Building the Exascale Software Center

The Exascale Software Center Planning Team
Presentation at the IESP meeting, October 2010

Pete Beckman
Director, Exascale Technology and Computing Institute (ETCI)
Argonne National Laboratory

Jack Dongarra, University of Tennessee, Oak Ridge National Laboratory

+ Large team of folks from national laboratories and universities
# Potential System Architecture Targets

<table>
<thead>
<tr>
<th>System attributes</th>
<th>2010</th>
<th>“2015”</th>
<th>“2018”</th>
</tr>
</thead>
<tbody>
<tr>
<td>System peak</td>
<td>2 Peta</td>
<td>200 Petaflop/sec</td>
<td>1 Exaflop/sec</td>
</tr>
<tr>
<td>Power</td>
<td>6 MW</td>
<td>15 MW</td>
<td>20 MW</td>
</tr>
<tr>
<td>System memory</td>
<td>0.3 PB</td>
<td>5 PB</td>
<td>32-64 PB</td>
</tr>
<tr>
<td>Node performance</td>
<td>125 GF</td>
<td>0.5 TF</td>
<td>7 TF</td>
</tr>
<tr>
<td>Node memory BW</td>
<td>25 GB/s</td>
<td>0.1 TB/sec</td>
<td>1 TB/sec</td>
</tr>
<tr>
<td>Node concurrency</td>
<td>12</td>
<td>O(100)</td>
<td>O(1,000)</td>
</tr>
<tr>
<td>System size (nodes)</td>
<td>18,700</td>
<td>50,000</td>
<td>5,000</td>
</tr>
<tr>
<td>Total Node Interconnect BW</td>
<td>1.5 GB/s</td>
<td>20 GB/sec</td>
<td></td>
</tr>
<tr>
<td>MTTI</td>
<td>days</td>
<td>O(1 day)</td>
<td></td>
</tr>
</tbody>
</table>
Biggest Disruption:
Node Architecture is Changing

- 100x – 1000x more cores
- Heterogeneous cores
- New programming model
- 3d stacked memory
- Smart memory management
- Integration on package
Context: Planning for Exascale

Platforms
• Systems: 2015
• Systems: 2018

Cross-cutting Technologies

Co-Design Application Teams

Exascale Software

Goal: Ensure successful deployment of coordinated exascale software stack on Exascale Initiative platforms
Exascale Software Center
within co-design framework

**Ultimately responsible for success of software:**

- Identify required software capabilities
- Identify gaps
- Design and develop open-source software components
  - Both: evolve existing components, develop new ones
  - Includes maintainability, support, verification
- Ensure functionality, stability, and performance
- Collaborate with platform vendors to integrate software
- Coordinate outreach to the broader open source
- Track development progress and milestones
Exascale Software Center  (in 1 slide)

- **Scope**
  - Deliver high quality system software for exascale platforms
    - ~2015, ~2018
  - Identify software gaps, research & develop solutions, test and support deployment
  - Increase the productivity and capability and reduce the risk of exascale deployments

- **Cost:**
  - Applied R&D: ~10-20 distributed teams of 3 to 7 people each
  - Large, primarily centralized QA, integration, and verification center

- **Schedule Overview**
  - 2010 – Q1 2011: Planning and technical reviews
  - April 2011: Launch Exascale Software Center!
  - 2014, 2017: SW ready for integration for 2015, 2018 systems respectively
Assumptions

- Several vendor platform partnerships
- ~2015 early scalability demonstration systems
  - Arch 2010-2011; System build 2015
- ~2018 exascale system
  - Arch 2014-2015; System build 2018

- Co-design centers provide initial applications

- ESC:
  - Partnership funding agencies, labs, and universities
  - Responsible for the common software environment for El systems
  - All development will be open source, with BSD-style license preferred over GPL
  - Some components will be integrated and supported by vendor, others will be provided atop basic platform, supported by ESC
  - Vendor-specific components will be part of their platform strategy
    - E.g.: system management, RAS, compiler, etc
ESC Organization Chart

Advisory Committee
Vendor Partnerships
International Partners
Co-Design Council Chair:

ESC Management
Director:
Deputy:

DOE Program HQ

Exascale Joint Management Council

Co-Design

Software Components

Programming Models
Operating Systems and Run Time
Application Programmer Tools
Numerical Libraries and Frameworks
Data Management and Analysis
System Management and Cybersecurity
Testing, QA, HPC Center Integration, Support
Vendor Integration Team
Application Integration and Performance
The Exascale Software Center and Co-Design Processes

- Platform Architects (vendors)
- R&D Software Community
- Co-Design Centers

- Identify Needs
- Identify Gaps
- Research
- Initial Prototypes
- Test & QA
- Integration, Deployment, Support

Initial System Design

Refining Design

Applied Research and Development
**Vendor Co-Design Model**

- Want something like ESC to coordinate and take *real* responsibility for features and milestones
  - Improved leverage over projects that are currently less responsive than needed
- Do not want “toss over the wall” strategy. “hardening” cannot be done by different team
- Need to manage risk of final machine functionality, performance, stability and acceptance
- Key ESC models:
  - ESC developed -- vendor integrated and supported
    - Two test and development environments needed, with careful planning, linking, and tracking of known issues
  - ESC developed – ESC provided, and supported
- Formalized roles between ESC and Vendors for development, risk, support, and acceptance
- Feedback and progress tracking between ESC and vendors must be shared
- Application co-design centers should coordinate discussions of system software through ESC
- NDA material for roadmaps, across co-design centers, etc will be difficult to coordinate

<table>
<thead>
<tr>
<th>Examples of software package</th>
<th>Primary developer</th>
<th>First-level Support Provider</th>
<th>Second-level Support Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 RAS, system mgmt, compilers</td>
<td>Vendor</td>
<td>Vendor</td>
<td>Vendor</td>
</tr>
<tr>
<td>2 OS, MPI, PAPI, math libraries</td>
<td>ESC</td>
<td>Vendor</td>
<td>ESC</td>
</tr>
<tr>
<td>3 Performance tools, I/O libraries</td>
<td>ESC</td>
<td>ESC</td>
<td>ESC</td>
</tr>
<tr>
<td>4 Perl, Python</td>
<td>Broader Community</td>
<td>Vendor</td>
<td></td>
</tr>
<tr>
<td>5 Eclipse IDE</td>
<td>Broader Community</td>
<td>Broader Community</td>
<td>Broader Community</td>
</tr>
</tbody>
</table>
Application Co-Design Model

- Want something like ESC to coordinate and take *real* responsibility for features and milestones
  - Improved leverage over projects that are currently less responsive than needed
- Want to know specifics about hardware and available software
- Applications will provide best estimates of needs for exascale science:
  - Data movement, memory sizes, programming models, etc
- Applications will test and evaluate prototype system software
- Need help managing risk of final machine functionality, performance, stability and acceptance
- Formalized roles between ESC and App Co-Design Centers for development, risk, support, and acceptance
- Feedback and progress tracking between ESC and App Co-Design Centers
- Coordinate discussions of system software through ESC
- NDA material for roadmaps, across co-design centers, etc will be difficult to coordinate
International Co-Design?

- Tomorrow’s breakout
- Europe --- Asia --- US
Selecting ESC Components
Making the hard choices

- **ESC is responsible for delivering successful software**
  - Technical evaluation:
    - Criticality to successful deployment and key applications
    - Technical risk for achieving goal
  - Project team evaluation:
    - Team history of delivering high-quality, applied software
    - Management and institutional support

- ESC will make component selection and resource decisions based on criticality and risk
  - continuous evaluations of progress; adjust resources

ESC will have a range of components

---

### Technical Evaluation Matrix

<table>
<thead>
<tr>
<th></th>
<th>Low Risk</th>
<th>Moderate Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESC Supported</td>
<td>Important</td>
<td></td>
</tr>
<tr>
<td>Vendor Supported</td>
<td>Critical</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Most Critical</td>
<td></td>
</tr>
</tbody>
</table>
ESC Software Development

- Successful applied R&D teams are built around clear goal of delivering working, supported packages
- Good software hygiene can’t be someone else’s job
- ESC must **work with successful teams existing processes** or in some cases, boot new teams within institutions with excellent history of deployed software
  - Probably not feasible to launch new team at site without history of software success
- Formal plans and milestones and reviews are necessary for each component
- Co-design feedback and risk-based assessments work well with spiral development discipline for software (common in R&D)

Classic “Waterfall” model  “Spiral” model
Required Processes for ESC Components

- Formulation of clear deliverables with specific targets for functionality, performance, and stability
- Defined team management plan and risk tracking
- Documented software development plans
  - QA (unit tests, integration, etc)
  - Performance testing
  - Documentation, support
  - Bug and new feature tracking
- Resource accounting
- Technical review schedule
- Release schedule
- Integration plan
Distributed Project Staffing Approach

- “ESC Component Teams” should be located where their center of mass has demonstrated success
  - E.g: Math libraries at UTK, Performance tools at UOregon and Rice, etc.

- Each Component Team will have at least one “embedded” QA and testing staff member provided by ESC
  - Position will be held by professional QA/build engineer (i.e., not a student or postdoc)
  - Candidates will be approved by ESC director of QA and have performance appraisal “matrix input”

- Each of the 4 sites (2 NNSA, 2 SC) must have local ESC team members responsible for integration
  - Will belong to production computing division, not R&D division

- QA, integration, and support team will be primarily at one site

- Resources dedicated to collaboration and software development infrastructure is required
Community Engagement

IESP Activities

European Exascale Initiative

Japanese Exascale Initiative

ASCR/NNSA Institutes

Domain Science Institutes

Computer Science Institutes

ESC

Co-design Centers

Applications Co-design

Hardware Co-design

Third Party Software
## ESC High-Level Milestones (under development)

<table>
<thead>
<tr>
<th>Computer</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018</td>
<td>Launch the Exascale Initiative and start the Exascale Software Center.</td>
<td>2Q11</td>
</tr>
<tr>
<td>2015</td>
<td>Report on 100PF needs, identified gaps, and researched solutions</td>
<td>4Q11</td>
</tr>
<tr>
<td>2015</td>
<td>Vendor agreements on integration roles and timelines complete</td>
<td>2Q12</td>
</tr>
<tr>
<td>2015</td>
<td>Deployable packages created in all five target areas</td>
<td>3Q13</td>
</tr>
<tr>
<td>2015</td>
<td>QA and Support infrastructure to create production quality developed</td>
<td>3Q13</td>
</tr>
<tr>
<td>2015</td>
<td>Initial version of software on prototype hardware complete</td>
<td>2Q14</td>
</tr>
<tr>
<td>2015</td>
<td>Integration of packages, testing, and enhancement of SW stack complete</td>
<td>3Q15</td>
</tr>
<tr>
<td>2015</td>
<td>Integrated SW stack deployed on 100+PF systems</td>
<td>4Q15</td>
</tr>
<tr>
<td>2018</td>
<td>Report on 1EF needs, identified gaps, and researched solutions</td>
<td>2Q15</td>
</tr>
<tr>
<td>2018</td>
<td>New target areas and teams incorporated into ESC based on assessment</td>
<td>4Q15</td>
</tr>
<tr>
<td>2018</td>
<td>Deployable packages in all targeted Exascale areas created</td>
<td>3Q16</td>
</tr>
<tr>
<td>2018</td>
<td>QA and support infrastructure for new areas developed</td>
<td>3Q16</td>
</tr>
<tr>
<td>2018</td>
<td>Initial version of software on prototype hardware complete</td>
<td>2Q17</td>
</tr>
<tr>
<td>2018</td>
<td>Integration of packages, testing, and enhancement of SW stack complete</td>
<td>3Q18</td>
</tr>
<tr>
<td>2018</td>
<td>Integrated SW stack deployed on 1EF systems</td>
<td>4Q18</td>
</tr>
</tbody>
</table>
Next Steps

- Develop software planning documents:
  - Definition of review materials
  - Formal review in April 2011
- Build application co-design liaisons, develop plan for jointly evaluating key software
- Build links to IESP organizational plan
- Begin technical evaluation and ranking of key software components
- Link to NSF, NASA, DARPA, and other groups