SAND2007-0894C

SNL Perspectives on Petascale Environments

ASC Level 2 Petascale Environment Planning Meeting
Las Vegas, NV
February 23, 2007

James R. Stewart
Sandia National Laboratories
Albuquerque, NM

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

Challenges to PetaScale

- Software and Algorithmic Issues
- Hardware Issues
- Pre and Post Processing
- Usability, Reliability, Porting and Testing

Software and Algorithmic Issues

- More complex programming models will be needed for our mechanics codes
  - Will MPI be sufficient?
  - What can we afford?
  - Will the software environment be mature enough to support these complex programming models?
- New models should cover full spectrum of heterogeneous parallelism
  - Multicore through petascale
  - Grid computing

Example

- With more processors, there will be a problem with longer synchronization times for barriers and global reductions (e.g., global dot product). This could be a performance killer.

What Programming Models Might Work?

- Nested programming models
  - MPI with something else at the node?
- Cluster of clusters
  - Split MPI_COMM_WORLD into a different communicator for each cluster
  - Use a single communicator between the master processes of all the clusters
  - Implies need for new communication backbone
    - *Fast, low-latency* communication within a cluster
    - *Fast, low-latency* (but perhaps not as fast as above) between master processes of a cluster
    - *Low-latency between any two arbitrary nodes in different clusters not needed!*

Implications for Multi-Physics Coupling

- We will need to rethink our multi-physics code coupling strategies to effectively use any new programming model
  - *Whole-machine coupling* (where each processor shares pieces of each single-physics application) should be avoided
  - Instead, let each single-physics piece "own" an appropriate processor subset (perhaps aligned with a cluster…), swapping information with other physics only at synchronization points

Implications for Multi-Physics Coupling

- NEW ISSUE: Exchange information (between different physics/clusters) less often, overall performance improves; however, additional numerical errors or mathematical instabilities could arise
  - This will be a research area

There will be tradeoffs between coupled multi-physics performance and numerical accuracy

Other applications (e.g., “multi-point” problems such as parameter estimation optimization) would be well-suited for the cluster of clusters model

Sandia National Laboratories

Software and Algorithmic Issue: Linear Solvers

- Achieving scalability to and beyond 1000's of processors is a challenge
  - FETI is an iterative solver used by the Salinas and Adagio solid mechanics codes
    - FETI uses an *algebraic multigrid strategy* where the each grid level is a function of the number of processors
    - With a very large processor count, solving even the coarsest level becomes an issue, and could impede scalability
- Scalable Algebraic Multigrid is a critical research area
  - Subject of current LDRD
  - New technical approaches being tested in the Fuego fluid mechanics code

Hardware Issues
Communication vs. Computation

- Need faster communication relative to computation speed
- Example: Crash simulation using Presto solid dynamics code
  - Very fine mesh leads to very small time steps
  - Currently would need 50-100 days of runtime on 1000's of processors
  - Contact-dominated, implying lots of inter-processor communication!
  - Scalability currently levels out at ~4000 elems/proc
  - To approach petascale performance, would need faster communication to allow scalability down to ~400 elems/proc

Hardware Issues

Communication vs. Computation

- The need for highly efficient scalability (i.e., speedup) for a fixed total problem size is ubiquitous!
  - Must efficiently handle a small computational workload per processor

Hardware Issues
Memory Bandwidth vs. Computation

- Increasing imbalance in computation vs. memory bandwidth performance (ever-decreasing bandwidth per operation)
  - Our algorithms are already bandwidth-limited
  - We will get far less than linear improvement on large-count multicore systems, regardless of how we program them
  - We can make improvements in algorithms and data structures, but this trend will be an issue

Need improved memory bandwidth

Sandia National Laboratories

Pre and Post Processing

- Meshing becomes a real concern
  - Increases pressure on the need for faster model creation (currently a big bottleneck for end-to-end analysis efficiency)
  - Meshes will become larger; Meshing algorithms sufficient? Huge data transfers required if meshing can't be done on the petascale machine itself
    - Cubit group is currently working on a parallel hex meshing capability and is ready to begin testing on Purple or Red Storm

Pre and Post Processing

- ExodusII database can currently only handle meshes of ~250M elements
  - This finite-element database is used by many mechanics applications at Sandia
  - We already have a request by the Fuego team to enhance ExodusII to handle larger meshes so they can run a scaling study to test their new solvers on Red Storm and Purple
  - The drive toward Petascale will increase the need for these kinds of improvements -> significant investments will be required!

If you build it, will they come?

(The trip won't be cheap!)

Pre and Post Processing

- Need faster transfer speeds between machines
  - E.g., between Red Storm and Edison (according to a Presto user)
- Need improved ability to visualize large models, and at a reasonable speed
- Need better storage and archiving of computational models and results
  - Sandia's disks are filled to 85-90% capacity on average
  - Currently no official model archive location

Infrastructure improvements are needed

Usability, Reliability, Porting and Testing

- Need good reliability to decrease the need for restarting applications
- Issues with getting and keeping a problem running (beyond restart)
  - Progress monitoring
  - I/O debugging
  - File system usage
- These issues occur between
  - The application and the system
  - The analyst and the system admin
- Getting and keeping these issues right can consume system time; addressing them would improve our productivity

Usability, Reliability, Porting and Testing

- Need support for standard-as-possible operating systems, compilers, debuggers, etc. to minimize costs of porting, development, and support
  - Dependence on non-supported features in our codes like threads would be huge problems
- Need a build and test environment for nightly builds and testing
  - Could be a stand-alone, synced, machine, or simply fence off several processors for this purpose
  - This should be included in the plans and budget!
  - We currently are in desperate need for such an environment for Purple, but are lacking one!

Summary

- To achieve a *usable* Petascale environment, will need greatly improved
  - Communication / Computation ratio
  - Memory Bandwidth / Computation ratio
  - Improved Pre and Post processing and infrastructure
  - Build and test environment
- *The above won't be enough.* In combination, we need significant investments in improving our parallel programming models and understanding the impact on the numerical accuracy of the simulations

Acknowledgements

- Bob Ballance
- Ross Bartlett
- Ted Blacker
- Stefan Domino
- Ray Dukart
- Carter Edwards
- Arne Gullerud
- Terry Hinnerichs
- Mike Heroux
- Pat Notz
- Kendall Pierson
- Jim Strickland

Thanks for your input!