

to appear in the proceedings of Frontiers '92: The Fourth Symposium on the Frontiers of Massively Parallel Computation, Oct. 1992, co-sponsored by the IEEE Computer Society and NASA

## Summary of the Report of the NSF-Sponsored Purdue Workshop on Grand Challenges in Computer Architecture for the Support of High Performance Computing

H. J. Siegel and S. Abraham, Purdue Univ., Workshop Co-Chairs

B. Bain, Intel Corp. SSD  
T. L. Casavant, Univ. of Iowa  
J. B. Dennis, M.I.T.  
T-Y. Feng, Penn. State Univ.  
A. Huang, AT&T Bell Labs.  
J. R. Jump, Rice Univ.  
A. J. Smith, UC-Berkeley  
L. Snyder, Univ. of Washington  
R. Tuck, MasPar Computer Corp.

K. E. Batcher, Kent State Univ.  
D. DeGroot, Texas Instruments  
D. C. Douglas, Thinking Machines Corp.  
J. R. Goodman, Univ. of Wisconsin  
H. F. Jordan, Univ. of Colorado  
Y. N. Patt, Univ. of Michigan  
J. E. Smith, Cray Research, Inc.  
H. S. Stone, IBM T.J. Watson Research Ctr.  
B. W. Wah, Univ. of Illinois

### Abstract

*The Purdue Workshop on Grand Challenges in Computer Architecture for the Support of High Performance Computing was held at Purdue University on December 12 and 13, 1991. The workshop was sponsored by the National Science Foundation to identify critical research topics in computer architecture as they relate to high performance computing. Following a wide-ranging discussion of the computational characteristics and requirements of the grand challenge applications, the workshop identified four major computer architecture grand challenges as crucial to advancing the state of the art of high performance computation in the coming decade. These are:*

- (1) idealized parallel computer models,*
- (2) usable peta-ops ( $10^{15}$  ops) performance,*
- (3) computers in an era of HDTV, gigabyte networks, and visualization,*
- (4) infrastructure for prototyping architectures.*

*This invited paper overviews some of the demands of the grand challenge applications and summarizes the above four grand challenges for computer architecture.*

---

This work was supported by the National Science Foundation Division of Computer and Computation Research Systems Program under grant number CCR-9200735.

### 1: Introduction

#### 1.1: Origin of the Workshop

“Grand Challenges: High Performance Computing and Communications” is the title of the widely distributed “blue book” [2] that describes the United States Federal High Performance Computing and Communications (HPCC) program. The goal of this program is “to accelerate significantly the commercial availability and utilization of the next generation of high performance computers and networks.” The booklet presents a set of “grand challenge problems” — applications that need the major gain in processing power that the HPCC initiative is expected to provide. These problems are characterized by massive data sets, complex operations, and irregular data structures that exceed the limits of current supercomputers and programming paradigms.

However, the blue book does not explicitly explore what developments in computer architecture are needed to support the grand challenge applications. This topic arose in discussions between Dr. Zeke Zalcstein of the National Science Foundation and Prof. H. J. Siegel of Purdue University. Dr. Zalcstein felt it was important to explore, in a workshop environment, what the relevant key issues in computer architecture are.

“The Purdue Workshop on Grand Challenges in Computer Architecture for the Support of High

Performance Computing" was held at Purdue University on December 12 and 13, 1991 to identify critical research topics in computer architecture as they relate to high performance computing. The workshop was sponsored by the Computer Systems Program of the Division of Computer and Computation Research at the National Science Foundation and brought together a small but diverse group of computer architecture researchers. Professors H. J. Siegel and Seth Abraham, both of the School of Electrical Engineering at Purdue University, were the workshop co-chairs, and Dr. Zeke Zalcstein was the NSF liaison.

The results of the workshop are reported in [4]. This invited paper, to be presented by Professor Jack B. Dennis at the symposium, is a summary of those results.

## 1.2: The Workshop Charter

To fully appreciate the architectural grand challenges that were the "output" of this meeting, it is instructive to keep in mind the "input" to which the group was responding. To clarify this, the plan for the workshop is quoted from the invitation sent to the participants. It was as follows.

There is a desire to advance significantly the state of the art of high performance computing. The grand challenges for high performance computing have been discussed in terms of the applications that can make use of the computing power to be made available. The focus of this workshop can be stated succinctly as follows: what are the grand challenges facing computer architecture that must be met to build high performance computers? The workshop will focus on the design and construction of the hardware architecture. While the hardware cannot be considered in isolation, application and system software issues are beyond the scope of this workshop. This workshop will consider software aspects and application characteristics only where there is an impact on the hardware design.

The goals of the workshop are to list, characterize, categorize, assess the difficulty of, and interrelate these "grand challenges" for computer architecture for the support of high performance computing. This meeting will indicate the areas of computer architecture research that the participants feel are most important and should receive the most attention.

Computer architects from both academia and industry were invited to the workshop. Some invitees could not attend due to scheduling conflicts. Those who attended are the co-authors of this paper (see Appendix for complete names and addresses).

While it was recognized that technology and software are important considerations and are strongly interrelated with architecture, the group's instructions from NSF were to focus mainly on the hardware architecture organization. Such a focus was necessary due to the limited time duration of the workshop.

## 1.3: The Workshop Results

The report of the workshop results [4], summarized in this paper, presents four architectural grand challenges whose achievements would make significant advances towards the goals of high performance computing and communication. These four challenges were distilled from a great variety of views expressed by individual participants and the report is closer to a union of those views than an intersection.

The workshop co-chairs assembled the report from draft material contributed by all workshop participants. This report attempts to reflect fairly the (sometimes conflicting) views expressed, while maintaining a coherent style.

Section 2 of this paper sets the stage for the group's selection of grand challenges in computer architecture by discussing the demands on architecture implied by the U.S. national commitment to supporting the solution of the grand challenge problems. The grand challenges in computer architecture the group felt were most important are explained in Section 3.

## 2: Grand Challenge Application Problems and Computer Architecture

### 2.1: Grand Challenge Application Problems

The U. S. Committee on Physical, Mathematical, and Engineering Sciences has identified a set of "grand challenge problems" that set a goal for HPCC initiative, now funded by the U. S. Congress through several agencies. The grand challenge application problems concern pressing issues of human welfare on planet Earth, and problems at the exciting frontiers of science that may open doors to better living for future generations.

The blue booklet published by the committee [2] lists ten areas as posing "problems whose solution is critical to national needs."

Climate Modeling	Quantum Chromodynamics
Fluid Turbulence	Semiconductor Modeling
Pollution Dispersion	Superconductor Modeling
Human Genome	Combustion Systems
Ocean Circulation	Vision and Cognition

It is estimated that a serious attack on any of these problems will require computer performance in excess of one trillion floating point operations per second (one teraflops).

The grand challenge problems are awesome in their computational requirements. Consider, for example, the problem of modeling the weather. In five years time, data collection facilities will be in place to define detailed atmospheric structures and permit significant advances in forecasting capabilities. However, today's most powerful supercomputers cannot meet the computational requirements. The goal of improving atmospheric modeling resolution to a five-kilometer scale and providing timely results is believed to require 20 teraflops of performance.

## 2.2: One Teraflops and Beyond

Even though substantial progress remains to be achieved in the uniprocessor arena, due to the inherent limitations of uniprocessor technology it is assumed that high performance computing will employ parallel systems. Currently available massively parallel computers of practical size and cost perform at most at the level of hundreds of gigaflops ( $10^9$  floating point operations per second). To produce practical massively parallel computers having at least one teraflops ( $10^{12}$  floating point-operations per second) performance, only engineering effort is needed to fully utilize existing, demonstrated technology. These teraflops computers can become available in a few years; however, there is much debate whether such machines can be produced at a low enough cost that will make them commercially viable for a large customer base. Furthermore, there is a need for environments that will allow application programmers to realize a significant fraction of such a machine's peak speed.

Providing performance significantly beyond teraflops will require major innovations in computer hardware architecture, packaging, and device technology. Optical technology may offer a breakthrough in performance, but it will require a radical rethinking of computer structure and how the technology can support appropriate models of computation. Of course, cost and usability concerns remain.

Many supporting and related areas must also be developed. Improvement is needed in the infrastructure that supports the design, prototyping, and construction of advanced computer hardware. This is also true for high performance peripherals to match

the capabilities of the processors. Reliability and fault tolerance will become increasingly serious issues as high performance machines become incorporated into networks, begin to handle communications-intensive information processing, and satisfy real-time demands. Programmability and usability must be facilitated by new programming models and environments.

## 2.3: Effective Use of Potential Performance

Achieving ever greater levels of peak performance is not the only challenge resulting from the goals of high performance computing; a significant challenge is to make those levels of performance easily accessible to the end user. We are living in a new era of computing in which the U.S. national laboratories will no longer be the dominant users of high performance computation, and it is no longer feasible to spend ten person-years of effort to put an important problem on a supercomputer. In contrast to this circumstance, in many situations the computational models used with current massively parallel computers are dismal in comparison to those familiar to users of conventional computers and workstations. The feeling one senses among some in the community is that increased difficulty of programming is a necessary price to be paid for the benefits of high performance. One of the challenges is to show that this need not be so.

In the near future, most high performance computing will be at the level of 100 megaflops to several gigaflops and will be performed by machines assigned to individuals or small groups of workers, or used in operational information/communication systems of business and industry. The effective use of large-scale parallel machines in these roles requires programming support at least comparable in power and generality to that available on present day workstations. The required programmability demands the adoption of more general models of computation. The development of satisfactory computational models for parallel computers that are efficiently supported by the hardware is a grand challenge of computer architecture. Without support for such computational models, the impact of architectural advancements will be severely impaired.

## 2.4: Programming for Massively Parallel Computation

Current programming practice for most massively parallel computers is based on the *data parallel* model of computation [3]. In this model, the principal data structures of a problem (usually large data arrays) are partitioned and assigned to the processors of the machine. It is rare to see large-scale parallel

computation where hundreds of processors are performing functionally distinct parts of a job (this is sometimes referred to as *functional parallelism*).

In the case of machines having a distributed memory architecture, a data parallel algorithm is expressed as machine code that is executed by all processors and the necessary communication among processors is implemented by manually coding explicit message-passing commands or by the use of a logically shared address space; the former approach is currently prevalent. Compilers available and under development will automate this process by letting the programmer specify data partitioning and by automatically generating the communications code for the given data partitioning.

A widespread misconception is that the important parts of the high performance field are architecture and algorithms. The effective programmability of the machine is limited by the computational model, and how well that model is supported by the hardware and software of the system, as mentioned in the previous subsection. That is, the interface between the architecture and the algorithm is a crucial issue. A major challenge is to move toward architectures that can efficiently implement a truly general-purpose parallel computation model. Architectures must support environments that facilitate functional parallelism in a massive way, as well as data parallelism.

## 2.5: The Goal of General-Purpose Parallel Computation

General-purpose computation is not well defined. At one extreme, the term means simply that one is able to perform any algorithm expressed in a complete language. At the other extreme, a general-purpose computer is expected to be efficient for applications ranging from science and engineering to business and industry.

Important programmability features that are standard for general-purpose workstations are not typically available for massively parallel computers. One of these is the ability to execute programs much larger than the physical main memory of the machine without having to program the swapping of information between main memory and disk; this is the familiar virtual memory idea implemented in all workstations. Another limitation concerns the linking of separately compiled programs; there are no standards for communicating large partitioned data structures between compiled modules. Realizing these features within the framework of massively parallel machines is a major challenge in computer science -- one that is often lost amid the concentration on hardware and algorithms.

Two of the major issues to be addressed are: (1) providing a global virtual memory for massively parallel computers; and (2) expressing and supporting parallelism and the interaction of concurrent activities. The model of computation supported by the architecture must have the properties necessary to create the desired programming environment. A basic approach to the challenge is to choose a model of computation that at once serves as the specification of an architecture and the target language for high-level programs. However, portability of parallel programs is also an important consideration.

## 2.6: Demands of the New Applications

The enormous rise in computer performance is making qualitative changes in the expectations and interests of users. For example, experience with larger computational grids and three-dimensional modeling of physical phenomena is motivating the use of more sophisticated data structures. In weather modeling, more effective methods are possible if computing resources are concentrated on unstable portions (e.g., storm systems) of the simulated space. However, unstructured grids make efficient usage of the processors in a parallel machine difficult.

Other areas include symbolic manipulation, compiling, heuristic search, etc. These types of computation are important in image analysis and may be crucial to solving the human genome problem [2]. University research has shown that these problems often have high levels of parallelism. However, as mentioned earlier, these problems are characterized by massive data sets, complex operations, and irregular data structures that exceed the limits of current supercomputers and programming paradigms. Making massive parallelism readily available in an effective and "user-friendly" manner for applications involving these characteristics requires the development of new techniques for mapping tasks onto parallel architectures.

Finally, the computing technology of the 1990s will enable access to vast information sources such as digital libraries, visualization images, and multimedia information objects [1]. Future computers must deal with such data entities as though they were the simple textual messages of today. The challenge is to incorporate into computers a high capacity to handle and transform these data.

## 3: Grand Challenge Problems in Computer Architecture

### 3.1: The Architectural Grand Challenges

The workshop opened with a wide-ranging discussion surrounding the computational characteristics and demands of the grand challenge *application*

problems. From these requirements, the participants translated the application-centered grand challenges into grand challenges for *computer architecture* for high performance computing. From a lengthy list of challenges, the attendees selected four primary challenges for presentation:

1. idealized parallel computer models,
2. usable peta-ops performance,
3. support of I/O and intensive communications, and
4. infrastructure for prototyping architectures.

It was recognized that the list from which these four were selected was by no means exhaustive, and that these four challenges overlapped and interacted.

This subsection summarizes these grand challenges for computer architecture. In [4], each problem is examined in more detail, and approaches for attacking them are considered.

#### *Grand Challenge 1: Idealized Parallel Computer Models*

The model of parallel computation is fundamental to progress in high performance computing because the model provides the interface between parallel hardware and parallel software. It is the idealization of computation that computer architects strive to support with the greatest possible performance. The model is the specification of the computational engine that language and operating systems designers can assume as they seek to enhance the power and convenience of parallel machines. It is not clear that a single model can fulfill all of the requirements, but it is essential to reduce the multitude of alternatives to the fewest possible number. Therefore, it is important to identify one "universal" or a small number of "fundamental" models of parallel computation that serve as a natural basis for programming languages and that facilitate high performance hardware implementations.

#### *Grand Challenge 2: Usable Peta-ops Performance*

Grand challenge applications require usable computer performance many orders of magnitude greater than the giga-ops performance available today, and the tera-ops performance which may be achieved soon. This computer performance cannot be obtained by simply interconnecting massive quantities of existing cpu, memory, and I/O resources, because the collective overhead associated with these interconnected resources can produce a system that is both unmanageable to program and ineffective to utilize. The challenge is to (1) dramatically improve and (2) effectively harness the base technologies impacting processors, memory, and I/O into a comput-

er system such that the grand challenge applications programmer has easy access to a peta-ops ( $10^{15}$  operations per second) of usable processing performance.

#### *Grand Challenge 3: Computers in an Era of HDTV, Gigabyte Networks, and Visualization*

Technology will be able to support startling new communications-intensive applications. For example, concurrent access by thousands of people to a digital version of the Library of Congress may be within reach in this decade. Digital video will enable workstations of the future to treat images as easily as characters and words are treated today. How can computer architecture and new communications technology evolve to facilitate such applications?

#### *Grand Challenge 4: Infrastructure for Prototyping Architectures*

Given that computer generations change every two to three years, new ideas on architecture must be evaluated and prototyped quickly. Prototype development involves hardware as well as software in the form of compilers and operating systems. The effects of new technologies and different application requirements must also be assessed. This computer architecture challenge is to develop sufficient infrastructure to allow rapid prototyping of hardware ideas and the associated software in a way that they can be realistically evaluated.

### **3.2: Multidisciplinary Approach**

The architectural grand challenges stated above are inherently multidisciplinary and involve team efforts that cross boundaries from software to hardware to applications. The early efforts in the development of parallel computers have shown that their viability and usability is a strong function of the supporting software systems. A substantial component of effort must be devoted to the automation of the software development process to exploit the power of the underlying hardware. This includes such problem areas as algorithm selection, algorithm optimization, data mapping, and parallelization. In the arena of high performance parallel computers, it is more important than ever for computer architects to attend to system software, application needs, and usability issues when designing and implementing machines. Computer architects must design systems that will efficiently support the software tools that will make the systems useful; it is a symbiotic relationship that must be leveraged to the fullest extent.

## 4: Conclusions

The grand challenge application problems are far more difficult than any problems yet attacked by computers. They require systems of unheralded capability. Such systems appear to be within reach by the year 2000 at reasonable cost, but only if significant advances are made in a large number of interrelated technology areas. Advances in device technology can supply only some of the improvement. The remainder has to be provided by architectures, algorithms, matching architectures and algorithms, system models, and new ideas in structuring systems to meet the challenges.

Computers for the grand challenge application problems will necessarily have characteristics not present today, such as advanced visualization, access to geographically distributed data bases, multigigabyte main memories, and terabyte per second communications links. These characteristics need to be factored into architectures to create the hardware and software features that can support and exploit them.

This paper has stated some of grand challenge problems in computer architecture for the support of high performance computing. In particular, (1) inventing a useful and widely accepted idealized parallel computer model or small set of models; (2) implementing systems that provide sustained usable petapops performance; (3) designing architectures that provide the capabilities needed in an era of HDTV, gigabyte networks, and visualization; and (4) creating an infrastructure for the rapid prototyping of new architectural organizations with the associated system software. The full report [4] discusses each of these in more details.

These problems are presented to the technical community as issues in computer architecture that demand further study if success is to be achieved in this nation's grand challenge applications. The purpose of this paper is to help stimulate some of the research needed to make high performance computers for the grand challenge application problems a practical reality.

## 5: References

- [1] E. A. Fox, "Advances in interactive digital multimedia systems," *Computer*, Vol. 24, No. 10, Oct. 1991, pp. 9-21.
- [2] *Grand Challenges: High Performance Computing and Communications*, Committee on Physical, Mathematical, and Engineering Sciences, National Science Foundation, 1991.
- [3] W. D. Hillis and G. L. Steele, Jr., "Data parallel algorithms," *Communications of the ACM*, Vol. 29, No. 12, Dec. 1986, pp. 1170-1183.
- [4] H. J. Siegel, S. Abraham, B. Bain, K. E. Batchner, T. L. Casavant, D. DeGroot, J. B. Dennis, D. C. Douglas, T.-Y. Feng, J. R. Goodman, A. Huang, H. F. Jordan, J. R. Jump, Y. N. Patt, A. J. Smith, J. E. Smith, L. Snyder, H. S. Stone, R. Tuck, and B. W. Wah, *Report of the Purdue Workshop on Grand Challenges in Computer Architecture for the Support of High Performance Computing*, Purdue University, School of Electrical Engineering, Technical Report Number 92-26, July 1992.\* (Also to appear as an invited paper in the *Journal of Parallel and Distributed Computing*.)

## Acknowledgments

The co-authors express their gratitude to Dr. Zeke Zalcstein of NSF for suggesting the workshop and making it possible. In addition, the workshop co-chairs thank Dr. Zalcstein for his advice in organizing and focusing the workshop. The co-chairs appreciated the efforts of the following people at Purdue University for their assistance in administering the workshop and producing the report and this paper: their secretaries Mary DeBruicker and Carol Edmundson, the financial coordinator Brenda McElhinney, the technical typists Vicky Spence and Kitty Cooper, and Continuing Education Administration Conference Coordinator Susan Umberger. We also thank Pearl Wang for her comments on the paper.

---

\*To receive a copy of this report, send a letter requesting "Purdue University, School of Electrical Engineering, Technical Report Number 92-26" to: Technical Reports, School of Electrical Engineering, 1285 Electrical Engineering Building, Purdue University, West Lafayette, IN 47907-1285.

**Appendix: Complete Names and Addresses of Co-authors**

Workshop Co-Chairs: H. J. Siegel and Seth Abraham  
Parallel Processing Laboratory  
School of Electrical Engineering  
Purdue University  
West Lafayette, IN 47907-1285

Bill Bain  
Intel Corporation SSD  
15201 NW Greenbrier Pkwy  
Beaverton, OR 97006

Thomas L. Casavant  
Dept. of Electrical and Computer Engr.  
University of Iowa  
Iowa City, IA 52242

Jack B. Dennis  
Massachusetts Institute of Technology  
Laboratory for Computer Science  
545 Technology Square  
Cambridge, MA 02139

Tse-yun Feng  
Dept. of Electrical Engineering  
121 Electrical Engineering East  
The Pennsylvania State University  
University Park, PA 16802

Alan Huang  
AT&T Bell Laboratories  
Room 4G-514  
Crawfords Corner Road  
Holmdel, New Jersey 07733

J. Robert Jump  
Dept. of Electrical Engineering  
Rice University  
P.O. Box 1892  
Houston, TX 77251

Alan Jay Smith  
Computer Science Division  
573 Evans Hall  
University of California-Berkeley  
Berkeley, CA 94720

Lawrence Snyder  
Dept. of Computer Science FR-35  
University of Washington  
Seattle, WA 98195

Russ Tuck  
MasPar Computer Corporation  
749 North Mary Avenue  
Sunnyvale, CA 94086

Kenneth E. Batcher  
Dept. of Mathematics and Computer Science  
Kent State University  
Kent OH 44242-0001

Doug DeGroot  
Texas Instruments  
Advanced Technologies & Components  
6550 Chase Oaks Blvd., MS 8435  
Plano, TX 75023

David C. Douglas  
Thinking Machines Corporation  
245 First St  
Cambridge, MA 02142

James R. Goodman  
Dept. of Computer Sciences  
University of Wisconsin-Madison  
Madison, WI 53706

Harry F. Jordan  
Depts. of Electrical & Computer Engr.  
and Computer Science, Campus Box 425  
University of Colorado  
Boulder, CO 80309-0425

Yale N. Patt  
The University of Michigan  
Dept. of Electrical Engr. & Computer Science  
1101 Beal Avenue  
Ann Arbor, MI 48109-2110

James E. Smith  
Cray Research, Inc.  
900 Lowater Rd.  
Chippewa Falls, WI 54729

Harold S. Stone  
IBM Thomas J. Watson Research Center  
P.O. Box 704  
Yorktown Heights, NY 10598

Benjamin W. Wah  
Coordinated Science Laboratory  
University of Illinois, Urbana-Champaign  
1101 West Springfield Avenue  
Urbana, IL 61801

