DEVELOPING A HIGH-PERFORMANCE COMPUTING/NUMERICAL ANALYSIS ROADMAP

Anne Trefethen[1]
Nick Higham[1]
Iain Duff[2]
Peter Coveney[3]

Abstract

A roadmap activity in the UK has leveraged US and European efforts for identifying the challenges and barriers in the development of high-performance computing (HPC) algorithms and software. The activity has identified the Grand Challenge to provide:

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

Through a series of workshops and discussions with UK HPC application groups and numerical analysts, five areas of challenge have emerged.

Key words: high-performance computing, numerical analysis, roadmap, applications and algorithms, software

The International Journal of High Performance Computing Applications,
Volume 23, No. 4, Winter 2009, pp. 423–426
DOI: 10.1177/1094342009347710

1 The High-performance Computing/ Numerical Analysis Roadmap Activity

1.1 Methodology

The activity 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 1; 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.

1.2 Roadmap Themes

A number of themes have been developed through the consultation. Not surprisingly 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.

1.2.1 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.

[1] UNIVERSTIY OF OXFORD, OXFORD, UK (ANNE.TREFETHEN@OERC.OX.AC.UK)

[2] COMPUTATIONAL SCIENCE AND ENGINEERING DEPARTMENT, RUTHERFORD APPLETON LABORATORY, UK.

[3] DEPARTMENT OF CHEMISTRY, UNIVERSITY COLLEGE LONDON, UK.

Development of a Community

- There was a general desire to organize further 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.

1.2.2 Applications and Algorithms

Here we provide a high-level view of some of the issues and challenges for applications and algorithms; a more detailed and in-depth consideration can be found in the final report of the first stage of the activity on the project website[1].

Cross-application Commonality

- There is a commonality of algorithms across the applications that allow advances in a particular algorithm to have a broader impact – these have been identified for the leading applications in the UK.

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 a number of issues, including a lack of standards for data models and formats, interoperability of programming models and lack of knowledge of error propagation through the integrated system.
- In many applications there is a pipeline of activities: firstly, setting up the model; then the actual calculation; and finally visualization 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 recognized 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.

Adaptivity

- There is a need to have adaptive algorithms to 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, achieving scalability is a huge problem for many application areas. The desire to solve larger 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 there is a need to develop algorithms that minimize communications to enable that scaling.

Partitioning and Load Balancing

- As systems become larger and more heterogeneous, load balancing and problem partitioning will be increasingly difficult. The need for dynamic load balancing may arise from hardware, faults or the application/algorithm 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 often 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.
- The 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 a larger number 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.

1.2.3 Software

Language Issues

- There is a need for mixed-language support as 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 compatibility, 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

- It is necessary to have the ability to manage locality and data movement and to schedule for latency hiding.
- Performance transparency and feedback are needed to provide the user with a layering of capability and tuning.
- The capability to control energy efficiency provides the most optimal use of hardware to minimize energy usage and to allow the control of hardware (e.g. the ability to shut down hardware and wake-up appropriately).

Support for the 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

- There were concerns from many application developers that there are no 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 and 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.

1.2.4 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.

1.2.5 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 helpful to the community to develop mechanisms for collecting information on existing software and tools and disseminating this 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 in mathematics, software engineering and high-performance computing (HPC). 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 are likely to be areas 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.

2 Conclusions

This activity is ongoing and the UK roadmap for algorithms and applications will continue to evolve. For more a more detailed analysis of the findings and additional information see http://www.oerc.ox.ac.uk/research/hpc-na.

Acknowledgements

We are grateful for the support provided by the Engineering & Physical Sciences Research Council (EPSRC) for the algorithm/application roadmapping activity. The activity has benefitted from the work of the team at OeRC and in particular Mark Hylton and Stef Salvini.

Author Biographies

Anne Trefethen is the Director of the Oxford e-Research Centre and Professor of Scientific Computing at Oxford University. She has been active in HPC applications and algorithms since 1989, starting with four years at Thinking Machines Corporation Inc. developing linear algebra algorithms for the Connection Machine Scientific Software Library (CMSSL). She was a researcher in parallel computing at the Cornell Theory Centre, one of the NSF's four national HPC facilities, where she later became the Associate Director for Computational Support and Software. From 2001 to 2005 she was deputy Director, then Director of the UK e-Science Core Program.

Nick Higham is Richardson Professor of Applied Mathematics in the School of Mathematics, University of Manchester. His degrees (BA 1982, MSc 1983, PhD 1985) are from the University of Manchester and he has held visiting positions at Cornell University and the Institute for Mathematics and its Applications, University of Minnesota. He is Director of Research within the School of Mathematics, Director of the Manchester Institute for Mathematical Sciences (MIMS) and Head of the Numerical Analysis Group. He was elected Fellow of the Royal Society in 2007, is a SIAM Fellow, and held a Royal Society Wolfson Research Merit Award (2003–2008). He has more than 90 refereed publications on topics such as rounding error analysis, linear systems, least squares problems, matrix functions and non-linear matrix equations, condition number estimation and generalized eigenvalue problems.

Iain Duff is a CCLRC Senior Fellow in the Computational Science and Engineering Department. After completing his D Phil at Oxford, he was a Harkness Fellow in the United States visiting Stony Brook and Stanford. He then spent two years lecturing at the University of Newcastle upon Tyne before joining the Harwell Laboratory where he became Group Leader of Numerical Analysis in 1986. He has had several extended visits to Argonne National Laboratory, the Australian National University, the University of Colorado at Boulder, Stanford University and the University of Umea. He is the Project Leader for the Parallel Algorithms Group at CERFACS in Toulouse and is a Visiting Professor at the University of Strathclyde. His research interests include numerical linear algebra, sparse matrices, parallel computing, scientific computation and mathematical software.

Peter Coveney, BA MA DPhil (Oxon) CChem CPhys FRSC FInstPhys, holds a Chair in Physical Chemistry, is Director of the Centre for Computational Science (CCS) and is an Honorary Professor in Computer Science at UCL. His research includes atomistic, mesoscale and multiscale modeling, including quantum and classical molecular dynamics, dissipative particle dynamics, lattice gas and lattice-Boltzmann techniques, and he exploits state of the art HPC and visualization methods. He is the recipient of an HPC Challenge Award at Supercomputing 2003 for the TeraGyroid Project, an inaugural HPC Analytics Challenge Award at SC05 for the SPICE Project and the International Supercomputing Conference Awards in 2004 and 2006. He is leading the Virtual Physiological Human (VPH) Network within the EU's seventh Framework Program as part of the VPH initiative with the ambitious goal for integrative biomedicine.

Note

[1] www.oerc.ox.ac.uk/research/hpc-na