SAND2011-0681P

System Software (SSW) Report from ASC/CSSE-FOUS Exascale Planning Workshop, Washington DC, June 15-16, 2010 Outbrief Questions

Ron Minnich

Bob Ballance, Ron Brightwell, Chris Dunlap, Jim Garlick, Maya Gokhale, Pam Hamilton, Sue Kelly, Mike Lang, Scott Pakin

1) SSW Scope

System software encompasses:

kernel, communication libraries, job scheduler, systems health and systems monitoring

(out of scope: compilers, math libs)

- Dependencies
  - Architecture
- Close collaboration needed with
  - Programming models
  - I/O
  - Tools
  - Data/Viz

2) SSW Current State of the Art

- Large-scale Systems (petaflop class)
  - Cray's XT6, IBM's BG/P, IBM/LANL Roadrunner
  - Current SSW
    - Linux on 2 out of 3 – and arguably on all 3
    - Lightweight Kernel on XT6 and BG/P
    - Plan 9 on BG/P
  - What works
    - Most features are borderline at Petascale, ***not*** likely to work for exascale
    - Statically allocated resources work but not reliable at petascale
  - What doesn't
    - Reliability/resilience
    - Dynamic resources
    - ? I/O – filesystems ?
    - Memory management
    - Power management
      - Hardware features may be offline for power reasons
    - Posix/Pthreads API needs to go
    - Heterogeneity everywhere, processors, network

3) SSW Exascale Needs

System software that manages millions of nodes is fundamentally different from SSW that manages thousands of nodes

- Need tighter tie-in with programming models
- Standard Linux will not be a good fit at exascale
- Global/Regional Power management
- Better access to hardware info
- Reliability/resilience
  - Good HW notification
  - Run-time and application need to expect and react to failures
- Dynamic management of resources
- I/O – filesystems ?
- Memory management
  - Hardware features may be offline for power reasons
  - 3D Memory
    - A defined access method; cache, app managed, coherent?
- Posix/Pthreads API needs to go, not a good fit for exascale
- Heterogeneity everywhere, processors, network

NNSA
National Nuclear Security Administration

4) SSW Technology Gaps

- Research Needs (in house)
  - Scalability (all)
  - Resilience (all)
    - SSW, runtime, apps all assume checkpoint for solving resilience
    - How do we build systems that survive in presence of constant failure, without using checkpoints
  - Power (Sandia, LLNL)
    - Runtimes/applications could be a lot smarter but the information is frequently not available
    - Requires a change in thinking about power management
  - Direct research in new kernel models beyond current lightweight kernels and Linux (Sandia, LANL)
  - Direct research in runtime systems, task schedulers, communications layers, i.e. software that ties into programming models
    - Scalable Services (LLNL, LANL)
    - Communication Layers (Sandia, LANL)
  - File system virtual layer (LANL, LLNL)
  - Strong co-design impact on vendors and platforms
- Research Needs (academia)
  - Data Movement API in collaboration with in house researchers (Institution TBD)
- Development Needs (in house)
  - APIs for power, resilience, 3D Memory (all)
- Development Needs (FastForward)
  - Architectural controls for power (AMD, Intel, other processor and chipset vendors)
  - 3D Memory controls (US supplier)
  - Management and control systems (Vendors supporting ConMan, Cray, ?)
  - Failure management support (Chip, network, I/O vendors)

5) Proposed technology to address gaps with time phases

Technology	FY11-15 Simulation/codesign	FY16-20 Move to real systems	FY21-25
APIs	Data Movement, resilience, power, Post-posix file system interface		
Kernels	Evolutionary (Linux++?) and revolutionary framework	-Scheduler, with new philosophy on what's valuable (it's not the CPU anymore) - Memory manager	New VFS layer, new security model
Run time systems	Resiliency	Dynamic node and other resource allocation, Heterogeneity, support new programming models	"Fix all the stuff we got wrong" a.k.a. refactor
Data Movement	Evolutionary – MPI3 – and revolutionary – resilient data movement framework	Testing on real systems	
Control Systems	Resilient, distributed, self-healing, autonomous, power-aware		

6) SSW Co-design and Cross-cutting issues

- Tools and other needs to facilitate co-design
  - Need good simulators and early access to test bed for new and proposed architecture designs
  - Virtualization to explore new hardware and scalability
  - Mini applications and/or synthetic workload
- Interface and interdependencies addressed
  - Need well defined requirements from programming models and architecture
  - Need to provide interfaces for tools and programming models
  - Need APIs for data movement, resilience, power, and control systems
  - Need well defined requirements for visualization and analysis
- Interface and interdependencies NOT addressed
  - ?

7) SSW Partnerships

- Technical Coordination
  - DOE
    - ASCR funded OS research
  - Industry examples
    - FAST-OS research on BG/P (IBM)
    - Cray Linux Environment (Cray)
    - New OS work from Bell Labs
  - Academia examples
    - Boston University research on very large scale virtualization on Blue Gene (funded on ASCR X-stack)
    - Tesselation group at Berkeley
    - “Self-healing OS” at MIT
- Co-Funding Coordination
  - DARPA – OHPC, UHPC
  - OFFICE OF SCIENCE – X-STACK
- Academic Fellowships
  - Place professors/students at National Labs

U.S. DEPARTMENT OF ENERGY

Set of questions

- DAM questions
- Codesign (what does it mean for SSW?)
- PM
- How does SSW API change for IO?
  - How do we incorporate incoherent storage
  - Data base model – how do we use it – SSW change
  - If data base scales to exascale – do we use that for operation of HPC

National Nuclear Security Administration

What services does DAM need

- Does app see a fault or just runtime? (ASCI RED experience)
- "lost rank" example – what fraction can we lose before it is game over (and what do we do)
- How to deal with "approximate" resource specs that change over time (min/desired/max/hotspare) (memory,cpus,IO,power,mem hierarchy)
- PGAS?
- X10? Fortress? Chapel? Any others?

U.S. DEPARTMENT OF ENERGY

- It won't be just two levels
- It will be 4 or more
  - Rich cost model (time, power)
- Will apply to all resources
  - Memory, file system, messages, etc.
- How much should be visible

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY

DAM questions (cont)

- Do you want a “virtual perfect machine?”
  - How much would you pay? (you pay 20% for SECDED)
  - All run as slow as worst case (HP Superdome)
- What would you consider reasonable
- And what would you abstract if so
- How important are BSP/collectives
  - Can we do async collectives/domains/

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY

DAM

- Are your needs a fundamental property of the algorithm or the implementation?
- Example: move to cell-based algorithms
  - How might SSW change to support cell-based algorithms? (not IBM Cell)
  - Convergence/data integrity/recovery from node disappearing (“punch a hole in the grid”)
- How configurable software stack?
  - Your own kernel/message primitives/

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY

Hardware

- TM?
- Power knobs (besides ACPI)
- Local persistent storage (Big disk? Big FLASH?) How do we use it/ Security issues?
- Is it realistic to have “perfect networks” a la BG?
  - And if so how would that be done
- Cache coherence between cores?
- How is core failure/restart handled?
- How much future is there for x86?

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY

Hardware

- How much hardware support for resilience?
  - Should we look at cloud on a chip
- What if a core, e.g. wedges the bus – what do we need to do?
- Given that we have no influence
  - How do we get influence?
- If we have tens of thousands of threads per core how would future hardware support it

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY

Hardware

- What does 3D really look like?
  - Super cache? Software managed? L4?
  - SRAM? DRAM? Hybrid?
- How much can we influence the module/board/system level?
- What do interconnects look like
  - Flat/hierarchy/PGAS/free space optics/
- Does IO/CPU node differentiation continue?
  - Can IO nodes get more memory?

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY

Programming models

- Review DAM questions

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY

Programming models

- Does app see a fault or just runtime? (ASCI RED experience)
- "lost rank" example – what fraction can we lose before it is game over (and what do we do)
- How to deal with "approximate" resource specs that change over time (min/desired/max/hotspare) (memory,cpus,IO,power,mem hierarchy)
- PGAS?
- X10? Fortress? Chapel? Any others?
- Does PM subsume SSW?

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY

- It won't be just two levels
- It will be 4 or more
  - Rich cost model (time, power)
- Will apply to all resources
  - Memory, file system, messages, etc.
- How much should be visible
- What services does PM need to manage this hierarchy?

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY

PM questions (cont)

- Do you want to provide a “virtual perfect machine?”
  - What do you need from SSW in that case
- And what would you abstract if so
- How important are BSP/collectives
  - Can we do async collectives/domains/

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY

More PM

- How configurable software stack?
  - Your own kernel/message primitives/
- What functions might move from PM to SSW
- What functions might move from SSW to PM
- Does OS bypass make sense in 100M CPU world? If not, what?

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY

Tools

- What is needed for debugging?
- How do we monitor 100M CPUs
- What would you like to see from SSW?

National Nuclear Security Administration

System Software (SSW) Report from ASC/CSSE-FOUS Exascale Planning Workshop, Albuquerque September 14-16, 2010 Outbrief

Ron Minnich

Bob Ballance, Chris Dunlap, Jim Garlick, Maya Gokhale, Pam Hamilton, Sue Kelly, Mike Lang

National Nuclear Security Administration

PEM key points

- Hardware events must NOT be hidden, must be available to the SSW, and it MUST be possible to drive those errors to the very highest levels in the application.
- Quality of service for IO would be helpful.
- SSW must provide
  - High performance and efficient thread management (M;N or N:N?)
  - Richer set of communication alternatives (typed channels)

PEM key points

- Performance need not be predictable, as long as the app can measure it and react
  - Information should be preserved as it goes up software levels, i.e. information hiding is “bad”
  - This is counter to a common OS (Linux) design pattern
- Don’t see an end to use of bulk synchronous parallel programming

U.S. DEPARTMENT OF ENERGY

PEM key points

- Interrupt-based vs. Polling-based notification
  - Polling is preferred for non-fatal events
  - Interrupts still have a role

National Nuclear Security Administration

PEM key points

- Collective power management
  - More hysterisis (i.e. don't change it 2x/second)
- Uses for NVRAM
  - Defensive I/O
  - Extension of DRAM
  - Inter job scratch space
  - Debug, performance, staging data for debug
- Willing to invest up to 10% of system resources for resilience

IC key points

- Some apps expect to take on the challenge of dealing with failure
- Jobs can specify min/desired/max i.e. range of resource fluctuation
  - Frequency-dependent (see:PEM hysteresis note)
  - Every couple hours is doable today (via checkpoint)
  - SSW should force exit if a required resource drops below min

IC key points

- Programming models present an ideal machine
  - Apps have to implement “programming model bypass”
- Depending on the type of failure
  - Want the information at the app level
  - Want the underlying infrastructure to solve the problem, this may impact performance so the app still wants notification

V&V key points

- Python is the glue that holds the runs together
  - Sets up the runs
  - Initiates them as an ensemble
  - Handles post processing
- Python on “beefier” node could steer 1000 “compute nodes”
  - Hence we need more heterogeneity than we get with BG/P and XT
  - E.g. some fraction of CPU nodes with big memory

V&V key points

- Macro level exascale simulator seen as useful
  - Near term task for SSW could be to develop such a simulator to run on HPC resource
    - Forces further development of SSW
  - Simulator can be used be used to try out design alternatives e.g. NVRAM and how it fits in
  - Note: we need to make trilab HPC systems available for SSW research (as ANL does today)
  - Counting on DST is not an answer

U.S. DEPARTMENT OF ENERGY

V&V key points

- Residency between multiple app runs

National Nuclear Security Administration

HW/Tools key points

- SSW must be able to take cores offline
  - Bringing them back online would be nice
- App can “Park” cores for power savings and BW saving without long system call
  - Dynamic – very dynamic – in a loop
- SSW needs to assume that core offline/online transition will be quick (hardware changes)
- Fine grain power control of core subsystems

HW/Tools key points

- Re-examine hardware privilege boundary
  - i.e. user-level halt of a core
- 3D: another level in the memory hierarchy
  - Some NVRAM, some not, may be user-managed fast memory subset
- Semantics of NVRAM are not quite memory, but we would not like them to be a disk

U.S. DEPARTMENT OF ENERGY

HW/Tools key points

- Tools would like as much info as possible
  - Both polling and interrupts
- Need to provide more info with less overhead
  - That might have been considered privileged
  - Information on resources in use, PTEs, etc.
  - Health/status memory segment
- Taxonomy of errors needed

National Nuclear Security Administration

HW/Tools key points

- (software) Broadcast/reduction overlay networks for efficient data collection and distribution
- There are two levels of OS
  - On node
  - Internode

U.S. DEPARTMENT OF ENERGY

Other

- Quick reminder that you want to add a point about resource managers needing to support more dynamic allocations. I agree that the RMs really need to be rearchitected.

National Nuclear Security Administration