SAND2014-17256PE

FY14 L2 Milestone Overview

Rich Barrett(SNL), David Daniel(LANL), Todd Gamblin(LLNL),
Mike Glass(SNL), Rob Hoekstra(SNL), Louis Howell(LLNL),
Christoph Junghans(LANL), Al McPherson(LANL), Rob Neely(LLNL)

August 18, 2014

Outline

- **Milestone**
  - Description
  - Completion Criteria
- **Scope of Work**
  - Proxy Applications Chosen
  - Platforms
  - Types of Analysis
- **High level messages**
  - Milestone Completed!
  - Lessons learned
- **Lab led talks**
  - Brief outline

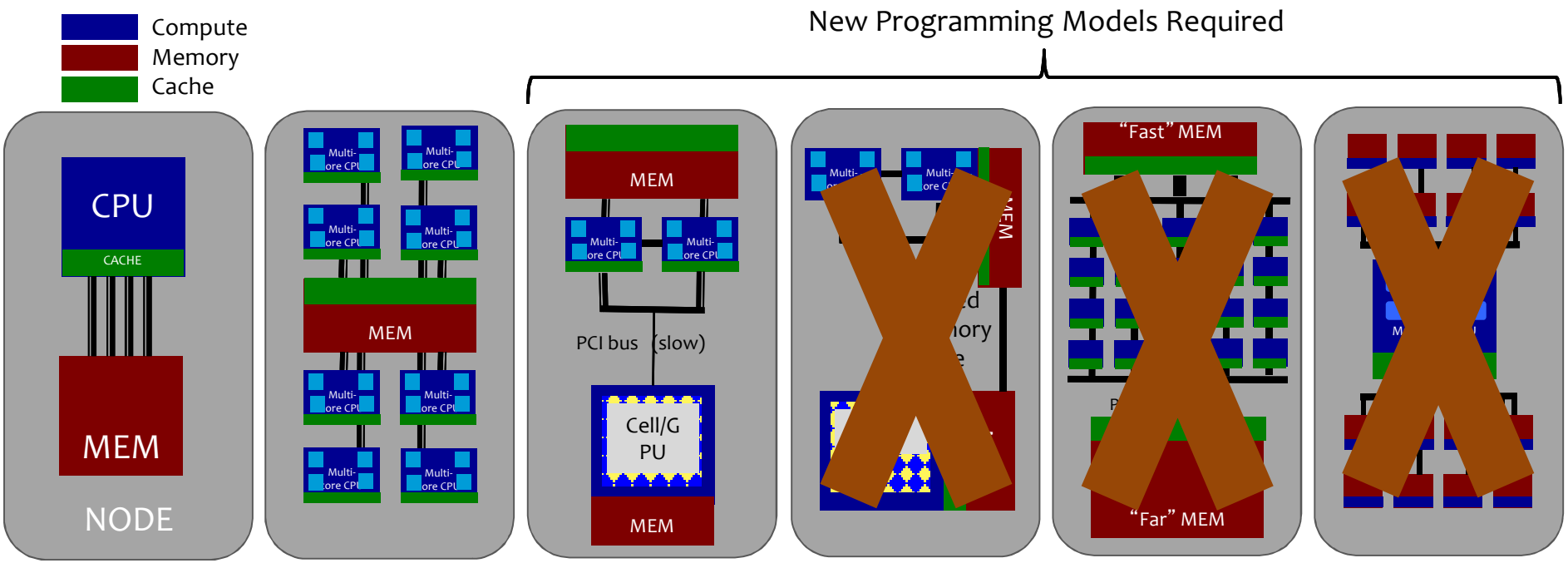
Milestone #4875: Evaluate Application Performance on Advanced Architectures

Milestone (ID#4875): Evaluate Application Performance on Advanced Architectures		
Level: 2	**Fiscal Year**: FY14	**DOE Area/Campaign**: ASC
Completion Date: 9/30/14		
ASC nWBS Subprogram: IC, CSSE		
Participating Sites: LLNL, LANL, SNL		
Participating Programs/Campaigns: ASC		
Description: Each lab will identify two proxy applications that have been demonstrated to be representative of key performance aspects of ASC integrated codes. These proxy applications will be exercised on test beds, advanced systems, or simulators to analyze both performance and scalability issues.		
Completion Criteria: A tri-lab report will detail key performance indicators related to hardware (such as memory bandwidth or latency, and interconnection fabric performance) or software (such as runtime support for task level parallelism or DSLs, advanced compilers, or application development tools).		
Customer: ASC		
Milestone Certification Method: A program review is conducted and its results are documented. Professional documentation, such as a report or a set of viewgraphs with a written summary, is prepared as a record of milestone completion.		
Supporting Resources: Co-design teams from IC and CSSE		

Completion Criteria Proxy Applications

- **LANL**
  - **SNAP: Deterministic Sn Transport**
  - **PENNANT: Unstructured Hydrodynamics**
- **LLNL**
  - **UMT: Deterministic Sn Transport**
  - **MCB: Monte Carlo Particle Transport**
- **SNL**
  - **MiniFE: Implicit Unstructured Finite Element**
  - **MiniAero: Explicit High Mach Aerodynamics**

We are testing against the range of architectures we currently have available



c. 1995. Single CPU per node with main memory.

c. 2000-2010. Multiple CPU's per node sharing common memory

c.2008. Specialized accelerators usher in heterogeneity.

c.2016. Accelerators share common view of memory with CPU

c. 2015. Simple low-power cores and Non-Uniform Memory Access

c. 2020. Processor-in-memory (PIM).

Types of analysis performed by the labs

- Weak Scaling
- Strong Scaling
- MPI Performance
- Memory Usage
- Threading vs. MPI
- Hardware Counters

Meeting the completion criteria

- You will see detailed performance analysis of at least 2 proxies from each lab
- Annotated versions of these 4 talks will serve as the report along with an executive summary
- This will be a snapshot of a much broader collection of performance data for our proxies

Plans for follow up:

- Present findings at SC14 BOF
- Joint Tech Report FY15Q1 (Journal Article?)
- Broader briefings to ASC staff at 3 labs

Lessons learned

- **MPI + threads effective but:**
  - **Hard to get performance win over MPI-only in many cases**
  - **Almost always a significant memory usage win though**
- **Performance bottlenecks vary for applications**
  - **Effective memory bandwidth often a big one**
  - **But, in some cases integer instructions can be the bottleneck**
- **Abstractions have the potential to aid in performance portability**
- **Feedback loop to compiler team crucial to improve performance**
- **Proxy applications are valuable for these activities but we must never mistake them for the real thing!**

3 more detailed 30 minute talks

- **David Daniel & Christoph Junghans: LANL**
- **Louis Howell: LLNL**
- **Rich Barrett: SNL**

Completion

- 4-5pm committee deliberation
- Deliver annotated slides by Wednesday
- Deliver executive summary by Wednesday
- Committee's memo of completion by Friday?

Extra Slides

Proxy Applications Value

- 1) What is the value of proxy applications?
  - An effective means to isolate specific issues for current and future systems
  - Greatly ease communication between computational scientists, computer scientists and computer vendors
  - Enable rapid exploration of programming models, abstraction techniques, and optimization approaches in a quasi-realistic context, for subsequent adoption by a full application
- 2) What are their short-comings?
  - Simplicity can be misleading: a single-physics proxy application may be significantly easier to optimize on challenging architectures (e.g. GPUs) than multi-physics applications with their more dynamic behavior
- 3) What recommendations could be followed to increase their value?
  - Better documentation on how to do scaling studies (particularly weak scaling), physics (parameter ranges that, etc.)
  - Better documentation on how to vary physics (i.e. how to select parameter ranges that test the limits of interest wrt the target computational science)
  - Caveat: Can all of the above be done without a testing framework as complex to grasp and maintain as a full application?