Defining Requirements, Meeting Requirements

Phillip Colella

Applied Numerical Algorithms Group

Computational Research Division, LBNL

NERSC ASCR Requirements for 2017
January 15, 2014
LBNL

Research projects

- Base program research in numerical methods for partial differential equations.
- Participation SciDAC FASTMath Institute activities in structured grid and particle methods.
- "Exascale" research: participation in ExReDi project (RXSolver program) and D-TEC and XTunes projects (XStack program).
- End-to-end applications development (see talks by Martin, Trebotich).

Research projects

- Multiple applications stakeholders

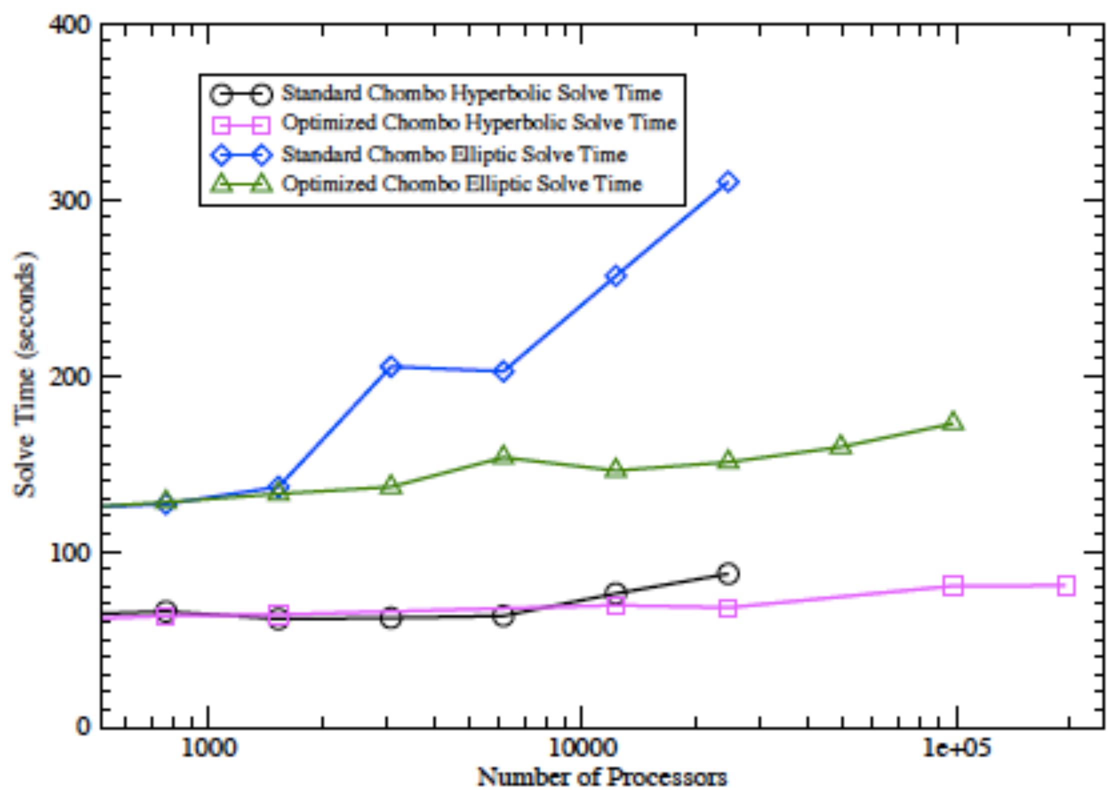
- Anticipate challenges and develop solutions.

- Engagement with the rest of the HPC community: domain scientists, computational mathematicians, computer scientists.

We are both part of the problem and part of the solution.

Chombo Framework

- Layered framework for partial differential equations.
- Supports a broad range of applications.
- Bulk-synchronous parallelism, with locally-static domain decomposition.



Challenges - Applications

- Kinetic problems (4-6 dimensional independent variables, + time). Fusion, cosmology, accelerators.

- Eddy-resolving (oceans) / cloud-system resolving (atmospheres) models for CFD in climate.

- HPC implementations of complex models in systems biology with rapid turnaround.

Challenges - HPC Computer Systems

- More complex processor architectures and deeper memory hierarchies.

- Heterogeneity.

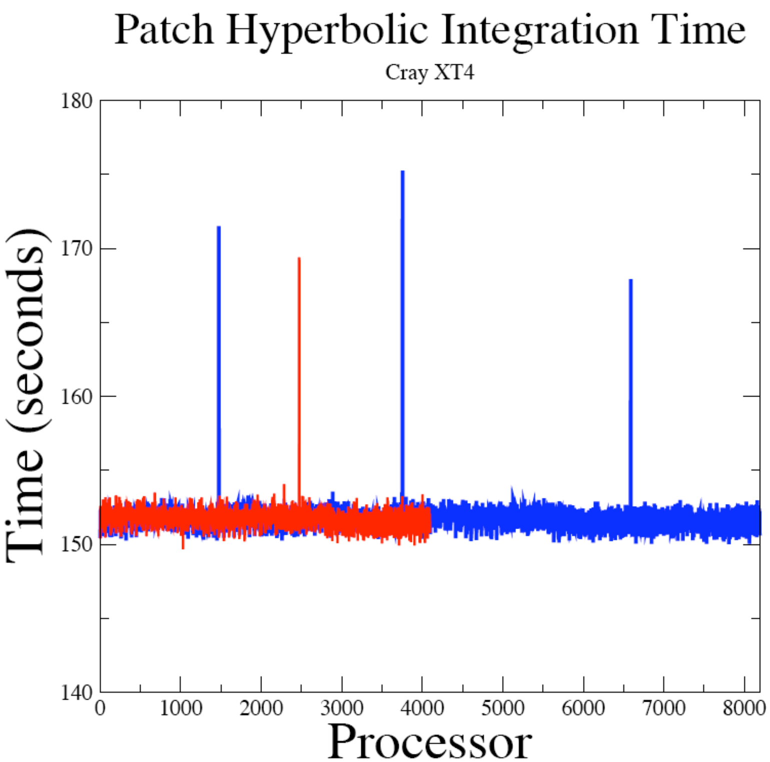
- Fault tolerance. In the short term, a form of heterogeneity. Errors are either detected and corrected, or detected and cause node failure. The latter can be dealt with using distributed redundant storage (containment domains).

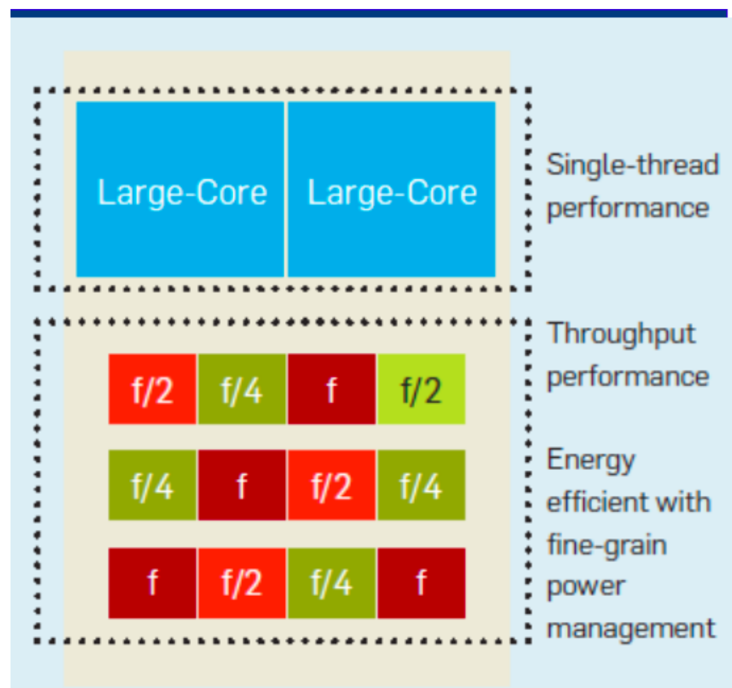
Response to these Challenges: Mathematics

- Communication-avoiding methods for for Poisson's equation based on potential theory; FFT-based AMR methods for electromagnetics (Computational kernel: FFTs for small 3D grids on a single node).
- AMR / Embedded boundaries for PDEs on a sphere; AMR in phase space; complex geometries. (Anisotropic solvers, visualization and data analysis).
- Two-grid methods for PIC (fast parallel sorting).
- Ameliorating the solver bottleneck in PIC by new approaches to time integration.

Addressing Challenges: Computer Science

Locally-static load balancing is a dead end. How do we program heterogeneous systems, algorithms, models and still get decent performance ?





Addressing Challenges: Computer Science

Reuse of a framework across applications is obtained by use of a layered architecture, with callbacks to application-dependent code. This leads to unnecessary reads / writes to DRAM.

- Monolithic code for each application: no reuse, expensive to maintain and extend.
- Embedded DSLs /compilers that eliminate unnecessary loads and stores by fusing framework and application-specific code at compile time. Requires significant investment by the facilities and buy-in from developers/ users.

Addressing Challenges: Computer Science

In fields where models or their coupling are poorly understood (e.g. systems biology), you want to be able to experiment with models and discretizations with a rapid turnaround time (days to weeks), and have these models run at near-production performance.

- Tool-rich environment: algorithm components, workflow tools, data analysis and visualization.
- High-level DSLs that are expressive of the algorithms for a specific application.

Approach: choose a specific science domain as a focal point for development of such a toolchain.

Final Comments

I have deliberately omitted any discussion of present or future hardware requirements.

- I believe the critical performance bottlenecks are software in the 2017 timeframe and beyond.
- The costs, and lead times, for software are long, so now is the time to start discussing them.
- Who owns what part of these problems: vendors, facilities, R&D community ? How do we coordinate ?
- We need a layered organization that reflects the layers in the enterprise: science questions and models <-> mathematical formulation <-> discretization methods <-> HPC software <-> programming systems <-> computer systems.