PERCU Results in a A Reawakened Relationship for NERSC and Cray

William T.C. Kramer[1]
National Energy Research Scientific Computing (NERSC)
Facility and Lawrence Berkeley National Laboratory
kramer@nersc.gov

ABSTRACT: *NERSC is DOE's Flagship High Performance Computing Facility supporting 1,000s of users and 100s of computationally challenging projects. This paper provides a brief overall of NERSC usage and then discusses the processes NERSC used to evaluate and acquire NERCS-5 our latest computational system – the world's largest Cray XT-4. NERSC uses the PERCU methodology (Performance, Effectiveness, Reliability, Consistency and Usability) to assess systems – not just before purchase but throughout their life. Combined with the Sustained System Performance (SSP) and Effective System Performance (ESP) metrics, NERSC is able to assure its client community and stakeholders that is will provide highly productive and cost effective systems. This paper also discussed the configurations and planned evolution of the NERSC-5 systems.*

KEYWORDS: XT-4, Sustained System Performance, System Evaluation, HPC, PERCU, SSP, ESP

Introduction

NERSC has served the DOE Computational Research community since its inception in 1974 as one of the very first "Supercomputing Centers". In 2005, NERSC set the stage for the next five years of its evolution through a series of important planning activities that determine how NERSC will continue to be a premier HPC center for open science. Through the well-established Greenbook process, our active user community provided their input to the planning process. NERSC management then developed a new five-year plan for 2006 to 2010[1], which was then thoroughly reviewed in a programmatic review by DOE. This plan includes the acquisition of the major new computational systems, NERSC-5 and later NERSC-6 and NERSC-7. The reviewers fully endorsed our plans, stating in part: "NERSC is a strong, productive, and responsive science-driven center that possesses the potential to significantly and positively impact scientific progress.… NERSC is extremely well run with a lean and knowledgeable staff."

Indeed, NERSC facilitates highly efficient use of the computational resources. The combination of excellent, leading edge systems, highly effective user assistance, efficient system management and open, protected access has enables NERSC's scientific users to be highly productive. This is evidenced by the fact that each year on their allocation renewal form, PIs indicate how many refereed publications their project had in the previous 12 months.

Year of request renewal	Number of refereed publications
2007	1,437
2006	1,448

---

[1] This work was supported by the Director, Office of Science, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract number DE-AC03-76SF00098.
.

2005	1,270

Table 1 shows the number of peer-reviewed publications that were based on NERSC resources over the past 3 years.

NERSC added significant capacity in 2005 by introducing two new clusters, named “Jacquard” and “Bassi.”, approximately doubling it computational capability. NERSC is currently in the process of increasing the total sustained computational capability for the Facility with the addition of NERSC-5. NERSC-5 is named “Franklin”, and is the largest Cray XT-4 delivered to date. All these systems were selected in competitive procurements. This increase in capacity was highly welcomed by the NERSC community.

NERSC and Cray – a Long and Productive Relationship

NERSC has a long history of using early delivery Cray systems to meeting the demands of the DOE computational community. Cray systems at NERSC include:

- 1974 - NERSC began with a CDC 6600
- 1975 – Used LBNL CDC 7600
- 1978 – Cray 1 (SN 6)
    - CTSS first used - NERSC joins CUG
- 1981 – Second Cray 1
- 1984 – Cray XMP
- 1985 – First Cray-2 (SN 1)
    - Demonstrated UNICOS
- 1990 – Only 8 processor Cray-2
- 1992 – 8 processor XMP
- 1993 – 16 processor C-90 (SN 4005)
- 1994 – Installed early T3D
- 1996 – NERSC moves to LBNL
- 1996 – 128 processor T3E-600 (SN 6306) and J-90 (SN 8192)
- 1997 – Added 512 processor T3E-900 (SN 6711)
    - Unicos/mk - First C/R on an MPP
- 1998 – Increase T3E-900 to 696 processors
- 1998 - Installed first SV1s (SNs 9601, 02, 05)
- 2007 – Installed largest XT4 (SN 4501) – 19,584 processors

Note the NERSC has done similar systems from other vendors such as IBM, Linux Networx and SGI following similar processes.

NERSC Users

Meeting the computational science needs of the DOE Office of Science encompasses a broad range of research projects in terms of scientific disciplines, geographic location or home institution. Here are some statistics on the NERSC user community.

NERSC served 2,677 scientists throughout the United States in 2005 and 2,978 scientists in 2006. These researchers work in DOE laboratories, universities, industry, and other Federal agencies. Figure 1 shows the proportion of NERSC usage by type of institution and 2006. Figures 2 and 3, show laboratory, university, and other organizations that used large allocations of computer time in 2006. Computational science conducted at NERSC covers the entire range of scientific disciplines, but is focused on research that supports the DOE’s mission and scientific goals, as shown in Figures 4.

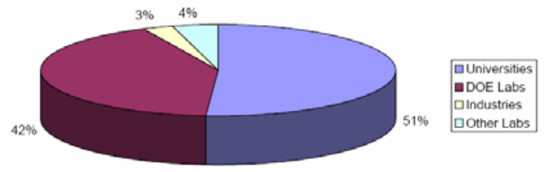

Figure 1. NERSC MPP usage by institution type, 2006 (Percent).

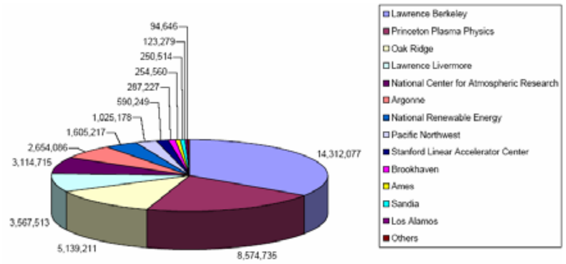

Figure 2. DOE and other Federal laboratory usage at NERSC, 2006 (MPP hours).

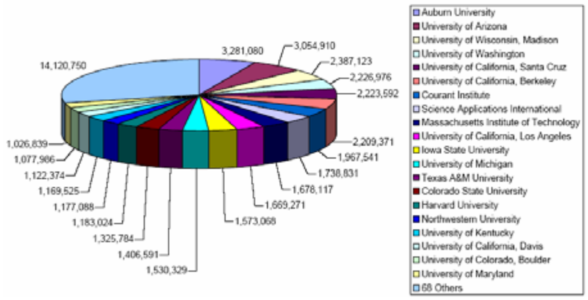

Figure 3. Academic and private laboratory usage at NERSC, 2006 (MPP hours).

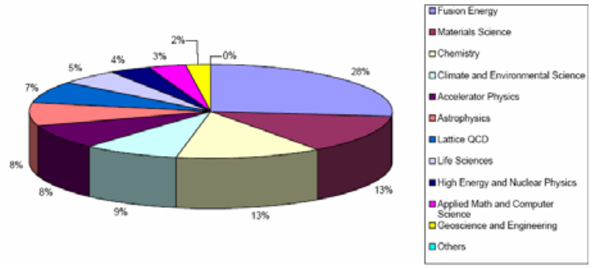

Figure 4. NERSC usage by scientific discipline, 2006.

The NERSC Workload

Based on the diversity of Science, the NERSC workload is also complex and diverse[2]. Table 2 shows the number of projects NERSC has supported over time. The table shows there is a continued increase in projects, in large part due to the unmet demand for computation and storage resources. The Production projects are selected to support DOE programmatic needs, while INCITE projects are awarded as a result of an international competition. At DOE's request, NERSC began the INCITE program in 2003 (then called "Big Splash") with the goal of providing exceptional amounts of computational resources and services to high payoff projects that can demonstrate marked progress within a year. Since then NERSC has provided up to 15% of all its resources to s very small number of projects. This program was expanded to ORNL and ANL's computational facilities in 2006. SciDAC projects are those in support of DOE's SciDAC-I and SciDAC-II programs which stretch application areas that need to achieve Petascale computing. Start-ups are small allocations to enable new science areas and research teams to demonstration their expertise and readiness for large scale computing.

Allocation Year	Production	INCITE & Big Splash	SciDAC	Startup
2007 (as of May)	291	7	45	44
2006	286	3	36	60
2005	277	3	31	70
2004	257	3	29	83
2003	235	3	21	76

Table 2: The number of scientific projects NERSC supports over time.

Table 3: State-of-the-art computational science requires increasingly diverse and complex algorithms.

The NERSC scientific applications require well balance computational systems[3]. Table 3 shows the different algorithmic methods[4] that science disciplines use. High performance computing requires architectures capable of achieving high performance across the spectrum of applications. It is very challenging, but possible, for architectures to address all these algorithmic areas in the same system. On the other hand, scientists become less productive if they have to use many different, specialized systems since they spend significant time managing data, codes and jobs. For HPC computing, balanced system architectures are extremely important.

Science areas	Multi-physics, Multi-scale	Dense linear algebra	Sparse linear algebra	Spectral Methods (FFT)s	N-Body Methods	Structured Grids	Unstructured Grids	Data Intensive
Nanoscience	x	x	x	x	x	x		
Climate	x			x		x	x	x
Chemistry	x	x	x	x	x			
Fusion	x	x	x			x	x	x
Combustion	x		x			x	x	x
Astrophysics	x	x	x	x	x	x	x	x
Biology	x	x						x
Nuclear		x	x		x			x

Combining Table 3 with Figure 4 dictates only a balanced system function well in NERSC environments.

Large Scale Jobs on Efficient Systems

At the same time, the demand for computation has far exceeded what can be provisioned, so efficient operation of the NERSC systems is essential. NERSC has been at the forefront of assisting scientists to increase the scale of their computational applications[5]. Indeed, NERSC has been given the goal of at least 50% of its computational time going to jobs 1/8th or more of the overall system. Over the past 4 years since DOE set this goal, NERSC has exceeded it, as shown in Figure 5. Furthermore, NERSC is able, through focus on system resource management, to provide high amounts of time to science projects – thereby keeping the system busy. This is shown in Table 4.

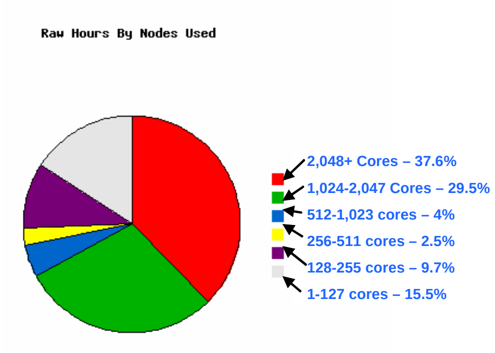

Figure 5: Percent of time used by different job sizes on NERSC's largest system – Seaborg – with 6080 computational processors for the period January 9, 2007 to May 4, 2007. Note more then 67% of the time is used by jobs more the 1,024 processors.

NERSC Allocation Year	Percent of Overall Time used by science codes on Seaborg
AY 2004	90.0%
AY 2005	93.5%
AY 2006	87.5%
AY 2007 to date	88.5%

Table 4: Amount of time used on the largest NERSC system.

Detailed examples of the science applications and results are found in numerous papers[6], and in the NERSC Annual reports[7]. The end result is systems selected for operation at NERSC have to have high performance, good balance, and robustness. The next sections explain how NERSC evaluates and selects such systems, and how that selection process for NERSC-5 resulted in re-energizing the NERSC-Cray relationship.

Best Value and the PERCU Method of System Evaluation

NERSC is required to performance full and open competitions for all its major computational hardware. To accomplish this in the most efficient and flexible manner, NERSC uses the Best Value Source Selection Process. BVSS was originally implemented at Lawrence Livermore National Laboratory and then used and refined at LBNL on NERSC 3, NERSC 4, NCS, NCS-b and NERSC 5. The BVSS process has been adopted by other labs as well. BVSS intends to reduce procurement time, reduce costs for technical evaluations, and provide efficient and cost effective ways to conduct complex procurements. It is used in competitive, negotiated contracting to select most advantageous offer. BVSS benefits include flexibility, allowing offerors to propose (and sites to consider) different solutions from what may have been envisioned at the outset and allows sites evaluate and compare features in addition to price. It is un-weighted and un-scored and focuses on strengths and weaknesses of proposals. Despite the flexibility, the end result at NERSC is usually a firm, fixed price contract with 100's of criteria that is completely agreed upon by NERSC and the vendor.

Within the BVSS framework, NERSC translates scientific requirements into about 50 total system requirements– all at high level. These reflect the attributes computational scientists want in a large system, which are

- **P**erformance - How fast will a system process their work if everything is perfect
- **E**ffectiveness - What is the likelihood they can get the system to do their work

- **R**eliability - The system is available to do work and operates correctly all the time
- **C**onsistency/Variability - How often will the system process their work as fast as it can
- **U**sability - How easy is it for them to get the system to go as fast as possible

This categorization of needs is called the *PERCU* method which we will discuss in a little detail now.

Benchmarks Have Four Purposes

Most reports in the literature discuss only one of four distinct and important purposes benchmarks play in the life of a system. Many tests do well on one or possibly two of the goals for benchmarks, but few are effective in all. This section will briefly explain how the Sustained System Performance Test (SSP)[8] methodology can be used for all purposes. The four purposes of benchmarks can be organized as the following:

1. Selection of a system from among its competitors. Typically applications and kernels are used for this.
2. Validating the selected system works the way expected once a system is built and/or arrives at a site. This purpose may be more important than the first reason and is particularly key when systems are specified and selected based on performance projections rather than actual runs on the actual hardware. This is also where the use of only simple tests and kernels fails for the first time since they can not capture the complex interplay that exists in large scale systems.
3. Assuring the system performs as expected throughout its lifetime (e.g. after upgrades, changes, and regular use.) Again, due to complex interactions, applications are typically used.
4. Finally, benchmarks can be used to help guide future system designs. This is where kernel benchmarks are most often successful. This area is where kernels and simple, focused tests may be used in lieu of full applications.

HPC procurements require sophisticated methods to gauge the potency and value of the system. Simple "speeds and feeds" that are typically supplied by hardware vendors or from simple, one dimensional tests are not nearly sufficient for the complexity of today's and tomorrow's systems.

Motivations for SSP

Whenever one buys technology that is driven by Moore's Law, be it consumer electronics, personal computers or supercomputers, there is a fundamental issue that can be expressed as "If I wait a little longer, I can get a better system for the same cost. Should I wait?" This of course becomes more critical when selecting HPC systems due to cost and long lead times. When looking at a single system from a single vendor this may be a simpler question because one has to assess how long the wait is and how much better the later system would be. However, even when just going to the local computer store it is clear the simple case never exists because different systems are available at different times. How does one decide?

The primary motivation of the SSP is to help address the "when to buy" as well as "what to buy" questions by providing a quantitative assessment of sustained computer performance over time and to address how to represent a complex workload with a metric that is meaningful and balanced (i.e. not dominated by one architectural attribute). The SSP metric has value in addressing the first three of goals for benchmarks and has potential to address the fourth goal. The steps in creating an SSP metric start like any other benchmark activity[9].

1. Select potential representative codes from the target workload
2. Refine the selection of the codes to give the best coverage of discipline areas, algorithmic methods, use of resources (CPU, memory, I/O), portability and run time.
3. Select problem sets and set concurrency of the codes to balance the run time on existing systems and the run time on the target systems. HPC systems are selected well in advance of general availability so often vendors will have to run the benchmarks on current systems and project performance to systems to be delivered in the future.
4. Define a way to make a composite metric out of individual benchmark runs.

Defining the SSP

Rephrased, the purpose of the Sustained System Performance (SSP) metric is to provide an objective basis for evaluating and tracking the performance of installed systems for their lifetime. The high level description of the SSP is to use a suite of tests to establish a measure of per processor operation rate. For the purposes of this paper, floating point operations per second is the operation rate of interest but, integer operations, transactions or other work units can be used as well. The operation count for each test code can be determined by the processors hardware performance counters on existing systems. Similar to using different operations, the SSP tests can use any type of programming models and communication methods. In fact, the methodology works as well for serial codes as it does for parallel codes.

To obtain the SSP metric[†], an operation count is obtained from microprocessor hardware counters for the applications and by the concurrency of the application. A computational rate per processor is calculated by dividing the floating-point operation count by the sum of observed elapsed times of each application. The per processor SSP rate is an aggregate measure for all the component applications, which, in the simplest form, can be a basic mean of the component per processor rates. Once a per processor SSP rate is established the system-wide SSP value is determined by multiplying the per processor SSP rate by the number of computational processors in the proposed/delivered system.

Different vendors introduce technology at different times, and it may be to organization's advantage to have current technology installed and then have a predetermined upgrade to new technology that has higher performance. That is, having phased improvements of the system to have the best value. Alternatively, it may be better for an organization to wait for delivery for some time in order to install later generations of Hardware. In order to account for different delivery dates and phase scales, the calculation for the area under the curve uses a common start and end date for all bid systems. This normalizes for systems that are delivered "late" and also takes into account the staged delivery of systems to the site. A vendor can make up for a later delivery of a system by increasing the total size of the delivered system and/or providing faster technology. Either will compensate for the loss in area under the SSP curve caused by the later delivery. Because of Moore's Law, this may be an advantage to both provider and purchaser.

The capability or *Potency* of a system is then represented by the potency of a system integrated over a given time period. The SSP value (currently in Mega/Giga or Tera flops/s-months) indicates the effective average performance of the system on an organization's scientific workload at any given point in time. In order to enable a comparison between systems, the potency of the system is the total area under the SSP curve over a given time period (NERSC uses 3 years, but any time period is possible). The potency of the system for the entire workload at any point in time, and indeed, throughout any period, can be quantified to assess the price performance, or "value" of the system by dividing the potency by the cost of the system – basically Tflops/s-years per $. This gives an important and straight forward way to determine the system with the best value out of all systems.

SSP is one aspect of a PERCU system evaluation

The PERCU approach to system evaluation stands for Performance, Effectiveness, Reliability, Consistency and Usability. All five factors need to be taken into account and measured before a conclusion can be drawn as to how well a computer system can support

---

[†] Note a complete mathematical definition of SSP will be available in UC Berkeley dissertation nearing completion. Please contact the author to receive notice of release.

scientific work. The SSP metric, while very useful, is not sufficient to evaluate systems. Indeed, NERSC always combines SSP with a suite of complementary measurements covering all areas of the PERCU approach to system evaluation.

In addition to SSP, NERSC selects real applications that come from the client community codes and represent the future workload that is strategic to the DOE science community. Each application test assesses at least one, and often multiple attributes of the systems, programming methods and algorithms. Additional applications may be selected from projects in other discipline areas, although practical considerations limit the number of full applications feasible to between five and eight. Added to the applications are specific functionality tests, reliability tests, and benchmarks for storage and network I/O. NERSC uses specialized (kernel) tests that measure internal communication bandwidth in the memory subsystems of the processors, or across the communication fabric.

Effective System Performance

Explicit metrics including the Effective System Performance Test[10] (**ESP**)[11], throughput tests, consistency runtime measures, functionality and reliability metrics all are used in assessing and validating performance.

ESP measures system utilization and effectiveness with the primary motivation in developing this to aid the evaluation of high performance systems. ESP is used to monitor the impact of configuration changes and software upgrades in existing systems.

The ESP test extends the idea of a throughput benchmark with additional features that mimic day to day supercomputer center operation. It yields an efficiency measurement based on the ratio of the actual elapsed time relative to the theoretical minimum time assuming perfect efficiency. This ratio is independent of the computational rate and is also relatively independent of the number of processors used, thus permitting comparisons between platforms.

Reliability

To date, almost all reliability metrics and/or requirements for HPC systems have been reactive and only visible after the evaluation decision. Yet there is a great need, as systems get exponentially more complex, to have a more common, semi-proactive test to evaluate and project eventual reliability of the overall system. Examples attempts to do this including requirements to “run specific application code without interruption for N hours” or having the “system run a given workload for X days with Y% availability."

It is more likely to be able to estimate discrete hardware MTBF and MTTR and use that to decide hardware configurations. The hardware only measure is of limited value. Most major system wide failures, at least at NERSC, are software based. Yet, little data on HPC software reliability or performance is tracked by vendors these days.

In system evaluation and selection, there should be as precise and complete understanding of software as there is for hardware, but he question is how to assess reliability proactively. This is even more important with the new world of horizontal integration, many reliability issues stem from component interaction and are not visible to any individual component provider.

One modest attempt to assess this area for NERSC-5 was to see how well providers understand the reliability of their components and then of the integration of the components. There has been some work in reliability assessment for systems that have not been used for HPC. Such work includes injecting failure modes and assessing corrective reaction of systems, probing for weak areas and applying statistical learning theory/control theory to observe and then improve response. Unfortunately for this area, most research is in discrete systems or Web oriented farms

Consistency of performance

Performance variation is caused by many factors and can be detected by SSP at initial delivery as well as though the life of the system. On HPC distributed memory systems, it is rare the compute-intensive parallel applications share SMP nodes.   So, many factors contributing to variation are not present on these systems. HPC application run times can still vary widely[12]. The SSP is useful in detecting anomalies and situations that contribute to unexpected variation.

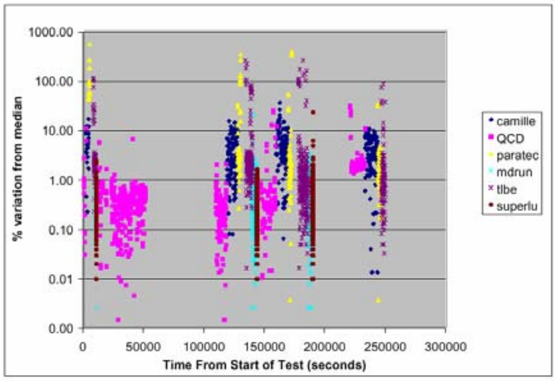

Figure 5: shows performance variation of SSP-2 applications on NERSC-3. The codes ran over a three day period with very little else on the system. The run time variation shows that large-scale parallel systems exhibit significant variation unless carefully designed and configured.

There are many examples of variability[13]. At NERSC, we have seen 10-20% more work coming from systems after consistency issues are address. This results in loss of computational resources that can be avoided.

Explicit variability metrics can make a significant difference. Such measures include using the Coefficient of Variation on multiple benchmark runs, throughput tests, the actual workload etc. This needs large amounts of information to prove cause of degradation. For example, one investigation took 9 months to determine the cause of a 10% performance difference between ½ the nodes in our system. Solving it immediately generated the equivalent of a ½ TFlops/s more computing for NERSC users!

Another example at NERSC, the NERSC-3 Phase 1 system that had a persistent degradation of performance, measured both by SSP-1 and user applications. The system consistently slowed down by 5% every month until it was totally rebooted. A reboot would return the system to the expected performance level. This was only detected because of proactively running the SSP-1 benchmarks on a regular basis. Since the system was in place less than 18 months before replacement with Phase 2 and it took time to identify the reduction, it was not possible to definitively determine the cause of the slowdown. However, recognizing the degradation meant a work around of rebooting the system every month was worthwhile.

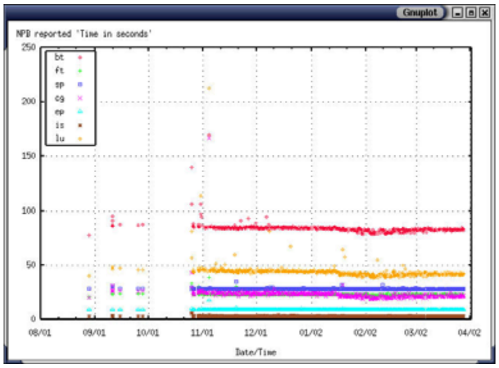

Figure 6 shows seven months of run times for six SSP-1 codes on the NERSC-3 system. The earlier runs show variation in performance before and after adjustments made to the MP_RETRANSMIT_INTERVAL interval. The interval controls how long an application waits before retransmitting messages.

Figure 5 shows the variation present on the NERSC IBM SP system when it was first installed. Previous experience had shown a number of software factors could cause variation, including different system software on nodes, system management event timing and application performance tuning. These issues were all mitigated on this system before the period being discussed. However, configuration problems, bugs, and architectural issues remain that cause variance.

Despite the challenges, some of which are outlined about, it is possible to make such a large system operate in a consistent manner. Figure 7 shows results of running the SSP-1 codes on the same system 7 months after the period show in Figures 5 during production. It shows that in a heavily used (85-95% utilization) system, the benchmarks have consistently low performance variation over multiple runs.

Usability

What scientists really want to know is how much harder it to use this system than their standard platform/tools. In the past this "standard platform" was most likely a Sun workstation, but today most scientists are using Linux desktops. So, for HPC, we could conceive of a relative measure rather than an absolute measure of usability. One example is to assess, relative to a scientist desktop, how much more effort is required to get X amount more work done on HPC systems than on their desktop. Alternatively – is it worth learning how to use a much more sophisticated and efficient tool.

Currently this area is handled either by checklists of functional features or qualitative assessments (note some qualitative assessments may use numerical values). More should be done in this area, but due to the lack of space it will have to be discussed in another forum.

PERCU Overall

PERCU holistically assesses systems. It takes into account many of the key aspects the computational science community's needs in HPC systems. Note PERCU does have not specify how a system is acquired. While NERSC uses the PERCU method with its Best Value procurement process, PERCU simply points out what a system should do for it to be effective for users. It can be used in any assessment of technology and any purchase method.

PERCU is a good way to address risk, particularly if there is a commitment to certain levels of performance by a provider. PERCU also is relevant and explainable to the science community, and traceable to their requirements

The Cray XT-4 – The result of PERCU

NERSC used BVSS and PERCU in its evaluation and assessment for NERSC-5. The original goals of NERSC 5 set in 2005, derived from the NERSC User requirements and budget, were:

- Sustained System Performance over 3 years - 7.5 to 10 Sustained Teraflop/s averaged over 3 years
- System Balance
    - Large Aggregate memory
    - Ability to use at least 80% of the available memory for user code and data.
    - Global usable disk storage
    - At least 300 TB
    - Ability to integrate with the NERSC Global File system (NGF)
- Expected to significantly increase computational time for NERSC users in the 2007 Allocation Year (AY)
    - January 9, 2007 – January 8, 2008
    - Have full impact for AY 2008

It is not possible to go into detail about the RFP assessments, but the main point is the Cray XT-4 provided the best overall performance and the best cost – yielding the best value – of all the proposed systems.

Franklin Characteristics

The Cray Xt-4 was names "Franklin" after America's First true Scientist – Benjamin Franklin, on the 300th anniversary of his birth. It is appropriate since Benjamin Franklin, performed ground breaking work in energy efficiency, electricity, materials, climate, ocean currents, transportation, health, medicine, acoustics and heat transfer. These are all areas the DOE community is currently engaged

Franklin is the largest XT-4 yet delivered. The major system characteristics are that it is made up of 102 cabinets with 9,740 dual core nodes (19,480 CPUs). Each node has a dual core AMD Opeteron processor running at 2.6 GHz and has 2 GB of memory running at 667 MHz. The system has 39.5 TBs of total, aggregate memory. It is using the Cray SeaStar 2.1 3D Torus Interconnect in a 17x24x24 configuration. This interconnect has a 6.3 TB/s Bi-Section bandwidth with each link providing 7.6 GB/s peak bi-directional bandwidth. There is expected to have 345 TBs of usable shared disk for permanent and scratch storage.

The system has significant external transfer capabilities, including sixty 4 Gbps Fibre Channel Data connections, four 10 Gbps Ethernet network connections, sixteen 1 Gbps Ethernet network connections

Franklin Performance and Status

At the date of this report, the XT-4 is undergoing testing at NERSC, so details are not yet available. It is clear the system will provide at least 16.1+ Tflops/s Sustained System Performance (SSP) over the course of the first 36 months of service at NERSC. In contrast, using the same metric, the IBM Power-3 Seaborg has a rating of .9 TFlops/s. The IBM Power 5, Bassi has a rating of .8 TFlops/s.

It is expected that Franklin will be in full service by the end of the summer 2007. At that time it will be running the Catamount OS on the compute nodes. It is expected that Franklin will migrate to Cray's Compute Node Linux in early to mid 2008. In fact, NERSC worked with Cray to define many of the performance and functional requirements for CNL during our negotiations.

Further, NERSC and Cray have established a Center of Excellence for Resource and Storage Management. Two of the first tasks for this Center are to incorporate Berkeley Lab's Checkpoint/Restart (BLCR) [14]and develop the "Petascale I/O Interface". The Petascale I/O Interface is designed to provide the ability to integrate the XT systems with different parallel file systems. This will allow the XT compute nodes to integrate with NERSC's Global File system[15] (NGF) which currently provides a single, uniform namespace across all NERSC systems (5 systems, 4 architectures from 4 vendors) at very high performance. Hence, the eventual software target for Franklin is

- SuSE SLES 9.0 – 10.0 Linux on Service Nodes
- Compute Node Linux O/S for all compute nodes
  - Cray's light weight Linux kernel
- Portals communication layer
  - MPI, Shmem
- Compute node integration with the NERSC Global Filesystem
  - Global file systems (e.g. GPFS, Lustre, others) directly accessible from compute nodes with a "Petascale I/O Interface"
- Torque with Moab
  - Most expected functions including Backfill, Fairshare, advanced reservation
- Checkpoint Restart
  - Based on Berkeley Linux Checkpoint/Restart (Hargrove)
- Application Development Environment
  - PGI compilers - assembler, Fortran, C, UPC, and C++
  - Parallel programming models include MPI, and SHMEM.
  - Libraries include SCALAPACK, SuperLU, ACML, Portals, MPICH2/ROMIO.
  - Languages and parallel programming models shall be extended to include OpenMP, and Posix threads but are dependent on compute node Linux
  - Totalview or equivalent to 1,024 tasks
  - Craypat and Cray Apprentice
  - PAPI and Modules

Beyond 2007, NERSC and Franklin will continue to evolve and expand. For example, in the winter of 2007/2008 NERSC has an option to upgrade Franklin to quad core opterons. While these nodes have 4 times the peak performance increase, initial assessment (see the related paper) [16] is that applications are unlikely to see more than a 2x sustained performance increase. NERSC can double the memory per node to keep the constant B/F ratio at that time.

In the spring to summer 2008 NERSC will have the option – based on how well it performs according to our continuing metrics – to change to Compute Node Linux. In the Winter/Spring 2009 there is also the option to migrate to a 1 Peta Flops/s system.

In 2011, NERSC will take occupancy of a new, 40,000 sf computer facility so it can continue to meet DOE's computational demands.

Conclusion

NERSC supports the rich and diverse workload of the DOE Office of Science. In doing so, it provides service thousands of scientists and hundreds of challenging projects. NERSC has a history of providing robust and highly effective systems, managed to provide high utilization while at the same time have the majority of its computational time go to large scale (1,024+) jobs.

As the next step in that line of success, the Cray XT-4 was competitively selected as NERSC's next major computational system, using the proven PERCU methodology. The expectation is the XT-4 will greatly enhance NERSC's computational infrastructure and continue to make its computational science user community highly productive.

Finally, NERSC is excited at the potential of our re-energized relationship with Cray.

Acknowledgments

The author would like to thank colleagues, users and vendor staff for a wealth of experiences we have shared together. In particular, the NERSC staff who does the work to assess, test, operate and improve the NERSC systems for our user community.

About the Author

Bill Kramer is the General Manager of NERSC and Head of the High Performance Computing Department at Lawrence Berkeley National Laboratory (LBNL). He is responsible for all aspects of the NERSC Facility and shares division-wide management responsibilities for the NERSC Division. At NERSC, Bill led the acquisition, testing, and introduction of the early, very large systems, including his 18th computational supercomputer - the largest Cray XT-4. He was instrumental in managing the paradigm shift for NERSC from traditional vector computing to massively parallel and is one of the primary contributors to LBNL's "Science Driven Computer Architecture" initiative, which has yielded ideas like the Virtual Vector Architecture (ViVA) and the "Blue Planet" node. Under his leadership NERSC was first to use checkpoint/restart in a MPP production environment and the first to demonstrate the ability to manage very large capability systems with utilization over 90-95%, among other innovations. Bill led the development and implementation of a new 28,000-square-foot computing facility from advocacy and requirements to implementation.

Bill's research interests include large-scale system management, scheduling, performance evaluation, and the integration of high performance networks and computers. Bill was the General Chair of the SC 05 conference in Seattle. Prior to Berkeley Lab, Bill worked at the NASA Ames Research Center, where he was Branch Chief of the NAS Computational Services. In this role, he was responsible for all aspects of operations and customer service for NASA's principal supercomputer center. He was responsible for putting the first UNIX supercomputer into production and led the introduction of cluster computing at NAS. He holds a BS and MS in Computer Science from Purdue University and a ME in Electrical Engineering from the University of Delaware. Bill is completing his PhD in Computer Science at UC Berkeley.

---

[1] Horst D. Simon, William T. C. Kramer, David H. Bailey, Michael J. Banda, E. Wes Bethel, Jonathon T. Carter, James M. Craw, William J. Fortney, John A. Hules, Nancy L. Meyer, Juan C. Meza, Esmond G. Ng, Lynn E. Rippe, William C. Saphir, Francesca Verdier, Howard A. Walter, Katherine A. Yelick. Science-Driven Computing: NERSC's Plan for 2006-2010., May 18, 2005, LBNL Technical Report, LBNL-57582. http://www-library.lbl.gov/docs/LBNL/575/82/PDF/LBNL-57582.pdf

[2] LBNL-58927. A. Aspuru-Guzik, V. Batista, E. W. Bethel, J. D. Borrill, J. Chen, P. Colella, D. J. Dean, T. DeBoni, D. Donzis, S. Ethier, A. Friedman, A. Harrison, E. Hawkes, M. Head-Gordon, B. E. Hingerty, S. C. Jardin, Y. Jung, D. Keyes, W. A. J. Lester, F. Lui, M. Mavrikakis, A. Mezzacappa, D. Olson, K. Refson, C. Ren, D. Rotman, R. Ryne, V. Sankaran, C. Sovinec, D. Spong, G. Sposito, G. M. Stocks, R. Sugar, D. Swesty, H. Wang, V. Wayland, and P. K. Yeung (2005). DOE Greenbook: Needs and Directions in High Performance Computing for the Office of Science. A Report from the NERSC User Group.

[3] D.A. Reed, editor, *Report of the Workshop on the roadmap for the revitalization of high-end computing*, Computing Research Association, June 2003.

[4] D. Keyes, editor, *Science case for large-scale simulation*. DOE Office of Science Workshop, June 2003.

---

[5] LBNL/PUB-904 (2006). D. Skinner, F. Verdier, H. Anand, J. Carter, M. Durst, and R. Gerber (2005). Parallel Scaling Characteristics of Selected NERSC User Project Codes. http://www-library.lbl.gov/docs/PUB/904/PDF/PUB-904_2006.pdf |

[6] http://www.nersc.gov/news/reports/

[7] http://www.nersc.gov/news/annual_reports/

[8] LBNL-58868. W. Kramer, J. Shalf, and E. Strohmaier (2005). The NERSC Sustained System Performance (SSP) Metric. http://www-library.lbl.gov/docs/LBNL/588/68/PDF/LBNL-58868.pdf

[9] Bucher, Ingrid and Joanne Martin, *Methodology for Characterizing a Scientific Workload* ,1982, Los Alamos National Laboratory Technical Report, LA-UR-82-1702, Los Alamos, NM 87545

[10] Wong, Adrian T., Leonid Oliker, William T. C. Kramer, Teresa L. Kaltz, and David H. Bailey, *Evaluating System Effectiveness in High Performance Computing Systems*. Proceedings of SC2000, November 2000

[11] Wong, Adrian T., Leonid Oliker, William T. C. Kramer, Teresa L. Kaltz, and David H. Bailey, *System Utilization Benchmark on the Cray T3E and IBM SP*, 5th Workshop on Job Scheduling Strategies for Parallel Processing, May 2000, Cancun Mexico.

[12] Kramer, William and Clint Ryan, *Performance Variability on Highly Parallel Architectures*, the International Conference on Computational Science 2003, Melbourne Australia and St. Petersburg Russia, June 2-4, 2003.

[13] Skinner, David and William Kramer, *Understanding the Causes of Performance Variability in HPC Workloads*, 2005 IEEE International Symposium on Workload Characterization (IISWC-2005), October 6-8, 2005, Austin, Texas

[14] Duell, J., Hargrove, P., and Roman., E. *The Design and Implementation of Berkeley Lab's Linux Checkpoint/Restart.* Berkeley Lab Technical Report (publication LBNL-54941) and http://ftg.lbl.gov/CheckpointRestart/CheckpointRestart.shtml

[15] http://www.nersc.gov/nusers/resources/NGF/

[16] Jonathan Carter, Yun (Helen) He, John Shalf, Hongzhang Shan, and Harvey Wasserman, National Energy Research Scientific Computing Center (NERSC), **The Performance Effect of Multi-Core on Scientific Applications**, Cray User Group, May 2007, Seattle, WA