HPC Software Layers
Breakout Session 4

System Software: The power within that you can't do without

smkelly@sandia.gov

scanon@lbl.gov

Session 4 Participants

Shane Canon , LBNL, Session Co-Chair
Sue Kelly, SNL, Session Co-Chair

Participants

Jon Stearley SNL
Robert Bohn NCO/NITRD
Krystyne Supplee Cray Inc.
Tinu Ogunde ASC/NNSA/DOE
Gary Grider LANL
Paul Iwanchuk LANL
Buddy Bland ORNL
Andrew Uselton NERSC
Janet Lebens Cray Inc.
William Spotz DOE
Manojkumar Krishnan PNNL
Bob Ciotti NASA NAS Division
Daniel Hitchcock DOE - Office of Science
Patricia Kovatch NICS/UTK
Vicky White ORNL
Susan Turnbull DOE- ASCR
Mark Gary LLNL
Howard Walter LBNL/NERSC
Jeff Broughton NERSC/LBL
Stephen Scott ORNL
Pam Hamilton LLNL
William Kramer NCSA/University of Illinois

Desire to hear from everyone

Get acquainted questions

- Name, organization, your role
- Is your site purely a consumer or does your organization play a role in supporting and maintaining system software?
  - We buy it and pay for support
  - we downloaded it and rely on a community
  - we download it and participate in the community support
  - we develop it and support it
- Do you have concerns with the support and longevity of any critical system software?

Actions for this session

- Identify best practices that are unique to System Software development, engineering, and sustainability

After Lunch:

- Address cross-cutting questions (1 hour)
- Create 1-3 survey questions that will be responded to by all workshop attendees (.5 hours)

System Software Best Practices

- Are there any Certified Software Development Professionals (CSDP) in attendance? In anyone in your group back home? Is it relevant?
- How does or doesn't the SWEBOK apply to HPC system software?
- Identify and provide an assessment of your
  - Life cycle methodology (Waterfall, Iterative, Agile (e.g. SCRUM), code and fix, etc.)
  - Bug tracking system
  - Distribution/release mechanism
    - What is an optimal release schedule for system software?
  - Testing engine or process
- How can Centers help sustain and improve commercially and community developed software?
- What system software do you rely on for your operation and how critical are they to your mission?
  - (Examples, Linux distribution, File Systems, Lightweight kernels, hyper-visors, InfiniBand stacks)

Critical Issues in Sys SW

- Consortium dynamics
- Cost of ownership/ support
- testing/integration
- scalability
- resilience
- sustainability
- power management
- Strive for commonality
  - strengths in numbers
- too small a community; pie is shrinking
- Funding is at micro scale, so collaboration is equally small
- Inability to get patches into mainstream (e.g.Linux)
  - buy (fund) influential developers
- Best practices
  - Path Forward
    - Totalview
    - Lustre
    - pNFS
    - HPSS
  - Open source
    - MPI
- How to let others know what we're funding
- Funded projects need to be tried on big machines

Cross-cutting Questions

- What are the best practices and tools?
  - Inside HPC
  - Outside HPC
  - Education and Training
- What are the top challenges?
- What new technologies are needed?

Workshop-wide Survey Questions

- Suggestion: What is the preferred support model in your organization for these components?

	Vendor supported	Open Source Consumer	Open Source contributor	Develop our own	Something else (specify)
Operating System					
Serial file system					
Parallel file system					
Network stack					

Breakout Report

Challenges

- **Funding**
  - **In too small of increments to have an impact and to allow collaboration**
  - **for maintenance and support**
- **Scalability**
- **Resiliency**
- **Complexity**
  - power management
  - Interoperability of components? Need to take everything down to change 1 component
- What is the right balance between the above? What is proper frequency of course correction?
- Gathering requirements
- What is the right licensing model for system software?
- Outside HPC has a slightly different language

Best Practices in Sys SW

- Policy for sustainability
- **Forming relationships with vendors, open source providers through contracts, testbeds (Hyperion), active participation in the community**
- **Testing at scale**
  - **community test environment open to all**
  - **work with vendors to test at scale**
    - **(This was missing from PathForward)**
  - **use decommissioned large scale?**
  - **Access to test systems for specialized platforms**
- Successes
  - Structured Collaborations (e.g. HPSS, Open Fabrics)
  - **PathForward (e.g. Lustre)**
  - **ASC Alliances with active engagement, including funding, of oversight group**
- More commonality to help address funding issue (e.g. TLCC)
- Engage with non-HPC that have overlapping requirements
- **Developers gets funds cycled back in for sustainment**

Required Technologies

- System software suitable for hetero computing
- Exploiting virtualization in HPC
- Machine diagnostics (self organizing and self diagnosis)
- System software to provide other sw layers information to be fault tolerant
- Power aware software

Proposed Findings

- Need for Alliance-style or PathForward-style initiatives
  - level and scope of the problems and commensurate funding
- Current software stack breakdown won't sustain us to exascale. There are too many cross cutting issues (e.g. resiliency and scaling)
- Access to specialized test systems and large scale systems are critical.