Franklin: User Experiences

Yun (Helen) He, William T.C. Kramer,
Jonathan Carter, and Nicholas Cardo
National Energy Research Supercomputing Center
Lawrence Berkeley National Laboratory
Berkeley, CA 94720

ABSTRACT: *The newest workhorse of the National Energy Research Scientific Computing Center is a Cray XT4 with 9,736 dual core nodes. This paper summarizes Franklin user experiences from friendly early user period to production period. Selected successful user stories along with top issues affecting user experiences are presented.*

KEYWORDS: Cray, XT4, Franklin, NERSC, User Experiences

1. Introduction

1.1 The Role of Franklin at NERSC

NERSC is US Department of Energy's (DOE) keystone high performance computing facility that serves the needs of the DOE and open science computational research community on a broad range of scientific disciplines, including astrophysics, fusion, climate change prediction, combustion, energy and biology.

The addition of the Franklin system at NERSC, a powerful Cray XT-4, with nearly 20,000 processor cores and peak speed of 100+ TFlop/sec, puts in place the next "flagship" system at NERSC after our legacy IBM SP3 machine (Seaborg) was retired after seven years service in January 2008. The addition of Franklin increases the available computational time by a factor of 9 for our ~3,100 scientific NERSC users. It serves the needs for most NERSC users from modest concurrency (~100-200 processors) jobs to extreme concurrency (>8000 processors) jobs. We expect a significant percentage of time to be used for capability jobs on Franklin.

Figure 1 shows the AY2008 computer resources allocations at NERSC, categorized by different science categories. Table 1 lists the NERSC allocations for different types of projects from 2003 to 2008. The Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program provides computing resources and consulting support for a small number of computationally intensive large-scale research projects. DOE's Scientific Discovery through Advanced Computing (SciDAC) program brings together the nation's top researchers to tackle challenging scientific problems.

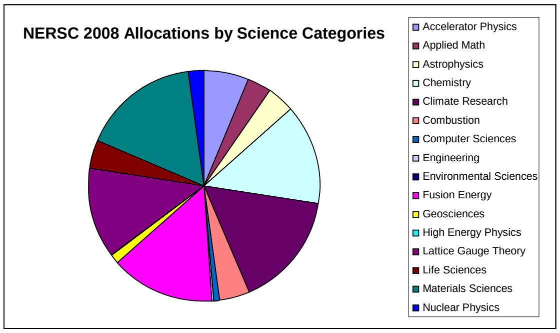

Figure 1. NERSC 2008 allocated computer resources by science categories.

Allocation Year	Production	INCITE & Big Splash	SciDAC	Startup
2008	275	11	47	40
2007	291	7	45	44
2006	286	3	36	70
2005	277	3	31	60
2004	257	3	29	83
2003	235	3	21	76

Table 1. NERSC allocations from 2003-2008 by project categories (Courtesy Kramer [2]).

1.2 Franklin

The NERSC Cray XT4 system, named Franklin, is a massively parallel processing (MPP) system with 9,660 compute nodes. Each node has a dual processor chip, and the entire system has a total of 19,320 processor cores available as computational resources for scientific applications. The system is named in honor of Benjamin Franklin, the first internationally recognized American scientist.

Each of Franklin's compute nodes consists of a 2.6 GHz dual-core AMD Opteron processor with a theoretical peak performance of 5.2 GFlop/sec. Each compute node has 4 GBytes of memory, and each service node (e.g. login node) has 8 GBytes of memory. The full system consists of 102 cabinets with 39 TBytes of aggregate memory. The theoretical peak performance of Franklin is about 101.5 TFlop/sec. Each compute node is connected to a dedicated SeaStar2 router through a Hypertransport link to a high speed network with a 3D torus topology which is designed to provide high performance, low-latency communication for MPI and SHMEM jobs [1].

Franklin uses two different operating systems. Full-featured SuSE Linux is run on service nodes (16 login nodes, 28 IO nodes, 4 network nodes, and 4 system nodes). A light weight OS based on Linux, Cray Linux Environment (CLE), is run on each compute node. CLE reduces system overhead, and is critical for the system to scale to very large concurrencies [1]. The parallel file system on Franklin is Lustre with approximately 350 TBytes of user disk space.

1.3 User Environment on Franklin

Franklin has SuSE SLES 9.2 Linux with a SLES 10 kernel on service nodes, and CLE for all compute nodes; Torque/Moab for batch system resource managements and batch job scheduling; ALPS utility (aprun) to launch compute node applications; and Lustre parallel file system.

The user programming environment includes PGI, Pathscale, and GNU compilers for Fortran, C and C++ codes; Portals communication layer that supports MPI and Shmem parallel programming models; a special port of the glibc GNU C library routines for compute node applications; a rich set of Cray LibSci scientific libraries (ScaLAPACK, BLACS, SuperLU) and ACML (AMD Core Math libraries: BLAS, LAPACK, FFT, Math transcendental libraries, Random Number generators, and GNU Fortran libraries); Cray performance and profiling tools (CrayPat and Cray Apprentice2); performance API (PAPI); and modules environment for managing system and custom built softwares.

1.4 NERSC User Services

The User Services Group is the user community's primary point of contact with NERSC. This group is responsible for problem management and consulting; help with user code optimization and debugging; strategic project support; documentation; online, remote, and classroom training; and third-party applications and library support. User Services also supplies user account and allocations management support; and maintains and makes enhancements to the NERSC Information Management system, including database management, client server code, and PHP web front end.

2. Franklin Early User Program

NERSC has a diverse user base compared to the most other computing centers. During the period of Franklin early implementation, configuration, testing and acceptance, previous experience had shown that a subset of NERSC users could help us to mimic real production work load, and identify system problems. So we launched an early user program, which was designed to bring early users in batches. We could work with small number of users and have more in-depth communications with them in the beginning, and gradually increase the user base as Franklin became more stable and was ready to host more applications.

2.1 Enabling Early Users

Early users were enabled on Franklin in seven different batches. Central NERSC staff (Batch 1) was granted access in early March 2007, followed closely by additional NERSC staff and a few invited Petascale projects (Batch 2). We call these users our Franklin “pre-early” users.

A solicitation email to all NERSC PIs and Project Managers was sent out on February 28, 2007. Information requested included: the readiness of the code to be ported on to Franklin, science goals for the runs at scale, the list of codes intended to be used, a brief description for scaling targets, and 3 to 4 user names that would need accounts. We reviewed, pre-approved or deferred each request based on if the user codes are easily ported to and ready to run on Franklin. These formed Batch 3 users, who are further categorized into different sub-batches to have a balance of science category, scale range, and IO need, etc. Each sub-batch has about 30 users, who were enabled when Franklin stability and capacity allowed us to do so. Batch 3a users were enabled in early July 2007. Batch 3b users were enabled in mid July. Batch 3c users were enabled in early August. Batch 3d users were enabled in late August. Batch 3e users were enabled in early September.

Batch 4 users are those who requested early access, but dropped the request or were deferred from the initial offer. They were enabled in mid September 2007. Batch 5 users are those who registered for the NERSC User Group (NUG) meeting and the following Franklin User Training on September 17-20, 2007. Batch 6 users are a few others who requested access later, followed by the massive group of Batch 7 users, i.e. all remaining NERSC users, who were enabled September 24-26, 2007.

2.2 Pre-Early User Period

The Franklin pre-early user period lasted from early March to early July, 2007. During that time, we only had Batch 1 (58 users) and Batch 2 (40 users) users on the system. This included NERSC staff, and five active LBNL research projects. We collected our first User Feedback from March 8-19, 2007. The Franklin-early-users email list was created for communicating with the users. Franklin web pages, containing documentation for compiling and running jobs on Franklin, and quick start guides for NERSC users who had previous experience on our IBM SP and Opteron Cluster platforms, were provided for reference. Staff training for Applications Programming and Optimization on the XT4 was conducted on April 16-20, 2007.

Franklin was not available for some extended periods during March 22 to April 3, 2007 for defective memory replacement; April 10-25, 2007 for file loss problem, and May 18 to June 6, 2007 for file system reconfiguration.

Examples of problems we encountered during this period were the file truncation issues and applications that make very heavy use of IO crashing the system. A simple IO test of full machine run was used to reproduce the IO problem. Other problems included wrong time stamps for output files, and some jobs failed with aprun reading directory time out. All the above problems have been fixed.

A Cray and NERSC collaboration called the "Scout Effort" was performed to bring in new applications to Franklin or new inputs to existing applications for potential problems exposure. A total of eight new applications and/or new inputs were examined: GAMESS, the Lattice-Boltzmann (LB) code, FLASH, SWEEP3D, GTC, MILC, POP, and LS3DF.

NERSC had a chance to evaluate CLE (then called Compute Node Linux) for two weeks in early June, 2007 [3]. NERSC users had access in the second week, which was also the same week CLE exited the development process. Please see more details of evaluating CVN and CLE and how a decision was made to go forward with CLE on Franklin in Section 6 of this paper. A quick start guide for CLE was written to provide our pre-early users the ability to quickly get on CLE. Besides the official procurement benchmarks testing, 12 user applications were examined.

2.3 Early User Period

Franklin early user period started from the enabling of Batch 3 users in early July, 2007. We then had about 150 users outside LBNL. Franklin compute nodes now running CLE.

User feedback was collected from August 19 to September 5, 2007. We followed-up with the user concerns raised in the feedback reports, and every communication was recorded in the NERSC trouble ticket database. Franklin user training was conducted from September 17-20, 2007. As of mid September 2007, top projects had used more than 3 million hours.

An issue during this period was NWCHEM [4] and GAMESS [5] codes crashing the system due to a flood of messages to the portals layer. A common thread between these two applications is that they both use SHMEM as the parallel programming model. The first patch provided by Cray trapped the shmem portals usage with an error exit, and the second patch addressed the issue by throttling messages traffic. Other problems included compute nodes losing connection after application started, jobs intermittently running over the wallclock limit, a problem related to a difficulty in allocating large contiguous memory in the portals level, specifying the node list option for aprun did not work, and aprun MPMD mode does not work in batch mode. All the above problems have been fixed.

User Quotas on Franklin were enforced on October 14, 2007. A quota bug existed so that no user quota could be set over 3.78 TB. This problem has been fixed although currently quota is disabled (see Section 5.6).

Queue structure regarding "regular" batch classes was simplified on October 14, 2007 to have only "reg_small", "reg_big" and "reg_xbig" classes to replace the original 10+ buckets. The change is transparent to users.

2.4 After Acceptance Early User Period

Franklin was accepted on October 26, 2007. The official announcement from Cray and NERSC was made on November 1, 2007. Franklin web pages were open to the public. Another set of user feedback was collected from November 1-26, 2007.

Users were productively running jobs without being charged to their allocations. We accommodated some very large applications and provided massive amounts of time during this period.

Batch queue backlogs started to grow. Some small to medium jobs showed long queue wait times. We modified the idle limit and global run limit to address these issues. Also users were advised to specify as accurately as possible the wall time in their scripts. The job throughputs were closely monitored for guidance on how to best make adjustments after system goes into production.

There was an inode quota bug that impacted users occasionally. Lustre behaves as if the users were over the quota while they were actually not. It did not allow users to cross over certain inode unit boundaries (multiple of its unit size, which is 250 on /home and 1000 on /scratch). This problem has been fixed.

We also discovered a problem resulting from the implicit barriers for MPI_Allreduce not functioning properly that generated a specific exit code 13. An SPR was filed to request Cray to check their MPI implementation for collective operations and the problem has been fixed.

Two other problems opened in September 2007 were user jobs intermittently over wall clock limit due to bad memory nodes left by previously over-subscribing memory jobs and multiple apruns simultaneously do not work in batch. The second problem has been fixed.

2.5 User Feedback

Overall, the early user program on Franklin was very successful. The overall user feedback was quite positive. Most applications were relatively easy to port to Franklin, the user environment (via modules) was familiar; the batch system was working well. Users got a lot of useful work done during the early user period. Many were able to run high concurrency jobs to tackle much larger problem sizes and model resolutions that were impossible before. Several got the equivalent of multiple years of run time.

Total of 51 science projects participated in the early user program. We worked with users in a way that the Franklin early user period benefited both sides. Users got a chance to get hands-on experience with a new architecture and a relatively lightly loaded system, and user jobs were free of charge from their allocations. Running the broader range of user applications was also good for helping us and Cray to find any potential problems in the system and develop fixes. We communicated with users often to help them to successfully port and run their jobs. Below are some selected early users feedbacks:

- "Franklin has been easy to use in both programming (porting) and running codes. I am very pleased and impressed with the quality of this machine. I believe it is an exceptional asset to the computational physics community in the US."
- "The friendly user period on Franklin has significantly impacted our science by allowing us to test the capabilities of our code and to establish that such high resolution simulations will be useful and constructive in understanding within-canopy turbulent transport of carbon dioxide."
- "I have been able to compile and run large scaling studies with very few problems. The queuing system has worked well and I have not had any problems with libraries, etc."
- "Overall, I am impressed with the performance and reliability of Franklin during the testing stage."

3. Franklin into Production

3.1 System usage

Franklin entered formal production, with user allocation charging from the start of 2008 allocation, January 9, 2008. Maximum runtime limit for production queues were increased from 12 hrs to 24 hrs. Other queue restructuring included setting the maximum tasks for the "reg_small" class to 1/8 of the machine, i.e., 2,416 compute cores, and enabling the "premium" queue for general use.

Figure 2 shows the daily Franklin usage from the start of the allocation year. Figure 3 shows the usage of the top 10 projects, which have used over 20 million CPU hours. Figure 4 and 5 show the Franklin usage by number of cores used and by science categories. Over 50% of machines hours are used by jobs using more than 2,048 compute cores.

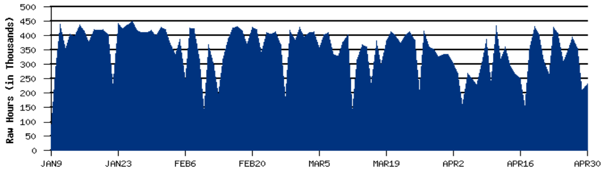

Figure 2. Franklin daily usage from January 9, 2008. Max 24 hour usage on this machine is 463,680 CPU hours.

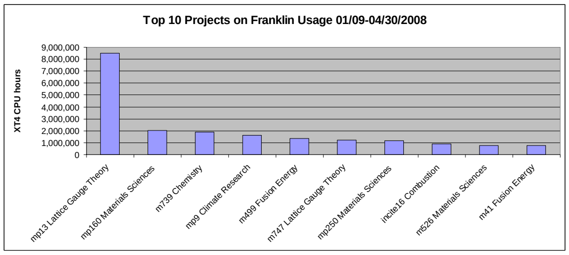

Figure 3. Usage of top 10 science projects on Franklin from January 9 to April 30, 2008.

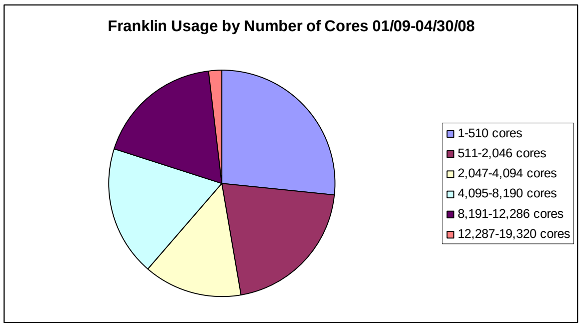

Figure 4. Usage on Franklin from January 9 to April 30, 2008 by number of cores used.

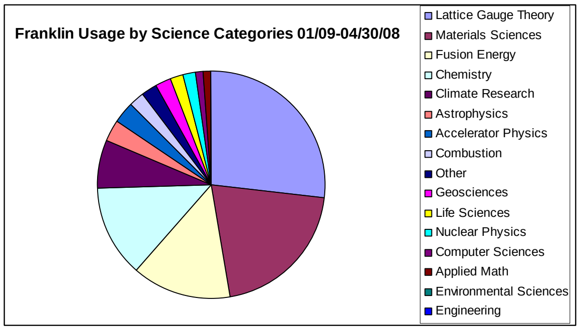

Figure 5. Usage on Franklin from January 9 to April 30, 2008 by science categories.

3.2 Scaling Reimbursement Program

NERSC and DOE launched the Franklin Scaling Reimbursement Program on February 19, 2008 with resources of 26 million MPP hours, equivalent of 4M Franklin CPU hours. This program is intended to help projects understand and improve the scaling characteristics of their codes and to be able to scale efficiently to at least 2,416 processors (1,208 nodes). This number corresponds to 1/8th of the computational processors, and NERSC has to meet the DOE metric that at least 40% of the time used on Franklin is by jobs running on 1/8th or more of its processors. The target projects for this program are those whose codes are already scaled to the 1,000 or so processors range, but are not yet typically run at 2,416+ processors.

NERSC evaluated the user applications based on the current scaling of their codes, their scaling bottlenecks and the work it might undertake to overcome the bottlenecks. We have now 23 users enrolled in the scaling reimbursement program as well as a number of "graduates" from the past year's program. Users and repos enrolled in the program will be partially reimbursed for jobs running at more than 2,416 cores. We will be working closely with some users, profiling codes etc., while other users will work more independently.

4. Selected Successful User Stories

4.1 Planck Cosmic Microwave Background Map

One of the NERSC users, Julian Borrill, from the Computational Research Division, Lawrence Berkeley National Laboratory, reported in October 2007 during Franklin early user period: "I am delighted to report that we have just successfully made a map of the entire Planck Full Focal Plane 1-year simulation (FFP-1). This is the first time that so many data samples (one mission year, 74 detectors, 3TB, 50K files) have been analysed simultaneously, and doing so has been the primary goal of our group's early Franklin efforts." Figure 6 shows a Planck Full Focal Plane (FFP) all frequency map using one year of Planck data from all detectors at all frequencies (100% data). The massively parallel MADmap code, a Preconditioned Conjugate Gradient (PCG) solver for the maximum likelihood map given the measured noise statistics was used to map the 750 billion observations to 1.5 billion pixels [6].

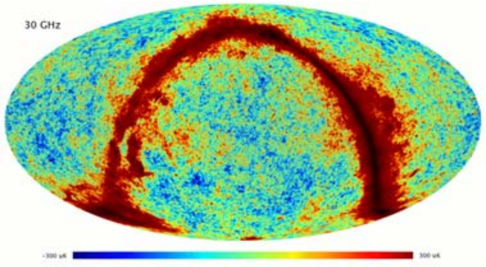

Figure 6. Planck Full Focal Plane (FFP) all frequency map. (Courtesy Borrill [6])

CMB data analysis is a computationally challenging problem that requires well-balanced, state-of-the-art and persistent HPC capabilities. This team also developed MADbench2 [7], which is a stripped down MADcap code that retains full computational complexity (calculation, communication & IO), but removed scientific complexity by using self-generated pseudo data. It is used as one of the application benchmarks in the NERSC-5 procurement that resulted in Franklin purchase. The Madbench2 running with 16,000 cores was the first user code that crashed Franklin during early user period and was used to develop a simple IO test and to validate the bug fix.

4.2 WRF Nature Run

A team from National Center for Atmospheric Research (NCAR), San Diego Supercomputer Center (SDSC), Lawrence Livermore National Laboratory (LLNL) and IBM Watson Research Center set the performance record for a U.S Weather model by running the Weather Research and Forecast (WRF) model on Franklin. They achieved a milestone of 8.8 TFlop/sec on 12,090 cores (Figure 7) – the fastest performance of a weather or climate-related application on a U.S. supercomputer. The team became one of the SuperComputing 2007 Gordon Bell finalists for high performance computing competition [8].

WRF model is a model of the atmosphere for meso-scale research and operational numerical weather prediction. The nature run involves an idealized high resolution rotating fluid on the hemisphere; at a size and resolution never before attempted – 2 billion cells @ 5km resolution. The science goal of the nature run is to provide very high-resolution "truth" (Figure 8) against which more coarse simulations or perturbation runs may be compared for purposes of studying predictability, stochastic parameterization, and fundamental dynamics. The initial input data is 200 GB, and the restart file size is 40 GB per simulated hour output [9].

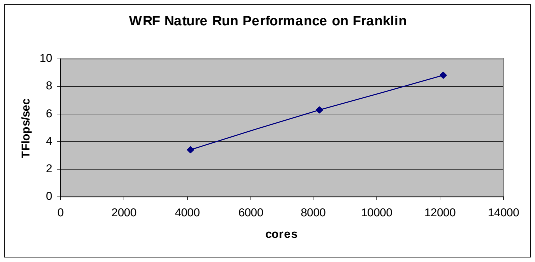

Figure 7. WRF Nature Run scaling performance on Franklin.

4.3 OS Jitter or Something Else?

This story is presented here to illustrate the positive and healthy vendor collaboration with NERSC users. As the user wrote after this work that "… that this Franklin research is some of the best vendor interaction I have had in my time using a supercomputer." and thanks for "taking us seriously and being careful, open and honest vendor collaborators."

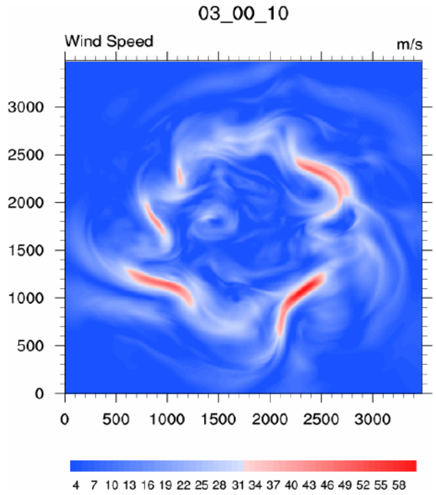

Figure 8. WRF Nature Run with 5km (idealized) resolution captures large scale structure such as Rossby Waves. (Courtesy Wright [9])

To study the potential of strong scaling of an Adaptive Mesh Refinement (AMR) Gas Dynamics Benchmark and AMR Poisson Benchmark, it is needed to study the effect of OS jitter. OS jitter can be thought of as the effect on application performance of the many OS interrupts that take place without synchronization across all the compute nodes involved in a simulation. The LBNL Applied Numerical Algorithm Group that is responsible for the Chombo AMR framework created an embarrassingly parallel benchmark by extracting a Fortran kernel from the AMR gas dynamics code, giving every processor the same amount of computation work, with no IO, no MPI messaging, no communication barriers, and no system calls.

It was expected to see almost perfect load balancing except for possible OS jitter effects. However, the initial result was somewhat surprising [10,11]. Figure 9 shows the histogram of run time for this stripped down AMR gas dynamic benchmark on ORNL Cray XT3 (Jaguar) CVN, Jaguar XT4 CVN, Jaguar XT4 CLE, and Franklin XT4 CLE.

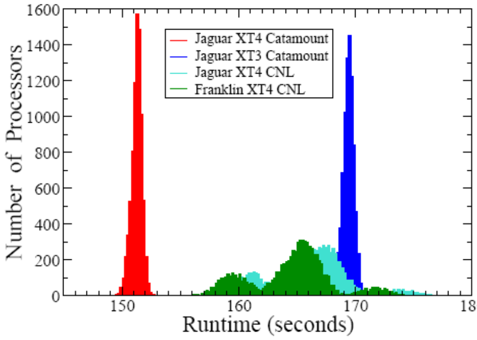

Figure 9. Histogram of run time on Jaguar and Franklin. (Courtesy Van Straalen *et al.* [10])

Jaguar CVN did not show the same tri-mode pattern. The initial suspect in this difference was the "OS jitter". More tests done on the Franklin test system with CLE and Jaguar CLE showed the same pattern.

The Chombo team met with Cray on-site support and discussed a few hypotheses. One of them being CLE has a more sophisticated, but stochastic, heap manager than CVN. The test of simplifying the memory allocator by using two environment variables that influence the operation of malloc, “MALLOC_MMAP_MAX_” and “MALLOC _TRIM_THRESHOLD_”, was carried out. Another test was to change the order of memory allocation and free operations. Chombo was internally tested with its own memory allocation routine “CArena”. This reduces the number of explicit malloc and free operations. It was confirmed that both methods (see 1-peak mode in Figure 10) were able to reduce the run time variation and to improve the overall performance.

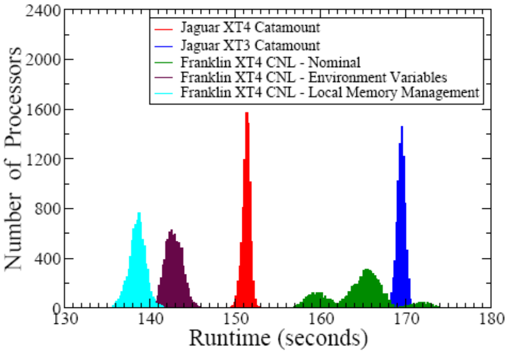

Figure 10. Histogram of run time on Jaguar and Franklin with CLE malloc environment variable setting and with AMR local memory management show a reduced single peak distribution. (Courtesy Van Straalen *et al.* [10])

We continue investigation why the run time variation on CLE, with the libc heap manager completed removed, is still twice larger than on CVN for the same hardware.

4.4 Large Scale Electronic Calculations

The LS3DF method is an O(N) method (compared to conventional O(N^3) methods) for large scale *ab initio* electronic structure calculations developed and optimized by Lin-Wang Wang *et al.* [12,13]. It uses a divide-and-conquer approach to calculate the total energy self-consistently on each subdivision of a physical system. This leads to almost perfect scaling for higher numbers of processors. LS3DF achieved 35.1 TFlop/sec, 39% of the peak speed, on Franklin using 17,280 cores [14]. LS3DF is capable of simulating tens of thousands of atoms, and is a candidate for petascale computing when the computing hardware is ready.

Figure 11 shows the scaling of LS3DF and its key component PEtot_F on Franklin. The speedup (and parallel efficiency) for 17,280 cores (from the base 1,080-core run) for PEtot_F and LS3DF are 15.3 (and 95.8%) and 13.8 (and 86.3%), respectively. This team has submitted the above results to the SuperComputing 2008 Gordon Bell category for high performance computing.

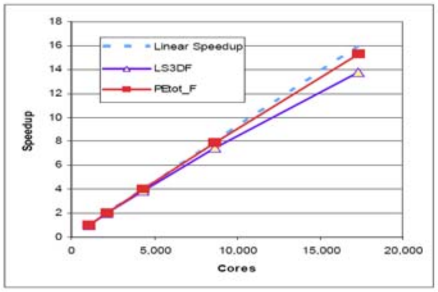

Figure 11. LS3DF and PEtot_F speedup. (Courtesy Wang *et al.* [14])

5. Top Issues Affecting User Experiences

5.1 Problems Filed and Fixed

Since the test system for Franklin (a single-cabinet XT-4 named “silence”) and thereafter Franklin itself arrived at NERSC, we have opened a large number of problem reports with Cray via SPR tracking. Figure 12 shows the accumulative number of SPRs opened and ended during the period. There is an increased gap between the number of problems opened and ended since October 2007 when all NERSC users were enabled, and Franklin was exposed to a larger user community and more diverse science load. The number of problems being solved is a great credit to the efforts of Cray development and support teams, however, there are still issues remain to be solved.

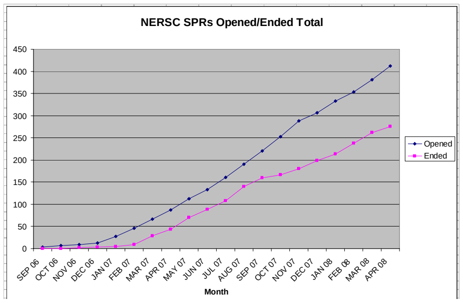

Figure 12. NERSC accumulative number of SPRs opened and ended (Courtesy Dan Unger of Cray).

5.2 System Stability

System stability is a top issue that affects user experiences. When a system crashes, all the user jobs fail automatically. If a system crashes too often, and mean time between failures (MTBF) is short, longer user jobs become unrealistic. One heavy user reported a 27% job failure rate from late March to April due to the combination of system outages, compute node failures, and wall clock limit exceeded (job hung or slow IO performances).

System stability issues also directly affect the system usage rate. For some significant periods during April 2008, Franklin was lightly loaded with little or no backlogs. Figure 13 shows the number of system wide outages from January to April by week. There are also about twice as many software related outages than hardware related.

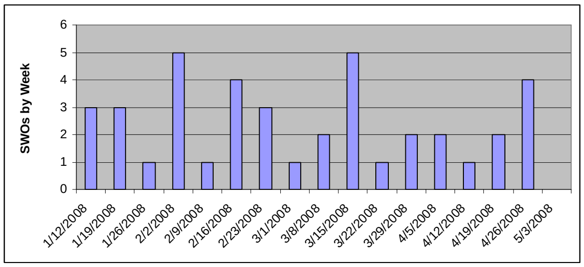

Figure 13. Franklin system wide outages by week since January 2008.

Cray recognizes the importance of this issue and is working hard with NERSC to improve the situation. For example, a patch was provided on February 13 to correct the top failure cause due to a Seastar heartbeat fault, and there has been no-reoccurrence of system crash due to this bug. The system has been up for the last 8 days since the scheduled maintenance of April 28, 2008.

5.3 Shared Login Nodes

Franklin login nodes are shared among users for interactive usages and for processing batch jobs by the system. User jobs launched without aprun (so they run on the login nodes), running large-scale parallel makes on the login nodes, other resource intensive scripts such as python or visualization packages, or the combinations of above have caused login node failures on Franklin.

When a login node fails, NERSC has made the decision to kill all the jobs that launched from that node as the execution host. This prevents these jobs using up requested wall clock time but not getting job outputs propagated back from the staging area.

We have now put essential points onto Running Jobs on Franklin web page to educate users that Franklin login nodes are shared resources, shell commands in a batch job still run on a login node, and only to use aprun to launch executables onto compute nodes. We have also set a limit for the process CPU limit to 60 min on the login nodes.

5.4 Hung Jobs and "Bad" Nodes

Users have reported their jobs on Franklin hung, especially large jobs, with aprun not actually starting (and of course no job outputs obtained), until the wall clock limit is exhausted.

These informally correlated with system problems or precursors to system crashes. Rerunning the code would usually correct the problem. An ultimate solution would be to decrease system related problems.

During March 18-22, 2008, there were an unusual number of jobs hung being reported. 7 nodes were found "bad" on March 20, and system was rebooted on March 22. We were only able to correlate 10 out of 16 hung jobs with some "bad" nodes detected a day or two later being used in these jobs. One hypothesis was that these nodes were already in unhealthy state when the jobs that used one of these nodes were running. Work is ongoing to better define this.

Hung jobs reported after March 22 were mostly related to memory overuse and Craypat profiling tool being used. One hung job was also correlated to a "bad" node being detected a day later. But there are still two hung jobs with no apparent causes. All hung jobs were reported to Cray for analysis.

5.5 Job Error Messages

Users frequently complained about not getting enough details of why their jobs failed: error messages are not being propagated back from the compute nodes to users' stdout or stderr files. Also, currently user jobs that run out of memory are not getting error messages about memory usage so users have no clue that their jobs actually failed due to memory reasons.

We are working with Cray on ways to improve accounting exit codes and more detailed error messages. This will help us to understand more of the currently unknown category of user job exits. For example, one of the paths to be explored is to provide memory utilization for compute node.

5.6 Quota Related Issues

Some quota related bugs that are severe enough to crash the system are preventing user quota limit being enabled on Franklin. NERSC has temporarily set the user inode quota to zero since January 4, 2008 and space quota to zero since February 4, 2008. The /scratch file system was filling up quickly with no controls, so NERSC user services had to contact users to ask them to reduce usage.

Although quota was set to be zero for all users, there have been several users hit the "quota exceeded" bug. At the time we asked the users to run specific quota commands and send the results back to understand more of the issues. It has been fixed in a released future OS version.

5.7 Slow Interactive Response Time

Users report occasional slow response, usually associated with heavy IO load on the system. Following one user's suggestion, the default "ls" option is now set to not using the color display, so there could be less file stat calls. Other issues related to slow interactive responses are pending more effective IO load monitoring and investigation.

5.8 Run Time Variations

Large variability of Interleaved or Random (IOR) benchmarks (developed by LLNL) has been observed since October 2007. After an OS upgrade (OS 2.0.39 and OS 2.0.44), performance degraded the first few days, then rates improved, but still with very large variability. The large IO variability issues are still under investigation. The possibility that several OSTs might have slower performance (to be proved) could be one of the reasons.

Some users reported jobs are slower than normal at some stages and then would pick up speed. Some users noticed large variation of application performance, sometimes with a factor as large as 10. At the same time, interactive responses are also slow. Information has been collected and sent to an IO team for further analysis.

6. CVN vs. CLE

The initial evaluation for CLE was carried out in early June for two weeks, and a decision [3] was made later to go on with CLE for Franklin additional evaluation and then for entering Franklin acceptance period and production period with CLE. Cray finished CNL development ahead of schedule, needed testing time at scale, so CLE was installed on Franklin the week it was released from Cray Develop to Cray Testing. The decision to move forward with CNL also mitigated risks and benefited DOE and other Cray XT sites for their system upgrade plans.

Another reason behind this was that since CLE is the path forward eventually, it would be better for our users not to have to go through an additional step of using CVN, then going to CLE. Other advantages and disadvantages of CLE *vs.* CNL were considered extensively.

The advantages of CLE over CVN are a bigger set of ported glibc GNU C library routines for compute node applications, so users have more control for their applications, and less the need to rewrite the source codes. Having more OS functionalities of CLE also leads an easier port from other platforms then to CVN. At least in some cases, compilations are also quicker. CLE provides (or a path to) other needed functions, such as OpenMP, pthreads, Lustre failover, and Checkpoint/Restart. CLE is also required for possible future quad core upgrade. CLE allows the potential for Franklin to be on NGF (NERSC Global File System) sooner. CLE gave us more options for debugging tools, such as DDT (Distributed Debugging Tool), which is now the operational debugger running on Franklin.

Some disadvantages of CLE over CVN are the increased memory footprint for OS so that it leaves less usable memory space for user applications. The difference is about 170 MB/node from our measurement out of about 3.66 GB/node of total usable memory for users. Our benchmark results showed a small increase in runtime variability for CLE (0.4%) *vs.* CVN (0.35%) for a set of scientific applications. Also, MPI latency for farthest intra-node is a little higher under CLE than CVN. There are rooms for improvement for future CLE OS releases.

Kramer *et al.* [3] summarized the holistic evaluation of CVN and CLE with 6 to 8 weeks exposure time for each OS on Franklin. CLE showed benefits over CVN in performance, scalability, reliability and usability, while showing only slight, acceptable decreases in consistency.

7. Other Topics

7.1 DDT vs. Totalview

Totalview is the standard choice of parallel debugging tool for Cray XT series. The adoption of Compute Node Linux enabled NERSC a chance to evaluate, to help in development, and finally utilize another much more cost-effective debugging tool, Allinea's DDT (Distributed Debugging Tool).

NERSC evaluated DDT from October to December 2007. One of the NERSC User Services consultants, Antypas [15], reviewed the differences between DDT and Totalview regarding user interfaces, features with different programming languages and parallelism models on three other NERSC platforms. DDT has a very similar user interface to Totalview, so the learning curve is rather small for NERSC users since Totalview has been our major parallel debugger on several other NERSC computing platforms. The major improvement from Totalview is the parallel stack view, parallel data comparison, and easy group control of processes. DDT's disadvantage being it is still relatively immature thus not all platforms are supported, and some features have limited support.

We have one floating license for 1,024 cores. This means one user can run at 1,024 cores or 1,024 users can run on one core or any combination of multiple users using a total of 1,024 cores. NERSC users are so far happy to use it as an alternative to Totalview.

7.2 ACTS PETSc vs. Cray PETSc

The ACTS (Advanced CompuTational Software) group maintains a collection of DOE ACTS softwares, including PETSc, SuperLU, Scalapack, TAO, HYPRE, etc., on all NERSC high performance computing platforms.

An ACTS PETSc module named "petsc" has been installed on Franklin before the Cray PETSc package was available. However, the Cray PETSc also has a default module name of "petsc", and is ahead of ACTS PETSc in the system module search path. So once we also had Cray PETSc installed, users depending on ACTS PETSc unknowingly started to use the default Cray PETSc, and there were a few things reported broken. For example, another eigenvalue library (SLEPc) is not working with the Cray PETSc since it was built with ACTS PETSc. Another library with the same problem is TAO, designed for non-linear optimization. TAO is not called via PETSc wrappers but instead they call PETSc routines.

Some advantages of Cray PETSc include more official support for the software; performance tuning for XT4 via Cray Adaptive Sparse Kernels (CASK); support on all three compilers (via a simple PrgEnv change) with the compiler wrappers pick up the correct libraries; has support for ParMETIS, HYPRE, and SuperLU packages within PETSc. Some advantages of ACTS PETSc and other related ACTS tools: has more varieties of PETSc modules for different versions: such as optimized, C++, and debug versions; software likely to be more up-to-date; and has more complete support for ParMETIS, HYPRE, and SuperLU standalone packages. It would be nice to have both installed on Franklin.

The same module name conflict issue would not only affect NERSC, but other DOE sites that ACTS softwares are officially supported. It would be hard to coordinate name changes for ACTS PETSc and other ACTS softwares. So we contacted the Cray Math Software Group for a possible Cray PETSc module name change. We learned that Cray doesn't really have the flexibility to change the names of the modules after it is released due to the difficulty of coordination with existing customers and that Cray has a strict name convention that a prefix of xt with a specific module name (xt-petsc) would imply a product that was available on multiple architectures.

One thing we noticed is that after loading the Cray PETSc, the library shown in compiler wrapper (%ftn –v) is –lcraypetsc, not -lpetsc. Maybe there is a chance to rename CRAY PETSc module to be cray-petsc or xt4-petsc?

We have since temporarily removed Cray PETSc from Franklin. Resolving conflicts between Cray PETSc and ACTS PETSc does not seem to be a short term item, and the current default setting of Cray PETSc could have caused some confusion of users who need ACTS PETSc and other softwares that depend on it.

There is another interesting thing related to PETSc libraries. One user reported flushing IO did not work on Franklin, which has generally proved to be working on most other user codes. It turned out that this user code used PETSc, and the PETSc link command adds libraries that the PETSc installation script automatically detected at installation. However, the compiler wrapper "ftn" has already included these libraries and the fact of explicitly adding them again changes the order in which they appear on the link command line. In this "flushing IO" case, the library "libpgftnrtl.a", is the culprit. The ACTs group has recompiled PETSc with a configure option that turns off the automatic detection of those libraries.

8. Summary

Franklin has delivered large amount of high performance computing resources to NERSC users during its friendly early user period and the early production period. NERSC users have been able to make progress and accomplish scientific goals that were impossible before. Users are continuously reporting problems they encounter on Franklin to NERSC user services, and we work with system groups, Cray onsite and remote teams to mitigate the problems. The support demand for this platform during these early days is still very high.

Two teams were formed at NERSC in mid April 2008 to address key Franklin general issues and IO issues. With the objectives of stability and quality of service, these teams are to provide Cray with information about issues NERSC users are experiencing on Franklin; to solicit updated information from Cray; and to make recommendation with a "path forward" to management priorities for both NERSC and Cray.

Some of the top priorities for the stability team are improving system stability, fixing quota related problems, propagating application error messages back from the compute nodes, and developing tools for better detecting bad nodes and identifying hung jobs.

The IO team's specific issues are slow responsiveness of login nodes, and the performance and variation of IO from batch jobs. There are suspicions that IO intensive jobs (either on the login nodes such as large tars and gzips or on the compute nodes with massive data reading/writing) are affecting other users interactive commands responsiveness and batch job performances. A high priority task for the IO team is to study and implement an IO monitoring tool to help understanding associations between user application related problems and the IO load at the OSS and OST layers.

Cray technical staff are involved along with NERSC on the two team efforts. We are looking forward to an improved Franklin user environment in various aspects and more satisfied Franklin users.

Acknowledgments

We would like to thank the Chombo team and the LS3DF team to allow us to include their newest results as our successful user stories in this paper. We would like to thank Cray teams (remote and on site) and also our NERSC colleagues for their hard work on Franklin. We would like to thank NERSC users for their valuable feedbacks.

The authors are supported by the Director, Office of Science, Advanced Scientific Computing Research, U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This work used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy.

References

1. Cray XT Overview and Cray XT4 Datasheet. http://www.cray.com

2. W. T.C Kramer. Cray's XT4 Integration and Progress at NERSC. Cray Technical Workshop North America 2008, San Francisco, February 26-27, 2008.

3. W. T.C. Kramer, Y. He, J. Carter, J. Glenski, L. Rippe1, and N. Cardo. Holistic Evaluation of Lightweight Operating Systems using the PERCU Method. Submitted to SuperComputing 2008, April 2008.

4. NWCHEM Home Page: http://www.emsl.pnl.gov/docs/nwchem/nwchem.html

5. GAMESS Home Page: http://www.msg.ameslab.gov/GAMESS/

6. J. Borrill. High Performance Computing For Cosmic Microwave Background Data Analysis: From T3E To XT4, Cray Technical Workshop North America 2008, San Francisco, February 26- 27, 2008.

7. L. Oliker, J. Borrill, J. Carter, D. Skinner, R. Biswas, Integrated Performance Monitoring of a Cosmology Application on Leading HEC Platforms, International Conference on Parallel Processing: ICPP 2005.

8. J. Michalakes, J. Hacker, R. Loft, M. O. McCracken, A. Snavely, N. J. Wright, T. Spelce, B. Gorda, R. Walkup. WRF Nature Run. Proceedings of Supercomputing 2007, Reno, CA, Nov 10-16, 2007.

9. N. J. Wright. Smoothing the Path to the Petascale Using Performance Modeling. Cray Technical Workshop North America 2008, San Francisco, February 26-27, 2008.

10. B. Van Straalen, J. Shalf, T. Ligocki, N. Keen, and W.-S. Yang. System Architecture Challenges for Massively Parallel AMR Applications. Submitted to SuperComputing 2008, April 2008.

11. P. Colella, N. Keen, T. Ligocki, B. Van Straalen. Performance and Scaling of Locally-Structured Grid Methods on Cray XT Systems. Cray Technical Workshop North America 2008, San Francisco, February 26-27, 2008.

12. L.-W. Wang, Z. Zhao, and J. Meza. Linear scaling three-dimensional fragment method for large-scale electronic structure calculations. Phys. Rev. B, 77:165113, 2008.

13. Z. Zhao, J. Meza, and L.-W. Wang. A divide and conquer linear scaling three dimensional fragment method for large scale electronic structure calculations. J. Phys. Cond. Matter, 2008. In press.

14. L.-W. Wang, B. Lee, H. Shan, Z. Zhao, J. Meza, E. Strohmaier,and D. Bailey. Linearly Scaling 3D Fragment Method for Large-Scale Electronic Structure Calculations. Submitted to SuperComputing 2008, April 2008.

15. K. Antypas Allinea DDT as a Parallel Debugging Alternative to Totalview. Lawrence Berkeley National Laboratory Technical Report, LBNL-62564, December 2007.