SST Simulation Framework (and Complex Memory)

SST Team and Collaborators

Center for Computing Research
Sandia National Laboratories, NM
What is SST?

**Goals**
- Become the standard architectural simulation framework for HPC
- Be able to evaluate future systems on DOE/DOD workloads
- Use supercomputers to design supercomputers

**Technical Approach**
- Parallel
  - Parallel Discrete Event core with conservative optimization over MPI/Threads
- Multiscale
  - Detailed and simple models for processor, network, & memory
- Interoperability
  - Many components
- Open
  - Open Core, non-viral, modular

**Status**
- Parallel Core, basic components
- Current Release (7.1)
  - Improved components
  - Modular core/elements
  - More Internal documentation

**Consortium**
- “Best of Breed” simulation suite
- Combine Lab, Academic & Industry

How can we use SST?

- Virtual prototyping environment for studying complex future node designs
  - Processors, caches, network-on-chip, memory systems/controllers ..

- Simulate and analyze nodes and hardware we don’t yet have
  - Use vendor specifications to design virtual environment
  - Use our imagination to design future node designs

- Support software, application and algorithm design and codesign
Example: Non-Volatile Memory

- **DIMM: 1+ ranks** → ranks consist of 1+ banks + row buffer
- Ancillary structures: write buffer, request buffer, scheduler, wear leveler (Start-Gap), power management
- Key Latencies: \( t_{CMD} \) (Command), \( t_{RCD} \) (read to row buf), \( t_{CL} \) (read col), \( t_{BURST} \) (transfer data), \( t_{CL\_w} \) (write)
- Can model fundamental timing/scheduling parameters
Write Latency & Cancellation

- Write latency stepped by 100 cycles from 100 to 1k
- Mitigation: Write Cancelation
  - Cancels pending write operations in order to service read operations
  - Can hurt performance at low write latencies (higher bank occupancy)
  - Adaptive thresholds can be used to balance read latency with the number of outstanding writes
Multi-Level Memory

- Main memory comprised of NVM and DRAM
  - Capture low cost of NVM and performance of DRAM
- Controller can implement a variety of management policies
  - addMFRPU – More Frequent More Recent Previous Use w/Threshold
  - addT – Simple Threshold
  - LRU – Least Recently Used
Multi-Level Memory Evaluation

- **MLM Policy**: addMFRPU, varying the threshold and presence of cache
  - Results are application dependent
- **Varying amount of DRAM**
  - Most applications were insensitive to changes
MLM only useful if cost and performance effective
Most of the applications have worse performance than DRAM
NVM systems also much lower cost than DRAM