Post-Mortem of the NERSC Franklin XT Upgrade to CLE 2.1

James M. Craw, Nicholas P. Cardo, Yun (Helen) He
Lawrence Berkeley National Laboratory
National Energy Research Scientific Computing Center
Berkeley, CA 94720
craw@nersc.gov, cardo@nersc.gov, yhe@lbl.gov

And

Janet M. Lebens
Cray, Inc.
jml@cray.com

ABSTRACT: *This paper will discuss the lessons learned of the events leading up to the production deployment of CLE 2.1 and the post install issues experienced in upgrading NERSC's XT4™ system called Franklin.*

KEYWORDS: XT4, Install, Configuration, Upgrade, CLE, Mortem

1. Introduction

NERSC is the flagship scientific computing facility for the Office of Science in the U.S. Department of Energy and a world leader in accelerating scientific discovery through computation. NERSC is located at Berkeley Lab in Berkeley, California.

2. The NERSC-5 Systems

2.1. Franklin

Franklin consists of 102 Cray XT4™ cabinets connected via a 3-Dimensional Torus high-speed switch. The system consists of a total of 9,592 quad-core Opteron nodes for computational work and 100 dual-core nodes for system services.

2.2. Silence

The NERSC 5 system includes a dedicated test system that is completely isolated from the main production system. This allows for specialized testing to occur without the fear of interruptions to the main production system. The system consists of a single Cray XT4™ cabinet with two full chassis. The configuration is designed to mimic the configuration of the main production system, although not at scale.

2.3. Gulfstream

NERSC upgraded Franklin to a quad-core XT between July and October 2008. The upgrade was done in phases in order to have maximum system availability and job throughput. During the transition period all users had access to the Franklin production system, which were, for a given phase, a mixture of dual- and quad-core nodes. The production system will experience brief periods of system unavailability while nodes were migrated into a separate "test environment" system where the hardware was physically replaced. The test system was called Gulfstream, which had limited access by selected users, who were able to stress test the nodes. After a 2 to 3 week period of testing, those quad-core nodes would be integrated back into the production (Franklin) system.

The last phase of this conversion allowed a unique opportunity for both NERSC and Cray. Both Cray and NERSC used Gulfstream for large-scale testing of CLE 2.1 before installing on the production system Franklin.

3. Cray Test Environment

Cray uses a product lifecycle process to manage its projects. This process includes seven gated product phases: Concept, Planning, Development, Verification, Introduction, Production, and End of Life.

While the OSIO Test group is involved in all of these phases for CLE, it participates significantly in the Development, Verification and Introduction phases to ensure a quality release.

3.1. Cray Test Methodology

During the Development Phase, OSIO Testing creates and executes feature tests for the new functionality included in the release. As new features are added in the daily "dev" build, regression tests are also run.

Weekly system stress and performance testing commences after significant functionality is enabled in the Development phase. Once "dev" is functionally complete and a set of split criteria are met, a release branch is created from "dev", starting the Verification Phase.

Verification Phase is the point at which system level testing begins in earnest. Testing continues on weekly release candidate builds (across a number of platform types), which include fixes to critical and urgent problems.

When testing is complete and a set of Verification Phase criteria are met, the Introduction Phase begins with creation of the Limited Availability (LA) release. It is at this point that Cray believes the release is ready for small production systems and seeks input from larger system customers via a Customer Test. Cray and NERSC partnered to perform such a Customer Test for the CLE 2.1 release. Once a successful Customer Test has occurred and the software has met the Introduction Phase criteria, the General Availability (GA) release is created and made available to all customers. Through this testing process, Cray helps to ensure correctness of Cray value-added functionality to the software stack, and that all software components of the Cray system—regardless of origin—function well together and scale to Cray-sized machines.

3.2. Cray Test Planning

During the Planning Phase of a release, the OSIO Test Group creates a Test Plan describing how each new feature will be tested. The group uses design documents and direct developer input to understand the internal workings of the feature and to understand how the user (system administrator or end user) will interact with the feature.

3.3. Cray Unit Testing

Unit Testing is done during the Development Phase by individual programmers, who ensure their new code works correctly with the existing code. Once Unit Testing is complete, the code is checked into the "dev" line of development.

3.4. Cray Functional Testing

The manual and automated feature tests that were created by the OSIO Test group are run during Functional Testing. A subset of these tests (both manual and automated) becomes part of the ongoing regression tests.

Features are usually tested individually, unless there are inter-feature dependencies. For high-risk features, such as new Sun Lustre versions, functional testing is performed before the feature is included in "dev". For lower risk features, which can be easily disabled, testing is performed on the "dev" system.

A total of 19 major features were tested by the OSIO Test group for the CLE 2.1 release.

3.5. Cray System Testing

System Testing is the main focus for the Verification Phase of the release, and consists of the following test types: Regression, Stress, Reliability Runs, Performance, Installation, and Exposure.

System test components used during Regression Testing, Stress Testing, Reliability Runs, and Performance Testing consist of a series of suites. The Operating System (OS) suite tests system calls, commands, and OS features. The Interconnect suite tests Portals, Seastar, and inter-node communication. The MPI suite contains MPI-based applications and test codes. Similarly, the SHMEM, and UPC test suites tests contain both applications and test codes. There is a CUST suite, which consists of 22 current customer applications. The Applications suite consists of over 500 older applications, many of which have found system problems previously. The PERF suite is used to specifically measure the performance of the system, and the IO suite exercises the IO and networking capabilities and the file system. Finally, since it is so important, the Application Level Placement Scheduler (ALPS) has its own suite. All suites are run in conjunction with released versions of CRI Moab/Torque and Altair PBS Pro batch schedulers.

These suites are used in different ways to accomplish specific goals of the test. The goal of Regression testing is to ensure that new code has not introduced a regression to previously existing functionality. Each test is analyzed as Pass/Fail. Approximately 6,450 test cases are run for each regression run, with the exact number dependent upon the architecture and which features are enabled.

The goal of Stress testing is to place a heavy load on the system lasting four to six hours to see how well system components interact. A stress run uses the same basic test cases as a regression test but with different core counts or memory sizes to put a load on the machine. In a six hour period, approximately 20,000 test cases are executed.

Once a release-in-progress can pass a Stress test, Reliability Runs begin.

The goal of a Reliability Run is to determine if the software can remain up for 72 hours under a heavy load without any overall system failure or node drops. Because our stress on the system is so severe—with frequent starting and stopping of shorter running jobs of varying sizes—passing a 72-hour run usually translates into a much larger Software Mean Time To Interrupt (SMTTI) at customer sites.

As the release progresses, the Reliability Run moves to larger system sizes in the Cray Data Center, culminating in runs on our largest in-house system. For CLE 2.1, that system was 16 cabinets. Because of the system load, Cray believes our Stress and Reliability Runs are able to simulate a system up to four times actual size. In other words, our 16 cabinet in-house test system for CLE 2.1 covered systems of up to 64 cabinets. Our internal testing, however, cannot substitute for running a release at a customer site to get real user and production load feedback, as well as test additional configuration options.

The goal of Performance testing is to ensure that the software meets performance targets set for the release. Depending on the release, these targets can be set to either demonstrate there is no regression in performance or to show an improvement in performance. Tests are run to measure node-to-node throughput, ping-pong, multi-pong, all-to-all, HPCC latency, and 8 node barrier times.

Additionally, the OSIO Test group runs the following the specific performance suites: HPCC 1.0, IMB, Pallas, Comtest (Sandia), Memory usage (service and compute nodes), and Lustre read/write. Also tested are boot and dump times, job launch times, MDS file creates and removes, and single and multi-stream reads and writes.

Other important testing includes Installation tests, where we ensure that both upgrade and initial installations will work for the new release. Installation testing is performed first by the Software group. Near the end of a release, Cray Service performs the testing to provide Software with an independent audit. The installation documentation is used for this testing, and feedback is provided to the Customer Documentation group.

The last type of testing is Exposure testing. During the Verification Phase, the weekly builds are run in a shared user environment. In addition to OS personnel, Cray Programming Environment and Benchmarking and Application groups become users of the system, exposing it to a larger set of users. Customer applications are run for correctness and performance. In addition to ensuring application "health", these users give feedback on usability of the release, and issues are reported via our bug tracking system.

As part of the system test process, tests are run on a variety of hardware configurations and systems to look for platform specific problems. CLE 2.1 was tested on Cray XT3-5™ single, dual and quad core systems with various memory sizes. Cray XT5h and XMT testing was also performed.

In addition, different software options are turned on and off to help identify software configuration issues. For CLE 2.1 these included VC2, Huge Pages, DVS, and PBS Pro / MOAB and Torque. Additionally, both Compute Node Linux and Catamount Virtual Node were also tested.

3.6. Cray Customer testing

Once we've passed our internal Cray testing and criteria, the Verification Phase is completed and the LA release is created. Very early on in this Introduction Phase, Cray partners with one or more customers to gain additional exposure and testing for the upcoming release. This is done with the understanding that customers will be able to find additional problems that Cray would not otherwise find before the release because of system size, the behavior of specific features in a real user environment, and real-world production workload. Since the test is done early in the Introduction Phase, Cray has the opportunity to fix many problems found before moving into the Production Phase and the General Availability release (GA).

The Customer Test is divided into three phases: Cray dedicated time testing, dedicated time "friendly user" application testing, and the production phase. The entire testing phase lasts from two to three weeks. Problems are reported via Crayport and Bugzilla. Daily meetings are held to track progress of the Customer test and any problems encountered.

The CLE 2.1 test schedule for Gulfstream at NERSC was 13 days in length. Time was scheduled in four-hour blocks and greater increments. These blocks of time were either Cray-only, NERSC-only or shared time. After installing the software, a NERSC security scan was run on the system. Next, Cray Operating System and I/O (OSIO) Testing executed tests to ensure the overall health of the system, including memory tests and basic regression tests. From there tests were scaled to the system size, and application, IO functional and some feature tests were run. NERSC then performed its workload checkout, which included both functional and performance tests.

It is at this point that friendly users were allowed on the system, as the workload test continued. Over time, Cray testing expanded its dedicated time to include testing of specific features for the release, including DVS and NFS, performance and stress testing. Checkpoint/Restart, a Limited Availability feature for the CLE 2.1 release, was also tested. About half of the time scheduled for the test included friendly user, NERSC workload and Cray application testing. No Franklin production testing was done as part of this test.

4. NERSC 2.1 Test Strategy

4.1. Silence

Before any software is installed on the NERSC production system, Franklin, it is installed and checked out on a single cabinet independent test system. This allows for procedural steps to be worked out and problems encountered to be addressed (and fixed) first.

The primary goals of this part of the testing is to:

1. Identify procedural issues
2. Become familiar with the upgrade process
3. Validate the new functionality achieved by the upgrade
4. Gain insight into the stability of the upgrade
5. Perform basic functionality tests
6. Perform limited performance tests

However, these tests are limited due to the small size of the test system. It is difficult to evaluate applications on this system due to its limited size. Franklin is significantly larger and problems induced by scale won't be encountered on Silence. Even with this limitation, the knowledge gained by this first test scenario is invaluable.

4.2. Gulfstream

The upgrade to XT 2.1 coincided with a rolling quad-core hardware upgrade being performed on Franklin. The system, Gulfstream, was actually a partition of the actual Franklin system. Gulfstream functioned as the quad-core "burn-in" system before moving the nodes back into the Franklin system. The system size changed, over time, as the hardware upgrade proceeded with a maximum size of 48 cabinets. It was decided to take this opportunity to run XT 2.1 on Gulfstream to further check out its viability. It was now possible to perform some application level testing as well as scale testing.

But there were limitations. Gulfstream contained only four I/O servers, which hampered any attempt to gain insight into I/O related issues. Furthermore, in order to create Gulfstream, it was necessary to convert the 3D Torus into a mesh in the X dimension. While Gulfstream brought us closer to Franklin's production configuration, it was not identical. The test user base that had access to Gulfstream was limited and the applications tested were limited also.

At the end of the Gulfstream 2.1 testing, in October 2008, there were no known major issues that would suggest we shouldn't upgrade Franklin to 2.1. So, Cray and NERSC decided to proceed ahead.

4.3. Franklin

No separate or special dedicated time was used on the new fully quad-core Franklin given the successful testing already performed on Silence and Gulfstream.

5. Franklin Post 2.1 Install

On December 3rd, 2008 Franklin was upgraded from CLE 2.0 to 2.1. Service nodes were upgraded to SLES 10 Service Pack 1 (from SLES 9.2) and Lustre was upgraded from 1.4.12 to 1.6.5.

The first problem encountered had the symptom that certain users could no longer connect directly to Franklin. This was believed to be a networking problem connecting to the system and was promptly investigated by NERSC's Networking Group. The problem though was identified as a bad netmask on the SeaStar network and was quickly corrected. The details were reported to Cray and subsequently released as Field Notice 5565.

Also, access controls into the system included the use of pam_access.so in the sshd PAM stack. The system contains a very large group, over 7000 entries, to control login access. This functionality broke, and an alternative method of using AllowGroups in sshd_config was employed to get around this issue.

5.1. Benchmarks Results

5.1.1. Kernel Benchmarks

NERSC consistently runs NAS Parallel Benchmarks (NPB): Serial NPB 2.3 Class B and Parallel NPB 2.4 Class D at 64 and 256 processors before and after CLE 2.1 upgrade. There were no noticeable performance differences for all NPB benchmarks, except the NPB 2.4, 64-way SP, which increased from 287 Mop/sec/process with CLE 2.0 to 306 Mop/sec/process with CLE 2.1. The SP performance has been seen to be very sensitive to compiler options and user environment changes.

There were also no significant performance differences in the memory benchmark STREAMS TRIAD operation using three different configurations: 60% memory of each node, 60% memory of each core, or full node.

There was significant latency changes resulted from underlying portals software change. Under CLE 2.0, within each quad core node, there is one favored core and three unfavored cores.

Measured latency between two nodes could land in three different buckets: favored/favored core pairs with the average of 5.46 us, favored/unfavored core pairs with the average of 6.09 usec, and unfavored/unfavored core pairs with the average of 6.74 usec.

Under CLE 2.1, there are no more favored/unfavored cores in each quad core node, the latency between different cores are averaged out to be 6.46 usec.

5.1.2. Application Benchmarks

Seven application benchmarks that represent 85% of NERSC workload, and also cover most frequently used programming libraries and programming languages, were chosen as Franklin application benchmarks and are run on the system periodically. These applications are CAM (climate model), GAMESS (computational chemistry), GTC (fusion), MADbench (astrophysics), Milc (QCD), Paratec (materials science), and PMEMD (computational chemistry).

Each application has a medium test case (run on 64 processors, except CAM on 56 processors) and a large test case (run on 256 processors, except CAM runs on 240 processors and

GAMESS runs on 384 processors). There is also an xlarge case for MADbench (runs on 1024 procs) and an xlarge case for MILC (runs on 2048 processors).

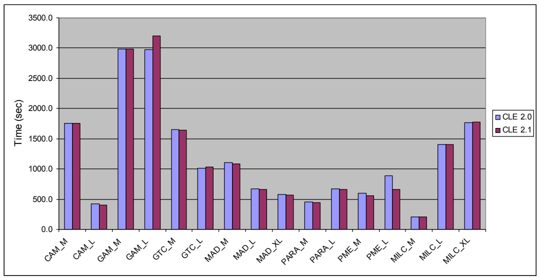

Figure 1 Run time comparisons for all application benchmarks under CLE 2.0 and CLE 2.1.

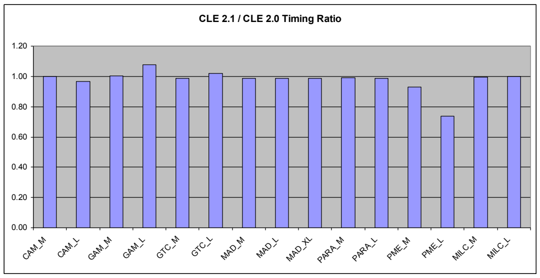

Figure 2 Run time ratio from CLE 2.1 to CLE 2.0 for all application benchmarks.

Figure 1 shows the actual run time comparisons for these application benchmarks under CLE 2.0 and CLE 2.1. Figure 2 shows the run time ratio for these benchmarks from CLE 2.1 to CLE 2.0. Most applications see within 3% performance differences, except GAMESS Large is 8% slower, PMEMD Medium is 7% faster and PMEMD Large is 26% faster. GAMESS slowdown may be contributed to a message passing library used in the application which is not quad core optimized. The speedup of PMEMD may be explained by large amount of the short communication messages in the code taking advantage of latency changes and the memory caching improvement in CLE 2.1.

5.2. Franklin Stability

All the testing at Cray, on Silence and Gulfstream, could not come close to the reality of having the full customer base utilizing the production system, Franklin, to its fullest capability. A number of stability issues began surfacing quickly after opening the system up to full user community.

The High Speed Network also appeared to be unstable. The symptoms all appeared to be congestion related issues. As part of this upgrade, the Virtual Channel 2 capability was enabled. Part of the instability was believed to be this. The situation appeared to improve when we turned off VC2. However, the system was still plagued by HSN congestion problems. These issues were serious and frequently hit the system. The problems manifested themselves in a number of ways including Lustre problems (or hangs).

5.3. MPT 2.0 verse 3.0 Apps

The upgrade release notes indicated that users needed to recompile their applications. The problem turned out to be more serious than simply recompiling. Applications compiled with MPT2 libraries have the potential to bring down the HSN causing the system to fail.

Once this was identified, a test could be performed to identify if applications were safe to run. In early January, this was turned into a wrapper around aprun, which performed the test and rejected applications that required recompilation with new libraries.

By the end of February, Cray had diagnosed and installed a new firmware patch for a CAM overflow condition that was causing system-wide outages.

6. Light at the End of the Tunnel

At this point the system was beginning to show signs of improvement. The patches installed to resolve SeaStar related issues and the wrapper for aprun that blocked MPT2 compiled applications appeared to be working. But by this point, the system still had a large number of individual patches installed and getting new fixes was becoming increasingly more difficult.

6.1. The Mother of Patch Sets (UP01)

XT 2.1.UP01 (update 1) contains a large number of fixes (100+). Included were most of the fixes that were currently installed as individual patches. After much debate, it was decided to apply UP01 along with selected Patch Sets (PS01, PS01a, & PS02). Careful analysis showed that this level of software would include all the fixes that were already installed as well as provide new fixes that were also needed.

There was much fear with installing UP01 after having gone through the problems with the XT 2.1 upgrade. The fear of introducing yet another bug that would just start the cycle over again lingered. Unfortunately this fear was realized after UP01 was installed in the middle of March on Franklin. A problem was immediately found when LDAP communicates with the NERSC central LDAP server it was being interrupted on downloads when a large group was encountered. A new RPM for PAM was found (and installed) which contained a fix for this problem. This issue was not discovered earlier on Silence, even though UP01 was installed there first. The belief was the groups on Silence didn't scale to the same size as some of the large groups on Franklin (another form of scaling not tested).

7. Summary

Best practices for the Cray Customer Test included the right level of planning and execution of the test itself. Having a single focal point to drive planning made joint Cray/NERSC planning easier.

The level of cooperation between NERSC and Cray was excellent.

The timeframe for the Cray internal test was appropriate—CLE 2.1 was ready for a Customer Test.

After nearly five months, the end result was a significant improvement in the software stability of the system.

Even with all of the shared pain, amongst Cray and NERSC staff, and even NERSC users, regarding the 2.1 upgrade of Franklin; the eventual benefits (2.1 stability and functionality) out weighed the pain. Many lessons were learned along the way.

7.1. Observations

A tremendous amount of effort was put into this evaluation particularly in preparations, actual testing and post production activities.

Key observations:

- Test duration at NERSC, was likely too short
- Lack of adequate I/O bandwidth on Gulfstream, to fully test I/O issues
- Current and up-to-date release notes are very important
- Private vs. public bugs posed an access problem for NERSC staff
- NERSC was unaware of LA versus GA differences
- Ability to partition the system very useful
- Some bugs slipped through the testing process
- Metric for success missing or not communicated effectively, both for NERSC & Cray
- Risk management process not explicitly planned for
- "Install" versus "upgrade"; both have issues (differences)
- Explicitly test the install process—have a customer validate the installation process prior to release
- Release notes need to highlight software that's "bad" to run on the system (recompile codes or not)
- Provide a method to not let users run "bad" software that crashes whole system
- Customer Test Program needs to disclose all known problems (even if not encountered at customer site) communicated to customer prior to production use

7.2. Lessons Learned

Cray took away many lessons learned, including ideas for improving in-house testing and changes to the Customer Test process. One of those lesson learned was that the level of Cray Programming Environment software supported for the OS release must be clearly identified before the start of the test.

Also, because of the large number of changes incorporated in CLE 2.1, including upgrades to SuSE SLES and Sun Lustre, the release would have been better named "CLE 3.0".

Another lesson learned was the assumption that a successful test on Gulfstream meant that CLE 2.1 was ready for NERSC production. That wasn't a good assumption by NERSC or Cray.

One area to be improved is Cray's determination of which problems found by the OSIO Test will and won't be found at customer sites. Also, Cray needs to track compatibility much more closely from release-to-release.

Cray Software routinely performs post-mortems on software releases to aid with process improvement for the next release. With CLE 2.1, for the first time a formal post-mortem was performed jointly with a customer after the Customer Test. This collaboration proved to be extremely valuable to Cray; NERSC developed many good suggestions for product and process improvement.

A major lesson learned by NERSC, even when testing is going well; don't schedule a major upgrade right before a major holiday period.

Other key lessons learned:

- Open, two-way communications are key to the project success
- Better define and share risks for NERSC/Cray
- Need to really run on a large "production" system (not just a set of test systems like Silence and/or Gulfstream) at a customer site before officially GA'ing 2.1
- Customer needs ability to review all outstanding bugs before deciding to go production (GA) – first large site
- Cray should provide a Tiger Team during the initial cut over to production use of a new release/upgrade
- Cray should consider loaning I/O H/W to help customers that volunteer for the Customer Test Program to better test I/O issues and performance
- Test 3rd party software at production levels, for example CSA/Moab/Torque combination was not tested before going production at NERSC (tested different or newer versions not current production levels)
- Reinforced the necessity to do large scale testing
- Set expectations for benchmark results
- Understand pros and cons to the install process verses upgrade process, upfront
- Need to provide users and incentive to test system (more free time)
- Better track system patches; ensure no regression due to missing patch(s)
- Cray needs to understand why problems slip through the test process
- Utility was needed for non-compatible software (MPT2 vs. MPT3)

7.3. Recommendations

Specific recommendations to add additional tests to the Cray test suite include:

- Injection of additional HSN traffic to simulate congestion
- 3D Torus test
- I/O stress test, e.g. IOR test

Don't implement a MAJOR upgrade right before or during a major holiday. The schedule was too optimistic for both NERSC and Cray.

Establish or better define Metrics of Success prior to starting the test for both NERSC and Cray individually and jointly.

Highly recommend increasing the size of Cray's test system to better validate scaling issues, beyond the current 16 cabinet test system.

NERSC and Cray both agreed to the immense value of the formal CLE 2.1 Post-Mortem and suggested to continue this practice with future test partners.

NERSC and Cray should formally and jointly write a "Post-Mortem" document.

Cray should share internal problems at each step of testing with Customer.

Finally, Cray should allow NERSC to share all of its CLE 2.1 bugs/SPRs with other interested sites.

8. Acknowledgments

The authors would like to thank the many Cray staff that helped with the Franklin upgrade, from pre-planning to post-mortem. Particularly the Cray On-site: Verrill Rinehart, Terence Brewer, Randall Palmer, Bill Anderson, and Steve Luzmoor; Jim Grindle, Brent Shields and the rest of the OSIO Test Group. Kevin Peterson, for excellent overall planning and as the Cray focal point.

The authors would also like to thank the NERSC staff that helped and worked long hours to make 2.1 a success on Franklin.

The NERSC authors are supported by the Director, Office of Science, Advanced Scientific Computing Research, U.S. Department of Energy under Contract No. DE-AC02-05CH11231.