Valuation of Ultra-Scale Computing Systems

Larry Rudolph[1] and Paul Smith[2]

[1] MIT Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02139
rudolph@lcs.mit.edu

[2] MRJ Tecnology Solutions
10560 Arrowhead Drive
Fairfax, Virginia
phsmith@mrj.com

Abstract. The goal of the Ultra-Scale Computing Valuation Project is to understand utilization issues for both users and managers of the largest scientific computing systems and to begin developing appropriate metrics and models for such system. This paper describes a few aspects of the project.

1 Introduction

Ultra-scale computers are general-purpose computers in actual use, whose computing power (the combination of aggregate processor speed, memory size, and I/O speeds) is about an order of magnitude larger than the highest performance machine available. These ultra-scale computers, normally outside the area of focus of the commercial market forces, enable much larger and more complex computations than can be performed presently on more conventionally available platforms. However, one is never fully outside of commercial market forces nor can one completely ignore practical realities; it is important to measure the value of ultra-scale computers.

To this end, the Ultra-Scale Computing Valuation Project was undertaken to see how to evaluate the efficient use of these computers (the full version of the report is available at `www.dp.doe.gov/valuation`). The bottom line is that the value provided by an ultra-scale machine can only truly be measured, in the long run, by the scientific output produced by using it. But, in the short run there is much that can be done to improve the utilization of these expensive resources.

The valuation project included experts from universities, the federal government, national laboratories, and the computing industry. Several meetings were held, operational data were analyzed, and many discussions took place to arrive at the conclusions and recommendations. Participants were able to identify several operational similarities at the Ultra-scale computing centers while recognizing that there are very few general practices for measuring use and assessing value that will hold across all sites. What is needed, therefore, is a sufficiently flexible and graded approach that can be used by each site to measure the contributions of advanced computing systems to scientific thinking and meeting programmatic objectives. Such an approach recognizes that the first-of-their-kind status of ultra-scale platforms directly impacts initial utilization.

What cannot be measured cannot be managed, and what cannot be managed cannot be improved. Assessing the overall value of a highly sophisticated resource dedicated to pushing the forefront of knowledge requires complicated analysis and calculation. Ultimately, the value of the ultra-scale computing platform must be defined and measured in terms of usefulness to the user and the return on investment (ROI) provided to the stakeholder. The original needs of the program must be assessed. Why was the platform purchased in the first place? Are the original objectives being met? Are the numbers of projects and users served meeting expectations? To determine ROI, the value of returns must first be assessed and understood, and then it must be assigned an overall, aggregate, weighted value. ROI is not measured in dollars alone, but in value to the users and the stakeholders. This means the dimensions of value, such as the following, must initially be acknowledged, then measured:

	ASCI	NERSC	NCSA	NPACI
Availability	5	5	4	4
Capability	5	5	5	5
Response	3	3	3	3
Throughput	2	2	2	2
Allocation	2	5	3	3

Fig. 1. Value Dimensions for Ultra-Scale Computer Systems by Site. A scale of 1 to 5 is used with 5 being the most important.

- Availability: Are enough system resources available to support mission critical applications in some acceptable manner? Are users able to achieve objectives on priority jobs?
- Capability: Was enabling capability jobs a reason the platform was purchased in the first place? Can an important task be performed overnight? Can the resources required to accomplish the necessary simulations be reasonably acquired?
- Response time: Do the simulations enabled by the resources exhibit a demonstrable decrease in turnaround time, as expected by users? For specific classes of applications or users, is the response time appropriate?
- Throughput: Is the throughput (number of jobs) meeting expectations?
- Allocation: Is the system sufficiently agile to meet diverse user needs? Are the numbers of projects and users served meeting expectations?

To assess the dimensions of "value," focus should be placed on these areas. Each site performed a self-assessment on the dimension of "value," using a scale of 1-5 with 5 being the highest value (see Figure 1). As expected, there were variations in the assessment of the value of the selected parameters.

Accelerated development computers are purchased to shorten the time to develop critical new software, create new capabilities, perform key new calculations, increase the productivity of key scientists and engineers, and decrease time to market. For simulations performed on highly parallel platforms, this means advancing parallel simulation software, which in turn requires experimentation for a full range of problem sizes up to and including use of the largest system sizes available.

A very large, accelerated development computer would be considered fully utilized if adding more work to the queue of jobs awaiting execution serves only to increase the average delay for jobs in the waiting queue without increasing the throughput. Note that this definition makes no reference to the utilization rate for any of a computer's many sub-system components such as memory, disks, or processors. It does assume that when a job is assigned to a particular node (set of tightly coupled processors, memory, disks, etc.) within a parallel computer, all of those resources are unavailable for use by other jobs, and therefore are considered utilized. The peak theoretical utilization would then be achieved when jobs were assigned to all nodes all of the time.

2 Utilization as a Metric

Utilizition is the most natural metric for a computing facility. But what is meant by *utilization*? There are at least two commonly accepted definitions: One is the fraction of node hours used out of the total time the advanced computing platform is available for use. Another is the fraction of time the platform is in use regardless of its availability. The distinction between these two definitions is relevant mostly for new machines where the machine is unavailable for significant periods of time and, as a machine matures, there is less down time and the two definitions converge. According to either definition, a computer would be considered fully utilized if adding more jobs to the queue of jobs awaiting execution serves only to increase the average delay for jobs.

Neither of these definitions refer to the utilization rate for any of a computer's sub-systems such as memory, disks, or processors. Rather, it assumes that when a job is assigned to a particular node (set of tightly coupled processors, memory, disks, etc.) within a parallel computer, all of those resources are unavailable for use by other jobs and therefore are considered utilized. Theoretically, peak utilization would be achieved when jobs were assigned to all nodes all of the time.

A serious problem with utilization as a metric is that driving utilization to too high a level almost always results in an overall slowdown in system performance. When the slowdown is significant, the effect of achieving very high utilization is a counter-productive decrease in the ability of the system to support the applications for which its acquisition was justified. Another and more subtle weakness with utilization is that it does not measure the capability quality of the machine. In fact, the replacement of many capacity jobs by any capability job requiring the same total amount of resource can only decrease the utilization. Utilization as a measure penalizes exactly those capability jobs that are the driving rationale for the creation of large, integrated, ultra-scale machines.

Historically, managers of advanced computing platforms have used a variety of approaches to assess system utilization. The NASA Numerical Aerospace Simulation (NAS) Facility, for example, has operated parallel supercomputers for the past 11 years, including the Cray C-90, Intel iPSC/860, Intel Paragon, Thinking Machines CM-5, IBM SP-2, and SGI Origin 2000. The variability of the Available Node Utilization of some of those machines is shown in Figure 2. In last year's workshop (*Job Scheduling Strategies for Parallel Processing*, D.G. Feitelson and L. Rudolph (Eds.), Lecture Notes in Computer Science, Vol 1659, Springer-Verlag, 1999), James Jones and Bill Nitzberg presented a historical perspective of achievable utilization. Recognizing the range of machine architectures, a time span of more than six years, large numbers of different users, and thousands of minor configurations and policy changes, they show that the utilization of these machines reveal three general trends (see Figure 2):

- scheduling using a naive first-in, first-out, first-fit policy results in 40-60% utilization
- switching to the more sophisticated dynamic backfilling scheduling algorithm improves utilization by about 15 percentage points (yielding up to about 70% utilization)
- reducing the maximum allowable job size further increases utilization This last policy however defeats one of the purposes for buying ultra-scale machines, namely to gain new capability. Most surprising is the consistency of these trends across different platforms and user communities.

2.1 Experience with a New System

One example of the process a major facility must go through in placing a new ultrascale capability into service is demonstrated in Figure 3. In 1997, NERSC at Lawrence Berkeley National Laboratory (LBL) transitioned its primary production computing capability to a massively parallel ultra-scale computer by placing into service a large, early delivery Cray T3E. At the time of introduction, this system was the largest unclassified supercomputer in the US and represented a 20-fold increase in raw computing power to the 2,500 scientists who use NERSC. NERSC, working closely with Cray Research, was able to improve utilization through the gradual introduction and exploitation of major system software functionality such as job migration and system initiated checkpoint/restart. During the first 18 months in service, the T3E utilization increased from approximately 55% to over 90% while still focusing most of the system resources on large jobs. This represents almost a factor of two in price performance increase for the system or the equivalent (in 1999 costs) of $10.25M. At the same time the system was improving, T3E users were making improvements in applications to better utilize the system and improve its scientific output.

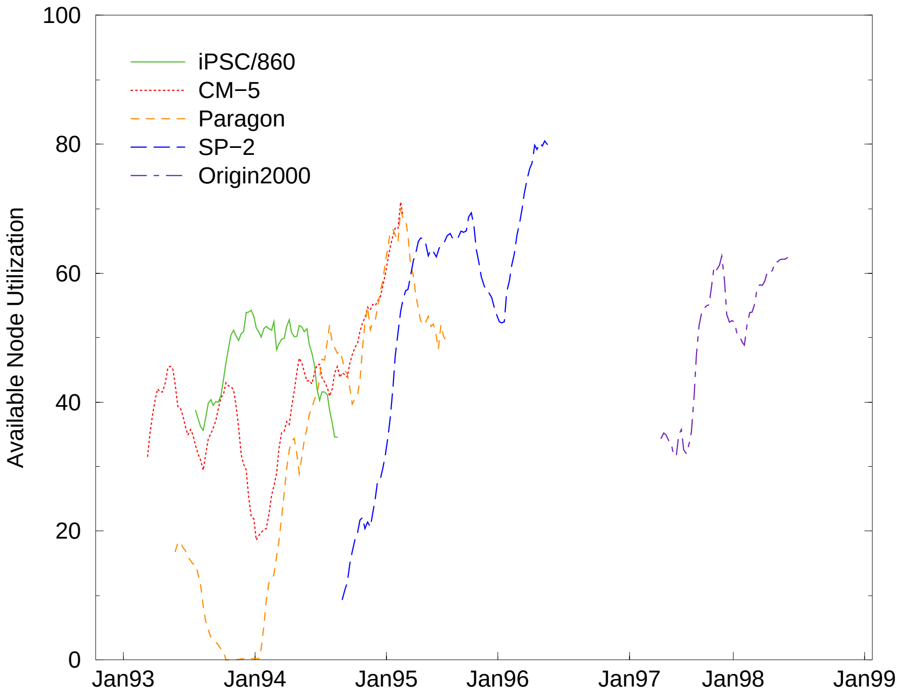

Fig. 2. NASA Overall Comparison of Parallel Supercomputer Utilization

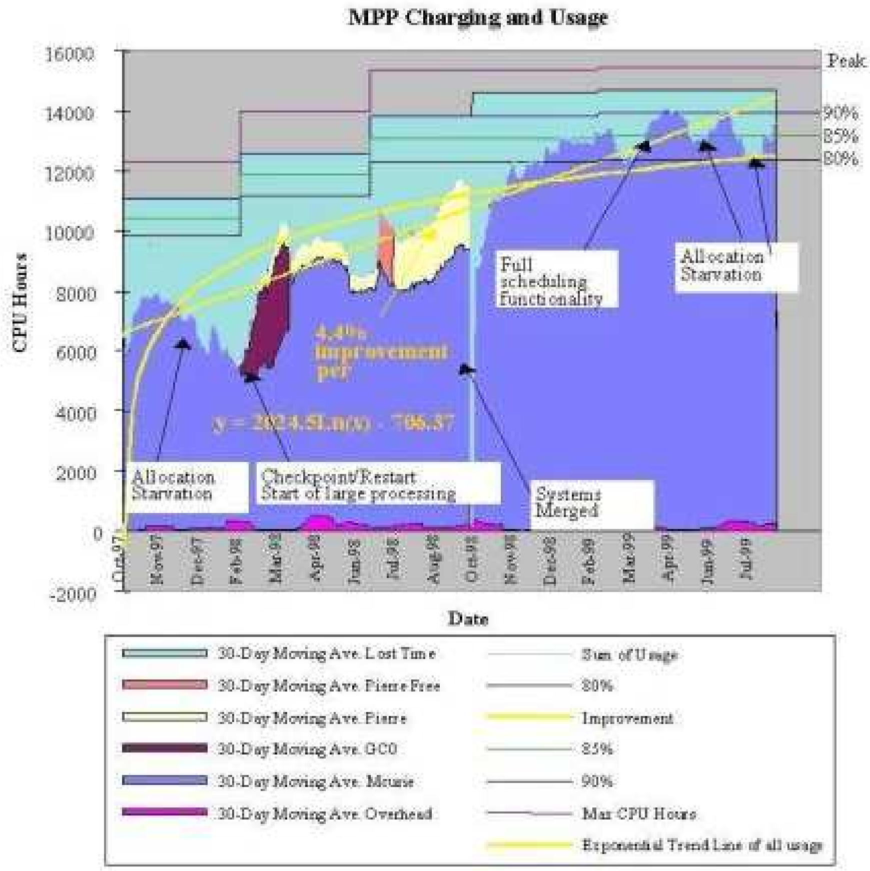

Fig. 3. NERSC Evolution of T3-E Utilization

It is unfortunate that this scenario must be repeated for every new architecture delivered. But rapid changes in the high performance computing industry make it virtually inevitable that each new type system will experience the same learning curve. This is partly due to the facility's and its users' needs to learn how best to implement, tune, and run the new system and applications. But it is also due to the fact that few of the basic system software capabilities are transportable to the general system or can be shared among vendors. High performance computing vendors have little incentive to invest and maintain advanced system software capabilities since, until recently, there has not been a reward structure for creating a system that is more effective rather than yielding faster performance.

Generally, ultra-scale computing platform managers assign preference to large jobs to ensure there are sufficient resources to run and there is a trade-off between quick turnaround for development jobs and maximum efficiency for production jobs. This trade-off translates into decisions that must be made between asynchronous, interactive user behavior and using a batch queue system to provide sustained loading over longer time versus peak loading to maximize interactivity over shorter time periods.

2.2 Impacts of Allocation of Resource Decisions

The ways that access to an ultra-scale system is authorized also greatly influences utilization. If system resources are either under or over allocated, usage and client satisfaction can be greatly degraded. Consider the typical impact the following allocation methods can have on utilization:

1. Exact division of resources (typically CPU time and storage) and strict enforcement of limits subdivide the resources of the system on some priority basis. By exactly cutting up the total resources and limiting users to their share, a system can easily become underutilized. There is a "feast or famine" mentality that causes users to hoard time early in the allocation period. If enough users hoard time, then utilization is low early in the period because there are not enough jobs to keep the system busy all the time. Later, when everyone tries to use their time, they may not be able to use it completely. The results are underutilized resources AND degradation of quality of service.
2. Some sites mitigate the previous impact by periodically taking time away from users who have not used it and redistributing it to users who are short on time. Since time lost due to idle processors cannot be recreated, this typically results in a scarcity of work early in the period and a severe backlog late in the period. It also is often cumbersome to implement and manage. If the recapture of time is done too frequently, it results in several mini feasts and famines throughout the allocation period.
3. Another way to address the concern of underutilization is to oversubscribe the system by allocating more resources than it can possibly deliver. This is done in one of two ways. If there is too much oversubscription, users expect

to be able to do more work than the system can perform. If enough users attempt this, the system will be clogged with work and users. More resources will be needed to manage the overload and the system loses efficiency as the quality of service degrades even though it may appear to be using all the CPU time. The second way to overallocate system resources is to institute a priority system, so that a user can submit work at a lower priority, risking expanded run times, while being charged less. Alternatively, a user can ask for higher priority and "pay" more. While this typically yields higher utilization, it also extends the slowdown factors of the system sometimes to the extent that it takes so long to run a job the scientists do not even bother to try. Most sites use one or more of these methods, often in combination, to try to balance keeping the system utilized on the one hand but still responsive on the other hand. No method is entirely successful, so it takes dedicated system managers to be constantly monitoring and tuning the methods.

2.3 Utilization Trade-Offs

Managing utilization of the ultra-scale computing platforms requires systems operators to decide among a large number of complex trade-offs. Factors to be considered include:

– Job mix - This includes the categories and size of job described above (for example, large production runs requiring thousands of processors versus smaller development runs to test and validate code). Job mix requires adequate management of memory, internal network bandwidth, and file system concurrently; the risk is that a large capability job may be starved if any single resource is not managed well. This is complicated by the fact that different types of jobs will require memory and CPU resources in differing proportions (for example, large jobs in chemistry versus computational fluid dynamics).
– People priorities - Some users and/or projects might be considered "more equal" than others because they are completing higher priority work. This means that resources must be available to meet the high priority needs – sometimes to the exclusion of other users and other jobs – forcing managers to provide guaranteed access to fewer, key users at the possible cost of lower utilization.
– Learning curves - The optimal target for a platform is usually running on 1/4 to 1/2 of the entire machine. Use of such large fractions of systems, particularly early in existence, is not likely as system operating software is still being developed and applications programmers are still becoming familiar with the scheduling processes and operational algorithms required to make effective use of the ultra-scale platform. One conceivable alternative is to run only small jobs. However, in practice, the only way to ensure that a machine is ready is to subject it to real jobs and real workloads. Therefore, when users are kept off the machine with the goal of fixing all the problems, the net result is serious delays in the development and scaling of applications

to make use of the capabilities and features of the system. This in turn leads to further utilization problems.

– Absence of Tools - Because the ultra-scale platforms are first-of-their-kind, tools for measuring efficiency, accounting for use, and for tuning system parameters for higher levels of efficiency are not yet in place. There is an imbalance between the size and diversity of the software needed and the size of the new systems. Initially, accusations of low utilization are often met with anecdotal evidence and little systematic data; time and sponsored efforts are needed to evolve better tools for these platforms.

All of these factors, and the trade-offs that must be made among them, have to be balanced when managing ultra-scale computing platforms. Managers must respond to a highly complex problem with a large number of degrees of freedom. Scheduling efficient use of all of the resources is like a "Tetris" problem; the right job at the right time is needed to consume whatever resources are available. If there is conflict or overlap, utilization efficiency may decrease.

2.4 Utilization Should Not Be the Sole Metric

As these arguments are meant to demonstrate, utilization is not a universally defined term and different organizations use different approaches to define it. The Project participants believe strongly that the true measure of the value of ultra-scale computing systems in the long run should be the scientific output of these systems. Are the systems doing what they were designed and funded to do? How is this measured? The answer is that the overall value of the ultra-scale platform must be assessed to those that have purchased it and taken advantage of its capabilities. This is very effectively achieved by periodic peer review of the facility, as is done with national facilities. In the end, the facilities that operate ultra-scale computing systems should be judged in the same way other national facilities such as accelerators are judged. Typically, periodic peer review is used to assess whether they are meeting their missions and goals. Assessments evaluate and provide guidance in the areas such as: ? Does the facility operate well? Are the systems run well, are they reliable, is the facility meeting user expectations, etc.? ? Is the facility doing the appropriate research and development necessary to keep it at the forefront of its discipline? ? Is the facility doing what it can to ensure, in the aggregate, that the best science is being produced from its resources? Such peer reviews have worked very well to ensure the effectiveness and efficiency of facilities that serve the targeted scientific community. The value of ultra-scale computing facilities and the scientific output of the systems should be evaluated in a similar manner.

There is no single metric for utilization because every platform manager, program, and complex problem to be solved is working towards specific (and somewhat different) objectives. The managers of the programs and the platforms must first define the overall value of the new tool in meeting objectives and then assess how successfully those objectives are being met with respect to the use of this sophisticated

3 A New Conceptual Approach

Although some consideration of utilization is appropriate, a slowdown effect in the system can result when utilization is driven too hard (Figure 4). If the slowdown is significant, the effect of focusing on utilization can be counterproductive on overall system performance and on the ability of the system to be used for the type of applications for which its acquisition was justified. It was found that a "smoothly" running system (for example, ultra-scale computer systems) will find optimum utilization at the "knee" of the curve. One would want to increase utilization from small values until the slowdown becomes too large. Acceptable slowdown values may be different for different operations. For example, the exact slowdown-utilization curve depends on the type of machine, software, and job mix (e.g. Case 1 and Case 2 in the figure). The curves all look the same but have different constants, and hence the knee occurs are different places. Slowdown impacts user behavior which, in turn, affects the amount of load on the system (reduced utilization) and, more importantly, ultimately affects what the user is able to accomplish.

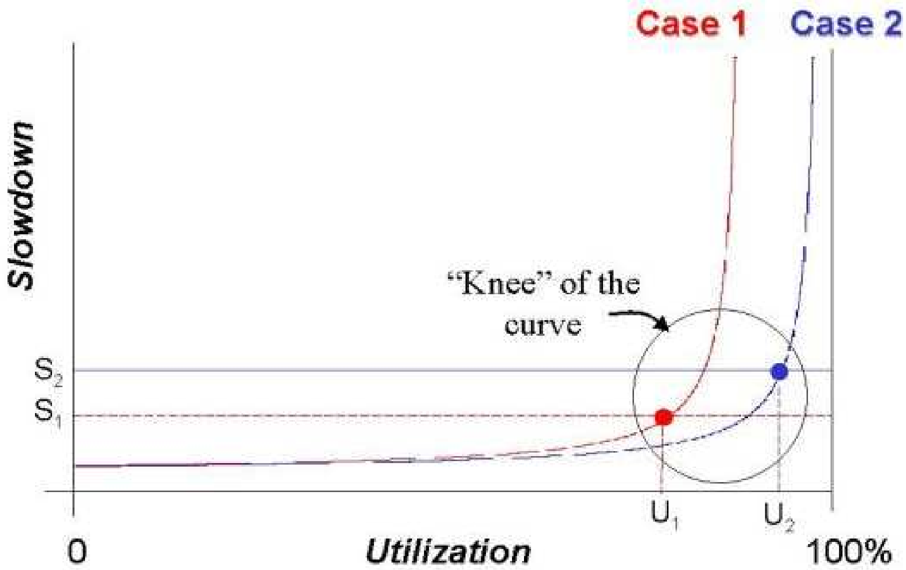

Fig. 4. Slowdown vs. Utilization

Preliminary examination of the sample data showed that this normally expected "slowdown-utilization" curve does occur although it is not immediately evident from raw trace data. The implication of this hypothesis is that systems operating at the "knee" are operating at the best range for those systems, that is, at the ideal point.

3.1 Trace Data Analysis

As part of the Project, trace data from several sites (see Figure 5) were collected and examined. It should be noted that the data did not cover the same time period or even the same length of time at each site. Furthermore, different machines collected the data, used different schedulers, and had different workloads. Because some of the trace data contained partially complete records, some information was lost as it was converted to a standard format. Despite all these differences across the data sets, a standard pattern was detected. Although acceptable slowdown associated with utilization was found to be near 60%, the data clearly show that there is no absolute acceptable utilization number. For the purpose of this section, instantaneous utilization is the percentage of total time that is used by running jobs – not percentage of available time or fraction allocated to jobs.

Before describing the analysis, it is important to highlight a standard queuing theory expectation. It is a well-established fact for service systems that the average response time increases as the "offered load" increases. The response time is flat until the load crosses a threshold, at which point the response time increases exponentially.

Org	Machine	Max CPUs*	Period	#Jobs	#Queues
LANL	SGI Origin	2048	7/24/99 – 8/31/99	30,000	18
LLNL	SP-2	1344	Months	20,000	3
NASA	SGI Origin	256	Months	32,000	2
NPACI	SP-2	128	1/1/99 9/27/99	22,000	4
NPACI	T3E	272	5/1/99 9/27/99	5,000	40
NPACI	T90	14	1/1/99 9/27/99	25,000	45
NCSA	SGI Origin	512	6/30/99 7/30/99	10,000	36
NERSC	T3E	644	1/1/99 7/1/99	90,000	12

Fig. 5. The data analysis was based on trace logs from these sites. The largest number of CPUs for which trace data were available, not the size of machine are presented.

Since a high performance computer is an example of a service system, such a pattern should occur. In many systems, it is possible to submit jobs to "closed queues" that may not be "opened" for quite some time, for example, the weekend queue. For this and other reasons, the offered load was not used. Instead, the average system utilization during the lifetime of each job was measured. Utilization was taken to be the fraction of the total available CPU hours, during the lifetime of a job, that were being used to execute jobs.

Instead of response time, the related measure of slowdown was computed. Slowdown is defined as the elapsed job time (from submission to completion) divided by the run time. For example, a slowdown of two indicates that a job spent as much time waiting to run as it did actually being run. Some sites have job queues that are active only during certain time periods, such as late night and weekends. A job submitted on Monday, for a weekend queue, would incur at least a five-day waiting time. In this analysis, the submit time was changed to be just before the queue open time. Two other modifications were made to the data: (1) jobs with run times of less than one minute were excluded, and (2) jobs with very high slowdown values (due either to queues that were turned on/off or due to an inability to determine exactly when a queue becomes active) were excluded. Both these job classes obscured the results. Finally, the average instantaneous utilization (considering all the included jobs) is noted on the plots below.

The plots that follow reveal that indeed, at higher utilization levels, the slowdown (for example, response time) does increase. It appears that the facility managers do try to keep the response time reasonable. Two types of anomalous situations were found. The first happens when the response time decreases at higher utilization levels. The other occurs when response time increases at lower utilization levels. Further investigation revealed that one must first separate the jobs into different classes because some systems have batch queues for large jobs, others for interactive daytime jobs, and even queues for very long, highly parallel jobs. The slowdown versus utilization curves all fit the same pattern but each has a different Desired Operation Range (DOR). When the analysis is focused on the important queues, most of the jobs are found to reside in the DOR.

Major conclusions to be drawn from the analysis of trace data are as follows:

- High-end and ultra-scale computer workloads exhibit a pattern of acceptable response time up to a certain instantaneous utilization level, which one refers to as the DOR. When instantaneous utilization is pushed higher than that level, average response time increases precipitously and to levels that negatively impact human productivity.
- For many of the systems studied and for the job classes that matter most, the DOR occurs around 60% instantaneous utilization.
- The location of the DOR can change through improvements in system software (for example, gang scheduling) and scheduler queues that are particularly well matched to the workload characteristics. Thus, more mature systems with more capable system software and a well-characterized workload can achieve desired operation ranges at higher instantaneous utilization levels in the later stages of the system life cycle. The figures that follow show average slowdown as a function of system instantaneous utilization for individual sites. This requires some explanation. For each job, the average system instantaneous utilization was computed for the lifetime of that job, and the job was assigned to one of ten utilization buckets (from 10 to 100%). In addition, the slowdown for that job was calculated as the ratio of job lifetime divided by job runtime. Finally, the weight for each instantaneous utilization bucket was computed, expressed as a fraction of the whole weight, and displayed as the size of the bubble. Bubbles with high slowdown values indicate poor system response time. Bubbles with low utilization levels indicate poor system usage. Ideal performance has large bubbles.

A vertical line was drawn indicating the percentage of total node hours for all trace jobs divided by the total number of node hours in the time period. This line is not the average instantaneous utilization of the jobs in the curve, since there may be periods when the system was unusable.

The Site A plot reveals the characteristic rise with most of the big bubbles at the DOR of the curve. At eighty percent, many jobs are seen to suffer from a large slowdown value.

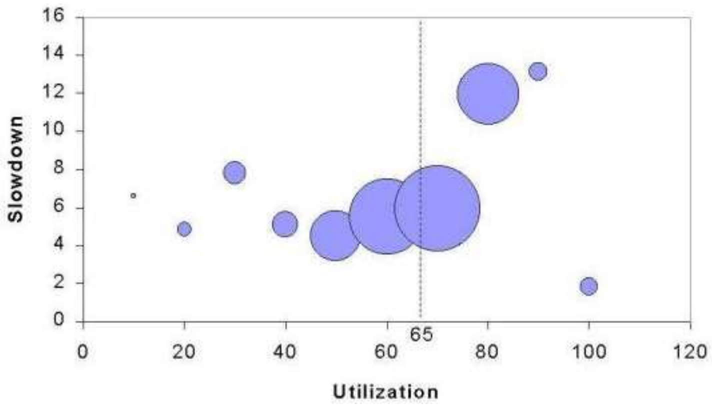

Fig. 6. Site A

The Site B curve looks very similar to the one before it, except that everything happens at a lower utilization level. At 60% utilization, the response time rises, so the DOR occurs at a lower utilization level.

The Site C curves below show a slightly different pattern. Slowdown values are very low and it is easy to see the increase at higher levels. As one can see, the vertical line appears more to the left than would be evident from the distribution of the bubbles. This is because either a long downtime or a short trace period exists. Site C1 has smaller jobs than Site C2; thus the desired operation ranges are in different places, although both seem to manage their systems very well.

The plot for Site D does not show the typical pattern. Most of the jobs have a low response time (not just most of the jobs, but most of the job weight). But there are high slowdown values at low instantaneous utilization values. The reason for this counterintuitive pattern is that there are a number of job classes that are overlaid in this data.

Finally, a yet different phenomenon is observed for Site E. The bubbles appear to have no real pattern. When the plots for the individual job classes are examined (the four plots), however, it is evident that most of the plots follow the usual pattern. The instantaneous utilization appears a bit on the high side for the response time.

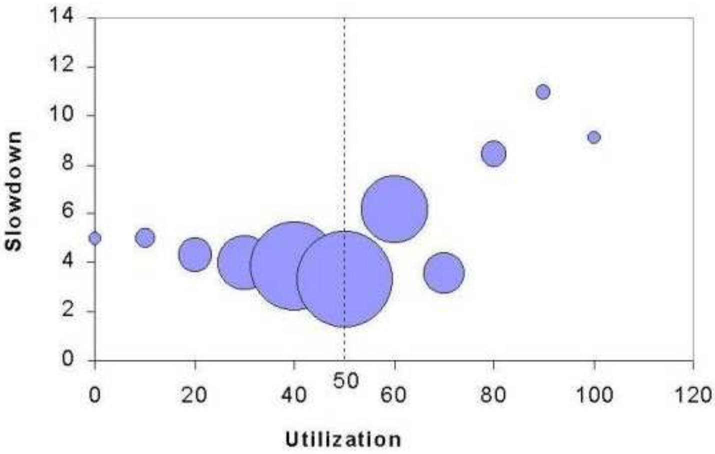

Fig. 7. Site B

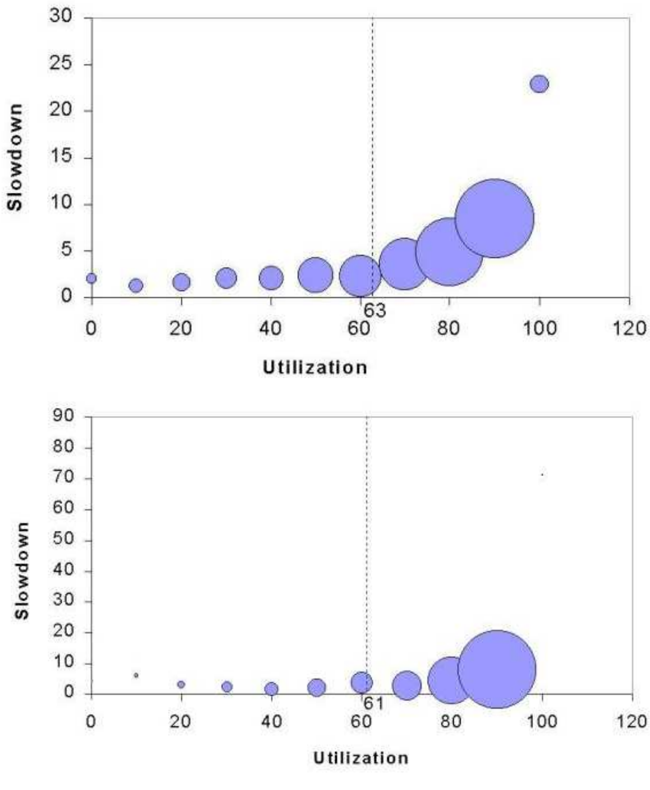

Fig. 8. Site C - C1 is on the top; C2 is on the bottom

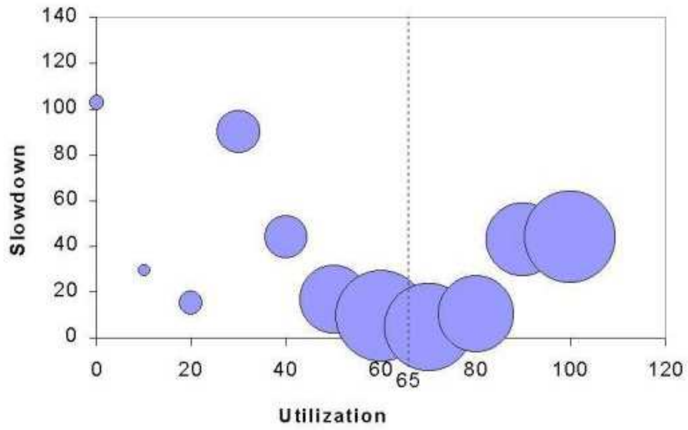

Fig. 9. Site D

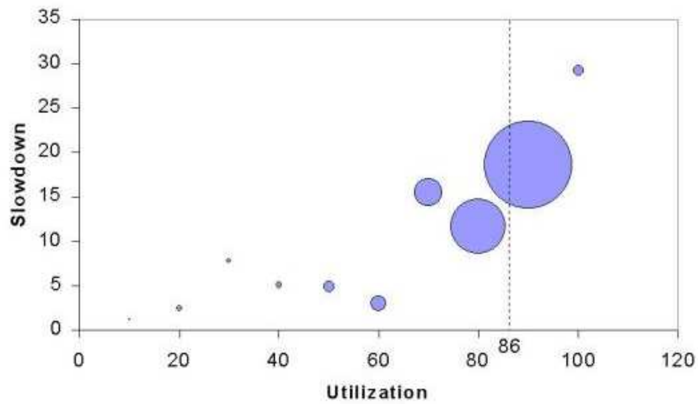

Fig. 10. Site E

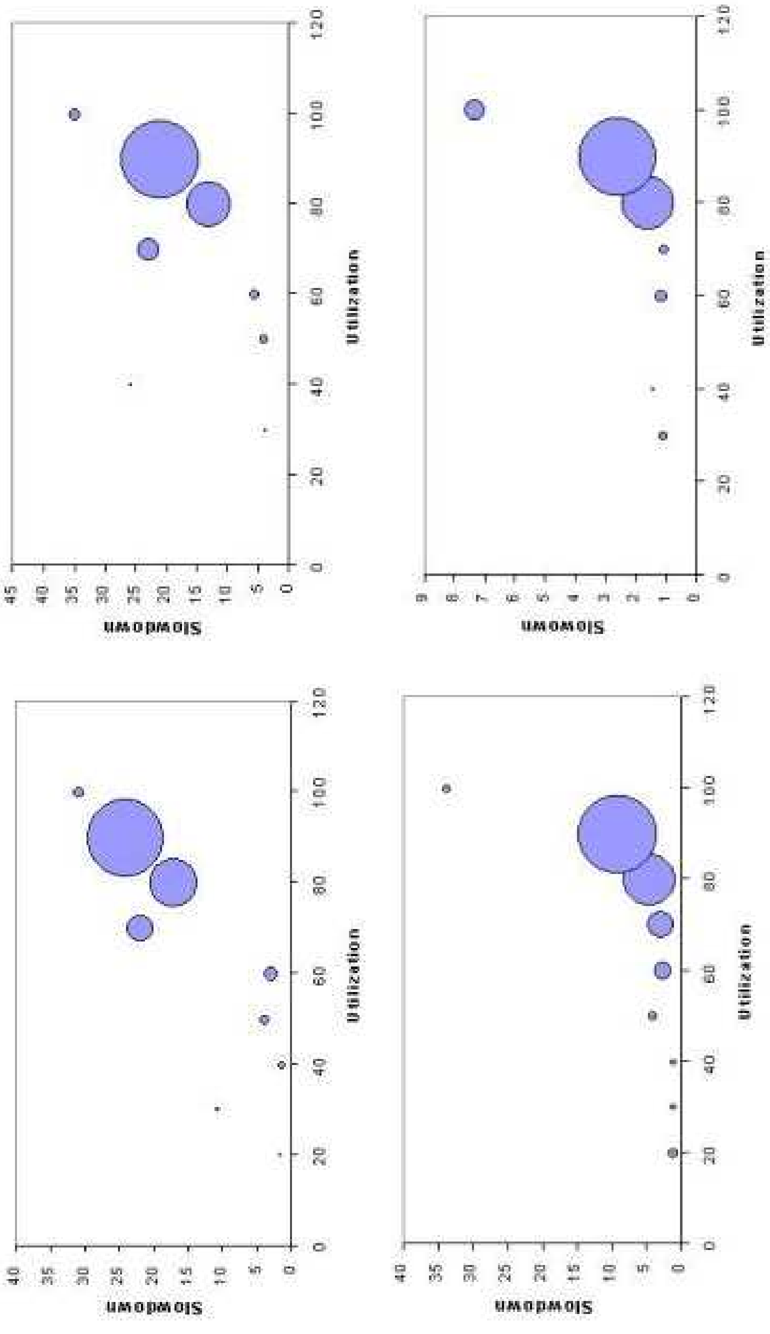

Fig. 11. Different Queues of Site E

4 Recommendations And Conclusions

Project participants recommend that changes such as the following be considered by the ultra-scale computing community to better measure system usefulness and assess the value of advanced platforms:

- Change the day-to-day operations of ultra-scale platforms so that the use of this sophisticated resource lies close to the left side of the "knee" of the "slowdown vs. utilization" curve. Exact changes to be made, both near-term changes in practice and longer-term modifications to policy, should be dictated by the mission requirements of each computing platform and the particular needs of its key users.
- Undertake concerted efforts to better characterize the workloads for each platform. Managers must thoroughly understand the sizes and types of jobs that are to run on their ultra-scale platforms in order to move utilization towards the optimal point on the curve for each site.
- Design benchmark suites and performance models to predict the effectiveness of the systems instead of solely measuring utilization after the fact. It is recommended that the results of the ESP effort mentioned in the previous section, for example, be brought to the attention of advanced platform managers in government, industry, and at educational institutions.
- Adjust the configuration of advanced computing platforms. This may require a re- balancing among processor, memory, and interconnect bandwidth capabilities to better address the specific job mix of the particular computing system. It is recognized that additional resources may be required.
- Analyzing performance statistics across a wider variety of system resources.
- Establish performance requirements for collecting trace and other data as part of future procurements and encourage the inclusion of better schedulers and other tools in the system.
- Require system vendors to provide access to more system statistics as a performance specification.
- Develop procurement specifications for future purchases of advanced platforms so that the importance of valuation metrics and related system software is evident to the vendors.
- Ensure open and ongoing dialogues with platform users and make changes needed to increase the ability of the users to benefit from new capabilities afforded by massively parallel systems.
- Assess the potential impact of various system enhancements, such as faster job launch, checkpoint-restart, gang scheduling, more sophisticated job scheduling, and backfill schemes. Which is likely the most cost-effective? Encourage software developments in these areas. Establish ways to share the testing and validation needed across sites to develop full production tools.
- Change scientific applications codes to take advantage of new ultra-scale computing resource capabilities.
- Establish a balanced investment strategy for obtaining and managing ultra-scale computing platforms that seeks to both improve scheduling and increase the efficiency of individual jobs.

It is important to undertake research in many key areas to accomplish, among other things, at least the following:

– Designing more efficient scheduling algorithms. Determine how to incorporate other job parameters (memory usage, interruptible jobs, and dynamic node size flexibility) into the scheduling software.
– Techniques for increasing the efficiency of applications by employing moldable and malleable jobs. What percentage of user jobs could be configured, with reasonable effort, as interruptible, as adjustable CPUs at job launch time, or as adjustable CPUs during execution? What improvement in system efficiency could be achieved by means of such changes?

The following research questions were identified during the Project:

– Which system software enhancements and operating strategies have the greatest potential for improving system effectiveness?
– How do the job size/run time distributions vary between various HPC centers? How do these distributions vary between different disciplines?
– What percentage of jobs has power-of-two CPU sizes?
– What impact does job size distribution have on system-level efficiency?

The purpose of the Ultra-Scale Computing Valuation Project has been achieved; acceptable ways to evaluate "ultra-scale" computing systems are being defined and a consensus on these approaches is emerging within the ultra-scale computing community. Reaching agreement on understandable and defensible measures is important to the large-scale computing research and applications programs in government and academia, such as the ASCI program at DOE and the PACI program at NSF, as well as others. Presently, generally accepted metrics do not exist for this evaluation. It is evident, however, that the answer is not found by merely assessing node utilization.

The Project Co-Chair's contention given all the system limitations and constraints is that "things are in good shape as far as the running of the advanced computing platforms is concerned," is based on sound peer review of the approaches currently used to manage utilization of advanced computing platforms. Participants agree that a balance of research, development, and implementation considerations is necessary, but argue that the success of high-end computing efforts aimed at enabling new classes of applications should be measured primarily by whether the use does, in fact, result in new knowledge. If so, then the advanced computing tools used were worth the investment.

Participants agreed that the ultra-scale computing community should focus on creating the right-size tools for every scientific and programmatic mission. There is recognition of the responsibility of computing systems managers and the overseeing agencies to determine how best to measure the overall value of each system to its users. In addition, ways must be defined to make needed measurements and compare against recognized benchmarks and to establish operational practices that are optimal for each site and the scientific goals that site is designed to achieve.