SAND2016-9027R

Sandia National Laboratories

Operated for the U.S. Department of Energy's
National Nuclear Security Administration
by **Sandia Corporation**

Albuquerque, New Mexico 87185-0351
Livermore, California 94551-0969

date: September 8, 2016

to: Bruce Hendrickson, 1400
David Womble, 1220
Ken Alvin, 1420
Rob Hoekstra, 1422
Gwen Voskuilen, 1422
Scott Collis, 1440
Scott Hutchinson, 1540

from: Review Panel: Micheal W. Glass, SNL (1545, Chair); Dennis Dinge, SNL (9326); and Jeff Keasler, LLNL

subject: Final Review of FY16 ASC CSSE L2 Milestone #5676 entitled "Impact of Advanced Memory Architectures on ASC Codes"

The final review for the FY16 ASC Computational Systems and Software Environment (CSSE) L2 Milestone #5676 was conducted on August 17, 2016 at Sandia National Laboratories in Albuquerque, New Mexico. The review panel unanimously agreed that the milestone has been successfully completed.

Gwen Voskuilen (1422) led the milestone team and presented the results. The review panel was comprised of staff from several Department of Energy national laboratories that are involved with next generation platform development from both a hardware and software perspective. The Panel consisted of Micheal W. Glass, SNL (1545, Chair); Dennis Dinge, SNL (9326); and Jeff Keasler, LLNL. The presentation documented the technical approach of the team and summary of the results with sufficient detail to demonstrate both the value and the completion of the milestone. A separate SAND report was also generated with more detail to supplement the presentation.

The purpose of the milestone was to analyze the feasibility of next-generation multi-level memory (MLM) system architectures to impact the performance of ASC applications. The milestone team made extensive use of simulation tools such as the Structural Simulation Toolkit (SST) and hardware evaluation testbeds where available. The presentation consisted of 6 major topics (1) Methodology; (2) Design Space Exploration; (3) Software / manually managed MLM; (4) Hardware / automatically managed MLM; (5) Costs / Overheads of managed MLM; and (6) Recommendations and Conclusions.

Milestone Completion Criteria

1. Analyze the feasibility of next-generation memory system architectures to increase performance of ASC applications

   The team chose a crosscutting subset of miniapps from the APEX suite. Using the APEX miniapps was a very good choice for a couple of reasons. The APEX RFP will soon be released and the contract is scheduled to be awarded in early FY17. This will present a great co-design opportunity going forward with the APEX miniapp suite and this work to evaluate MLM impact using the SST. These miniapps are also larger and more complex than previous miniapps. This pushed the bounds of simulation with the SST and resulted in enhancements to the SST to improve its capabilities, performance, and scalability.

   The team evaluated the miniapp performances on two different architectures and the affect of manually or automatically managing MLM.

   The panel agrees that the milestone team met this criterion.

2. Quantify improvements in the memory system performance or power compared to conventional memory systems

   Memory system performance on the miniapps was explored with the SST on both a lightweight and heavyweight architectures. The lightweight architecture consisted of a tile of 72 less powerful cores with HMC and DDR memory. The heavyweight architecture consisted of ring of 8 more powerful cores with HMC and DDR memory. Performance comparisons between running in DDR vs. running in HMC were made for each of the architectures. The lightweight architecture mimics the KNL architecture of the ASC Trinity phase 2 KNL system that will go into production in FY17Q2. Because of this, comparisons where made between the lightweight architecture simulation results and actual results obtained on an early KNL hardware testbed. For the most part, the trends from the simulator matched the trends from the real hardware testbed.

   The team plans to use SST to examine the energy impacts of multi-level memory system design in the future.

   The panel agrees that the milestone team met this criterion.

3. Explore the memory system design space

   Memory system design space was explored with the SST on the miniapps by varying memory bandwidth and latency on the heavyweight architecture mentioned above. The bandwidth was doubled (a likely next generation improvement) and latency was increased by ~15% and ~75%. Some of the miniapps were particularly sensitive to latency and bandwidth. A general conclusion was to not increase bandwidth at the cost of higher latency. The HBM (vs DDR) proportion of the total application memory footprint was also varied from $1/8^{th}$ to 100% and simulated.

   The panel agrees that the milestone team met this criterion.

4. Determine the overall impact on the applications, system software, and system balance

   Various schemes for manually and automatically managing MLM were explored.

   For manually managed MLM, both OS-managed and programmer-managed approaches were considered. A new SST based tool, MemSieve, was developed to profile application memory behavior. The tool captures the application's memory accesses and correlates them to the application's memory allocations. The key metric is malloc density with the hypothesis that dense mallocs should be put in fast memory. MemSieve was used to perform a malloc density analysis on several of the miniapps to identify the dense mallocs. As a case study, the miniFE miniapp was run on a KNL testbed with the top dense mallocs identified by MemSieve being manually placed into HBM. This resulted in a 1.5x performance improvement over pure DDR but was less than the 4.5x improvement by having everything in HBM. A SST MLM API was developed and implemented to enable simulation of various policies for MLM management. Four policies were explored: (1) greedily allocating pages in HMC; (2) greedily allocating mallocs in HMC; (3) statically allocate dense mallocs in HMC (using MemSieve for guidance); and (4) dynamically allocating the most dense mallocs in HMC (using MemSieve for guidance) for each phase of the application. In general, the OS managed MLM with programmer hints showed the most promise. The results were similar between the lightweight and heavy weight architectures which means applications may not have to repeat the malloc density analysis for each architecture changes.

   For automatic hardware managed MLM, caching pages in HMC was simulated with the SST. Automatic page-level swapping with a matrix of seven page addition policies and six page replacement policies were explored on the heavyweight architecture and a variety of miniapps. The addition policy choice was found to have much greater significance than any choice of replacement policy. "swap throttling" (not issuing swaps to or from fast memory when the DRAM is busy) had a significant impact when a simple touch threshold add policy was used.

   A cost/performance analysis was performed for several of the miniapps to determine the benefits of utilizing MLM in a system design.

   The panel agrees that the milestone team met this criterion.

5. Determine the impact of performing some computation, synchronization, or data movement operations in the memory system

No simulation results were presented for processing-in-memory (PIM) but a methodology was presented where the newly developed MemSieve tool's dense allocation ranking could be used to identify allocations that would benefit from PIM. Allocations likely to be in DDR would incur an overhead to migrate them to HMC for PIM. Allocations likely to end up in HMC (manually or automatically) could benefit from PIM without incurring the overhead of memory migration. The team suggested that PIM computation might be useful for "some of the small OpenMP loops with no child calls".

The panel agrees that the milestone team met this criterion.

The review panel would have liked access to the presentation and report (even an early draft) prior to the review to better familiarize ourselves with the scope and content for the review. That said, after reviewing the material after the fact, we believe this milestone has significantly advanced the current

state-of-the-art with respect to system simulation. The milestone team was particularly impressed by the advances made to the SST for improved performance, scalability, and capabilities (MemSieve). Being able to effectively utilize the SST for on the larger and more complex APEX miniapps requiring 1GB to 8GB data requirements shows hope for being able to run a real application through the SST in the future to assess various system designs on real applications in a timely manner. Even though this milestone heavily relied on simulations with the SST, it was good to see validation with results from real hardware testbeds where available.

Re:

- Presentation: **Multi-Level Memory Milestone**, SAND2016-8726C, September 2016
- Final Report: **Evaluating the Opportunities for Multi-Level Memory – An ASC 2016 L2 Milestone**, SAND2016-8845, September 2016