Scaling Efforts to Reach a PetaFlop

Kevin Peterson, *XT Software Director, Cray Inc.*

ABSTRACT: *This paper describes the activities leading up to sustained PetaFlop performance on the Jaguar XT5 system fielded at the Department of Energy National Leadership Computing Facility in Oak Ridge National Laboratory (ORNL). These activities included software development, scaling emulation, system validation, pre-acceptance application tuning, application benchmarking, and acceptance testing. The paper describes what changes were necessary to the Cray software stack to execute efficiently over 100K cores. Changes to Cray System Management (CSM), Cray Linux Environment (CLE) and Cray Programming Environment (CPE) are described, as well as methodologies used to test the software prior to execution on the target machine.*

KEYWORDS: XT5, CNL, Jaguar, PetaFlop

1. Introduction

In late 2008 Cray installed its largest supercomputer in the company's history at Oak Ridge National Laboratory (ORNL) and demonstrated that the system hardware and software delivers sustained PetaFlop (PF) performance. The hardware comprised 200 cabinets of Cray's XT5 Series supercomputer with ECOphlex liquid cooling. The software comprised the Cray Programming Environment (CPE), Cray Linux Environment (CLE), and Cray System Management (CSM) software. This Cray system ran applications across the entire machine (about 150K cores) and achieved sustained performance exceeding a PetaFlop.

Besides meeting ORNL's acceptance criteria, the Jaguar PF system was used as a demonstration vehicle for Defense Advanced Research Projects Agency's (DARPA) High Productivity Computing Systems (HPCS) program. Specifically, this milestone demonstrated that Cray system and software architecture can meet the goals of the HPCS program sponsored by DARPA.

This paper describes the activities leading up to this milestone, which were planned and executed to deliver and demonstrate a Petascale system with scalable software before the end of 2008. The key activities included system validation, pre-acceptance application tuning, application benchmarking, and acceptance testing.

2. Petascale Software Scaling Goals

There were several goals associated with this milestone. First and foremost, Cray had to develop management software, an operating system, and programming environment capable of running at Petascale. Second, Cray needed to deploy this software stack on a XT series system of sufficient size to validate the stack. Cray also had a goal of delivering production software for the Jaguar PetaFlop system fielded at the Department of Energy (DoE) National Leadership Computing Facility (NLCF) at ORNL. This system was shipped and installed at ORNL in the third quarter of 2008 and was accepted at the end of 2008.

Cray required the complete 200 cabinet Jaguar system for validation of software that was written or modified to support a system of such large scale. Existing software for the Cray XT series had a number of hard-coded limits that needed to be changed to support a larger number of network nodes and processor cores. Specifically, the Jaguar system exceeded the 16K node limit in the Portals network layer and the 64K Message Passing Interface (MPI) rank limit in the Cray Message Passing Toolkit (MPT) library.

Cray needed to evaluate the impact of changing these limits on Compute Node Linux (CNL) in terms of node memory usage and high-speed network (HSN) communication. Cray also had to validate changes to the HSN software that had been made to enhance the performance of the XT5 two-socket, eight-core NUMA

node. Changes had been made to distribute HSN interrupts among processors to balance core performance. These changes were validated on Jaguar PF. The ability for CNL to scale to tens of thousands of nodes and over a hundred thousand cores was a fundamental requirement for achieving the performance and scalability of applications across the system.

Prior to Jaguar PF, the largest systems that had successfully run CNL were the National Energy Research Scientific Computing Center (NERSC) Franklin XT4 system and ORNL Jaguar 250TF XT4 system. The following is the compute capability of those systems.

- NERSC Franklin:102 cabinets, 9680 nodes (1 socket dual-core), 19360 cores
- ORNL Jaguar: 84 cabinets, 6296 nodes (1 socket quad-core), 25184 cores

By comparison, the Jaguar PF XT5 systems had double the number of cabinets, twice the number of nodes, and six times the number of cores of any prior Cray system.

- ORNL Jaguar PF: 200 cabinets, 18688 nodes (2 socket quad-core), 149504 cores

The Jaguar PF system proved to be a worthy vehicle for demonstrating that Cray's software stack was capable of scaling and running applications with sustained PetaFlop performance.

3. PetaScale System Acceptance Criteria

In addition to Cray's goal of demonstrating system software capabilities and stability, there was also comprehensive criteria for the acceptance of the Jaguar system that needed to be met. The following is an excerpt from the Jaguar Acceptance Test document (JAT):

The activities of the Jaguar Acceptance Test (JAT) are designed to demonstrate the usability of the Cray XT based systems after each upgrade and after the 1000T-CS Baker system is delivered to ORNL. A usable system is defined to be one that first and foremost exhibits stability for the DOE Office of Science (DOE-SC) workload. Stability exists if the correct results for established DOE-SC applications can be generated repeatedly and reliably over a specified period of time. Secondly, the functionality of the system for developing and executing DOE-SC applications must be verified to operate correctly. Finally, the system must perform computation, communication and I/O at the specified rates. All test results must be demonstrated with the system configured for normal ORNL usage using system hardware configured for normal operation and software developed and released through Cray's software release processes[1].

The acceptance test was divided into two phases: initial hardware acceptance and a final integration acceptance. The initial hardware acceptance required that the cabinets power up and pass hardware diagnostics including running the operating system and key applications stressing the system. This Jaguar Acceptance Test-Hardware (JAT-HW) verified the correct operation of the processors (sockets), cores, memory, interconnect, and base software.

The final integration acceptance consisted of three tests for functionality (JAT-FT), performance (JAT-PT), and stability (JAT-ST). The functionality test was designed to demonstrate that system operations worked as expected. The performance test was designed to validate that the system met the test metrics. The stability test was designed to show that the system would run under load with acceptable variability.

The functionality and performance tests were run for 12 hours. The final stability test ran a mix of simulated code development activities and production simulations for one week. The following exit criteria were established for the stability test.

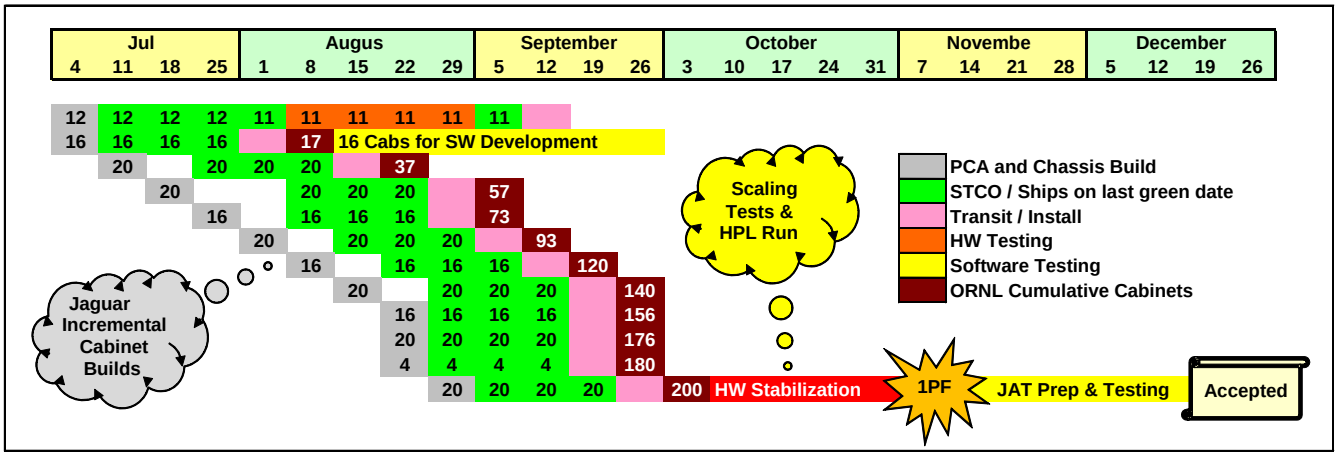
- Stability for 168 hours
  - Downtime for reboot and repair not counted toward duration
  - Overall availability of >90% with downtimes counted
  - Fewer than 2% of compute nodes failures over the test
  - Fewer than 3 system reboots during the test
- Job completion
  - 95% of applications submitted are completed successfully
  - 100% of applications are completed correctly at least once
  - 100% of completed applications generated the correct answer
- At least 4 applications
  - Run with a range of problem sizes
  - Run within 10% of metric
  - Scale to at least 80% of available sockets
- All applications run within 30% of metrics

A summary of application acceptance results is provided in Petascale SW Scaling Summary (Section 9).

4. PetaScale System HW Acceptance

Jaguar PF was the largest single system Cray has ever delivered. As with several other large Cray systems, the 200-cabinet system could not be fully built and tested in manufacturing. Rather, the system was built and tested in groups of up to twenty cabinets, and then shipped incrementally to the customer site. At ORNL, a combined Cray and ORNL site team installed and tested the incremental shipments of cabinets as shown in Figure 1.

Figure 1-Jaguar Manufacturing and Shipment Plan[6]



Due to the inability to test the system and its software at full scale before it was shipped, Cray had to test software scalability as the system was assembled. This was very effective as the system grew to about one hundred cabinets, about as large as the Jaguar 250TF, but with twice the processor density. However, the final assembly was accelerated and all two hundred cabinets were brought up without any software exposure at intermediate scaling increments. Fortunately, this final leap afforded Cray and ORNL additional time to stabilize the system and tune applications at full scale.

5. Software Scaling Efforts

In preparation for the final PetaScale configuration, Cray employed several techniques to test software on smaller systems while emulating the behavior of larger systems. To that end, Cray both over-subscribed resources such as memory and created multiple virtual resources such as nodes and MPI endpoints. Cray made every effort to validate the scalability of its three software platforms, the Cray Linux Environment (CLE), the Cray Programming Environment (CPE), and the System Management Workstation (SMW) prior to exposure on Jaguar PF.

Cray System Management Scaling

The SMW is the management interface for a Cray XT Series supercomputer, providing a single view of a system. Jaguar PF was about twice the size of existing systems in the field, as measured by number of cabinets, resulting in a fan-out ratio of 1:200 from the SMW to cabinet controllers. The XT5 compute blade also represented a doubling of cores per node because of the dual quad-core socket SMP configuration. Hence, the overall ratio of SMW to cores increased fourfold.

One of the powers-of-two boundaries that was crossed with Jaguar PF was the 16K node threshold. In preparation for the crossing, the development team did code inspection to identify any hard-coded limits or structure fields that were undersized. Fortunately, the management code had been written to support the architectural limit of the HSN of 32K endpoints.

Appropriate adjustments were made to the cabinet (L1) controller, and more importantly to the compute blade (L0) controller, to accommodate two sockets per node and four cores per socket. These changes had already been made during the product development of the XT5 compute blade. However, further work was done on the overall Hardware Supervisory System (HSS) to reduce message data and/or throttle messages where appropriate to avoid saturation of upstream components in the management hierarchy. This was especially true for the

console traffic between the L0 controller and the four Linux nodes that it managed.

Cray Linux Environment Scaling

Much of the operating system scaling work was done in the development of Compute Node Linux (CNL). This involved reducing the number of system services and daemons to only those that were required for our HPC customers' applications. These changes resulted in a significant reduction in operating system jitter and memory footprint and a return of cycles and space to the user. By reducing the overhead, CNL performed well at large scale. The CNL pioneering began on the NERSC Franklin system and incrementally improved within their production environment. Subsequently, it was tested on the ORNL Jaguar 250TF system where additional CNL tuning and scaling work was done as a proof of concept for Jaguar PF. Soon thereafter, it was installed on Jaguar 250TF for production work.

Several areas within the operating system environment were modified in anticipation of the Jaguar PF deployment. Specifically, Portals and Application Level Placement Scheduler (ALPS) were changed, though these changes were driven more based on changes to the node architecture than purely for scaling. Additionally, InfiniBand support was included in the CLE based on ORNL's requirement to support site-wide shared storage and InfiniBand attached storage.

Portals Constraint Based Testing

On Cray XT systems with the SeaStar network, Portals is the software layer that supports programming models such as MPI, Cray Shared Memory Access library (SHMEM), and Aggregate Remote Memory Copy Interface (ARMCI). Typically, the Portals interface is transparent to applications and users. However, to artificially stress the system, the Portals resource settings were opened for manipulation by developers and testers to permit constraint based testing. The strategy was to create tests that put pressure on one or more of those resources, and then to reduce the resource setting to the point where the available resources are exceeded. The premise being that this would exercise code paths in Portals error recovery that might only be seen on much larger systems and applications, hence, uncovering issues that could be resolved prior to execution at large scale.

A collection of tests was used for constraint based testing. On each test pass, a particular Portals resource was set to a constrained value, and all of the tests in the test suite were run. Only one Portals resource is constrained at a time, and all tests are run at every constrained value. This was done for simplicity and completeness.

During this testing, no problems were found by constraining certain resources such as user_tx_credits or rx_pendings. However, constraining other resources such as message buffers or tx_pendings resulted in Portals errors. In these cases, appropriate tuning of these resources prior to scaling on the Petascale system prevented these errors from surfacing. In these exercises no "hard" failures, such as node or system crashes, were seen in any of the tests.

ALPS and Job Scheduling

ALPS was already designed to accommodate systems of the scale of Jaguar PF. In fact, ALPS can handle much larger systems due to its n-way fan-out of shepherd processes.

The team made changes to ALPS to support the XT5 dual socket quadcore cc-NUMA node. First was the change to support 8 cores per operating system instance. Additionally, the concept of NUMA nodes was created along with CPUset support for assigning processor and memory affinity. Last were changes to the external interface to expose these changes to job schedulers such as Moab.

Additionally, a test version of ALPS was created to oversubscribe nodes. This was to enable testing the Cray MPI library at sufficient scale to cover the Jaguar PF system. This specialized version of ALPS was run with the modified MPI libraries on the Jaguar 250TF system to validate the scaling changes.

Lustre & InfiniBand scaling

Prior to the Jaguar PF system, the primary storage solutions for Cray XT series systems were based on Fibre Channel. ORNL required an InfiniBand solution on Jaguar PF in order to connect with a site-wide shared storage architecture.

As a result, Cray integrated the OpenFabrics Enterprise Distribution (OFED) InfiniBand stack into its Linux environment. Two InfiniBand storage solutions were supported. About half of the XT service nodes were configured to support fabric-attached InfiniBand storage with traditional Lustre Metadata Servers (MDS) and Object Storage Servers (OSS). This configuration was used throughout the acceptance period. The bulk of the remaining XT service nodes were set aside as Lustre routers to the shared Lustre service. The final acceptance of this I/O configuration is targeted for the first quarter of 2009.

6. Cray Programming Environment Scaling

There were several areas in the programming environment that needed work to scale applications across the machine. The team looked at MPI, SHMEM, and Global Arrays.

MPI Scaling

In preparation for the Jaguar PF system, changes had been made in Cray's MPI Toolkit (MPT) to scale to 150K ranks. The development team was given access to ORNL's Jaguar 250TF XT4 system with 7832 nodes of quad-core processors (32256 cores). In order to emulate more cores, the ALPS team provided a debug feature to over-subscribe each node. To validate that that MPT would operate correctly at 150K ranks, each node was oversubscribed by a factor of six or effectively running 24 processes on each node.

Launching a 156,000 process MPI "hello world" job demonstrated that basic MPT mechanics functioned at that scale. Several "multi-pong" tests were run at different process counts. This kind of test uses MPI_Send and MPI_Recv to transfer data messages back and forth between each pair of processes. The data values are modified and verified for each pass. The largest multi-pong test case used 180,000 processes, each sending one thousand 16KB messages between the pairs. MPI collective tests were written and run at 150K processes to test MPI_Allreduce, MPI_Bcast and MPI_Barrier functions, as well as an Intel MPI Benchmarks (IMB) Reduce test at 100,000 processes using message sizes from 0 up to 4096 bytes.

During the testing, the MPT initialization sequence COLL_OPT_ON had to be disabled due to excessive run time. This COLL_OPT_ON issue was debugged and resolved before acceptance. The MPI_Allgather algorithm within it was rewritten to improve performance at large scale. Access to a large scale system, albeit partially emulated, was critical to resolving issues prior to system acceptance testing.

SHMEM Scaling

Similar to MPI, SHMEM had a hard limit on the number of Processing Elements (PEs) it supported of 32K. The constant SHMEM_LOG_MAXPES was previously defined to be 15, representing 32,768 PEs (2^{15}). This user-accessible constant and other dependent constants define the sync space needed for barriers and the work arrays needed for various collective operations. In order to provide sufficient headroom for future generations of Cray systems, SHMEM_LOG_MAXPES was increased to 24, representing 16,777,216 PEs (2^{24}).

The SHMEM library also uses SHMEM_LOG_MAXPES and associated constants when accessing the user's sync space. Therefore, it is necessary that the value used in the user code matches the value used in the library. As an aid to the user, a check was added to the 3.1.0 library to detect if the user code was compiled with the old value. In these situations, the job is aborted and an informative message to recompile is displayed. This prevents the library code from overrunning the user space. Using an old library with user code defining the new limit will work, because the library would use the smaller limit and not overrun the user's array, but the user would be limited to 32K PEs.

Global Arrays Scaling

In order to run the NWChem computational chemistry package efficiently at larger scale, significant changes needed to be made to the Global Arrays software stack. First of all, a decision was made to remove SHMEM from the software stack and port ARMCI directly to the Portals interface. As a result of this change, modifications were made to Global Arrays to use a launching mechanism in Cray's MPT library. These changes provided a more efficient software stack, and a proven and scalable launch mechanism.

During debug of Global Arrays at scale, a performance issue was discovered in and out of band communication protocol on the compute node side. As a result changes were made in the Portals software to replace a polling loop with a block, sleep, and wake event. This significantly improved performance of Global Arrays.

7. Pre-Acceptance Work on Jaguar PF

As the Jaguar PF system was assembled onsite in groups of 16 to 20 cabinets, Cray's installation team tested the system at incremental scale. Note that groups of cabinets had been tested together at Cray's manufacturing facility prior to shipment and passed a rigorous fifty-hour stress and stability test. This made the job of testing joined groups on site much easier and allowed the focus to be on scaling issues.

Hardware Stress & Stability Testing

The primary goal of the Cray site team was to stabilize the system as quickly as possible as it was assembled so that the final combined system testing would be as short as possible. The team ran basic diagnostics on individual cabinets, and also ran three key applications that had proven useful in stressing the system and identifying marginal parts by accelerating infant mortality. These key applications were Intel MPI Benchmarks (IMB), High

Performance LINPACK (HPL), and S3D for hardware stability.

IMB was used not only as a functional test and benchmark, but also as an HSN exerciser. HPL was used to stress the system and provide maximal loading, which was useful in both power and performance measurements. S3D was used because it stressed both the memory subsystem and the HSN.

HPL and Autotuning

During the scale out of the system, Cray employed autotuning of HPL at each of the system increments. The basic idea behind autotuning is the use of automated techniques to develop platform-specific optimizations for benchmarks, libraries, and applications. Autotuning automatically explores a wide input parameter space, code optimizations, and problem configurations to find the best performing set for a given benchmark, library, or application. This process was particularly effective for HPL.

One autotuning approach was to predict end-performance early into a run, and then abort minimally performing trials. Another approach was to tune certain size-independent parameters on a subset of the system, allowing multiple runs to be tiled across the system to tune multiple parameters concurrently. Without these capabilities, tuning the entire parameter space would have taken far too much time to explore manually.

Hence, autotuning was enormously helpful in tuning HPL to each of the system increments within the short window of operation at each scaling step. It also made it possible to compute a parameter set that would enable a PetaFlop HPL run over the entire system and within the mean time between failure (MTBF) window. This was critical for a system of this size where the MTBF is measured in hours, not days.

Scientific Application Tuning

More important than HPL benchmark tuning, several large application programs were ported to Jaguar PF and tuned to achieve world record-breaking performance levels in approximately one week. While this achievement was accomplished with more traditional performance tuning techniques, it reflects the stability, portability, and performance of the Cray software stack, even at extreme levels of scaling. Specifically, the Cray Linux Environment was robust at 19K nodes and the Message Passing Toolkit performed well at over 150K MPI ranks. Although these application codes had been run previously on smaller Cray XT platforms, they were restructured (data structures, communication and I/O patterns, etc.) for Jaguar PF. The Cray performance tool suite proved effective at tuning these applications for PetaFlop execution.

Pre-Acceptance Application Results

The Jaguar PF system was assembled and stabilized quickly and in time to submit application results for the SuperComputing 2008 conference. One of the highlights of this effort was the Jaguar PF High Performance LINPACK (HPL) run, which achieved 1.059 PetaFlops or 76.6% of its peak 1.3814 PetaFlop performance, scoring a close second in the Top 500.

In addition, a number of real science applications were run in the remaining days before SC08 and demonstrated unprecedented performance. A summary of these applications and their performance on Jaguar PF is shown in Table 1 from an SC08 presentation by Arthur S. Bland of ORNL.

Table 1 - Scientific Applications Run Prior to SC08 Conference [2]

Science Area	Code	Cores	% of Peak	Total Perf	Notes
Materials	DCA++	150,144	97%	1.3 PF*	**Gordon Bell Winner**
Materials	LSMS/WL	149,580	76.40%	1.05 PF	
Seismology	SPECFEM3D	149,784	12.60%	165 TF	**Gordon Bell Finalist**
Weather	WRF	150,000	5.60%	50 TF	
Climate	POP	18,000		20 Sim yrs / CPU day	
Combustion	S3D	144,000	6.00%	83 TF	
Fusion	GTC	102,000		20 Bil Particles / sec	
Materials	LS3DF	147,456	32%	442 TF	**Gordon Bell Winner**

Several of these scientific applications demonstrated the work of ORNL teams and received awards and recognition at SC08 in November, 2008.

The ACM Gordon Bell Prize for Peak Performance was awarded to the team of Gonzalo Alvarez, Michael S. Summers, Don E. Maxwell, Markus Eisenbach, Jeremy S. Meredith, Thomas A. Maier, Paul R. Kent, Eduardo D'Azevedo and Thomas C. Schulthess (all of Oak Ridge National Laboratory), and Jeffrey M. Larkin and John M. Levesque (both of Cray, Inc.) for their entry, "New Algorithm to Enable 400+ TFlop/s Sustained Performance in Simulations of Disorder Effects in High-Tc." [5]

8. Petascale System Acceptance Work

Jaguar PF completed the Jaguar PF acceptance testing for hardware (JAT-HW) shortly after the entire system was assembled and diagnostics had been run across the machine. By the time the SCO8 benchmarking activities had completed, Jaguar PF had reached a stability level suitable for acceptance testing. However, prior to starting the acceptance work, the Cray site team upgraded the operating system to Cray Linux Environment 2.1 GA and replaced a number of parts that had failed prematurely. Once upgraded, the system was turned over to the ORNL to prepare their test harness for Jaguar PF acceptance testing.

Test Harness Development

The ORNL National Center for Computing Science (NCCS) team ported and enhanced their test harness that was used to control the execution of numerous scientific applications and their associated test data for the JAT. The test harness was devised to automate the creation of system workloads that encompassed all aspects of a traditional site workload, including representative applications.

The test harness repeatedly builds the applications, submits executables to batch, collects application results, and verifies the test data. The harness uses Python scripts that interact with the Subversion source repository to perform application builds. It then interacts with the batch scheduler (Moab/Torque) to submit application test jobs. The harness provides robust logging, restart capabilities, and permits real-time review of results via a web interface.

Though much of the test harness development had been done on the Jaguar XT4 system, final integration and testing was done on Jaguar PF. This final integration went smoothly and there were very few problems in scaling the test harness and its applications to the full size of the system. After the harness work was completed, the functionality, performance, and stability tests were begun, starting the formal phase of the acceptance.

Functionality Test

The purpose of the functionality test (JAT-FT) was to ensure basic operation of the system and consisted of a 12-hour run. A number of basic system operations were performed from system reboots to Lustre restarts.

Performance Test

The purpose of the performance test (JAT-PT) was to ensure the performance and scalability of a selection of Department of Energy Office of Science (DOE-SC) applications and I/O subsystem. A full range of problem sizes were run for each application code and each was determined to meet the performance criteria. Additionally, the I/O tests were run to validate the scalability and performance of the Jaguar PF I/O configuration and Lustre file system.

The primary focus of the performance test was to demonstrate that several key DOE applications could both scale and perform well on Jaguar PF. The set of applications chosen were determined to be a fair cross section of the expected Jaguar PF workload. Each of these applications was run over a range of problem sizes.

In order to pass the performance test, at least four of the applications were required to attain within 10% of the negotiated performance metrics and scale to at least 80% of the compute processors (sockets) in the system. All applications had to attain within 30% of their performance targets. A summary of all the applications run, their target performance, and final results are shown in Table 3.

One aspect of the performance test was validating the I/O sub-system performance. The most aggressive target was the 100 GB/sec aggregate I/O bandwidth. The IOR benchmark test was used to verify that the storage system could hit this target with the Lustre file system. This goal was achieved for both sequential and parallel reads and writes, as shown in Table 2.

Table 2 -Jaguar PF I/O Performance

Metric	Description	Goal	Actual
InfiniBand Performance	Send BW Test	1.25 GB/sec	1.54 GB/sec
Aggregate Bandwidth	Seq. Write Seq. Read	100 GB/sec	173 GB/sec 112 GB/sec
	Para. Write Para. Read	100 GB/sec	165 GB/sec 123 GB/sec
	Flash I/O	8.5 GB/sec	12.71GB/sec

Additionally, Lustre was tested to be sure that file system operations would scale linearly on the system. With a single file open time of 706 microseconds, Jaguar PF demonstrated 18,666 file opens in 12.04 seconds, achieving better than linear scaling.

Stability Test

The purpose of the stability test (JAT-ST) was to fill the machine with a large number of applications spanning the full range of problem sizes, with run times varying from a few minutes to several hours. Both the functionality and performance tests were run as entry criteria for the stability run. These tests served to root out problems before the long stability run.

The time spent in benchmarking for SC08 and in pre-acceptance work served a dual purpose. Not only did significant application tuning and benchmarking take place but, throughout the process, the system hardware matured as early part failures were replaced and the system gradually grew more stable. As a result, the final leg of the acceptance test, the Stability Test, completed December 23, 2008.

Table 3 - Jaguar PF Acceptance Results[3]

Application	Metric	Metric Value	Observed performance
LSMS-WL	Percentage of aggregate peak performance	70% of peak on at least 90% of compute nodes	77% of peak 17921 nodes (96%)
HPL	Percentage of aggregate peak performance	> 1PF	1.059 PF (#2 11/2008)
Pop	execution time in simulated model years per CPU day for a fixed size problem at various socket counts	>15 simulated model years per CPU day	16.0 simulated model years per CPU day using 18000 nodes
Aorsa	Percentage of aggregate peak performance for dense linear solver kernel	50% of peak on at least 85% of compute nodes.	61.7% of peak on 90.5% of compute nodes
Flash	Collective output bandwidth to a single (HDF) file using =>5000 Sockets/writers	Avg wall clock restart dump time (based on 5 instances) > 8.5GB/s	9.97 GB/s on 5000 nodes
Chimera	Weak scaling (numbers of zone evolved per wall clock second per processor)	Within 10% of being constant up to at least 90% of compute nodes. Minimum achievable is 300 Kzones/sec/PE	Within 10% of being constant on 91.6% of compute sockets and achieved 329.7 Kzones/sec/PE
DCA++	Time to solution on Gordon Bell Problem	Weak scaling on disorder configs to at least 90% of xt5 nodes with runtime <100min.	5901 sec (98.4 min) on 148352 cores (95% of system)
MADNESS	Scalable computation of the DFT energy	moldft - run (h2o)35 on 40K, 90K, 140K processors to completion and obtaining final converged energies that agree to at least 4.d.p.	3 moldft/35H2O runs completed, final converged energies agreed to within .00004.
NWChem	Time to solution on CCSD(T) restart problem	30K cores less than 1500 seconds	1041.5 sec on 30K cores
S3D	Weak scaling (wall clock time per time step per grid point)	190 us per grid point per time step.	178.6 us per grid point per time step on 144K cores.
GTC	Weak scaling (number of particles pushed per step per second)	[Time(2*N,2*Particles)/Time(N,Particles) < 1.15] where N is number of cores [N must be multiple of 64], where the 2N run has to push at least 20B particles*cycles/time.	[Time(2*N,2*Particles)/Time(N,Particles) = 1.02]. 27.3B particles were pushed using 51K and 102K cores
Global Arrays	Benchmark GA performance	put/get operations gives 80% of MPI-2 put/get operation bandwidth and a latency <= 25 usec.	put/get operations exceeded (2.33x) MPI-2 put/get operation bandwidth & get latency = 16.6 usec, put latency = 15.5 usec.

9. Petascale Software Scaling Summary

Cray demonstrated the ability of the XT software stack to scale from the initial install throughout the acceptance period. The completion of the Jaguar Acceptance Test confirmed that the operating system, programming environment, and system management software were well suited for PetaScale operation. The customer's acceptance of the system underscored this achievement.

Specifically, this demonstration proves that a tuned Cray Linux Environment (CLE) and Cray Programming Environment (CPE) can execute applications efficiently and reliably over 100K cores. The Cray System Management Workstation (SMW) also scaled administrative and management tasks from day one.

The swift progress made by reaching PetaFlop performance across several scientific applications is evidence that the Cray software environment simplifies

the task of porting, scaling, and tuning applications on a Petascale class system. Several applications reached record breaking performance and key DOE applications scaled well on Jaguar PF. Though most of the applications run on Jaguar PF had been developed earlier, Cray's tools demonstrated their value in both programmability and portability, as applications were quickly ported and tuned to Jaguar PF in hours and days, not weeks and months.

Options in the XT5 node configuration and CLE environment improved the reliability and robustness of the system in the presence of failures by reducing the impact of memory errors and number of system reboots. Consistency in the Cray Linux Environment and Programming Environment across product generations as well as scaling improvements in these software stacks were key to ensuring that these (and other) programs could be quickly ported and run on Jaguar.

Acknowledgments

The author would like to thank the combined Cray and ORNL team that not only contributed information for this paper, but more importantly made it possible to reach PetaFlop performance on a Cray XT Series system. Without the strong partnership of Cray and ORNL from the conception of the Jaguar PF system, through the machine installation on site, and to the execution of the JAT, the Jaguar PF system would not have been accepted. Both Cray and ORNL personnel put in significant effort above and beyond the call of duty in order to get acceptance by 2008 year's end. This was a monumental accomplishment and reflects Cray's dedication to its HPC customers such as ORNL and its ability to work intimately with them to reach common goals.

About the Author

Kevin Peterson has been with Cray for five years and is currently the Software Director for the Cray XT Series supercomputers. He joined Cray as the Software Technical Project Leader for the Cascade Program. Prior to Cray he was a software professional for over two decades at several other computer companies including Hewlett Packard, Compaq, Digital Equipment, and Texas Instruments. He can be reached at Cray, Inc. 1340 Mendota Heights Road, Mendota Heights, MN 55120 or by E-mail at kpeterso@cray.com.

References

1. *NLCF Acceptance Test Plans (50T, 100T, 250T, 1000T-CS) and (1000T-G)*
   DOE Leadership Computing Facility
   Center for Computational Sciences
   Computing and Computational Sciences Directorate
   December 10, 2008
2. *Jaguar & Kraken – The world's most powerful computing complex (PowerPoint Presentation)*
   Arthur S. (Buddy) Bland
   Leadership Computing Facility Project Director
   National Center for Computational Sciences
   November 20, 2008
3. *ORNL 1PF Acceptance Peer Review (PowerPoint Presentation)*
   ORNL Leadership Computing Facility
   Center for Computational Sciences
   December 29, 2008
4. *Acceptance Status (PowerPoint Presentation)*
   Ricky A. Kendall
   Scientific Computing
   National Center for Computational Sciences
   October 30, 2008
5. *SC08 Awards Website*
   http://sc08.supercomputing.org/html/AwardsPresented.html
   November 21, 2008
6. *Cray XT Manufacturing Plan*
   William Childs
   Cray Inc., Chippewa Falls, Wisconsin
   October 2008