Compute Node Linux:

Overview, Progress to Date & Roadmap

David Wallace
Cray Inc

ABSTRACT: : *This presentation will provide an overview of Compute Node Linux(CNL) for the CRAY XT machine series. Compute Node Linux is the next generation of compute node operating systems for the scalar line of Cray systems. The presentation will provide a vision of the long-term objectives for CNL.*

KEYWORDS: 'Cray XT3', 'Cray XT4', software, CNL, 'compute node OS', Catamount/Qk

1. Introduction

The Compute Node Linux (CNL) is the next generation of lightweight kernels for compute nodes on the CRAY XT3 and Cray XT4 computer systems. The CNL operating system provides a runtime environment based on the SUSE SLES distribution. Modifications are being made to the system software to match the legacy performance and scaling characteristics of Catamount/QK as well as address new marketing and customer requirements.

This paper will provide a brief overview of CNL objectives as well as the product requirements for a new compute node operating system. The majority of the paper will describe the current state of progress against these objects. The paper will conclude with a brief discussion of topics related to transitioning to CNL.

2. A New Compute Node OS: Objectives and Requirements

There are four major objectives of the Compute Node Linux project:

- The system must be stable and robust,
- The system must meet or exceed Catamount/Qk in terms of performance and scalability,
- The system should provide enhanced functionality to accommodate new applications and to expand the Cray XT3/XT4 system market presence,
- The system should support a concurrent Capability and Capacity computing environments on a single system.

The assumption is that Cray XT3/XT4 customers will be transitioning to CNL at a point in time when the requirements for maintaining a stable, production environment will be paramount. The expectation is that CNL will be equivalent to or better than Catamount/Qk in terms of stability, performance and scaling. Current and future requirements demand that CNL scale to 20,000 and

100,000 cores. CNL is also expected to support capacity application environments. CNL must support ISV and 3rd party applications. This means that CNL must support better application portability and increased OS functionality including standard Linux system calls, support for application networking, and an expanded set of programming models. And finally, CNL is expected to support a mixture of capability and capacity applications on a single system. A key objective is to provide support for flexible configuration of compute nodes.

The product requirements for the new compute node OS are derived from Catamount/Qk functionality and the need to maintain scalability to very large core counts. Below is a list of requirements based on Catamount:

- Scales to 20K compute sockets
- Application I/O equivalent to Catamount
- Launch applications as fast as Catamount
- Boot compute nodes almost as fast as Catamount
- Maintain a small memory footprint

CNL is comparable to Catamount/Qk with respect to the scaling and I/O requirements. Later in this paper, test results will be presented that show similar performance in terms of computation and I/O.

In order to expand the market for the Cray XT4, the following additional requirements must be met:

- Support for N-way cores
- Provide support for ISV and 3rd Party applications
- Provide support for multiple programming models including:
  - MPI
  - SHMEM
  - OpenMP
  - Global Arrays / ARMCI
  - PGAS language support (CoArray FORTRAN and UPC)

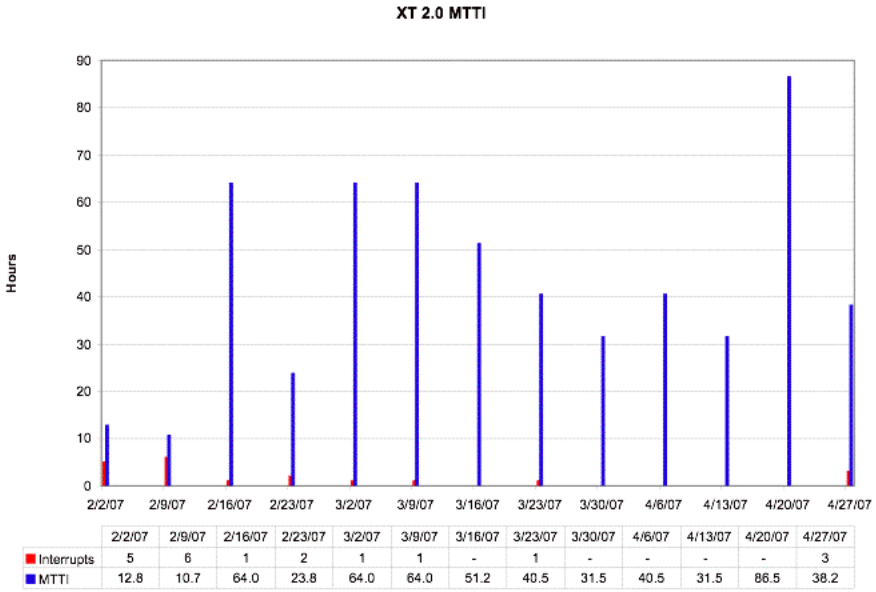
CNL will support all of the requirements listed above. Over time, Cray will enhance and improve support for ISV applications (further discussion on this topic can be found later in this paper).

3. Progress against Goals

3.1 CNL Stability

CNL is based on the SuSE SLES 10 Linux kernel (Linux 2.6.16). The OS kernel was updated to take advantage of enhancements to memory management and improved handling of out-of-memory conditions. In general, CNL has been fairly stable on the small internal systems used for development and shared use within Cray.

Most noticeable is the reduced frequency of system interrupts and resulting reboots (see Figure 1). This is largely due to the early work with the CNL prototype. The approach that was taken was to leverage as much common software from the existing UNICOS/lc base and augment it with code specifically required for CNL. This approach resulted in the re-use of the majority of the CRMS/HSS and Service node software, which is quite mature. CNL also benefited from early exposure at Oak Ridge National Laboratory (ORNL) on a large system. Testing exposed a number of scaling issues in the system administration tools and utilities. Resolution of these issues resulted in reliable booting of the system. System testing of CNL at ORNL was split between OS testing and performance testing. OS testing focused on functional regression, I/O, application and system stress testing. This work formed the basis of a stable environment to do performance and scaling work.



	2/2/07	2/9/07	2/16/07	2/23/07	3/2/07	3/9/07	3/16/07	3/23/07	3/30/07	4/6/07	4/13/07	4/20/07	4/27/07
Interrupts	5	6	1	2	1	1	-	1	-	-	-	-	3
MTTI	12.8	10.7	64.0	23.8	64.0	64.0	51.2	40.5	31.5	40.5	31.5	86.5	38.2

Figure 1 CNL stability

3.2 CNL Exposure

One of the keys to stabilizing a new OS is to provide broad exposure to end-users. CNL has be running on in-house system since June 2006. The total amount of time CNL runs on the in-house systems has been steadily increased. Early in 2007, Cray extended an offer to Army High Performance Computing Resource Center (AHPCRC/NCS) to run their applications on CNL and provide feedback.

The feedback from AHPCRC/NCS was generally favorable. The system ran without need of a system reboot over the two days Army ran on the system. Army ran a number of applications (WRF, CPMD, Adsorb, BenchC) and reported that system functionality was good and in general performance was close to Catamount.

3.3 Performance and Scaling

One of the CNL requirements is that it must meet or exceed Catamount/Qk in terms of performance and scalability. In order to measure this, a set of applications were selected to be used in determining how well CNL scales and performs in comparison to Catamount/Qk. The set of applications are listed below:

- CCSM: a fully-coupled, global climate model that provides state-of-the-art computer simulations of the Earth's past, present, and future climate states
- GTC: a 3-dimensional particle-in-cell code in toroidal geometry
- HPCC: the HPC Challenge benchmark
- LSMS: a first-principles computer model that simulates the interactions between electrons and atoms in magnetic materials.
- MILC: a set of codes used for doing simulations of four dimensional SU(3) lattice gauge theory
- NAMD: a parallel, object-oriented molecular dynamics code designed for high-performance simulation of large biomolecular systems
- Paratec: a materials science total energy planewave pseudopotential Fortran 90 code
- POP: an ocean circulation model
- VH1:

The initial goal is to demonstrate the performance of these applications on CNL within 10% of Catamount/Qk as measured by elapsed time. Scalability would be measured by running these same applications at 1K, 5K and 10K cores and comparing the performance at each core count (note that some of these applications do not scale to 10K cores). POP was chosen because of its sensitivity to OS jitter, which was considered to be a significant problem that CNL would have to overcome.

Figure 2 shows the status of performance and scaling of each of these applications with respect to Catamount/Qk. 'Green' indicates that the application is within 10% of the performance of Catamount/Qk, 'Red' indicates that the application is outside of the target performance range. Figure 2 represents a summary of the performance and scaling for these applications.

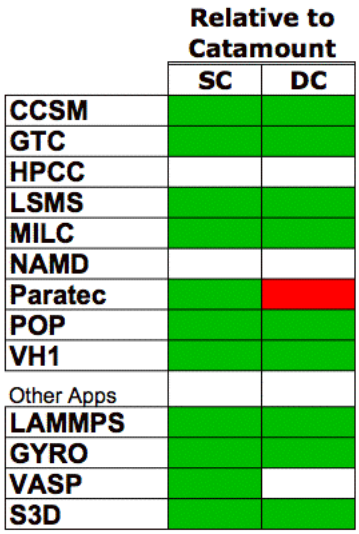


Figure 2 CNL performance and scaling comparison

Also represented are several other applications that are commonly run on the Cray XT4. Blank boxes (those without a color) are cases where results have not yet been obtained. In general, most of these codes are within a range of 1-5% of Catamount/Qk performance. A few codes show improved performance over Catamount/Qk.

Figure 3 highlights the performance of POP over a period of time. As previously mentioned, the performance of POP is sensitive to OS jitter. Detailed investigations into the possible sources of OS jitter in CNL were conducted and experiments run to reduce OS jitter. While this work showed some small improvement, the biggest effect on CNL performance was gained by changes and enhancements to Portals. Initial tests of POP showed a performance degradation of as much as 200% on dual core at 8000 cores and almost 100% on single core at 1000 cores. After incorporating the Portals changes, POP performance is now within 5% of Catamount/Qk performance for core counts between 1000 and 8000. In a few cases, POP under CNL was measured to be faster than Catamount/Qk. Further enhancements are expected to further improve the performance of POP.

# Cores	Routine	April 17/April22 (- Slower, + Faster) SC	DC	13-Apr SC	DC	7-Jan SC	DC
1000	Step/Total	-1%	2%	-15%	-9%	-23%	-46%
1000	Baroclinic	4%	-1%	2%	-2%	-2%	-15%
1000	Barotropic	-7%	6%	-40%	-22%	-40%	-95%
2000	Step/Total		0%	-6%	-5%		
2000	Baroclinic		-3%	0%	-1%		
2000	Barotropic		2%	-26%	-14%		
4000	Step/Total	3%					
4000	Baroclinic	1%					
4000	Barotropic	4%					
8000	Step/Total		-5%				-150%
8000	Baroclinic		-4%				-22%
8000	Barotropic		-5%				-179%

Figure 3 Performance results for POP

Other applications have shown a benefit from running on CNL. Table 1 shows the results of the Himeno-BMT benchmark (a 3D CFD Poisson kernel) that ran 1.26 times faster under CNL.

Table 1 Comparison of Himeno-BMT performance

Compute Node OS	CPU Sped	# of Cores	Performance
Catamount/QK	2.6 GHz	256	232 MFLOPS
CNL	2.6 GHz	256	292 MFLOPs

A limited set of results has been obtained for applications using OpenMP on CNL. Figure 4 shows the results of CPMD using OpenMP.

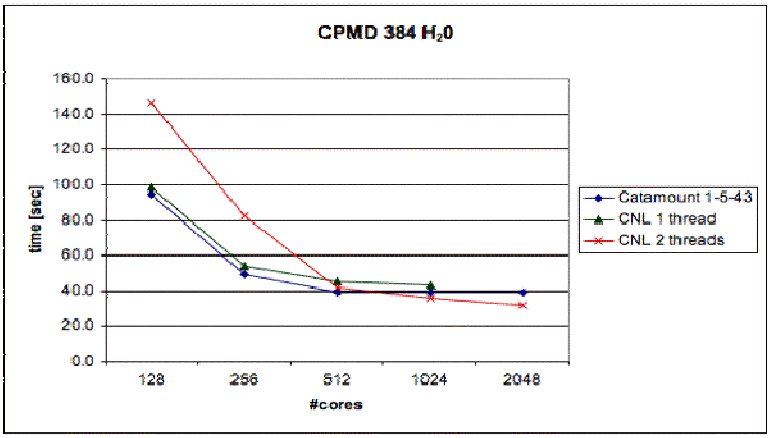


Figure 4 Sample application using OpenMP on CNL

A summary of the results is follows:

- Catamount is always faster than CNL when using one thread only. Performance is inside the 10% the OS group has as a target (except for the 512 core run).
- Using 2 OpenMP threads is slower for smaller number of cores, but as expected (or hoped) it scales better than Catamount/Qk and CNL using 1 thread beginning with 1024 cores. In the latter case, OpenMp is the fastest version.
- When comparing the 2048 core Catamount and CNL-2threads numbers, most of the time gained is from the communication needed for the FFTs (AlltoAll).
- It appears that the CPU part is slower on the CNL side compared to Catamount (2048 cores),
- For comparison, all runs were done using the "default" MPI settings, i.e. no MPI environment settings were used.
- CNL runs were not done on a dedicated system; the Catamount runs were run on a dedicated system.
- All runs were done in 'packed' mode using both cores (-VN for catamount, -N2 (default) for CNL)

3.4 I/O Performance

The goal for CNL is to match the performance of Catamount/Qk for I/O to a Lustre file system. Tests using applications as well as I/O benchmarks have shown that CNL meets or exceeds Catamount/Qk in terms of I/O performance. A number of applications that do large amounts of file I/O have demonstrated higher performance under CNL. Figure 5 shows CNL I/O performance using IOR running on eight client nodes to a single file. The figure shows read, write and metadata rates.

Test	Size	Operation	QK	CNL	Measure	Diff %
IOR	65536	Write	48.734	678.953	MB/sec	1293%
IOR	65536	Read	123.220	569.569	MB/sec	362%
IOR	1048576	Write	524.558	688.652	MB/sec	31%
IOR	1048576	Read	404.950	517.401	MB/sec	28%
IOR	4194304	Write	725.626	693.292	MB/sec	-4%
IOR	4194304	Read	603.823	537.106	MB/sec	-11%
mdtest	1	Dir_create	2402.801	3484.987	ops/sec	45%
mdtest	1	Dir_stat	4854.606	4887.840	ops/sec	1%
mdtest	1	Dir_rm	2000.467	3161.859	ops/sec	58%
mdtest	1	File_create	1514.006	2716.896	ops/sec	79%
mdtest	1	File_stat	4686.043	4740.014	ops/sec	1%
mdtest	1	File_rm	1658.783	2685.556	ops/sec	62%

Figure 5 IOR results (stripe: 1MB, count: 4)

3.5 CNL Functionality

CNL is meeting the product requirement expectations with respect to increased functionality. A summary of these requirements is listed below:

- Support for N-way cores
- Provide support for ISV and 3rd Party applications
- Provide support for multiple programming models including:
  - MPI
  - SHMEM
  - OpenMP
  - Global Arrays / ARMCI
  - PGAS language support (CoArray FORTRAN and UPC)

CNL is currently running on Cray XT3 and single and dual core XT4 systems. Progress towards running on quad core Opteron processors is proceeding according to plans. Support for Cray XT4 systems with quad core processors is expected to be introduced before the end of 2007. The CNL programming environment currently supports all of the compilers, libraries and tools that are available in UNICOS/lc. Cray is working with 3rd parties vendors to provide support for Global Arrays and UPC under CNL.

The primary emphasis at Cray is to be the leader in HPC capability computing. A secondary requirement is to be able to support a broad range of ISV and 3rd Party vendor applications. Over time, Cray will enhance its ability to support arbitrary ISV and 3rd Party applications.

4.0 Converting to CNL

CNL is intended be the compute node operating system for all Cray Scalar systems beginning with the Cray XT4 Quad Core system. CNL will be supported on the Cray XT3 and the Cray XT4 (single, dual and quad

core) systems. Customers will have the option of upgrading to CNL beginning with UNICOS/lc version 2.0.

Customers who are planning to convert to CNL are encouraged to plan ahead! Utilizing a small test system (if available) is an excellent way to gain exposure to CNL and it provides a platform independent of the main production system. This platform can be used to port system administration tools and utilities, convert or port customisations for site accounting and the Batch subsystem. UNICOS/lc 2.0 has features that allow for creating and maintaining multiple versions of the system software as well as utilities to facilitate switching between different versions of the installed software.

Getting users exposed to CNL prior to switching the production system over to CNL requires additional planning. Using dedicated time is a familiar means for testing new software and getting friendly users on the new system. Another strategy is to employ the System Partitioning feature introduced in UNICOS/lc version 1.5. System partitioning allows the administrator to create logically separate systems that can be operated concurrently, each running a different version of the operating system. There are strict requirements for using system partitioning which includes separate SIO nodes (I/O and login Service nodes) and separate file systems for each partition. System partitioning is not a particularly flexible migration aid, but it does insure complete separation of the production and test systems while allowing concurrent operation.

For customers with a test system, the first opportunity to install and begin testing CNL will be the UNICOS/lc 2.0 Limited Availability release targeted for 2Q07. Customers with large systems or production requirements should wait for the UNCIOS/lc General Availability release slated for 4Q07.