IESP meetings summarized

Peter Michielse (NWO/NCF)
April 2011

• Based on EESI living deliverable D2.3
• With Patrick Aerts
Contents

- Overview meetings
- Attendee statistics
- Initial objectives
- Achievements so far
- Topics covered
- Evolution of these topics over the meetings
Overview Meetings

- Meetings so far:
  - Santa Fe, NM, US, April 2009
  - Paris, France, June 2009
  - Tsukuba, Japan, October 2009
  - Oxford, UK, April 2010
  - Maui, HI, US, October 2010

- SC08 (Austin), SC09 (Portland) and SC10 (New Orleans)

- Overall goal:
  - To develop a plan for producing a new software infrastructure capable of supporting exascale applications
Attendee Statistics

- Attendees from universities, research institutes, government, funding agencies, research councils, hardware and software vendors, industry

- Amount of attendees roughly between 65 and 85

- Rough distribution per meeting is rather constant:
  - 70% universities/research institutes
  - 15% vendors/industry
  - 15% government/funding agencies
Initial Objectives

- To develop a plan for producing a new software infrastructure capable of supporting exascale applications
  - Thorough assessment of needs, issues and strategies
  - Develop a coordinated software roadmap
  - Provide a framework for organizing the software research community
  - Engage and coordinate with the vendor community
  - Encourage and facilitate collaboration in education and training

NCF - Peter Michielse - IESP San Francisco - april 2011
IESP has created version 1.0 and 1.1 of a software roadmap:

- Science and technology trends
- Software stack components (with cross-cutting aspects)
  - Not only the usual suspects (MPI, compilers, OpenMP, ...)
  - But also new challenges, like resiliency, bit-wise reproducibility, ...
- Application involvement
  - Business cases – why do we need exascale-ready applications?
  - Co-design vehicles
- Vendor involvement
- Organisation and governance
IESP has created awareness:

- We all find it exciting
  - But does the science community have the right sense-of-urgency?
- We are getting to some kind of community building
  - Reports by IDC, HPCwire, ...
- It is on the radar of funding agencies and governments
  - DoE, NSF, Japan, EC programs, ...
  - But maintenance is needed
Evolution

- Many important topics have been discussed and brought forward during the IESP workshops
- We will go through these, show their evolution and current position:
  - Science drivers
  - Applications
  - Software stack
  - Open source
  - Vendor involvement
  - Hardware
  - Co-design
  - Exascale software centers
  - Initiatives
  - Funding and governance models
Science drivers

- Input through many activities, among others:
  - DoE workshops in US
  - Science and Technology Basic Plan in Japan
  - Science case for PRACE in Europe

- Many areas of scientific interest and societal impact:
  - Resolution increase
  - Addition of complexity
  - Multiscale simulation

- These aspects, combined with hardware futures, pose requirements to applications and underlying software stack
Applications - 1

- Current findings:
  - Bulk synchronous, MPI, models decomposed in some way
  - But also global shared memory, global arrays, ...
  - Hybrid MPI/OpenMP already in place
  - Load-balancing challenges
  - Weak scaling to Petaflop level ok

- Coming from petascale, what are we facing towards exascale?
  - Concurrency – 1000x increase for a single job
  - Energy usage (also in relation to data movement)
  - Resiliency – how to handle device errors
  - Heterogeneity – combine standard CPUs with GPUs and accelerators
  - I/O and memory – data rates
Applications - 2

- Higher-level questions:
  - Programming models
  - Coherence domain
  - Fault tolerance
  - Reproducibility of floating-point results
  - Big data

- Applications rely on software stack
Software stack (X-stack)

- The exascale roadmap addresses software components:
  - Each subdivided in capabilities
  - Each with a level of uniqueness for exascale, and a level of criticality

- Components:
  - Frameworks
  - Algorithms
  - I/O
  - Programming models
  - Operating Systems
  - Power
  - Resilience
  - Numerical libraries
  - Debugging
  - Scientific data management
  - Compilers
  - Performance
  - Programmability
  - Runtime systems

- Inventory of what is in use at computer centres (done later today by Bernd Mohr)

- We all agree on joint development, but what will be the model?
Open Source

- Keep in mind that hw and sw vendors have been developing their own compilers, numerical libraries, runtime environments, MPI, ...
- We seem to agree we need co-development (joint ownership and responsibility with a formal agreement)
  - Academic + vendor
- Would that be: just Open Source, or Open Source + (e.g. with paid support contract)
- Probably a minimum requirement: open API’s published and supported
- Probably the toughest area for the vendors:
  - Developed by community, and supported by provider
Vendor involvement

- Vendors like to embark on the exascale train:
  - Systems contain huge amounts of devices
  - Share research and development effort
  - But still there is the business model: do the investments by vendors pay off in lower-than-exascale systems?
    - And also: can customers afford the running costs?
- Many aspects on software approach:
  - Support and maintenance by vendor in case of Open Source software components?
  - Responsible for components which are not under vendor’s control?
  - What about low-level proprietary software components?
  - IP questions
Hardware

- Basic core design in place
- Trade-offs at node level
  - Heavy vs. lightweight cores, how many, data movement, threads per coherence domain
- Hardware features for resilient algorithms?
- Memory hierarchy levels?
- Power management tools?
- Programming models?
- General observation:
  - Many questions in each area (hardware, software, apps) which require answers, in a collaborative effort
  - Have been addressed in IESP through break-out groups
Co-design

- All tough questions posed so far require answers from more than one point of view: co-design

- Co-design of computer system is a design process in which:
  - Scientific problem requirements
  - Architecture design
  - Technology developments
  - Software components and capabilities are considered together

- Requires integrated teams of scientific researchers, applied mathematicians, computer scientists and computer architects

- Co-design centers around each important scientific discipline
Exascale software centers

- US (DoE funded) planning effort for an Exascale Software Center (ESC), with the following initial 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;

- Organisation:
  - Applied R&D: ~10-20 distributed teams of 3 to 7 people each;
  - Large, primarily centralized Quality Assurance, integration, and verification center;
  - Relation to co-design effort

- Schedule Overview (as of Oct. 2010):
  - 2010 – Q1 2011: Planning and technical reviews
  - April 2011: Launch
  - 2014, 2017: SW ready for integration for 2015, 2018 systems respectively
Initiatives

- **US**
  - Exascale Software Center planning effort
  - Co-design Centers proposal evaluation (21)
  - Bring the next Petaflop/s systems online

- **Japan**
  - Strategic program to promote HPC activities
  - Next-generation supercomputer project (Kei)

- **China:**
  - Ambitious schedule towards 100’s Petaflop/s and Exaflop/s
  - Own hardware?

- **Russia:**
  - Presidential agenda
  - Vendor presence (T-platforms)

- **Europe:**
  - FP7 calls for computing systems, exascale initiatives, under evaluation
  - EESI
  - National Petaflop/s systems in PRACE

- **G8 calls**
Governance and funding

- Some basic ideas for governance:
  - At least coordination is needed
  - International software roadmap group to overview for planning and definition of components/capabilities
  - Funding agencies to coordinate their support of IESP-related R&D so that they complement each other
  - International monitoring team
  - Testing facilities to be organised

- First timeline discussed:
In summary

- **Current status/approach in IESP:**
  - Maintain the science drivers side
  - Investigate software stack and prioritise
  - Co-design hardware, software, applications
    - Set up co-design centers per application area
  - Find a proper model to include vendor participation
    - OpenSource, Support, Responsibility, IP, proprietary components, APIs
  - Applications, hardware, software:
    - Answer each others’ questions!
  - Find a proper funding and organisational model
    - International software roadmap team
    - International role of Exascale Software Center(s)
    - Make sure all initiatives align