Characteristics of HPC Scientific and Engineering Applications *

G.C.Fox[†], K.A.Hawick [‡]
Northeast Parallel Architectures Center,
111 College Place, Syracuse, NY 13244-4100
&
A.B.White [§]
Advanced Computing Lab, MS B287, C-ACL,
Los Alamos National Laboratory, NM 87545

January 1996

Summary

This document summarises the discussions, conclusions and recommendations of Working Group 2 of the Second Pasadena Workshop on System Software on Tools for High Performance Computing Environments. The group discussed the outlook for High Performance Computing(HPC) in the context of the disappointingly low uptake of HPC by industry and commerce. We identified a number of novel applications areas not hitherto able to exploit HPC as well as a number of applications and algorithmic characteristics, such as irregular data structures, code size, complexity, real-time to solution, data storage requirements and data access rates, indicating their potential for studying the relevance of high performance computing systems. We also identified a number of important non-technical issues such as the need for collaborative enterprise models between industry, developers, and customer end-users to further the successful uptake of HPC technology and to reap the potential economic benefits of so doing.

*Report of Working Group 2: Second Pasadena Workshop on System Software on Tools for High Performance Computing Environments, January 10-12 1995

[†]Group Chair

[‡]Group Secretary, Present address: UK Meteorological Office, London Road, Bracknell, Berkshire, England.

[§]Group Co-Chair

Contents

Introduction

The charter of Working Group 2 was centred on the characteristics of scientific and engineering applications and algorithms that require high-performance computer systems. In particular, the group tried to address the following points:

- What are the highest priority issues for system software and tools implied by these applications and algorithms?
- How will these applications and algorithms contribute to the long term commercial viability of high-performance computing systems?
- What particular system software and tools will attract independent software vendors to participate in the commercialization of these applications and algorithms?

More broadly, the group's charter was to consider possible national initiatives in HPC software from an applications perspective and taking into account the findings of the 1st Pasadena Workshop [2]. The group's charter directed us to scientific and engineering applications although the group took a broader view and in fact a major recommendation of the group is to build HPC software from a **broad** end-user base with a viable commercial market. This implies that one needs to consider applications such as the National Challenges, as described in the HPCC National Coordination Office (NCO) "Blue Books" for 1994 and 1995 , to get an appropriate set of requirements and standards for HPC software.

Although the group focused on HPC, many of the issues and indeed some of the groups recommendations also apply to HPCC. The distinction between HPC and HPCC can be rather subtle depending upon the country and the market sector being considered. HPC is obviously a subset of HPCC and in the scientific and engineering market sector HPC is at least currently more important than the communications aspect of the second 'C'.

Working Group 2 started its work with a set of short position papers presented by each of the (26) participants. These represented a broad spectrum of interests including academia (6), National Laboratories (9), Software and Systems Vendors (6), Government Agencies (5). Industrial research laboratories and the industrial user community were not directly represented, however. Applications expertise included: applied mathematics; astrophysics; biology; computer aided engineering; computational fluid dynamics; chemistry; medicine; real-time systems; physics; satellite data analysis and remote sensing and weather forecasting. Several Working Group 2 members had broad experience in a wide variety of applications outside this list. We have distilled comments from these position papers and the ensuing discussion into this present article.

The comments in the position papers of the individual Working Group members fell into two classes: relatively precise technical comments on the structure of applications such as their need for substantial I/O or or of adaptive irregular data structures; the second class of comments concerned the structure of the HPC enterprise and the nature of its evolution. This is illustrated by a concern that the HPC industry was widely perceived to be in financial difficulties and it was important to find ways to encourage progress in the field.

There was a lengthy discussion about HPC standards and software models. Clearly standards such as HPF and MPI were deemed to be very valuable but they have the characteristic of being motivated by the parallel computing (HPC) world. They are consistent with mainstream computing but not required by it. For example,a software developer on a PC or workstation would use perhaps C, C++, Fortran 77 or even Visual Basic and such software is not "HPC compliant". We recommend that "trickle-down" [1] strategies be complemented with a "trickle-up" standards and software model. For this, one starts with a software model for an area of computing which appears to have a solid fiscal and user base. We discuss some such areas in section 3.

Integration forums such as recent efforts for HPF and MPI are examples of the trickle-down phenomena. While they have had some impact, they have been driven largely by developers rather than end-users. Forums like these could use modern (Web based) collaborative technology to collect input from wider audi-

[1] The concept being: trickling **down** from **academia**

ences from **industry** and from **commercial users** as well as from academia and developers. Other examples which could be the starting point of a "trickle-up" approach are the software for symmetric multiprocessors, distributed computing or the software used on the World Wide Web itself.

HPC Software models built on this broad base of user input may not be as optimal as something specialised for a narrow user base but this trickle-up philosophy may in the long-run have greater viability and vitality. This idea was embodied as a major recommendation in the group's presentations to the other workshop participants on both of the last two days. The material here represents an expansion of these two presentations.

Another important theme was the identification of viable **enterprise models** or "Industry, Government & Academia business partnerships" for HPC applications, tools and system software development. Our trickle-up strategy represents one such viable model whereas most of the others required varying amounts of significant government intervention and therefore resource investment. In particular, we also recommended that more attention be given to a pulse or seed funding activity similar to the EUROPORT model as another especially important enterprise model. Experience of this sort of matching-funding from the UK's Parallel Applications Programme (PAP) and from the European Union's EUROPORT programme indicates a relatively high uptake of HPC by industry and commercial organisations.

1 Key Difficulties in HPC Applications

The consensus of the group was that there was a great need:

1. for code development tools;
2. for greater reliability of HPC systems;
3. for code migration tools;
4. to reduce system software inefficiencies;
5. for looking at exciting and innovative applications areas, (to help the HPC industry by stimulating new demands). This might involve very data intensive applications (in contradistinction to compute intensive ones) but also harder and more complex problems, irregular data structures and less obviously load balance-able problems;
6. to generally increase market confidence in HPC as technology that is becoming mainstream and that is now robust enough for real commercial and industrial applications.

These technical and economic needs are strongly correlated. The group believes that the technical deficiencies and general unreliability of HPC platforms are the key to why industry and commerce are still only gradually uptaking HPC technology in mainstream and core business activities. However, the technical deficiencies are largely due to insufficient commercial funding and industry-driven backing. It is not clear how this loop can be broken as HPC alone is probably too small an area to viably support commercial strength software. Government intervention in the form of programmes like EUROPORT or the UK's Parallel Applications Programme have been suggested as possibilities.

The group identified a number of general observations about HPC technology and Applications:

- It was felt generally that the HPC industry is perceived to be in a state of financial instability and that this is highly detrimental to long term planning both for developers and end-users.
- HPC as a field is technology-led rather than applications-driven at present. This is true internationally and is a serious concern for its long term future.
- The group felt that in the current maturing phase for HPC, vendors and developers should be starting with applications and not with the technology per se.
- We noted changes in the balance between usage of: workstation; mid range platforms such as departmental-sized compute servers or mainframes; and the very large supercomputer range of systems. These three sectors overlap to a significant extent and compete with each other and price/performance dictates the balance in user demand between sectors. We note a contrast in trends between USA and Europe in 5.
- New code development efforts (eg new companies) are more likely to adopt HPC than companies who already have (large) legacy codes.
- We noted we had an insufficient number of commercial end-users in the group. This reflects the limited number of active participants in the HPC field who are from industry and commerce. How do we encourage more industrial and commercial participation?
- A viable HPC vendor and software tool industry needs many viable HPC applications. There is a need to broaden the application base outside science and technology (since S&T is too small to sustain a viable HPC industry alone). Perhaps education for industrial decision makers in HPC exploitation needs greater attention and funding? Might funding be made available for HPC academics to participate in end-user community symposia such as in geophysics, chemistry, aerospace meetings? This would carry the message to the end-users.

- Can the HPCC development community set standards without full input from end-user communities and have the "tail wagging the dog"? HPCC is too important a technology to be controlled solely by its developers.
- as a group of international composition we note the dilemma of duplicated efforts between Europe and the USA as well as increasing input from the far east. To what extent is it healthy for HPCC developments to be reinvented internationally? Should national competitiveness be left to the end-user fields? There exists a clear trade off: competitiveness of the US computer-making industry vs. US computer-user industry. At present the latter may be suffering at expense of the former.

2 Applications Characteristics that drive HPCC Requirements

The group collective experience covered a number of applications areas in science and engineering. From the summarised applications list in section A, we identified the following interesting characteristics of scientific and engineering applications:

- computational performance - either FLOPS or just OPS in some cases;
- data storage access rates - MByte/s;
- data communications transfer speeds - Mbit/s;
- target range of platforms - and changing targets;
- number of separate disciplines needed in development team;
- size of software effort to develop and build needed software;
- size of software effort to maintain software;
- economic model for developing software - who pays and why.

These characteristics arose time and again in many of the HPCC vignettes the group discussed (see section 3. A key issue was that HPCC software should be well engineered just the same as software for conventional platforms is if it is to be suited for use in industry and commerce.

3 Vignettes in HPCC Applications

We believe that applying modelling and simulation to everyday life would expand the market for HPCC systems. Addressing industry and new and innovative applications therefore seems very important, since only by attracting the necessary industry-driven investment will HPC survive in the long term.

The group identified the following areas of special interest for understanding the relevance of HPCC and its software. Some are already being addressed, others may have to involve HPCC in the future for a solution.

1. NASTRAN and the Structural Engineering problem (see section 3.3 below);
2. Real time embedded systems for medical and military applications (see section 3.1 below);
3. Aerospace manufacturing, multidisciplinary analysis, and high end CFD (see section 3.2 below);
4. Crisis and emergency management;
5. Nuclear Weapons and the stewardship of the nuclear stockpile;
6. Environmental modelling;
7. Mission to Planet Earth;
8. Financial instrument modelling;
9. National and international power grid modelling and optimisation ;
10. Computational chemistry;
11. Intelligent vehicle highway systems;
12. QCD.

Three of these are especially worth focusing upon in terms of current issues for HPCC applications characteristics and requirements: Real-Time Embedded Systems; Aerospace Engineering; and Structural Engineering. A brief account of industrial uptake of HPCC in the UK and Europe is also given (section 3.4) to outline a possible means of addressing the issues identified.

3.1 Real-Time Embedded Systems

(Based on information supplied by Frank Blitzer, Honeywell)

Real-Time embedded systems applications are:

- processing intensive - already requiring teraFLOPS ;
- software intensive - with typically 100,000's of lines of code per application;
- dual use - both military and commercial markets (medical systems for example);

The market size will be around \$300M per year by the year 2000 for military applications. Deployed military systems need highly mobile processing equipment for both ground-based and airborne applications. This implies very high reliability of the hardware. We also anticipate a sizeable medical imaging and diagnostics market.

Some typical applications include systems for:

- wide area surveillance;
- command and control;
- battle management;
- real-time mission planning;
- real-time information to the warrior;
- gas and oil exploration;
- ocean floor mapping;
- autonomic ship;
- medical imaging.

The current proposed software development model for real-time software systems has the following characteristsics:

- algorithm development cycle involves use of distributed work stations with subsequent port to target (MPP) machine type;
- high algorithm coupling efficiency - between 50% and 75%;
- deployed resources must meet meet real-time schedule constraints;
- balanced resource loading problems occur - both static and dynamic;
- light weight kernel operating systems are needed.

Current barriers to successful development include:

- algorithm mapping performance studies are hindered by poor real-time support of workstation hardware;
- necessary software tools may **not** be applicable to **both** MPP systems and distributed workstations;
- static and dynamic load balancing HPC studies may not be possible until the application is hosted on the final target system.

The potential result of these barriers is that:

- software development costs will rise;
- will need **two** sets of tools;
- software sizeing estimates will be incorrect;
- overall cost and schedule delays will have a **negative impact** on the user.

Some sought after characteristics of the HPCC development environment are:

- accurate size estimates of code and HPC ;
- efficient mapping of complex algorithms ;
- capability of providing load-balancing;
- capability of efficient development of millions of lines of code;
- capability of software re-engineering and re-use;

The key challenges are to drive the market with hardware and software to make a self-sustaining industry and also to use government funding to prime the market development.

3.2 Aerospace Engineering

(Based on information supplied by Bill Feiereisen, NASA)

The Aerospace Engineering field is in a sense a revised Grand Challenge with a business model having the following characteristic stages:

- identify the market - typically 300 passenger for a subsonic 777 - competitors to be monitored too;
- conceptual design - relatively small part of the total cost;
- preliminary design - lines freeze - much of the final system life-cycle cost identified even though the expenditures at this stage are a smal fraction of total costs;
- final design - full cost identified;
- production;
- maintainance.

There is a tradeoff between how good or efficient the system is (in terms of $'s or performance) and the time to market. Earlier delivery generally captures a greater market share and is the main key to greater revenue. Shortening the design process by six months will typically capture much more of the market - potentially $0.5B for a typical airliner. Delaying the lines freeze, eliminating some steps (such as the pressure model) also help and this is where simulation, and HPC can make a difference. There are however problems with proprietary codes and problems with proprietary data outside of the companies direct control and many security concerns. This means a lot of the work must be carried out internally to the corporation and makes it difficult to transfer new research technology into the companies in a timely fashion.

Currently, the design cycle in bringing a large aerospace systemn to market involved running large CFD codes with a some structural engineering input. Typically this would draw upon the disciplines of four grand challenges oriented around four types of vehicle. Relying solely on computation for all its development is not a trusted option. This issue of trust is entirely separate from computing capability and has to do with the numerical algorithms and significantly turbulence modeling. The current design process is not one big CFD code but rather a combination of **many** CFD calculations and experiments. Typically no simulations are run concerning manufacturing or maintainance which is unfortunate as these factors are the really big drivers in cost, as well as the cost of capital itself.

Improved operational cycles will combine experiment with computation and will involve CFD computation at **all** levels. Collaborative technologies will contribute substantially in the form of: remote access to wind tunnels; remote control of experiments; and remote access to real-time data. Access to large experimental (and CFD) data sets for analysis are expected to lead to cooperative relationships. This industry is very risk adverse, and companies cannot afford to depend upon a process that might not work. Government sponsored work may be helpful in developing processes that are more efficient but which have a greater risk of failure.

Most of the aerospace companies say that they will **not** buy the next generation supercomputer, since it is perceived as **too risky** an investment. Instead, companies are exploring networked workstations and some such as Pratt & Whitney and MacDonnell Douglas, have shown great success.

3.3 Structural Engineering

(Based on information supplied by Louis Komszik, MacNeal-Schwendler Corp.)

The industrial success of high performance computing requires reliable hardware developed with production considerations, industrial usage and maintenance issues in mind. Systems must be equipped with good analysis, monitoring, debugging and evaluating tools to facilitate software development. They must have reliable compiler, linker and runtime tools, such as MPI. Systems should produce high sustainable performance on production applications. The current form of HPF is **not** useful in structural analysis with its complex data structures.

The commercial failure of many HPC systems recently must be linked to their being architectural experiments released into the commercial world before their time. There is a difficult economic balance to be reached between funding for new HPC architectures, and resources invested to produce stable commercially useable systems.

Uptake of HPC would be helped by an organized, timely and efficient government funding program to motivate ISVs to port their software onto the new hardware, even when the end-users have not purchased those platforms yet. Current experience with the EUROPORT program shows that many ISV's are willing to port to a machine not in use at the end-users premises yet, if the cost of port and a certain amount of guaranteed revenue or at least advance customer interest can be generated.

ISVs need to have staff well trained in the state-of-the-art computer science and computational mathematics issues. While MacNeal-Schwendler has such staff, this is by no means common amongst ISVs.

ISVs are only going to convert their industrially well established software to computers which are *already* available or are specifically requested by the end-users (paying customers). These customers are naturally cautious however and only want to buy established software. This situation makes hardware procurement decisions difficult as the choice of system by *industrial users* is strongly coupled to the established software base [2]. Only government intervention in some form may be able to break this deadlock.

The industrial HPC adoption process needs to start with the availablity of *reliable* HPC platforms; progressing through some form of Government intervention to aid the ports to new HPC systems, and **ending** up with the stable base of serious end-user customers.

The current process starts at the end-user organisation, who arrives at a procurement decision, followed by the ISV, who then converts the software to the procured hardware system. There is currently no practical government funding programme, although one is being established in connection with an ARPA funding for porting 5 ISV applications onto the IBM SP architecture. Finally, a hardware vendor, who is not selected by the end-user finds it difficult to fund development of the relevant ISV software for their platform, and therefore cannot break out of the cycle.

This scenario is a particular concern in the case of systems that are procured with the express objective of running some set of ISV products. It is very hard to obtain benchmark data in advance to obtain the right procurement decision.

HPC uptake also requires industrial end-users, such as FORD, General Mo-

[2] Academic end-users have been less concerned and have often been content to develop their own software - particularly if there has been grant funding available to do so.

tors, Boeing and Rockwell, to stay in the USA. These end-users are driven by their particular technical area and all they want to do is to solve their ever larger and more difficult problems even faster. Organisations like MacNeal-Schwendler has these customers today and the needs of these users ought to be the dominant driving force behind HPC developments, since it is they who can drive the commercial resource allocation to HPC.

3.4 Government Intervention Scenarios

(Based on experiences of one of us - KAH - at the Edinburgh Parallel Computing Centre)

Of particular interest are the national and international government sponsored programmes such as the European Union's EUROPORT programme, or the UK's Parallel Applications Programme. These programmes provide some degree of matching funding to encourage industry to uptake HPCC and exploit it at a lower cost and lower risk by building consortia of a mix of academics, developers, end-users and potential customers.

Experience at HPCC academic centres in the UK has been particulary favourable with this sort of matching funding. The Edinburgh Parallel Computing Centre (EPCC) was originally an academic organization within Edinburgh University formed on the basis of experience in computational physics. The UK Government Department of Trade and Industry initiated the Parallel Application Programme (PAP) as a matching funding programme to bring together academic HPCC code developers and industrial end users, as well as HPCC suppliers. EPCC evolved under relatively modest PAP funding into a flourishing commercially oriented organization and has recently spun off a commercial company to service the growing base of HPCC customers. British companies and other organizations such as: Rolls Royce (CFD for Turbofan simulation); British Aerospace (CEM for radar simulation); British Telecom(communications stacks for network simulation); The UK Home Office(fingerprint recognition); The UK Meteorological Office (weather and climate simulation); AEA Technology(reactor design); British Gas (network optimisation); Ford (parallel databases); Intera Information Technologies Ltd (oil reservior simulation); Shell Exploration and Production (seismic data processing); SIAS Ltd (road traffic simulation) all became customers of EPCC. Many of these projects led to HPCC procurement and uptake by the companies or organizations involved.

The programme matched funding to collaborative projects between EPCC and end-user organizations and also HPCC vendors and ISVs on the basis of how much resources the companies put into the project. This latter was a combination of cash; manpower; software; hardware. Matching funding was typically tapered so that satisfied customers would eventually fund projects entirely out of their own resources. This was a clear incentive for success measured in industrial relevancy and satisfaction rather than other more academic criteria.

The value to EPCC in helping to grow a customer base; to the end-users of subsidising early adoption of HPCC technology; and to the HPCC vendors and ISVs of attracting new customers made the programme highly successful. Other centres in the UK such as the HPC Centre at Southampton University were also successful under the same programme and the European Union's EUROPORT project was driven in part by the clear success of the UK programme.

The USA could benefit from such matching funded programmes rather than simply funding development of what may be unwanted products without pre-established customer bases.

4 Long Term Viability for HPC Applications

The long term success of HPC software projects like those under PAP and EUROPORT depends upon how HPC utilisation effort will be sustained in the future and in particular on the maintainance schedule for the HPC application codes that resulted from the projects. Two main factors affecting this will be: which organizations decide to adopt and maintain the parallel applications codes; and either the continued stability or an organized development of the underlying HPC systems level software such as message passing systems, libraries or parallel languages that have been used to develop parallel applications.

In the case of EPCC projects under the PAP scheme, an early decision was made to allow the copyright of the HPC applications to remain with the industrial development partner who originally owned the serial application code. This was felt the best way to encourage these industrial partners to maintain the parallel applications codes. Other projects have split copyright amongst members of a consortia, which may provide less motivation for individual consortia members to pursue maintainance of HPC applications codes beyond the timescales of the original funded project, unless some motivating means keeps consortia members together. It is too early to forecast how successful most of the PAP or EUROPORT projects in the 'long' timescales of 5 years hence. However, we stress that independent software vendors will only continue to maintain the 'pulse funded' parallel codes if there is a viable enterprise model for this. In many cases it might be that slow user uptake and continued volatility in HPCC systems will lead to abandonment of parallel versions. It is critical that this issue be monitored and suitable resources be made available to tide these projects over until a sustaining enterprise model can be maintained without government intervention.

At the time of writing the underlying HPC systems level software such as the definitions for message passing or data parallel programming languages is still in a state of flux. There has been a degree of paradigm convergence over the last 3 years and it is now realistic to expect HPC software codes written with message passing calls for example to be maintainable to the same industrial standards as vector codes currently are. Similarly, the widespread acceptance of Fortran 90 and the continued (albeit gradual) uptake of HPF suggests that data parallel programs will be maintainable to similar levels soon. This is in clear contrast to the situation at the start of the decade when experience was that many industrial development partners viewed parallel applications as purely research projects with no prospect of maintaining the resulting applications codes.

On a technical level, it is not clear how new developments in parallel languages and libraries for example will be embraced and incorporated into what will become the 'HPC applications legacy codes' of the future. It is important that some degree of **stability** in HPC systems level interfaces (like MPI or HPF) be perceived so that industry has the confidence to maintain applications codes that use them. From this perspective it is probably better to develop MPI-2 or HPF-2 which have well defined mupgrade migration routes and a large degree of backward compatibility, rather than to keep introducing completely new packages with no degree of compatibility. This phenomena is demonstrated by the continued lifecycle of the Fortran series of languages and to a less extent more recently the C series of languages. This contrasts with the practically non-existant industrial uptake of completely new programming languages.

It is fairly certain that the backwards compatibility provided by Intel series of processors for the PC was instrumental in ensuring a stable and broad base of software developed for PCs. This backward compatibility, although unfortunately anathema to the research world, will be necessary for the future prosperity of HPC software and by implication HPC as a whole.

5 Contrasts between USA and European Computer Usage

It is worth noting a significant difference between the computing usage climate in the USA and in smaller countries like the UK and other European nations. End-user companies in the USA have tended to be large and have been able to supply more significant compute resources to their staff. The reasons for this are not obvious, but the point is that US companies generally command greater budgets for traditional vector supercomputers and more recently have been able to supply individual staff with workstations more readily than companies in the UK and Europe.

In contrast, only very large UK companies have been able to afford traditional vector supercomputers and have relied upon department sized computers - mainframes and other 'medium range' systems. The presence of departmental machines has perhaps slowed the incursion of workstations into UK companies. However, parallel computers have often been marketed as a cheaper alternative to traditional vector supercomputers and as such have been embraced more openly by UK organizations than US ones. This is because many UK ornagizations have never had traditional vector supercomputers and therefore have no staff infrastructure to change. It is much easier for them to introduce the new parallel technology into a vacuum where there is no traditional vector supercomputing infrastructure to change. The success of centres like EPCC in introducing parallel computing into the operational cycle of companies like Rolls Royce, British Telecom or SIAS must be due in part to this effect.

This situation may now change however. Workstations are already common throughout USA companies and are gradually encroaching into UK companies. Workstations provide 'competition' with departmental machines and with parallel systems. Indeed workstations are often used at night as 'clustered parallel computers' in companies. That this is possible is thanks in large part to the software and hardware development work carried out originally with HPC systems in mind.

It is worth noting that success for organizations like EPCC will ultimately lead to their demise. Successful transfer of parallel computing to industry and industry's adoption of this technology as mainstream will obviate the need for the academic HPC centres except as developers of the *next* technological advances.

6 Applications Enhancement by Systems Software Improvement

The group identified a number of enhancements in applications that would result from improvements in systems software.

1. Better debuggers would speed up code development.
2. Better code profilers and performance monitoring tools would help identification of areas where HPC could really improve applications.
3. More stable operating systems releases for HPC platforms would enhance user confidence and increase chance of HPC uptake by industry, commerce and other mission critical software needs.
4. Better collaboration software (and compatibility with existing systems such as the world wide web) would lead to less time spent reinventing and redeveloping algorithms and multiple unmaintained versions of utilities that do almost the same thing.
5. More resource-efficient library codes (in terms of storage capacity, processing speed as well as bandwidth) would substantially enhance the capability of existing HPCC systems.
6. Since isolationism and commercial forces do not always produce the optimal solutions - forums to introduce some standards from end-user requirements (independent of the suppliers) would be powerful. HPF and MPI forums are examples.

7. Code browsing and analysis tools to aid in code migration to parallel systems, whether it be in the form of libraries, or other software engineering tools, would aid uptake of HPC.

8. Reduction of the latencies due to system software due to under-engineered communications libraries would allow better use of existing HPC hardware and give greater confidence in it.

7 Working Group 2 Conclusions and Recommendations

To summarize the conclusions of Working Group 2, some means of investigating long term economic, social, legal, technical, viable models for HPCC evolution in the USA must be found and a solution identified. We note:

1. the current incentive model for vendors, ISVs, and industries is incomplete;

2. in Working Group 2 only two ISVs were parallelizing codes;

3. the business case for HPC(C) uptake in a industry is typically inadequte;

4. the use of MPPs in gas and oil industry is a counter-example and we need to examine why this **has** been successful.

5. the example of government intervention in Europe to provide product development and advance customer uptake for HPC(C) may be worth following.

Working Group 2 therefore proposes the following actions:

1. As an HPCC community we must identify platform independent software standards scaling from workstations to MPP with tools using these standards. This requires continued support and involvement in forums like those for HPF and MPI but also requires great efforts to broaden the user base who provide input to such forums.

2. Industry and commercial users must be more actively involved in HPCC - academics must find a way of becoming more involved in end-user symposia as a first step to this. Currently, the tail (academic developers) is wagging the dog (end-user communities). In particular, can we identify big enough industrial markets upon whose requirements to base viable HPCC standards (eg SMP, distributed systems, WWW) and upon whom to focus the new enterprise model first?

3. HPCC developers must find collaborators to broaden the application base outside traditional science and engineering areas and into applications such as:

(a) integrated manufacturing (eg car body design);

(b) Low-latency (eg real-time systems);

(c) Data intensive (eg crisis management systems);

(d) Business (eg financial instrument simulation);

(e) Event driven modeling and simulation (eg defense and road traffic simulations).

A Applications Categories

These application categories [1] have been developed for their relevance to scientific and engineering as well as industrial and commercial HPCC activities. Another good list of applications categories is given in [3]. These categories form the basis for a "roadmap of HPCC applications", an online information resource available on the World Wide Web as part of the National HPCC Software Exchange [4]. The headings are given here.

a Information Creation - Simulation

- **1** Computational Fluid Dynamics
- **2** Structural Dynamics
- **3** Electromagnetic Simulation
- **4** Scheduling
- **5** Environmental Modeling
- **6** Health and Biological Modeling
- **7** Basic Chemistry
- **8** Molecular Dynamics
- **9** Economic and Financial Modeling
- **10** Network Simulations
- **11** Particle Flux Transport Simulations
- **12** Graphics Rendering
- **13** Integrated Complex Simulations

b Information Analysis - Data Mining

- **14** Seismic Data Analysis
- **15** Image Processing
- **16** Statistical Analysis
- **17** Healthcare and Insurance Fraud

18 Market Segmentation Analysis

c Information Access - InfoVision

19 Online Transaction Processing (OLTP)

20 Collaboratory Systems (eg WWW)

21 Text on-Demand

22 Video on-Demand

23 Imagery on-Demand

24 Simulation on-Demand

d Information Integration - Systems of Systems

25 Command, Control and Intelligence (C2I)

26 Personal Decision Support

27 Corporate Decision Support

28 Government Decision Support

29 Real Time Control Systems

30 Electronic Banking

31 Electronic Shopping

32 Agile Manufacturing

33 Education

Historically parallel and HPCC systems have only addressed science and engineering simulation applications. This is now changing, as indeed it must to allow HPCC a broader base of support for long term viability.

B Participants of Working Group 2

- Ackerman, Michael, NIH/NLM, ackerman@hpcc.gov
- Blitzer, Frank, Honeywell, fblitzer@space.honeywell.com
- Bredekamp, Joe, NASA HQ Code S, jbredekamp@hq.nasa.gov
- Bromley, Keith, NCCOSC, bromley@nosc.mil
- Coley, Terry, Biosym, trc@biosym.com
- Darema, Frederica, NSF, fdarema@nsf.gov
- Decyk, Viktor, UCLA, decyk@physics.ucla.edu

- Dosanjh, Sudip, Sandia national lab., email-unknown
- Eldredge, Mike, Centric, eldredge@centric.com
- Ermler, Walter, DOE, ermler@er.doe.gov
- Feiereisen, Bill, NASA Ames Res Ctr, Bill_Feiereisen@qmgate.arc.nasa.gov
- Fischer, Jim, NASA Goddard SFC, fischer@nibbles.gsfc.nasa.gov
- Fox, Geoffrey, Syracuse Univ, gcf@npac.syr.edu (Chair)
- Gaasterland, Terry, Argonne National Lab., gaasterland@mcs.anl.gov
- Glimm, James, SUNY at Stony Brook, glimm@ams.sunysb.edu
- Hawick, Ken, Syracuse Univ, hawick@npac.syr.edu / khawick@meto.govt.uk (Secretary)
- Hitchcock, Dan, DOE, hitchcock@er.doe.gov
- Karin, Sid, San Diego Supercomputing Ctr, karin@sdsc.edu
- Komzsik, Louis, MacNeal-Schwendler Corp., L_KOMZSIK@macsch.com
- Lin, Norman, Univ of Oklahoma, nlin@tornado.gcn.uoknor.edu
- Maizel, Jacob, NIH/NCI, jmaizel@fcrfv1.ncifcrf.gov
- Martino, Robert, NIH/DCRT, martino@alw.nih.gov
- Mesirov, Jill, Thinking Machines Corp, jill@think.com
- Payne, David, Intel, payne@ccsf.caltech.edu
- White, Andrew, Los Alamos National Lab, abw@lanl.gov (Co-Chair)
- Williams, Leland, DOD-Modernization Office, williams9@ccf.nrl.navy.mil
- Wong, Joanna, IBM Kingston, yjw@vnet.ibm.com
- Woodward, Paul, Univ of Minnesota, paul@s1.arc.umn.edu

References

[1] Fox, G.C., Williams, R., and Messina, P., "Parallel Computing Works!", Published by Morgan Kaufmann, 1994.

[2] Messina, P., "Proceedings of the first Pasadena Workshop on System Software and Tools for High Performance Computing Environments"

[3] Thomas Sterling, Paul Messina, Paul H. Smith, Proceedings of Workshop on "Enabling Technologies for Peta(FL)OPS Computing"

[4] "The National HPCC Software Exchange", a project of the members of the Center for Research on Paralle Computation (CRPC) at Caltech, Rice, Syracuse and Tennessee Universities, Argonne and Los Alamos National Laboratories. On the World Wide Web at: http://www.netlib.org/nse/home.html