IESP Software Break
Day 2

Oxford
## Software Efforts: Exascale Uniqueness and Criticality

<table>
<thead>
<tr>
<th>Unique-to-Exascale/Non-critical</th>
<th>Unique/Critical</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Explicit long term funding</td>
<td>• Explicit near term funding</td>
</tr>
<tr>
<td>• Explicitly managed</td>
<td>• Explicitly managed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exascale+trickle-down/Non-critical Funding? Management?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exascale+trickle-down/Critical Funding? Management?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-Unique/Non-critical</th>
<th>Non-unique/Critical</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Other party funding</td>
<td>• Other party funding with explicit funding contingency</td>
</tr>
<tr>
<td>• Moderately Tracked</td>
<td>• Explicitly tracked</td>
</tr>
</tbody>
</table>

- Funding and Management decisions are based on the criticality and uniqueness of the software efforts.
- Critical efforts require explicit funding and management, while non-critical efforts can be managed implicitly.
- Funding decisions are influenced by the nature of the funding source (internal vs. external) and the level of tracking required.
Write-up Template Elements

• For each activity:
  – Justification/rationalization of uniqueness/criticality.

• Procedure:
  – Look at current draft report.
  – Generate narratives of the results of this meeting with classification in mind.
  – Update the draft accordingly.
HW Configuration strawmen

• Need information on potential system design tracks.
• What is the best way to engaging vendor community in the codesign process with IESP?
• Suggestions:
  – Continual interaction (based on experience in embedded env).
  – Embed postdocs in vendor environment?
• Example: What are the technologies we can use a baseline for I/O capabilities?
  – Service nodes spread out on the system.
  – Local NVRAM.
• New scenarios:
  – Example: Eliminate traditional I/O storage. Use NVRAM to replace disks.
  – What are the probabilities new scenarios will emerge? High, medium, low.
• New strawmen:
  – Kogge Exascale report
  – DARPA UHPC award
  – others in future
  – Expected design freeze 2015.
Applications

• Application roadmap needed for co-design.
• How do we prioritize applications?
• What are the critical apps?
  – Without which we fail.
  • Which of these are obviously scalable (low hanging fruit)? Will run on Day 1.
  – Beyond that, how much effort to get success?
  – How soon for the rest?
  – How do we decide within multi-agency environment?
• Alternative: From science perspective what are the priorities?
  – What are low hanging fruit?
  – What are the political priorities?
• How can we incorporate performance proxies, e.g. Motifs?
Program Management

• Funders monitor whole program.
• Critical dependencies throughout program.
• Who is going to manage, coordinate and monitor:
  – Internal dependencies?
    • Especially Codesign.
  – External dependencies?
• What is the level of engagement at the international level?
• What form of funding will be ideal?
R&D Timeline

• Long term timeline: 8-12 years.
• Challenge: Typically funding comes in 2-3 chunks.
• Can we obtain long term funding?
• Staged R&D effort?
  – Years 1-4:
    • Research focus,
    • Mostly unconstrained,
    • Multiple paths.
    • Evolution: Delivering on current systems.
    • Revolution: Early prototypes and demonstrations for future system.
  – Years 5-8:
    • Development focus,
    • Delivery oriented.
    • Start R&D for post-Exascale (pipeline).
  – Year 9-12:
    • Deployment and improvement.
Co-design Methodologies

• Exchange of people, co-location:
  – Example: Postdocs at providers. Providers at labs.

• Simulation, emulators, small scale prototypes:
  – Systematically part of co-design.
  – Each group: Determine what it can add to this process.
  – All parties need to make strong commitment, co-owned deliverables.

• Co-design should be explicitly planned, funded and managed.

• Co-design in HPC:
  – Need co-design approaches that support vastly complex effort (when compared to embedded approaches).
  – Need to understand the constraints on the design space.
  – Need to understand risks of co-design:
    • Example: Provider departing market.
Additional Topics Discussed
Exascale SW vs Commodity Node SW

• Much discussion about relationship between exascale needs/development and commodity needs/development.

• Can we factor efforts to depend on commodity?
Questions for Apps

Answer for short, medium and long term

• What kind of I/O will be needed by Apps?
• Will apps manage power and need an API?
New Programming Model Features

• Resilience model (with library extensions?)
• New Model of Computation?
  – Need to be explicit about roadblocks.
  – How will new model overcome them.
Risk vs. Likelihood

• Software Roadmap
• Deadlines
• Useful for our efforts?