The NERSC Sustained System Performance (SSP) Metric

William T.C. Kramer, John M. Shalf, Erich Strohmaier
wtkramer@lbl.gov

Background

Most plans and reports recently discuss only one of four distinct purposes benchmarks are used. The obvious purpose is selection of a system from among its competitors, something that is the main focus of this paper. This purpose is well discussed in many workshops and reports. The second use of benchmarks is validating the selected system actually works the way expected once it arrives. This purpose may be more important than the first reason. The second purpose is particularly key when systems are specified and selected based on performance projections rather than actual runs on the actual hardware. The third use of benchmarks, seldom mentioned, is to assure the system performs as expected throughout its lifetime[1], (e.g. after upgrades, changes, and regular use.) Finally, benchmarks are used to guide system designs, something covered in detail in a companion paper from Berkeley's Institute for Performance Studies (BIPS).

HPC procurements require more sophisticated methods to gauge the effectiveness of the system than the "speeds and feeds" that are typically supplied by hardware vendors or from simple, one dimensional tests. HPC systems are typically presented based on raw hardware characteristics -- the interconnect bandwidth, the peak floating point performance of the processors, the number of processors, the memory bandwidth, etc. The HEC Community is left with the situation that we sometimes have a good idea of how much work a system can produce if everything is perfect, but little idea of what it will do in the real world. Without additional information, the systems may end up being compared based on their peak performance even though they are likely to deliver only a fraction of their peak performance. Selecting a system with benchmarks and tests is akin to a constrained optimization problem. There is serious concern however whether a strict optimization approach goes too far - back to quantitative scoring rather than an informed evaluation. Having worked in both realms, the qualitative approach produces superior results and is much more likely to move to the revolutionary change rather than a slow, evolutionary system improvement. Rigid weighing and scoring of values will be a step backwards in this regard. Using qualitative judgment based on quantitative date works best. Hence any methods investigated have to be on the "fuzzy" side of Operations Research. HEC Systems are too complex for simple assessment. These systems have components based at least in part on commodity technology. Since HEC systems are most often used to run highly parallel, tightly coupled application codes, the interaction of components plays a disproportional role in how well the systems meet expected productivity levels.

The overriding question for an HEC procurement team is: "What performance will this system deliver to *our* workload?" A procurement team needs a normalizing metric that tells them how efficiently the architecture can execute their typical workload given the proposed system's peak performance so that different systems can be compared on an equal footing. The general methodology for approaching the problem was outlined in a 1982 technical report by Ingrid Bucher and Joanne Martin entitled "Methodology for Characterizing a Scientific Workload." These principles are embodied in the construction of the NAS kernel benchmarking program (Bailey & Barton, 1985), and are consistent with the approach taken for the Sustained System Performance (SSP) metric developed by NERSC for its procurements. The methodology outlined in 1982 by Bucher and Martin is as follows:

1) Workload Characterization: Conduct a workload characterization study to understand the

[1] A brief comment on why this is important. One system at NERSC consistently slowed down by 5% every month it was up. A reboot would return the system to the expect performance level. This was only detected because of proactive and continuing benchmark runs that combined to give SSP values. Other reasons include detecting variation over time.

type and distribution of jobs run on your systems.
2) Representative Subset: Select a subset of programs in the total workload that represent classes of applications fairly. Assign weights according to usage.
3) Application Kernels: Designate portions (kernels) of the selected programs to represent them.
4) Collect Timing: Time the kernels on the system under test.
5) Compute Metric: Compute the weighted harmonic mean of kernel execution rates to normalize the "peak" performance of the system to a number that would likely be delivered in practice on the computing center's mix of applications.

As stated, the implementation of this methodology suffers from a number of pragmatic problems:

1) It is difficult to collect an accurate workload characterization given that many tools for collecting such information can affect code performance and even the names of the codes can provide little insight into their function or construction (the most popular code, for instance, is 'a.out').
2) NSF Centers and DOE centers like NERSC, support a broad spectrum of users and applications. The resulting workload too diverse to be represented by a small subset of programs.
3) The complexity of many supercomputing codes has increased dramatically over the years. The result is that extracting a kernel is an enormous software engineering effort, and maybe enormously difficult.
4) The complexity of the software can make codes more difficult to port, and particularly, to tune for a wide variety of architectures.
5) The weighted harmonic mean of timings presumes the applications are either serial (as was the case when the report was first written) or that they are run in parallel at same level of concurrency. However, applications are typically executed at different scales on the system and the scale is primarily governed by the science requirements of the code.
6) This metric does not take into account other issues that play an equally important role in decisions such as the effectiveness of the system resource management, consistency of service, the reliability/fault-tolerance of the system. The metric also is not accurate in judging heterogeneous processing power within the same system – something that may be very important in the future
7) Finally, the simple metric does not take into account the delivery date of the system. A system delivered late will provide less value to the user community than one delivered on time. It is also the case that very large systems are delivered in phases, and there is no way to assess the value of how much computing should be in each phase.

The NERSC Approach to Procurement Benchmarks

With these issues in mind, we introduce the methodology developed for NERSC procurements to develop a normalizing metric for inter-machine comparisons.

1) *Workload Characterization:* Use non-invasive profiling tools to collect workload characterization data. For instance, the IPM tool collects communication and hardware counter profiles of parallel codes without a significant impact on code performance. This data is followed by direct contacts with users in order to properly identify the code, the numerical algorithm, and the scientific purpose behind the codes that are run by a particular user or group. There is no automated way to collect the latter information with accuracy.
2) *Representative Subset:* It is impossible at this point to get full coverage of a diverse workload, so the selection of an appropriate subset of codes ranks codes in part by their prominence in the workload (a combination of the total number of cycles they consume and their importance to their respective scientific community). The subset is chosen with both past performance and future workloads in mind. By having close interaction with the scientific community[2], emerging algorithmic methods may be included even if they do not

[2] The fosters this communication in a number of ways, with month discussions with users, semi-annual meetings, large amounts of informal interaction, etc. However, there is also a formal process in which the

play a dominant factor in the current workload since the systems being acquired may have a life time of 4 to 6 years after the date of the RFP. Too much balance on past workload may mean slow evolution of the computing capability, while too little attention will mean large and disjoint transitions for the science community.

3) *Application Kernels:* The selection of representative codes is also influenced by what codes are both portable, assuring there is an appropriate distribution of algorithmic methods across the suite and are made available by the scientists. In complex codes, it is not feasible to extract all the necessary computational kernels. While kernels are very useful as part of the test suite to measure single parameters they are inadequate for selection unless the workload is essentially homogeneous. Furthermore, because kernels are greatly simplified, they are not capable of capturing the complex interactions of how difference system features interact. Thus they are not sufficient for either the initial or the on-going performance validation of HEC systems. Complex tests are needed to assess complex interactions that cannot be foreseen with a set of 1 dimensional kernel tests. The emerging of multi-dimensional, *parametric* kernels such as APEX may improve this situation. Kernels also play a key role in the fourth use of benchmarks, influencing system design. Consequently, there are many advantages to using full application code-base.
4) *Timing:* RFPs expect vendors to run the benchmarks and providing the results and often projections for as yet unreleased hardware as part of their response to the RFP. Since RFPs are periodic, and there are issues with vendors gathering resources to run benchmarks at scale, it may be necessary to embark upon a continuous effort outside of the RFP process to actively benchmark emerging systems from vendors. This would also allow a continuous feedback process with the vendors that would enable them to optimize their system architecture over time. This is a process that LBL refers to as "Science Driven Architecture" and is consistent with the practices laid out in the 2001 PITAC, 2003 HECRTF and 2004 NRC reports.
5) *Metrics:* The data collected from scalability studies, where the code is run at a variety of scales, is not necessarily a good reflection of what the production throughput of the code will be. The codes must be executed at the scale that their implementers intend to use for production runs. Therefore, the total wall clock resource utilization at scale forms the basis for the metric rather than the scalar execution rate. The harmonic mean of wall clock times is probably appropriate in cases where the workload balance is expected to remain relatively static. But other methods of calculation a single measure are necessary for complex workloads and complex systems.

The NERSC-5 SSP

The effectiveness of a benchmarking metric for predicting delivered performance is founded on its accurate model of the target workload. A static benchmark suite will eventually fail to provide an accurate means for assessing systems. Several examples, including LINPACK, show that over time, kernel benchmarks become less of a discriminating factor. This is because once a simple benchmark gains traction in the community, system designers customize their designs to do well on that benchmark. The Livermore Loops, SPEC, LINPACK, NAS Parallel Benchmarks, etc. all have this issue. It is clear LINPACK now tracks peak performance in the large majority of cases. Erich Strohmaier showed through statistical methods, that within 5-7 years, only three of the eight NPBs were true distinguishers of system performance.

Thus it should not be a goal that the kernel benchmarks that they are long lived – except as regression tests to make sure the features that make them work well stay in the design scope. There will have to be a constant introduction/validation of the “primary” tests that will drive the design features for the future, and a constant “retirement” of the benchmarks that are no longer strong discriminators. Consequently, the SSP metric continues to evolve to stay current with

---

NERSC User Group generates a *Greenbook* or requirements every 3-4 years that cover their current approaches, future plans and needs. This document is key input to many things, including the major decisions on system selection.

current workloads and future trends. We will now describe the 5th generation of the SSP metric that is being used for the NERSC-5 procurement.

The table below shows the set of applications used in NERSC-5 SSP. The procurement team looks at all benchmark values and assesses their implications. It also uses kernel benchmarks such as the NAS Parallel Benchmarks, CPUbench, Membench and IObench. But the SSP provides the best overall expectation of performance and price performance for a proposed solution. It also is attractive to vendors who prefer composite tests over many discrete tests.

Application	Science Area	Basic Algorithm	Language	Library Use	Required Concurrency
CAM3	Climate (BER)	CFD, FFT	FORTRAN 90	netCDF	64 and 240
GAMESS	Chemistry (BES)	DFT	FORTRAN 90	DDI, BLAS	64and 384[3]
GTC	Fusion (FES)	Particle-in-cell	FORTRAN 90	FFT(opt)	64 and 256
MADbench	Astrophysics (HEP & NP)	Power Spectrum Estimation	C	Scalapack	64 and 256 and projection required to 1,024
MILC	QCD (NP)	Conjugate gradient	C	none	64 and 256 Projection required to 2,048
PARATEC	Materials (BES) Nanoscience	3D FFT	FORTRAN 90	Scalapack	64 and 256
PMEMD	Life Science (BER)	Particle Mesh Ewald	FORTRAN 90	none	64 and 256

The SSP value is now calculated as the geometric mean of the Flop rate of each program. In the past, a weighed harmonic mean and a straight arithmetic mean were used at NERSC, but recent experience indicates the geometric mean, while giving somewhat lower value, provides a better balance amongst the codes used and is a better predictor. It should also be noted that the SSP does not have to be made up only of full applications. SSP can be used with kernels and indeed the first time SSP was used at NERSC, it was calculated as the average of the NAS Parallel Benchmarks rates.

The capability of a system is then represented by the Sustained Performance (SSP) of a system integrated over a given time period. The SSP number (in Tflop/s-years) indicates the effective average performance of the system on a center's scientific workload at any given point in time. In order to enable a comparison between systems that are bid as part of a procurement process, the "capability" of the system is the total area under the SSP curve over a given time period (NERSC uses 3 years). Since the capability of the system can be quantified for the entire workload at any point in time, and indeed, throughout any period, it is then possible to assess the price performance, or “value” of the system by dividing the SSP metric by the cost of the system – basically Tflop/s-years per $. This gives an important and straight forward way to determine the system with the best value out of all the system proposed.

[3] Note this benchmark and its data input is identical to the DOD HPCMP TI06 Gamess benchmark. NERSC and HPCMP have coordinated benchmarks for the NERSC-5/TI06 RFPs.

Different vendors introduce technology at different times, and it may be to sites 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 in order to have the best price performance. 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 parties.

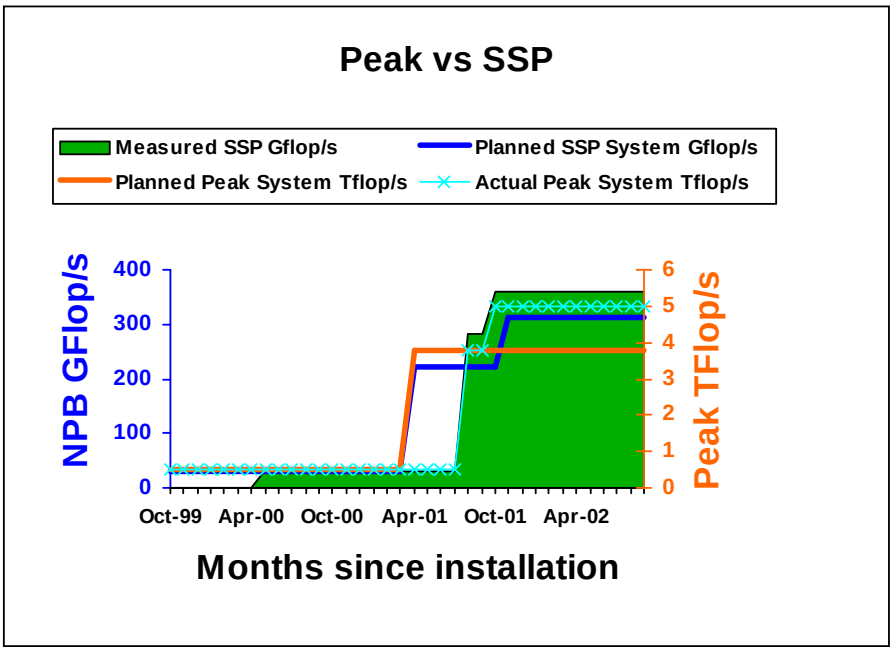

Figure 1: Peak vs. measured SSP performance.

The Effective System Performance (ESP) Metric

High performance scientific computer systems are traditionally compared using individual job performance metrics. However, such metrics tend to ignore high-level system issues, such as how effectively a system can schedule and manage a varied workload, how rapidly the system can launch jobs, and how quickly it can recover from a scheduled or unscheduled system outage. The productive work that can be extracted from a computational system is dependent not only on computational performance but also on the software infrastructure. In particular, resource management functionality (e.g. scheduling, job launch and checkpoint/restart) has become an increasingly important issue given the difficulty of managing large-scale parallel computers.

Therefore, the SSP metric must be adjusted in two ways. The first adjustment is by a metric that takes into account the effective throughput of the job scheduling system. For instance, a job scheduler that must schedule jobs on a torus interconnect (egg. and XT3 or a BG/L system) may have additional overheads to make room for new jobs or may not be able to densely pack the torus with work. Similarly, some job launchers have considerable startup overhead for launching parallel jobs that cuts into the effective throughput of the system. So the scheduler candidate system must be subjected to a simulated workload in order to estimate the efficiency with which it can schedule the resources of the system. NERSC refers to this metric as the ESP (Effective

System Performance)[4]. ESP[5] has several characteristics that set it apart from traditional throughput tests. First, it is specifically designed to simulate "a day in the life" of a supercomputer. It has scheduling shifts that mimic the typical supercomputer center which often changes priority of job times between daytime and night. ESP also requires shifting between multiple jobs running and a single "full configuration" job and in fact requires the full configuration jobs run with different scheduling parameters. Finally it attempts to estimate system management effort involved in running large scale computers.

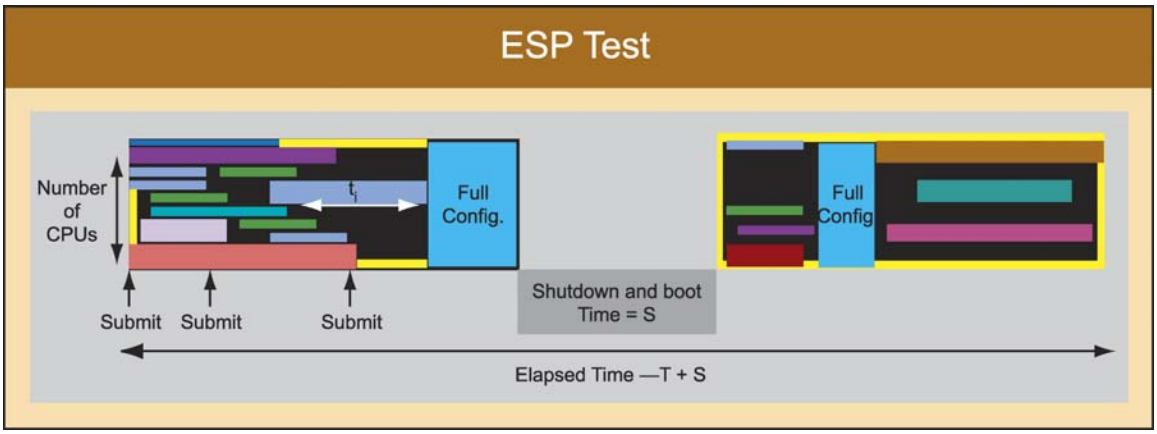

Figure 2: The Effective System Performance Test schematic shows how the operational paradigm of a system is challenged by the test

The second adjustment to SSP is variability – which increasingly causes lost capability in HPC systems[6]. NERSC uses the variability shown by the SSP metric as a prime indicator of how variable the system is. Both the ESP metric and variation are assessed with proposals, and during the life of the system. The SSP metric is used to assure the system still operates as expected after upgrades and through the life of the system.

Conclusion

The SSP metric is the most important performance metric of the procurement and contract. Vendors are required to supply a promised aggregate life-time integral of the SSP metric on the supplied system. On the other hand, the vendor has flexibility in how best to provide the required performance although at NERSC any major change requires concurrence. This means that the SSP metric has to be measured in regular intervals during the life-time of the system. On-going use of benchmarks ensures that all effects on performance from system upgrades or deterioration, system software and compiler upgrades, and communication library changes are reflected in the actual measured SSP value.

Hence, the Sustained System Performance metric, along with quantifying the impact of Effective System Performance and variability, provides an excellent approximation of how well HEC will serve the scientific community.

---

[4] 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

[5] 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.

[6] 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.