Heterogenous Memory Challanges

Mike Lang*

*My thoughts not LANL's

Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA

MCHPC questions

Usability and programmability of complex and heterogeneous memory systems remain significant challenges facing the HPC and data analytics communities. Existing memory systems that include DRAM, SRAM, discrete memory, software unified memory, and distributed memory are difficult to exploit while maintaining portable performance. Approaches include programming language constructs and runtime libraries, OS enhancements, and even hardware mechanisms to enable the competing goals of programmability and portability.

We would like the panel to address challenges and solutions that address the problems of maintaining portability in applications that must navigate the complex memory hierarchy without sacrificing performance and capability.

Challenges/Solutions from a mechanism view

Discovery – Still not fixed –Some promising work from ARM and the HMAT Vendors not really populating the structures. Hwloc is on good place and multiple folks are looking at make it work the systems coming now, but not a long term adopted solution.

Attributes: *default, large_cap, const, high_bw, low_lat) _mem_spaces*, OpenMP opted for high level attributes, would runtime developers or app developers want more performance details?

- If you want more how do you characterize it: Stride, data size, …

Allocation and Deallocation: Focus on data used together, locality (not new).

Migration: Move data between heterogenous memories. Applications need to share or stage these resources to solve real problems, capacity is still an issue for HBM.

Arbitration: multiple libraries want to use different memories, how do we allow this with out rewriting them all to use the same memory-libraries?

**Portable mechanisms

Other Challenges/Opportunities

Semi-Custom – With the Denard/Moores dead folks are looking to semi-custom to keep an influx of performance on future systems.

- A chance to tune the architecture for you particular workload mix
- In both memory, processing power and accelerators.
- MCM populated with CPU of choice and accelerators

Solution

Hard code it – There are so few examples should we just hard code things and worry about a portable solution when it becomes problematic to maintain the hardcoded parameters of the system available?

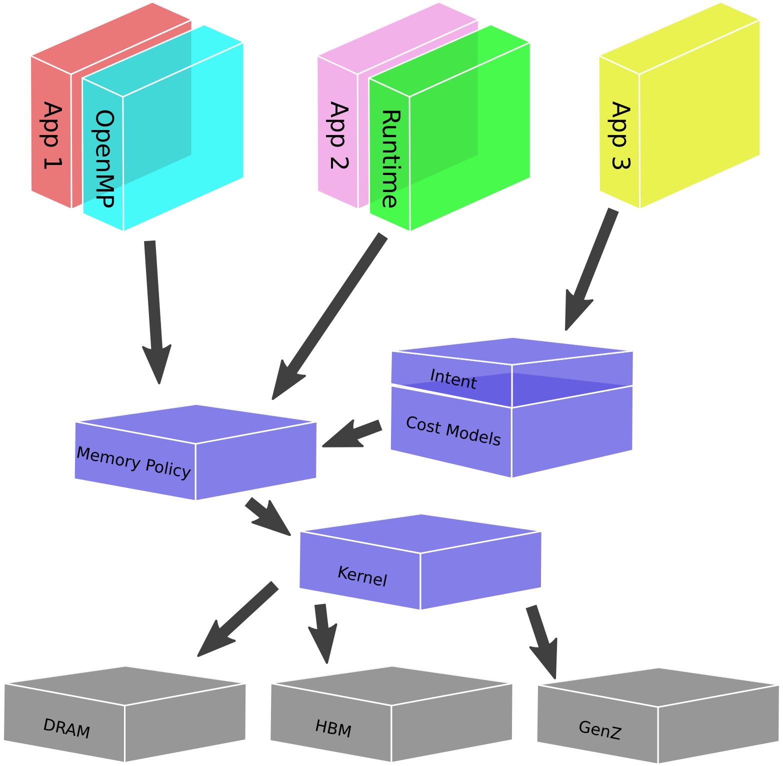
I talked to an HPC vendor … no interest, there won’t be a heterogenous memory hierarchy.

This also depends on if you target outside of HPC for adoption

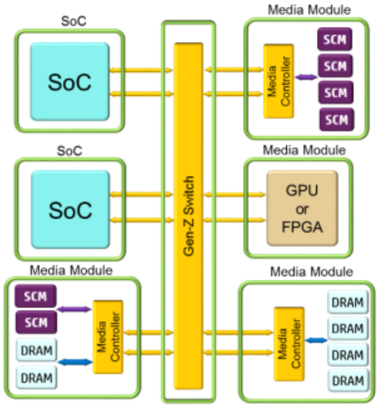
Memory should be unified, anything else is a step backwards.

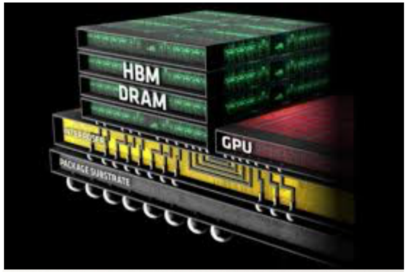
Power as an example … HW will take over eventually

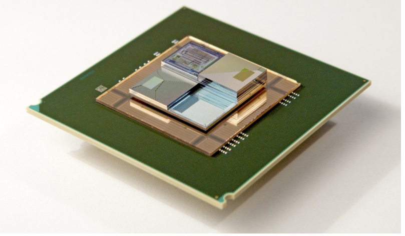
Simplified Interface to Complex Memory (SICM) Overview



Allocate
Deallocate
Migrate
Arbitrate
&
Introspect
Memory
in a
portable
manner

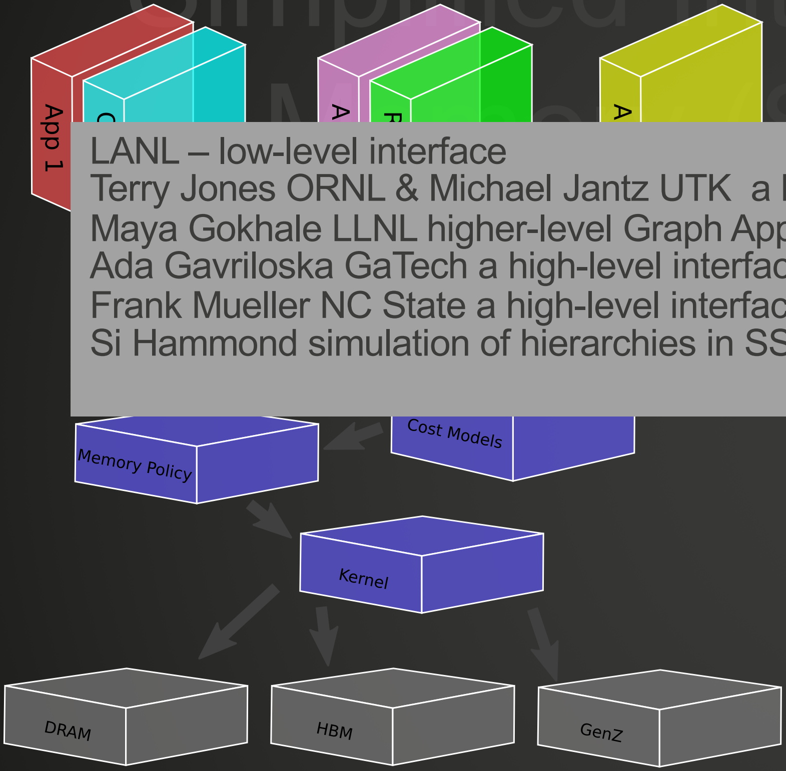




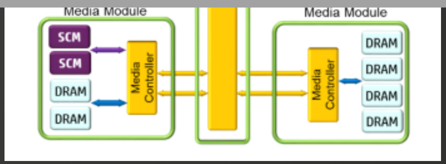


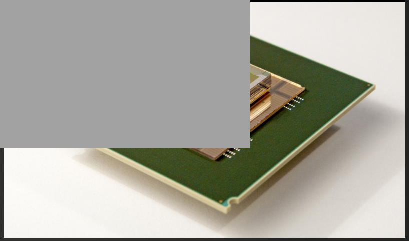
Simplified Interface to Complex Memory (SICM) Overview

LANL – low-level interface
Terry Jones ORNL & Michael Jantz UTK a high-level interface via active profiling for data placement
Maya Gokhale LLNL higher-level Graph App interface
Ada Gavriloska GaTech a high-level interface, app characterization, ML approach
Frank Mueller NC State a high-level interface using static analysis in the LLVM layer
Si Hammond simulation of hierarchies in SST



&
Introspect
Memory
in a
portable
manner





SICM approach & preparation for exascale platforms.

- **Provide abstraction for Heterogenous memory for runtimes and applications**
  - focusing on arenas for data structures that are used together
  - Trying for inclusion in existing open source projects rather than starting a new one (CLANG, OpenMP, hwloc, Umpire, Jemalloc …)
- **Pre-exascale environments you are using.**
  - Sierra/Summit nodes P9+Volta,
  - Intel CascadeLake + Optane,
  - Intel KNL with MCDRAM,
  - Intel w/ GPU, AMD w/ GPU -- Builds in the Aurora environment. (last week)
    - **Only** supporting unified memory architectures
- **Investigating automation of data placement**
  - **UTK, GaTech, NC State**

SICM low-level: Generate SICM call from OpenMP Pragmas

Patches to Clang to turn OpenMP memory spaces in OpenMP 5.x into sicm library calls in the LLVM/OpenMP runtime.

Supports Compile SICM runtime with CLANG/LLVM

At init time it does DLOpen to find SICM library, if found it it uses SICM to satisfy the

pragma openmp allocate

OpenMP memory types: *omp_ (default, large_cap, const, high_bw, low_lat) _mem_spaces*

Currently supports KNL, Optane, testing Sierra

Same codepath that supports memkind library, refactored to support multiple custom memory allocators – more general than SICM support.

SICM low-level: SICM initial investigation of Intel’s 3DXpoint & update of SICM build system

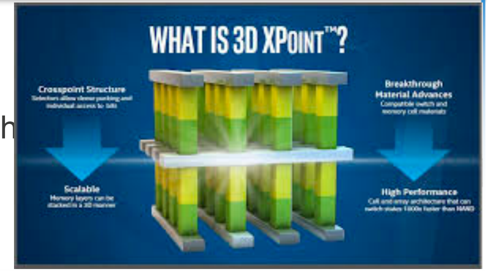
ECP WBS	2.3.1.16 SICM
PI	Michael Lang, LANL
Members	LLNL, ORNL, SNL

Scope and objectives

- Testing of 3dxpoint which is expected on A21
- Rework of build system to use Cmake and Space and Travis for continuous integration.
- To get experience on A21 hardware and to realign build system with ECP common practices.

Cool image

Intel 3Dxpoint allows Non-Volatile memory to be mapped into traditional DRAM space allowing much high memory capacity for applications but at a lower performance.


WHAT IS 3D XPOINT™?
Crosspoint Structure
Selectors allow denser packing and individual access to cells
Scalable
Memory layers can be stacked in a 3D manner
Breakthrough Material Advances
Compatible switch and memory cell materials
High Performance
Cell and array architecture that can switch states 1000x faster than NAND

Impact

- Experience with 3dxpoint will allow support by SICM library.
- Rework of build system will allow easier integration into ECP SDK. Cmake and Spack.

Project accomplishment

- Availability of 3Dxpoint lifted the priority of this milestone.