SURVEY

In Pursuit of Petascale Computing: Initiatives Around the World

Steve Conway
Jie Wu
Richard Walsh
Charlie Hayes
Earl Joseph, Ph.D.
Daniel Lee
Lloyd Cohen

IDC OPINION

The high-end high-performance computing (HPC) community has a penchant for marching toward milestones — the first sustained gigaflop (circa 1989), the first sustained teraflop (circa 1998), and now the first sustained petaflop. The pursuit of each milestone has led to important breakthroughs in science and engineering. Today, leading HPC sites have begun to identify applications that will benefit from petascale performance. Governments around the world, in partnership with HPC vendors, have launched initiatives to support the development and deployment of petascale systems. Much has already been written and spoken about the challenges and benefits of petascale computing; meanwhile, the U.S. Department of Energy, its labs, and DARPA are actively exploring requirements for the next milestone — exascale computing. Highlights of our analysis are as follows:

- The petascale era will arrive in stages — peak petaflop systems (late 2007–2010), Linpack petaflops, sustained petaflops on "embarrassingly parallel" applications and, finally, sustained petaflops on challenging applications.
- Gone is the era when only the United States and Japan had the wherewithal to develop and deploy the highest-performing class of HPC systems. Petascale initiatives also exist in Europe and China — although the U.S. and Japanese initiatives are the most fully conceived and may still be the highest performers.
- Petascale architectures will range from ultralarge clusters to "hybrid" designs that aim to deliver more performance by coupling together (loosely or tightly) multiple processor types via high-bandwidth interconnects.
- Environmental factors (power, cooling, and facility space), the number 2 or 3 issue for system administrators today, will be even more important for petascale systems.
- Petascale computing will have major implications for other areas, among them storage, programming languages, and software — including system software, scalable algorithms, and applications software. IDC studies show that few third-party software applications will be ready to exploit petascale systems.

Global Headquarters: 5 Speen Street Framingham, MA 01701 USA P.508.872.8200 F.508.935.4015 www.idc.com

Filing Information: September 2007, IDC #208623, Volume: 1

TABLE OF CONTENTS

P

LIST OF TABLES

P

IN THIS STUDY

This IDC study presents detailed summaries of presentations on petascale computing by a variety of HPC users and vendors, along with highlights from IDC studies on HPC user requirements for petascale systems and essential guidance for market participants. The presentations occurred at meetings of the HPC User Forum, which is a user-directed, IDC-operated organization that holds HPC conferences twice per year in the United States, with additional meetings in international locations. The HPC User Forum was founded in 1999 to advance the state of high-performance computing through open discussions, information sharing, and initiatives involving HPC users in industry, government, and academia, along with HPC vendors and other interested parties. The organization has grown to more than 150 members.

SITUATION OVERVIEW

Petascale Initiatives Around the World

DARPA HPCS Program: Cray Cascade Program (Peter Ungaro)

"We've dedicated our entire company to building machines that scale as high as possible." Supercomputing is increasingly about managing scalability. [Ungaro showed a slide on the average number of processors in the top 20 supercomputers over the past five years.] In 2006, the figure was 18,000 processors.

There are increasingly complex application requirements. In earth science, there is increased complexity, and the number of components in the models is increasing, as are the scales. This means that there isn't a single machine that does all of this well. So, Cray came up with Adaptive Supercomputing, where you have multiple architectures within a single system. You start with a scalable system infrastructure and put multiple processor technologies on top of that, depending on the customer's applications and workload. On top of that, we put Linux; and on top of that, an adaptive software stack that isolates the user from the underlying complexity. So, the system starts to adapt to the application rather than the other way around.

It will be 2010 by the time we come out with a prototype, so Cray is building steps toward this vision. The first step, called the Rainier Program, integrates various common elements into the Cray XT infrastructure (e.g., interconnect, global file system), and then you can mix and match one or more types of compute resources (e.g., XT4 compute blades, XMT compute blades).

The next step will be Cascade. The first Cascade goal is to design an adaptive, configurable machine that can match the attributes of a wide variety of applications — serial performance, SIMD data-level parallelism, fine-grained MIMD parallelism, regular and sparse bandwidth of varying intensities, and so on. The focus is on increased performance and productivity. The second goal is ease of development of parallel codes. This will be a great MPI machine that will also run OpenMP. Finally, there will be programming tools to ease debugging and tuning at scale.

For processing technologies, we start with the best-of-class microprocessor. For us, this was AMD because of the integrated memory controller, but HyperTransport was even more important. On top of this, we built two ASICs, one to accelerate communications, and the other to accelerate computation. However, we allow the user to make it very productive. It will merge multithreading and vector processing.

I'll give a high-level view of the Cascade architecture. You have single core as well as SMP nodes, so you can mix and match, plus service and I/O nodes. There's a configurable network, memory, processing and I/O, and a globally addressable memory with a unified addressing architecture. It supports heterogeneous processing. There's a separate set of service nodes with back-end compute nodes of various kinds. It's a very lightweight, jitter-free OS with a very small OS footprint on the compute nodes. It provides robustness with reliable services and common interfaces for programmability, and we leverage standard interfaces for portability.

We will have support for C, C++, and Fortran compilers, plus newer languages. Chapel is a new parallel language developed by Cray for HPCS. Our productivity goals are to improve programmability, performance, portability, and code robustness. The draft language specification is available, and the portable prototype implementation is under way. With Chapel, there are 5–15x fewer lines of code (e.g., STREAM and FFT).

WRF is a great example of an application where one size doesn't fit all. Part of the code is serial, and this will run best on Opteron; part will run best on the vector processor. Other parts don't vectorize well, such as cloud physics. We can run this in multithreaded mode.

DARPA HPCS Program: IBM PERCS Program (Rama Govindaraju)

Over 300 people at IBM are working on this project. I'm responsible for the software side and will focus mostly on that today. It requires significant innovation in all aspects of the system.

We are focusing on a high-level software architecture. We have done a lot of work on the software stack. File systems are very important for serving files to highly parallel systems. Resource managers and schedules are also very important. The rest is systems management and reliability software. The current IBM software stack is reliable and proven.

There are challenges for transport protocols. The PERCS system will have 0.8μs latency, 10GBps x 4 unidirectional bandwidth, disk drives (1TB+ drives) greater than 10GBps, network bandwidth of tens of GBps x N., and performance over 10TBps.

Key observations include:

- Peak performance is easy to add, but getting applications to scale has been a tremendous challenge. The domain of operation for the HPC software stack needs to expand beyond a single supercomputer cluster. The domain needs to extend to the entire datacenter. The operational efficiency of supercomputers is often overlooked.

- MTBF is increasingly critical at scale. Continuous operation means always being able to serve applications. This is critical, even in the event of multiple failures of components; reliability is most important of all.

- In the area of productivity domains, programmer productivity gets the most attention and is very important. Operational efficiency is also important. Administrator productivity involves storage management, network management, installation/upgrades, and system monitoring. RAS entails continuous operation, problem isolation, FFDC, and serviceability.

- Sustained performance is not scaling well. Sustained performance is the percentage of time the processor is doing floating point application work. Anything else takes away from this (e.g., a cache miss, or if I send a message). This falls into architectural reasons and algorithmic reasons (i.e., an application has attributes that cause one or more of the processors to wait). We need to address all the architectural reasons, such as avoiding all cache misses. Vendors also need to provide the right tools for users to alleviate algorithmic issues.

Another key theme is remote direct memory access (RDMA) value and exploitation. RDMA came along 5–6 years ago, but is still difficult to exploit because applications were not written to do this. New techniques are now being developed. RDMA provides more efficient data transfers from CPU to memory. Data goes directly from the user buffer to the adapter. This frees up the CPU. Using intelligent pipelining, you can make this happen even with older applications without having to change the application.

We'll have a cache injection strategy. Cache-miss-free execution is critical. The solution is to inject DMA write transactions into cache. For programming models, today we support MPI, LAPI, and SHMEM. Future models for PERCS include X10 concepts being applied to HPLCS initiatives. We'll support UPC. Component libraries will include ESSL, PESSL enhancements, and COIN-OR. The compilers will have a productivity focus. [Govindaraju showed a chart of the overall PERCS programmer productivity solution vision.]

For reliable operation of the system, RAID5 and RAID6 storage are not sufficient. In a networking infrastructure with thousands of optical cables, packet drops occur more frequently than anything in previous experience. These need to be detected and dealt with, so that the applications can continue running. MetaCluster provides application mobility (checkpoint/restart), based on the notion of containers. MetaCluster detects, say, that a fan has failed, which means a node is about to fail, so it moves that work to another node.

Later this year, we'll offer enterprisewide file sharing across wide area networks (e.g., a datacenter). This works across multiple clusters.

The PERCS OS vision is that for these systems to be productive, you can't depend on a kernel-based approach because vendors don't support it. You also have to address OS jitter.

Comment: Can you give a brief description of the hardware architecture?

Speaker: It will be based on Power7, part of our mainline system. Power6 will come out this year. It will have a very unique integrated network, with very low latency, high bandwidth, and some support for minimal vector operations.

A Petascale Initiative in China (Zeng Yu, Dawning)

[Zeng Yu is general manager of the R&D center for Dawning Information Industry Co. Ltd., Beijing.] From a $20 million company in 1995, Dawning has grown to net $2 billion in annual revenue today. Dawning is the number 1 vendor in China's server market and has a 90% share in China's HPC market. The company is also the top supplier to the education industry in China. For example, Dawning will deliver 3,000 servers for a Beijing project for high school admissions. The company is also a major supplier for the Beijing 2008 Olympic Games, the Shenzhou manned airship, and many more projects. Dawning can help non-Chinese vendors with access to the Chinese market, in the form of know-how, support, cost-effective manufacturing, and other assistance.

Dawning is using industry-standard systems for building a 100-teraflop system (the Dawning 5000A). The system will include 131 blade servers, 22 cabinets with 100.608 peak teraflops, and an estimated Linpack performance of 60.36 teraflops. It will use InfiniBand and Gigabit Ethernet, with fiber channel connectivity to storage and the Lustre file system and will have 670kW of power.

The Dawning 5000A will be based on blade servers (7U 10 computing blades), and it will have a management and monitoring module that manages the entire system. The design will be heat balanced. The cluster operating system will include basic and high-level functions (e.g., resource management, load balancing, performance evaluation, backup, and recovery). It will also feature a job scheduling system, a load balancing system, and a role management capability that manages the system based on the nature of the workload. There will also be an Internet-based cluster monitoring system.

The three HPC developers in China are:

- The National University of Defense Technology (vector/MPP/SMP systems)
- Jiangnan Institute of Computing (JIC) (SMP/cluster system)
- Dawning (MPP/cluster systems, mainly for the civilian and commercial markets); the company is strong in the petroleum and other commercial markets

JIC will complete a system surpassing the current Blue Gene/L by the end of 2006. The NCIC/Dawning nonclassified system has been funded by the Ministry of Science and Education for 100 teraflops in 2007–2008.

Major challenges of petascale systems include price, power, balance, the "memory wall," and the programming model. The Dawning system will use the Godson-3 four-core CPU, due out in mid-2008 with a peak performance of 20GFLOPS. In mid-2010, this will advance to 8–16 cores and 100GFLOPS. Key architectural features of the Dawning system include cross-platform support, inverse-pgraph topology, and register-level network.

In the future, Dawning's goals are to capture a sizeable market share for HPC and servers, introduce new product lines, enter more vertical markets, and employ more creative financing. We must have a global vision. There will be huge demands for servers and HPC in China in the future. Relationships need to be mutually beneficial.

European Petascale Planning (Michael Resch, HLRS/Stuttgart)

We started to discuss HPC with the European Commission in the late 1990s. Some people in Europe thought the ASCI program made sense and might be a threat to Europe. In the 1990s, politicians said European HPC was competitive; so we had no impact on politicians. This changed with the Earth Simulator and the publicity around it. After the CNN coverage, people started talking much more about HPC, and European users started caring more about the ASCI program. They said, "we've got to do something about this." This led to committees being set up during 2002–2004, one on the European Strategy for Research Infrastructure and another called the Infrastructure Reflection Group. They produced papers saying that Europe needs to be competitive in HPC. A decision was made in 2006 that HPC should be a goal for European infrastructure. An organization called HPC for Europe (HPCEUR) put together the justification. Also, over the past three years, there have been discussions on research funding for Europe under the framework programs. In 2006–2007, we made the transition to Framework Program 7, and HPC is integrated into this. This is where we stand today.

The HPC for Europe Task Force (HET) is another committee. Its goals are to create a persistent supercomputing infrastructure to support research in Europe, deploy and operate a petaflop system as early as 2009 at one or more sites, maintain European leadership in software and provide researchers and industry with world-class systems, and work together with the European HPC and grid ecosystem. HET assumes that Europe is now ahead in HPC software. The task force came up with tier 1–3 centers in Europe. The tier 1 facilities are national centers. The tier 2 sites are midrange, and the tier 3 sites are universities. The goal is to have a petaflop system in 2009. The big questions are, how many of the top-tier centers in Europe really need this, and who's going to pay for all this? Spain has been investing a lot in HPC recently. Spain's large center is in Barcelona. So, now there are four big investing countries that may be hosts for the top centers — Germany, France, the United Kingdom, and Spain — with the Netherlands being another possible location.

The next step is for HET members to submit proposals for the preparatory phase of the research infrastructure. The European Commission wants a two-stage approach to determine whether a system like this makes sense that includes exploring financial, legal, and administrative issues and doing some technical work. The question is, how do you find a European basis for a program that so far has been about national programs? Also, in cases like Stuttgart, the State of Baden-Württemberg has been a funder, so there can be at least three levels of funding. I also have an audit committee that says if we pay 10% toward the funding, the State of Baden-Württemberg should get exactly 10% of the CPU time. For the grid discussion, the question is, what is the value of one CPU hour? For the legal framework, they also need to discuss TCO. The cost of manpower is different in Spain than in Germany or France.

The second stage will be implementation, including the actual construction of the petaflop system. The community support will focus on the preparatory phase. This proposal is due on May 2. The budget is €10 million for this, and the project is supposed to start at the end of 2007. Competing proposals are not expected under the same topic. Existing projects can be linked to this proposal.

DEISA unites the main HPC players in Europe, linking all the supercomputers together. There is a common scheduler and file system, and then we distribute the CPU time. It's been quite successful. The link is a 10Gb network. We're working on a global parallel file system and a common scheduling policy. There is a lot of overlap between HPC Europa and DEISA.

There are also national unification processes, mainly in France and Germany. Germany is the most federalist country in Europe, where nothing is done without the consent of every state. We have to bring together HLRS, LRZ, and NIC. Germany has the Gauss Center for Supercomputing (GCS) to organize HLRS, LRZ, and NIC activities (**www.gcfs.eu**). France is the most centralized country in Europe but has the same problem, with both the government and universities working on HPC. France has the Grand Equipment National pour le Calcul Intensif (GENCI), a *société civile* (partnership with full liability) that brings together all the major players. About 50% is held by the French government, 20% by CEA, 20% by CNRS, and 10% by universities. All over the world, HPC is mainly about politics.

There is also a technology platform we set up with some of these companies, called the TALOS HPC Alliance. The partners are Bull (France), CEA (France), HLRS (Germany), Intel (Europe/the United States), and Quadrics (the United Kingdom/Italy). Bull is building systems based on Intel products, so we needed Intel to work with us. The goals are to meet technology challenges related to large-scale computing systems and provide these leading-class tools to European researchers and industry.

Los Alamos National Laboratory: The Roadrunner System (John Morrison)

Roadrunner is critical for our stockpile stewardship mission. The contract was awarded on September 7 to IBM. The Cell processor, in 2007, will have a peak performance on the order of 100GFLOPS versus the 1GFLOPS CM-5 board in 1994. The hybrid final system will achieve petaflop performance. Our vision is about faster computation, not more processors. Cabinets start showing up next week for the initial system.

We've coasted during the past decade, exploiting clock-speed improvements from the microprocessor industry; so we've used the same programming model and haven't really worked much on the algorithms. Over the next decade, performance gains will come through architectural innovations; so now is the time to prepare for that. We need a lot of investment in our applications. We need to exploit these architectural innovations by looking at the algorithms.

Roadrunner will be implemented in multiple phases. The first phase will include about 70 teraflops in the secure computing environment, plus 5 teraflops in the open environment. Phase 2 will be a technology refresh. In phase 3, we will populate the entire classified system with AA and achieve sustained petaflop Linpack performance.

The base system for the near term is an 8-way IBM x3755 server, with 76 teraflops of Opterons and 1.7 petaflops of Cell, for the sustained petaflop Linpack Top500 run. The compute node rack houses both Opterons and Cell processors. In Phase 3, we just add Cell blades to the same chassis.

The IBM Cell is the heart of our advanced architecture phase. There will be one Cell processor per Opteron core, or 16,000 of each. The Cells will be connected via InfiniBand links. The Cell chip in late 2007 will have about 100GFLOPS of double precision floating point capability, giving us about a 7x performance boost.

Cell is an 8-way heterogeneous engine. It's an independent engine. The PowerPC feeds them. Each Cell has its own DMA access to 4GB of DDR-2 RAM.

The host system hands off work to the Cell processor, which spawns parallel tasks that iterate and generate results that go back to the host system memory. We're working on some of our key algorithms, experimenting with different approaches. We should be able to give an update at SC06.

Phase 1 will begin this month and next. The system is scheduled to be in production in March 2007. The phase 2 technology refresh will include Cell Blade Plus and Software Development Kit 3.0. For phase 3, we'll do a technical assessment in August and September 2007 and decide on the phase 3 option in October 2007. If the decision is yes, we'll take delivery in the first quarter of 2008, and the acceptance will be in the second quarter. Then we'll do the 1-petaflop Cell-accelerated Linpack run.

Regarding infrastructure requirements, the phase 1 system will require 3.5MW in total, and the phase 3 system will double this to 7MW. We'll need to upgrade our facility for this.

Comment: You said when you go to AA, you will cooperate with IBM to do the hybrid programming model. What will the programmer responsibility be for using this model?

National Energy Research Scientific Computing Center (Horst Simon)

The term *petascale* is much used but ill defined. There's a big difference between a peak, Linpack petaflop and a sustained petaflop on a real application. The age of petascale computing will occur when there is widespread use of petascale technology for application performance above 1 petaflop (i.e., when all Top500 machines must have at least 1-petaflop Linpack performance). If you do a straight-line extrapolation of the Top500 results from 1993 through the future, you can see some interesting results.

The first Linpack teraflop was in June 1997. Eight years later, in June 2005, all Top500 systems had teraflop Linpack performance. Based on this extrapolation, I predict the first petaflop system on the list will appear in November 2008, and that it will take seven-and-a-half years, until June 2016, to enter the age of petascale computing, when all the Top500 systems will have petascale Linpack performance. So, it will take a whole decade from now to get to the age of petascale computing. The first Linpack exaflop will be in 2018.

One of the big challenges is scaling. NERSC estimates that a 2010-era sustained petaflop system might have 150,000–500,000 multicores. Almost no application today is ready to scale to that. We have stagnated for a decade. We have not increased parallelism on our systems since the mid-1990s. In June 1993, the CM-5 had 1,024 processors; ASCI Red had 10,000 processors in 1996, and we didn't get beyond that until Blue Gene/L in 2006 with more than 100,000 processors, yet we're still using the same MPI programming model.

In the early 1980s, there were fewer than 100,000 transistors per chip. With Blue Gene/L, there are more processors than these chips had transistors. Using MPI is analogous to having an explicit programming instruction for each transistor on the early 1980s chips. We need at least to get to "assembly level" language.

The next problem is power. NERSC estimates that HPCS systems in the 2010 time frame will require 20MW, 16,000 sq ft, and $12 million per year in electricity costs. We project this could go to 60MW in the 2015–2017 time frame. Power is an industrywide problem, not just for HPC. Google is building a plant in Oregon that can use river water for cooling; it is on the site of a closed-down aluminum smelter that was there because electricity is cheap along the Columbia River. Does it make sense to build HPC systems that require as much power as an aluminum smelter factory? Only a few HPC sites in the United States can afford to house petaflop sites. Midrange installations will soon hit the 1–2MW wall. When operating expenses exceed acquisition costs, we'll hit a turning point where we need to switch to low-power processors.

The HPC industry will switch to low-power technology within 3–4 years. Embedded processors or game processors will be the next step (Blue Gene, Cell, and SiCortex). SiCortex is building a cluster with embedded processors. Zeng Yu of Dawning talked about Godson-3 using MIPS processors, which exploit a low-power core.

Challenge number 3 is the Petascale bubble. There are many petascale announcements from users, most of which will be vaporware. There's a petascale mania. The reality is, there is only one general-purpose system worldwide with more than 100TFLOPS Linpack performance: Blue Gene/L.

Now for my petascale announcement! Cray won a seven-year, $52 million NERSC contract, with delivery in 2007 of an initial 100-teraflop system with an expected 16 teraflops sustained on our workload. There are options for upgrades to a petascale.

The allure of game processors is significant. The Cell processor has real potential for scientific computing. The danger is creating a petaflop before its time. In November 2008, there will be a Linpack petaflop system, but this is likely to create an unwarranted sense of accomplishment. It will take eight more years to make petascale computing commonplace in production environments.

Until 2011, in one sense, we'll have the best of times. The HPC market will continue to expand. Scalability to a few thousand processors is feasible with current technologies. It will also be the worst of times. Cheap clusters will drive out the good, and there will be a further decline of the HPC capability market. Early and easy successes will detract from solutions to the difficult problems of building petascale systems.

All computing will be highly parallel by 2010. The current ecosystem will become untenable at this point. The HPC community has dealt with this for 15–20 years, and there is a great opportunity for the HPC community to lead the way in addressing the whole computer industry's problems of parallelism. We must make the change now.

National Science Foundation Initiative (Jim Kasdorf, Pittsburgh Supercomputing Center)

NSF is trying to create a powerful, stable, persistent, and widely accessible infrastructure for scientists, engineers, and educators. Steve Meacham leads the HPC part of the NSF Office of Cyberinfrastructure. The fiscal-year 2007 budget request for the NSF Office of Cyberinfrastructure is almost $600 million. The HEC program elements include acquisitions, operations, HEC system software development, and HEC petascale application development. This is coordinated with other agencies.

The NSF acquisition strategy includes an annual set of planned acquisitions for Track 2 (capacity systems), each at $30 million. Track 1 is a single system, and the aim is for a petascale computer in 2010. There will be four chunks of $50 million each in funding.

The Track 1 acquisition is to permit revolutionary science with petascale performance on a range of interesting problems (turbulence, QCD, and others). The system must scale and be robust, and it must be a single system. There's a huge laundry list of research problems. It needs to go into production operation by June 2011. Preliminary proposals have been submitted and reviewed.

Track 2 acquisitions are for capacity systems. The first Track 2 award recently went to the Texas Advanced Computer Center (TACC). It was for $59 million, with $30 million for the kit plus $5 million per year for operations. The system is a Sun Constellation Cluster, with a 421TFLOPS peak, 105TB of memory, 3,288 compute nodes, and 4x AMD Deerhound/nodes. It will have 52,608 cores, 32GB/node, PathScale adapters, and a Sun Magnum InfiniBand switch. We don't know if Texas put extra money into this or it got a great deal from Sun. The question is, what will the second Track 2 bid need to look like one year later?

Oak Ridge National Laboratory (Doug Kothe)

Kothe said he is an applications person and is relatively new at ORNL after spending 20 years at Sandia. His talk focused more on the applications and the science drivers than on the Cray machines at ORNL. Very impressive science can be done on the current 54-teraflop and the future petaflop machines. Most of the work is related to Department of Energy (DOE)–funded projects that span a broad spectrum of scientific disciplines. The hardware requirements are very broad. There is no hardware sweet spot, given the range of applications, including accelerator physics, fusion, astrophysics, biology, combustion, chemistry, computer science, engineering (Boeing CFD tools), materials science, high-energy physics, nanoscience, nuclear physics, climate science, and nuclear energy (on the horizon).

ORNL hosts 22 projects, about 50 codes, and a few hundred users. The lab is trying to understand how to make the applications run better, using better algorithms. In the past few months, lab personnel sat down with experts and asked what science they would perform on the petaflop machine that ORNL is scheduled to have in 2008–2009 if they had the whole system for one month. ORNL was interested in learning more about both scientific quality and productivity.

The result is a 60-page document focusing on 20 codes and whether they could scale to this level. Many of these codes are at least a decade old and still need work, but can get at multicore processors better than you might think. They present lots of petascale possibilities.

The study also looked at user requirements and matching system attributes with applications. There were over a dozen attributes per machine, such as peak flops, interconnect, bandwidth, disk latency, and so on. ORNL also noted which behaviors drive these attribute requirements, such as for memory latency and random access patterns for small data. Then ORNL sat down with various codes and charted out which system attributes were needed for specific types of codes. It was clear that "one size does not fit all." Some trends appeared. For example, most of the applications want as much node-memory capacity as they can get. Interconnects and memory bandwidth will become more and more important. If we want to do broad science on these machines, a major challenge will be the variety of requirements.

ORNL also looked at algorithm requirements, using Corella's Algorithm "Seven Dwarfs," which says that most applications tend to have one or more of seven specific algorithm requirements. ORNL charted this out by type of application (e.g., structured grids, unstructured grids, and FFTs).

Kothe said ORNL's Leadership Computing Facility today includes a Cray XT3 upgraded to 54 teraflops and 21 terabytes of memory, along with a Cray X1E that raises the aggregate peak performance to 74 teraflops. ORNL's road map calls for a peak petaflop Cray system in late 2008 or early 2009, for which ORNL has funding. This will be followed by a sustained petaflop system. ORNL is currently upgrading to 100 petaflops.

Petascale Computer Projects in Japan (Makoto Taiji, Riken)

We have multiple projects, including:

- MDGRAPE-3, also called "Protein Explorer." This petaflop system for molecular dynamics was finished in June 2006 for the Genomic Sciences Center at RIKEN. Its purpose is to help determine 3D protein structures.

- GRAPE-DR (now through 2008). This is a 2-petaflop quasi-general-purpose system for the University of Tokyo.

GRAPE stands for GRAvity PipE. It's a special-purpose accelerator for classical particle simulations. It's used for astrophysics and molecular dynamics. GRAPE computers started in 1991 and have won seven Gorden Bell Prizes.

One GRAPE advantage is that special-purpose machines can solve problems of memory bandwidth and power consumption/heat dissipation. Another advantage is the "broadcast parallelization" architecture that can broadcast data through as many pipelines as you need.

The MDGRAPE-3 has a peak performance of 180 gigaflops. The power efficiency of these special-purpose computers is remarkable — MDGRAPE-3 runs at 0.1W/GFLOPS compared with the Pentium 4's 124W/GFLOPS. The MDGRAPE-3 chip is a Hitachi HDL4N. There are 12 chips/board, 2 boards/2U subrack, and 5TFLOPS/subrack. The 4,800-plus chips are connected via a PCI-X bus.

For a system with nominal peak performance of 1 petaflop, there are 400 boards. The host cluster is an Intel Xeon with 330 cores, operating at 200kW. It has 22 standard 19in. racks and costs $8.6 million, including labor. System integration was done by Japan SGI.

On the software side, we're porting molecular dynamics packages and also in-house codes. The OS and compiler are only needed on the host cluster. The OS is CentOS 4.3 (x86-64), with the Lustre file system. It was a Gorden Bell 2006 finalist for modeling systems with 14,000 atoms at nominal peak performance of 415 teraflops (~40% of the whole system). After correction, the peak was 192 teraflops and the sustained performance was 55 teraflops, or 29% efficiency. Full system performance will be reported at SC06. The applications include protein structuring, x-ray crystallography, protein folding, drug design, and systems biology.

For MDGRAPE-4, we're looking for a sponsor and $20 million in funding for a 20-petaflop peak system in 2010. Applications suitable for the broadcast memory architecture include codes with multiple calculations using the same data.

The GRAPE-DR project will use a SIMD accelerator with this architecture. The full, 2-petaflop system is scheduled for 2008. It will have peak performance of 0.5TFLOPS/chip (single precision) or 0.25TFLOPS/chip (double precision).

There is also a next-generation national project to develop a leading general-purpose supercomputer in Japan. Today, there is no national center like this in Japan because the Earth Simulator was a single-purpose machine. The next-generation, 10-petaflop system, scheduled for fiscal year 2012 (with hardware by fiscal year 2011), will cost about $900 million. RIKEN is charged with developing the system. The project manager is Tadashi Watanabe, and the director is Ryoji Noyori.

The current status is that there are two hardware proposals: NEC with Hitachi, and Fujitsu. Both proposals have been selected to outline designs. One of these will become the final architecture.

Twenty-one applications have been selected for this next-generation system. We are building benchmarks using these applications (several are finished). The applications are from life science, nanotechnology, physics, astronomy, earth science, and engineering. They include grand challenges such as the next-generation Integrated Life Simulation (multiscale, multiphysics simulation of life), which spans from molecular simulation to whole body simulation.

My personal opinion on future HPC processors is that on the high end there will be vector and Blue Gene/L, and on the low end there will be GRAPE, ClearSpeed, and Cell. These are the two extremes. In the middle, we will have microprocessors (bytes/flop).

Perspectives from HPC Experts

Petascale Algorithms (David Bailey, Lawrence Berkeley National Laboratory)

Not long ago, a 1TFLOPS system seemed like a lofty goal, and a Congressional budget analyst called me and said, "If we fund this, will it once and for all meet all your needs?" I didn't laugh. I told him, "when more power arrives and scientists scale up their codes, add more physics, and increase the resolution of their grids."

Petascale System Trends

A major transition is underway to multiple CPUs per node and multiple cores per CPU. In part, this is driven by the realization that increasing the clock rate will only increase energy consumption and heat. Intel has announced a 1TFLOPS research chip. ClearSpeed, NVIDIA, and others are developing many-core HPC systems.

Little's law says that the average length of a queue equals the average arrival rate times the average delay per customer. Applied to multiprocessor HPC systems, this means that concurrency equals latency times bandwidth (Burton Smith).

Assume a 1PFLOPS system with 1 Pword/s memory bandwidth and 100ns of DRAM main memory latency. Now assume there are two different systems, a commodity design and a superconducting processor design. Each gets concurrency of 10^8.

Concurrency and Amdahl's Law

Assume a system with 100,000 CPUs and a program that can efficiently use these 99% of the time. Assume 1% serial code. Then, the maximum performance will be 1/1,000 of peak performance. To sustain 50% of peak, 99.999% of the operations must use the CPUs. Can we beat Amdahl's or Little's law? Yes, if we can develop "latency tolerant" numerical algorithms. We would need to provide large amounts of high-speed cache on all nodes, although this greatly increases power consumption and heat. The alternative would be new devices, such as nanotubes.

The Challenges of 100,000-Way Concurrency

The OS and programming models must efficiently deal with many-core, many-node systems (you need a jitter-free OS). The algorithms must be redesigned and the applications rewritten. There are reasons for optimism. So far, predictions have been wrong that we wouldn't be able to effectively use new levels of computing power. Many users have scaled codes to more than 1,000 CPUs. When the NERSC computer at LBNL doubled in size in 2004, it was saturated within two weeks. As of January 2006, 20% of compute cycles on the Seaborg system at NERSC were for jobs of 1,024 CPUs or above. By January 2007, this had gone up to 60%.

High-Resolution Climate Modeling on NERSC-3

Only at 50km do we begin to see realistic weather features. Until very recently, climate models could not show hurricanes. How do you scale-up existing codes and still get good performance (not just the right answers)? Performance Engineering Research Institute (PERI) is a 10-institution consortium. One thing that PERI is doing is automatic performance tuning of scientific code, automating the process of tuning software to maximize performance. The aim is to reduce the performance portability challenges facing computational scientists. PERI is trying to address the problem that performance experts are in short supply and build on 40 years of human experience and recent success with linear algebra libraries.

In our focus on performance, we're forgetting numerical precision. Even 64-point floating arithmetic won't be enough. It's already not enough for some numerically sensitive applications. (See paper at **www.crd.lbl.gov/~dbailey/dhbpapers/high-prec-arith.pdf**.) Possible remedies include vendors providing hardware support for 128-bit IEEE, using software libraries that permit C++ and Fortran codes to perform, and making algorithm modifications. Algorithm cost scaling is another issue. For a wide class of 3D physical simulation algorithms, work scales as n^4 while memory scales as n^3. Time and again, the scientific community has adapted to new technology. Numerical precision and algorithm scalability may be significant challenges.

What Software Architecture Will Cope with 100,000 Processors? (John Gurd, Manchester University)

Abstraction conflicts with performance; so in HPC, everyone ignores abstraction and concentrates on performance. I believe this is not sustainable for 100,000 processors. In HPC, as time goes on, everything is stretching: what people expect to be done, the complexity of the application, and the systems we run them on.

People expect software to be readily maintainable. They want software that's able to be easily reused and changed so you don't have to write it from scratch each time. They want predictability (how accurate the computation will be and how long it will take). They also want veracity: Are we computing what we thought we were?

Application trends are toward more data and more operations — mixed models (multiphysics, numerical, and symbolic data/information/knowledge). The driver is better science, so you need more performance.

System trends are more toward distribution than even parallelism (e.g., grid computing). The nature of communications is becoming more complicated. There will be heterogeneity of components of the computing platform. The OS will need to support reliable and dynamic multicomputing, rather than batch processing. Scale is increasing: more processors, more memory, more everything. Each of these topics is challenging on its own.

Gurd's laws of parallel processing state that parallelism is only easy for p = 0 and p = 1; the difficulty level rises as "p" increases. Even uniprocessors have parallelism within them (e.g., four-way superscalar pipeline).

Following are the efficiency expectations:

- Small-scale SMP: ~80% (10 processors, 8x speedup)
- Large-scale SMP: ~20% (100 processors, 20x speedup)
- Large-scale clusters: ~5% (1,000 processors, 50x speedup)
- Petascale systems: Much less than 5%

A key problem for petascale systems is electrical power consumption. This can be solved with enough funding. Another problem is the number of processors required (generally agreed to exceed 100,000). We may need a change in software architecture. (Even if petascale doesn't need this, exascale certainly will!) What might this be?

The 50th anniversary of Fortran is near. This represents the imperative way of programming: Make the programming language a model of the machine, and instruct the machine what to do at every step. This worked for a while, but people wanted a higher-level way, so we started using abstractions, such as the declarative. This describes in an abstract way what the compiler should be instructed to do. This has had a fair degree of success in academic circles, but industry doesn't think in terms of mathematical abstractions, so now the trend is imitative, an object-oriented way of writing codes that exposes a more human view of the problem and lets the compiler figure out how to tell the machine what to do. Both these abstract approaches are difficult to turn into machine instructions that do things quickly. It requires very good compilers, and we don't seem to be able to construct these (ones that turn abstractions into fast code). Many, including our group, have been trying to do this. Performance-oriented people have walked away from abstraction, so the state of the art in HPC programming is almost like the 1950s, oriented toward performance and explicitly instructing the machine what to do now. This is horrendous. As innovation is going on in architectures and applications, compliers and other tools always lag behind this rapid innovation.

Some petascale programming choices are:

- Carry on as before. Better the devil you know. However, diminishing returns or unreasonable costs will almost certainly kill off this route.
- Incremental improvement (better tools). Doubling the utilization halves the cost, but ultimately this will delay the inevitable. Except in very amenable cases, this has been hard for people to do.
- Radical change ... to what?

Analysis

Explicit programming has to go at some point. We need to abstract the programs away from the hardware and create the tools we've failed to create so far. This means creating languages and tools together, with million-way parallelism in mind. Autoparallelizers working dynamically could help (versus speculation), along with languages that make parallelism easier to find, algorithms that expose parallelism, and ready-made parallel libraries.

Rule of Thumb

The problem needs at least one order of magnitude more parallel tasks than the number of processors. You need 1 million tasks available to keep 100,000 processors busy. To make 1 million tasks, you could use $DOACROSS in Fortran, but I don't think this will work. Parallels with the biological world are probably a better approach. Hierarchy is anathema to the memory of the system using Fortran.

In sum, the petaflop era is arriving and poses substantial challenges. Sooner or later, we must change our ways. We need to invent new weapons.

Petascale or Petaflops? (Paul Muzio, Chair, HPC User Forum)

I'll say some of the same things as Dave Probst. (This presentation does not represent the views of the U.S. government.)

Petaflop does not necessarily mean petascale. We need to recognize this. It becomes more of a reality as we go to multicores on commodity processors. If we talk about a petaflop, do we mean 50 teraflops or less (sustained performance) or a petaflop? We need to address balanced systems, ease of programmability, global addressing, and so on. HPCS is heading in the right direction on these issues, but the HPC industry is not motivated economically to do all these things. It's much like the airline industry, maybe worse. It amazes me we're supposed to be in technology but are showing huge flops and implying all the problems can be solved this way. We stress cheap flops, including on the applications development side. User sites are too flops centric and not enough user centric. We need to be more user-centric. There is more emphasis on hardware rather than systems. HPC lacks a stable programming environment. Every time you move from one architecture to another, it's sideways progress. Vendors' profit margins are razor thin, maybe collectively negative. Where are we going? Over the past 10 years, HPC has been driven by commodity processors. Are these becoming so specialized that they are no longer appropriate for technical computing? If we're only driven by low-cost flops, we don't need systems vendors. We can buy out of catalogues.

Muzio showed a parachute problem from the U.S. Army, which is interested in delivering large loads by parachute (e.g., 40,000lb). Each test is expensive. If you want a tough problem to solve, how about modeling parachute deployment, a very complex fluid-structure problem. It's a complicated multiphysics problem that you can't solve today on a commodity-processor system.

The dynamic mesh generation for the fluid-structure interaction was developed at the AHPCRC, using Unified Parallel C. [Muzio showed a visualization of the parachute deployment.] This is an unstructured finite element mesh. It uses a general code that we can use for many applications.

The View from Berkeley (David Patterson, University of California-Berkeley)

I'll talk more about single chips that go into high-performance computers. With the trend toward multi- and many cores, it's as if the HPC industry has thrown a Hail Mary pass. Can people run and catch it? Everything is changing, and new ideas are needed. The Berkeley paper is the work of many people.

The old wisdom was that power is free and transistors are expensive. Now, we've hit the power wall. We can design more transistors than we can afford to turn on. We've also hit the memory wall: loads are slow, and multiplies are fast. The era of getting 2x CPU performance every 18–24 months is over. All this has led to a brick wall. We are about a factor of three off of the Moore's law pace today. The processor is the new transistor.

This time, the industry is betting its future on handling parallelism. A group from many disciplines began meeting at Berkeley in February 2005, and meetings continued through December 2006. This led to seven questions that frame the parallel landscape:

- The biggest goal should be to make easy-to-write programs that run efficiently on highly parallel systems.
- The target should be processors with thousands of cores each.
- Use 13 "dwarfs" to design and evaluate parallel programming models and architectures. A dwarf is an algorithmic method that captures a pattern of computation and communication. Seven of these important numerical methods are based on Phillip Colella's Seven Dwarfs.
- "Autotuners" will be more important than traditional compilers for translating parallel programs.
- Future programming models should focus more on human psychology than on hardware or applications.
- Programming models should be independent of the number of processors.
- Programming models should support a wide range of data types and parallelism models: task-level parallelism, word-level parallelism, and bit-level parallelism.

The 13 dwarfs are: dense linear algebra, sparse linear algebra, spectral methods, N-body methods, structured grids, unstructured grids, Monte Carlo, combinational logic, graph traversal, graphical models, finite state machines, dynamic programming, and backtrack and branch-and-bound.

Parallel systems must at least do well in these 13. The dwarfs have four major roles:

- Give us a vocabulary to talk across disciplines
- Define building blocks for libraries

- Serve as "anti-benchmarks," not tied to code or language artifacts (This encourages innovation in algorithms and languages.)

- Allow us to do parallel research in parallel

The hardware issues are power limits, yields dropping, and validation of chips becoming more expensive than design. The hardware solution has been lots of modestly pipelined processors (5- to 9-stage). Where the number of cores per socket is concerned, the current path is evolutionary. We think many cores is the way to go if we can solve the programming problem. Multicore architectures and programming models good for 2–32 cores won't be able to evolve to systems with more cores than this per processor and systems with thousands of processors.

With programming models, the primary focus recently has been on correctness (of results), not performance. Why write parallel programs if you don't care about performance? Performance must be added to this. Previous parallel programming models were hardware centric, application centric, and formalism centric, not productivity centric. Human-centric (i.e., productivity centric) programming models need to integrate research on human psychology.

The issues with support software are that compilers and operating systems have been very large and complex. It may take a decade for complier innovations to show up in production compilers. For code generation, we're more excited by autotuners. We're looking more at what's best for specific machines.

There is a resurgence of interest in virtual machines (VMware, etc.) to help security and reliability. Mendel Rosenblum sees the OS going back to old days, with thinner operating systems and virtual machine images running on top of hypervisors.

How to Measure Success

Only companies, not universities, can build hardware nowadays, and it takes years. Software people usually don't get to start working hard until the hardware arrives. Then they'll tell you in 3–6 months what's wrong with it, and the cycle repeats. We need to get 1,000-CPU systems into the hands of researchers much earlier. Our idea is to use FPGAs to emulate future machines. FPGAs are still on a Moore's law curve. The idea is that the research community creates "gate shareware," which we call research accelerated many processors (RAMP), and then the hardware companies take over and build the systems. We're not trying to build real hardware, just to emulate it with FPGAs. We got this idea one-and-a-half years ago, and since then we've built a couple of multiprocessors. We ran the NAS benchmarks across a 256-processor RAMP system like this. Stanford did one too. RAMP can bring together all the right people to work on the parallelism problem.

Why Should This Work

This should work simply because the era of faster sequential processors is over. The software/hardware community is fully committed to parallelism. (Moore's law continues with the FPG). Processor-to-processor communication is fast, even if memory is slow. All the cores are at equal distance to the shared main memory. And open source software is moving ahead fast.

Summary

- Parallel machines must be easy to program and run efficiently.
- The 13 dwarfs are stand-ins for future parallel applications.
- Simple processors are the way to go. Many-core beats multicore.
- We need human-centric programming models.
- Use autotuners and deconstructed operating systems and virtual machines.
- We need RAMP to accelerate hardware/software generations.

The big questions are: Where do we go from here? What bold new applications will many-core enable? Can we really design architectures that make parallel programming easier?

A Productive Petaflop is a Hard Sell (David Probst, Concordia University — Montreal, Canada)

A productive petaflop is a hard sell to the HPC community. The philosophical message of this talk, assuming you think petaflop systems are worth building, is that the only way is to use a complex form of heterogeneity; and none of this will have a useful payoff until you can develop system software. Other things are also happening in the world of computing, such as the transition to multicores, and the HPC community can learn from these trends.

Why is petaflop computing a hard sell? It's a fact of life that no vendor can design and manufacture a productive petaflop system without considerable resources. By productive, I mean a system that's easy to program and delivers sustained petaflop performance on a broad range of demanding applications. Users are not willing to pay the price for this type of machine. I'll talk about finding a compromise position.

Many people ask, "what's a fair price for a computer, and what metrics should we use?" You buy a computer because you have some mission that's important, and you're concerned whether it will impact the mission in a profound way. You must do an analysis of utility versus cost, but there's no scientific standard for this.

We talk about demanding applications. Maybe only 12 of us have these, and you don't need the kind of machines I'm talking about. What are demanding applications? They have globally dispersed data structures, so you're stuck with long-range communication and there's no way around it. For this, we need high global system bandwidth, which is expensive and power hungry. However, users won't pay for this. The middle position requires heterogeneity. A main theme is that for advanced computing to exist in the next 20 years, sophisticated software must be radically different from what we have today; and if we don't solve this, the hardware systems will not be useful. We won't get anywhere without high global system bandwidth. You need this for any reasonable programmability. The middle position is to find ways to decrease the cost of bandwidth.

Moving from SPARC to Cell is not a new architecture. I'm talking about system architecture. "System level heterogeneous processing" is my buzzword. This is not new. We are moving to weirder and weirder applications, and one thing these do is they're dynamically diverse in the amount, kind, and granularity of their parallelism and locality. How do you deal with applications that are always shape-shifting on you?

There are many proposals for and different forms of heterogeneity. For example, take a serial processor, a vector processor, and a multithreaded processor — how do you program this mix? The answer: you wouldn't and couldn't. The only way to use this machine is if someone built you a programming model that would translate it into a diverse, heterogeneous execution model. The only way to decouple the programming model from the execution model is a revolution in system software, with totally different compilers, OS, and programming languages. If you don't do this, these systems will be just for a few geeks, and this will be a waste.

My view is that it's not critical whether the processors are serial or vector, and so on. The heterogeneity that matters is of thread state, with some having less state and some more state. You need system software that addresses the heterogeneity of the thread state. A vector processor has a fair amount of state. The only processor without an absurd amount of state is a multithreaded processor. How big the thread is has nothing to do with the ability of the processor to tolerate latency.

Then there's heterogeneous locality. Processor-based latency tolerance won't scale to a petaflop. As a community, we haven't yet figured out locality. The locality that's familiar to us is caches, or temporal locality. The other one is spatial locality. Suppose when you programmed a code, you managed from time to time to get moderate amounts of data to be in some place (pre-position data). And suppose your threads were small enough that you could shoot a thread at one of these clouds of data. Then your latency would go away. If we could solve thread migration, we could solve much of the latency issue. If you get the state of the thread low enough, you can afford to ship it anywhere you want. The compiler would divide the code into threads with various states: heavyweight, moderate-state, and lightweight threads. Sometimes you need each, and need a way to make them synchronize. The important thing is that they should not have any state when they need to synchronize or migrate.

I used to be a multithreaded bigot and thought it would scale. Now I think processor-based latency tolerance of any type will not scale. To use processor-based latency tolerance, the processor needs to generate a lot of concurrency and bandwidth. When you use this, it needs a lot of bandwidth, so it's very expensive.

Conclusion: To some extent, we will need to worry about locality of data. Heterogeneity won't work until we redo system software, everything that's not silicon and wires. Petaflop systems must combine von Neumann and non–von Neumann styles. Processor-based latency does not scale. Advanced supercomputing must incorporate thread migration. Sophisticated system software must provide scheduling strategies that extract performance from rare resources. Finally, if you ignore programmable spatial locality, you're not being very sensible.

Are Programming Model Changes Needed for Petaflops? (Panel Session, Manchester University)

The panel was chaired by Andrew Jones, University of Manchester, and introduced by Paul Muzio.

Andrew Jones, University of Manchester

Petascale and exascale will mean orders of magnitude, more threads than today. This might mean architectural complexity. The experience of most programmers is that scaling to 1,000 processors even on homogeneous architectures is difficult. Performance and usability are competing demands. Do we need a new programming paradigm?

The HPC industry has sustained one big change, moving from Fortran to MPI and from single processor to MPP. This was successful only because it enabled something that couldn't be done otherwise. Will the programming model remain Fortran plus MPI? Why haven't PGAS languages been successful? The HPCS options are not revolutionary. With parallel Matlab, performance may not scale enough. Will there be something new from computer science? Or from the games industry? Or from Microsoft? There are really three questions: Do we or don't we, why, and what will it be?

Ben Ralston, AWE

HPC has to find emerging technologies such as the lightweight kernel on the Cray XT3. This allows scaling but is not disruptive. It doesn't change the programming paradigm. Clayton Christensen from Harvard wrote the paper on disruptive technologies. Reprogramming software typically takes many person-years and isn't undertaken lightly. We won't undergo this unless we're certain of the new direction. I don't think we're ready for a new software paradigm because we don't know where the hardware is going yet. We need to choose the right time to do these things.

Paul Muzio, Chair, HPC User Forum

Seymour Cray asked 25–30 years ago what programming language would be used in the year 2000 and said it would still be called Fortran. If you look at what GM, Dassault, and other major companies are using, it's Fortran. There's a huge investment in that software. People won't throw this out. It represents years of knowledge gained from solving critical problems in science and engineering. Look at the applications NSF is interested in for petascale computing. Which company will invest in QCD software? The applications proposed for petascale computing are not the ones companies or even the defense establishment will invest in. However, there are opportunities for languages like Fortran to change. CAF started on the T3E and provides new capability. PGAS languages are not widely accepted because a limited number of hardware platforms are available to support these languages. Languages need to go step in step with the hardware and have stability within the industry, and the industry has not been stable over the past 10 years. There is much emphasis on petaflops and not on problem solving. We have to get back to the scientific/engineering results, not how many petaflops I can buy.

The human cost in developing the applications is far greater than the cost of the hardware. It's been about "Let's get the cheapest available flops," not about user productivity. Evolution, not revolution, is better.

Ian Reid, NAG

It's great to hear about Fortran. That's still our core technology. Yes, a new programming model is needed, or maybe just probably. I agree it's a question of timing. The days of the single-core treadmill are over. Multicore is here to stay. This raises big software issues: Will your software slow down? Memory? These issues will force us at least to hybrid architectures (distributed plus shared memory) in many places. We're heading for an era of renewed thinking. These are exciting times. There have been lots of different architecture enhancements recently. We also need to enhance the software stacks. We need something pretty special to excite the HPC community. I expect many of the proposed architectures will fail for HPC, though they may succeed for games and other things. You must have portability. There are many processor options, and you could program in their languages and get good speedups, but you can't port our programs onto them. There is a lot of uncertainty about whether the HPCS programming languages will deliver on their promises. We need to be able to migrate the vast existing software stack. Don't forget the single core, because a lot of my stuff is still on there. Software issues are coming to the fore. These are very worrying times for people trying to justify budgets based on promised improvements, but these are also very exciting times.

Stephen Wheat, Intel

Clearly, it's too late for programming model shifts before petaflops computing comes, as petaflops computing is coming in the next year or so. Maybe there's enough time until volume petaflops systems arrive 4–5 years from now. There are many petaflops deployments planned for the next few years; petaflops systems will be in industrial organizations in maybe 4–5 years. If not in time for the petaflops, maybe there's time before the exaflop in the middle of the next decade. Even so, how do we get this new thinking done in a decade? From our perspective as a supplier for volume petascale/exascale systems, you need a homogenous architecture. There is much activity already under way in this realm to apply applications to the entire system. We must conceive parallelism before we can represent it. We need to rationalize the problem first. To make this easier, you need stability of the architecture going forward. That's Intel's viewpoint. Thread count is greatly increasing. We expect that distributed memory will be the de facto environment in the volume space. I don't think globally addressable memory will take hold in the volume space, though it may do so at the smaller SMP scale. We need to look at efficiency. If you look at the first teraflop Linpack system, it used 800kW of power. Today, a teraflop takes less than 10kW. That's an 80x performance improvement for the same power. We expect power requirements to continue coming down substantially in relation to performance. We look at efficiency as time to solution versus power consumed.

We Have 10 Years to Construct the Petaflop Ecosystem (Martin Walker, HP)

I'm responsible for HP's HPC in EMEA. We're blade based, and we've tripled the cooling capacity of standard racks. We also have scalable I/O systems and visualization systems. HP recently installed a 10TFLOPS cluster at the Helsinki compute center.

HP supports the U.S. National Research Council's 2004 report, *The Future of Supercomputing.* We agree with Horst Simon that if things continue on track, the first petaflop system will be on the Top500 in two years, but the real petascale era will begin when you need a petaflop to get on the Top500 list. Processor manufacturers are responding to requirements for more performance. Intel is talking about a teraflop processor. These kinds of devices will be very interesting for petaflop systems.

We have 10 years to construct the petaflop ecosystem. Sustained petaflop performance will require a lot of work in the applications domain to exploit hundreds of thousands of cores. What applications need this, and in which domains?

One potential area for using a petaflop ecosystem is biomedical research. 3D protein structures are important for how proteins function. Leventhal's paradox says that many proteins fold to their native state. The folding problem is enormously complex. You'd want a very intelligent computational way to arrive at the 3D structures. It would take a petaflops system three years to simulate the folding process of a single protein, so people take known structures from x-ray crystallography and look for proteins with similar known structures. For example, in Switzerland there is research proceeding on a cancer cell (melanoma). The ultimate goal is to develop a vaccine to stimulate the right T-cell to attach to and kill the cancer cell. This is being done on an HP system. The researchers came up with a small number of peptides that were used in human trials that produced eight remissions (out of 16 people), two of them permanent. This involved rational peptide design, plus peptide-based immunotherapy. This work was published in *Science.* The work was done by Vital-IT, an HPC center dedicated to the life science. It's a joint venture involving several universities, EPFL, and others.

We don't have petaflops computing yet. Even when we do, it won't be enough. However, we still can do useful things with HPC today.

Thermals are the Key (Stephen Wheat, Intel)

To us, petascale means a petaflop of performance, plus a petabyte of memory (or greater). More performance will be needed tomorrow. Petascale computing will come. According to Horst Simon, it takes about eight years for specific performance to go from the top position of the Top500 to the bottom of the Top500.

Intel is enabling this capability. We already have 65nm products today. Next year, Intel will premiere 45nm products, and so on. We expect to have 8nm products in 2017 (not yet on the official road map).

There is a trend toward many-core. We went from multiprocessor to hyperthreading to dual core and multicore. We'll see hyperthreading come back, but as we go forward it will be many-core. We have fabricated and tested 80 core already. The key to all this is what's happening with the thermals. Energy per instruction has increased. Mobile processor technology has helped with this and moved us back to where we were some time ago. We've reset our thermal story. Dempsey has 130W/socket, Woodcrest is at 80W, and the rest of the family is as low as 40W. This is because we're bringing to bear the mobile processor technology. Intel is also looking at application accelerators. There is a slew of new instructions coming, including POP count.

In 2006, we had a test chip with a teraflop on it. This yields a density of 80TFLOPS/sq ft. The test was to show that we could deploy into a given die a large number of cores interconnected for cooperation, and do this in a reliable manner. Full-speed I/O on this processor (JTAG) is relatively slow. We also need to communicate between the cores.

If we look at the standard Intel road map, reaching a petaflop with 100,000 processors in 2010 seems reasonable. This assumes sustained petaflops performance on an application, meaning tens of petaflops peak. Multithreaded cores in our quad cores could alter this. Multicores are more energy efficient, in terms of flops/watt, than just running up the clock. Regarding interchip interconnect challenges, Intel is looking at various possibilities. Today, interchip interconnect technology is all electrical. Intel is looking at silicon photonics, bringing this right onto the silicon itself. Memory bandwidth is falling off, but by bringing the memory stack onto the processor (not PIM) we can bring great bandwidth to the processor: several hundred gigabytes per second to stack DRAM, moving to terabytes per second. Two years ago, memory manufacturers weren't interested in working this problem in a five-year horizon; now, they are. Generation 2 PCI Express is due next year, with five giga-transactions per second; 0.4 bytes/flop on I/O capability is where this seems headed. We're working on additional options for managing power and efficiency. We're looking at 25–30kW/rack.

As process density increases, particle strikes on the processor itself are more likely, so you need to find a way to avoid or recover from this. This is not just for HPC, where you can run a simulation again. It's an even worse problem for Wall Street, and others, where you can't run it again. We have a number of processes to address this. We concur with Martin Walker that we'll need to move even beyond petascale, to exascale and beyond. We foresee a petaflops laptop in the 2027 time frame.

IDC Survey Highlights Related to Petascale Computing

HPC computer servers can be equipped with hundreds, thousands, or (soon) tens of thousands of powerful processors, yet few independent software vendor (ISV) applications today can take advantage of more than 128 processors. In practice:

- About 82% of application codes scale to 32-way or lower (see Table 1)
- One in four are single-CPU applications

ISV applications that are able to scale to large numbers of processors in many cases do so because the underlying problems are relatively easy to parallelize. Some of the most complex and consequential industrial problems are far more difficult to scale to large numbers of processors. In the best-case scenario, the majority of ISV codes will not be able to take full advantage of petascale systems until 3–5 years after the computers are introduced.

Changes in market dynamics, especially the adoption of clusters, have allowed most ISVs to increase revenue with only normal feature enhancements, or "technology updates." Even if an ISV had the resources for a major rewrite, the ISV might choose to spend that R&D money on other projects rather than on increasing scalability for a small part of the total market. For many applications, ISVs know how to improve scalability but have no plans to do so. For example:

- **Scaling to thousands of processors.** For 57% of codes, there are no plans to scale, scaling is not possible, or it is hard to scale (see Table 2).
- **Scaling to tens of thousands of processors.** For 74% of codes, there are no plans to scale, scaling is not possible, or it is hard to scale (see Table 3).

ISVs have a number of key factors, as shown in Table 4, which would help them scale their codes. Most important is funding (25%) and a clear business case (19%). Additional technical expertise as a whole was seen as critical by 15% (for external experts) and 18% (for internal experts) of the ISV providers.

Three-quarters of U.S. industrial firms (73%) surveyed by IDC said they could make use of a petascale computer to run today's crucial problems faster or to tackle next-generation problems of great competitive importance (see Table 5). The majority of the industrial end users said they would (83%) or might (91%) use a petascale computer to run more complex heterogeneous problems (see Table 6).

Many U.S. HPC end users have significantly larger problems that they would like to be able to solve, as shown in Table 7. One-quarter have problems that are 100 times larger than what they can solve today, and 8% have problems that are 100,000 times larger than what they can solve today.

TABLE 1

Typical Number of Processors That ISV Applications Use for Single Jobs

CPU Range	Number of Respondents	% of Respondents
1	19	24.4
2–8	25	32.1
9–32	20	25.6
33–128	9	11.5
129–1,024	4	5.1
Unlimited	1	1.3
Total	78	100.0

Source: IDC, 2007

TABLE 2

Ability of Application to Scale to Thousands of Processors

Status	Number of Respondents	% of Respondents
Already does	28	31.8
Yes, and plans in place	10	11.4
Yes, but hard	19	21.6
Yes, but no plans	20	22.7
No, not possible	11	12.5
Total	88	100.0

Source: IDC, 2007

TABLE 3

Ability of Application to Scale to Tens of Thousands of Processors

Status	Number of Respondents	% of Respondents
Already does	17	19.3
Yes, and plans in place	6	6.8
Yes, but hard	19	21.6
Yes, but no plans	34	38.6
No, not possible	12	13.6
Total	88	100.0

Source: IDC, 2007

TABLE 4

Key Factors Needed for ISVs to Improve Applications

Factor	Number of Respondents	% of Respondents
Money/investments	50	24.9
Business case/many customers	39	19.4
Internal people or experts	35	17.4
External tech expertise	30	14.9
Partnerships to share costs and risks	28	13.9
A whole new approach to their code	19	9.5
Total	201	100.0

Note: Multiple responses were allowed.

Source: IDC, 2007

TABLE 5

Use for a Petascale Computer

Q. Do you have use for a petascale computer?

Response	% of Respondents
Yes	73
No	9
Unsure	18

n = 11

Source: IDC, 2007

TABLE 6

Willingness to Run Heterogeneous Problems

Q. Would you run heterogeneous problems?

Response	% of Respondents
Yes	83
No	8
Maybe	8

n = 11

Source: IDC, 2007

TABLE 7

Size of Problems to Solve

Q. How much larger are the problems you'd like to solve?

Multiple	% of Respondents
5–10	58
100	25
10,000	8
100,000	8

n = 12

Source: IDC, 2007

FUTURE OUTLOOK

IDC studies have shown that the prospect of sustained petascale performance is exciting for HPC users, not only in government and academia but also in industry. Petascale systems are just around the corner, but the world is not ready for them yet; few software applications can efficiently exploit today's high-end HPC systems, much less petascale computers. Algorithms may need to be rewritten or, in some cases, entirely rethought. Programming, managing, providing bandwidth and storage, and running problems to completion on computers with this many (fallible) components are just some of the serious challenges. It may take a decade or more for sustained petaflop performance to become commonplace.

Many of today's leading-edge users are running problems on 5,000 or more processors at sustained speeds approaching 100 teraflops. This creates confidence that the HPC community will advance into the sustained petaflop era within five years.

ESSENTIAL GUIDANCE

Guidance for HPC Users/Buyers

- **Petascale initiatives will benefit a substantial number of users.** Although the initial crop of petascale systems and their antecedents will be few in number, many thousands of users will have access to these systems. Through the U.S. Department of Energy's INCITE program and analogous undertakings, industrial users will also gain access to these systems for their most-advanced research.

- **Petascale initiatives will intensify efforts to increase scalable application performance.** Sites that are advancing in phases toward petascale systems, often in direct or quasi competition with other sites for government funding, will be highly motivated to demonstrate impressive sustained performance gains that may also result in scientific breakthroughs. To achieve these performance gains, researchers will need to intensify their efforts to boost scalable application performance by taming large-scale, many-core parallelism. This effort will benefit the larger HPC community. Over time, it will likely also benefit the computer industry as a whole.

- **Some important applications will be left behind.** Petascale initiatives will not benefit applications that do not scale well on today's largest HPC systems unless these applications are fundamentally rewritten. Rewriting is an especially daunting proposition for some important engineering codes that required a decade or more to certify and incrementally enhance. Some poor-scaling codes may perform even worse than today on systems that turn down processor speeds to reduce power consumption.

- **Petascale systems will drive pricing to aggressive new levels.** As happens today with the largest, most prestigious procurements, vendors will propose the most aggressive pricing possible to win the contracts for petascale systems. Mindshare may be a bigger factor than profitability from the vendors' standpoint. Petascale contracts will set new per-flop pricing levels that will affect other large-scale HPC procurements. Not surprisingly, the more clusterlike the petascale system (i.e., the less custom technology it includes), the more aggressive the pricing is likely to be.

- **Some petascale developments will "trickle down" to the mainstream HPC market, while others may not.** Evolutionary improvements, especially in programming languages, will be most readily accepted by mainstream HPC users but may be inadequate for exploiting the potential of petascale systems. Conversely, revolutionary improvements (e.g., new programming languages) that greatly facilitate work on petascale systems may be rejected by the bulk of HPC users as requiring too painful a change. The larger issue is whether petascale developments will bring the government-driven high-end HPC market closer to the HPC mainstream or push these two segments further apart. This remains to be seen.

Guidance for HPC Vendors

- **"Swim" at your own risk.** Bidding and delivering on contracts for petascale systems is not for the faint of heart. Contracts for ultrascale HPC systems, today and in the future, can bring enormous mindshare and turbocharge a vendor's R&D efforts, but they may be marginally profitable at best and can distract a company from the mainstream HPC market, especially if the petascale contract requires substantial custom R&D work.

- **Monitor petascale pricing.** Whether or not they are participating directly in a petascale bid or contract, vendors should closely track petascale pricing, which will become the new benchmark for aggressive HPC pricing and could influence all large-scale procurements.

- **Selectively exploit petascale developments.** HPC vendors should watch the progress of leading-edge technologies and approaches within the petascale initiatives and look for opportunities to exploit any developments that could provide competitive advantage. Storage vendors should pay special attention to the strategies adopted by the petascale sites for storage and data management.

LEARN MORE

Related Research

Additional research from IDC in the Technical Computing Systems hardware program includes the following documents:

- *HPC User Update: Tokyo Institute of Technology, Global Scientific Information and Computing Center* (IDC #205701, May 2007)

- *Worldwide High-Performance and Technical Computing Server 2007–2011 Forecast* (IDC #206170, April 2007)

- *Steve Conway Joins IDC as Research Vice President, Technical Computing Systems* (IDC #prUS20587207, March 2007)

- *Directions 2007* (IDC #IDC_P13816, March 2007)

- *September 2006 HPC User Forum Meeting Notes* (IDC #205648, March 2007)

- *October 2006 HPC User Forum Meeting Notes* (IDC #205850, March 2007)

- *Introducing Altus 600 — A New Addition to the Penguin Family: Setting a New Price Level for HPC Clusters?* (IDC #205746, February 2007)

- *HPC Technical Computing Market Trends and Areas of Growth: Looking at Market Drivers, User Preferences, and Opportunities* (IDC #DR2007_SISEJ, February 2007)

- *IDC Analyst Briefing at the SC06 Conference* (IDC #205031, January 2007)

- *What Will Be the Next Phase Change in HPC, and Will It Require New Ways of Looking at HPC Systems?* (IDC #205025, January 2007)

- *Worldwide High-Performance Technical Computing System 2005 Vendor Shares: 2005 Market Census Results* (IDC #204866, December 2006)

- *DARPA Selects IBM for Phase III of HPCS Program* (IDC #204606, December 2006)

- *DARPA Selects Cray for Phase III of HPCS Program* (IDC #204582, December 2006)

- *HPC Market and Research Overview: 2006 and Beyond* (IDC #TB20061130, November 2006)

- *IBM Covers the Bases in Capability Computing* (IDC #lcUS20455306, November 2006)

- *DARPA to Fund Cray and IBM to Research the Next Generation of Super Computers* (IDC #lcUS20446406, November 2006)

- *Liquid Computing Announces a New Approach to Scalable Computing: Merging HPC and Telecom Architectures* (IDC #204386, November 2006)

- W*orldwide Technical Utility Grid: Initial Market Sizing and 2006–2010 Forecast* (IDC #204310, November 2006)

- *Utility Grids in Technical Computing Markets: Taxonomy, Opportunities, Drivers, and Inhibitors* (IDC #204283, November 2006)

- *Worldwide Technical Server 2006–2010 Forecast by Geographic Region* (IDC #204219, November 2006)

- *EPFL Targets First Functional Model of the Human Brain* (IDC #204194, November 2006)

- *Looking Back and Forward — What Has Changed in the Technical Computing Market?* (IDC #204099, November 2006)

- *The DoD HPCMP: A Mission-Driven, User-Centric Resource Network* (IDC #203700, October 2006)

- *Worldwide Technical Computing Systems 2006–2010 Forecast by Industry/Application Workload Segment* (IDC #203691, September 2006)

- *Star-P at Air Force Research Lab* (IDC #203273, August 2006)

- *IBM Pushes System p Performance Higher with POWER5+ Processors* (IDC #203165, August 2006)

- *Intel Brings Dual-Core Capabilities to Itanium 2 with Montecito Processor* (IDC #202865, July 2006)

- *Sun Deepens Its Investments in Next-Generation x64 Products* (IDC #202704, July 2006)

- *May 2006 HPC User Forum Meeting Notes* (IDC #202661, July 2006)

- *The Keisoku Project: Reestablishing Japan's Leadership in Supercomputing?* (IDC #202243, June 2006)

- *Storage Systems for the HPC Community: Pains and Possibilities of the "Super-Size Me" World* (IDC #202185, June 2006)

- *Worldwide Technical Computing Systems 2006–2010 Forecast* (IDC #201733, May 2006)

- *Cray Launches Adaptive Supercomputing as a New Technology Approach* (IDC #201017, March 2006)

- *Definitions for HPC Industry/Application Workload Segments and 2005 Technical Computing Market Results* (IDC #35055, March 2006)

- *Project Fastball, IBM's General Parallel File System Addressing the FLOPS to GBps Challenge* (IDC #35027, March 2006)

- *New Supercomputer Design Approach: Sun and Luxtera Partner on DARPA HPCS Program* (IDC #34861, February 2006)

- *Grid-Based Advanced Storage, Motivations, and Storage Infrastructure Development Stages, 2006* (IDC #34795, February 2006)

Copyright Notice

Published Under Services: HPC User Forum; Technical Computing Systems