SAND2018-1117D

# ECP Memory Working Group Progress and Plans

Hardware & Integration
Hardware Evaluation
Team

LLNL: Maya Gokhale <gokhale2@llnl.gov>, Alfredo Gimenez <gimenez1@llnl.gov>
ORNL: Shirley Moore <mooresv@ornl.gov>, Jeffrey Vetter <vetter@ornl.gov>, Ivy Peng <pengb@ornl.gov>
SNL: Gwen Voskuilen <grvosku@sandia.gov>

## Overview

**The memory working group is responsible for providing analysis and solutions to questions about memory when designing future HPC systems. This includes:**

- Predicting the performance of future and hypothetical memory architectures, e.g., HBM-based
- Providing data-driven recommendations for hardware design and procurement decisions
- Analyzing how HPC codes utilize memory resources, and how they should be modified for future systems

### Analysis Questions

- How will our codes benefit from HBM2/HBM3?
- What are tradeoffs between GPU GDDR5 vs. HBM?
- How can we benefit from NVM?
- Can ECP applications that heavily use indirect addressing A[B[i]] benefit from gather/scatter hardware?

### Working Milestones

- Identify set of orthogonal memory-intensive benchmarks
- Develop a unified trace format that covers what different memory data collection tools can express
- Generate a repository of memory traces on different compilers x architectures x benchmarks

## ASPEN + RTHMS

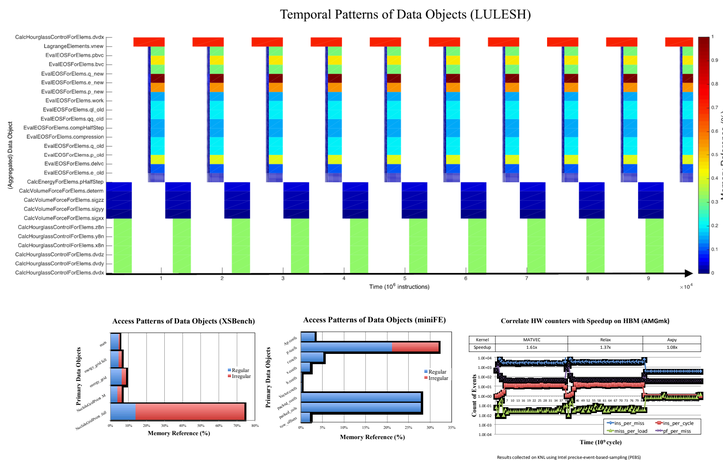
**RTHMS** PIN-based tool provides customizable data-centric profiling and tracing
**ASPEN** analytical performance modeling tool suite is being extended to model heterogeneous memory systems (not yet released, available by request)

**Users:** AMD and Cray. Cray has used ASPEN for PathForward evaluation and extended ASPEN with the following contributions:

- Cache model
- Full application models (not just kernels) for CoMD, MCB, and LULESH.
- Full execution models for performance and power for Intel Broadwell and an abstract heterogeneous exascale node

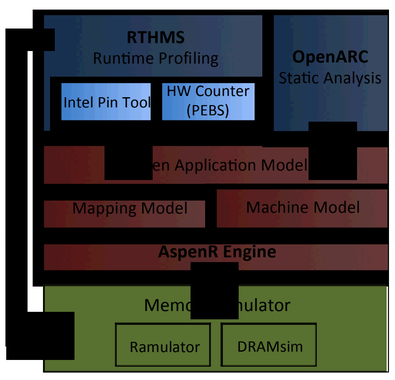
### Progress

- Used RTHMS to survey data access patterns in apps (e.g., XSBench, Graph500, miniFE, LULESH)
- Extended ASPEN to support common access patterns in app abstraction
- Analyzed data placements on hybrid memory systems like KNL
- Developed ASPEN interface to memory simulators (e.g., Ramulator, DRAMsim)




### Future Plans

- Continue survey of ECP app data access patterns
- Develop ASPEN models of various memories and app data access patterns
- Generate traces from parameterized app models for input to simulators
- Validate that traces are representative of real app execution

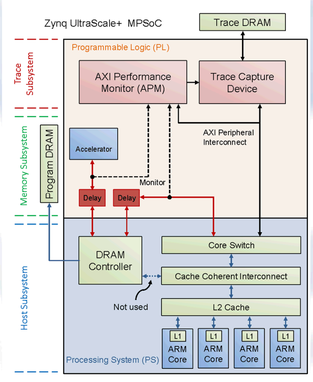


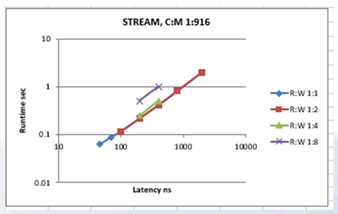

## LiME – Logic in Memory Emulator

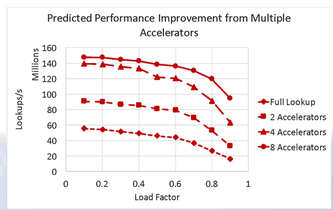
**Logic in Memory Emulator (LiME)** on Xilinx ZC706 FPGA board provides non-intrusive, complete memory trace capture and emulation of wide range of memory latencies
**Users**: UCSC; ARM adopted and extended our ideas on near memory gather/scatter units as demonstrated in LiME, built proprietary simulator

### Progress

- Ported LiME hardware/software/benchmarks to UltraScale+ -- from 32b ARM to 64b ARM
- Ported Zynq 7000 Linux address map to LiME (capture and delay memory references)
- Developed a hybrid emulator/simulator framework that models heterogeneous hardware acceleration components built using LiME framework to SST
  - Enable additional scaling experiments such as multiple accelerators, enable interoperation with other SST memory module




- Emulator has been ported to Xilinx Ultrascale+ MPSoC
  - Quad core ARM A53
  - Provides capability for non-intrusive memory trace capture
  - Provides capability for emulating wide range of memory latencies
  - Four GB DDR4 on one board, 16GB on a new board

### Future plans

- Linux boot modifications to map memory addresses into Programmable Logic port to Ultrascale+
- Add hardware modules in PL to support additional concurrently accessed memory types including cache
- Run Memory Group benchmarks under LiME environment to generate memory traces for vendor and community use
- Run latency sensitive memory group benchmarks on LiME for independent assessment of performance with complex memory hierarchy
- Continue to interoperate LiME with SST
- **Overall goal is to provide independent assessment of complex memory hierarchy effects on latency sensitive code**

## Analysis of HBM2 vs HBM3

### Approach

- Evaluate performance gap between HBM2 and HBM3 for ECP proxy apps
- Simulate proxy apps in SST
  - CramSim (from IBM FastForward2) for HBM2/3
  - Ariel PIN-based processor to run proxy apps on a KNL model
  - Validate KNL model against hardware for HBM2
- Use Caliper to collect/compare memory performance from hardware
  - Traces, samples, data structures, bandwidth, access patterns
- Extrapolate results to larger codes
- Initial proxy app targets:
  - Quicksilver, Laghos, SW4Lite, SWFFT, MILC, AMG microbenchmark

### Progress

- KNL model running on SST
- Validating against KNL hardware using STREAM
  - Simulated bandwidths within 10% for 10M element array size
  - Evaluating a new NoC model with split data and control networks

## Memory Traces

**Objective:** provide common baseline for collecting, simulating, and comparing memory access patterns

- Establish common trace format for memory tools
- Collect memory traces from benchmarks, proxy apps, mini apps
- Provide data base of traces to ECP community

| Source | Type | Address | Length | AXI ID | Time |
|---|---|---|---|---|---|
| 0 | W | 0x400082E80 | 64 | 525 | 415552 |
| 0 | W | 0x400082EC0 | 64 | 525 | 415890 |
| 0 | W | 0x400082F00 | 64 | 525 | 416228 |
| 0 | W | 0x400082F40 | 64 | 525 | 416566 |
| 0 | W | 0x400082F80 | 64 | 525 | 416905 |
| 0 | R | 0x400001180 | 64 | 1293 | 417208 |
| 0 | R | 0x400001100 | 64 | 1421 | 417210 |
| 0 | W | 0x400082FC0 | 64 | 525 | 417247 |
| 0 | R | 0x400001200 | 64 | 1037 | 417770 |
| 0 | R | 0x400001240 | 64 | 1165 | 417779 |
| 1 | W | 0x400080000 | 8 | 3 | 418494 |
| 1 | R | 0x400082000 | 16 | 0 | 418309 |
| 1 | W | 0x400080008 | 8 | 3 | 418501 |
| 1 | W | 0x400080010 | 8 | 3 | 418508 |
| 1 | W | 0x400080018 | 8 | 3 | 418515 |
| 1 | W | 0x400080020 | 8 | 3 | 418600 |
| 1 | W | 0x400080028 | 8 | 3 | 418607 |
| 1 | W | 0x400080030 | 8 | 3 | 418614 |
| 1 | W | 0x400080038 | 8 | 3 | 418621 |
| 1 | W | 0x400080040 | 8 | 3 | 418711 |

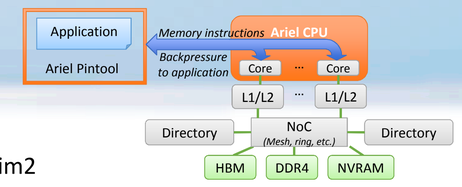
## SST Memory Simulation

The **Structural Simulation Toolkit (SST)** enables simulating novel memory systems both standalone and as part of a larger node architecture. SST supports memory models from a variety of industry, academic, and lab sources. These models include DDR3/4, HBM, HMC, and non-volatile.

**Users:**

**Source:**

- https://github.com/sstsimulator
- http://sst-simulator.org




### Progress

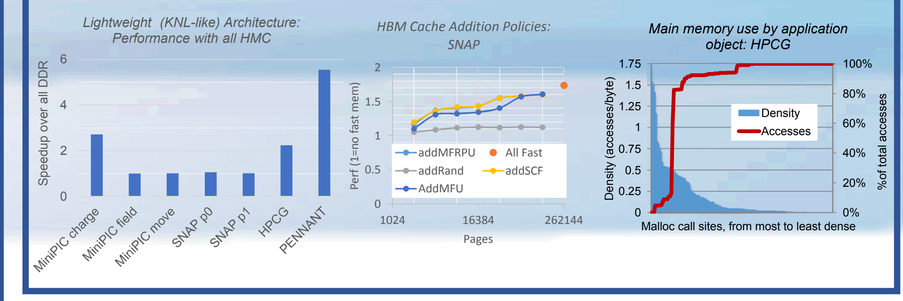
- New simulation models
  - Scratchpad memory
  - NoC-oriented network
  - Integration of HBMDramSim2
- Cache banks
- Improvements to TimingDRAM (DDR3/4 model)
- Custom memory command support
- Memory/scratch controller initiated cacheline shootdowns

### Current Work

- Analyzing proxy app performance on HBM2 and HBM3

### Future Plans

- Analysis of proxy app performance on projected exascale architectures and memory systems
  - Where the memory system is over and under-performing
  - Where applications are unable to effectively use memory
- Memory behavior analysis for proxy apps



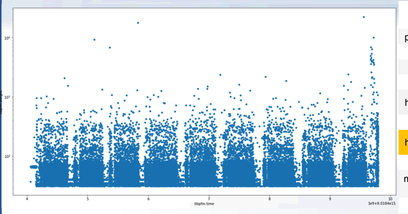

## Caliper Memory Performance Data Collection and Analysis

**Caliper** memory analysis tools enable data-centric memory analysis, memory hardware + instruction profiling, allocation tracing/analysis, attribution to source code and data structures https://github.com/LLNL/Caliper
**Users:** Used in many codes at LLNL (ARES, ALE3D, Ardra, ParaDiS) and proxy/mini-apps (Laghos, Kripke, Quicksilver, LULESH). Integrated with other performance tools (Intel Vtune, NVIDIA NVProf, HPCToolkit, Allinea)

### Progress

**Memory Performance Data**

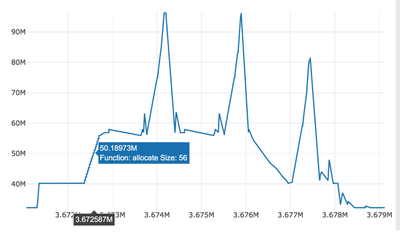
- Libpfm service enables cross-architecture memory performance data collection
- Intel (SNB,IVB, HSW, BDW), Power8, AMD
- Memory access sampling + latencies + data addresses, memory counters

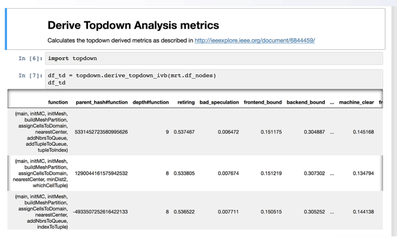


| source.function | source.line | source.file | memory_level | counts |
|---|---|---|---|---|
| psm2_mq_ipeek2 | 0.0 | UNKNOWN | L1 | 24317 |
| _int_malloc | 0.0 | UNKNOWN | L3 | 1177 |
| __strcmp_sse42 | 0.0 | UNKNOWN | L3 | 1159 |
| UNKNOWN | 0.0 | UNKNOWN | L3 | 987 |
| hash_lookup_elm | 79.0 | ares/sources/hash/hash_lookup_elm.cc | LFB | 986 |
| hfi_get_mylabel | 0.0 | UNKNOWN | LFB | 810 |
| hash_lookup_elm | 84.0 | ares/sources/hash/hash_lookup_elm.cc | L3 | 652 |
| UNKNOWN | 0.0 | UNKNOWN | LFB | 606 |
| malloc_consolidate | 0.0 | UNKNOWN | L3 | 509 |
| _init | 0.0 | UNKNOWN | L3 | 503 |

**API for Data Tracking**

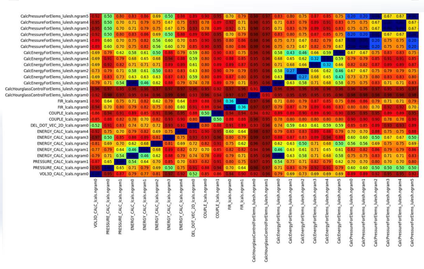
- Allocation statistics (total allocations, maximum active allocations, count per size), instrumented within UMPIRE
- Memory address samples (e.g., PEBS) resolved to allocation containers




### Future Plans

- Extrapolate results from small tests (simulators, mini-apps) to full-scale codes
- Identify optimization potentials (NUMA, HBM, arena allocators)
- Compare memory performance across different platforms



Los Alamos NATIONAL LABORATORY
Sandia National Laboratories
Lawrence Livermore National Laboratory
ECP EXASCALE COMPUTING PROJECT
OAK RIDGE National Laboratory
Argonne NATIONAL LABORATORY
BERKELEY LAB
NNSA National Nuclear Security Administration
U.S. DEPARTMENT OF ENERGY | Office of Science

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Science Administration under contract DE-NA0003525.