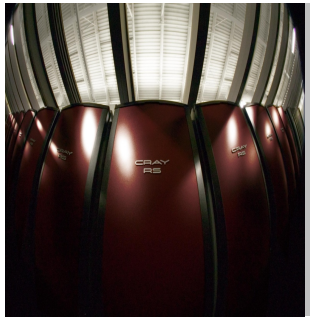

Sandia National Laboratories

Exceptional service in the national interest

Memory vs. Storage: Is There a Difference and Do We Care Anymore?

Jay Lofstead

gflofst@sandia.gov

July 23, 2019

PIRL 2019

SAND2019-8513 PE

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

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 Security Administration under contract DE-NA0003525.

What I Work on

Sandia National Laboratories

- Large scale scientific and engineering compute
  - scale up--not out
  - Think weather simulations as an example
  - Single task on 200K+ cores, 1 PB memory data footprint
  - Save all data every 15 minutes for 24 hours (short run) or maybe 3-6 months (long run)
- Typical high end machine
  - 16,000+ nodes
  - 1.5+ PB DRAM
  - 200K+ cores
  - 100+ PB storage
  - Dual 200 Gb network ports per node
  - 2 TB/sec storage bandwidth

Persistent Memory Questions

Sandia National Laboratories

- Can PMEM better support compute or storage operations?
- Are there other purposes where PMEM can help?
- How much advantage does PMEM offer over NVMe or remote fast memory/storage?

The Shifting Storage Landscape

Sandia National Laboratories

- "Modern" computing storage had disk and tape for decades
- Solid state storage becoming "affordable" started to shift this about 15 years ago with PCIe-based flash
- Burst Buffer coined MSST 2012
  - Earlier work goes back to 1996 at Sandia (Ober SAND96-0691C)
- Now QLC flash essentially the same price for capacity as disk
  - Write endurance doesn't really matter
  - Performance characteristics MUCH better than disk

The Shifting Memory Landscape

Sandia National Laboratories

- "Modern" memory storage started with registers and main memory
- Then layers of cache added
  - Different instruction and data caches too
- Now high bandwidth, on package memory being deployed
  - My first question when encountering multi-core was, "why aren't we doing on package memory instead?"
- In network "memory" being deployed (HPE's The Machine)
  - On network switches and dedicated locations

- How does PMEM fit into this for scale up compute?
  - Price, performance, functionality, and usability all matter

Considering PMEM (scale up)

Sandia National Laboratories

- The memory bus outside the CPU package is now empty
- SATA is too slow to bother
- NVMe is expensive, but compelling—and slower than PMEM

- New architecture
  - Memory: registers->cache->HBM on package->memory bus?
  - Storage: memory bus?->NVMe->remote storage->disk/tape

- If we add PMEM, what is it? How should it be used?

How is “Storage” Used?

Sandia National Laboratories

- Assumption 1: Data lifetimes exceed the application’s lifetime
- Assumption 2: Knowledge to understand data format and types encoded with the data
- Assumption 3: Writing architectural assumptions may not hold when reading
- Assumption 4: Writing/Reading is “slow” compared to memory

How is “Memory” Used?

Sandia National Laboratories

- Assumption 1: Data only lives as long as the application
- Assumption 2: Knowledge for understanding the data is encoded in the application
- Assumption 3: writing and reading environment identical
- Assumption 4: writing/reading is “fast” compared to storage

Now add PMEM

Sandia National Laboratories

- Question 1: Is it slow memory or fast storage?
- Question 2: Does persistence matter?
- Question 3: If it is across a fast interconnect, does that matter?
- Question 4: Does it have to be natively byte addressable?
  - Consider cache lines vs. blocks or pages; isn't everything byte addressable through a suitable API?

Pop Quiz! Memory or Storage?

Sandia National Laboratories

1. Organizing data for highest density?
2. Organizing data in an archive/long-term reuse format?
3. Application writing checkpoint to disk using a raw memory dump?
4. ECMWF computing on data in a format identical to archive and network transmission format?

What do Apps People See?

Sandia National Laboratories

- PMEM is slow memory
  - But faster than interconnect bottleneck to access remote memory/storage
  - Extra capacity!
- CPU load/store instructions work so programming is easy
- Opportunity: Out-of-core computations become feasible
  - This is the HBM model
  - But now I need to write memory swapping code into my application
    - And it needs to try to overlap compute and copy with the best net performance

What do Storage People See?

Sandia National Laboratories

- PMEM is fast storage
  - Quickest path to persist data in case of power loss or crash
  - Make an in-memory output buffer directly in PMEM and then migrate to off node storage somehow
- Can map directly to storage, but access via memory ops
  - Use same API and semantics, but different interface behind the scenes
- Opportunity: Can extend remote storage onto the node
  - Auto-migrate off node for better resilience over time

Who Wins?

Sandia National Laboratories

- Why do we have big machines?
- What is the easiest way to program for performance?
- What usage model is most important?
- What about transient data staging?
- What about workflows and network communication?

Memory vs. Storage: Bottom Line

Sandia National Laboratories

- Data Lifetimes not particularly interesting difference
  - Burst buffers and other transient data storage
- Architectural assumptions almost 100% standardized now
- Speed differences from fastest to slowest in “type” not meaningful
  - Registers vs. PMEM or PMEM vs. disk (or even tape)
- Data Encoding might be the only meaningful difference
  - Except that pesky ECMWF example eliminates even that

What do we need to do?

Sandia National Laboratories

- If memcpy is the desired IO API, what primitives do we need to implement this effectively?
- How does a programmer know if something is in “slow” or “fast” media? How much do they care?
  - Kokkos has some semblance of a model for this
- What about the byte-to-block translation? Is it a meaningful difference a programmer needs to really care about? Can middleware take care of that?
- How long until AMD, Power, and ARM support PMEM? What about RISC-V?

What do we need to do?

Sandia National Laboratories

- How do we encode appropriate metadata in the output blob assuming a memcpy interface?
- When machine architectures differ/change, how do we identify data encoded in an old vs. new format?
- Since we have to archive and migrate data regularly, we are hitting the memory bus to move it. Do we just assume synchronous IO? Can we decouple PMEM access from HBM on the CPU package? Can we prioritize access to avoid compute jitter?

Questions

Sandia National Laboratories

gflofst@sandia.gov