Applications/Algorithms Roadmapping Activity. First Stage Final Report

Trefethen, Anne and Higham, Nicholas J. and Duff, Iain and Coveney, Peter

2009

MIMS EPrint: **2009.85**

ISSN 1749-9097

Developing a high performance computing / numerical analysis roadmap

HPC/NA

Prof. A. E. Trefethen, University of Oxford
Prof. N. J. Higham, University of Manchester

Prof. I. S. Duff, Rutherford Appleton Laboratory
Prof. P. V. Coveney, University College London

Applications/Algorithms Roadmapping Activity

First Stage Final Report

June 2009

http://www.oerc.ox.ac.uk/research/hpc-na

Contents

August, 2009

Preamble

This final report of the initially funded stage of the HPC/NA roadmapping activity brings together elements from the individual workshops to date, the input from various groups and the completed desk work reflecting related activities.

The report is the result of a 9-month funded roadmapping activity in which there were three workshops bringing together HPC application researchers and developers, computer scientists and numerical analysts, a significant desk-based study of existing activities and contact with many groups around the UK.

The resulting findings should not be taken as the final word, as there are still groups who may not have had the opportunity to contribute or comment, and the picture is by necessity evolving. The work that has started with this initial project will continue and the roadmap as presented will continue to grow in evidence base.

During the latter part of this project an international activity, the International Exascale Software Project (www.exascale.org) was initiated to develop an international collaboration to form a roadmap for software infrastructure for extreme computing. We have included the findings of that activity to date in this report as it is clear that the UK roadmap must sit within the context of the IESP roadmap.

Executive Summary

This first version of a roadmap document is the outcome of three community meetings together with input from similar activities in Europe, the USA and collaborative international projects. The roadmap activity aims to provide a number of recommendations that together will drive the agenda toward the provision of

- algorithms and software that application developers can reuse in the form of high-quality, high performance, sustained software components, libraries and modules
- a community environment that allows the sharing of software, communication of interdisciplinary knowledge, and the development of appropriate skills.

During this activity five themes emerged, namely

1. **Cultural Issues**: building cross disciplinary teams and community development; models for sharing information, algorithms, software and ideas; engaging with international activities.
2. **Applications and Algorithms**: identifying exemplar applications; mapping applications and algorithms; integration across models; adaptivity; efficiency; scalability; partitioning and load balancing; data management; scalable i/o; portability and requirements for new architectures.
3. **Software**: language issues; ease of use; efficiency and performance; support for development of software libraries and frameworks; validation methods; software engineering skills; standards and compilers; and active libraries and code generation.
4. **Sustainability:** developing models for sustainable software; sustainable programming models and interoperability.
5. **Knowledge Base:** gathering and disseminating knowledge on existing projects; education and training strategies.

This report provides the evidence to support the requirements gathered under these themes. The section on implementing the roadmap includes six recommendations built on a number of suggested instruments to meet the requirements identified.

Recommendation 1: The network of HPC applications, numerical analysis and computer scientists within the UK should be facilitated using the instruments as indicated in table 2. This network should provide communication of best practice across the UK community and provide a common interface to international activities. It should also provide mechanisms for raising awareness of existing applications, algorithms software and activities.

Recommendation 2: Key application exemplars should be developed.

Recommendation 3: There needs to be continued investment in new algorithm development to underpin existing applications move to the new architectures and to enable new applications. Table 1 should provide an indication of the most important algorithms in the sense of impact across a number of application areas.

Recommendation 4: There needs to be a continued engagement of computer scientists within key application areas and support for fundamental computer science research on abstractions, code generation and adaptive software systems and frameworks for reuse within the context of those applications.

Recommendation 5: Sustainability of software should be addressed through the development of models of sustainability including collaboration with software foundation, industry and international activities. This will require an initial investment from EPSRC and other research councils and funding agencies.

Recommendation 6: There is a need for a more joined up approach for skills training and education. There are a number of disparate courses around the UK that could be leveraged to provide a better platform for graduate students. A DTC in this area would also provide a more comprehensive consideration of the cross-discipline requirements. Key areas have been identified where there is a clear lack of education a particular example is optimisation.

The Challenge

The applications/algorithms roadmapping activity has the goal of developing the first instantiation of a high performance numerical algorithm roadmap. The roadmap l identifies areas of research and development focus for the next five years including specific algorithmic areas required by applications as well as new architectural issues requiring consideration. It will provide a co-ordinated approach for a numerical algorithm and application development.

Many applications from different fields share a common numerical algorithmic base. The roadmap aims to capture the elements of this common base, to identify the status of those elements and, in conjunction with the EPSRC Technology and Applications roadmapping activity, to determine areas in which the UK should invest in algorithm development.

A significant sample of applications, from a range of research areas has been included in the roadmapping activity and we are interested in adding any new or underrepresented area.

The applications should provide the basis to understand:

- The role and limits of a common algorithmic base
- How this common algorithmic base is currently delivered and how should it be delivered in the future
- What the current requirements and limitations of the applications are, and how these should be expanded
- What are the “road-blocks” that limit the scope of the future exploitation of these applications
- A better comprehension of the “knowledge gap” between algorithmic developments and scientific deployment
- How significant computing languages as well as other “practical” issues weigh in the delivery of algorithmic content

The Grand Challenge is to provide

- Algorithms and software that application developers can reuse in the form of high-quality, high-performance, sustained software components, libraries and modules that lead to a better capability to develop high-performance applications.
- A community environment that allows sharing of software, communication of interdisciplinary knowledge, and the development of appropriate skills.

We hope that the roadmapping activity will also help to identify UK strengths and weaknesses to assist in choosing appropriate investment areas.

Background

The Changing Face of HPC

High performance computing (HPC) has moved from being a somewhat esoteric interest of a few "bleeding-edge" scientists to a necessity for any computational scientist, any software developer, and any industry that uses simulation as a tool. This shift has come about as processor chip designs now mimic the architecture of high performance computers, with multiple processing cores on a single chip, making efficient programming of a single processor computer as complex as it once was to develop software for a high performance computer. Furthermore today's HPC systems comprise many thousands of such processors connected by a high performance switch requiring hierarchies of different programming models.

Advanced computing is an essential tool in addressing scientific problems of national interest, including climate change, nanoscience, the virtual human, new materials, and next-generation power sources, but as importantly it is equally essential to solve commercial and industrial problems in financial modelling, engineering, and real-time decision systems. Complex surgery, tumour imaging and cancer treatment, effects of drugs on the human system, and many more health-related applications increasingly depend upon advanced computing. A standard dual-core laptop is equivalent to one of the top500 machines of only 12 years ago. High performance computing is no longer for the select elite!

Yet our capability to use these machines is diminishing with the increasing complexity of hardware. The algorithms underlying current software are not able to cope efficiently and are not able to fully exploit advanced computing architectures. We can no longer take it for granted that for each successive generation of microprocessors the software applications will immediately, or after minor adjustments, run substantially faster. Companies who supply computational software for simulation development will be hit by the fact that all underlying algorithms will need to change to take advantage of the new evolving machine architectures or will likely run slower on newer computer systems.

As a nation we are poorly equipped to address these challenges. There is a lack of cohesion across the disciplinary groups that need to be brought together, namely mathematics, computer science, engineering and domain scientists. There are few existing channels to allow a flow of knowledge and expertise across the academia/industry divide. We have an inadequately trained next generation of researchers and we do not have the required skilled workforce.

It is widely recognized that, to date, algorithms and libraries have contributed as much to increases in computational simulation capability as have improvements in hardware [7]. The developments in computer systems throw even greater focus on algorithms as a means of increasing our computational capability [2]. Enhancing the national capabilities in advanced computing algorithms and software will have a major impact on the UK's future research capacity and international impact in the ever increasing number of domains within which high performance computing is, or is set to become, a core activity. In addition to all the domains within EPSRC's own remit, there are many areas of interest to other UK Research Councils for which it is absolutely essential that the UK make strategic investments now in high performance computing algorithms and software to safeguard our ability to be internationally leading in the next ten to twenty years and beyond.

In the UK, the OST Science and Innovation Framework 2004-2014 [8] identified the need for computational research and the underpinning research infrastructure to support it. The International Reviews of Mathematics (2004) [11] and High Performance Computing (2005) [12] both identified the UK as having internationally leading research groups in the areas of numerical analysis and HPC; however, the lack of collaboration between the numerical analysis community and HPC researchers was also identified by both as a significant concern that will result in the UK missing important scientific opportunities. In countries such as France, Germany and the USA that have large government funded laboratories there are strong links between heavy users of numerical algorithms for solving grand challenge problems and numerical analysts and computer scientists in academia, with significant funding from various agencies to facilitate these collaborations (for example see the US DOE SciDAC effort [5]). In the UK, such links between scientists in academia and industry are underdeveloped and do not lead to the symbiosis that we see in these other countries. The consequences of the UK situation are that algorithmic breakthroughs are slow to be picked up and used by computational scientists, and numerical analysis research efforts are not necessarily being targeted at the problems of most interest to the scientists.

The roadmap aims to bring these communities together, to collaboratively identify the algorithmic and computational areas that need investment to ensure the UK is able to compete internationally. It identifies potential priority areas where the UK can have impact, and proposes actions to bring together the stakeholders to take those priority areas forward.

Architectural Trends

Physical barriers mean that while the number of transistors per chip continues to double at the historic rate, processor clock rates are ceasing to increase. This is the primary driver for chip designs changing to multi-core architectures. Intel will ship 6-core this year, 8-core next year, and the 16-24 core Larrabee chip in 2010. Meanwhile, NVIDIA is already providing 128-core today to 256-core later this year, and AMD's forthcoming GPU-based product is 320 core. But the number and throughput of pins on chips are reaching their limits, meaning that multi-core chips will have an increasingly large gap between processor performance and memory performance.

Meanwhile, economic necessity forces the use of many thousands of commodity CPUs to scale up to the next generations of high end systems. Hence in order for scientists to tackle increasingly complex problems using HPC, algorithms will need to be developed that employ novel mathematical and coding techniques. Moreover, new and revised algorithms will need to be rapidly translated into software developed with careful adherence to standards and portability---which in turn support both maintainability and adaptability---to ensure a long lifetime.

In the longer term it is likely that by 2015 memory will be layered on a chip in a 3 dimensional configuration and the amount of memory per core is uncertain. The transition from HDD (Hard disk drives) to SSD (Solid State Disks) may also impact applications capability and algorithm performance. And the fact that undoubtedly the world will be heterogeneous to the extreme will require a dynamic capability of applications and algorithms that has not been seen to date.

Relevant activities

A number of activities in other countries are relevant to the roadmap and where possible we would like to leverage and borrow from those strategies. It is important that any roadmap for UK activity makes sense within the context of the global picture.

In the second workshop we learned from Michel Kern of INRIA about the “Thinking for the Petaflop” activities in France where the initial areas of interest coincide with this roadmap. The report from these activities has now been published [14] and reflects remarkably closely the findings of this activity.

There have been a number of studies in the USA [1, 2, 3] focused on this issue and we aim to leverage their findings. The recent report on exascale computing for energy and environment notes “The current belief is that the broad market is not likely to be able to adopt multicore systems at the 1000-processor level without a substantial revolution in software and programming techniques for the hundreds of thousands of programmers who work in industry and do not yet have adequate parallel programming skills.” [4]

Other significant activities in the USA that are highly relevant to this activity include the ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems [3], the DOE Modelling and Simulation at the Exascale for Energy and the Environment [4] study and the Oak Ridge report on Scientific Application Requirements for Leadership Computing at the Exascale [6]. The first of these reports provides a comprehensive review of both hardware and software issues for extreme computing, together with the practical issues surrounding development of suitable data centres for exaflop computers.

The reports have focussed on key application areas such as energy, climate, weather simulation, biology and astrophysics, which not surprisingly mirror many of the applications that are of interest here. However all these reports are focussed on exascale computing, the peak of high-performance computing, and our focus is somewhat broader than this.

The DOE has instigated a series of Scientific Grand Challenges Workshops that have focused on Climate, HEP, Nuclear Physics, Fusion Energy Sciences, Nuclear Energy, and plan to cover Basic Energy Sciences and Biology later in 2009. At the time of writing only the report for Climate Science was complete. All reports will appear on the workshop series website http://extremecomputing.labworks.org/ in due course.

A further initiative is the International Exascale Software Project (www.exascale.org) that is funded by the DOE, NSF, EU and has been supported by other international funders including Japan and the UK. This activity aims to develop a roadmap for the software infrastructure required for exascale computing and to formulate a framework that will enable the development of the infrastructure through international collaboration on open source software. The roadmap that is being developed is the closest in form to the UK roadmap presented here and naturally the UK roadmap should be considered in the context of the international activity. The IESP roadmap will be developed over the next six months with a workshop in Japan in October 2009 followed by a final workshop in the Spring of 2010. The recent workshop in Paris (June, 09) provided an outline of the roadmap which is included here in annex 8 and to which we will refer later in the report.

Our interests are broader than the exascale high-end -- we are hoping to enable a wider range of capability computing applications. However the complexities of the exascale within a node will reflect across a broad group of architectures and scales and we can therefore glean and hopefully leverage a great deal from the efforts at this level.

The HPC/NA Roadmap Activity

Methodology

A full description of the methodology used to develop the roadmap is provided in Annex 1. In essence the effort has been a combination of background desk work, a series of workshops and a collaborative community site. The latter has not provided input to this version of the roadmap but should provide significant input in the future. The three workshops held in Oxford, Manchester and London are described in full in Annex 2, 5 and 6; they brought together applications developers, numerical analysts, computer scientists, industry scientists and computer vendors. The outputs from the workshops have been distilled and circulated to the broader community.

Roadmap Themes

A number of themes have been developed through the consultation. Not surprising these are largely mirrored in other international activities although some are local to the UK. We note particularly that the general themes that have emerged appear to match those in the French activity “Thinking for the Petaflop”. It is important that we focus on understanding and considering the UK areas of strength within the context of these themes to make sure that investment and development build on them.

Cultural

Cultural issues around sharing

- Some application domain scientists are used to sharing models and codes, and reusing other people's software. For other domains this approach is almost completely alien with codes being entirely developed within a particular group and little use being made of libraries or other third-party software.

International boundaries/collaborations

- Many of the application groups have international collaborators or in some cases depend upon software developed in other countries (particularly the US) that may or may not continue to be supported. It is suggested that a map of international developments is created and a repository of information about ongoing activities is developed.

Development of a Community

- There was a general desire to have activities such as this workshop to develop more of a community across applications and across application/numerical analysis and computer science borders. Bringing together these interdisciplinary groups is very valuable and allows a transfer of knowledge from one field to another. A sequence of events and activities should be developed to assist in the communications across the community.

Applications and Algorithms

This section provides a high-level view of some of the issues and challenges for applications and algorithms; a more detailed section follows that will provide a more in depth consideration.

Cross application commonality

- The following section identifies and articulates the commonality of algorithms across the applications considered and the challenges for the future.

Integration across models

- Many applications involve multiple models at different scales or for different elements of the application. Bringing independent codes together can be difficult due to any number of issues – lack of standards for data models and formats, interoperability of programming models, and lack of knowledge of error propagation through the integrated system.
- Integration of application pipeline (e.g. CAD, simulation, visualisation) again these components often require different data formats and the like.
- In many applications there is a pipeline of activities: first, setting up the model; then the actual calculation; finally visualisation and analysis. A common concern was the lack of integration of the pipeline thus requiring a lot of effort to go, for example from the calculations/simulations to the analysis.

Error propagation across mathematical and simulation models

- It was recognised that there is a great deal to be understood regarding error propagation through a given model. This is compounded in the integration across models and pipelines.
- As architectures become heterogeneous there is also the need for algorithms that support mixed arithmetic of different precisions.

Adaptivity

- There is a need to have algorithms that adapt to problem characteristics and also architectural constraints. This may include dynamic algorithms that adapt at runtime and algorithms that might adapt according to experience.

Efficiency

- As architectures become more heterogeneous and components might be power hungry, there is a need to develop algorithms that are energy efficient.

Scalability

- As noted above, scalability is a huge problem for many application areas and a desire for all application areas. The desire to solve bigger problems faster is one of the main drivers of this community. Most applications do not scale beyond a few hundred processors, and this is widely perceived as inadequate as we move to petaflop-scale machines.
- As applications scale in size there is a need to develop underpinning algorithms that minimize communications to facilitate performance scaling.

Partitioning & Load balancing

- As systems become larger and more heterogeneous, load balancing and problem partitioning will be increasingly difficult. The need for load balancing may arise from the hardware, faults, and or the applications/algorithms requirements. Methods for dynamic concurrency generation and dynamic runtimes that default to a static model as needed will be required.

Data management

- As applications scale so too does the data be it analyzed data, output or other, and there are many issues around data distribution, replication, integration, integrity and security that need to be addressed. This includes management of metadata and ontologies.
- Ability to manage locality and data movement will be of increasing importance as memory hierarchies increase in complexity, making efficient use of bandwidth and scheduling for latency hiding will continue to be important.

Scalable i/o

- Input and output is important for applications not just in terms of writing out results but also in terms of enabling efficient and effective checkpointing. As applications scale to larger numbers of processors, this capability will become increasingly important.

Exemplar applications

- It is suggested that baseline models for a set of specified applications are developed to enable communication and benchmarking of new algorithms.

Software

Language issues

- There is a need for mixed language support: a variety of languages are used for application development. There is a need to consider how best to support this mixed language environment to allow better code re-use. This needs to allow composability, portability and support for standards.
- Similarly there is a need for sustainable software that through backward compatiblility provides interoperability.

Ease of Use

- Higher level abstractions should allow application developers an easier development environment. The provision of efficient, portable "plug-and-play" libraries would also simplify the application developers' tasks.

Efficiency and Performance

- Ability to manage locality and data movement and to schedule for latency hiding.
- Performance transparency and feedback providing the user with a layering of capability and tuning.
- Capability to control energy efficiency.

Support for development of software libraries and frameworks

- More effective code reuse is essential. This could be achieved by supporting software library development and frameworks for reuse.

Validation of software and models

- Many application developers are concerned that there are not well defined methods and techniques for validating scientific software and the underlying models. In some application areas observational data can play a role in validation, but for many this is not the case.

Software engineering

- It is often the case that application teams developing scientific software are not as skilled in software engineering as would be desired. Guidance on best practice for software engineering development would be a step to assist the community.

Standards and Compilers

- There is a need for standards to enable composability of models and it is clear that there will be a need for more sophisticated compiler and development suites. (The latter is likely to be an industry development.)

Active libraries & code generation

- In order to be able to move from one platform to another it would be beneficial to have underlying libraries that “do the right thing” for any given platform. This is becoming increasingly important with the plethora of new architectures that need to be considered.

Sustainability

There is general concern regarding the sustainability of application codes, software libraries and skills (we consider skills in the next section).

There is a need to develop models for sustainable software that might include

- Long term funding
- Industrial translation
- Open community support
- Other

The question of sustainability is also linked to the issues identified above in programming models and the need to maintain compatibility and interoperability.

Knowledge Base

Lack of awareness of existing libraries/packages

- It became clear through the workshops that there is patchy awareness of what is already available. It would be helpful to the community to develop mechanisms for collecting information on existing software and tools and disseminating effectively.

Skills and training

- All presentations at workshops mentioned skills in academic research groups and industry alike. There are simply insufficient students being trained with the required skills, mathematical, software engineering and high-performance computing. Approaches to this include MSC and graduate training, computational science internships and short courses or summer schools.
- As well as integrated approaches to high-performance algorithms it was noted that there were some specific areas such as optimization where there is scant education for graduate and postdoctoral researchers, but which is likely to be an area of increasing importance across a number of application areas.

Lack of awareness of expertise

- Providing a repository of expertise of numerical analysis and application domains in the UK may assist in developing appropriate teams for activities.

UK Strengths

The 2004 International Review of Mathematics [11] highlighted linear algebra, multiscale and adaptive algorithms, stochastic differential equations, preconditioning techniques and optimization as areas of strength in numerical analysis and scientific computing in the UK. In the workshops other areas were identified where groups in the UK are making international contributions; these include multi-scale and multi-level problems, approximation and neural networks.

The 2005 International Review of HPC highlighted a number of application areas where groups in the UK were internationally leading and specifically were developing codes that are being used worldwide. These included GAMESS-UK, a general purpose *ab initio* molecular electronic structure program; CASTEP, which uses density functional theory to compute the forces on the atoms and to simulate the time evolution ("dynamics") of molecular systems; DL_POLY, a general purpose molecular dynamics simulation package; MOLPRO, a system of *ab initio* programs for molecular electronic structure calculations with extensive treatment of the electron correlation problem; Chroma, a software system for lattice QCD calculations which is portable and efficient on a wide range of architectures (UKQCD Collaboration); and HiGEM, a new high-resolution integrated climate modelling code which includes atmospheric chemical influences. However the review also observed that there was a lack of integration of computational science and computer science and that the HPC research groups were "deficient in adapting to modern programming practices, anticipating the need for data management and visualization, and developing interoperable software and algorithm environments". Since the review new groups have formed, notably in Oceanography and biomedical simulations, who are also contributing to the wealth of application software and who are perhaps rather more integrated with other domains than indicated in the review.

Within the UK computer science community there are strengths in fundamental computer science research, algorithms, languages, and compilers as well as security, photonic materials and devices, and people and interactivity. But there is again a perceived weakness in the lack of integration with application areas and industry.

Applications and Algorithmic Challenges

Within the activity to date we have focussed on applications that are of interest or importance within the UK. There has been an attempt to identify new and upcoming application areas as well as those that are long established. A major issue that all the application areas face is scaling. There are generally four reasons that applications want to be able to scale:

1. A desire to include more realistic physical models. This implies higher resolution, more physical parameters and in general a greater complexity.
2. Simply to solve a larger problem with the same physical parameters.
3. A need to move to real-time simulation; to be able to solve the same problem but much faster.
4. A desire to complete far more time steps of the same simulation.

A good analysis of the requirements on a computer architecture, with respect to memory, storage and communications, is given in [6] for each of these categories. There it is also noted that the maturity of algorithmic approaches within applications, such as adaptive mesh refinement, mean a complexity in the application that did not exist with regular grids and the like. Similarly the complexity of the hardware with multi-core processors and vector units combined in 10s of thousands of processor units means that algorithms need to be designed to be efficient across a broad range of memory hierarchies and chip architectures.

Applications

Through the workshops and survey we have considered the following applications: Gas Turbine CFD Applications, Cosmological Hydro Simulations for Galaxy Formation, Multiscale Mathematical Models of the Heart, Spin dynamics for cancer diagnostics, Computational finance, MD Simulation of Complex Biomolecular Systems: Computational Challenge, ICOM (Imperial College Ocean model): 3D adaptive unstructured mesh ocean modeling, Industrial CFD for design (virtual engine), Biophysics: membranes with lateral phase separation, Materials science: species diffusion, EHL – Elasto-Hydrodynamic Lubricant simulation, Phase-field modelling (PFM) (solidification/crystallisation of molten metals), Chemical diffusion through skin (CDS), Astrophysics, and in more general terms Climate and Meteorology. There are a few application areas that we were not able to get input on to date and these include QCD, text mining and agent-based applications (the latter two were thought to be emerging HPC applications).

Considering the application portfolio on HECToR (see Annex 7) we feel that we have gathered data from a realistic set of applications that represent the main users of HPC in the UK. This on reflection may be a different set, or at least provide a different bias than those chosen to be of strategic importance to the wealth and health of the UK.

The general challenges for the applications considered were:

- Much bigger problems
  - high scalability essential
  - much better load balancing
  - performance overall issue
  - much larger data set sizes
- Parallelism
  - Hybrid parallelism: DMP & SMP
  - Hierarchical parallelism to map multi-level approaches
  - increase modularity: separation of computation and communication
  - Parallel I/O
  - Efficient one-sided communication
    - MPI-2 inadequate
    - Global array technolgies
  - Libraries abstracting multi-core architectures
- Hardware
  - Better use of multi-core technologies
  - GPUs and other novel architectures
  - Automatic mapping of algorithmic content to hardware/system
    - software cycle >> hardware cycle
  - Vectorisation
    - better use of SSE on Intel etc
    - other forms of vectorisation less useful (on the wane, in general)
- Error Analysis
  - Analysis of particular algorithms
  - Sensitivity/uncertainty analysis for problem
  - Error propogation across coupled models
  - Considerations of single vs. double vs. higher precisions, especially with GPU implementation
- Coupling of different codes
  - APIs?
- Multi-scale problems
- Multi-physics
- Better training
  - Tackling current dearth of HPC/NA specialists
- Physics consideration to drive problem size reductions
- Integration with post-processing and visualisation
  - standard interfaces for visualisation and analysis software
- Legacy provisions
- Improved validation and verification
- Long term managed support for libraries

Algorithms

Through the workshops and continuing discussions we have identified the key algorithmic areas that underpin the applications developed in the UK. These are:

- Parallelism
    - MPI (dominant)
    - Multithreading (incl. OpenMP) - very limited use
    - Hybrid/hierarchical - not used
- Multigrid
    - Algebraic Multigrid (AMG)
    - Classical MultiGrid
- Direct solvers
    - dense matrices
    - sparse matrices
- Iterative solvers (Krylov subspaces)
    - CG
    - BiCGStab
    - GMRES
- Poisson solvers
- Diagonalisation
    - dense eigenvalues
        - tridiagonalisation
        - QR algorithm (lack of parallel performance)
        - DC Divide-and-Conquer)
        - MRRR (Multiple Relatively Robust Representations)
        - bisection and inverse iteration
    - sparse eigenvalues
        - Davidson (Jacobi-Davidson)
        - Davidson-Liu
        - Symmetric subspace decomposition
    - SVD - dense and Lanczos (sparse)
- Preconditioners
- FFT
- PDE discretisation
    - FD (Finite Difference)
    - FE (Finite Elements)
    - FV (Finite Volume)
- Spectral methods (rare in all application areas at HPC/NA)
- Meshes
    - structured and unstructured
    - adaptive and adaptive refinement
- Domain decomposition
    - mesh partitioning
    - domain partitioning for particle dynamics
- ODE (mostly time-marching for PDEs)
    - explicit Runge Kutta 2nd to 4th order
    - implicit for stiff cases (unspecified techniques)

- Arnoldi propagators for TD-Schroedinger equation
- Particle dynamics
    - explicit short-range interactions
    - approximation for long range interactions (Ewald sum, FFT, etc)
    - Verlet algorithm
- Adjoint methods
    - data assimilation
    - sensitivity analysis
- Monte Carlo and quasi-Monte Carlo methods
    - Stochastic differential equations
- Random Number Generators
    - Currently, from standard numerical libraries (MKL, ACML, NAG)
- Optimisation –
    - Search (for local optima) is essentially sequential.
        - Parallelism is via
        - function and derivative evaluation
        - linear system solution
    - optimization often involves inequalities => needs its own (convex) analysis
    - Most real optimization problems are at least NP hard!
        - non-convex optimization
        - integer programming
        - global optimization
    - Optimization currently often uses implicit elimination of constraints
        - adjoints
        - inefficient optimization

We have also identified within the numerical algorithms area some challenges for the future that include:

- FFTs
    - More scalable alternatives to FFTs and convolution
- ODE
    - explicit algorithms likely to be favoured
- PDE
    - Better preconditioners for hyperbolic and elliptic operators
- Multigrid
    - Algebraic Multigrid (AMG) also as a preconditioner
- Meshing
    - Adaptive meshing
    - partitioning techniques for adaptive and moving meshes
    - good standard for mesh input/output
- Adjoint technologies for data assimilation and sensitivity analysis
- Sparse solvers
    - Direct - new, more efficient methods
    - Iterative
        - Better parallel preconditioners
        - Block Krylov methods
- R-Matrix technologies
    - splitting into inner-outer regions matching at interface
- Arnoldi propagator toolbox
- Partitioning and domain decomposition

  - better partitioning algorithms
  - bandwidth reduction
- Diagonalisation
  - sparse
    - block Krylov methods
    - better Davidson-like algorithms
  - dense
    - nonlinear problems
    - more scalable algorithms
- Fast Methods for dense matrices
  - H-Matrices (Herarchical Matrices)
  - FMM (Fast Multipole Methods)
- BLAS - efficient parallel BLAS (PBLAS)
- Optimization
  - Better polynomial methods for linear/convex quadratic programming
  - Polynomial approximations to NP hard problems
  - Scaling (or scale-invariant methods!)
  - Derivatives (automatic differentiation)
  - Good branching strategies
  - Good bounding strategies
  - Warm-starting
  - Semi-definite optimization (state of the art is small, systems are inevitably dense)
  - Better techniques than BFGS for molecular geometry optimization
- ALE - Arbitrary Lagrange Eulerian
- Particle dynamics
  - Better solution for mesh mismatchin for long-range interactions

In report [3] a number of key algorithmic issues were also identified. These included research into a new generation of algorithms that

- repeat calculations to avoid storing intermediate values, to minimize movement from DRAM
- will have appropriate mechanisms to allow fault-tolerance and resilience
- take account of the amount of power that will be consumed
- allow mixed precision computation
- include error detection and correction

and of course we need to have algorithms that will adapt to the heterogeneity of the system architecture.

Mapping Algorithms to Applications

Both [6] and [3] provide mappings of application areas to algorithmic requirements. Figure 1 shows the table from [6] which was developed through an analysis of the applications at Oak Ridge and using the seven dwarf classification of Colella - seven algorithmic tasks that he believed will be important for science and engineering for at least the next decade [9].

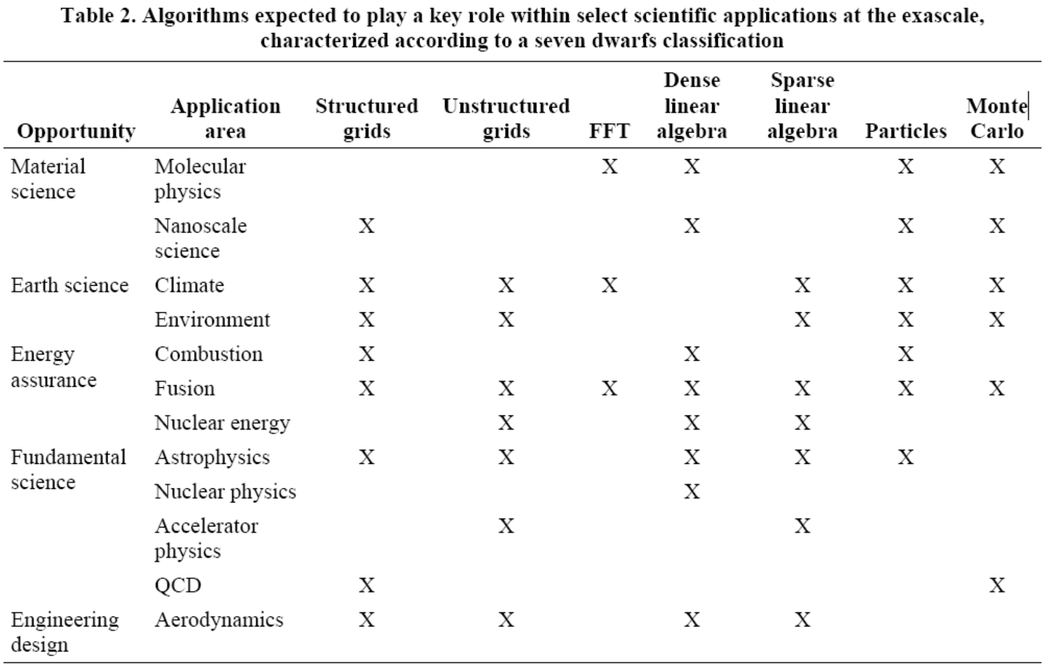


Table 2. Algorithms expected to play a key role within select scientific applications at the exascale, characterized according to a seven dwarfs classification

Opportunity	Application area	Structured grids	Unstructured grids	FFT	Dense linear algebra	Sparse linear algebra	Particles	Monte Carlo
Material science	Molecular physics			X	X		X	X
	Nanoscale science	X			X		X	X
Earth science	Climate	X	X	X		X	X	X
	Environment	X	X			X	X	X
Energy assurance	Combustion	X			X		X	
	Fusion	X	X	X	X	X	X	X
	Nuclear energy		X		X	X		
Fundamental science	Astrophysics	X	X		X	X	X	
	Nuclear physics				X			
	Accelerator physics		X			X		
	QCD	X						X
Engineering design	Aerodynamics	X	X		X	X		

Figure 1: Application analysis from [6], Scientific Application Requirements for Leadership Computing at the Exascale

As we can see from our own analysis we can learn much from this classification due to the overlap of application areas and therefore algorithmic requirements.

Figure 2 shows a similar analysis provided in [3] with its origin in the work of David Keoster & others [10]. Here they have provided a measure of the importance of the algorithm to the application by varying the shade of the box (see legend). Again the application areas have an overlap, not surprisingly, with our own and in this case are somewhat broader in their reach.

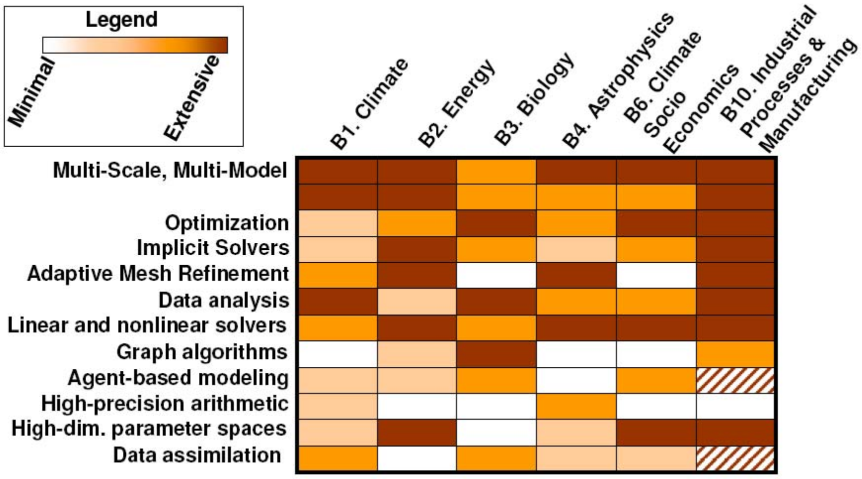


Figure 2: Application analysis from [3]

On the next page is a similar characterisation of the applications considered in this activity. We have mapped the algorithms required for the application area and we have also identified the software packages that are being used to support the application development.

It is notable that the majority of the software used is open source and very few packages or libraries in use are either “private” or commercial.   Many are developed in the US as part of the DOE, Darpa or NSF initiatives, which means that the UK is at risk to the continuing support from these sources.

Table 1: Application and Algorithm Overlap

	CFD/turbulence	nuclear fusion	galaxy formation	biomedical simulations	Biomolecular Systems	Spin dynamics	Industrial pdes	Computational Finance	Material science	Ocean model	Plasma physics	Computational Chemistry	Climate Modelling	Image Processing
Algorithms														
Adaptive Mesh Refinement	X		X	X			X			X	X			
Finite difference	X			X			X	X	X	X	X		X	
Finite Element	X			X			X		X	X	X			
Dense Linear Algebra	X			X		X	X		X			X	X	X
Sparse Linear Algebra	X	X		X		X	X		X	X		X	X	X
Evals/SVD	X					X	X		X			X		
Multigrid	X			X			X			X				
Preconditioners	X	X		X		X	X		X	X		X	X	
Domain Decomposition	X	X	X				X		X		X			
Monte Carlo		X					X	X						X
FFT		X	X						X		X	X		X
Particle dynamics		X	X		X									
Stochastic diff eqns				X				X						
Optimisation				X	X		X	X		X			X	
Software	HYDRA	CENTORI	FFTW	PETSc	VASP	GAUSSIAN	ARPACK	Many	DL-POLY3	FLUIDITY	GKW	NAMD	NAG*	FFTW
	ParMETIS	GS2	GADGET2	MUMPS	GROMACS	ADF	FFTW	ISVs	CRYSTAL	HYPRE	GS2	DLPOLY3	LAPACK	VXL
	OPLUS	ORB5	PARAMESH	OpenCMISS*	NAMD		LAPACK	providing	CASTEP	PROMETHEUS	EPOCH	CHARM++	UNIRAS	
	CGNS	MCNP	HDF5	SOFA	DESMOND*			software	FFTW	TERRENO	Osiris	ScaLAPACK		
	HDF5		FLASH	LIFEV					LAPACK	PETSc		PRMAT		
	PADRAM			OPENFEM					SCALAPACK			CRYSTAL		
	ALBERTA			BLAS								GAMESS-UK		
	PETSC											Molpro		
	LAPACK											Peigs		
												PLAPACK		
												SIESTA		

General Challenges

We have captured most challenges within the application or algorithmic domain but there are a number that are more general and they are usually to do with the programming environment. Most of these have been articulated in the section on algorithms and applications within the roadmap themes, however one that is an urgent issue and not yet addressed is that of novel architectures. Novel architectures such as FPGAs, GPUs and multicore processors are increasingly the norm and unfortunately at this time the programming environments and the software stack to support applications is sadly lacking. Many of the issues identified above such as mixed arithmetic and the like apply equally to these platforms, but their rise to prominence has brought into focus the need for a concerted effort to develop algorithms and application software that can run effectively on them.

A further general area that has not been dwelt on is that of data. In some application areas researchers are drowning in a data deluge and this has a number of implications in the context of this roadmap; data management and storage, I/O and other data related areas need as much attention as that of the computational algorithms; visualization is of increasing importance as understanding and analyzing the increasing complex data becomes increasingly difficult; planning for data infrastructure alongside computational infrastructure will be critical.

Implementing a Roadmap

In the sections above we have identified a number of requirements and challenges. In this section we will try to provide these in the context of a timeframe of what might be achieved and a plan according to the themes that evolved.

As identified above there are several themes that need to be addressed. We have provided an overview of those and the instruments we feel might be applied to these themes in the table below.

Table 2 Instruments for Roadmap Implementation

Theme /Activity	Culture	Apps & Algorithms	Software Development	Knowledge Base	Sustainability
EPSRC Network	x	x		x	x
APACE Website	x	x	x	x	x
HPC Software Development Calls	x	x	x		x
Exemplar Apps		x	x	x	x
IDEAS Factory		x	x	x	x

The “APACE” website (http://apace.myexperiment.org/) is an activity underway that will provide a community base for anyone interested in this area. It is planned to provide a way to link together algorithms and applications and to collect requirements for both whilst at the same time sharing information about existing activities.

In terms of a specific timeline for activities we have put together the following suggested timeline.

Recommendations

Note there are existing activities including the CSE support in annex 7 where some of the near-term issues are being solved as well as the applications get first-hand input from numerical analysts. Th recommendations here assume that those activities are continued.

Recommendation 1: The network of HPC applications, numerical analysis and computer scientists within the UK should be facilitated using the instruments as indicated in table 2. This network should provide communication of best practice across the UK community and provide a common interface to international activities. It should also provide mechanisms for raising awareness of existing applications, algorithms software and activities.

Recommendation 2: Key application exemplars should be developed.

Recommendation 3: There needs to be continued investment in new algorithm development to underpin existing applications move to the new architectures and to enable new applications. Table 1 should provide an indication of the most important algorithms in the sense of impact across a number of application areas.

Recommendation 4: There needs to be a continuted engagement of computer scientists within key application areas and support fundamental computer science research on abstractions, code generation and adaptive software systems and frameworks for reuse within the context of those applications.

Recommendation 5: Sustainability of software should be addressed through the development of models of sustainability including collaboration with software foundation, industry and international activities. This will require an initial investment from EPSRC and other research councils and funding agencies.

Recommendation 6: There is a need for a more joined up approach for skills training and education. There are a number of disparate courses around the UK that could be leveraged to provide a better platform for graduate students. A DTC in this area would also provide a more comprehensive consideration of the cross-discipline requirements. Key areas have been identified where there is a clear lack of education a particular example is optimisation.

Conclusions

This report is based on a relatively short activity with limited support. It has been successful in bringing together groups of researchers to provide this initial input into a high-performance computing applications and algorithms roadmap. We recognise that there is a need to continue the discussion between groups to provide further input to the evolving roadmap and will aim to facilitate that continuing community development.

Acknowledgements

We are grateful for the support provided by EPSRC for the algorithm/application roadmapping activity. We also want to acknowledge the work of Mark Hylton and Stef Salvini without whom this activity would not have succeeded.

References

1. Report to the President: *Computational Science: Ensuring America's Competitiveness*, President's Information Technology Advisory Committee, 2005.
2. A Report of the NSF Blue Ribbon Panel on Simulation-Based Engineering Science, *Revolutionizing Engineering Science through Simulation*, 2006.
3. ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems, www.sdsc.edu/~allans/Exascale_final_report.pdf*, September 2008*
4. DOE Report on Modeling and **Simulation** at the Exascale for Energy and the Environment, http://www.sc.doe.gov/ascr/ProgramDocuments/Docs/TownHall.pdf, June 2007
5. US DOE SciDAC activity - http://www.scidac.gov/missionSD2.html
6. Scientific Application Requirements for Leadership Computing at the Exascale, ORNL/TM-2007/238, http://www.nccs.gov/wp-content/media/nccs_reports/Exascale_Reqms.pdf, Dec 2007
7. Mathematics at the Interface of Computer Science and Industry, the Smith Institute for Industrial Mathematics and System Engineering, 2005.
8. Science and Innovation Framework (2004-2010) http://www.hm-treasury.gov.uk/spending_review/spend_sr04/associated_documents/spending_sr04_science.cfm
9. P. Colella, "Defining Software Requirements for Scientific Computing," DARPA HPCS presentation, 2004.
10. Frontiers of Extreme Computing 2007, Applications and Algorithms Working Group, http://www.zettaflops.org/fec07/presentations/Thursday-1130-Zetta-apps-r7.pdf, October 2007
11. International Review of Mathematics, http://www.cms.ac.uk/irm/irm.pdf, March 2004
12. International Review of Research Using HPC in the UK, http://www.epsrc.ac.uk/CMSWeb/Downloads/Other/HPCInternationalReviewReport.pdf , December 2005
13. The International Exascale Software Project: A Call to Cooperative Action by the Global High Performance Community, Jack Dongarra, Pete Beckman, Patrick Aerts, Frank Cappello, Thomas Lippert, Satoshi Matsuoka, Paul Messina, Terry Moore, Rick Stevens, Anne Trefethen, Mateo Valero

Annex 1: Individuals and Groups who have given input

Contributor	Institution
Cliff Addison	University of Liverpool
Mark Allan	BAE Systems
Jamil Appa	BAe Systems
Tony Arber	University of Warwick
Wayne Arter	UKAEA
Bruce Boghosian	Tufts University
Garfield Bowen	Schlumberger
John Brooke	University of Manchester
Kevin Burrage	University of Oxford
Ian Bush	NAG Ltd
Stewart Cant	University of Cambridge
Anthony Ching Ho Ng	Smith Institute/NAG
George Constantinides	Imperial College London
Peter Coveney	UCL
Sue Dollar	STFC Rutherford Appleton Laboratory
Iain Duff	STFC - Rutherford Appleton Laboratory
Mihai Duta	Oxford University
Massimiliano Fatica	NVIDIA
Gregorz Gawron	HSBC IB
Mike Giles	Mathematical Institute
Mike Gillan	UCL
Jacek Gondzio	University of Edinburgh
Christopher Goodyer	University of Leeds
Gerard Gorman	Imperial College London
Nick Gould	RAL, STFC
Ivan Graham	University of Bath
John Gurd	University of Manchester
Matthias Heil	University of Manchester
Nick Higham	University of Manchester
Mark Hylton	Oxford University
Peter Jimack	University of Leeds
Emma Jones	EPSRC
Crispin Keable	IBM
Steve Kenny	Loughborough University
Michel Kern	INRIA, France
Igor Kozin	STFC Daresbury
Ilya Kuprov	University of Durham
Leigh Lapworth	Rolls-Royce Plc
Charlie Laughton	University of Nottingham
Ben Leimkuhler	University of Edinburgh
Zoe Lock	Technology Strategy Board
Osni Marques	LBNL & DOE, USA
Milan Mihajlovic	University of Manchester
Maziar Nekovee	BT Research & UCL

Bruno Nicoletti	The Foundry
Duc Nguyen	Culham Science Centre, UKAEA
Ross Nobes	Fujitsu Laboratories of Europe
Stephen Pickles	STFC Daresbury
Matthew Piggot	Imperial
Ram Rajamony	IBM Research, Austin
Ben Ralston	AWE
Graham Riley	University of Manchester
Colin M Roach	UKAEA
Sabine Roller	HLRS, Stuttgart
Radhika Saksena	UCL
Stef Salvini	Oxford University
Mark Sansom	Oxford
Rob Scheichl	Bath
Stan Scott	Queen's University Belfast
Jennifer Scott	STFC - Rutherford Appleton Laboratory
Gareth Shaw	Schlumberger
Paul Sherwood	STFC Daresbury Lab
David Silvester	University of Manchester
Nic Smith	Oxford
Edward Smyth	NAG
Björn Stinner	University of Warwick
Kevin Stratford	University of Edinburgh (EPCC)
Andrew Sunderland	STFC - Daresbury Laboratory
Ken Taylor	Queen's University Belfast
Tom Theuns	University of Durham
Anne Trefethen	Oxford University
Philip Treleaven	UCL
Jan Van lent	University of Bath
Dominic Walsh	Schlumberger
David Worth	STFC - Rutherford Appleton Laboratory
Zahari Zlatev	Aarhus University

Annex 2: Methodology for this Roadmap

Many applications share a common numerical algorithmic base. We aim to capture the elements of this common base, identify the status of those elements and in conjunction with the EPSRC Technology and Applications roadmapping activity, determine areas in which the UK should invest in algorithm development.

A significant sample of applications, from a range of research areas, will be included in the roadmapping activity. The applications chosen will include those in the EPSRC Technology and Applications roadmap, and others that represent upcoming and potentially new HPC areas.

The roadmapping activity consists of a set of subactivities:

1. Ground work: Evaluation of current situation (general activity)
2. Detailed discussion of numerical aspects, current and future in specific applications (workshop 1)
3. Definition of a road map for future development (workshop 2)
4. Iteration on Road map definitions and requirements (workshop 3)
5. Community Engagement

Each of the following three sections will briefly deal with each of the activities. Finally, the last section will introduce some ideas which could be of use in classifying numerical algorithms and the applications they are used in.

The Access Grid and other Web-based technologies could naturally be used to widen the attendance and interaction to wider audiences. The establishment of a collaborative forum, easily accessible from the Web or other means should also be seen as a primary target of this work.

Ground work

Some initial grounding work will be done to evaluate the current situation.

Application development is an extremely time consuming activity. Many applications have extremely long life spans, certainly much longer than the platforms on which they were initially deployed. That makes the tracking of applications to developing HPC platforms as well as to numerical algorithms a very critical, but an all too often *ad hoc* activity.

This activity aims to elucidate aspects of the current status

- Classes of algorithms used, their importance and centrality in the applications considered
- Current requirements and observed limitations in desired achieving scientific aims. This would, of course, include a study of current numerical and computational performance.
- Analysis of current algorithmic content delivery methodologies, their strength and limitations

Classifying Algorithms and Applications

The algorithmic content of applications should be quantifiable with respect to a number of parameters. This, however, must also consider the context in which algorithms appear. For example, a parallel version of an algorithm may be required in one application, while multiple versions of the equivalent serial algorithm running concurrently may be employed in a different application.

At the same time, the relative importance of the classifying parameters must, of course, reflect the current and future technological trend as well as the suitability of particular numerical solutions for their future deployment.

For example, criteria could include:

1. Range of applicability, in the sense of how widely distributed these algorithms are across the applications considered, and how important they will be for any future developments.
2. Scalability. This should be seen both as a function of performance vs. number of processing units as well as a function of required performance vs. problem size.
3. Parallelism type
   - Pure message passing
   - Pure SMP
   - Hybrid (message passing + SMP)
   - Flexible (any of the above, reconfigurable at will)
4. Memory access properties. In most current architecture, data re-use carries a considerable performance premium given the layered structure of memory access. This situation is unlikely to change in the near future, at least for most architectures being developed.
5. I/O vs. computationally-bound algorithms.
6. Existing equivalent/alternative algorithms and their properties.
7. Deployment issues
8. Complexity of development
9. Testing
10. Ease of use across several applications
11. Language and other computational issues
    - Development language
    - Ease of cross-language usage
    - Delivery: package/library structure

A careful analysis of the outcome of this classification would allow a more focused approach to the roadmap.

Annex 3: Workshop 1

Overview

Seventeen application areas were discussed by forty three workshop attendees over three days. The agenda for the workshop is provided in below together with the attendee list. The attendees came from a mix of academic institutions, research laboratories and industry.

The workshop provided short presentations by application experts with break out groups to consider the questions provided on a Proforma (below). In this report we have attempted to bring together all pertinent suggestions and ideas from the presentations and from the discussion groups. A synopsis of all the presentations has been provided in Annex 7 that identifies the key messages from the presentations.

Algorithms Identified and challenges for the future

The general algorithmic areas include:

- Scalable FFT
- Mesh refinement
- Eigenvalue/eigenvector (all or few)
- Optimisation
- Iterative & implicit solvers
- Visualisation
- High-performance one sided communications
- Out of core algorithms to enable larger problems

The major issues for the future were seen to be:

1. Load balancing
   - ➢ meshes
   - ➢ particle dynamics and computation of interactions
2. FFTs and Poisson solvers
3. Sparse and dense diagonalisation
4. Sparse and dense linear solvers
5. Use of novel architectures (in the immediate future)
   - ➢ FPGAs
   - ➢ GPUs
   - ➢ IBM Cell
   - ➢ Clearspeed
6. domain decomposition
7. Coupling between different codes
8. Meshes
   - ➢ generation of accurate surface mesh
   - ➢ partitioning

At the next workshop we hope to determine where these lie on the roadmap in the context of existing activities and further applications input.

Further Application Areas and Other Requirements

Although we were able to include a broad set of applications in this workshop we have identified some key application areas that have not been captured here. These include:

- NERC (Climate, Met etc)

- Acoustics & Electromagnetics
- QCD
- Microfluidic flows
- Astrophysics
- Biology & systems biology
- Large scale agent simulations
- Text mining
- Data mining and statistics
- Digital signal processing (compression)
- Complex networks
- Medical sciences
- Genome Sequencing

There is also a need to make a collection of information on existing software libraries and activities across the application areas and numerical software.

This workshop did not include any vendors' presentations or inputs. As we move forward it will be important to bring in vendors to ensure that we understand their development directions both in terms of hardware and software environments.

It was suggested that model application codes that can be used as a baseline by algorithm developers would be helpful in providing an effective collaborative framework.

Conclusions for Workshop 1

The first workshop was very successful in bringing together an initial set of application scientists, numerical analysts and computer scientists. It has provided a base set of information on which we can build. It is clear, however, that there is much to be done if we are to succeed in developing a first instantiation of the roadmap in the next four months. We can only do this by engagement of the wider community and we ask anyone reading this report to get engaged and provide input on the material presented here or indeed any element related to this activity.

Workshop 1 timetable

HPC/NA Workshop 1: Applications: underlying algorithms and challenges for the future

Oxford e-Research Centre

Day 1: Wednesday 5th November

11:00 Welcome

Introductions & aims of workshop by Anne Trefethen, OERC

Morning Sessions chaired by Iain Duff, STFC

11:15-13:15 **Large Scale Simulation**

- Stewart Cant, Cambridge
- Leigh Lapworth, Rolls Royce
- Wayne Arter, UKAE
- Tom Theuns, Durham

Group discussions

13:15-14:00 Lunch

Afternoon Sessions Chaired by Nick Higham, Manchester

14:00-15:30 **Biomedical Simulation**

- Nic Smith, Oxford
- Mark Sansom, Oxford

Group Discussions

15:30-17:30 **Computation for Physical Sciences Session I**

- Ken Taylor, QuB
- Ilya Kuprov, Durham
- Ivan Graham & Robert Scheichl, Bath

Group Discussions

17:30-18:00 Plenary review & discussion of first sessions
18:30 Pre-dinner drinks, St. Hugh's College
19:00 Dinner, Wordsworth Room, St Hugh's College

Day 2: Thursday 6th November
09:00 Welcome to the workshop & summary by Anne Trefethen, OeRC
Morning Sessions Chaired by Iain Duff, STFC
09:30-10:45 **Computational Finance**

- Mike Giles, Oxford

Group Discussions
11:30-13:00 **Computation for Physical Sciences II**

- Chris Goodyer, Leeds
- Björn Stinner, Warwick
- Ian Bush, NAG

Group Discussions
13:00-14:00 Lunch

Afternoon Sessions chaired by Anne Trefethen, OeRC
14:00-16:00 **Oceanography and Climate Modelling**

- Gerard Gorman, Imperial

Group Discussions
16:30-17:30 Review & discussion of day 2
18:30 Pre-dinner drinks, St. Hugh's College
19:00 Dinner, The Boardroom, St Hugh's College

Day 3: Friday 7th November
09:00 Welcome to the workshop & summary by Anne Trefethen, OeRC
Morning Sessions chaired by Nick Higham, Manchester
Developing a high performance computing / numerical analysis roadmap

09:30-11:30 **Computation for Physical Sciences Session III**

- Tony Arber, Warwick (Physics)
- Paul Sherwood (Comp Chem)
- Cliff Addison, Liverpool

Group discussions
11:30-12:30 Review of workshop and outputs
12:30-13:30 Lunch

Discussion – first thoughts for roadmap
Plans for second workshop, Manchester December 2008
15:00 Close

Synopsis of Presentations for Workshop 1

Session 1 – Large Scale simulations -

Title Gas Turbine CFD Applications and Challenges for the Future

Name and Affiliation Leigh Lapworth, Rolls Royce

Subject

ACARE Environmental Goals for 2020:

- Reduce fule consumption by 50%
- Reduce external noise by 50%
- Reduce NOX by 80%

All development in-house but combustion

Current Packages, Libraries, Tools etc

HYDRA

- CFD Solver
- Steady and unsteady flow
- Hybrid unstructured mesh; moving mesh
- Parallel on DMP and SMP systems
- Linearised, unsteady and adjoint CFD capabilities
- Libraries: OPLUS, CGNS, ParMETIS, HDF5
- F77 mostly
- RANS (Reynolds Averaged Navier Stokes); LES (Large Eddy Simulation)

PADRAM

- Multiblock structured and unstructured mesh generator
- All geometry and meshing parametric

Current Algorithmic Requirements

- MultiGrid (HYDRA)
- Preconditioners (HYDRA)
- Partitioning (HYDRA through ParMETIS)

Future Developments, Requirements, Challenges and Issues

- Virtual Engine
  - Multi-physics
  - Different models (RANS for compressor, LES for combustor; RANS for turbine
- Coupling of different codes (Multicode)
- Industrial CFD code development
  - Much increased number of nodes
- Hardware
  - Better use of multicore: through an API?
- Software shelf life >> hardware shelf life
- Recruitment an issue: not enough HPC specialists available

Title Computational Fluid Dynamics

Name and Affiliation Stewart Cant, Cambridge University

Subject

Review of CFD current state and challenges

- Multi-scale: cannot resolve all scales at reasonable cost: turbulence models required
- Major limitation: all practical flows are turbulent
- Complex geometries
  - CAD data format, cleanup and repair issues
  - Surface meshing accurate (by hand); volume meshing hopefully automatic
  - Visualisation and post-processing

- Main techniques
  - DNS: Direct Numerical Simulation of the Navier-Stokes equations
  - Large Eddy Simulation (LES): actively developed – modelling required at sub-grid level
  - Reynolds Averaged Navier Stokes (RANS) – average the governing equations – model all scales – inexpensive hence standard approach
- Numerics
  - Finite Volume – standard in almost all CFD, second order accurate
  - Finite Differences – for high-order accuracy in DNS
  - Finite Elements – rare in FD (common in structural eng.)
  - Spectral Methods – rare, turbulent research
- Solution Algorithms
  - Incompressible flow
    - Poisson equation
    - Conjugate Gradient, MultiGrid
    - Semi-implicit in time
  - Compressible flow
    - Density-based time-marching
    - Explicit integrators (Runge-Kutta) 2nd to 4th order
  - DNS
    - Explicit Runge Kutta 3rd to 4th order
- Performance
  - Critical issue (e.g. LES)
  - MPI/Linux basic technology

Current Packages, Libraries, Tools etc

Current Algorithmic Requirements

Future Developments, Requirements, Challenges and Issues

- Parallelism
  - Distributed memory: minimise global operations
  - Little scope for vectorisation techniques
  - FFT is dead, Poisson solvers are struggling
- Explicit algorithms are favoured
- Exploit synergy between DNS/RANS/LES
- LES requires a major effort to achieve robustness
- Numerics
  - Adaptive meshing
  - Optimal time-stepping
  - Parallel tools for all tasks: CAD → mesh → solution → postprocessing
- Non-standard processors: Cell, GPU, etc

Title Algorithms for Nuclear Fusion Plasma Physics

Name and Affiliation Wayne Arter, EURATOM/UKAEA, Culham

Subject

Tokamak modelling (MAST,JET,ITER)

Current Packages, Libraries, Tools etc

CENTORI

- Fluid model

GS2

- Gyrokinetic phase-fluid model

ORB5

- Gyrokinetic particle (trajectory) model

MCNP

- Monte Carlo neutron transport (for nuclear fusion safety analysis)

Current Algorithmic Requirements

(Gyrokinetic unless stated)

- FFT (exchange between different meshes) (fluid and phase-fluid)
- Domain decomposition (phase-fluid and particle)
- Matrix splitting, solvers and preconditioning (phase-fluid and particle)
- Particle tracking on mesh (particle)
- Particle tracking/ray tracing through geometry (neutrons)

Future Developments, Requirements, Challenges and Issues

- Good preconditioners for hyperbolic and elliptic operators
- Better FFTs and domain decomposition, multiscale generally
- Visualisation

Title Cosmological Hydro Simulations of Galaxy Formation: physics and numerics

Name and Affiliation Tom Theuns, Durham University

Subject

Simulation of the time evolution of galaxies (formation etc)

Current Packages, Libraries, Tools etc

GADGET2

- Particle dynamics model
- Time-steps depending on particle (saving on force evaluation)
- Domain decomposition
- Compute explicitly forces between nearby objects
- FFT techniques for objects between far objects (approx)
    - Load imbalance; different distribution required than mesh for particles (memory layout issues)
- MPI-parallel
- No multithreading
- Extensively ported
- Load imbalance for systems with large dynamic range

FLASH

- Block-structured adaptive mesh
- Eulerian hydrodynamics
- MPI-parallel
- Domain decomposition

FFTW

- In GADGET2

PARAMESH

- In FLASH

HDF5

- Parallel I/O

Current Algorithmic Requirements

Future Developments, Requirements, Challenges and Issues

- Bigger simulations
- Multi-physics
- Greater dynamic range
- More physics
- Training
- Legacy provision
- Improved validation/verification
- Increase modularity (separate comp. and comm..)
- Standard interfaces (“I/O”) for visualisation and analysis software
- Much increased data sets

Session 2 – Biomedical Simulation

Title Developing Multiscale Mathematical Models of the Heart

Name and Affiliation Nic Smith, Oxford University

Subject

Multi-scale, multi-physics modelling of the human heart

- Coupling of
  - Mechanical heart simulation
  - Heart fluid-flow
  - Electro-stimulation
  - Coronary blood flow
- 15 orders of magnitude between molecular → cell → macroscopic levels
- Analysis → medical diagnosis and intervention in the future

Current Packages, Libraries, Tools etc

- Mostly in-house development.
- PETSc
- MUMPS (direct solver)
- A number of other packages/libraries listed in one slide, but not mentioned otherwise (OpenCMISS, SOFA, LIFEV, OPENFEM: Finite Elements; GIMIAS: visualisation; CMGUI: data assimilation?)
- List of numerical techniques given in the same slides but not mentioned elsewhere (FEM, FD, POD, ALE)
- Paralllelism currently limited to 64 processors? (from talk)

Current Algorithmic Requirements

Future Developments, Requirements, Challenges and Issues

- Validation (data and codes)
- Multi-physics
- Multi-scale
- Load balancing
- Parallel I/O
- Visualisation
- Re-engineering codes and models for new architectures
- Adaptive meshing and MultiGrid to cut down the computing requirements

Title MD Simulation of Complex Biomolecular Systems: Computational Challenges

Name and Affiliation Mark Sansom, Oxford University

Subject

MD (Molecular Dynamics) simulation of complex biomolecular systems:

- Cell membrane transport mechanisms (proteins embedded in lipid bilayers of lipids)
- From experiments, only static structure of proteins – simulation essential
- “particle dynamics” type of approach

Current Packages, Libraries, Tools etc

- GROMACS
- NAMD
- DESMOND (industrial/commercial application)

Current Algorithmic Requirements

- Newtonian physics
- Verlet algorithm for Time-Dependent (TD) integration
- Bottleneck: long range interactions (particularly electrostatic)
- Load balancing (need more)
- Approximation by “clustering” portions of molecules (rather than atomic level)
- Emerging architectures must be considered: GPUs, Clearspeed, Anton from DE Share

Future Developments, Requirements, Challenges and Issues

- Multi-scale modelling: quantum mechanics → MD atomistic (hybrid?) → MD
- Very large systems
  - Load balancing
  - Large amount of data and technologies required for storage and access
  - Large-scale visualisation
  - Multi-level integration: how?
  - Hybrid systems (very difficult)

Session 3 – Computation for Physical Science I

Title Solving time-dependent high dimensional PDEs on MPP architectures

Name and Affiliation Ken Taylor, Queen’s University of Belfast

Subject

Modelling electron dynamics in atoms/molecules exposed to high intensity laser light to complement laboratory studies. Allows investigation on the Atto-second (10^{-18} s) time-scale, fundamental to electronic motion in atomic/molecular systems. The Ti-sapphire laser is the laboratory “WORK-HORSE” for such studies but complementary theoretical work on just the two-electron atom helium demands the full power of HECToR. The time-dependent Schrödinger Equation describing the particular atom/molecule in the intense Ti:sapphire laser pulse is the high-dimensional PDE.

Current Packages, Libraries, Tools etc

- HELIUM
  - Extremely large memory and data transfer requirements
  - Fortran 90
  - MPI-parallel
  - Finite-difference methods

Current Algorithmic Requirements

- High efficiency on MPP systems (minimum communication overheads)
- Arnoldi propagator for accuracy in time-stepping
  - Extremely large problems
  - Very small Krylov subspace required for time-propagation
- Domain decomposition upper triangular part of two-electron radial space
  - Only nearest-neighbour communication

Future Developments, Requirements, Challenges and Issues

- Very important to widen application to other atoms/molecules, by partitioning space into:
  - OUTER region, where HELIUM two-electron finite-difference methods apply
  - INNER region, where a full multi-electron basis set description is possible and already largely coded for traditional collisional work
- Multi-electron systems would lead to yet bigger problem sizes; INNER and OUTER must be Arnoldi propagated simultaneously and careful load-balancing would be needed over these regions.

Title Spin Dynamics: from sparse matrices to cancer diagnostics

Name and Affiliation Ilya Kruprov, Durham University

Subject

Spin-selection in chemical reactions in the presence of magnetic fields.

- Bird navigation: chemical receptors affected by Earth magnetic field

Ilya made a strong case for physics considerations to make large problems tractable

Current Packages, Libraries, Tools etc

- Gaussian
- ADF

Current Algorithmic Requirements

- Parallel diagonalisation of large systems
  - Eigenvalues
  - SVD
  - Dense and Lanczos (Davidson for smallest eigenvalues) methods
- Restriction of basis set cause very large reduction in problem size
  - Physics driven: not through NA
  - Polynomial scaling algorithm

Future Developments, Requirements, Challenges and Issues

- Standard Arnoldi propagator toolbox for TD problems

Title Future algorithm and software development in industrial applications of PDEs

Name and Affiliation Ivan Graham, Bath University

Subject

Overview of the requirements for complex engineering/biological processes modelling using PDEs

- Often huge ill-conditioned systems
- Additional complications
    - Data uncertainty
    - Multiscale
    - multiphysics

Current Packages, Libraries, Tools etc

- ARPACK
- FFTW
- BLAS, LAPACK

Current Algorithmic Requirements

- Path following techniques
- Adaptive FE for complex flows
- Preconditioners for high Reynolds numbers
- Iterative methods for smallest eigenvalues of large sparse systems (inexact solvers)
- Multigrid
- Parallel domain decomposition preconditioners
- Matrix-free inverse power method
- Monte Carlo methods
- Geometric tracking algorithms

Future Developments, Requirements, Challenges and Issues

- Multi-physics
- Multi-scale
- Algebraic MultiGrid (AMG) preconditioners
- Fast methods for dense matrices
    - H-Matrices (Hierarchical matrices)
    - FMM (Fast Multipole Method)
- Iterative methods for dense system (preconditioning)
- Robust computation of (4D) oscillatory integrals

Session 4 – Computational Finance -

Title Computational Finance

Name and Affiliation Mike Giles, Oxford University

Subject

Overview of current HPC use in the financial world

- Very large number of systems used for financial modelling (centres: New York and London)
- Throughput requirements not capacity computing
- “brute force” approach
  - Limited search for efficient algorithms
  - Buy bigger systems, cost no objection
  - Important metrics
    - Quick deployment
    - Easy modification
    - People expensive, hardware cheap
- Multi-task “trivial” parallelism
  - Large number of independent small serial jobs
  - Handled by specific tools (see below)
- Other parallelism
  - MPI: limited use
  - OpenMP: limited use? (to increase with Multicores?)
- People involved
  - Traders – new financial products through scripting languages
  - “Quants” (quantitative analysts) – many PhD in sciences, develop the models and write codes
- New technologies
  - Not of great interest to quants – limited development
  - FPGAs (of minor interest)
  - GPUs
  - IBM Cell

Current Packages, Libraries, Tools etc

- Data Synapse – low latency distributed task submission system
- Symphony (Platform Computing) - low latency distributed task submission system
- Tier 1 banks
  - Software developed in house
- Tier 2 banks
  - Software from ISVs (Algorithmics, SunGard, SciComp)
- Standard RNG from vendors’ libraries (Intel, AMD, NAG)

Current Algorithmic Requirements

- Monte Carlo simulation (60%)
- Finited Difference methods (30%) (currently, no FE)
- Semi-analytic methods involving FFTs (10%)
- Stochastic Differential Equations (SDE)
- Answer to AET’s question: optimisation not important
  - Only some classical optimisation for calibration purposes

Future Developments, Requirements, Challenges and Issues

- Increasing complexity of models
  - Increasing importance of MC methods

- Many more calculations
  - More “stress” tests required by regulators, calibrations and sensitivity
  - Costs and power consumption starting being an issue
- Multicores
  - Better use of SSE vectorisation
  - OpenMP or multithreading
- GPUs
  - “Easy” for MC methods
  - Much more difficult for FD

Session 5 – Computation for Physical Science II -

Title Parallel Implementation and Application of Adaptive and Multilevel Numerical Algorithms

Name and Affiliation Chris Goodyer, Leeds University

Subject

An overview of HPC NA activities at Leeds concentrating on three applications

- EHL – Elastohydrodynamic lubricant simulation
  - MPI-parallel
- Phase-field modelling (PFM) (solidification/crystallisation of molten metals)
  - Parallel version is in development
- Chemical diffusion through skin (CDS)
  - MPI-parallel

Also, research into

- Multi-level techniques
- Mesh adaptation
- Adjoint methods

Current Packages, Libraries, Tools etc

- PARAMESH - Parallel mesh generator (PFM)
- Netgen (CDS) - Mesh generator not suitable for large meshes
- TETRAD (CDS) – mesh refinement
- SPARSKIT (CDS) – sparse solvers for serial code
- PETSc
- Metis (sparse matrix reordering)
- gViz for interface to visualisation framework

Current Algorithmic Requirements

- EHL
  - Regular Grid FD
  - High order Discontinuous Galerkin Fes
  - Multigrid – MLAT, FAS
  - Geometrical decomposition
  - MPI-parallel
- PFM
  - FD

    - Continuous FE Solvers
    - Multigrid – MLAT, FAS
    - Implicit, stiff ODE solvers (time)
- CDS
    - FE solver
    - Mesh: periodic, anisotropic, 3-d unstructured tetrahedral
- Global error estimation through adjoint

Future Developments, Requirements, Challenges and Issues

- Better parallel preconditioners (CDS)
- Bandwidth reduction, particularly for periodic domains (CDS)
- Combining DMP and SMP parallelism (multicores)
- Hierarchy of parallelism
- Partitioning and load balancing
    - General not just geometric
- Multi-level algorithms to map onto hierarchical hardware
- Automatic mapping of software to hardware
- Plug-and-parallelise libraries
- Inter-application communication (APIs?) for multiscale multiphysics problems
- Long term managed support for libraries

Title PDEs on Moving Surfaces - Computational Issues

Name and Affiliation Björn Stinner, Warwick University

Subject

Free boundary problems

- CFD: surface active agents in two-phase flow
- Biophysics: membranes with lateral phase separation
- Materials science: species diffusion

Coupled surface + bulk problems

Current Packages, Libraries, Tools etc

- ALBERTA – Adaptive Hierarchical FE Toolbox
- PETSc (in ALBERTA)
- BLAS, LAPACK

Current Algorithmic Requirements

- FE on moving polyhedral surfaces
- Level set approach and phase-field method with unfitted bulk FE
- Gauss-Seidel iteration
- Krylov subspace methods (GMRES, CG, BiCGstab) with preconditioning
- Monotone and classical Multigrid

Future Developments, Requirements, Challenges and Issues

- Efficient parallel BLAS (PBLAS)
- Efficient direct solvers?
- Parallel iterative solvers
    - In answer to question: block Krylov iterative solvers
- Parallel assembling

- Parallel MultiGrid with information about mesh

Title Algorithms In Use In Material Science Applications

Name and Affiliation Ian Bush, NAG

Subject

Overview of some of the main packages used in materials science computation when periodic boundary conditions apply:

- DL-POLY3 – classical MD
- CRYSTAL – Electronic structure
- Castep – Electronic structure

Current Packages, Libraries, Tools etc

- DL-POLY3 – classical MD
  - General parallelisation
  - No external libraries
  - Direct evaluation of short-range interactions, FFT approximation to long range interactions
- CRYSTAL – Electronic structure
  - Gaussian basis set
  - ScaLAPACK
  - Fortran 90
- Castep – Electronic structure
  - Plane wave expansion
  - Potential hierarchical parallelism

Current Algorithmic Requirements

- Domain decomposition (DL-POLY 3)
- General parallelisation
- Dense and sparse (Davidson method) matrix diagonalisation
- BLAS, LAPACK (Castep)
- FFTs (FFTW + vendors' libraries)

Future Developments, Requirements, Challenges and Issues

- Better alternatives or development of FFTs
- Better, more scalable, optimisation methods than BFGS for optimising the structure of materials
- Hierarchical parallelism (all three codes could benefit)
- More scalable methods for the Hermitian matrix diagonalization problem

Session 6 – Oceanography and Climate Modelling -

Title ICOM (Imperial College Ocean model): 3D adaptive unstructured mesh ocean modelling

Name and Affiliation Gerard Gorman, Imperial College

Subject

3D modelling of the ocean circulation, waves etc a the global and local scales

- Multiscale problem
- Need to represent highly anisotropic and complex domains

It aims to develop an open source framework for multiscale ocean modelling

Current Packages, Libraries, Tools etc

- FLUIDITY
  - Open source (LGPL) FE solver for CFD
  - Robust parallel implementation
- HYPRE – multigrid
- PROMETHEUS – multigrid
- PETSc
- Terreno – meshing package for multiscale gridding (avoids grid nesting)

Current Algorithmic Requirements

- FE discretisation
- Adaptive unstructured anisotropic mesh refinement and movement → dynamic load balancing
- Adjoint model for data assimilation and sensitivity studies
- Theta time-stepping
- Linearization by Picard iteration
- Iterative solvers (CG, GMRES) with standard preconditioners
- MultiGrid

Future Developments, Requirements, Challenges and Issues

- Working with adaptive fully unstructured meshes in the vertical
- New numerical techniques:
  - Adjoint
  - Mesh movement
  - Multi-physics
  - Fast solvers
  - AMG (Algebraic MultiGrid)
- Validation and comparison with data and other models (e.g. DYNAMO, MITgcm, etc)
- Adjoint data assimilation and sensitivity analysis
- CGNS for mesh input/output (though better standards would be useful)

Session 7 – Computation for Physical Science III

Title Computational Plasma Physics

Name and Affiliation Tony Arber, Warwick University

Subject

Overview of codes/models required for Tokamak (ITER) and fast igniter fusion modelling (HiPER):

- Gyrokinetic models (GKW) (continuum based kinetic) codes
- Particle-in-Cell codes
- Fluid dynamics codes

Current Packages, Libraries, Tools etc

- GKW – explicit gyrokinetic code
  - FD based

  - Massively parallel (MPP)
- GS2 - implicit gyrokinetic code
- PIC codes (EPOCH, Osiris)
- Vlasov codes (e.g. Kalos)
- No libraries are used

Current Algorithmic Requirements

- Riemann solvers
- FE method
- FD methods
- Lagrangian codes (ALE or remap)
- PIC algorithms
- Domain decomposition

Future Developments, Requirements, Challenges and Issues

- ALE (Arbitrary Lagrangian Eulerian)
- Adaptive mesh refinement
  - Load balancing difficult
- Implicit solvers for stiff parabolic terms
- Robust and scalable matrix inversions for parabolic, linear, implicit schemes
- Modelling QED (Quantum ElectroDynamics) processes
- For PIC: implicit EM field updates
- Improved FFTs on domain decomposed grids
- Replace FFT/spectral methods with FE (full tokamak simulation)
- Stay with Fortran 90
- Mostly known algorithms:
  - Issues are implementation, verification and validation not NA

Title Numerical Algorithms in Computational Chemistry

Name and Affiliation Paul Sherwood, STFC Daresbury

Subject

Overview of current state of computational Quantum Chemistry and bottlenecks

- Classical methods (empirical potentials)
- MD (Molecular Dynamics)
- Ab-initio quantum computations, finite basis set, Density Functional Theory (DFT)

Current Packages, Libraries, Tools etc

- NAMD – MD package relying on CHARM++
- DL_POLY 3 – MD package
- CHARM++ - communication/relocation layer for NAMD
- ScaLAPACK – including some pre-release code for MRRR
- PRMAT – atomic and molecular physics (scattering) code (finite basis)
- CRYSTAL – periodic ab-initio code (Gaussian basis)
- GAMESS-UK – molecular code (Gaussian basis)
- Molpro – molecular code (Gaussian basis)
- Peigs – bisection and inverse iteration
- PLAPACK – QR and MRRR
- Libsci, ACML – CRAY
- ESSL, PESSL – IBM
- SIESTA – molecular code (finite basis)
  - Minimisation rather than diagonalisation
  - Multigrid solver instead of FFT

Current Algorithmic Requirements

- Diagonalisation, partial and complete
    - tridiagonalisation
    - QR algorithm
    - Divide and Conquer
    - Multiple Relatively Robust Representations (MRRR)
    - Bisection and inverse iteration
    - Jacobi
    - Davidson-Liu
    - Symmetric subspace decomposition
- 3D FFT

Future Developments, Requirements, Challenges and Issues

- Block DC methods
- One-sided communication (seen as very important)
    - Global Arrays used in a number of Chemistry codes but issues with portability
    - MPI-2 offers only poor implementations on many platforms
- Poisson equation to replace multi-centre Coulomb integrals (Manby, Bristol)
    - Highly parallel Molpro implementation demonstrated on Clearspeed
- MADNES Project (Oak Ridge) recasts Schroedinger Equation as an integral equation over a grid
    - Multi-resolution analysis
    - Prototype code using wavelet basis
- MD codes require efficient 3D_FFT
- Libraries abstracting multi-core architectures

Title A tale of two applications

Name and Affiliation Cliff Addison, Liverpool University

Subject

Benchmarks for the new AMD Barcelona chip using the Liverpool University new cluster

- xhpl (aka parallel LINPACK benchmark) – dense LU factorisation and solution of equation
- VASP - ab-initio MD (Molecular Dynamics) code

Current Packages, Libraries, Tools etc

- xhpl
    - MPI-parallel
- VASP
    - MPI-parallel
- BLAS

Current Algorithmic Requirements

- FFT
- Diagonalisation

Future Developments, Requirements, Challenges and Issues

- Blocked storage schemes for dense matrices
- Tiling (Jack Dongarra’s recent work)
- Recursion (LU, Cholesky, QR)

Proforma for discussion groups.

HPC/NA Workshop 1: Applications: underlying algorithms and challenges for the future 5th-7th November,
Oxford e-Research Centre

Session/Application Area:
Breakout Group:
Current numerical and computational performance required
Forecasted numerical and computational performance required to tackle future problems of interest
Algorithms needed, if known, and their characteristics
Areas of overlap with other applications
Numerical capabilities required, otherwise, in order to map these to existing algorithms or help the design of new ones
Current and required algorithmic deployment vehicles (i.e. packages, libraries, etc)
Mapping to advanced HPC platforms
Knowledge of existing activities in this area

Annex 4: Information on Existing Software

Packages Used or Mentioned		Used?
ADF	DFT (Density Functional Theory) for molecular electronic structure	Y
ALBERTA	Adaptive Hierarchical Finite Elements Toolbox	Y
Castep	Molecular Electronic structure, plane wave basis set	Y
CENTORI	CFD for Plasma physics	Y
CRYSTAL	Molecular Electronic structure, Gaussian basis set	Y
DESMOND	Molecular Dynamics (MD) - ISV proprietary	
DL-POLY3	Molecular Dynamics (MD)	Y
EPOCH	Particle-In-Cell (PIC) code	
FLASH	Eulerian hydrodynamics for astrophysics (galaxy formation)	Y
FLUENT	CFD – uses FV discretisation	
FLUIDITY	CFD - general purpose multi-phase CFD code (oceanography)	Y
GADGET-2	Particle dynamics model for astrophysics (galaxy formation)	Y
GAMESS-UK	Molecular Electronic structure, plane wave basis set	
Gaussian	Molecular Electronic structure, Gaussian basis set	Y
GKW	Gyro-kinetic code for plasma physics	Y
GROMACS	Molecular Dynamics (MD)	Y
GS2	Gyro-phase fluid model package for plasma physics	Y
HELIUM	Time-Dependent Schroedinger Equation for two-electron systems in laser	Y
HYDRA	CFD used at Rolls-Royce	Y
Kalos	Vlasov code	
MADNES	Oak Ridge project: molecular SE recast as integral equation (under development)	
MCNP	Monte Carlo neutron transport (reactor safety)	
METIS	Graph partitioning (reordering for sparse matrices)	
Molpro	Molecular Code (Gaussian basis)	
NAMD	Molecular Dynamics (MD)	Y
Netgen	Mesh generator for small-ish problems	Y
ORB5	Particle dynamics modelling for plasma physics	Y
Osiris	Particle-In-Cell (PIC) code	
PADRAM	Mesh generator	Y
PARAMESH	Parallel mesh generator	Y
ParMETIS	Parallel version of METIS	
PRMAT	Atomic electronic structure code (finite basis set)	Y
SIESTA	Molecular Electronic structure, finite basis set	
Terreno	Meshing for multi-scale avoiding nested grids	Y

TETRAD	Mesh refinement	Y
VASP	Ab-initio Molecular Dynamics (MD)	Y

Parallel Tools

Data Synapse	Low latency distributed task submission system	
Symphony	(from Platform Computing) - low latency distributed task submission system	

Libraries and Supporting Packages

ACML	AMD maths library
ARPACK	Arnoldi eigensolvers for non-symmetric (non-Hermitian) sparse matrices
BLAS	Only BLAS from vendors (Intel MKL, AMD ACML) mentioned
CGNS	Mesh information input/output
CHARM++	Communication/relocation layer for NAMD
ESSL	IBM serial maths library (similar to MKL, ACML)
FFTW	FFT
gViz	For interfaces to visualisation framework
HDF5	Parallel I/O
HSL	Harwell Sparse Libraries (Linear algebra library for sparse matrices)
HYPRE	Multigrid
LAPACK	Linear algebra for dense and band matrices (generally from vendors - see BLAS)
Libsci	CRAY scientific library
MKL	Intel maths library
MUMPS	Direct linear solver for sparse matrices
NAG	NAG numerical libraries
OPLUS	Communication layer for HYDRA
PARPACK	Parallel version of ARPACK
Peigs	bisection and inverse iteration for symmetric (Hermitian) eigenproblems
PESSL	IBM MPI-parallel maths library
PETSc	Iterative linear solvers for sparse matrices
Prometheus	Multigrid
ScaLAPACK	MPI-parallel version of LAPACK
SPARSKIT	Serial (non-parallel) sparse matrix solvers

Annex 5: Overview of Workshop 2

This workshop focused on the numerical and algorithmic details that had been identified in the application workshop 1.

There were 20 attendees including two international participants at the workshop, held at the University of Manchester – a full list of attendees is provided below. The majority were academic numerical analysts.

The agenda for the meeting is provided below. The two international participants gave overviews of activities in France and at the Department of Energy in the US [their presentations are available to download from the project website). The rest of the programme focused on discussion around specific numerical areas to begin to understand what exists in each, what the barriers are and what planned activities there are.

International Keynote Presentations

Osni Marques – LBNL – now at IPA assignment at the DOE HQ, OASCR

The full presentation given by Osni may be downloaded from the project website. The slides cover a number of projects in the DOE.

Osni provided an overview of the DOE efforts in computational science applications and software. He gave an overview of the tools and libraries that are developed and supported by DOE under the Advanced Computational Software Collection: these can be found at acts.nersc.gov . This effort started in the late 90s and continues to be supported into the future. The idea is to make the software tools available on the various computing facilities sponsored by the DOE. Education and training also feature as an important feature.

Each year DOE hold a workshop to bring together stakeholders and pay for students to attend. Osni also noted the DOE Computational Science Graduates programme that provides support for students who are able to spend time in the DOE laboratories.

The other relevant DOE program is SciDac, details of which can be found at www.scidac.gov. This supports breakthrough science enabled through HPC by partnerships among discipline experts, applied mathematicians and computer scientists.

The DOE have a number of workshops relevant to this activity planned in the next 12 months. These are under the umbrella of “High risk, high payoff technologies for applications” and are called Extreme Scale Computing Workshops in areas such as climate, high-energy physics, nuclear physics, nuclear energy, fusion, biology, and material science. Details can also be found on the website extremecomputing.labworks.org and may provide valuable inputs to this roadmap. There are also a number of DOE reports that provide insights for this.

Michel Kern, INRIA and Ministry for HE and Research

Michel reported on several activities that are related to the roadmapping activity. Michel’s full presentation is provided may be downloaded from the project website.

Michel began with an overview of the hardware purchases in France for the provision of national HPC. France (together with the UK, Germany, Spain and the Netherlands) is one of the principal partners of the Prace project (http://www.prace-project.eu/) that prepares the creation of a pan-European HPC service.

Michel explained that Genci (Grand Equipement National de Calcul Intensif) provides coordination of the national centres. Genci is owned for 50 % by the French State, represented by the Ministry for

Higher Education and Research, for 20 % by the CEA, 20 % by the CNRS and 10 % by the Universities. Genci also actively promotes the use of HPC in fundamental and industrial research.

He noted that while the CSCI provides the high level strategy for HPC in France, the main funding mechanism for (software related) HPC projects is the Cosinus programme from the Agence Nationale de la Recherche (http://www.agence-nationale-recherche.fr/Intl). These are usually 3 yr projects and generally 10-15 projects per year at a value of around 10-15 M€. The details of existing resources are given on the attached slides.

One of the items he discussed that seemed very pertinent to the roadmapping activity was the Seminar – Thinking for the Petaflop (CEA-CNRS). This project has had four working groups: Sharing, Algorithms, Organization, Teaching. These map very well onto the areas we have identified as keys for the future. The project will have a report out soon.

Michel provided an overview of several algorithmic and application software projects in France that are related to the topic of the workshop. Details can be found in the attached presentation.

Numerical Algorithmic Areas

Load balancing (meshes, particle dynamics) – Lead Peter Coveney

The discussion was concerned with load balancing for codes involving meshes, particle codes, and how one deals with complex interactions. Peter uses two classes of codes which are coupled with other codes. In CFD, the Lattice Boltzman algorithm is very attractive as it scales very well as we go to higher core count machines as the communications are very local. It is a nice way of modeling complex fluids and turbulence. The other class of algorithms is Molecular Dynamics (MD) with long range interactions, used for materials and biomolecular applications.

Load balancing has been a concern for many years, when trying to deploy Lattice Bolztman codes onto machines like CSAR, T3E etc. In those cases the model you are trying to describe is heterogeneous with lots of things happening in different parts of the simulations. In some applications you have interfaces in fluids across different regions or domains and you need to do a lot more computation at their interfaces. As it turned out, this class of application has never really been problematic. For example one of the codes, LB3D, has been happily studying highly heterogeneous systems and there are classes of liquid crystalline materials under flow and shear that scale well up to core counts of order 65k. Our applications are run on state of the art machines in the US such as Ranger (TACC, Austin, 62K cores) and Intrepid (The IBM Blue Gene/P at the Argonne Leadership Computing Facility (ALCF), 164K cores). The Lattice Boltzmann codes scale to those sort of numbers without too much problem probably due to the heterogeneity of the machines.

A distributed form of application can be run using MPIG (Grid extension of MPI) running over multiple machines. MPIG does a good job of overlapping communication and computation. They have examples of applications running better on two or more machines than on a single system, even if those machines are thousands of miles apart. MPIG could also be useful for running on a single system when there are heterogeneities to load balance.

In the second class, Molecular Dynamics for long range interactions, one code used is NAMDE which uses Charm++ to help with load balancing. This, in Peter's opinion, is a very messy environment for developers. Questions include how important load balancing is and how specific it is to your problem, and if there is something you can do that is generic and reusable for other people's purposes. As we go to very large core count machines, we are interested in looking at single "capability" applications that would require over half of the machine to run. Would it ever be sensible to run applications across such large number of cores? On Ranger or Intrepid, we can now run huge numbers of jobs that used to be 'large' (i.e. in the "capability" class on HPCx) and therefore do things in MD that are quite challenging such as properly sampling the system: if you can run a huge number of replicas of a system for a short period of time, the properties you get are far superior to those obtain through a single simulation.

Other people's experiences of load balancing in HPC are different. Their experience is typically problem dependent – constraints appear to be different but this might be an underlying optimization problem. Optimization libraries could be applied to load balancing problem if we were able to express the constraints appropriately and feed them into an appropriate optimization tool.

Ocean modeling and environment codes have load balancing problems of the type where each processor has to calculate according to number of sea points it has and night/day will change the amount of radiation computation. The load balancing problems might be too specific and there might a generic approach to load balancing may not be suitable in this case.

Can these problems be offered to the Optimization community?

- Never going to get 100% perfection so solving optimization problems requires assessing the payoff
    - If it takes too long to solve to optimsation problem, then the gains are small
- This might be a max flow problem: in this case, there are good polynomial algorithms for solving this efficiently
    - There are very fast graph algorithms that can deal with this sort of problem, it's not NP hard.
- If you have access to a Grid of resources, performance and checkpointing are key, to relaunch onto more cores. The optimization issues are to do with the dynamics of the usage of the machines. MPIG is one way of handling this.

Optimization – Lead by Nick Gould and Jacek Gondzio

For a number of reasons the first workshop did not discuss the importance of optimization. This was due in part to the range of applications included but also to the focus of the individuals involved. The DOE report *Scientific Application Requirements for Leadership Computing at the Exascale*[1] includes optimization as one of the key areas for the future:

"The new algorithm categories that application scientists expect to be increasingly important in the next decade include adaptive mesh refinement, implicit nonlinear systems, data assimilation, agent-based methods, parameter continuation, and optimization."
APPLICATIONS

- ***Simulation-based optimization***
    - ***PDE constraints***
    - ***US DOE highlight for communication systems (civil and military) (1)***
- ***Scheduling (electricity, gas, water)***
    - ***network constraints***
    - ***variables***

[1] Scientific Application Requirements for Leadership Computing at the Exascale (ORNL/TM-2007/238)

http://www.nccs.gov/wp-content/media/nccs_reports/Exascale_Reqms.pdf

    - ***global optimization***
- Engineering
    - Structural design -> ideally zero-one variables
    - Truss-topology design -> semi-definite programming
    - Chemical process engineering - network nonlinear programming
    - VLSI design -> semi-definite and nonlinear programming
    - Contact problems -> complementarity
    - Elasto-hydrodynamic lubrification -> nonlinear complementarity
- Traffic equilibrium
    - Wardrop principle -> nonlinear complementarity
- Physics
    - minimize potential energy
    - protein folding (e.g. Lennard-Jones models)
    - global optimization?
- Finance
    - Portfolio selection -> quadratic programming
    - Risk management -> linear programming
    - Asset management -> stochastic (linear) programming
    - European option/GARCH/Black-Scholes models - > nonlinear (liklihood) fitting
    - American options -> dynamic programming
    - Arbitrage models -> semi-definite programming
    - Multi-period portfolios -> robust optimization
    - (Nash) games -> complementarity
- Energy: quotes from report[2]
    - "First-principles computational design and optimization of catalysts will become possible at the exascale, as will novel design of biologically mediated pathways for energy conversion."
    - "Nuclear fission reactor design and optimization would help accelerate understanding of key plasma physics phenomena in fusion science"
    - "Exascale systems should also enable a major paradigm shift in the use of large-scale optimization techniques to search for near-optimal solutions to engineering problems. Many energy and industrial problems are amenable to such an approach, in which many petascale instances of the problem are run simultaneously under the control of a global optimization procedure that can focus the search on parameters that produce an optimal outcome."
    - "Of great interest are methods that will enable the power of exascale computing to advance the use of mathematical optimization in many areas of science and engineering. Examples include the use of ensembles and outer loop optimization to iterate design parameters of new nuclear reactor designs that would simultaneously improve safety margins and lower cost, or to explore the parameter space of technology choices and how they might impact global energy security strategies."
    - "Robust and reliable optimization techniques that exploit evolving architectures and are easy to use"
    - "Appropriate algorithms for novel optimization paradigms that can be implemented only at the exascale (e.g., hierarchical optimization problems over multiple time stages) Handling of problems with hundreds of thousands of discrete parameters."

[2] Modeling and Simulation at the Exascale for Energy and the Environment (Town Hall meeting, 2007)

http://www.er.doe.gov/ascr/ProgramDocuments/Docs/TownHall.pdf

- Accelerator physics[3]
    - design and optimization for better efficiency at lower costs
- Meteorology
    - 4D variational data assimilation
    - O(109) unknowns

CURRENT OPTIMIZATION

Search (for local optima) is essentially sequential.

- Parallelism is via
    - function and derivative evaluation
    - linear system solution
- optimization often involves inequalities => needs its own (convex) analysis
- Most real optimization problems are at least NP hard!
    - non-convex optimization
    - integer programming
    - global optimization
- Optimization currently often uses implicit elimination of constraints
    - adjoints
    - inefficient optimization

NB. Often only require inaccurate solution until convergence (c.f. inner-outer iteration). Often better to use all-at-once approaches.

OPTIMIZATION USES

- linear systems (sometimes)
- generically symmetric, usually indefinite, frequently very ill conditioned
- eigensolvers
- other solvers for constraints (ODE/PDE/quadrature)

PARALLELISM

- branch and bound for integer and global problems

BIG CHALLENGES

- Better polynomial methods for linear/convex quadratic programming
- Polynomial approximations to NP hard problems
- Scaling (or scale-invariant methods!)
- Derivatives (automatic differentiation)
- Good branching strategies
- Good bounding strategies
- Warm-starting
- Semi-definite optimization (state of the art is small, systems are inevitably dense)

TEACHING OF OPTIMIZATION

- Needs more emphasis in the undergraduate curriculum
- c.f. Europe and North America

Eigenvalues: dense and sparse - Lead by Nick Higham

Nick identified the standard eigenvalue transformations within an application, namely:

[3] Science Prospects and Benefits for Exascale Computing (ORNL/TM-2007/232)

We begin with a rational form $R(\lambda)x = 0$

Which generally becomes a polynomial form $P(\lambda)x = 0$

Which is linearized to become $Ax = \lambda Bx$

And finally $Ax = \lambda x \quad (A := B^{-1}A)$

It is usually the latter form that application developers bring to numerical analysts to solve. Nick pointed out that we need to consider solutions over the range of forms as at each stage information is lost and often cases arise in the earlier forms.

The following table provides an overview of software available for the dense and sparse cases.

	Dense	Sparse
$Ax = \lambda x$	LAPACK	ARPACK
$Ax = \lambda Bx$		EA19 (symm) Jacobi-Davidson
$P(\lambda)x = 0$	Polyeig	
$\mathrm{R}(\lambda)x = 0$		

It was noted that we need more detailed information regarding the sorts of applications that require eigenvalue decomposition. For example, for what problems are half or more of the eigenvectors required.

Methods for $P(\lambda)x = 0$ are under active development and a LAPACK code for the dense case is foreseeable in the next couple of years. Very often in practice $P(\lambda)$ has structure, such as symmetry, hyperbolicity, palindromicity or gyroscopic structure and algorithms that exploit these structures are required.

It was suggested that it would be a good idea to bring application scientists and numerical analysts together to consider the problems differently and to formulate higher level solutions. Various policy vehicles exist to enable this including from simply a workshop, to an EPSRC Ideas Factory.

This question was raised: how many problems are related to pseudo-spectra? It was noted that no one had heard an application scientist mention pseudo-spectra and it hadn't really taken off as a standard tool at this stage.

PDEs, domain decomposition, adaptive meshing – Lead by David Silvester

The last workshop identified adaptive mesh refinement as a key area with six of the presentations indicating that this was an area of concern for them. It was noted that the UK has some disparate groups working in this area including groups at Bath, Imperial, Leeds and Manchester and some in Oxford.

David identified the PDE's that are being solved using HPC:

- Navier Stokes
- RANS
- Schrödinger Equations
- Porous differential equations
- No mention of elasticity
- Maxwells equations
- Einstein Equations
- Stochastic PDEs

Codes are running faster with bigger machines however the algorithmic approach will need to change with parallelism and multi-core. Many people are still using packages developed in the 1970s. The new architectures won't be suited to these old codes. The codes are increasing in complexity as more physics is being added to the model, higher resolutions are being used and there are more coupled models.

FFT and related transforms - Lead by Jan Van Lent

(with input from Ivan Graham & Rob Scheichl)

FFTW[4] is the standard high performance software for FFT which is widely used. From wikipedia: FFTW, for "Fastest Fourier Transform in the West," is a software library for computing discrete Fourier transforms (DFTs) developed by Matteo Frigo and Steven G. Johnson at the Massachusetts Institute of Technology. It is public domain under GNU; there is also a commercial version from MIT and it underlies the fft and ifft commands in MATLAB. FFTW handles data of any length N, but works best when N has small prime factors: powers of two are optimal size; a (large) prime sizes provide the worst cases.

FFTW is widely used by the scientific community in particular computational Physicists and Chemists.

FFTW runs under MPI so supports parallelism but its parallel performance is controversial. Several participants at the Oxford Roadmapping meeting reported dissatisfaction. Recent versions of FFTW are optimised for special architectures like multicore, cell, GPU or FPGA's.

There is a page on parallel FFT at http://www.sandia.gov/~sjplimp/docs/fft/README.html

FFTW assumes equally spaced data but there are recent versions of FFT for non-equally spaced data.

The NFFT (a form of the FFT that allows non-equally spaced sample) has been developed by Daniel Potts (now at Chemnitz) and co-workers: www-user.tu-chemnitz.de/~potts/nfft. A number of FFT spin-offs are offered at their web site: for example, a fast Gauss transform, a fast summation of radial functions on the sphere, polar FFT, etc. Key names are Keiner, Kunis and Potts.

In the USA, a more popular (and older) version of the FFT that allows non-equally spaced data is called USFFT (Unequally spaced FFT), see www.fmah.com/IMAGES/SEISMIC/MANUAL.PS

OTHER TRANSFORMS

A, by now, fairly old fast Legendre transform software written by Mohlenkamp is available. This involves computational costs proportional to C N log N, unfortunately with a large constant C, so that it can only be used efficiently for large values of N. As far as we know, this has not been widely successful.

A fast Legendre transform together with an FFT could be used for fast computation of spherical harmonic expansions for functions on spherical domains such as occur in many problems in geophysics. Important names in this area are Driscoll and Healy and more recently Kunis and Keiner, Suda and Takami. The big group of Freeden in Kaiserslautern makes heavy use of such technology in geophysical applications. There are at least two suites of software to do ``spherical FFT'': www.cs.dartmouth.edu/~geelong/sphere/ and www-user.tu-chemnitz.de/~potts/

It was agreed that there is international demand for such transforms e.g. in fields like Numerical Weather Forecasting. One example is Theora available through the Xiph.org Foundation: www.theora.org/faq/#32

[4] www.fftw.org

The JPEG2000 standard uses wavelet compression: en.wikipedia.org/wiki/JPEG_2000

Many examples of wavelet compression methods for pictures and images are listed at en.wikipedia.org/wiki/Wavelet_compression

Many technology companies, cameras etc, are working on video compression and the like (including the BBC).

Iterative solvers, including Krylov methods, multigrid – Lead by Peter Jimack

Peter noted that while the implementation of most iterative in methods in parallel is not particularly demanding, the development of effective, parallel preconditioners is very challenging. Peter focused on sparse problems although he noted that sparse preconditioners may also be employed with boundary element methods.

Peter reported how, for most cases, preconditioners based on the approximate inverse could be used, and gave an overview of methods for calculating approximate inverse preconditioners, highlighting their strengths and weaknesses with regards both to the problems domains and to the type of hardware architecture. It was noted that often the structure of the matrix and physical properties of the problem can provide clues.

A number of packages were discussed including HYPRE from Livermore and AZTEC from Sandia National Labs. It was clear that there is much more activity in the US than the UK in this area.

Nick Gould stressed that reliable, stopping criteria for convergence need be developed. He also asked at what level parallelism would be most effective for non-linear equations: in the inner iterations (i.e. the underlying linear iterative solvers) or in the outer iterations (i.e. exploring in parallel the non-linear equation solution space).

Peter Jimack noted that there are applications that need to provide reproducible results: this imposes considerable constraints on any algorithms using an asynchronous or random communication pattern.

COMMENTS ON THE “ITERATIVE SOLVERS” DISCUSSION

- Many systems are structured, either globally or through structured sub-matrices -> structure-exploiting not generic methods needed
- A key issue is when to stop (often determined by physics, etc)
- All issues discussed also relevant for nonlinear equations, but then a delicate balance between how accurately to solve inner (linear) system vs. pay-off for overall progress
    - trade time for function/derivative evaluations against time for linear solvers

Direct Methods – lead by Jennifer Scott

Bascially, there are two separate cases: dense and sparse. The dense case is the remit of such projects as BLAS, LAPACK and ScaLAPACK. The UK has had involvement in this over the years (notably, people at NAG, RAL and Manchester). Much of the current effort is being done in the US, in particular, Jack Dongarra and his team is leading the way with the PLASMA project.

Dense	**Sparse**
BLAS	MUMPS -MPI
LAPACK	Pardiso – Basel, Intel, OpenMP
SCALAPACK	WSMP - IBM
	SuperLU
	PASTIX

Most sparse direct solvers take advantage of software for dense solvers, as they rely for efficiency on dense. However, improved dense kernels only lead to modest improvements in sparse solvers: it is necessary to design new sparse algorithms that can exploit more general parallelism. This is a tough but very important problem because the solution of sparse linear systems often lies at the heart of computational science, engineering and finance problems.

As models become more sophisticated, ever larger systems need to be solved accurately and efficiently. A small number of parallel sparse solvers are available. Main codes are:

- MUMPS: developed in France. This is an MPI-based code.
- PARDISO: originally developed at the University of Basel. It has now been taken over by Intel and is distributed with the latest version of the Intel Library. It uses OpenMP.
- SuperLU: it comes from Berkeley and is widely used in the USA. It was designed for unsymmetric systems only. There are versions for both shared and distributed memory.
- WSMP: IBM code (commercially available).

All of these codes have drawbacks and none will solve all the problems users are currently interested in.

Memory is a key issue for direct solvers. Recently, there has been considerable interest in working out of core, but this is another challenge in the parallel case.

Jennifer noted that we should stop looking at direct methods as black boxes but look more carefully at the structure of a problem. We should probably also research into hybrid methods, crossing over the separation between direct and iterative methods as, for example, iterative refinement techniques become more important.

COMMENTS ON THE “DIRECT SOLVERS” DISCUSSION

- iterative refinement should be employed - blurs distinction between iterative and direct methods
- sometimes require many solves per factorization - pays off to compute "sparse-est" possible factors
- third class of problems which are dense but structured - loose information if simply use LAPACK
- under/over-determined systems very important - least squares and regularisation methods needed (QR vs. LU)

Software: engineering, language, libraries, validation, standards – Lead by John Brooke

John began by saying how a well documented body of research showed that usability (how software is viewed and approached by users) was an often contentious issues issue, arising, for example, in the choice of programming languages, development methodologies, look-feel of a package, etc. This can have an even greater impact that algorithmic choice. These questions were seen as paramount:

- Who may use the software, the actual products of algorithms?
- What are the groups of users of these algorithms and what is their degree of sophistication. in the context of HPC?
- How do they want algorithmic content to be delivered: libraries, own implementation of algorithms, components of general packages like Matlab?
- Why do they need algorithms and software? The needs of scientific applications, aimed at research and discovery, or engineering where validation and verification actual physical model are essential, may well differ.
- What machines do they want to use? Hardware architectures affect algorithms, of course.

If we can answer these questions to some extent then they will help with thinking about the 'engineering' aspects of the numerical algorithms. In some cases the results of the algorithms need to be reproducible and repeatable, and in others the ability to do discovery are more important. In general the whole body of practice in software engineering maybe hasn't been taken up sufficiently in the science domain.

In terms of reliability we can't ignore underlying architectural issues. Very large systems by their very nature are likely to have physical errors of nodes or memory – as their size increases so does the probability of hardware/software failure. Parity errors are a particular problem. Systems are designed to continue even if a node fails.

The final point is about distribution. How are the software products going to be used? Will they go into libraries, or to commercial partners for hardening, or is the aim to help the actual users of the algorithms to incorporate them into their own user-generated codes? There will be different answers for different users, which has an impact on how the software is engineered.

Discussion

- It is important to bear in mind that the best delivery system may not be a library – could still be appropriate for a numerical service – particularly, for example, on a GPU. Integrating physics into the model might be easier if not calling library codes
- "Frameworks" may become essential for delivery.
- Some codes are "horrendous" because of complexity and because of poor documentation

Matthias Heil noted that we need to make sure that the framework doesn't get so big that we lose the capability to deliver it.

There was a lot of discussion of frameworks and object oriented approaches that have been attempted in the past such as CCA and PETSc. It is clear that some have been more successful than others and we need to learn from those experiences.

Annex 6 :Overview of Workshop 3

This workshop focused on discussing and reviewing the first instantiation of the National Roadmap for HPC Algorithms and Applications, published just before the workshop in January 2009.

Additional aims of the workshop were to:

- Gain national and international perspectives from a range of speakers
- Review the roadmap document
- Present a community view to the Research Councils via EPSRC
- Think about the next steps and continuing the dialogue

There were 34 attendees including four international participants at the workshop, held at the Royal Society, London over 26th-27th January 2009. There was a good mix of application developers, computer scientists, numerical analysts and vendors including system architects.

Overview of Presentations

Copies of the presentations may be downloaded from the HPC-NA project website.

Prof Stan Scott (Queen’s University, Belfast)
Emerging HPC Technologies: back to the future?

Prof Scott gave a comprehensive overview of the current hardware developments having an impact on the immediate future of HPC: GPUs, FPGAs, heterogeneous chips (e.g. Intel Larrabee), floating point accelerators (e.g. Clearspeed), and heterogeneous systems (e.g. IBM Roadrunner). He commented that in the future hybrid computing, i.e. collections of heterogeneous components, would become more common. However, he sounded a warning as novel technologies often appeared unstable and might well be discontinued. Stan highlighted a study by Sandia, showing the impact of the “memory wall”, i.e. the inability to ”feed” data fast enough to hungry processors due to bandwidth constraints, and how this resulted in decreasing parallel efficiency as the number of cores grew. He added that there was the possibility of a rebirth of SIMD algorithms, at least in some specific cases. Commenting on GPUs, he expressed some concerned about their potential non-compliance to IEEE 754 standards, and wondered whether that could have a detrimental effect on numerical stability. Stan highlighted some efforts to ease the situation: PLASMA, a multi-core aware follow-up to the LAPACK genealogy; mixed-precision numerical computation, to achieve best performance from mostly 32-bit engines such as GPUs; HONEI, a collection of libraries targeting multi-processing architectures; Sequoia, a memory layout aware programming language that could lead to self-adapting algorithms.

Finally, he described some recent work in the UK.

- A recently formed consortium, a joint effort by the Universities of Belfast, Imperial, Cardiff, Bristol and Southampton and Daresbury Laboratory, to study a collection of large codes from the CCP community. The aim being not only to provide high performance versions of these codes but also to abstract general principles and guidelines for the design of applications and algorithms for emerging and future HPC systems
- The recently formed Multi-core and Reconfigurable Supercomputing Network (MRSN), an initiative led by Oxford, Imperial, Belfast and Manchester.

He reported to the meeting that a conference dedicated to emerging HPC technology, MRSC 2009, would be taking place in Berlin on March 25-26 2009.

Dr Steven Kenny (Loughborough University)
Accelerating Simulations using Computational Steering

Dr Kenny reported on his group’s investigation on materials required for energy demand reduction, particularly on glass plates. Steven reported that the simulation of these glasses would require a three-fold iteration, each iteration been determined by a host of parameters that were very difficult to compute automatically. Convergence criteria and requirements were, in particular, difficult to assess and tended to vary considerably even across closely related problems. Hence, their solution consisted in allowing computational steering, in the sense of altering manually on-the-fly the parameters responsible for the computation at each iteration level. He commented that this allowed researchers to make best use of their own understanding of the problems studied and resulted in increased speed-up and better utilisation of resources.

He found that current set-ups, particularly at National centres, inhibited the possibility of running of a wide adoption of computational steering and that further developments would be required for complicated coupled simulations.

Prof Bruce Boghosian (Tufts University)

Spacetime computing: A Dynamical Systems Approach to Turbulence

Prof B.Boghosian reported on a class of algorithms being developed to tackle the very computationally intensive problem of turbulence at high Reynolds' numbers, such as in flow past aircraft. This spacetime computing, exemplified by so-called Parareal algorithms, employs coarse and fine time grids. The coarse grid, with purely sequential evolution, can be used as a predictor, and then a number of time slices (the fine grid) can be computed in parallel for each coarse time step, and acting as a corrector for the coarse grid results. In other words, these methods can be viewed as achieving domain decomposition in time. Bruce commented that these or similar methods could provide the only way of making turbulence simulation faster than real time.

Additionally, with periodic boundary conditions in time, it would be possible to use this method to generate the discrete set of unstable periodic orbits (UPOs) of a given flow. The enumeration of these, a project that requires petascale computing, would be of enormous help in extracting averages of observables over the turbulent flow. The so-called dynamical zeta function formalism reduces such averages to combinations of those over the UPOs.

Bruce added that other these ideas, particularly parareal algorithms, could affect other areas by providing means of parallelising evolution equations and efficiently extracting statistical results.

Dr Massimiliano Fatica (NVidia)

CUDA for High Performance Computing

Dr M.Fatica, from NVidia, first introduced the newest NVidia GPUs (Tesla T10), their architecture and their performance capabilities. Of particular relevance was the introduction of a number of dedicated Double Precision threads (cores) within the GPUs and their (the DP units, that is) compliance to IEEE 754 standards.

CUDA, based on standard C, is the language used for programming these GPUs. CUDA encapsulates a thread-based approach to parallelism and allows the mapping of threads to the GPU thread arrays. Massimiliano reported that through CUDA a number of applications had benefited by either being ported directly to GPUs or by employing GPUs as accelerators for specific computationally intensive portions, with considerale faster performances than achievable by conventional hardware.

Dr Ram Rajamony (IBM Research, Austin)

Productive Petascale systems and the challenges in getting to Exascale systems

Dr R.Rajamony reviewed the current contribution of IBM to advanced computing. In particular, he singled out several issues that affect scalable performance, but were not always evident from headlines performance figures such as overall network bandwidth and access to memory. Ram pointed out that the new Blue Gene/Q would be addressing these issues and would preturn sustainable performance figures beyond what could be achievable elsewhere.

Ram also reported on IBM efforts to create a new model of parallelism, based on direct access to remote data (PGAS), in a way akin to virtual shared memory model. This approach to parallelism would depend on the application code to guarantee memory integrity in the course of computation, using a number of provided locking primitives.

Dr Maziar Nekovee (BT Research & UCL)
High-Performance Computing for Wireless Telecom Research

Dr M.Nekovee gave an overview of computational modelling and optimisation for modern wireless telecom systems In particular, he concentrated on V2V (Vehicle-to-Vehicle) networks for future intelligent transports. In the future, he argued, such capabilities would be used for provision of broadband access to millions of vehicles, for traffic monitoring and optimisation, to convey and relay information to vehicles, to allow intelligent decision making, for example, for routing and reducing traffic congestion.

The simulation poses extreme difficulties as phenomena of widely differing time scales need be considered: vehicles on a slow time scale, of the order of seconds, the wireless network and its requirements on a time scale of microseconds. Maziar said that such coupled simulation played an increasing role in industry as well as defined the state-of-the-art and posed extreme challenges to HPC, possibly requiring altogether novel approaches and algorithms, e.g, for scalable parallel discrete event simulations.

Prof Philip Treleaven (UCL)
UK PhD Centre for Financial Computing

Prof P.Trevelean described the new Doctoral Training Centre for Financial Computing at UCL. The new Centre aroused considerable interest and was supported by a large number of major financial outfits. He commented that, despite contingent worldwide economic difficulties, HPC was viewed by all major banks and financial institutions as a key technology. The UCL-lead PhD programme had great appeal to them and aimed to facilitate the already strong links between UK banks and UK Universities.

Philip then gave an overview of the various aspects of Financial Computing: financial modelling, mathematical finance, financial engineering and computational finance. A wide range of algorithm was employed and more would need to be developed to cater for the needs of the financial world: e.g. automated learning, sophisticated statistical analysis, probabilistic methods, many flavours and techniques of optimisation, dynamic programming, Monte Carlo simulations, etc. On the computational end of the spectrum he saw the increasing importance of automatic and semiautomatic trading systems. He added that a number of key banking applications required HPC approaches, such as derivatives pricing, portfolio management and risk management. The importance of risk management was increasing also due to the expanding role of financial regulators.

Dr Charlie Laughton (Nottingham University)
Biomolecular Simulation: Where do we go from here?

Dr C.Laughton gave a comprehensive overview of the challenges facing MD (Molecular Dynamics) approaches to biomolecular simulations. He said that the focus of interest is shifting towards the study of complex systems over the millisecond-to-second timescale. This was unachievable by current technologies and algorithms as it was many orders of magnitude beyond their capabilities. The evaluation of the forces between interacting particles represented a serious bottleneck. In particular, various approximations used to compute long-range forces had not proved to scale sufficiently well to very large numbers of cores. A second bottleneck is the short simulation time step needed to keep the algorithms stable.

Charlie concluded that radically new approaches were necessary. Coarse graining, i.e. aggregating a number of particles into larger objects (e.g. several atoms in one molecular group) could speed-up things considerably, but still not adequately for the long-term requirements of the field. Larger still objects, such individual biomolecules, should be representable. However, Charlie said, these objects would have internal structure and flexibility and developing efficient methodologies to represent them, and the interactions between such complex objects, would pose a formidable challenge. Many of the properties of these larger objects with internal structure could be inferred from studies at a smaller scale of them and their components. This approach would then make best use of a body of knowledge already accumulated. Charlie surmised that should this approach prove feasible, then grand challenges of computational biology, such as the simulation of a whole bacterial cell, could be realistically tackled.

Dr Sabine Roller (HLRS, Stuttgart)
Challenges and opportunities in hybrid systems

Dr S.Roller described how HPC was currently organised in Germany and the role of HLRS (Stuttgart). She said that funding was divided into three, roughly identical, portions: the first for the three National centres (Stuttgart, Munich and Juelich); the second for the ten Regional Centres with specific domain focus (Aachen, Berlin, Hannover, etc); the third portion for University-based HPC-servers. This pyramidal structure, Sabine commented, served a number of purposes, from allowing applications to "scale up" to the large National platforms, to allow the "trickling down" of know-how and algorithms from the high-end to smaller systems. This last point, she added, was seen as having great importance, particularly in view of the strong ties between the research and industrial communities.

Sabine then reviewed the work carried out at HLRS to employ different architectures for different portions of a specific application. She explained that that was the meaning of "hybrid computing": creating a computing environment made up of different technologies and optimise applications on this. In the HLRS case, traditional cache-based as well as vector processors were made available to an aero-acoustic application and the grid and numerical mathods were mapped to the two architectures employed. She then proceeded to show that much higher performance could be achieved by a hybrid system than by a purely cache-based or vector-based system.

Dr Kevin Stratford (University of Edinburgh, EPCC)
HPC for Soft Matter Physics: Present and future

Dr K.Stratford first explained that by "soft matter" he meant the study and simulation of liquid, gels, foams etc., such as the study of liquid crystals "Blue Phases", binary fluid under strain, suspensions (ferrofluids).

Kevin showed that the study of blue phases of liquid crystals is acquiring great technological importance, for example for next generation of fast-switching, low-power displays. He reported that the phase transition could be simulated by solving the Navier-Stokes equations via a lattice Boltzmann method. A siimilar computational approach could be used to simulate binary fluids under strains as well we colloidal suspensions of particles subject to long-range forces (e.g. magnetic particles).

Kevin reported that their main code for lattice Boltzmann computations employed PI parallelism and had been ported successfully to a number of HPC platforms. The code was not publically available, and was unlikely to become so in the immediate future for a number of reasons. Work was underway to include better kernels (BLAS, PLASMA, etc), algorithmic enhancements and, possibly, to port part of the computation to novel architectures such as FPGAs, GPUs, etc. He also said that the computation of long-range electromagnetic forces inhibited scalability to large number of processors.

HPC-NA Roadmap Presentation & Discussion

The full National HPC-NA Roadmap for Applications and Algorithms is published in a separate document; this report provides a summary of the presentation and previous work:

- Workshop 1: Oxford, Nov 2008
- Workshop 2: Manchester, Dec 2008
- Background work considering DOE/DARPA/NSF workshops
- Discussions with applications outside of workshops

The first version of the roadmap document is the outcome of the two community meetings together with input from similar activities elsewhere. The roadmap activity aims to provide a number of recommendations that together will drive the agenda toward the provision of:

- Algorithms and software that application developers can reuse in the form of high-quality, high performance, sustained software components, libraries and modules
- a community environment that allows the sharing of software, communication of interdisciplinary knowledge, and the development of appropriate skills.

The first version of the roadmap is built around five themes that have evolved during the discussion within the community.

- Theme 1: Cultural Issues
- Theme 2: Applications and Algorithms
- Theme 3: Software Engineering
- Theme 4: Sustainability
- Theme 5: Knowledge Base

Each of these is represented in the roadmap. As the roadmap activity goes forward we expect that these initial actions to develop into a detailed map of priorities across a sensible timeframe.

Algorithms

- Optimisation
- Scalable FFT
- Adaptive mesh refinement
- Eigenvalue/eigenvector (all or few)
- Iterative & direct solvers
- Monte Carlo
- Out of core algorithms to enable larger problems

Major issues for the future

The roadmap identifies a number of major issues of high importance that future work should be focussing on:

- Load balancing
    - meshes
    - particle dynamics and computation of interactions
- Better software environments for complex application development
- Adaptive software to automatically meet architectural needs
- Use of novel architectures (in the immediate future)
    - FPGAs
    - GPUs
    - IBM Cell

    - Other....
- Coupling between different models and codes
- Error propagation
- Scalable I/O
- Visualisation

Prioritization Axes

- Key applications
- Algorithms
- New approaches due to architectural issues
- Software development issues
- Skills
- time frame for each

APACE Website

The APACE website is planned as a solution to support the development of a community environment that allows the sharing of software and communication of interdisciplinary knowledge.

- **AP**plication **A**dvanced **C**omputing **E**xchange
- Community site built on same lines as myExperiment[5], a collaborative environment where scientists can publish their workflows and experiment plans, share them with groups and find those of others. Workflows, other digital objects and collections (called Packs) can be swapped, sorted and searched like photos and videos on the Web. myExperiment enables scientists to contribute to a pool of scientific workflows, build communities and form relationships. It enables them to share, reuse and repurpose workflows and reduce time-to-experiment, share expertise and avoid reinvention.
- APACE will facilitate collection of information around
    - numerical analysis algorithms,
    - definition of applications in terms of algorithms
    - Expertise in applications and algorithms
    - Global activity in development etc
    - Build community groups and sharing ideas, information and software

Issues Identified

Arising out of the discussions over the two days, a number of issues were identified that would improve and focus the initial draft of the HPC-NA roadmap

- Need to identify exemplar “baseline” projects
- Develop scenarios & timelines
- Prioritisation of themes and/or algorithms
- NA specific “actions” for roadmap
- Getting & retaining engagement from the various communities
- “Sustainability” as one of the themes or as a cross-cutting issue?
- Next step – EPSRC Network application – participation & ideas

These issues were taken up through three breakout groups focusing initially on issue 2 – the development of scenarios and timelines.

[5] http://www.myexperiment.org/

Summary of Breakout Group Discussions

Breakout Group 1: Numerical Aspects of HPC-NA

Prof. N.Higham reported on behalf of the breakout group. He highlighted a number of key points:

1. Numerical precision aspects

This arises arising from the non-IEEE compliant (single or double precision) arithmetics on GPUs and FPGAs, along with variable and fixed precision on FPGAs. Its importance is enhanced by the large number of time steps required by integrators (order 10^5 or higher), which magnifies rounding errors. This issue has arisen only recently and its importance has become increasingly apparent during the 3 workshops (as well as at the Jan 09 MRS Network workshop in Belfast).
Urgency: high

Timescale: short. Good progress can be made over the course of a 3 year project. Work is already underway in Manchester (Jan - Mar 09) as part of the MRS network to survey the literature and identify key applications where precision problems arise.

2. Error propagation in coupled models

In particular, this includes error control in adaptive PDE solvers, a topic mentioned in previous workshop reports.

3. Input from numerical analysts to applications scientists

This could take the form of advice on choice of algorithms. While it will be facilitated by APACE., numerical analysts would find difficult to find the time to provide "free consultancy" - their time will need to be costed.

4. Study Groups

The annual Smith Institute “study groups with industry”[6] in applied mathematics have been very successful. An analogous activity could be undertaken here in the form of numerical analysts and computer scientists working with applications scientists in intensive EPSRC-funded workshops focused on a small number of key applications. These are a necessary follow-on to the 3 workshops so far in order to delve deeper into technical aspects. Experience from the workshops suggests there are willing participants, subject to their availability.

Breakout Group 2: Applications and Algorithms

Dr S.Salvini reported on behalf of the group. The group discussed a number of numerical aspects common to a range of applications that could provide exemplar “baseline” PROJECTS

- Multiscale problems/simulations: encapsulation, manipulation of complex physical objects (i.e. with an internal structure, e.g. molecule, cellular structures etc) and their interactions.
  Timescale: long term.
- Long range interactions for particle models. Several speakers from different fields (molecular dynamics, plasma physics, astrophysics, material sciences, etc.) reported that this constituted a serious bottleneck that inhibited scalability to large numbers of cores. Current algorithms, mostly based on FFT, have proved inadequate and new ideas and solutions need to be sought.
  Time scale: short to medium term.
- Generalised Hermitian/symmetric eigenproblems arise in many fields (quantum chemistry, material sciences, etc). Standard LAPACAK/ScaLAPACK provisions do not scale satisfactorily

[6] http://www.smithinst.ac.uk/Mechanisms/StudyGroups/index_html

with increasing number of cores; in many cases most of the computation time is spent in solving these eigenproblems.

There was some discussion also about the delivery vehicles for algorithmic content, possibly beyond simple libraries, automatic code generation to achieve optimal performance on specific target systems, high level abstractions and their suitability and use (possibly along the lines of PLASMA). Ufortunately, the time allocated was not sufficient to explore these themes in sufficient depth.

Breakout Group 3: Infrastructure & Architectures

Dr John Brooke reported on behalf of the third breakout group. The group discussed on a number of issues of general import relating to infrastructure, system architectures and suggested some practical steps that should be undertaken.

The group noticed that the UK HPC landscape if currently dominated by the National Centres. At the same time, porting codes across architectures has proved a particular bottleneck because of the lack of a systematic approach.

In order to improve the situation, it would be essential to ensure future proofing of codes, to avoid any much time consuming re-engineering. Future UK funding and purchasing decision could be based on actual delivered performance.

A number of practical steps should be undertaken:

- In similar fashion to Germany, effort should be spent to make sure that technologies and know-how can “trickle down” from the National Centres to smaller installations. This would be of benefit not just to the academic research world but to industrial and commercial concerns.
- The NA community should provide insight and support to application developers
- The APACE website could be a good starting point, but it should provide more than just a library of algorithms and knowledge on how to employ them
- Coordination with those involved in supporting applications on high-end systems, e.g. CCPS, NAG, etc, would also be essential

Final Discussion

It was agreed that the existing version 1 of the HPC-NA Roadmap should be circulated widely at once, beyond the immediate circle of the Workshop participants. In the light of this third Workshop, a redrafted HPC-NA Roadmap should be put on the Website at once and circulated among the participants for comments and corrections. It was agreed that the final revised version would be completed by the 20th February.

There was also agreement that the HPC-NA Roadmap presented to EPSRC would contain the following “exemplar” applications:

- Numerical precision issues, raised by breakout group 1.
- Coupled problems, and error propagations in mixed models, as described by breakout group 1.
- Scalable algorithms for the modelling of long range forces, as described by breakout group 2.

It was also agreed that others should be solicited and could be proposed from outside the Workshop. These would be added to the basket of “exemplar” issues/applications in due course.

Computational chemistry was highlighted as another possible exemplar, focussing on the bottleneck of the Hermitian generalized eigenvalue problem (GEP), as identified e.g. by Dr Kenny and Dr

Sutherland. This would have the advantage of UK NA expertise, as well as wide interest in its application.

Establishing a Network activity as a follow-up to this series of Workshop was unanimously supported. It was generally perceived as an important step in bringing together numerical analysts, computer scientists and application researchers/developers.

APACE was supported by all present; it was also felt that a prototype should be set up as soon as possible.

The Meeting also recognised the importance of an international dimension to all UK efforts and funding towards this should be actively sought. European funding for projects within this general remit should also be applied for.

Short duration study groups as in applied maths were seen as a very good idea, provided funding could be secured

Annex 7: HECToR information

Applications running on HECToR in 2008

Code Title	Funding	Discipline
EPSRC Projects		
UK Turbulence Consortium	EPSRC	Engineering
Materials Chemistry HPC Consortium	EPSRC	Chemistry
GENIUS	EPSRC	Chemistry
Large scale MD and quantum embedding for biological systems	EPSRC	Materials
Optimization of HPCx LES code	EPSRC	Engineering
Numerical investigation of aerofoil noise	EPSRC	Engineering
Micromagnetic simulations on HPC architectures	EPSRC	Engineering
Fluid-Mechanical Models applied to Heart Failure	EPSRC	Physics
Joint Euler/Lagrange Method for Multi-Scale Problems	EPSRC	Engineering
Numerical Simulation of Multiphase Flow: From Mesocales to	EPSRC	Engineering
Parallel Brain Surgery Simulation	EPSRC	Life Sciences
Unsteady Propeller Noise	EPSRC	Engineering
Nonlinear modelling of tokamak plasma eruptions	EPSRC	Physics
Computational Aeroacoustics Consortium	EPSRC	Engineering
The Modelling of New Catalysts for Fuel Cell Application	EPSRC	Physics
DEISA	EPSRC	Support
Hydrogen vacancy distribution in magnesium hydride	EPSRC	Chemistry
Non-adiabatic processes	EPSRC	Materials
Computational Combustion for Engineering Applications	EPSRC	Engineering
Turbulence in Breaking Gravity Waves	EPSRC	Engineering
UK Applied Aerodynamics Consortium	EPSRC	Engineering
Hydrogenation Reactions at Metal Surfaces	EPSRC	Chemistry
Simulations of a Subsonic Cylindrical Cavity Flow	EPSRC	Engineering
Computation of Electron Transfer Properties	EPSRC	Chemistry
Ultrascalable Modelling of Materials	EPSRC	Materials
Quantum Monte Carlo Methods	EPSRC	Materials
Terascale DNS of Turbulence	EPSRC	Engineering
HELIUM Developments	EPSRC	Physics
Q-Espresso CP/PWSCF Codes on HECToR	EPSRC	Chemistry
SMEAGOL	EPSRC	Physics
e-Collision experiments using HPC	EPSRC	Physics
ONETEP: linear-scaling method on High Performance Computers	EPSRC	Materials
Ab initio study of high pressure disordered ice	EPSRC	Physics
Vortical Mode Interactions	EPSRC	Engineering
Study of Interacting Turbulent Flames	EPSRC	Engineering

Single molecule vibrational microscopy and spectroscopy	EPSRC	Materials
Model Parameters for Unsaturated Elasto-plastic Models	EPSRC	Engineering
Support for UK Car-Parrinello Consortium	EPSRC	Physics
Network modelling of wireless cities	EPSRC	Engineering
Dynamo Action In Compressible Convection	EPSRC	Physics
e94 Porting the Linear Scaling DTF Code Conquest to HECToR	EPSRC	Physics
Materials Property Relationships	EPSRC	Materials
Discovery of innovative hydrogen storage materials	EPSRC	Chemistry
Non-linear magnetohydrodynamic modelling of tokamak plasmas	EPSRC	Physics
New Developments in Modelling Electron Energy Loss Spectroscopy	EPSRC	Materials
Materials simulation using AIMPRO	EPSRC	Early use Materials
DNS of NACA-0012 aerofoil at Mach 0.4	EPSRC	Early use Engineering
Turbulent Plasma Transport in Tokamaks	EPSRC	Early use physics
Testing	EPSRC	Early use support
NERC projects		
Global Ocean Modelling Consortium	NERC	Environment
NCAS (National Centre for Atmospheric Science)	NERC	Environment
Computational Mineral Physics Consortium	NERC	Environment
Shelf Seas Consortium	NERC	Environment
BBSRC projects		
Biomarkers for patient classification	BBSRC	Life Sciences
Int BioSim	BBSRC	Life Sciences
Circadian Clock	BBSRC	Materials
External projects		
HPC-Europa	External	External
NIMES: New Improved Muds from Environmental Sources	External	Environment

CSE Support in 2008

The following contracts were put in place for CSE support in 2008 (additional have been created in the meantime).

1. OCEANS 2025 (NEMO) Dr Andrew Coward (University of Southampton)
2. Parallel Algorithms for Efficient Massively-parallel tools for the Study of Catalytic Chemistry Dr Paul Sherwood (Daresbury Laboratories) & Professor Richard Catlow (UCL)
3. Improving the parallelisation and adding functionality to the quantum Monte Carlo code CASINO Professor Dario Alfe (UCL)
4. Parallel Algorithms for the Materials Modelling code CRYSTAL Professor Nic Harrison (Daresbury Laboratories)
5. Future-proof parallelism for the electron-atom scattering codes PRMAT Dr Martin
6. Plummer (Daresbury laboratories)

7. Cloud and Aerosol Research on Massively-Parallel Architectures (CARMA) Dr Paul Connolly (University of Manchester)
8. Porting and Optimisation of Code_Saturne on HECToR and Black Widow Professor David Emerson (Daresbury Laboratories)
9. Improve scalability of Domain Decomposition within CP2K Dr Ben Slater (UCL)
10. WRF code optimisation for Meso-scale Process Studies (WOMPS) Dr Alan Gadian (University of Leeds)
11. Support for multigrid improvements to Citcom Dr Jeroen van Hunen (University of Durham)
12. Hybrid time-dependent density functional theory in the Castep code Dr Keith Refson (Rutherford Appleton Laboratory)
13. Performance enhancements for the GLOMAP aerosol model Dr Graham Mann (University of Leeds)

Annex 8: Findings from IESP

IESP working group on Technical challenges and needs of academic and industrial software infrastructure research and development; software scaling

This document contains the outputs of the subgroup looking at the software infrastructure at the second workshop for the IESP that was held in Paris in June 2009. It is included here for reference and is clearly only a snapshot of the material being created from that activity but provides further evidence and input to our own roadmap. The document as has been created from the slides on the IESP website and has been kept in the bulleted- form of those presentations. (http://www.exascale.org/iesp/IESP:Documents) The first part of the document sets the scene and is then followed by the outputs from the discussions of four subgroups.

Roadmap Formulation Strategy (strawman) for IESP2

- Consider each software component / area, in operation at centers or close to deployment
- If standard / open source component exists
  - Then investigate status quo circa 2009 wrt scalability
  - If project exists to enhance scalability
- Then identify roadmap until project termination
- If need to continue then identify the timeline gap till 2018-20/exa
  - Else (R&D gap identification)
- Identify research challenges envision project req.
- Attempt to create scalability timeline to 2018-20 exa
- Else (component does not exist in open source)
  - Identify why the component does not exist
  - Conduct R&D gap identification as above

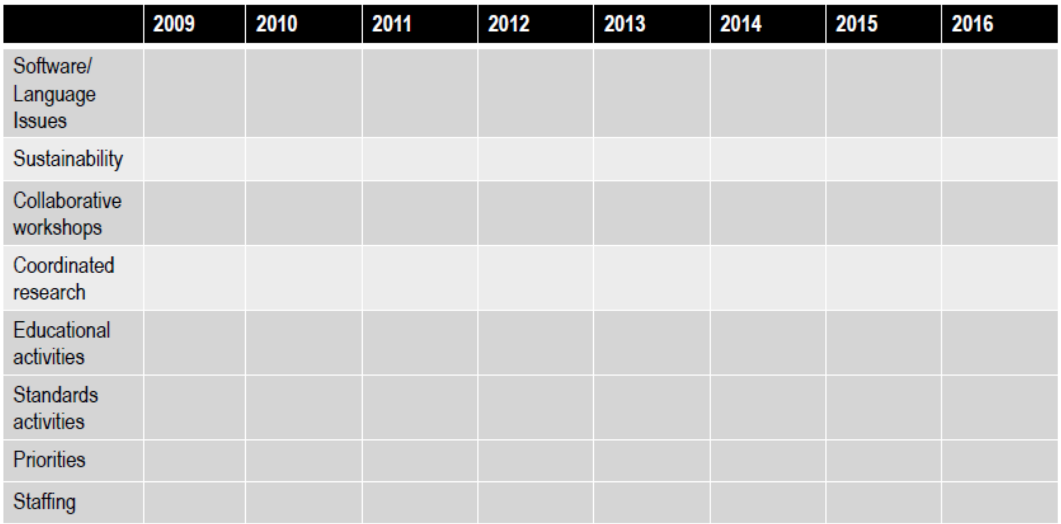
Roadmap Requirements (by Jack)

- Specify ways to re-invigorate the computational science software community throughout the international community.
- Include the status of computational science software activities across industry, government, and academia.
- Be created and maintained via an open process that involves broad input from industry, academia and government.
- Identify quantitative and measurable milestones and timelines.
- Be evaluated and revised as needed at prescribed intervals.
- Roadmap should specify opportunities for cross-fertilization of various agency activities, successes and challenges
- Agency strategies for computational science should be shaped in response to the roadmap
- Strategic plans should recognize and address roadmap priorities and funding requirement

Research Topics to consider (by Jack)

- Contributors
- Priorities
- Existing expertise
- SW sustainability
- Developing new programming models and tools that address extreme scale, multicore, heterogeneity and performance
  - Develop a framework for organizing the software research community
  - Encourage and facilitate collaboration in education and training

Roadmap/Milestone



	2009	2010	2011	2012	2013	2014	2015	2016
Software/ Language Issues								
Sustainability								
Collaborative workshops								
Coordinated research								
Educational activities								
Standards activities								
Priorities								
Staffing								

Strawman Roadmap Discussion (see Tech Assumptions Slide)

  - Memory requirements:
    - 3D layout in 2015.
    - Amount/core uncertain.
    - On-chip vs. off-chip performance difference grows.
- Migration from HD to SSD: How?
- Architecture support for GAS: Coming at some point.
- Node memory architecture:
    - SOC.
    - Deep memory hierarchy.
    - On-chip cache coherence, infeasible?
- Node PM: Will industry solve?
- Side comment: Bechtoldsheim: Up to half of industry might be in HPC.
  - FP precision: Role in this discussion?

Software Breakout Group 1A: Intra-node Day 1

Contributors: Kathy Yelick, Mike Heroux, Barbara Chapman, Mitsuhisha Sato, Mateo Valero, Jesus Labarta, Luc Giraud, Serge Petiton, John Shalf, Thomas Sterling, Jack Dongarra, Pete Beckman, Bernd Mohr, Jeff Vetter, Bill Gropp, Anne Trefethen

Agreement

  - Thousands of functional units on a socket
    - 100s of cores with 10s of FPUs per core (data parallel
  - hardware)
    - OR, 1000s of cores
    - OR, Lightweight PIM cores; heavy weight cores with dataparallel instructions and temporal locality needs
- DRAM Memory per FPU capacity will drop
    - Memory per socket will go up, but will not keep up with FPU growth
  - “Core” is not well-defined
    - PC / FPU >> 1 . multithreaded architecture
    - PC / FPU << 1 . vector/SIMD architecture
- Cache coherence across a 2020 socket is not feasible
- Memory bandwidth “wall” will be solved

    - Yes, technology will exists and market forces will cause this to happen (even outside HPC) Photonics, 3D stacking,…
- Memory latency problems will be solved by hardware and algorithms
    - Need massive concurrency, lightweight threading combined vectors
    - 1B way concurrency including threads, vectors, and/or prefetching

Programming Model Requirements

- Sustainability
    - Backward compatibility through interoperability
- Mixed language.
    - Incremental adoption support through interoperability
    - Composable
    - Portable and standardized
    - Develop in collaboration with industry
    - Familiarity with old models (no Haskel, remember that we are Fortran/C/C++ programming community).
- Address load imbalance (from the hardware, faults, and the applications/algorithms) (12)
    - Dynamic concurrency generation (Do you need it and can you afford it?)
    - Express more parallelism than you need (e.g., hierarchical task graph)
    - Dynamic runtime
    - Less synchronous than fork/join or SPMD
    - Dynamic runtime that relaxes to efficiency of a static model when dynamic is not needed.
- Ability to manage locality and data movement (9)
    - Make efficient use of bandwidth.
    - Scheduling for latency hiding
    - Ability to specify data scope and locality
    - Support for scratch pad memory
- Performance transparency and feedback (6)
    - Including runtime adaptation
    - Performance aware computing.
- Multiple precision support: ½, full, extended (3)
- Fault detection and ability to respond (1) Leave to inter-node sub-subgroup?. E.g., transient memory errors
- 1B way concurrency (1) Assumed true as part of memory wall solution.
- Support for strong scaling
- Global address space, but it is a challenge at exascale (1)
- Avoiding synchronization
- Message-driven computation for hiding communication and synchronization latency (1)
- One sided communication and active messaging, dataflow tokens
- Lightweight synchronization objects
- User-defined distributed data structures
- Energy management support (queries)
- Overlapping communication and computation

Breakout Group 1B:Inter-node Day 1

Contributors: Satoshi Matsuoka, David Skinner, Franck Cappello, Barbara Chapman, Alok Choudhary, Sudip Dosanjh, Yutaka Ishikawa, Barney MacCabe, Bob Lucas, Mateo Valero, Vladimir Voevodin

Inter-node topics

- I/O, I/O, I/O

- Resiliency (RAS,HA, Fault prevention, detection, recovery, management)
- Checkpointing
- Virtualization, Dynamic Provisioning, Partitioning
- Performance monitoring, feedback, parallel autotuning
- Resource provisioning (including heterogenous nodes, workflow scheduling)
    - Parallel programming models (distributed memory and PGAS)
    - Parallel debugging
    - Systems integration (SW bringup, factory testing)
    - Systems management (realtime, config management,)
    - Interaction with external resources : Clouds, Global FS (or non-FS), archiving, real-time data streams
    - Power and Facilities (SW only)
    - Security

Resiliency

- Faults everyday in PetaScale systems
    - Need to cope with continuous stream of failures
    - SW errors & HW errors, memory errors may dominate
    - MTTF < MTTC (mean time to checkpoint)
- Existing CPR falls short (full MPI CPR scales to 1k cpus)
    - Proactive actions (RAS analysis, Fault prediction)
    - Silent Errors (faults not monitored)
    - Lack Hardware Support (Diagnostics/Interfaces, SSDs?)
- Actions
    - Reduce number of errors, describe errors better
    - Make applications more resilient
    - Make systems situationally aware
- Roadmap & Research (immediate needs in 2012, figure out 2018 in 2012)
    - Fault oblivious algorithms, Error tolerant algorithms
    - Tunable advisement about key data/computations/communications to protect/check.
    - Relaxed model of correctness (from consistency)
    - Need C-PAPI for faults, make errors less silent
    - If faults are reproducible, inject them to test resiliency layers
    - Advanced diagnostics (monitor well-being of system, autonomous correction)

Software Infrastructure Intra-node Day 2

High level topics

- 1B concurrency and load balance (Thomas Sterling, Jesus Labarta)
- Locality and distributed data structures (Barbara Chapman, Vladimir Voevodin, Mitsuhisa Sato)
- Sustainability, especially interoperability (Bill Gropp, Mike Heroux)
- Operating systems (Thomas Sterling, John Shalf)
- Algorithms (Jack Dongarra, Anne Trefethen, Serge Petiton, Luc Giraud)
- Misc: performance.. (Bernd Mohr, Jeff Vetter, David Keyes)
- Fault tolerance (Kathy Yelick)

B way parallelism and load balance (Thomas Sterling, Jesus Labarta)

- Situation:
    - Need parallelism to feed the foreseable B cores hardware
    - Need further paralleism to let them tolerate latency
    - Dynamic scheduling for load balancing to tolerate not only algorithmic imbalances but the variance we are going to observe in platforms
    - There will be hierarchy both in algorithms and platforms (will they match?)

  - Need Low Overhead for synchronization and dispatch
- State of the art:
  - Generally static, compile/submission time specified
  - Based on preconception of knowledge of problem and machine
  - 100K processes and heterogeneous accelerators
  - Fork-join / spawn –wait / point to point synch + globally synchronizing (Zoltan, J-Machine,..)
- Needed Actions:
  - Develop advanced models of parallel programming models to expose dynamic parallelism
  - Develop advanced flow control model including advanced synchronization semantics and dependence handling mechanisms.
  - Runtime adaptive mechanisms and policies that converge to static if possible resource allocation
  - Methods for self aware resource allocation for dynamic load balancing
- Roadmap & Research: (immediate needs in 2012, figure out 2018 in 2012)
  - APIs for exposing fine grain/dynamic parallelism and enabling lightweight synchronization
  - Policies resource allocation and load balancing
  - Prototyping barebones parallel translation and runtime on “some” heterogeneous multicore

Managing Data Locality and Distributed Data Structures (Barbara Chapman, Vladimir Voevodin, Mitsuhisa Sato)

Situation: Locality essential for programming and performance in PetaScale systems, will be more complex in Exascale

- Extreme number of threads and memory distributed among nodes, cores and devices
- Complex cache hierarchy and resource sharing among cores
- New memory technologies are coming e.g. 3d stacking
- Explicit data transfer essential for use of accelerators
- State of the art: Programming models have different approaches to supporting locality
- Implicit and explicit distribution of data structures (e.g. MPI, PGAS languages, HPCS languages)
- Alignment of work and data (e.g. loop iteration mapping, Locale/place)
- Explicit data transfer between devices
- Use of manual program transformations to increase locality e.g. cache blocking

Needed Actions:

- Provide a model for the expression of scope and locality at both algorithmic and application code levels, especially for global view programming
- Develop techniques and features to enable efficient use of bandwidth and to support latency hiding
- Develop techniques for automatic optimization of data motion where possible, but user control for performance-aware programming
- Explore both implicit and explicit models for accomplishing locality including analyzable code

Roadmap & Research: (immediate needs in 2012, figure out 2018 in 2012)

- Features and notation for describing locality, e.g. algorithmic locality, user-defined distributed data structures, alignment of work and data, task mappings
- Develop support for data migration through non-conventional memory hierarchies (accelerators, scratchpads)
- Create tools for measuring, detecting, improving and exploiting locality
- Improve system-level approaches to managing locality
- Implicit locality models and automated locality support in long term

- Integrate with novel models for achieving concurrency and fault-tolerance for fine-grained state preservation recovery

Benefits

- For prefetch (identify data ahead of time)
- For software controlled memory (know what data needs to be copied in so you can set up your DMA transfers). Prefetch is just a different implementation
- For layout on-chip to reduce contention for shared resources: (because even on-chip there will be locality constraints will affect performance location of topological neighbors in a chip multiprocessor)
- For fault resilience (explicitly identify what data is changed by unit of computation so you know when it needs to be preserved)
- When dependencies are analyzed at even coarse-grained level, more freedom to reorder program units to increase slack for communication latency hiding and reorder for reuse between program units (not just) can restructure/reschedule at a program level
- Also enables functional partitioning to express more concurrency (makes it easier to create feed-forward pipelined parallelism when domain-decomposition reaches its limits

Algorithms & Software Libraries (Jack Dongarra, Anne Trefethen, Serge Petiton, Luc Giraud)

Situation: Algorithmic problems everyday in PetaScale systems, Exascale will be worse –

- Accumulation of round-off errors
- Adaptivity for architectural environment
- Fault resistant algorithms – bit flipping and loosing data (due to failures). Algorithms that detect and carry on or detect and correct and carry on (for one or more)
- Scalability : need algorithms with minimal amount of communication
- Coupling of multi-scale and multi-physics codes
- Amounts of data will increase (pre and post processing)

Roadmap & Research: (immediate needs in 2012, figure out 2018 in 2012)

- Fault oblivious algorithms, Error tolerant algorithms
- Hybrid and hierarchical based algorithms (eg linear algebra split across multi-core and gpu, self-adapting)
- Mixed arithmetic
- Energy efficient algorithms
- Algorithms that minimize communications
- Autotuning based on historical information
- Architectural aware algorithms/libraries
- Error propagation and sensitivity across mathematical and simulation models
- For application drivers identify key algorithmic areas

Sustainability (Bill Gropp, Mike Heroux)

Situation

- Huge software base
- Need to evolve legacy apps
- Need to introduce new models/approaches to address unique Exascale issues
- Need adoption strategy
- Industry recognition of multicore crisis in programming models

State of the Art

- MPI + C/C++/Fortran + OpenMP and/or pthreads etc.
- UPC, CAF; HPCS languages (research efforts); etc.
- Much of computational science primarily in serial parts of code
- No guarantee of interoperability between programming models
- Scalability demonstrated upto 100K nodes

Needed Actions

- Need effective multi/many core programming model(s)
- Need standards and/or mechanisms for efficient interoperability (no copies)
- Need interoperability with tool chain (debuggers, performance, OS, I/O)
- Determine division between industry and Exascale community

Roadmap and Research

- Research: Find commonality between models; common framework for describing interactions Memory model, synchronization model, etc.
- Research: Enhance (not replace) commodity multicore model for Exascale requirements (e.g., fault handling)
- Research: Tools to migrate legacy code to new models (e.g., exploit heterogeneous arch)
- Roadmap (short): Identify features to add to commodity programming models (incl MPI) and mechanism
- Roadmap (short): Identify issues in interoperability and composition
- Roadmap (long): Define and construct prototypes implementing interop and composibility in select pgmmodels
- Roadmap (long): Define and construct prototypes implementing tool chain/development environment

Operating Systems (Thomas Sterling, John Shalf)

Situation: Operating systems were designed with single processor or SMP node model

- Do not cope with heterogeneous hardware and nonconventional memory structures
- Poor scaling efficiency as cores/hardware added
- Serial path for exception handling (does not scale)
- Global locks for shared resources
  - Weak notion of locality and performance isolation
  - Requires cache-coherence and homogeneous ISA to work
- Unrecoverable fault result in kernel panic (reboot to recover from CPU error)
- Applications and runtime have very limited control of scheduling policy and resource management (OS interposes self with context switch for each protected/hardware resource request)

State of the art: Linux of various flavors assumes homogeneous shared memory system

- Hierarchical OS (offload OS calls to “service handlers”): e.g. Plan 9
- Lightweight Kernels (limited functionality, controls OS noise, mem footprint) e.g. CNK or CNL
- Full kernels (Full functionality, but complex, large memory footprint, OS noise) e.g. Linux

Needed Actions:

- Need to provide applications and runtime more control of the policy (scheduling & resource)
- Remove OS from critical path for access to resources (grant protected bare-metal access path and then get out of the way)
- Develop global namespace management
- Interoperability between local functionality and global functionality (e.g. make TLB be integrated with global memory model, global resource discovery and namespace management)
- Need to support for managing heterogeneous computational resources and non-cache-coherent memory hierarchies
- Expose mechanisms for finer reporting and control of power management (provide to app and runtime)
- Scalable parallel, locality-aware, interrupt dispatch mechanism
- Develop QoS mechanisms for managing access to shared resources (on-chip networks, memory bandwidth, caches)

- Scalable mechanisms for fault isolation, protection, and information propagation to application and runtime on-chip (for transient hardware errors and software errors)

Roadmap & Research: (immediate needs in 2012, figure out 2018 in 2012)

- Establish performance metrics and models to quantify scaling opportunities and robustness for global OS model
- Remove OS from critical path for hardware policy access
- Define asynchronous API for system calls with abstract service location for satisfying calls and global namespace approach.
- Derive an experimental platform (simplified OS) with strawman functional elements and interrelationships to facilitate exploration and quantification of competing mechanisms for managing OS concurrency. Quantify benefits, complexity, and hardware support requirements for competing approaches.
- Implement test X-OS experimental platform on medium/large scale testbed to integrated with global OS namespace and management features

Performance (Bernd Mohr, Jeff Vetter, David Keyes)

Situation:

- Functionality, Correctness, and only then, performance is taken care of (Telescoping)
- Too manual and labor-intensive
- Limited applicability and usage of performance models

State of the art:

- Simple statistical summaries, at best snapshots over time
- Can handle 25K events/s per thread => 5min, 64k threads => 2-10TB, mainly only thread count will increase
- Emphasis on data presentation rather than on analysis and necessary optimization

Needed Actions:

- Performance-aware design, development and deployment
- Integration with compilers and runtime systems
- Support for performance observability in HW and SW (runtime)
- Need more intelligence in raw data processing and analysis
- Support for heterogeneous hardware and mixed programming models

Roadmap & Research: (immediate needs in 2012, figure out 2018 in 2012)

- Make sure can handle envisioned number of threads in 2012, 2015, 2018
- Integrate performance modeling, measurement, and analysis communities and agendas
- Ensure performance-aware design of hardware, system software, and applications

Programming Model Support for Fault tolerance (Kathy Yelick)

Situation: Faults everyday in PetaScale systems, Exascale will be worse

- Need to cope with continuous stream of failures
- SW errors & HW errors, memory errors may dominate
- MTTF < MTTC (mean time to checkpoint)
- Need to distinguish failures from full system interrupts
- Detection problem alone will be major challenge

State of the art: Programming models assume fault free

- Research on fault tolerance MPI

Needed Actions:

- Programming model support for fault detection
- Programming model support for recovery (transactions, retry,…)

Roadmap & Research: (immediate needs in 2012, figure out 2018 in 2012)

- 2012: Model needed for classify faults and ability to tolerate (2012)
- 2015: Languages and compilers for hardware faults (2015)

  - Memory errors first
- 2018: Languages, compilers and tool support for software faults (2018) E.g., retry for rarely found race conditions; Parallel debugging of 1B unsolved

Breakout Group 1B: Day 2

Members: Satoshi Matsuoka, David Skinner, Franck Cappello, Barbara Chapman, Alok Choudhary, Sudip Dosanjh, Yutaka Ishikawa, Barney MacCabe, Bob Lucas, Mateo Valero

Inter-node topics

- I/O, I/O, I/O (HD/SSD, local, global parallel FS, non-FS) (1) (choudhary, ishikawa)
- Resiliency (RAS,HA, Fault prevention, detection, recovery, management) (1) (cappello)
- Virtualization, Dynamic Provisioning, Partitioning (3) (maccabe)
- Performance monitoring, feedback, parallel autotuning (2) (skinner)
- Resource provisioning (including heterogeneous nodes, workflow scheduling) (2)
- Parallel programming models (distributed memory and PGAS) (1)
- Parallel debugging (?, what is the programming model) (sudip)
- Eliminate bottlenecks to strong scaling (hidden latencies in SW) (1) (lucas, nakashima)
- Systems integration (SW bringup, factory testing, transition to production SW state) (3) (skinner)
- Systems management (realtime, config management, SW change management) (3)
- Interaction with external resources : Clouds, archiving, real-time data streams (2)
- Power and Facilities (SW only, thermal process migration, energy scheduling/charging) (2) (matsuoka)

I/O (Alok Choudhary, Yutaka Ishikawa)

Situation: Scalable I/O is critical - Scalability problems

- Programming and abstraction (how is I/O viewed from 100K+ processes), Is the file I/O abstraction necessary (e.g., what about highlevel data persistence models including databases)
- S/W Performance and optimizations (BW, latency)

State of the Art

- Applications use I/O at different levels, in formats, using different number of layers

Needed Actions

- Think differently? Purpose of I/O (e.g., checkpointing at OS/ application level, persistent data storage, data analytics, use it and throw away?); customized configurations?
- Define architecture hierarchy abstraction from S/W perspective

Roadmap and Research (Immediate Need, Intermediate, Long Term)

- Newer models of I/O (high level, DB, elimination of dependencies on number of nodes)?
- Exploitation of new memory hierarchy (e.g., SSD) for S/W layers, optimizations
- Power/performance optimizations in I/O
- Intelligent and proactive caching mechanisms
- Integration of data analytics, online analysis and data management
- Data provenance/management
- Derive I/O requirements from users or workload analysis

Parallel Debugging (Sudip Dosanjh)

Situation: Significant topic of research for many years

- Tools are available for applications with 1,000 MPI tasks
- Very few applications execute the first time on 10's of thousands of cores (even very mature, widely used codes)
- Debugging usually requires lots of time by expert parallel programmers

State of the Art:

- TotalView, Allinea's Distributed Debugging Tool
- Early work on a light-weight debugger

Needed Actions:

- Current methods will not scale to exascale. Most application programmers don't want to debug a code with 100,000 or 1,000,000 MPI tasks. A fundamentally new paradigm is needed.
- Automated tools and formal methods

Roadmap&Research

- 2010: Suggestion for an applications readiness team workshop, plan for community building/ information sharing
- 2012: Light-weight debuggers are needed that can supply limited information for 100,000 MPI tasks
- 2012: Simulation/testing tools are needed for large task counts -- i.e., so programmers can test their codes on O(1M tasks) on smaller systems
- 2015: Breakthrough needed for O(1M) tasks
- 2018: Near-production Exascale tools

Performance Monitoring and Workload Analysis (David Skinner)

Situation: At petascale application walltimes are variable and mysterious, exascale

- Workloads are studied anecdotally based on narratives or very limited data
- Performance often ascribed to an application as opposed to a series of runs on a specific machine
- Performance engineering done ex-situ sometimes away from users and production setting
- Good serial interfaces for perf data (PAPI), but with limited big picture view (no parallelism)
- Inter-job contention almost unstudied, poorly understood, aggravating at petascale toxic at exascale

State of the art:

- Performance monitoring of HPC resources lags that seen in autos (a dashboard)
- A spectrum of tools are needed (low overhead profiling, intelligent tracing, deep dive perf debug)
- Low overhead (< 2%) application profiling available at terascale, barely working at petascale
- Tool scaling varies from easy to use to heroic efforts, no one tool will meet all needs
- Asymptotically poor performance (failure) often undiagnosable, trial and error approach

Needed Actions:

- Tools must become composable allowing for deep dive tool use as well as background monitoring
- Continuous performance reporting required to maintain basic system operation + opt-in tools
- Pre-emptive verification of requested resource performance prior to job launch
- Shine light on app/system interactions, system status, contention weather, resource conflicts, wasted resources
- HPC managers and users need easy to use methods to provide common basis for productive performance dialogue
- Workload analysis will allows HPC facility managers to better procure, provision, schedule resources to mitigate contention

Roadmap & Research: (immediate needs in 2012, figure out 2018 in 2012)

- Extend performance counters to power, cooling, faults, interconnect, filesystem
- Accurate descriptions of job lifecycle (how long to start, reason failed, resources consumed, actions needed)
- Integrate multiple levels of monitoring into high level user and system contexts

- Clustering, compression, stream sketching, and synopsis generation of performance data (web methods)
- Data mining for performance prediction, workload patterns, and online analysis of system and applications
- Synthesis of measurements to connect inputs and outputs to research goals (high level metrics)

Virtualization, Dynamic Provisioning, Partitioning (Barney MacCabe)

Situation:

- Virtualization provides "a level of indirection" between the application and the systems
- Operating systems can exploit this level of indirection to support properties that are not associated with the physical system (e.g., dynamic node allocation, migration after a fault, etc)
- Isolation between applications is critical when multiple applications share a single system
- Applications frequently have very subtle dependencies on specific library and/or OS versions, there is a need for applications to "bring their own OS" to the nodes

State of the art:

- Applications are allocated a static a virtual machine (partition) at load time
- Systems like Blue Gene provide direct support for partitioning and isolation in the network, in other systems this partitioning and isolation is a key part of the system software dunning on the nodes
- Node virtualization is rarely supported, although Blue Gene easily support rebooting nodes when applications are launched

Needed Actions:

- Clarify programming models needs
  - what is dynamically provisioned?
  - Do all virtual nodes make progress when the number of virtual nodes exceeds the number of physical nodes?
  - What is the overhead? Can this overhead be eliminated for applications that do not
- Clarify other benefits that might accrue from node virtulaization
- Dynamic migration after fault detection or to balance resource usage

Roadmap & Research: (immediate needs in 2012, figure out 2018 in 2012)

- Support for, light-weight, node virtualization that supports a common API (e.g., the Xen API)
- Direct support for light-weight virtualization mechanisms (no node OS) based on small computational units (e.g., Charm++, ParalleX)
- Better understanding of the needs of computational and programming models

Power and Management (Satoshi Matsuoka)

Situation: 100TF-1PF systems at 500KW-6MW today, 100MW or more envisioned for Exascale

- Beyond Moore's law scaling is pushing power/energy requirements as systems grow larger
- Power/Energy may become fundamental limiting factor---$10s millions , CO2 footprint
- Need to drastically reduce energy consumption to be commercially feasible

State of the art: leveraging some datacenter/notebook power saving features

- DVFS (Dynamic Voltage & Frequency Scaling) within application
- System-level resource management tied to scheduling and DVFS
- Manual optimization of datacenter cooling to reach "reasonable" PUE ~= 1.5-2.0

Needed Actions:

- Altenative architectures and devices with fundamentally order(s)-of-magnitude better power-performance
- characteristics and their exploitation in SW, e.g., GPUs, phase change memory, SSDs, …

- Measure/predict power&energy, based on underlying sensors and power-performance models
- Aggressive cooling technologies (e.g., ambient cooling) coupled with machine oprations e.g., packing processes to achieve higher thermal gradient
- Auto-tune/optimize the entire system for best energy-performance levels, achieving the necessary x10 improvement beyond x100 offered by Moore's law over 10 years.

Roadmap & Research: (immediate needs in 2012, figure out 2018 in 2012)

- 2012: various software artifacts to exploit alternative architectures/devices along with their power models, open source sensor/monitoring framework to measure power and thermals on a 10,000 node scale machine
- 2015: comprehensive system simulation auto-tuning techniques to allow optimization, workload management tied to thermals/energy for energy optimization
- 2018: scaling of the software artifacts above to reach 100,000 node / 100million core exascale system with sufficient response time for effective energy and thermal control

Exascale Software Integration (David Skinner)

- Situation: System software, middleware, and system configuration settings are brittle
  - During machine bring up finding proper software configuration is a complex expensive search
  - Maintaining an optimal or reasonable state over time, across upgrades is difficult
  - Terascale downtimes are expensive, Exascale brings both greater complexity and cost of outage
  - Users bear the brunt of poorly integrate/configured software
- State of the art:
  - Impact of multiple changes often unverifiable
  - Change management not in widespread use (plan for changes, review changes)
  - Cfengine is not a change management plan
  - Many centers operate without test machines
- Needed Actions:
  - Transition from "out of the box" approach to a "trust and verify"
  - Revise expectations about when integration and testing happens
  - Incremental testing as system is brought up
- Roadmap & Research:
  - Improved software engineering across the board (users, HPC centers, industry partners)
  - Redefinition of acceptance period to allow for software bringup
  - Alternately add a SW integration period
  - Automated SW testing at all levels, connection to SLA/expectations

Removing Bottleneck to Strong (& Weak) Scaling (Bob Lucas)

- Situation:
  - many algorithms have some O(N) portions which should be severe bottleneck when N grows to 10^8-10^9
  - most scaling today is weak but memory/core will (should) decrease and even if exa-weak-scalable the problem should become TOO large
  - most systems provide communication means with high latency.
- State of the Art
  - MPI & PGAS programming models
  - algorithms (e.g. math kernels) with sequential bottleneck (e.g. dot product) and/or with global state
- Needed Actions = SLOW (thanks Tom)
  - Starvation: expose & manage concurrency
  - Latency: minimize & tolerate (hide)

– Overhead: minimize & expose fine-grain concurrency
– Waiting for Contention: remove global barriers

- Roadmap & Research (near, medium & long)
  - algorithms: force/allow app. guys to use nice math library
  - prog. models: global addr. spac, light weight sync., coarse-grain functional or dataflow
  - h/w mechanisms: active memory / messaging
  - near: remove barrier (& other global comm.) and replace with asynchronous global flow control which also is capable to hide global latency.
  - medium: algorithm research to eliminate any O(N) (or higher) portions
  - ultimate goal: to reach real speed-of-light (& physics) limits

Resiliency (Franck Cappello)

- Situation (problem statement, why a relevant topic):
  - Faults everyday in PetaScale systems, Exascale will be worse
  - Need to cope with continuous stream of failures (applications will have to resist to several errors, of different kinds, during their execution)
  - SW errors & HW errors, Undetected Soft errors (Silent errors) are already a problem. SW errors may dominate
- State of the art (and limitations):
  - Checkpointing on remote file system, however: MTTI <= Checkpoint time for Exascale systems
  - Proactive actions (RAS analysis, Fault prediction), however: 1) how to manage predicted software errors? 2) we need more event traces to improve fault prediction algorithms
  - Silent Errors (faults not monitored) are suspected and sometimes detected afterwards, however their is a need of characterization (what, where, how frequent, etc.)
  - Fuzzy event logging: Some errors are silently corrected (and so not reported) + some errors are reported by humans and not well integrated
  - No coordination between software layers (and errors are not reported across layers)
  - Almost no hardware support for Resilience (except at the node level), however we need to detect more errors and ease the job of the software
  - No experimental platforms
- Needed Actions:
  - Investigate Checkpoint/Restart limitations
  - Make applications more resilient
  - Develop novel application level tunable resilience techniques
  - Develop coordination mechanisms from HW to applications (through all SW layers)
  - Make errors less silent
  - Improve interaction mechanisms between automatic error correction systems and humans
  - Develop experimental platform
- Roadmap & Research: (immediate needs in 2012, figure out 2018 in 2012)
  - HW, SW, Soft, Silent Error characterization in HPC systems (Immediate, but should ve revised periodically)
  - Investigate the current scalability and bottlenecks of MPI checkpoint/restart (immediate)
  - Investigate how to reduce these bottleneck (in-situ checkpoint, hardware support: SSDs, Networks) (immediate)
  - Fault oblivious algorithms, Error tolerant algorithms, ABFT like techniques (medium and long term)
  - Tunable advisement about key data/computations/communications to protect/check. (Relaxed model of correctness -similar to consistency for memory)
  - (immediate and medium term)
  - Need uniform interface for faults, Improve hardware support (more sensors, detectors for Diagnostics/Interfaces) (medium term)

- Improve error description and report, make systems situation aware, provide advanced diagnostics (monitor well-being of system, autonomous correction) (medium term)
- Design and implement experimentation platform with sophisticated fault injectors (immediate)