BLUE WATERS

SUSTAINED PETASCALE COMPUTING

Blue Waters BW 2.0 Working Group Summary

Bill Kramer, Jeremy Enos, Greg Bauer, Bill Gropp

GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION

Announcements

- PI/Irene Lunch
- Tweet of the Day prize

Participation

- Day 1 had 6 science areas with 7 teams
  - MD – Tom Cheatham and Jim Phillips
  - QCD – Steve Gottleib
  - Atmospheric Science - Brian Jewett
  - Helioscience – Nikolai Pogorelov
  - Cosmology/Numerical Relativity – Scott Noble
  - Seismic – Phil Maechling
  - About 15 NCSA staff
- Day 2 had 3 science areas with 4 teams
  - MD – Tom Cheatham and Jim Phillips
  - Helioscience – Nikolai Pogorelov
  - Cosmology/Numerical Relativity – Scott Noble…
  - About 15 NCSA staff

Approach

- Spanning both days
- Discussions of S&E Team needs
- Discussion to understand where the limits and bottleneck are today
- General Discussions of technology options
- Discussion of relevant investment models
- Questions in science areas and technology areas
- Tried drawing conclusions

Science Workflows

- Some areas continue to need large highly parallel applications to add physics, resolution, etc. Trend to higher order methods to increase resolution. Using a courser grid to fit into memory, then more communication and more computation.
- Some areas have expanded insight potential with petascale codes (e.g 100M atoms enables new understanding for many problems) and now see themselves and other groups applying them to more problems.
  - Not ensembles
  - Still 1,000s of nodes
  - Also linked to getting high quality experimental data enable new studies
  - In some cases it is not possible to continue to scale up single problems with system size increase, as other issues (time to equilibrate, need for longer simulated time) start to take proportionally longer wall clock.
- Some areas want to do statistical sampling – ensembles
  - Still 1,000s of nodes
- Some areas need much longer simulated time at the petascale

Science Comments

- Long term forecast validation/testing resources are necessary.
- Validation means connection to data and observation sources
- Thinking from the start to have HW for data integration and larger scale aspects of entire simulation
- As data sets get bigger – it gets harder to collaborate. Need tools and analysis resources for others, remote visualization, etc.
  - Large frame rates and data sets – 3D, shared to remote locations
  - More collaborative opportunities for next generations
- No Moore's law for SW. Lots of legacy codes need to be made faster. Won't re-write in Cuda. Ability to run existing code in future architectures
  - Codes may not have to run efficiently
  - Codes have to run correctly
- HW changes 3 to 5 years, but codes are 20 years in life cycle.
  - Codes are used for 1000x resolution
- How will we deal with the multitude of architectures?
  - Scalar, GPU, Phi

Science Comments

- Some need for relational data bases as simulation and data grows.
- Some see adding stages onto the work flows to refine and improve workflows
  - Often some stages are in different locations – distributed – but for "policy reasons" not technical
    - E.g computation on BW, relational database at USC and needs privileges.
- Workflows and pipelines (biophysics) need to be more integrated.
- Expect to see more ensemble workloads or more complicated workflows to take advantage of larger systems.
- Lack of expertise with rapidly changing architectures continues to be a concern, as well as the potential loss of support for legacy codes. Single source becoming increasingly desirable.
- Some interest in system of systems that shares interconnect and file system in support of preproduction (mesh generation on fat nodes), production (on scalable compute), analysis (on lightweight cores) and visualization (on GPUs).

What are current constraints?

- Some communities need small number of large memory nodes for steps in workflow
  - VMD – shared memory multi-core – needs more memory on node
  - Discussion can SW mask or help to give impression of large memory or using PGAS of SW shared memory
  - Not as critical to high flop rates for the workflow steps
- Need to scale up support – more people to help transform code to new architectures.
- Where is PetaApps programs for next generation?
  - Effective and NEIS-P^2 was effective
  - SCEC had PetaApps helped qualify for PRAC applications
  - Teams of people with different expertise

What are current constraints?

- Need more memory to save distribution functions
  - Will have observations for the distribution functions
  - Will greatly increase memory requirements 1,000x
- Long term large data archive needed
  - [Note discussion on how long data is needed varied from a few years to 20 years]
  - Valuable to have a tool help teams decide to save data for a long term or recalculate if needed – tools to help quantify
- Unpurged file space needed – months between purges
  - Discussed more layers of storage – but also the trade offs of explicit movement

Science Thoughts

- Trend to use coarse grids and higher order methods makes communication becomes more of an issue.
- Do not want to have to use geographically distributed systems due to policy issues
  - Allocations, security and access, resource management, …
  - Use distributed system if technology requires
  - Perceptions of what is efficient use
    - High rates may not be efficient use
- Mixed resources: data analysis system (light weight cores) , compute intensive, shared network or disk. avoid data movement. Slower but larger data storage for intermediate time.

Discussion Points

- Is it cheaper to have focused systems or more general systems?
- Are multiple systems cheaper than fewer systems?
  - Answer tended to no
- Difficult to discuss requirements without implicit assumptions of cost influencing
- Migration to many core/GPU needs much more software support, people to do re-engineering and enabling technology independent source
  - Is domain specific languages useful here?

Technology Thoughts

- Very General –not disclose information so this is "Common Wisdom"
- Starting in 2016 it is possible there will be several viable technologies
  - Processing
    - Maybe 3 General Purpose Processor ISAs
    - Maybe one or two Many-core
    - Attached Accelerator GPU
  - Interconnects
    - IB derivatives
    - Ethernet
    - Intel STL 1/2
    - Aries
    - Specialized variants
  - Memory – did no talk about that too much but will have more hierarchies and packaging
  - Storage – many hierarchies – variety of approaches – did not spend time on it

Bill's Sizing ROM Estimates – major resource

- Quick and Semi Accurate Guesses – only certainty about estimates is they will be wrong
- May do better when the time comes but allowed some thinking
- ~$170 M purchase cost for main system
  - 20% for on-line storage
  - Assume an Aries equivalent interconnect
- Mid to late 2016 and early to mid 2018
  - Type of Processor – GPCPU, Manycore, CPCPU + Accelerator
  - Peak
  - Numbers of Nodes
  - Aggregate Memory
  - Power
- Beyond 2019 is pretty fuzzy
  - Technology and business conditions

Sizing ROM Estimates – 2016

Type	Peak (PF)	Nodes	Agg Mem (PB)	Power (MW)	Comments
GP CPU	28	~20,000	2.4	8-10	SPP 3-7X BW
Many Core	80	~22,500	2	8-10	SPP depends on code mix - in general 3-6X BW
GP CPU + Accelerator	80	~18,000	1	8-10	SPP depends on code mix - in general 3-6X BW

Sizing ROM Estimates – 2018

Type	Peak (PF)	Nodes	Agg Mem (PB)	Power (MW)	Comments
GP CPU	52	~16,000	5.6 NV	7-9	SPP >10X BW
Many Core	80	~20,000	3.7	8-10	SPP depends on code mix - in general - >10x BW
GP CPU + Accelerator	TBD				

Two Strawhorse Investment Strategies

- Multi use, Integrated System
- A System of Sub-Systems

Investment Model A – Multi use, Integrated System

Overall

- Main systems 52%
- Other subsystems 11%
- Other SW 2%
- Dev & Support Personnel 13%
- S&E Teams Improvement Support 3%
- EOT 1%
- Facility 17%

Main System

- Online Storage – 20%
- Compute Analysis Processors – 50%
- Memory – 15%
- Interconnect – 10%
- Node/frame Infra – 5%

Other Systems

- Nearline Storage – 65%
- WAN/LAN Connectivity – 30%
- Cyberprotection – 5%

Investment Model B - "System of Subsystems"

Sub-system	Initial Straw Horse	Group Investment Vector
CPU based High Performance Computation and Data Analysis SS with High BW/low latency Interconnect	35%	Higher
Accelerated SS	20%	Depends
High Throughput	5%	Lower
Visualization and Analysis	5%	Lower
Shared Data Store Hierarchy	25%	Ok
General Interconnect for SS	10%	

Technology Centered: Hardware

- What hardware limits the performance of your applications?
  - Socket performance (FLOPS)
  - Memory bandwidth and/or latency
  - Interconnect bandwidth and/or latency
  - Concurrency hazards (locks, remote atomic updates)
  - Synchronizing collectives (Allreduce)
  - I/O bandwidth and/or latency

Strawhorse Architectural Attributes Priorities

Attributes	S&E Teams (4)	NCSA POCs for their teams
Socket (CPU/GPU) Performance	16	9
Interconnect Bandwidth and/or Latency	13	14
Memory Bandwidth and/or Latency	4	8
I/O Bandwidth and/or Latency	3	13
Synchronization Collectives	0	1
Concurrency Hazards	0	0

Technology Centered: Software

- What software limits the performance of your applications?
  - Note: Need rigor in claims – many false claims have been made based on “there must be a better way”
- Examples of real issues:
  - Compiler/system fails to exploit processor vectorization
  - Complexity of managing distributed memory data structure forced design into a suboptimal compromise

Science Centered

- What is the smallest system on which you could accomplish your science goals?
  - Number of nodes/FLOPS/IO etc.
  - How many such systems would you need?
  - Is there an optimal system size for your applications?
    - Use your own assumptions about node performance
- How is that likely to change over the next 3 years? 6 years?
- Same questions, but for your community
- Does your community have a roadmap for its computation needs?
  - How frequently is it updated?
  - Is there community consensus?
- What problems are not being addressed in your community because computing resources are inadequate?
- What resources would make it possible?
  - Total memory? Total FLOPS? I/O? Other?

Summary

- Even within small group that was wide ranges of needs
- Key is driven by @scale simulation (at least for this group)
    - So most of the investment should be in the simulation SS
- Unclear how many communities and how much workload will be ready to exploit GPU or many core in 3 years