Project for Suggesting Computer Science Agenda(s) for High-Performance Computing: An Introduction

Uzi Vishkin*
The University of Maryland Institute for Advanced Computer Studies (UMIACS) and Electrical Engineering Department

Summary

Designing a successful agenda for the implementation of the High Performance Computing (HPC) program is a non-trivial technical challenge. The computer science and engineering research community has a vital interest in this challenge.

We view the technical problem (rather than the "political problem") of coming up with a coherent agenda for HPC as the main issue. The computing part of HPC is funded at a level of half a billion dollars per year, and a helpful way for thinking about the technical problem is as follows. Suppose that you are in charge of a budget of half a billion dollars per year for the next ten years for advancing HPC. What would be your technical plan for making the greatest impact?

Our project is aimed at developing implementable agendas. We would like that an agenda will have the support of researchers as well as meeting expectations of U.S. Congress[1].

We invited many computer science and engineering researchers in the area of HPC, who are affiliated with U.S. institutions, to participate. We made a considerable effort to give many in that research community, who have something to say on the topic, an opportunity to write a position paper. This explains why we view the project as placing a mirror in front of the community, and hope that the mirror indeed reflects many of the opinions on the topic.

The project had two stages. *Stage 1:* A workshop on developing computer science agendas for the High Performance Computing (HPC) program took place on March 14, 1994 in Arlington, Virginia, with nearly a hundred participants, including many leading computer science and engineering researchers in the area of HPC. The speakers were: Guy Blelloch, Carnegie Mellon, David Gelernter, Yale, John Hennessy, Stanford, Ken Kennedy, Rice, Tom Leighton, MIT, Steve Nelson, Cray, Avi Silberschatz, AT&T, Burton Smith, Tera, Uzi Vishkin, Maryland, and Robert Voigt, NSF. The panelists were: Frances Allen, IBM, James Demmel, Berkeley, Geoffrey Fox, Syracuse, Jerry Feldman, ICSI, and Vijaya Ramachandran, Texas. The workshop was meant to lead to the second, and main, stage. *Stage 2:* This edited book with agenda/position papers.

The initiative for the project has been voluntary and represents the feeling of the organizers that it is appropriate for the computer science and engineering research community to make suggestions concerning the future of high-performance computing. An executive summary of the position papers has been requested by the National Science Foundation (NSF). It is also invited for publication in the Communications of the ACM.

1 GENERAL OBJECTIVES

Given the dictates of Congress, a project of this nature is timely and important. A recent meeting of members of both houses of the U.S. Congress resulted in a report stating that the NSF could not spend beyond its current 1993 level until it submitted a report "articulating specific and measurable goals in this area". The report also called for "timetables and milestones" for achieving those goals. See also the above reference to Weingarten's summary for comments made by Congress with respect to ARPA and the NSF.

What this project implicitly suggests is to ask each group of researchers separately to put its research in context, and present its specific research plan in the way required by those members of Congress; this may include presentation of research as part of a global research agenda for high-performance computing. Yet it is important to emphasize that these global research agendas need not be the same for different projects.

The NSF will then be able to put together all these research agendas, and perhaps others, and submit them to Congress. This will serve two purposes:
(1) Respond to the questions raised by members of Congress.
(2) Let academic researchers continue to define their strategic research goals, as long as these goals can be

*Partially supported by NSF grants CCR-8906949 and 9111348.

[1] For a summary of two recent reports from congressional agencies, see: Fred W. Weingarten, From Washington: HPCC Research Questioned, CACM, Volume 36,11 (November 1993), 27-29.

justified.

2 TECHNICAL OBJECTIVE

The broad Federal High-Performance Computing (HPC) program is supposed to advance the world of general-purpose computing through a **transition** from serial to parallel computer systems and from serial to parallel programming. This means that: (1) parallel machines will be available, and (2) many people will be able to use them effectively (in co-existence with serial machines). The main technical objective, in my opinion, is how to bring about such a transition.

3 A PERSONAL PERSPECTIVE

Soon after I began my research activities in Computer Science in 1979, I became fascinated by the following challenge: having ubiquitous general-purpose parallel programming and parallel systems. While my research has naturally focused on one area of specialty, the design of discrete parallel algorithms, this has been the main general driving force behind my research activities.

In spite of that I was reluctant to get into initiating and organizing the current project. To understand my reluctance, note that a transition into general-purpose parallelism will probably be the greatest revolution in the world of computing since its early days affecting the way computer systems are built and many programmers (possibly million) do their job. It is simply a bit humbling to cope with such a challenge. Before taking on me the job, I held several discussions with some of the most senior researchers in computer science and engineering and decided to proceed following their encouragement.

4 THE STATE-OF-THE-ART?

The following quote, which is borrowed literally from Burton Smith's talk at the workshop of this project, summarizes his view on the state-of-the-art with respect to available parallel machines. *General-purpose computers have these important attributes: (i) They display adequate performance on any program; (ii) they require minimal programming effort to achieve it; (iii) their I/O capabilities match their computational power; and (iv) they use a "standard" scalable programming model. Today's high-performance parallel computer systems are clearly not "general-purpose".* The quote implies that the parallel machines are not yet here. Is this indeed the state-of-the-art?

5 HPC AGENDA(S)

HPC is a broad program which is supposed to advance the world of computing through a transition from serial to parallel computer systems and from serial to parallel programming, whose effect can be characterized as **revolutionary** on several frontiers. Still it is a **non-trivial technical problem** to develop an evolutionary **step-by-step implementable coherent agenda** for achieving HPC goals. For each step to be implementable, it should not only be doable but also agreeable to the specialists involved to guarantee their cooperation and an overall success.

5.1 WHAT DOES "AGENDA" MEAN?

Since scientific investigation often means "navigating in the dark", the term "research agenda" may sound like an oxymoron. We see two possible ways for reconciling that oxymoron. (i) Provide robust research platforms. (ii) Scientific investigation needs to encourage pluralism of opinions and directions ("let a thousand flowers bloom") and avoid discarding research directions without carefully considering their merit. However, it should not be as problematic if a research direction, which got a proper opportunity over time and did not lead to desired results, is considered less appealing than another direction which led to results.

Finally, the term "research agenda" should be understood as "vision" whenever appropriate. We note that comments made by Ken Kennedy of Rice University contributed to the current paragraph.

6 TRYING TO CHARACTERIZE HPC

The HPC program can be classified as a "basic applied science" program. The adjective "applied" is justified by the anticipation for building practical computer systems, whether in the short or long term, and the relevance towards building these systems or using them. The HPC program is still "basic science" since it calls for development of: (1) underlying technology required for high performance computing systems, and (2) generic software technology and algorithms. However, it is important to notice that even these aspects are not "curiosity-driven only". Furthermore, the case for the contribution of HPC to industrial competitiveness and as an enabling technology is not hard to make. Next, we try to characterize HPC by way of an analogy.

To stimulate debate we would like to draw an analogy between HPC, and its objective of achieving ubiquitous general-purpose parallelism, and the U.S. space program, announced in 1961 and accomplished in 1969, to put a person on the moon. Like any analogy, this also has its weaknesses, but we must draw analogies from other fields of science and technology since the field of Computer Science and Engineering has never before faced a challenge whose magnitude is similar to HPC.

Imagine the space program in 1961, where various vendors are developing their own products and are competing for selling them to universities or research laboratories or universities funded by by government funding agencies in rounds of three-year time windows. How successful would have this been in putting a person on the moon as early as possible?

While, we do not have the tools to compare the magnitude and complexity of the space program with HPC, we feel that at least in one aspect the HPC is more challenging. Recall that the space program was carried out by a small number of exceptionally skilled professionals (Wernher von Braun's leading team included a hundred people). However, the involvement of the masses was rather passive (essentially, watching the accomplishments in the media). While we do know how to compare the skills needed to lead HPC with the space program, the big difference with respect to HPC is that numerous programmers will have to update the way in which they perform their job. This active participation of so many people, not all with an advanced degree in computer science, should be taken into account in the implementation of HPC.

Try to add more detail to this analogy to the space program and question its relevance. Is this analogy helpful? if yes, does it mean that HPC needs a more strict plan? how much time will be needed for its completion? **if more than three years, then doesn't it mean that government must play a crucial role?** what should be the role of industry? what should be the role of the academic research community? how will "robust research platforms" (as per our discussion of what a research agenda may mean) to that community be provided? how should the implementation of such a plan be managed? who should do it?

7 REPRESENTATIVE QUESTIONS

We raise a variety of questions to stimulate debate. The questions in the next section should help in making an HPC agenda considered more pragmatic. The questions in the following section will help debate the concept of the formal HPC program. To focus the discussion, we highlight three questions as **focal questions**, and had asked that every position paper relate to them. However, before proceeding to the questions we bring several suggestions by the NSF.

7.1 NSF SUGGESTIONS ON POSITION PAPERS

The National Science Foundation "is very interested in the project and its output", and plans to have an executive summary of these position papers written for them. A summary was also invited for publication in the Communications of the ACM.

The NSF added the following suggestions to authors of the position papers. (1) Be constructive (rather than argue that others are wrong). (2) Present your agenda in a top-down manner. (3) An agenda should be coherent (or "hold water"). (4) Advocating a concrete subgoal even when not presented as part of a global agenda is acceptable, as long as this is explicitly stated.

8 QUESTION TO BE ADDRESSED BY AN AGENDA

Serial programs and software packages with many lines of code, representing a **huge investment**, exist among customers who are already interested in more computational power. They will have to switch over some transition period from a serial computer system to a parallel one. This represents only angle of the following question.

Focal Question 1 (revolution versus evolution): How will the transition from serial to parallel computation occur - instantly or gradually?

Focal Question 2: Should the development of parallel machines be done as a strategically planned progression of incremental steps towards an ideal end goal (to be called below Approach 1), or as a series of attempts, each of them attempting to build the ultimate parallel machine ("Approach 2")? Or, is there another approach?

We elaborate more on these questions. Suppose that a transition period from serial to parallel computation is projected to take ten years. Approach 1 will enable vendors to advance strategically and build several (two to four) generations of computer systems before finally building their best design at the end of the ten-year period. However, without specific planning the economic reality of high-performance computing will effectively dictate Approach 2, which, in effect, attempts to build the strongest possible complete system in each generation. **Question**: Are there opportunities which will be missed by Approach 2? **Question**: If the long term strategic Approach 1 is followed, can one state intermediate goals for machine design, application software development and education to use high-performance systems?

Strategic versus short-term research Ten years is too long a time horizon for industry to be actively interested in, and therefore Approach 1 will have to be done with proper support by government. This is a serious challenge for the following reason. Since resources are tight, some comparison of any proposed long-term strategic project which involves **feasible technology** with some alternative short-term kind of project which involves **available technology** by government agencies and Congress is probably unavoidable. We believe that even in cases where it is evident that industry will carry out a short-term project better and on its own, a special effort will be needed to steer funding towards the strategic project. Our intention is definitely not to play down the importance of short-term projects, but rather to make the case for the strategic ones and argue that strategic projects may be the ones where academia can not only make a difference but actually be a real leader. **Questions**: (1) How should the computer science research community go about achieving sufficiently long-term government support for a strategic project such as HPC? (2) Has CS achieved support for a similar project before? and, if not, what should be the model offered by established branches of science, such as Physics and Biology, that we should follow?

The following few general questions are beyond the HPC program. To what extent will excluding strategic projects doom universities to lose their prestige and become "second-class industry"? Will this make universities less interesting places? Will this result in having fewer of the better minds in the U.S. end up doing academic research? How strong is the computer science case relative to other disciplines in asking for research funding for strategic projects? How can we make the case for a potential impact on economic competitiveness? Is it true that academia in technological areas has a nearly-existential need for strategic long-term projects? To what extent do such projects fit the role that society (person in the street) expects from academia?

Commodity processors The performance of the processor used by standard computer systems has kept improving and is likely to continue to improve. A parallel system which is based on a non-standard processor may miss upgrades in performance of an improved standard processor. The industry has developed benchmarks for evaluating new standard "commodity" processors. These benchmarks measure some important performance features, but offer little reward for other features. For instance, faster context switching, which appears to be crucial for the overall performance of a parallel system, is not emphasized in these benchmarks. **Question**: How to bring about an update in the evaluation of standard processor to reflect the need of such processors to function in parallel systems? **A more general question**: Do current benchmarks harm progress in HPC by failing to emphasize certain aspects of performance? if so, what should benchmarks measure?

How much burden can be put on the programmer? There appears to be sufficient circumstantial evidence to suggest that it is possible to design efficient parallel algorithms for the permissive data-parallel programming paradigm. The literature includes also "more restrictive" parallel programming paradigms, as well as more restrictive models of parallel computation where, in addition to the Data-parallel programming requirements, the programmer is also required to control (or at least help control) issues such as locality of reference, synchronization, and communication among processors. Such programming paradigms tend to map better into some existing parallel machines. Unfortunately, only few methods for using these restrictive programming paradigms are known. It appears that programmers prefer and even like the basic Data-parallel paradigm and are not as successful with the more restrictive paradigms. **Question**: Is it too difficult (or at all possible) to use more restrictive programming paradigms for general enough problems and applications? Is there a middle way where programmers will have the option of using directives which will enable them to control locality of reference, synchronization, and communication among processors, but will not have to give up on getting good performance if they do not use that option?
Is it true that the **time of application programmers dominates all other costs** even for the relatively more expensive technologies which are under consideration?

Existing software Essentially all existing application software is in some serial programming language. **Question**: Will such software be effective (or at least acceptable) for any kind of high-performance machine?

Education to think and program in parallel Parallel algorithms represent a "new and different intellectual culture", requiring new ways of thinking to identify and express parallelism in programs representing known algorithms and new paradigms for designing new parallel algorithms. Designing parallel algorithms, even for a "relatively permissive" parallel programming paradigm such as "Data-parallel" is a demanding task. **Question**: Is it viable to have many users, who were not exposed to formal education on parallel algorithms, design parallel algorithms and write parallel programs? If the answer is negative, what should such education involve?
Computer scientists specializing in algorithms and data-structures wrote the standard textbooks which are used for teaching effective programming of serial computers and are developing their career around this issue. **Questions**: Are they also the right people to do the research which will eventually lead towards "writing the book" for teaching effective programming of parallel computers?

Considerations for curriculum development Available parallel computer systems are currently not ubiquitous nor is parallel programming. Some suggest to include parallelism as a core topic in the undergraduate curriculum in computer science and engineering based on the assumption that parallelism will become ubiquitous in five to ten years and a graduating student should be prepared for a life time career during which some transition to parallelism will occur. We avoid discussing this assumption in this paragraph, as it is discussed elsewhere in this paper. Given this assumption, it will be appropriate to have *education on parallelism focus on those robust principles that are likely to withstand the changing times.* What are those principles? To what extent should such education extrapolate possible future ubiquitous parallelism from currently available parallel machines?

Taking into account end-users of machines which are not yet ready Since general-purpose parallel machines are not yet ready, the question is how to guarantee during their design and construction stage that they are indeed general-purpose. Specifically, how to represent end-user application programmers (e.g., for the Grand Challenge problems) during the design and construction stage? **Questions**: Is it true that since system support for application software in most current parallel systems is below the standards of current serial systems, it is often impossible to analyze sources of difficulty in producing effective application programs? In other words, is it true that the **current state of affairs actually obscures such a study**? are there alternative ways for studying this issue? how helpful will it be to use a collection of representative "fundamental CS problems", which require a much smaller investment in programming, as a benchmark for predicting the ability of a reasonably skilled programmer to develop, in a time-effective manner, efficient and portable computer programs for much bigger application problems?

How does a cluster of workstations fit HPC Consider massively-parallel systems versus a cluster of workstations. **Questions**: To what extent do they span a continuum of of scale and performance? Can they all be viewed as part of a common scalable architecture? what are the **tradeoffs involved in terms of cost and performance**? Please include in the cost, the machine, software, as well as the time of application programmers. What are the cost-effective ways for balancing these costs?

Last, but not least, we would like to see research agendas which are both **promising and exciting**.

9 THE HPC PROGRAM CONCEPT

The **concept of the HPC program** appears to be rather unique. The literature accompanying the HPC program has documented extensively how an increase in computational power will benefit some important Grand Challenge applications. Indeed, relevance to problems, such as computational protein design and weather prediction, to name just two, is crucial for the credibility of HPC as a basic applied research program, since such problems provide concrete answers to members of Congress asking "what does HPC do for the person in the street?". However, what makes the program unique is that it went much further and actually required deliverable software for Grand Challenge applications on the new parallel computer systems, **as part of** HPC. Requiring such "deliverables" within the relatively short time-span of the HPC program may be a cause for concern, or even premature, since: (1) Parallel computer systems have not yet reached a stable situation, where various user, or even research, communities are satisfied with how they are supposed to use them; also, the issue of having computer programs which are portable from one parallel system to another has not yet been resolved. (2) The computational issues in some of the Grand Challenge applications are also not (yet) fully understood. (3) Past attempts at building parallel systems appear to indicate that researchers have repeatedly underestimated the difficulty of building them, even when such deliverables were not required. **Question:** Is it indeed a cause for concern?

The problem of building computer systems with increasing useful computer power has been the **"diamond in the crown of CS"** similar to the way that fundamental science problems such as sequencing the human genome are at the heart of life sciences. Also, the experience in the computing area has been that as computational power increases more useful applications are found. However, unlike other basic applied science projects, HPC required "deliverables" on the applications, **as part of** HPC.

Focal Question 3: Is high-performance computing a means to an end, or an end in itself? Should HPC be seen simply as building tools for use by scientists in other disciplines?

Back to the deliverable requirements of the HPC program. **Question**: Will it be better to instead have in the short-term a "federated program" concept, with two parts: (i) applications, and (ii) system-building plus basic techniques for using the system? The application part will focus on advancements towards reducing separately each Grand Challenge application to computational problems. The next stage may include developing "future dusty deck programs" for various applications using a standardized parallel language, such as FORTRAN 90. That language will be supported by the new parallel computer systems.

More on a possible federated HPC program concept Emphasis in applications among lawmakers is actually being gradually shifted away from the "Grand Challenges" toward "National Challenges", such as: electronic commerce, crisis management, health care, manufacturing and education. Of course, scientific applications, such as some Grand Challenges, will continue to play a major role in National Challenges, as well. However, National Challenges go far beyond these applications requiring the integration of communication and information in new ways, as per the evolving view of the National Information Infrastructure (NII). So, instead of narrowing down the already far-reaching goals of HPC, the opposite is actually happening. Below, we still raise the possibility of some narrowing down.

Questions: Try to define the part of system-building plus basic techniques of a federated HPC program in a way which: (1) would serve the Grand Challenges; (2) would serve the computing part of the National Challenges; (3) will meet Burton Smith's definition of general-purpose computers. If this can be done, will it not be a proper definition of a very challenging project? If yes, will it be necessary to combine it with other challenges? In other words, why not make this part the target, or explicit subtarget, of a federal program?

After all, a target definition which would have been robust enough to withstand the change of emphasis from Grand Challenges to National Challenges without violating a main stream computer science definition of general-purposeness, should be worth while.

We finish this section with a few related questions. **Questions**: What are the interests of the **wide users community**, not just other sciences, regarding more powerful computing power? how well-understood is this issue? are there comprehensive and reliable surveys that studied this issue? how to methodologically study the interests of the wide users community regarding more powerful computing and bring about a situation where they become a major factor guiding HPC?

The huge market of commercial **data base** applications is not directly represented in the HPC program. **Questions:** Should they be represented? In what way? Note that Focal Question 3, and its extensions, actually broadens this concrete question to the general interests of the wide users community.

10 PERSPECTIVE ON THE PROJECT

The project will (hopefully) have the following effect:

1. Make the community recognize that a coherent strategic plan for HPC is needed. (It took some effort to convince some people in government and academia that this is not a political-only issue, but rather a serious technical one as well.)
2. Get ideas for such plans from the research computer science and engineering community, and suggest to bodies as the NSF and ARPA to take these directions into account. (We indicated that the NSF already took initiative and agreed to do so.)
3. Start a serious discussion, which may lead to future workshops, etc.. This will also help younger researchers understand the main issues in looking for research directions.

10.1 GENERALISM IN COMPUTER SCIENCE

From a much wider perspective the project topic can be presented as follows. In some professions **generalists** are accepted as legitimate subspecialists. For example, consider general practitioners in Medicine and architects in the construction business. If one considers computer science as consisting of several disciplines then the missing specialty in CS is actually interdisciplinary within CS. As a field CS has not yet developed such a subspecialty and the project is about trying to somewhat rectify this situation in the context of HPC. Besides the direct interest, HPC is a good topic for demonstrating generalism, since building yet faster and more useful computers, and developing techniques for using them, have always been and will probably remain central issues in CS.

11 PROJECT ORGANIZATION AND ACKNOWLEDGEMENTS

Support is provided by The University of Maryland Institute for Advanced Computer Studies. The project is sponsored by the University of Maryland Institute for Advanced Computer Studies (UMIACS), and the National Science Foundation Center in Discrete Mathematics and Theoretical Computer Science (DIMACS).

The project is organized by Uzi Vishkin. A steering committee consists of Richard Lipton, Princeton and DIMACS, and Uzi Vishkin.

Larry Davis, the director of UMIACS, provided support for the project. Joseph JaJa was always available for consultation and moral support. Dick Lipton, Diane Souvaine and Bob Tarjan helped on the DIMACS side, and Bruce Barnes on the NSF side. I also had helpful discussions with Dick Karp (Berkeley), Randy Katz (ARPA), and Michael Rabin (Harvard). Last but not least, this project would not have been possible without the administrative support by Jo Weinstein, UMIACS. All this help is gratefully acknowledged.