Part III.
Computer Science / Engineering Open issues

Chandra Kambhamettu

Open Questions: Visualization

- How to use Visualization in-line to modeling (vs. the usual offline)
- How many parameters can we visualize (perceive?) in one screen. How many do we want to?
- Auto-Summarization of massive data
- Scale-space Data Visualization

Daniel Orozco

Programming language / protocol

MPI, OpenMP, C, Fortran, and Windows/Linux/Unix are standard today for historical reasons.

However it is fundamentally HARD to correctly design and optimize programs: There are fundamental problems in the models we use today.

Dataflow, and other execution models avoid these problems at their roots, but they are rarely found in commercial applications.

Should we lose millions of lines of code (and decades of investment) and embrace new computing models, or should the struggle for programmability, performance and correctness continue forever

Can we (should we?) make a smooth transition to dataflow models?

Future MPI trend

MPI has been an enormously successful standard as a message passing layer for parallel algorithms. The goal, as always, is portability of code.

Is the current MPI standard sufficient for this project?

Is the current MPI standard sufficient for practical scientific computing on massively parallel machines?

If not, what sort of abstraction do computational scientists want or need to separate the algorithm from the hardware?

Andii Wyszogrodzki

How to deal with new Petascale technologies?

- GPU systems
- millions of cores (threads)

Solutions for scalable/efficient I/O

Methods to increase peak performance on single node

Efficient domain decomposition methods on parallel systems

Xiaoming Li

Connection to Domain Applications

- Q1: How can computer people help?
  - Coding, optimizing?
  - Kernel vs whole application?
  - How do you want students in your domain to be trained in programming and performance tuning?
- Q2: Do you care about power consumption?
  - Megawatt level consumption per supercomputer.