LBNL-56545

Science-Driven System Architecture: A New Process for Leadership Class Computing

Horst Simon*[1,2,3], William Kramer[2], William Saphir[2], John Shalf[3], David Bailey[2],
Leonid Oliker[3], Michael Banda[1], C. William McCurdy[4], John Hules[1], Andrew Canning[3],
Marc Day[3], Philip Colella[3], David Serafini[3], Michael Wehner[3], Peter Nugent[3]

[1]Computing Sciences Directorate
[2]National Energy Research Scientific Computing Center (NERSC) Division
[3]Computational Research Division
[4]Chemical Sciences Division
Lawrence Berkeley National Laboratory
One Cyclotron Road, MS 50B4230
Berkeley, California 94720

*Email: hdsimon@lbl.gov

October 19, 2004

This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research of the U.S. Department of Energy under Contract No. DE-AC 03-76SF00098.

Science-Driven System Architecture:
A New Process for Leadership Class Computing

Abstract: Over the past several years, computational scientists have observed a frustrating trend of stagnating application performance despite dramatic increases in peak performance of high performance computers. In 2002, researchers at Lawrence Berkeley National Laboratory, Argonne National Laboratory, and IBM proposed a new process to reverse this situation [1]. This strategy is based on new types of development partnerships with computer vendors based on the concept of *science-driven computer system design*. This strategy will engage applications scientists well before an architecture is available for commercialization. The process is already producing results, and has further potential for dramatically improving system efficiency. This paper documents the progress to date and the potential for future benefits. An example of this process is discussed, using IBM Power architecture with a computer architecture design that can lead to a sustained performance of 50 to 100 Tflop/s on a broad spectrum of applications in 2006 for a reasonable cost. This partnership will establish a collaborative approach to modifying computer architecture to enable heretofore unrealized achievements in computer capability-limited fields such as nanoscience, combustion modeling, fusion, climate modeling, and astrophysics.

1. STRATEGIC APPROACH TO A LEADERSHIP COMPUTING TECHNOLOGY

This paper presents a plan that will maximize the return on the U.S. government’s investment in high performance computing, initiate a new wave of scientific discovery, and enable the solution of problems of national and global importance. Our vision is guided by the following analysis:

1. Government investments, such as the U.S. Department of Energy’s (DOE) Leadership Class Computing project, must lead to widely deployable new technology for high-end scientific computing. If such an investment leads merely to a series of experiments or the purchase of a single machine, it will not have a lasting impact.

2. The technology needed will not spontaneously appear on the market. By taking a passive approach that relies on evaluating and procuring existing vendor offerings, the high performance computing community has ceded leadership to other requirements that are increasingly incompatible with the needs of high-end computing.

3. Several national panels have concluded that the rules of engagement between the scientific community and the American computer industry must be revised [2,3,4]. Scientific applications must directly influence machine design in a repeating cycle: (a) scientific applications input, (b) computer design with increased performance, (c) deployment and delivery to the scientific community, (d) repeat.

4. Successfully changing the rules of engagement requires partnerships with the American computer companies with the resources and the track records of research and development in high performance computing. To justify the necessary commitments, there must be a national consortium of laboratories, computing facilities, universities and researchers equally committed to changing the future of the computing capability available to the scientific community.

5. Evaluating a representative array of applications to establish precisely their algorithmic characteristics provides a clear understanding of the limitations of current high-end systems of all designs, from clusters to vector computers.

6. Over the past two years, the Blue Planet partnership led by Lawrence Berkeley National Laboratory (Berkeley Lab) has worked closely with IBM to design a machine that better meets the needs of scientific applications. The goals and methodology of this partnership were validated by the successful design and implementation of the $100+ million ASC Purple machine at Lawrence Livermore National Laboratory, based on the Blue Planet node design. It is the first success of this science-driven design process.

2. SCIENTIFIC APPLICATIONS AND UNDERLYING ALGORITHMS DRIVE ARCHITECTURAL DESIGN

The central goal of this strategy is to deliver new scientific results on computations of a scale that greatly exceeds what is possible on current systems. It is possible, within a reasonable cost, to create by 2006 a system with sustained performance rates of 50 to 100 Tflop/s on scientific applications of national and global importance for an acceptable cost. We have identified the following example application classes as being ripe for breakthrough science using very high-end computing, and relevant to some of the most important national objectives: nanoscience, combustion modeling, fusion energy simulations, climate modeling, and astrophysics. Table 1 summarizes the goals, computational methods, and example applications of each science area.

The most effective approach to designing a computer architecture that can meet these scientific needs is to analyze the underlying algorithms of these applications, and then, working in partnership with vendors, design a system targeted to these algorithms.

Table 1
Science breakthroughs enabled by leadership computing capability

Science Areas	**Goals**	**Computational Methods**	**Examples of Breakthrough Applications**
Nanoscience	Simulate the synthesis and predict the properties of multi-component nanosystems	Quantum molecular dynamics Quantum Monte Carlo Iterative eigensolvers Dense linear algebra Parallel 3D FFTs	Simulate nanostructures with hundreds to thousands of atoms, as well as transport and optical properties and other parameters
Combustion Modeling	Predict combustion processes to provide efficient, clean and sustainable energy	Explicit finite difference Implicit finite difference Zero-dimensional physics Adaptive mesh refinement Lagrangian particle methods	Simulate laboratory-scale flames with high-fidelity representations of governing physical processes
Fusion Energy	Understand high-energy density plasmas and develop an integrated simulation of a fusion reactor	Multi-physics, multi-scale Particle methods Regular & irregular access Nonlinear solvers Adaptive mesh refinement	Simulate the ITER reactor
Climate Modeling	Accurately detect and attribute climate change, predict future climate, and engineer mitigation strategies	Finite difference methods FFTs Regular & irregular access Simulation ensembles	Perform a full ocean/atmosphere climate model with 0.125 degree spacing, with an ensemble of 8–10 runs
Astrophysics	Determine through simulation and analysis of observational data the origin, evolution, and fate of the universe; the nature of matter and energy; galaxy and stellar evolution	Multi-physics, multi-scale Dense linear algebra Parallel 3D FFTs Spherical transforms Particle methods Adaptive mesh refinement	Simulate the explosion of a supernova with a full 3D model

Table 2
Algorithm requirements

Science Areas	**Multi-physics & multi-scale**	**Dense linear algebra**	**FFTs**	**Particle methods**	**AMR**	**Data parallelism**	**Irregular control flow**
Nanoscience	X	X	X	X		X	X
Combustion	X			X	X	X	X
Fusion	X	X		X	X	X	X
Climate	X		X		X	X	X
Astrophysics	X	X	X	X	X	X	X

From this list of important scientific applications and underlying algorithms, several themes can be derived that drive the choice of a large-scale scientific computer system: (1) multi-physics, multi-scale calculations; (2) limited concurrency, requiring strong single-CPU performance; (3) reliance on key library routines such as ScaLAPACK and FFTs; (4) the use of particle methods, with couplings to grid-based methods that lead to large-scale interaction of two regular, but unaligned, data structures; (5) widespread usage of finite difference computations, requiring good performance on fairly regular accesses in multiple dimensions and high main memory bandwidth; (6) an increasing usage of sparse, unstructured, and adaptive mesh refinement

(AMR) methods, which entail some irregular control sequences that do not perform well on vector systems; (7) ubiquitous data parallelism providing the opportunity for fine-grained operation concurrency; and (8) irregular control flow inhibiting fine-grained symmetric operation concurrency. Table 2 presents a qualitative summary of this information:

The characteristics summarized here point to the need for a flexible system — one that can perform well both on random memory access calculations as well as regular memory access problems and that combines strong single-node performance (to minimize the required concurrency in the application) and a powerful system-scale network.

Of the two principal classes of high performance systems in widespread usage — superscalar systems and vector systems — each has a different set of advantages and disadvantages for these applications. Superscalar, cache-memory-based systems tend to do well on problems with spatial and temporal data regularity. These systems also do relatively well on irregularly structured algorithms and codes with heavy usage of conditional branching in inner loops. However, many cache-based systems feature low or oversubscribed main memory bandwidth, since they are not primarily designed for scientific computation. Thus, codes with low computational intensity typically do not perform well on these architectures.

Vector systems exploit regularities in the computational structure to expedite uniform operations on dependence-free data. Many scientific codes are characterized by predictable fine-grained data-parallelism and thus allow vectorization. However, vector systems tend to do poorly on codes with irregularly structured computations. These codes are characterized by irregular control flow, intensive scalar operations, and significant conditional branching — operations that inhibit vectorization. Performance on vector architectures degrades significantly even when a small fraction of the work is non-vectorizable, as described by Amdahl’s Law. This is particularly true for newly emerging multi-method, multi-physics codes that can only leverage vectorization for a subset of the numerical components.

These considerations suggest that an architecture that combines the best features of high-end superscalar and vector systems would be best suited for the workload that we project for future high-end computing of national and global importance.

3. A SCIENCE-DRIVEN SYSTEM ARCHITECTURE (SDSA)

Applications scientists have been frustrated by a trend of stagnating application performance despite dramatic increases in claimed peak performance of high performance computing (HPC) systems. This trend has been widely attributed to the use of commodity components whose architectural designs are unbalanced and inefficient for large-scale scientific computations. It was assumed that the ever-increasing gap between theoretical peak and sustained performance was unavoidable. However, recent results [12] from the Earth Simulator (ES) in Japan clearly demonstrate that a close collaboration with a vendor to develop a science-driven architectural solution can produce a system that achieves a significant fraction of peak performance for critical scientific applications. The key to the ES success was the long-term collaborative development strategy between the scientists of JAMSTEC (Japan Marine Science and Technology Center) and NEC Corporation.

Realizing that effective large-scale system performance cannot be achieved without a sustained focus on application-specific architectural development, Berkeley Lab and IBM have led a collaboration since 2002 that involves extensive interactions between domain scientists, mathematicians, computer experts, as well as leading members of IBM's research and product development teams. The goal of this effort is to change IBM's architectural roadmap to improve system balance and to add key architectural features that address the requirements of demanding leadership-class applications — ultimately leading to a sustained Pflop/s system for scientific discovery. The first product of this multi-year effort has been a redesigned Power5-based HPC system known as Blue Planet [1] and a set of architectural extensions referred to as ViVA (Virtual Vector Architecture). This collaboration has already had a dramatic impact on the architectural design of the ASC Purple system [5].

3.1 Leadership Computing Systems

The goal has to be to build an architecture balanced for leadership-class science requirements as described above, which presents the computational science applications that will be of critical importance to U.S. government-sponsored research in 2006 and are able to take advantage of an ultra-scale computing system.

The key science requirements for leadership-class computing can be distilled into three main system features: processor performance, interconnect performance, and software. Processors should have excellent sustained single-node performance across the spectrum of applications. The interconnect should provide high per-link performance (both latency and bandwidth) as well as high bisection bandwidth. Effective system utilization requires proven system software scalability and optimized numerical libraries.

The goal of SDSA is to enable new science discoveries. Implicit in this is a requirement for real working systems. Our plans take into account both credibility and risk in vendor roadmaps for architecture development.

3.2 Memory Contention Considerations with Multiple Processes per Node

An important concern with the use of symmetric multi-processor (SMP) systems as building blocks of large computers is memory contention within an SMP node. The per-processor performance of parallel applications is typically less than that of corresponding serial applications because of parallel inefficiencies (e.g., Amdahl's law), but also because of memory contention within a node. This has been a particular concern on IBM Power4 systems, which are based on a dual-core design in which two processors share the same interface to main memory, effectively halving the bandwidth. Power4 systems therefore perform particularly poorly on parallel applications — more poorly than one would expect based on single-processor benchmarks.

An estimate of the effect of memory contention can be obtained by running multiple simultaneous copies of a serial benchmark, and comparing their performance to that of a single copy on an unloaded machine. If there is no contention, performance is the same. We define a benchmark *NPB, which consists of running N-simultaneous copies of each NPB benchmark application [6] on an N-processor system. This can be seen for the Power4 in Table 3. This result is consistent with the earlier statement that increasing peak performance without increasing memory bandwidth typically improves performance by half the increase in peak. An analysis based on this rule of thumb predicts 6.9% efficiency.

Table 3
Effect of Memory Contention on the Power4

	Power4 (single copy)	**Power4 (8 copies in 8-processor partition)**
NAS Codes (Mflop/s)		
BT	827	682
CG	113	56
FT	514	345
LU	554	357
MG	430	333
SP	426	319
Average	477	**349**
% peak	9.2%	**6.7%**

The Blue Planet systems minimize the effect of memory contention through the following mechanisms:

- Dedicated memory system for each processor, including on-chip memory controller.
- “Single core” design. Many systems are now designed with two processor and even four cores on a chip. These processors share cache bandwidth and main memory bandwidth, effectively halving or quartering the memory bandwidth per processor.
- Small node design. By having fewer processors in an SMP, the memory interconnect is greatly simplified.
- Processor affinity. The scheduling system ensures that process memory is local to the processor on which the process is running.

We expect the effect of memory contention to be minimal in both the future Power systems. The Blue Planet design is incorporated into the new generation of IBM Power microprocessors that are the building blocks of future system configurations. These processors break the memory bandwidth bottleneck, reversing the recent trend towards architectures poorly balanced for scientific computations. The Blue Planet design improves the original power roadmap in several key respects: a dramatic improvement in memory bandwidth; 70% reduction in memory latency; eight-fold improvement in interconnect bandwidth per processor; and ViVA Virtual Processor extensions, which allow all eight processors within a node to be effectively utilized as a single virtual processor.

The Blue Planet node is a Power5 system with eight single-core CPUs per node. It is expected that average application performance will be 20% of peak, with several key applications well above that range. Key innovations in the Blue Planet architecture allow it to obtain a much higher percentage of peak performance than its predecessors, such as the Power4. These include:

- **High-memory bandwidth per processor,** including a memory architecture that achieves much higher bytes/flop, comparable to vector architectures.
- **"Single core" node design.** IBM's original roadmap called for two processor cores on a single chip to share the same memory system. Going to a single core design effectively doubles the memory bandwidth per processor.
- **Small node design.** With eight-processor nodes, it is possible to put the processors closer to memory, reducing memory latency. Furthermore, by reducing the number of processors per node, effective network bandwidth per processor exceeds IBM's original 32- or 64-way SMP roadmap.
- **ViVA Virtual Processing** that allows the eight processors in a node to be treated as a single processor with peak performance of 60+ Gigaflop/s. Codes that benefit from Cray X1 multistreaming, for example, will directly benefit from ViVA capabilities.

3.3 Building on the Blue Planet Collaboration: Addressing the Memory Bandwidth Bottleneck

Now is the time to look forward to new additions and accelerators that will lead to a set of enhancements known as ViVA-2. Science application collaborators are participating in the system design and refinement process. Berkeley Lab, Lawrence Livermore National Laboratory, and IBM hold quarterly meetings to review progress, create ideas, and refine the design decisions. These meetings integrate application scientists, system designers, HPC performance experts, and computer scientists. This community approach of directly engaging vendors in the collaborative process of designing leadership HPC systems was laid out by the High End Computing Revitalization Task Force (HECRTF) [2,4] and the DOE SCaLeS Workshop [3], and was demonstrated successfully by the Earth Simulator, the initial Blue Planet effort, and the Red Storm effort [7].

There is an opportunity to incorporate the ViVA-2 scientific enhancement technology into future Power processor design. During FY04 and FY05, IBM and the partners will evaluate various enhancements to the future processor, node, and interconnect design, including assisted processing capabilities and their impact on the associated components (e.g., compilers, libraries, tools, etc.). The collaborators will advise IBM on how to incorporate the resulting technology into subsequent systems to maximize its impact on scientific discovery.

3.3.1 ViVA Design Targets

ViVA and ViVA-2 are specialized enhancements to the Power architecture designed to significantly improve sustained performance on a wide range of scientific applications. ViVA is a compiler-supported programming model that combines processors to form more powerful virtual processors by making use of fast barrier synchronization technology available in Power5 and Power6 processors.

ViVA-2 is envisioned as a set of extensions to the Power6 architecture that will accelerate scientific applications by supporting deeper pipelining of memory requests in order to hide memory latencies. These extensions will improve the efficiency of memory accesses on both vectorizable and non-vectorizable codes. ViVA-2 is superior to strictly vector designs because it offers the flexibility of achieving high performance on non-vectorizable algorithms using state-of-the-art superscalar technology, while efficiently processing data-parallel code segments that are amenable to vectorization. These enhancements address a variety of scalar memory performance degradations often attributed to irregularities in the data-access patterns. Examples include ineffective hardware prefetching, load/store instruction issue-rate limitations, and wasted bandwidth due to partially used cache lines.

3.3.2 ViVA: Virtual Processors

The ability to combine CPUs to form more powerful virtual processors reduces coarse-grained parallelism requirements, and allows a wider spectrum of applications to effectively utilize the underlying computational resources. The ViVA Virtual Processing extensions enable this architectural enhancement through the tight synchronization of an eight-way CPU node. IBM originally developed the fast synchronization hardware for the earlier generation Power3 processor variants used in Hitachi’s innovative SR-8000 system [8,9,10]. This feature is similar to what Cray refers to as “multistreaming” on its X1 system, where four independent 3.2 Gflop/s vector processors (SSPs) are combined using fast synchronization hardware and compiler technology to form the 12.8 Gflop/s multistreaming processor (MSP). The Power5 node could use ViVA fast synchronization hardware to combine eight Power5 cores to form a single processor. Codes that benefit from Cray multistreaming will also benefit from ViVA. This feature will also improve the efficiency of OpenMP-enabled codes such as the latest generation of the Community Climate System Model (CCSM3). The ability to combine CPUs to form more powerful virtual processors reduces the apparent parallelism of the resulting system. This approach offers distinct advantages for a number of codes that have limited ability to manage massive parallelism, such as climate, sparse matrix methods, and adaptive mesh refinement (AMR) methods.

AMR, for instance, makes use of dynamically adapting hierarchies of meshes in order to follow shock fronts and other moving features that require additional refinement. AMR codes must therefore continuously rebalance the computational load as the meshes adapt to changing conditions in the simulation. The complexity of this load-balancing problem increases dramatically as the number of processors in the system increases. The ViVA virtual processors enable the AMR simulation to treat a 4,096-way supercomputing system as one that contains 512 much faster processors. By keeping scalability requirements to a manageable level, future systems will be applicable to a wider variety of application codes than could be supported using less powerful commodity processors.

3.3.3 ViVA-2: Application Accelerator

On March 31, 2004, IBM announced “plans to openly collaborate and build a community of innovation around its Power microprocessor architecture used in a vast range of products from the world’s most powerful enterprise systems and supercomputers to games and embedded devices” [11]. One of the features that might be added to this new chip platform is *application accelerators* — additional hardware collocated with the CPU on the chip to accelerate particular application-specific or domain-specific features. For instance, one potential use of this capability is a TCP protocol accelerator that is implemented entirely in hardware. A ViVA-2 scientific application accelerator will offer significant improvements in efficiency for a wide variety of scientific applications, as described below.

ViVA-2 is a science-driven application accelerator, targeting bottlenecks that degrade scientific code performance. Examples of performance limitations that ViVA-2 may potentially address include:

- **Irregular access patterns:** The Power architecture is optimized for strided and regular access patterns. The memory subsystem's automatic hardware prefetch streams provide deep pipelining of memory accesses that hides memory latency. However, hardware prefetching only recognizes regular memory access patterns and is not designed for irregular memory access patterns.
- **High load/store issue rates:** Aggressive issue of data prefetch instructions can fill the memory fetch queues. It is unfeasible to employ conditional logic to prevent redundant fetches of the same cache-line.
- **Low cache line utilization:** Sparse and strided operations may use as few as one 8-byte word in a 128-byte cache line, needlessly consuming memory bandwidth. This situation can arise in many scientific applications, including multigrid solvers and sparse matrix computations.

Some technology enhancements being considered for ViVA-2 to address these limitations include:

- Instruction set or auxiliary register set extensions that support efficient prefetch generation for moderately irregular data access.
- Instruction set extensions that support sparse, noncache-resident data loads. This is needed for strided accesses for multigrid methods, as well as for indexed-irregular loads required for sparse matrix methods.
- Additional registers for software pipelining of larger loop bodies. This decreases the need for loop splitting to control register spilling and thereby reduces the memory bandwidth requirements.
- Instruction set extensions that allow the CPU to initiate many dense or indexed/sparse loads using a single instruction in order to reduce load/store unit stalls. This must be done in conjunction with an increased number of memory request queue entries.
- Proper compiler support will be a critical component of these enhancements.

3.3.4 Refinements and Beyond

The ViVA-2 extensions are intended to benefit scientific codes that are characterized by the kind of predictable data parallelism that is typically associated with vector processing. Since the superscalar core performs all computations on operands fetched by ViVA-2, its advantages are available even for non-vectorizable algorithms. The collaboration will investigate design tradeoffs and define the final ViVA-2 architecture.

Additionally, custom hardware accelerators in network adaptors can be envisioned in the 2007–2008 time frame to efficiently support collective operations and global barrier synchronizations. Specialized hardware support for global operations would result in significant reduction in latency overhead. These interconnect enhancements allow a system to efficiently handle state-of-the-art scientific applications with fast global synchronization requirements in a scalable fashion.

The current roadmap for SDSA advancements is depicted in Figure 1. Based on the expertise gained from system design, and the extensive application knowledge represented by the application partners, it is possible to leverage the collaborative effort to assess the most effective and timely system options for a sustained Pflop/s system.

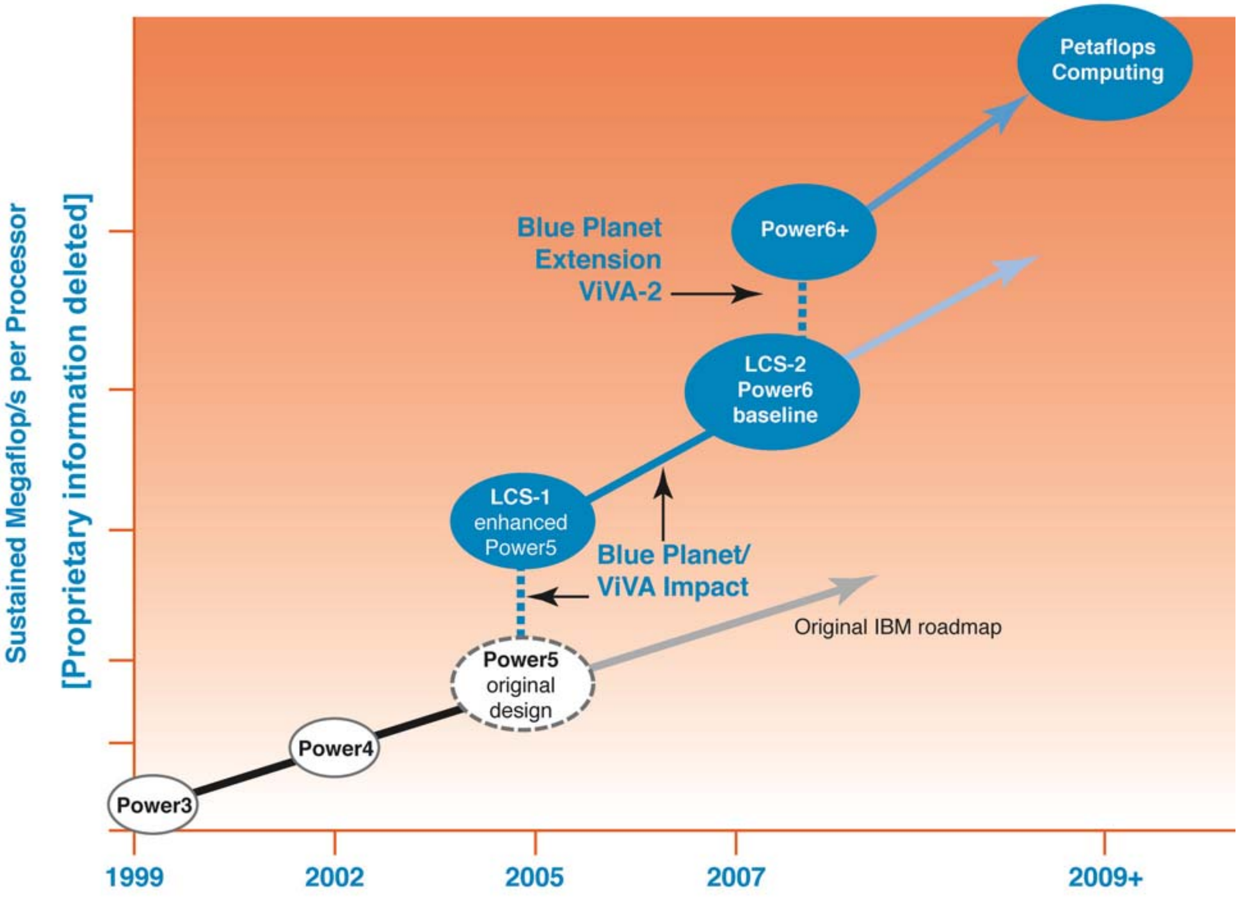

Figure 1. Science-driven architecture advancements.

4. BUILDING A NATIONAL LEADERSHIP COMPUTING CONSORTIUM

In order to fully engage the community in the SDSA process, collaborations with computational scientists in universities, research labs, and industry need to combine a patchwork of a nationwide computing resources into a common fabric serving the needs of the U.S. scientific research community across all branches of the U.S. government. The national computing fabric will lower barriers to user migration and resource sharing between facilities comprising our national computational infrastructure. A national Leadership Computing Consortium (LCC) needs to be established, which would include the leading high-end computing centers of the nation.

4.1 Functions of the Leadership Computing Consortium

The LCC is envisioned to have two functions:

- **Technology development:** LCC will be the main vehicle for implementing the SDSA development. LCC will engage major vendor partners in an ongoing dialogue of science-driven architecture development.
- **National facility operations:** LCC will be the vehicle to establish close connections and strategic collaborations with computer science programs and facilities funded by the DOE Office of Science, the National Nuclear Security Administration, the National Science Foundation, and the National Aeronautics and Space Administration, as well as universities.

Recognizing that the typical workload on a supercomputer follows a power-law-like curve of job sizes in order to satisfy users' development, data analysis, and post-processing needs, LCC members will establish a national computing fabric that will lower barriers to user migration and resource sharing between computing facilities. In particular, LCC sites will define systems that support coordinated access to accounts, federate archival storage devices across sites, establish a federated parallel file system (WAN-GPFS) that spans the U.S., and tie all of these services together with high performance network services to move data between all of these components. The goal is seamless migration across the U.S. computational infrastructure. LCC sites will also collaborate to jointly develop system documentation, mutual training, and support mechanisms, to conduct detailed performance analysis of applications, and to contribute to the direction of future systems development, drawing on their years of combined experience supporting a national user community. This collaboration will greatly reduce duplication of effort and free up resources to ensure that the U.S. supercomputing infrastructure will provide the highest quality platform for advanced scientific applications.

4.2 Leadership Computing Applications Teams

Computational science applications areas that require a leadership-class computing capability to make major computational advances include nanoscience, combustion, fusion, climate, life sciences, and astrophysics. In each of these applications, project teams must be assembled who will collaborate with national facilities and the LCC to accomplish their computational goals. In each team, one or more computational scientists will serve as points of contact, working with the applications scientists and developing a deep understanding of the algorithmic techniques and computational requirements of the applications areas. The points of contact will then communicate these requirements to the leadership computing facilities and to the vendor partners. This input from the science community is an important element in the process of driving future technology developments.

5. SUMMARY

In this paper we explain a new way to engage the science community and computer vendors in developing systems that are more effective for science, yet still cost effective overall. This we call the Science-Driven System Architecture process. This process replaces the traditional approach of letting vendors build systems designed for purposes other than science, and then evaluating and selecting the best from a set of poor choices. We show that this process is effective in producing significantly better-performing systems, with the first success demonstrated by the Blue Planet nodes being deployed as part of the ASC Purple systems. The long-term success of the SDSA process requires a commitment from both the science community and the vendors over a sustained period of time.

We show that high-performance systems of the future have to be balanced in many ways since the scientific applications of the future will combine many different methods. There is no longer a single method that dominates in any one area. We also discuss new ideas for enhancing current commodity processors, including designing nodes to maximize memory bandwidth, not peak flop/s. Another new idea, ViVA, is adding low-cost vector accelerators to commodity CPUs in order to further improve performance of codes that are characterized by predictable fine-grained data-parallelism and thus allow vectorization.

REFERENCES

1. C. W. McCurdy et al., *Creating Science-Driven Computer Architecture: A New Path to Scientific Leadership,* Lawrence Berkeley National Laboratory report LBNL/PUB-5483, October 2002; http://www.nersc.gov/news/blueplanet.html.

2. *Workshop on the Roadmap for the Revitalization of High-End Computing,* D. A. Reed, ed., Computing Research Association, Washington, D.C., 2003.

3. *A Science-Based Case for Large-Scale Simulation,* P. Colella, T. H. Dunning, Jr., W. D. Gropp, and D. E. Keyes, eds., DOE Office of Science, Washington, D.C., 2003.

4. J. Grosh, A. Laub, D. Nelson, S. Szykman, et al., *Federal Plan for High-End Computing: Report of the High-End Computing Revitalization Task Force,* National Coordination Office for Information Technology Research and Development, Arlington, VA, 2004.

5. *Facts on ASCI Purple,* Lawrence Livermore National Laboratory report UCRL-TB-150327, 2002; http://www.sandia.gov/supercomp/sc2002/flyers/SC02ASCIPurplev4.pdf.

6. D. H. Bailey et al., The NAS Parallel Benchmarks, *Intl. Journal of Supercomputer Applications,* vol. 5, no. 3, pp. 66–73, 1991.

7. *Red Storm System Raises Bar on Supercomputer Scalability,* Cray Inc., Seattle, WA, 2003; http://www.cray.com/company/RedStorm_flyer.pdf.

8. M. Brehm et al., Pseudo vectorization, SMP, and message passing on the Hitachi SR8000-F1, *Euro-Par 2000: Parallel Processing: 6th International Euro-Par Conference,* Munich, Germany, Aug. /Sept. 2000.

9. H. Nishiyama et al., Pseudo-vectorizing compiler for the SR8000, *Euro-Par 2000: Parallel Processing: 6th International Euro-Par Conference,* Munich, Germany, Aug. /Sept. 2000.

10. R. Bader et al., TeraFlops computing with the Hitachi SR8000-F1, *High Performance Computing in Science and Engineering. Transactions of the First Joint HLRB and KONWIHR Status and Result Workshop*, 2002.

11. IBM plans industry’s first openly customizable microprocessor, press release, IBM, New York, March 31, 2004, http://www-1.ibm.com/press/PressServletForm.wss.

12. S. Shingu, H. Takahara, H. Fuchigami, M. Yamada, Y. Tsuda, W. Ohfuchi, Y. Sasaki, K. Kobayashi, T. Hagiwara, S. Habata, M. Yokokawa, H. Itoh, K. Otsuka, A 26.58 Tflops Global Atmospheric Simulation with the Spectral Transform Method on the Earth Simulator, Proceedings of the IEEE/ACM SC2002 Conference, November 16–22, 2002, Baltimore, Maryland, page 52.

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or The Regents of the University of California.