expected and not have one big lump payment. This also may be beneficial to smaller vendors as well who rely on a more frequent revenue stream. Only having a lump sum at the end of acceptance, especially if some systemic HW issue is discovered, can be devastating to a company and they could go bankrupt. This type of event, while it does not happen often, can be extremely disruptive to a site. Selecting a smaller vendor to provide a system can be risky and should be part of the risk analysis criteria for selecting a vendor. One common mistake that sites make is to try and accept a machine at the end of the site's fiscal year and the site must pay the vendor by the end of the fiscal year due to various types of penalties imposed by the site's funding organization. This is undesirable for a number of reasons. One is that the vendor knows that the site has to pay them by then and has a fixed end date for acceptance unless everything goes terribly wrong. The site would hope that the vendor would still be invested in continuing to improve their product. In some cases this is true and there is still an incentive by the vendor to continue to improve its product, but they also have other customers who are more than likely going through acceptance as well and need to divert resources to ensure that acceptance completes as well. The vendor also has fiscal boundaries that it wants to complete the acceptance process by to realize revenue they have earned which can also put pressure on the site to accept a system. LANL sees this frequently as well and while the site does try to work with the vendor to help them, the more important aspect is to only accept the system when it fulfills the contract and is able to service the mission of the site. The timing of acceptance is usually out of the HPC system administrator's control unfortunately, but it is a topic that should be discussed with management. If the system is accepted even though it is not truly operational just because of a deadline, then the site is the one who suffers the most because they will be responsible for supporting and operating a system that is unstable.

8. ACKNOWLEDGMENTS

The authors would like to thank the diligent LANL employees who help with the acceptance process and ensure the systems that we acquire are optimally configured and the system fulfills the mission of the laboratory. The authors would also like to thank Phil Romero who generated the HPL graph from his runs across the Trinity system.

9. REFERENCES

[1] S. M. Strande, P. Cicotti, R. S. Sinkovits, W. S. Young, R. Wagner, M. Tatineni, E. Hocks, A. Snavely, and M. Norman, "Gordon: Design, performance, and experiences deploying and supporting a data intensive supercomputer," in *Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the Campus and Beyond*, XSEDE '12, (New York, NY, USA), pp. 3:1–3:8, ACM, 2012.

[2] N. DeBardeleben, S. Blanchard, L. Monroe, P. Romero, D. Grunau, C. Idler, and C. Wright, *GPU Behavior on a Large HPC Cluster*, pp. 680–689. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[3] C. L. Mendes, B. Bode, G. H. Bauer, J. Enos, C. Beldica, and W. T. Kramer, "Deployment and testing of the sustained petascale blue waters system," *Journal of Computational Science*, vol. 10, pp. 327 – 337, 2015.

[4] Virginia, "memory bandwidth: stream benchmark performance results." https://www.cs.virginia.edu/stream.

[5] LLNL, "Coral benchmarks." https://asc.llnl.gov/CORAL-benchmarks/#membench.

[6] OSU, "Osu micro-benchmarks - mvapich." http://mvapich.cse.ohio-state.edu/benchmarks/.

[7] Intel, "Intel mpi benchmarks." https://software.intel.com/en-us/imb-user-guide.

[8] J. Lothian, "System confidence - a system latency analysis benchmark." https://github.com/jlothian/sysconfidence.

[9] LANL, "Pavilion: Hpc testing harness." https://github.com/losalamos/Pavilion.

[10] N. Wichmann, C. Nuss, P. Carrier, R. Olson, S. Anderson, M. Davis, R. Baker, E. W. Draeger, S. Domino, A. Agelastos, *et al.*, "Performance on trinity (a cray xc40) with acceptance-applications and benchmarks," *Cray User Group*, vol. 2, p. 4, 2016.

[11] Warewulf, "Warewulf - scalable, modular, adaptable systems management." http://warewulf.lbl.gov/.

[12] LLNL, "nfsroot - linux cluster diskless support." https://github.com/chaos/nfsroot.

[13] LLNL, "netroot - linux cluster diskless support." https://github.com/chaos/netroot.