



Breakout Summary



Bill Gropp

Ron Minnich

Jarek Nieplocha

D. K. Panda

Neil Pundit

Tom Sterling

Stephen Wheat





**Bullet Two Issues:
Technical**





Major Issues



- Needs and Requirements for OS/RT Services
- How to provide support from lower layers for the required services
- Suitable measurements and metrics
- Testbeds to design, develop, experiment, and evaluate the support
- Roadmap and budget





Needs and Requirements



- Aimed at Petaflop systems in 2010
 - Custom
 - Commodity
- Large DOE applications base
 - understand their needs on the Petaflop systems
- What OS/Runtime services are (will be) critical?
 - Resource management (dynamic and adaptive)
 - Fault detection & tolerance
 - Memory hierarchy management



Support for the Required Services



- Where and How?
 - Runtime vs. OS (distinction)
- Identifying bottlenecks in runtime and OS wrt applications scaling and performance
- Defining major objectives when designing support for the services
 - Scalability
 - Reliability
 - Generality
 - Performance
 - Efficiency
- Communication and network requirements
 - Collective
 - Hardware support
- Break from Unix active process model/passive resource models
 - Active storage, memory (PIM), etc.





Suitable Measurements and Metrics



- Developing appropriate metrics to evaluate effectiveness of these schemes and support
- Total system measurement including file systems and I/O



Testbeds



- Required to
 - Design
 - Develop
 - Experiment
 - Evaluate
- Different kinds of testbeds
 - architectures, OS and Runtime
 - heterogeneous
 - Linux vs. Non-Linux
 - Custom (FPGA-based) vs. commodity
 - Different processors





Roadmap and Budget

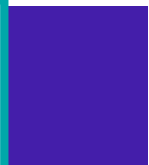


- Roadmap
 - Design and development stages
 - Adaptation to larger community
- Realistic total budget





Bullet Three Issues: Approach and Testbeds





Approach



- Are SciDAC ISICs the right model?
- Not really
 - there are no specific driving apps
 - more flexibility needed
 - do what is right rather than what requested
 - longer term perspective required
 - metrics of success will be more broadly defined
- Instead, a combination of:
 - basic research
 - applied research
 - development
 - prototypes & testbeds





Approach II



- Multi-year (10) renewable every 3 years projects
 - yearly progress reports
 - 3-year milestones
 - midterm review
 - COTS and Custom architecture tracks
 - periodic meetings to exchange ideas, explore synergies
- Start by
 - identifying hard problems
 - define long and short term research issues
 - consider
 - apps (what are they and their requirements)
 - different classes of target architectures
 - OS/RT boundaries and interactions





Phases





- Phase I (3 years)
 - concept development, models, metrics, proofs of concepts
 - design and build testbeds & simulators
 - labs and academia
- Phase II (3-5 years)
 - starts after Phase I review
 - prototype implementation (not deployment)
 - evaluation involving apps
 - vendor involvement





Testbed Assumptions and Requirements



- Components
 - h/w, s/w, networks, i/o
 - Multiple communities served
 - OS, RT developers and researchers, app experiments
 - Renewable every 2 years, refresh components more often
 - Large and extensible
 - emulate petaflop systems
 - expose scalability issues
 - Universal, remote accessibility
- 
- 



Participation and Outreach



- Partnerships between labs, academia, vendors
 - e.g., testbeds stimulate research in academia
- phase I
 - labs and academia
 - vendors welcomed but not required
- phase II
 - vendor participation encouraged

