Advanced Scientific Computing Advisory Committee Petascale Metrics Report

28 February, 2007

Petascale Metrics Panel [a subcommittee of the Department of Energy Office of Science Advanced Scientific Computing Advisory Committee]: F. Ronald Bailey, Gordon Bell (Chair), John Blondin, John Connolly, David Dean, Peter Freeman, James Hack (co-chair), Steven Pieper, Douglass Post, Steven Wolff

Report Contents

Petascale Metrics Panel Report
Executive Summary

Petascale Metrics Panel: F. Ronald Bailey, Gordon Bell (Chair), John Blondin, John Connolly, David Dean, Peter Freeman, James Hack (co-chair), Steven Pieper, Douglass Post, Steven Wolff

Introduction

Petascale computers providing a factor of thirty increase in capability are projected to be installed at major Department of Energy computational facilities by 2010. The anticipated performance increases, if realized, are certain to change the implementation of computationally intensive scientific applications as well as enable new science. The very substantial investment being made by the Department demands that it examine ways to measure the operational effectiveness of the petascale facilities, as well as their effects on the Department's science mission. Accordingly Dr. Raymond Orbach, the Department of Energy Under Secretary for Science, asked this panel, which reports to the Office of Advanced Scientific Computing Research Advisory Committee (ASCAC), "...to weigh and review the approach to performance measurement and assessment at [ALCF, NERSC, and NLCF], the appropriateness and comprehensiveness of the measures, and the [computational science component] of the science accomplishments and their effects on the Office of Science's science programs. Additionally, we were asked to consider "… the role and computational needs over the next 3-5 years".

Overview

Throughout their recent 50+ year history beginning with a few, one-hundred kiloflops computers, through the era of high-performance pipelined scalar and vector machines delivering tens of megaflops, to the modern teraflops, scalable architectures that promise petaflops speeds by 2010, supercomputer centers have developed and refined a variety of metrics to characterize, manage and control their operations. This report distinguishes between control metrics, which have specific goals which must be met, and observed metrics, which are used for monitoring and assessing activities.

In the report, we will discuss our findings given as *beliefs*, will provide *suggestions* for action, and will provide *recommendations and metrics* that pertain to the DOE High-Performance Computing Centers, Computational Science Projects, and management processes in these six elements of the charge:

1 &2: Facilities and projects metrics

3 & 4: Science accomplishments and their effects on the Office of Science's programs

5 & 6: Evolution of the roles of these facilities and the computational needs over the next 3-5 years.

Recommendations and Conclusions

Elements 1 & 2: Facilities and Project Metrics.

In addressing the "approach to performance measurement and assessment at the facilities" and their "appropriateness" and comprehensiveness" it became immediately clear that while useful metrics such as uptime and utilization for Centers have evolved for decades and are in wide use, the petascale challenge to projects introduces the need for much deeper understanding of the scientific project, the application codes, computational experiment management, and overall management of the scientific enterprise.

The panel believes that the introduction of new Center metrics is unnecessary and could be potentially disruptive. After careful consideration, the panel identified four existing "control" metrics that can be used for evaluating Centers' performance:

1.1 User Satisfaction (overall) of provided services, usually obtained via user surveys. A number of survey questions typically constitute a single metric.

1.2 System Availability in accordance with targets that should be determined for each machine, based on the age, capabilities and mission of that machine. These should apply after an initial period of introductory/early service. Although *reported* overall availability and delivered capacity should be of great interest, they should **not** be the primary measures of effectiveness because of their potential for misuse.

1.3 Problem Response Time in responding to users' queries regarding the variety of issues associated with complex computational systems as measured by appropriate standardized trouble reporting mechanisms.

1.4 Support for capability-limited problems at the leadership class facilities, measured by tracking and ensuring that some reasonable fraction of the deliverable computational resource is dedicated to scientific applications requiring some large fraction of the overall system. In addition to the control metric of a certain percent of the resource being used by large-scale jobs, the tracking mechanism for capability-limited jobs should include statistics on the expansion factor, which is a measurement of job turnaround time.

Centers use a number of additional "observed" metrics. The Panel believes these should be available to ASCR to inform its policy setting and facilities planning activities. As discussed in the body of the report, "observed" metrics, such as "system utilization[1]", while valuable for characterizing the Centers' operations, have the potential for distorting and constraining operation when used as management controls.

Measuring the status and progress of the scientific projects that utilize the centers on a continuous basis is equally important aspect of understanding the overall system.

2.1 Computational Project Evaluation based on a standard checklist described in Appendix 1 that includes the project goals and resources, centers resource

[1] Managing a center to have high system utilization usually has the effect of increasing job turn-around time, reducing the ability to run very large jobs, or both.

requirements, tools, software engineering techniques, validation and verification techniques, code performance including the degree of parallelism and most importantly the resulting scientific output.

2.2 **Code Improvement** measurement that includes mathematics, algorithms, code, and scalability. The Panel recommends and agrees with the Centers' suggestion of a halving the time to solution every three years – corresponding to one-half the rate of Moore's Law improvement.

Charge Elements 3 & 4: Science Accomplishments and Effects on the Programs

The Panel proposes the following recommendations aimed at addressing the science accomplishments of the centers and their effects on the Office of Science's programs as requested by the charge:

The Panel suggests that peer reviews of the projects be based on both their scientific and computational science merits for allocation of computational resources along the lines of the INCITE program.

The Panel further suggests that the Centers provide an appropriate and significant level of support for the scientific users of their facilities in light of the unique capabilities of the facilities and the leading edge computational science needs of the user base.

The Panel recommends the following be "reported and tracked" in a fashion similar to the centers "control" metrics to assist in the measurement of scientific output from its projects:

3.1 **Publications, Code & Datasets, People, and Technology Transfer** (as given in Appendix 1, Item 6) goes beyond the traditional scientific publication measures and extends to code produced, training and technology transfer.

3.2 **Project Milestones versus the Proposal's Project Plan** is a near term and useful metric as a measure of progress on the path toward well-defined scientific goals as reviewed by the Program Offices.

3.3 **Exploiting Parallelism and/or Improved Efficiency: aka Code Improvement** How well scientific applications take advantage of a given computational resource is key to progress for computational science applications. The improvement of algorithms that serve to increase computational efficiency is an equally important measure of code effectiveness. Scalability of selected applications should double every three years as described in the previous section as Code Improvement 2.2.

3.4 **Break-throughs; an immeasurable goal:** The panel could not identify metrics or any method that could be used to anticipate discoveries that occur on the leading edge of fundamental science.

The Panel makes the following suggestions to address the Computational Resource Effects on the Office of Science's science programs:

The Panel *suggests* that a clear process be implemented to measure the use and effects of the computational resources on the projects within each SC program office. The Centers will benefit from the feedback which will ensure that the computational facilities are optimally contributing to the advancement of science in the individual disciplines.

The Panel *suggests* that each SC office report the total investment in all projects, by including a rough conversion of computer time to dollars. The panel believes that computer resources need to be treated in a substantially more serious and measured fashion by both the program offices and project personnel.

The Panel *suggests* the process of allocating computer time at the Centers through the program offices be re examined in light of the diversity of architectures. Given the variety of platforms at the Centers and user code portability, the efficiency of a particular code will be highly variable.

Charge Elements 5 & 6: The Evolution of the Facilities' Roles and Computational Needs

The Panel believes the centers are on a sound trajectory to supply the broad range of needs of the scientific community, which will allow SC programs to maintain their national and international scientific leadership.

The Panel believes it is too early to assess the impact of the expanded role of INCITE on the facilities demand. Based on just the recent three decades of scientific computers whose performance has doubled annually, there is no end to the imaginable applications or amount of computing that science can absorb.

Regardless of their long-term evolution the panel *suggests* that project-integrated consulting should constitute a portion of the budget for all the centers in future funding scenarios. Petascale computers are going to be difficult to use efficiently in most applications because of the need to increase parallelism in order to get same fraction of use. As the SciDAC initiative demonstrated, the scientists need the help of computing professionals to make good use of the resources.

Final observations

The Panel makes these observations on the management of ASCR's portfolio with respect to the Facilities management and suggests two areas for possible improvement:

1. increasing communications between ASCR and Scientific Program Offices, and
2. improving the capability and support of various scientific codes and teams to use the petascale architectures by both general and domain specific Centers support

Concluding remarks

The Panel believes we have provided useful, actionable suggestions and recommendations based on our experience and those of our colleagues together with our recent review of the Centers and projects. We hope the Department will find it useful as it addresses the petascale challenge.

Petascale Metrics Panel Report

"If you can not measure it, you can not improve it." – Lord Kelvin

"The purpose of computing is insight, not numbers." -- Richard W. Hamming

Petascale Metrics Panel: F. Ronald Bailey, Gordon Bell (Chair), John Blondin, John Connolly, David Dean, Peter Freeman, James Hack (co-chair), Steven Pieper, Douglass Post, Steven Wolff

The petascale computing challenge

Petascale computers providing a factor of thirty (30) increase in peak operation rate, primarily through increasing the number of processors, are projected to be installed at the Department of Energy's Office of Advanced Scientific Computing Research (ASCR) computational facilities during 2007-2010. Two capability systems at Oak Ridge (ORNL) and Argonne National Laboratories (ANL), with peak performances of one petaflops and 200-500 teraflops, respectively, and a capacity system at the Lawrence Berkeley National Laboratory (LBNL) with a peak capacity of 500 teraflops are planned.

These new computing systems are certain to change the nature of computationally intensive scientific applications because of new capabilities and the challenge to efficiently exploit their capabilities. In order to exploit this change and the opportunity it provides, it is important to look both at how centers operate and how they interact with the scientific projects they serve. Interactions of particular interest are the workflow of scientific projects, scalability of application codes and code development practices. While providing new science opportunities, the increased computational power implies the need for more scalable algorithms and codes, new software and dataset management tools and practices, new methods of analysis, and, most of all, greater attention to managing a more complex computational experiment environment.

The Cray XT at ORNL's NLCF illustrates the challenge. At 50% peak processor computing rate the 10,400 processor Cray (Jaguar) at NLCF supplies 91 Million processor hours or 235 petaflops hours annually. On average, its 20 projects need to be running 500 processors continuously and in parallel all year long. Clearly large teams are necessary just to manage the programs, resulting computational experimental data, and data analysis.

Dr. Raymond Orbach, the Department of Energy Under Secretary for Science, charged ASCR's advisory committee, ASCAC, to examine metrics for petaflops computing that affects DOE's computing facilities at ANL (ALCF), LBNL (NERSC) and ORNL (NLCF) and the impact and interaction with scientists and sponsoring scientific programs. The ASCAC appointed our panel, (the Panel) to carry out this charge.

Panel response and approach to the Orbach Charge[2]

The Panel reviewed the charge described by Under-secretary Orbach of 10 March 2006 and identified six elements requiring analysis that are described herein.

The six elements of the “charge” constitute the structure for the six sections of the report:

1. the approach to performance measurement and assessment at these facilities;
2. the appropriateness and comprehensiveness of the measures; *(The Panel examined Project metrics as complementary and essential measures.);*
3. the science accomplishments;
4. their (*i.e. computational resources)* effects on the Office of Science’s science programs;
5. the evolution of the roles of these facilities; and
6. the computational needs over the next three - five years, *so that SC programs can maintain their national and international scientific leadership*.

In the last section, the Panel comments on “observed strengths or deficiencies in the management of any component or sub component of ASCR’s portfolio” as requested in the charge.

The Panel convened six times by teleconference to identify metrics relevant to the centers and computational science projects that it wanted to better understand, and engaged in a number of activities to address the charge. These activities included the submission of questionnaires to the centers regarding the centers operations and selected projects, meetings with centers directors, and a week-long meeting of the panel and center representatives to review metrics, center operations, and examples of large computational science projects. These discussions also included presentations by Brad Comes and Doug Post on metrics employed within DOD centers, and a project checklist-survey aimed at understanding the nature and state of DOD’s engineering-oriented high-performance computing applications. Peter Freeman (NSF), Michael Levine (PSC), and Allan Snavely (SDSC) discussed the operation of the NSF centers, and their metrics. Appendix 0 gives the contents of the 270 pages of the background Web Appendices http://research.microsoft.com/users/gbell/ASCAC_Metrics_Panel.htm.

After reviewing this comprehensive, multi-faceted charge, the Panel concluded that an important aspect of this report should include observations and suggestions on how the Secretary of Energy can follow the effectiveness of the scientific output of the Office of

[2] “The sub-panel should weigh and review the approach to performance measurement and assessment at these facilities, the appropriateness and comprehensiveness of the measures, and the science accomplishments and their effects on the Office of Science’s science programs. Additionally, the sub-panel should consider the evolution of the roles of these facilities and the computational needs over the next three - five years, so that SC programs can maintain their national and international scientific leadership.

In addition to these ratings, comments on observed strengths or deficiencies in the management of any component or sub-component of ASCR’s portfolio and suggestions for improvement would be very valuable.”

Science (SC) in addressing the mission of the Department. The Under Secretary for Science in turn, needs to know if the investment in present and planned petascale computational facilities (and the scientific projects they support) are producing, and will continue to produce, scientific results that are commensurate with the infrastructure investment.

Therefore, the Panel interprets as a part of its charge to investigate whether the Under Secretary has sufficient information – generated by metrics and other assessment mechanisms – to assist in answering the stated question. We note that other information, such as budgets, past history, science objectives, strategies, etc., is needed to fully answer the question. The Panel has not addressed whether or not this additional information is available. Addressing this overall question of effectiveness in broader terms is also clearly outside the scope and competence of the Panel.

DOE's Computational Science System

Figure 1 is the Panel's attempt to portray the system being analyzed. Simplistically, funds and scientific questions are the input and science is the output. Two independent, funding and management paths are responsible for experimental and computational science resources. First, ASCR funds facilities to deliver computational resources such as computer time, storage, and networking and to SIDAC to make coupled advances in computer and computational sciences. The second is the direct funding of projects, i.e., scientific personnel. This funding is provided by SC and other government agencies, such as NSF. Scientific projects from other agencies also apply to use the facilities, for example, through the INCITE program. A variety of mechanisms determine the computing resources particular projects receive including: direct allocations by ASCR and other SC program offices, and peer reviews.

Within the envelope of our charge, the Panel focused in some detail on the following two, key structural and "work" components of the Office of Science that are responsible for the scientific output:

1. the three *Facilities* or Centers supported by the Office of Advanced Scientific Computing Research (ASCR) consisting of ALCF (ANL), NERSC (LBNL), and NLCF (ORNL); and
2. the multitude of *Science and Engineering Projects* (supported by the other SC program offices and by other agencies) that utilize the computational services of the Facilities in making scientific discovery and progress.

The Panel believes all elements should be managed in a coupled way. The main overarching question is what degree of coupling and management processes is required to ensure the appropriate trade-offs between the funding of the ASCR computational infrastructure, and the investment in the scientific enterprise that exploits this infrastructure?

In addition to this broader examination of the investment in computational infrastructure, we have reviewed metrics that may be useful at various levels within the Office of Science, especially ASCR and the other SC Program Offices. The basic material that was used in the preparation of this report is contained in the 270 page Web Appendix located at http://research.microsoft.com/users/gbell/ASCAC_Metrics_Panel.htm. Appendix 0 gives the table of contents of the Web Appendix. The appendix material includes questionnaires, responses, DOE and DOD centers presentations and selected projects.

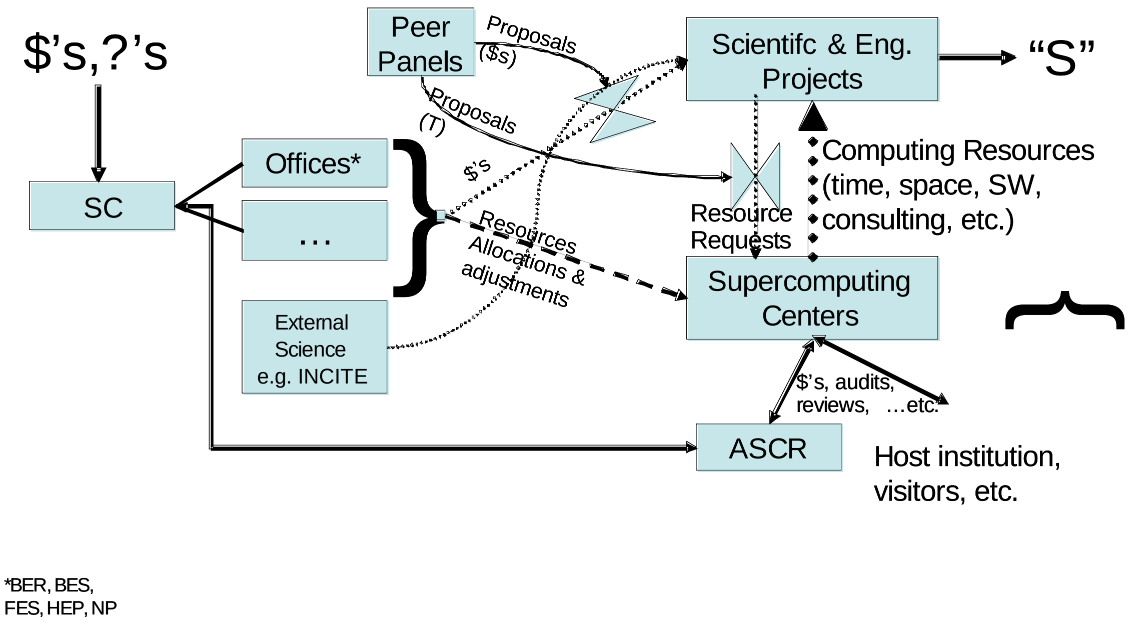
$'s,?'s
SC
Offices*
…
External Science e.g. INCITE
Peer Panels
Proposals ($s)
Proposals (T)
$'s
Resources Allocations & adjustments
Scientifc & Eng. Projects
"S"
Computing Resources (time, space, SW, consulting, etc.)
Resource Requests
Supercomputing Centers
$'s, audits, reviews, …etc.
ASCR
Host institution, visitors, etc.
*BER, BES, FES, HEP, NP

Figure 1 is a simplified diagram showing the flow and control of funds <u>and</u> computing resources from DOE's SC that create science, (S). SC Scientific Offices using funded, peer reviewed science projects and peer reviewed requests for computing resources control the allocation of project funds. Computing resources are provided by DOE's ASCR Centers at ANL, LBNL, and ORNL.

The following sections address the six elements of the charge. However, we viewed the first two elements regarding the Centers and projects as the most significant because these elements are more closely within our purview and areas of expertise.

Element 1. Centers performance measurement and assessment

The Panel believes new metrics are not needed, nor do refinements to existing metrics need to be imposed that potentially or radically alter the application of existing metrics. The Panel reached this conclusion after requesting and receiving metrics from the three centers that are used to measure their effectiveness. These metrics are briefly discussed below and in more detail in the Web Appendix (see Appendix 0 table of contents).

"Capacity" and "Capability" Centers: A utilization challenge

The three centers are presently in different states of maturity. They also have different foci relating to the services they provide, sometimes differentiated by the notion of:

- capacity (e.g. NERSC) – broad use of computational resources including processing, secondary and archival storage, analytics and visualization systems, advanced networking, and user support for 2,500 users working on 300+ projects across multiple architectures, but with minimal funding targeted to support specific projects; and

- capability (e.g. NLCF and ALCF) – focused use of a large amount of computational resources, including Center consultation, to about 20 large scientific projects that utilize one or two platforms at ALCF and NLCF, respectively.

For the foreseeable future, DOE considers the broad allocation of resources based on the capacity-capability segmentation. This allows for a capability center to be more or less intimately engaged with the small number of projects it hosts. Similarly, projects that use capacity facilities may also require significant help from the resource provider to attain the higher degree of parallelism to absorb the capacity from the increase in processor count.

It should be noted that NERSC pioneered the concept of "project specific services" which it continues to provide as part of SciDAC and INCITE projects. Furthermore, with time, the distinction of capacity and capability will be further blurred. A significant share of NERSC is dedicated to INCITE; likewise ORNL is supporting over 800 users in targeted areas under the "end station umbrella.

"Control" versus "Observed" Metrics

Centers utilize dozens of metrics for goal setting, control, review, and management purposes that we divide into "control" metrics, which have specific goals which must be met, and "observed" metrics, which are used for monitoring and assessing activities.

The Panel believes that there should be free and open access, including reporting, of the many "observed" metrics Centers collect and utilize.

> The Panel suggests it would be counter-productive to introduce a large number of spurious "control" metrics beyond the few we recommend below.

The Panel is especially concerned about using "control" metrics that are potentially harmful to the scientific projects that centers serve. For example, establishing a "control" metric for machine utilization too high, typically 70% will ensure longer turn-around times or "expansion factors" for very large jobs and reduce science output. Machine *utilization* should be "observed" (i.e. measured and reported) in order to assess demand and turn-around time -- but it should not be a "control" metric!

Recommended "Control" Metrics for the Centers

The Panel has addressed "control" and reported metrics for Centers. The Panel believes that individual code metrics (math, algorithms, software engineering aspects, code, and experiment management) are equally important measures as we discuss in project metrics of charge element 2.

> The Panel recommends the following to be good "control" metrics, such as those used as OMB's PART (Program Assessment Rating Tool) for the center's performance:

1. User Satisfaction (overall) of provided services obtained via user surveys.

2. Scheduled Availability described below is a “control” metric, with Overall Availability being an “observed” metric.
3. Response time to solve user problems as measured by the centers' trouble reporting systems.
4. Support for high capability work; with *observed and reported* distributions of job sizes.

1.1: User Satisfaction

The Panel suggests that all centers use a “standard” survey based on the NERSC survey that has been used for several years in measuring and improving service.

User feedback is a key to maintaining an effective computational infrastructure, and is important for tracking progress. NERSC conducts annual user surveys that assess the quality and timeliness of support functions using a questionnaire to measure many facets of their services – including properly resolving user problems and providing effective systems and services. An overall satisfaction rating is part of the survey. Interpreting survey results has both a quantitative and qualitative component. For quantitative results, different functions are rated on a numerical scale. Scores above 5.25 on a 7-point scale are considered satisfactory.

An equally important aspect of center operations is how the facility responds to issues identified in the survey and other user feedback. Does the facility use the information to make improvements and are those improvements reflected in improved scores in subsequent years? As a component of measuring user satisfaction each year the centers should quantify that there is an improved user rating in at least half of the areas for which the previous user rating had fallen below 5.25 (out of 7).

1.2: Availability- Systems are available to process a workload.

Meeting the availability metric means the machines are up and available nearly all of the time. Scheduled availability targets should be determined per-machine, based on the capabilities, characteristics, and mission of that machine. Availabilities are of interest both at the initial startup to understand the time to reach a stable operational state and later in the machine lifetime to understand failures.

The Panel recommends that scheduled availability be a control metric, where scheduled availability is the percentage of time a system is available for users, accounting for any scheduled downtime for maintenance and upgrades.

$$\frac{\Sigma \text{ scheduled hours} - \Sigma \text{ outages during scheduled time}}{\Sigma \text{ scheduled hours}}$$

A service interruption is any event or failure (hardware, software, human, and environment) that degrades service below an agreed-upon threshold. With modern scalable computers, the threshold will be system dependent; where the idea is that the failure of just a few nodes in a multi-thousand node machine need not constitute a service interruption. Any shutdown that has less than 24 hours notice is treated as an unscheduled interruption. A service outage is the time from when computational processing halts to the restoration of full operational capability (e.g., not when the system was booted, but rather when user jobs are recovered and restarted).

The centers should be expected to demonstrate that within 12 months of delivery, or a suitable period following a significant upgrade, scheduled availability is >95% or another value agreed to by ASCR.

The Panel recommends that overall availability be an observed metric, where overall availability is the percentage of time a system is available for users, based on the total time of the period.

$$\frac{\Sigma \text{ Total clock hours} - \Sigma \text{ (outages, upgrades, scheduled maintenance, etc.)}}{\Sigma \text{ Total clock hours}}$$

Using overall availability as a control metric may easily become counter productive as it can inhibit beneficial upgrades.

1.3: Response Time for assistance--Facilities provide timely and effective assistance

Helping users effectively use complex systems is a key service that computational facilities must provide. Users should expect that their inquiries are heard and are being addressed. Most importantly, user problems should be addressed in a timely manner.

Many user problems can be solved within a relatively short time period, which is critical to user effectiveness. Some problems take longer to solve – for example if they are referred to a vendor as a serious bug report. The centers should quantify and demonstrate that 80% of user problems are addressed within 3 working days, either by resolving them to the user¹s satisfaction within 3 working days, or for problems that will take longer, by informing the user how the problem will be handled within 3 working days (and providing periodic updates on the expected resolution).

1.4: Leadership Class Facilities (LCF) priority service to capability-limited science applications

The purpose of HPC Leadership Class Facilities is to advance scientific discovery through computer-based modeling, simulation, and data analysis, or what is often called computational science. Scientific discovery can be achieved through pioneering computations that successfully model complex phenomena for the first time, or by extensive exploration of solution space using accepted existing models of scientific phenomena. In either paradigm, computational scientists must be able to obtain sufficiently accurate results within reasonable bounds of time and effort. The degree to which these needs are satisfied reflects the effectiveness of an HPC facility. The effectiveness of HPC facilities is greatly determined by policy decisions that should be driven both by scientific merit and the ability of a computational science application to make effective use of the available resources.

The primary goal of Leadership Class computing facilities is to provide for capability computing, i.e., computational problems that push the limits of modern computers. The Panel believes there is also substantial merit to supporting the exploration of parameter space that can be characterized as capacity computing or an “ensemble” application. The latter class of computational problem can contribute to high overall utilization of the LCF resource, as demonstrated by experience at both the NERSC and NLCF facilities, but often with negative turnaround consequences for capability limited applications. Thus there is a natural tension between optimizing support for capability and capacity computing which will be paced by things like the allocation process.

The Panel recommends that the leadership centers track and ensure that at least T% of all computational time goes to jobs that use more than N CPUs (or equivalently, P% of the available resources), as determined by agreement between the Program Office and the Facility. Furthermore, for jobs defined as capability jobs, the expansion factor (a measure of queue wait time as a fraction of the required execution time) should be no greater than some value X, where $X \leq 4$ may be an appropriate place to start depending on the system's mission and workload. The final target should be determined through an agreement between the Program Office and each Facility and could be as high as 10.

Recommended "Observed" Metrics for the Centers

In addition to the four "control" metrics we recommend, "observed" metrics should be tracked and reported. These are essential for managing computational resources i.e. determining allocations, setting each center's policies, specifying priorities, assessing demand, planning new facilities, etc. Even more important, these "observed" metrics permit a broader comparison, calibration and benchmarking with centers at other agencies (e.g. DOD, NASA, and NSF). Some of the useful metrics to observe include:

- Constituent metrics that make up aggregate "user satisfaction" indices provide insight into user sophistication, level of support by the center, unproductive scientist time, software and hardware reliability, need for additional system software, etc. The NERSC user survey we recommend includes almost 100 useful service aspects.
- System uptime (overall and scheduled), by hardware and software reliability
- Utilization of centers resources. These provide an indicator of delivered computing resources as well as understanding bottlenecks. These are essential measures for understanding the load and utilization of the infrastructure components. This also provides insight into the time required to reach a steady-state operational capability after changes and upgrades.
- Degree of standard and specialized software utilization, including shared research application codes. The DOE centers are likely to evolve, like the IT world, to provide "web services" that can be "called" or accessed to carry out high level remote functions just as users access programs and files locally.
- Degree of use of shared, on line experimental data and databases, such as the Protein Data Bank at NSF's San Diego Center. The DOE centers are likely to evolve, like the IT world, to provide central databases and transaction processing services.
- Individual project metrics that need to be tracked over time include:
    - Total computer resources as requested in Appendix 1: Project Checklist and Metrics
    - job size distributions by runs, amount of time, and processors used;
    - percentage of successful job completion by number and by time

- Individual project program scalability and efficiency
*= speed-on-N-processors/(N*speed-on-one-processor)*
on each platform they utilize is an important *observed metric*. Efficiency is a potentially harmful metric if scientists are required to operate at minimal thresholds of scaling and/or efficiency. It is too early to understand efficiency in light of new chips with multiple microprocessor cores and/or multiple threads, yet have limited bandwidth to memory. Achieving peak and high efficiency will be an even greater challenge than in the past.

  Every scientist will make the trade-off of whether to improve their complex codes or do more science. Nearly all codes run on (utilize) at least two hardware platforms. The machines in the DOE centers differ in computing node characteristics (processor type, speed, processors per node, per node memory), interconnect, and I/O. Portability requirements often imply that every machine is utilized in a sub-optimal fashion! For peak performance, each code that uses significant time needs to be modified to operate efficiently on a specific machine configuration at an optimal scalability-level.
- Project use of software engineering tools for configuration management, program validation and verification, regression testing, workflow management including the ability for “ensemble” experiments that exploit parallelism and allow many “computational experiments per day” to be carried out.

Element 2. Project metrics: complementary, comprehensive, and essential measures

Computational science and engineering utilizing petascale computers offers tremendous promise for a continuing transformational role in the Department of Energy’s Office of Science programs. From Figure 1, the key to this potential is the ability of the project researchers to develop computational applications that can run effectively and efficiently on those computers.

Scaling applications to run on petaflops machines imply a wide range of challenges in all areas from codes to project management.

- Design and engineering of existing and new codes that operate efficiently on 10,000 to 100,000s processors representing the need for an order of magnitude increase in parallelism from many of 2005 codes
- Reacting to evolving and highly variable architectures and configurations for the targeted petaflops computers requiring machine-specific performance expertise
- Dealing with relatively immature, continually evolving research application codes, immature production tools, and environment for parallel program control and development that characterizes state-of-the-art computing
- Evolving small code development teams to large code development teams
- Increasing need for multi-disciplinary and multi-institutional science teams
- Greater need and utilization of software engineering practices and metrics

- Verifying and validating applications for substantially more complex systems against theory and experiments
- Developing problem generation methods for larger and complex problems
- Experiment management, including analyzing and visualizing larger and more complex datasets

Appendix 2 provides the rational behind our project review recommendation. Computational science and engineering encompasses different types of computational applications each of which presents its *petascale* challenge.

Suggestions and Metrics Recommendations

The Panel's belief in a clearly structured approach to reviewing the project's approach to the use of computational resources is based on these observations:

1. Computing resources provided by the centers is highly valuable that requires appropriate review and oversight from a project viewpoint with involvement of the Science programs. Centers have the critical information for such activities. In 2006, an hour of processor time costs a minimum of $1 at the Centers. Projects with little or no DOE funding can receive millions of hours i.e. dollars of computing resources. The range of computing resources versus direct project funding of the "average" project varies from 1:1 for the 2500 NERSC users to 20:1 and higher for projects with minimal SC funding.
2. Large codes can often be improved, which will free up computers for other important scientific applications. The Panel believes the return on this optimization investment will prove to be well worth the effort. This argues for balanced project investment of direct funding and computer resources.
3. Validation and verification is required both to ensure efficacy of the mathematics, algorithms, and code against both theory and experimental results.
4. The management of the code, datasets, and experimental runs used in petascale applications will undoubtedly require significant changes as we describe below.
5. Code quality and readiness as observed by the centers is highly variable. This includes the use of inappropriate or wasteful techniques, abandoned runs, etc.
6. In 2006, the "average" DoD code runs on seven platforms with an implication of non-optimality and a need to restrict and improve such code for a particular use.
7. For the many million dollar-plus projects, appropriate review is almost certain to payoff.

The Panel believes computational applications coming from the funded scientific projects should be reviewed using a checklist with metrics appropriate to the project size and complexity. While the use of metrics and checklists for projects is important for project success, their application must be carefully tailored. Projects vary in size from the use of standard program libraries e.g. Charm, Gaussian, MATLAB, or similar commercial or lab developed software, to small, single individual or team programs with less than 100,000 lines of code, to large coupled codes with over one million lines.

2.1 Project Evaluation.

The Panel's recommended checklist and metrics given in Appendix 1 cover the following seven aspects of projects that the Panel believes have to be well understood and tracked by the Projects, Centers, and Program Offices. They are:

1. Project overview that includes clear goals
2. Project team resources
3. Project resource needs from the center
4. Project "code" including portability, progress on scalability, etc. This is essential for PART measurement.
5. Project software engineering processes
6. Project output and Computational Science Accomplishments as we discuss in Section 3. This provides a comprehensive listing of results that cover publications, people, technology, etc.
7. Project future requirements

A discussion of the motivation for the checklist is given in Appendix 2, including:

1. Measures of scientific and engineering output (i.e. production computing)
2. Verification and Validation
3. Software project risk and management
4. Parallel scaling and parallel performance
5. Portability
6. Software engineering practices

2.2 Code Improvement

The Panel recommends a code improvement metric that is a combined measure of a scientific project's mathematics, algorithms, code, and scalability. Based on the Centers' recommendation, we support a goal of halving the time to solution every three years.

Element 3. The science accomplishments

The Panel believes the Centers play a critical role in enabling scientific discovery through computational techniques in a manner similar to the role played by large experimental facilities (e.g., accelerators).

The Centers not only provide computational hardware and software capability, but also provide support and expertise in ensuring that scientific applications make effective use of the Centers' resources. The panel is confident that the ability of the Centers to excel in their performance as measured by the preceding metrics will advance science accomplishment even further.

The Panel did not have the time, resources or qualifications to assess the science at the breath and depth required to produce a comprehensive and measured picture. This must be done by experts, applying appropriate measures, in each of the offices and programs of scientific domains supporting SC.

The Centers have a lesser role in evaluating scientific accomplishments, but work in concert with application scientists and the various Office of Science Program offices. The basic dilemma is finding a metric to measure scientific accomplishments or progress, which tends to be unpredictable and sporadic. The fruits of scientific discovery often have a long time scale and are therefore unlikely to be useful in short term planning and management. For example, NERSC's 300+ projects from its 2500 users generate 1200-1400 peer reviewed papers annually; these may take a year or more to appear and citations to them will take many years to peak.

Suggestions and Metrics Recommendations

> The Panel suggests that peer reviews of the projects be based on both their scientific and computational science merits for allocation of computational resources along the lines of the INCITE program.
>
> The Panel further suggests that Centers provide an appropriate and significant level of support for the scientific users of their facilities in light of the unique capabilities of the facilities and the leading-edge computational science needs of the user base.

Support could be in the form of user support staff familiar with the idiosyncrasies of their various resources and knowledgeable in the tricks of the trade. Expert staff can apply fine-tuning techniques that can often dramatically increase the efficiency of codes, reduce the number of aborted runs, and reduce the turn around time and the time to completion for a scientific project. The Panel's suggestion is based on the observation that the five year old, SciDAC program has demonstrated how teams of domain scientists and computational scientists focused on specific science problems can accelerate discovery. This also enables the Center's hardware, software and programming expertise to be brought to bear on selected scientific challenges.

The Panel recommends the following metrics be "reported and tracked" in a fashion similar to the centers "control" metrics to assist in the measurement of scientific output from its projects:

3.1 Publications, Code & Datasets, People, and Technology Transfer

> The Appendix 1, Item 6 project checklist goes beyond the traditional scientific measures. Publications, including citations and awards are important indications of whether the research is having an impact, but are not the complete picture. Equally important measures of output include professionals trained in computational science. With computing, application codes and datasets that others use are comparably important measures of *computational* scientific output and should be identified as such. In addition, technology transfer, including the formation of new companies for use in the private sector is important to the industrial and scientific community for advancing science.

3.2 Project Milestones Accomplished versus Proposal Project Plan

is a near term and useful metric as to whether a project is on the path towards meeting well defined scientific goals. These goals have *presumably* been peer reviewed by the scientific community, and certified as having a legitimate scientific purpose. Thus, the steps leading to these goals should be measurable. The Centers have suggested measuring how computation enables scientific progress by tracking computational result milestones identified in their project proposals. The value of the metric is based on an assessment made by the related science program office or peer review panel regarding how well scientific milestones were met or exceeded relative to plans for the review period.

3.3 Exploiting Parallelism and/or Improved Efficiency aka Code Improvement

How well scientific applications take advantage of a given computational resource is a key to progress through computation. Improved algorithms that increase code efficiency are critically important to the improvement of code effectiveness. Future processor technology for petascale computing and beyond is forecast to be multicore chips with no significant increase in clock rate. Therefore an increased computational rate for any given application can only be achieved by exploiting increased parallelism. The metric is to increase application computing performance by increasing scalability, where scalability is the ability to achieve near linear speed-up with increased core count. Scalability of selected applications should contribute to doubling the rate of solution every three years as described in the previous section as Code Improvement metric 2.2.

3.3 Break-throughs; an immeasurable goal

The Panel could not identify metrics or any method that could be used to anticipate discoveries that occur on the leading edge of fundamental science. The scientific communities can more effectively recognize breakthrough science, or even what constitutes a "significant advance." Unfortunately, we cannot identify a metric that tracks scientific progress that is guided by computation, - especially at the high end of the “Branscomb Pyramid”[3].

In order to take this kind of event into account, we suggest measuring scientific progress by some process that would enumerate “breakthroughs” or “significant advances” in computational science on an annual basis. The Panel observed what we believe are such breakthroughs taken from presentations at the June 2006 SciDAC meeting. The following would not have been possible without high performance computers.

[3] NSB Report: "From Desktop to Teraflop: Exploiting the U.S. Lead in High Performance Computing," chaired by Dr. Lewis Branscomb, October, 1993

(1) solving a problem that had not been solved before -- new method of solving the protein folding problem using Monte Carlo techniques by Charles Strauss, LLNL;

(2) increasing code efficiency by several orders of magnitude --combustion calculation code by John Bell, LLBL;

(3) greatly reducing the disagreement between theory and experiment --the QCD calculations which validate the standard model of particle physics, by Christine Davies of Glasgow U.;

(4) greatly expanding the scope and scale of computational simulation to provide accurate or new results -- Fred Streitz, LLNL, showed that at least 8 million atoms are needed in a simulation in order to get consistent results for metal condensation. (Streitz won the 2005 Gordon Bell Prize competition using an IBM BlueGene/L with 131 thousand processors operating at 107 Teraflops.)

Element 4. *The Computational Resources* Effects on the Office of Science's science programs.

While it is clear from past successes that the Centers are key to enabling advanced scientific discovery, the Panel did not feel it had the resources or expertise needed for a definitive assessment of the Center's effects on scientific programs. We suggest that this evaluation needs to be done by the Program offices and their advisors, as they assess the Centers' projected computational capabilities impact on the path to science discovery in their respective disciplines. A review by each Program office also provides a cross-check on Centers' effectiveness. The Panel provides some comments on the program effects and possible metrics.

SciDAC is now five years old, and has had an impact on the Office of Science programs. The results could provide metrics for evaluating the effect of computation on scientific progress. In the recently concluded second round of SciDAC awards the focus was preparing for the use of facilities which have a target of a Petaflops peak by 2010. This is a reflection of the fact that a major challenge facing computational science during the next five to ten years is the increased parallelism needed to realize the full potential of future computational resources.

Suggestions and Metrics Recommendations

The Panel believes the available computational capability paces the rate of scientific discovery and is a major factor in determining the scientific questions that can be addressed in selected scientific domains. Therefore, the level of computational capability made available by the Centers has been and will continue to be on the critical path to achieving discovery goals in several disciplines of the Office of Science.

The Panel believes that management processes and metrics are needed to demonstrate the significance of the Centers' capabilities to advance the science programs, especially in selected domains.

The Panel believes that measures and management are required to help understand whether changes in the levels of computational resources and support would change the output of science. For example, poor code has a negative effect on scientific output and the effective use of computational resources

The Panel was unable to ascertain whether the Office of Science has a good mechanism to evaluate the role of computation in the advancement of science. Evaluation might be done through the advisory committees for Basic Energy Science, Biological and Environmental Research, Fusion Energy, High Energy Physics and Nuclear Physics as they assess the Centers' projected computational capabilities impact on the path to science discovery in their respective disciplines.

> The Panel suggests that a clear process be implemented that measures the use and effects of the computational resources on the projects within each SC office.

The Centers also need feedback from these committees to ensure that the computational facilities are contributing in an optimal fashion to advancement of science in the individual disciplines.

> The Panel suggests that each SC office report the total investment in all projects, using a rough conversion of computer time to dollars. The Panel believes that computational resources need to be treated in a substantially more serious and measured fashion by the program offices and project personnel.

The computational resource investment often dwarfs the direct project funding. To the extent possible we plotted the size of the project grant versus the amount of computer time. Using computer time at NERSC at $1/hour, we observed that for projects using several million dollars of computer time, the SC project funding ranged from zero, one-half, two, and four times as much as the value of the computer time.

> The Panel suggests the process of allocating computer time at the Centers through the program offices be re examined in light of the diversity of architectures. Given the variety of platforms at the Centers and need for user code portability, the efficiency of a particular code will be highly variable.

An overall allocation procedure such as is used by the NSF Centers might be considered, in which scientific projects are mapped on to the computational resources appropriate to their needs. Unfortunately, the capacity or "broad focus" characterization (NERSC) versus capability or "limited focus" characterization (ALCF, NLCF) segmentation inhibits the use of unique or critical resources e.g. the shared memory, vector Cray X-1 at NLCF that could have a dramatic impact on specific codes!

Element 5. Evolution of the facilities and roles

It is useful to characterize the Centers by their experience, scientific domain or mission, and charter characterized simplistically by the notion of capacity versus capability computing. The DOE Centers support science very broadly as opposed to being directed at any particular mission including surrounding datasets that is best typified by the National Center for Atmospheric Research (NCAR) at Boulder.

NERSC has a long history of supporting a large and diverse user base including being a large storage facility. It currently supports over 2500 users on almost 400 projects. Its role has always been to host one of the largest DOE computer systems and to make it available to a broad community of users and projects. Some of these projects are very large and many are significantly smaller than those planned for NLCF and ALCF. There does not appear to be a significant change planned for the NERSC role as both a broad supplier and platform for large project evolution; a role the Panel believes is essential to the DOE Office of Science mission as many important projects do not require 1/20th of a leadership computer. However, both roles imply a need for application support.

The ORNL center (NLCF) has been operational as a service center for only one year; the ANL center (ALCF) is just starting up. Both centers are based on the assumption of about twenty large projects, each capable of utilizing a large fraction of the machine. A better characterization of these centers as being "limited focus" versus "broad focus" may be preferred to their characterization as "capability" or "capacity" centers.

The panel makes the following suggestions regarding the evolution of the facilities roles:

> The Panel believes the current computational resources plan is reasonable, but it is premature to consider their evolution in detail. However, the Panel does want to comment on some long-term questions that SC may wish to consider as the facilities evolve. Note that the Panel did not do a system by system review of the entire facility including storage, visualization, software tools, etc. as we felt the Lehman Committee reviews had covered these aspects.
>
> NERSC is a facility that has evolved to its present, highly tuned state over many years. In contrast NLCF and ALCF are young and their steady-state operation is not yet clear – including finding a balance for capacity. In 2006 they are focused on providing leading-edge systems representing the most advanced high-performance computing technology available by rapidly inserting new technology. A system can for a time remain on the leading-edge by upgrading its technology, but at some point a new system will be required. When this happens the older system may only be a few years old, in stable operation and providing reliable, capable and cost efficient computing. It may make sense for the older system to remain at the facility to serve a broader application base rather than be sited at some other facility. In other words, are the LCF's likely to evolve into a state of expanded service where the current leading-edge system provides capability computing while the previous one provides capacity computing service?
>
> Finally, NLCF and ALCF have taken significantly different architectural approaches in their leading-edge systems. Application codes from some scientific domains adapt better to one of these systems than the other, e.g., material science and the IBM Blue Gene, given the initial limitation of per node memory. If this distinction continues, should the facilities evolve to serve only a restricted set of domains, similar to NCAR?

Regardless of their long-term evolution the panel *suggests* that project-integrated consulting should constitute a portion of the budget for all the centers in future funding scenarios. Petascale computers are going to be difficult to use efficiently in most applications because of the need to increase parallelism in order to get same fraction of use. As the SciDAC initiative demonstrated, the scientists need the help of computing professionals to make good use of the resources.

Element 6. Computational needs over the next 3-5 years

The Panel believes the centers are on a sound trajectory to supply the broad range of needs of the scientific community, which will allow SC programs to maintain their national and international scientific leadership. This is based on the current facilities' plans and the scientific needs, including the recent charter expansion of INCITE to supply computational resources to the broader scientific community. Furthermore, while the science community has responded to increases in processor count in the past, it has taken time. For example, the NLCF processor count has just doubled, and will soon redouble. Experience with this increase in parallelism using multiple cores will provide valuable data about the multiple threads and multiple cores scaling approach.

Having a common, Cray architecture at NLCF and NERSC will serve the several thousand users who require the capacity of NERSC as well as the score of high capability users at NLCF, including the migration of the workload between the centers. The ALCF facility based on the IBM BlueGene system represents a different and potentially promising petascale approach utilizing a larger number of slower, but less power-consuming processors, connected by very fast networks.

The evolution of the computational needs of SC programs over the next 3 to 5 years is well-documented in numerous reports: ASCR Strategic Plan, SCaLeS report, SciDAC 5-year report, and NRC's "The Future of Supercomputing" which was sponsored by DOE. The push to petaflops computing is driven by the need for higher fidelity, faster throughput, improved resolution, and integrated modeling involving multi-scale and multi-physics simulations. Key examples include plasma fusion, where the simulation of a magnetically-confined plasma on the scale of the proposed ITER facility will require one trillion particles, and the next-generation climate model that will integrate biogeochemistry, atmospheric chemistry and a global carbon cycle. The Facilities are keenly aware of these needs through their strategic planning processes and user input, including the DOE Greenbook compiled every three years by the NERSC User Group and the Application Requirements Council at the NLCF.

The Panel believes it is too early to assess the impact of the expanded role of INCITE on the facilities' demand. Based on just the recent three decades of scientific computers whose performance has doubled annually, there is no end to the imaginable applications or amount of computing that science can absorb – especially at zero apparent cost. In other words, projections by scientists coming from the plethora of committees, task forces, and workshops to utilize any and every imaginable future computing resource is almost certain to outrun any and every realizable computational infrastructure. Science continues to demonstrate a nearly infinite demand for computing. Already researchers are discussing the needs and ability for exa-scale computers.

Observations on the management of ASCR's portfolio

ASCR's mission, to deliver forefront computational and networking capabilities to scientists nationwide that enable them to extend the frontiers of science, involves several strategic areas. Principal among these is the deployment and appropriate management of computational Centers that provide computational capabilities and computer science expertise for leadership in scientific computation. This leadership is directed toward the Department's strategy to ensure the security of the nation and to succeed in its science, energy, environmental quality, and national security missions. While the Panel believes the appropriate strategies for moving to the petascale in the computational arena are in place, and measures that will help define success are suggested above, we believe an integration of the Program (or Mission) needs with the scientific computing capabilities should be enhanced.

The Panel suggests two areas for possible improvement:

1 increasing communications between ASCR and Scientific Program Offices, and

2 improving the capability and support of various scientific codes and teams to use the petascale architectures by both general and domain specific Centers support

These involve strengthening and formalizing the discussions among program offices concerning which scientific codes are mission relevant – in essence priority setting as ASCR broadens its charter with INCITE. ASCR is not charged to direct the domain sciences, and neither are the Centers. Assessing the scientific relevance of computational projects and recommending them for computational cycles should reside with the Program Offices; however, a key component for successful computational activities will be an integration of science with computation. We are concerned that this integration may not be happening except in some circumstances (for example, in those circumstances where a SciDAC award has been made).

The Panel believes a tighter integration should take place between ASCR and SC's programs and that SC report a total budget that includes computation as described above. Specifically, decisions concerning computational readiness and the computational needs of a given science project should include ASCR expertise.

Concluding statement

The Panel believes we have provided useful, actionable suggestions and recommendations. These are based on our own and our colleagues' experience, and our recent review of the Centers and projects. We hope the Department will find these recommendations useful as it addresses the petascale challenge.

Acknowledgements

The Panel gratefully acknowledges the “Centers” cheerful, competent, and professional assistance in supplying information to “another headquarters panel” about their operation and projects, especially Ray Blair (ANL), Al Geist (ORNL), Douglas Kothe (ORNL), Bill Kramer (NERSC), and Francesca Verdier (NERSC). Douglass Post and Brad Comes of DOD’s HPC Modernization Program supplied valuable “benchmark” information about centers and project information that set a high mark for centers. Allan Snavely, Fran Berman and others from SDSC, Mike Levine PSC, and José L. Munoz (NSF OCI) provided glimpses of the NSF centers and their operation. Christine Chalk (DOE) provided data and plots for much insight e.g. computer time versus project funding. Nancy White (DOE) and Vicki Rozycki (Microsoft) provided logistic support and sustenance.

Appendix 0. Web Appendix: Petascale Computing Metrics Report Contents and Background Presentations

1. Orbach charge & attachment 3/10/06

The initial charge that created the Petascale Metrics Panel

2. Centers Computer Inventory at DOE, NSF, and DOD 7/19/06, rev. 8/20/06

Listing of supercomputers at DOE, NSF, and DOE with their processor counts, peak processing rate, processor speed, memory size, and capacity in TF-hours/year

3. Centers Proposal for Centers and Project Metrics 5/31/06, rev. 7/18/06

Bill Kramer (NERSC) committee of centers directors proposed metrics for measuring centers that the committee supported and adopted.

4. Centers Request from Panel 4/25/06; Responses metrics 5/31/06 ANL, ORNL, NERSC and Centers presentations 7/18/06 ANL, ORNL, NERSC

The panel request for information about the operation at the three centers and the per center response. The centers presented revised metrics at the meeting 7/18/06

5. DOD Centers Metrics presentation -Brad Comes 7/18/06 25

Brad Comes, who heads the DOD Centers Acquisition, provided an overview of his operation; and the metrics DOD for Centers management

6. Project Request to Centers 4/25/06

Spreadsheet responses from ORNL and NERSC 5/3/06. The NERSC .xls consists of several worksheets about computer use for various facilities and years. It has distributions of the parallelism for individual users.

7. Comp. Sci. and Eng. Software Development Issues-D. Post, DoD HPCMP

Projects require software development methodologies as the size and complexity increases.

8. DOD Engineering Projects Project Checklist & Metrics 7/18/06 – D. Post

DOD Project Survey of code, software engineering techniques, etc.

9. Project Presentation Template

Four Quadrant View of a Project including basic information and goals; resources being supplied by centers, code and project management, and scientific outputs

10. Ten Project Showcase by Centers, 7/17/06: from ANL, ORNL, NERSC

Ten projects used the project presentation template for description

Appendix 1. Computational Project Checklist and Metrics

The appendix provides a possible checklist that can be used to evaluate and track computational projects that utilize centers resources.

1. **Project overview**
2. **Project team resources**
3. **Project center resource inputs**
4. **Project "code"**
5. **Project software engineering processes**
6. **Science and Project outputs**
7. **Project Future Requirements**

1.0 Project Overview

a. Project name
b. Contact information for the project
    i. Principal investigators, emails; phones
    ii. URL
c. DOE Office support (SC Office (BES, BER, NP, HEP, ASCR, FES, other); DOE program manager;
d. Scientific domain (chemistry, fusion, high energy, nuclear, other.),
e. What are the technical goals of the project?
    i. What problem or "grand challenge" are you trying to solve?
    ii. What is the expect impact of project success? (e.g. better understanding of supernovae explosions, prediction of ITER performance, …)
f. Support for the development of the code
    i. Degree of DOE support to develop the code?
    ii. SciDAC, DOE SC, INCITE program
    iii. Internal institutional funding sources (e.g. LDRD,..),
    iv. Industry,
    v. Other agencies, ….
g. Size and names of the external communities that your code or datasets support.

2. Project Team Resources

a. Team size & structure
b. Team institutional affiliation(s). (e.g. all the institutions involved, including universities, national labs, government agencies,..). I.e. to what extent is the team multi-institutional?
c. To what extent are the code team members affiliated with the computer center institution? (e.g. are the team members also members of the computer center facility?)
d. Team composition and experience total
    i. domain scientists,
    ii. computational scientists, computer scientists, computational mathematicians, database managers

        iii. programmers
        iv. other
    e. Team composition by educational level (total)
        i. Ph.D.,
        ii. MS, BS, undergraduate students, graduate students, post-docs, younger faculty, senior faculty, national laboratory scientists, industrial scientists, etc.)
    f. Team resources utilization: time spent on code and algorithm development, maintenance, problem setup, production, and results analysis

3.0 Project use of Center's resources

a. Steady state use of resources on a production basis per month
    i. Number of processors and wall-clock time of typical jobs
    ii. Memory requirements, per processor and total job
    iii. Total Processor time for the month
    iv. On-line disk: temporary & permanent
    v. Tertiary storage rate of change
b. Annual use of resources
    i. Processor time
    ii. Disk
    iii. Tertiary storage rate of change
c. Software provided by center
d. Consulting including centers staff that are team members
e. Direct project support as a team member
f. THIS IS ITEM 3.0 a NEXT SECTION ITEM 3.0 a
g. How will usage change over next 3 yrs?
    i. Number of processors
    ii. Disk store
    iii. I/O
    iv. Transactions
    v. Tertiary store

4.0 Project Code

a. Problem Type (data analysis, data mining, simulation, experimental design, etc.)
b. Types of algorithms and computational mathematics (e.g. finite element, finite volume, Monte-Carlo, Krylov methods, adaptive mesh refinement, etc.)
c. What platforms does your code run on including your own local cluster?
    i. What is your preferred platform?
d. Approximate size of code in single lines of code (sloc)
    i. Total
    ii. Annual growth
    iii. Fortran 77
    iv. Fortran 90 or 95
    v. C
    vi. C++
    vii. Python
    viii. Java

xi. PERL
x. Other (List)

e. History and dates:
   i. Development started (month/year)
   ii. First usable version(month/year)
   iii. First significant applications(month/year)
   iv. Reasonably mature(month/year)
   v. Expected retirement(month/year)
f. What libraries are used?
   i. What fraction of the effort do they represent?
g. Code Mix:
   i. To what extent does your team develop and use your own codes?
   ii. Codes developed by others in the DOE and general scientific community?
   iii. Application codes provided by the center?
h. What memory/processor ratio do your project require? (e.g. Gbytes/processor)
i. What is the single-processor efficiency of the code
   i. How is it measured?
j. Parallelization model (e.g. MPI, OpenMP, Threads, UPC, Co-Array Fortran, etc.) E.g. Does your team use domain decomposition and if so what tools do you use?
k. <u>What is the scalability i.e. knee of the curve</u>
   i. <u>What number of processors can you efficiently use now?</u>
   ii. <u>Projected or maximum scalability</u>
   i. <u>How is measured?</u>
l. What are the major bottlenecks for scaling your code?
   i. Algorithms for parallelization
   ii. Inter-node bandwidth or latency
   iii. Memory/node
   iv. Processors/node
   v. I/O to/from disk or to/from tertiary store
m. What is the split between interactive and batch use?
   i. Why, and is interactive more productive
n. What is the split between code development on the computer center computers and on computers at other institutions including your own cluster?

5.0 Software Engineering, Development, Verification and Validation Processes

a. Software development tools used (
   i. parallel development,
   ii. debuggers,
   iii. visualization,
   iv. production management and steering
b. Software engineering practices. Please list the specific tools or processes used for
   i. configuration management,
   ii. quality control,
   iii. bug reporting and tracking,
   iv. code reviews,
   v. project planning,

    vi. project scheduling and tracking
- c. What is your verification strategy?
- d. What use do you make of regression tests?
- e. What is your validation strategy?
- f. What experimental facilities do you use for validation?
- g. Does your project have adequate resources for validation?

6.0 Science and project output metrics during the last 3 years

<u>Enumerate project output from the last 3 years:</u>
<u>Enumerate the effect on the Office of Science Programs:</u>
<u>In addition</u> provide:

- a. # Publications?
- b. Citations?
- c. Dissertations?
- d. Prizes and other honors?
- e. Residual and supported, datasets and/or databases accessed by a community?
    - i. Describe size of the external user community for the datasets
- f. Change in code capabilities and quality
- g. Code contributed to the centers
- h. Code contributed to the scientific community at large
- i. Company spin-offs based on code or trained people and/or CRADAs
- j. Corporation, extra-agency, etc. and any other external use
- k. Scientist output: Increase in trained scientists during last 5 years (list)
- l. Program Developers: Increase in trained code developers capable of writing project-level codes during last 5 years (list)

7.0 Project Future Requirements (qualitative)

- a. What is the greatest impediment in the use of the center's facilities?
- b. What do you believe the proposed increases in capacity at the facilities will provide your project (e.g. based on your observations of historical increases)?
    - i. Better turn-around time for the project
    - ii. More users and incremental improvement in use with little or no change in scale or quality of results at the project level
    - iii. Reduced granularity, resulting in constant solution time, though more accurate results or increased insight
    - iv. New applications permitting in new approaches and new science
- c. How, specifically, has your use changed with specific facilities increases?
- d. How is the project & effort projected to change in response to petascale computing?
- e. What is your plan for utilizing increased resources?

Appendix 2. Project Issues and Metrics

Computational science and engineering utilizing petaflops computers offers tremendous promise for playing a transformational role in the success of the Department of Energy Office of Science programs. The key to realizing this potential will be the successful development of the many different types of computational applications (Table 2.1) that can run effectively and efficiently on the DOE SC planned petaflops computers as well as the development of those computing systems.

Table 2.1 Taxonomy of Computational Science and Engineering Application Projects

- **Scientific discovery—**study of new scientific phenomena such as calculating the trade-off many different effects to determine the most important mechanisms; or calculation of the non-linear behavior of a complex system such as the generation of a high-resolution first principles turbulence simulation dataset
- **Experimental analysis and design—**the analysis of experimental data from DOE research facilities; or the design of a new high energy particle detectors
- **Prediction of operational conditions—**path of a hurricane, evolution of space weather, path of a satellite, exploration of potential operating modes for a tokamak reactor experiment, …
- **Scientific design and analysis—**analysis of large datasets (e.g. screening of all known microbial drug targets against the known chemical compound libraries, design of materials with specific properties), analysis of large datasets of turbulence simulations,
- **Engineering design and analysis—** Design of a passively safe reactor core for the Advanced Burner Reactor, tokamak reactors, high energy accelerators,…

The Panel and the DOE SC computer centers surveyed the DOE SC and other computational science and engineering communities to characterize the state of development of these applications. These surveys and case studies identified many of the challenges that the application development teams will need to address (Table 2.2).

Table 2.2 Petaflops application scaling challenges

- Design and engineering of existing and new codes that operate efficiently on 10,000 to 100,000s processors representing the need for an order of magnitude increase in parallelism
- Somewhat undefined and highly variable architectures and configurations for the targeted petaflop computers requiring machine-specific performance expertise
- Relatively immature code development, production tools, and environment for parallel program control and development
- Calculation the trade-off of many different strongly interacting effects across many more orders of magnitude of multiple time and distance scales

- Achievement of adequate levels of code performance and efficiency
- Evolution of small code development teams to large code development teams
- Increased use of multi-disciplinary and multi-institutional teams
- Greater utilization of software engineering practices and metrics
- Verification and validation of applications for substantially more complex systems against theory and experiments
- Development of problem generation methods for larger and complex problems
- Analysis and visualization of larger and more complex datasets

The general characteristics and metrics for existing DoD Teraflops applications (Table 3.3) help define the scale of the challenge. Development of codes with either a lot of users (e.g. commercial scientific codes) or that calculate many multiple effects (e.g. weather, climate, nuclear explosions, chemistry,..) requires relatively large teams. Smaller development teams are required for codes with fewer effects or few users. The successful large teams had team members from many disciplines, i.e. team members who were domain scientists, scientific programmers, software engineers, project managers, etc. Almost all of the teams were led by domain scientists with strong computational science and leadership skills as well as domain science expertise. The larger code teams generally found it useful to adopt greater degrees of software project management and software engineering. Most computational science and engineering codes are fairly large (100s of thousands of lines of code) and took 10 years or more to develop. FORTRAN is the dominant language, but the number of the newest codes had significant portions of C and C++. Almost all codes utilize a number of languages, including several scripting languages (Python, PERL, etc.). C and FORTRAN are fairly interchangeable and pose similar challenges. Object oriented languages (e.g. C++) are also slowly gaining acceptance. The successful C++ codes generally use only a few levels of inheritance or templating. Otherwise memory latency and intercommunication kills performance. In addition, the challenge of writing clear, understandable C++ code is much greater and the learning curve is much steeper for C++ compared to C or FORTRAN. MPI is the dominant parallelization model by far. The average age of these projects is between 15 and 20 years. Almost all of the codes have been under continual development for their whole life. They started being used to deliver results within a few years of the start of the project, and have been productive from that point forward. Codes that didn't deliver some useful capability within a few years of project start were usually unsuccessful. Codes that cease being developed usually cease being used and die within a very few years.

For the DoD survey, the "average" code runs on 7 platforms so that the ability to port to different platforms is a high priority. This usually results in sub-optimal utilization of any particular system. The age of the codes is much longer than the life time of computer platforms (3 to 6 years), so that performance optimization above what is necessary to achieve adequate performance is a lower priority than portability. Many of the codes have been able to scale to ~ 1000 to 3000 processors, although some exhibit poor scaling above 10 to 100 processors. Interconnect latency is one of the main reasons for poor parallel scaling. Most DOD applications typically use between 128 and 292 processors for a typical job as measured by job count, even if they have demonstrated that they can run with higher levels of processor counts. The bulk of computer times is used for larger jobs. Codes that scale well will typically use almost all the processors the scheduling system will let them use, especially those that are close to “pleasingly” parallel. The typical memory per processor varies form 0.75 to 4 GBytes.

Table 3.3 Characteristics of c2006 Teraflops computational science and engineering applications taken from a DoD application survey of top 40 codes

Metric	**Mean**	**Median**
Team size (FTEs)	38	6
Number of users	5000	27
Code size (Single lines of code)	820K	257K
Dominant Language		
Fortran	58%	
C	17%	
C++	13%	
Other	12%	
Parallelization model	Almost 100% MPI	
Project age (years)	20	17.5
Production version age (years)	15	15
Number of platforms	7	7
Largest degree of parallelization	1000 to 3000	1000 to 3000
Typical minimum of processors	225	128
Typical maximum of processors	292	128
Typical memory per processor	0.75 to 4 GBytes	

An analysis of these and other projects indicates that project success is enhanced by attention to verification and validation, software project management and software engineering, and risk minimization[4]. Almost all of the projects use some level of automated version control like CVS. Regression testing is not as common. Almost none of the projects have formal validation programs, or dedicated experimental support for validation. Validation is mostly done by comparing with the results of published data that were often taken long before the code was written and is not connected to the code project. Success for the code projects was measured by published results, customer satisfaction when there were external customers, invited papers, citations, and

[4] *Lessons Learned From ASCI*, D. E. Post and R. P. Kendall, The International Journal of High Performance Computing Applications, **18**(2004), pp. 399-416.

professional society and sponsoring organization prizes and awards, as well as grant or contract renewal.

This context and "lessons learned" identify the major ingredients of successful large-scale computational science and engineering projects, particularly the ingredients that will be important for success of the DOE SC petaflops program applications. The DOE SC computational science and engineering program will not succeed unless the applications are successful in producing significant scientific results. Each project represents a significant investment by DOE SC. Even small code development teams will consume significant resources. Including the cost of the computer and computer center support, a six-member project using 1/20th of a petaflops computer will cost up to $30M over a 5 year period on the leadership class facility at ORNL. It is thus essential that the application projects be well supported and well managed.

We identified six key measures and checklist items that can be tracked through peer review and DOE oversight:

1. Continual scientific and engineering output
2. Verification and Validation
3. Software project risk and management
4. Parallel scaling and parallel performance
5. Portability
6. Software engineering

It is essential that a balanced and graded approach be employed when applying these measures and checklists. Small projects work well with relatively few formal processes and would be crippled if forced to follow all the procedures necessary for much larger projects. However, even smaller projects need to organize their work and follow basic software engineering principles such as configuration management, testing, etc.

1. Continual scientific and engineering output

Successful code projects need to be continually applied to the solution of important and challenging problems. This provides a continuous set of reality checks for the application and the application development team. It ensures that the project tracks changing and evolving requirements (i.e. tracks the evolution of the emerging scientific progress in the scientific domain), and that the team members continue to be motivated and productive scientists. Measures for this include the normal ones for scientific output, e.g. publications, invited papers, patents, significant discoveries, design accomplishments, citations, etc.

2. Verification and Validation

Without verification and validation, there is little or no assurance that the code is free of important errors and defects and includes accurate treatments of all the important effects. Indeed, without validation and verification there is no assurance that computational results have any validity at all. Measures for verification include the frequency of regression tests, the fraction of the code tested, the number and types of verification tests (symmetry, predictable behaviors, truncation error convergence with grid size, comparison with analytic test problems, benchmarks with similar codes, etc.). Measures for validation include detailed numerical and statistical comparison of code results with experimental data for conditions as close to the problems of interest as possible. Every code project should have a validation plan. When possible, it should include collaboration with relevant experimental groups. Surveys and the case studies and there is a paucity of experimental data for the relevant regimes. The best validation data is ideally obtained from experiments designed specifically for validation, especially experiments conducted after the computational result has been obtained.

3. Software project risk and management

The history of the development of large-scale scientific codes, just as for the development of industrial software, indicates that code development has many risks. As many as one-half (or more) of large-scale scientific code projects fail to achieve their initial goals. A significant portion of those never produced significant results and were abandoned without ever achieving significant results. Like all complicated endeavors involving teams, it is important to organize the collective efforts of individuals to achieve a successful outcome. The code development tasks must be planned and organized. Progress needs to be tracked and periodically reported to management. The level of organization and planning depends on the size of the code team and scale of the project. A graded approach for the level and formality of software project management is essential. Teams with only a few individuals at a single institution require relatively little planning and organization. Success for teams with many individuals from several institutions developing a complex, multi-effect code will require significant levels of planning and organization. The team leaders will need to monitor progress and adjust the project schedule and task plans accordingly. The most successful projects placed a strong emphasis on identifying, minimizing and mitigating project risks. Appropriate measures include successful reviews by internal and external monitors, completion of milestones, successful delivery of code capability, continual scientific progress reflected by the usual measures of scientific results (published papers, invited papers, citations …). While software project management is important, it is essential to realize that scientific code development is a research activity, and requires agile processes that provide a proper balance between an organized development process and a flexible development process that can change based on the technical progress made during the project.

4. Parallel scaling and parallel performance

Most of the increased performance from the teraflops range of present computers to the petaflops range will be obtained through parallelization. Codes that can take advantage of petaflops computers will need to be able to incorporate algorithms that scale well from hundreds or thousands of processors to tens of thousand or hundreds of thousands of processors. Since this will be accomplished in stages for most codes, it will be necessary for the codes to exhibit continual progress in scaling. In addition, the code development teams will have to emphasize identifying and exploiting algorithms that have improved parallel scaling. The DOE SC should aggressively promote and support the development of such algorithms as well. Given the investment in computing the DOE is making, efficient use of the DOE petaflops computing facilities should be strongly emphasized in allocating computer time. Appropriate measures include continual demonstration of good parallel scaling and achievement of a reasonable fraction of peak performance.

5. Portability

DOE SC will be fielding at least 3 major computer facilities in the 500 to 1000 Teraflops range. Many, if not most, of the computer applications to be able to run on at least two, and possibly even all three of these platforms. Most of the applications will also run on other platforms as well. Much of the code development and problem setup and testing will be carried out on smaller scale platforms. Code portability is thus absolutely essential. Appropriate measures include demonstration of reasonable levels of performance on key platforms, including both large scale and smaller scale platforms.

6. Software engineering

The DOE SC petaflops applications represent substantial investments by the Department of Energy (as much as $30M/project or more over a 5 year period). Attention to efficient and effective code development procedures can improve the likelihood and level of the scientific success of the code applications. Fewer defects and early detection of those defects will improve the accuracy of the scientific results. Since most of the code development will be accomplished by multi-institutional teams, procedures to facilitate coordinated code development will also need to be emphasized.

Effective software engineering practices include utilization of the best software development tools (including tools for configuration management, defect tracking, parallel profiling and optimization, static analysis, etc.), use of effective development processes such as software architecture design, code review, definition of common interface specifications and uniform code styles to facilitate module development and integration, use of collaboration tools to facilitate development by multi-institutional teams, use of problem setup tools, remote and local visualization and data analysis tools, efficient and effective archiving of datasets of results, and sharing of datasets with other groups when appropriate,

Measures include review of the appropriate level for the use of these procedures by each team, the degree to which the teams use them and demonstrations of their effectiveness.