Breakout #1
Summary
Technical Roadmap Discussion (1/2)

- HPC Architecture Roadmap
  - What process do we use to create it?
  - What are the key components that affect software
  - What can we have done by the June meeting, & this year?
  - Who will do it?
  - Multiple paths?

- HPC Software Roadmap, Gaps, and CS research strategy
  - What process do we use to create it?
  - How should we improve software inventory? (acceptance test data?)
  - What does this community have leverage on (I/O, fault tolerance)
  - What are the key components
  - What can we have done by the June meeting & this year?
  - Who will do it?
  - How do application teams participate?
Gather and analyze existing R&D plans for addressing extreme scale

How does it match with the roadmap?

What is missing?

With software/hardware co-design, what can we address first?

What are the key new features that need the most R&D:

- E.g: power mgmt, multicore tools, math libraries, advanced memory models, etc

What is unique for our environment? (I/O, prog model)
Discussion

- Aims
  - CERN-like approach to development of an exascale system
  - vs
  - Loosely coupled research and development

- Evolution vs Revolution path?
- How broad is the base?
- Do we focus on the single node computing model?
The user-base we consider will effect the approach – if it is only the bleeding edge then many of the things we talked about yesterday are not important.

Different nations will have different focuses need to identify the specific strategic aims and how they may fit into a technical roadmap.

For some key exascale applications, flops is not the primary performance metric, eg graphical operations per sec – need to come up with appropriate metrics.
Discussion cont.

- We should develop a roadmap for the entire ecosystem of exascale leadership systems, petascale departmental, and terascale embedded to reap the broader benefits from the exascale software development.
- We need to identify the key applications to drive the technical roadmap.
- Working closely with vendors is key to success and we have the opportunity to develop hardware/software co-design.
Meta Technical Issues

- Increase in scale
- Heterogeneity
- Efficiency
- Memory
- Co-design (hardware/software)
- Intelligent Metrics
- Dynamic adaptability
Cross cutting functional issues

- Resilience (reliability & fault tolerance)
- Performance
- Programmability
- Computational model
- I/O
- Consistency & verification
- Resource Management
- Power management
Actions

- One page description of cross cutting issues
- One page on co-design – Sudip Dosanjh & Al Geist
- Development of software and activities registry – Pete Beckman & Bernd Mohr & Anne Trefethen & Mike Heroux & Jean-Yves Berthou
- How to revolutionize process used – integrating into programme – Bill Kramer & David Skinner
- End-to-End usage workflow and workload analysis- David Skinner & Alok Choudary
- Develop software taxonomy – Vivek Sarkar & Barbara Chapman & Alok Choudary
- Identify collaborator roles in technical R&D – David Skinner & Anne Trefethen
- Further down road risk analysis & external review
- Strategic decision – one track or two track approach
  - Define shareable and distinct activities - ????
Cross cutting issues

- Resilience (reliability & fault tolerance)
  - Al Geist & Bill Kramer & Nick Nystrom & Sudip Donsanjh & Franck Cappello
- Performance
  - Wolfgang Nagel & Bernd Mohr & Nick Nystrom
- Programmability
  - Vivek Sarkar & Jesus Labarta & Mitsuhisa Sato & Barbara Chapman
- Computational model
  - Thomas Sterling
- I/O
  - Yutaka Ishikawa & Bill Kramer & Alok Choudhary
Other issues

- Consistency & verification  
  - David Skinner & Bill Kramer

- Resource Management  
  - Barney MacCabe & Hugo Falter

- Power management  
  - Satoshi Matsuoka & Dan Reed