SAND2022-10516C

Multifidelity DRAM Simulation in SST

Patrick Lavin

Sandia National Labs

Ryan Lynch • Sudhanshu Agarwal • Jeffrey Young • Richard Vuduc

Georgia Institute of Technology

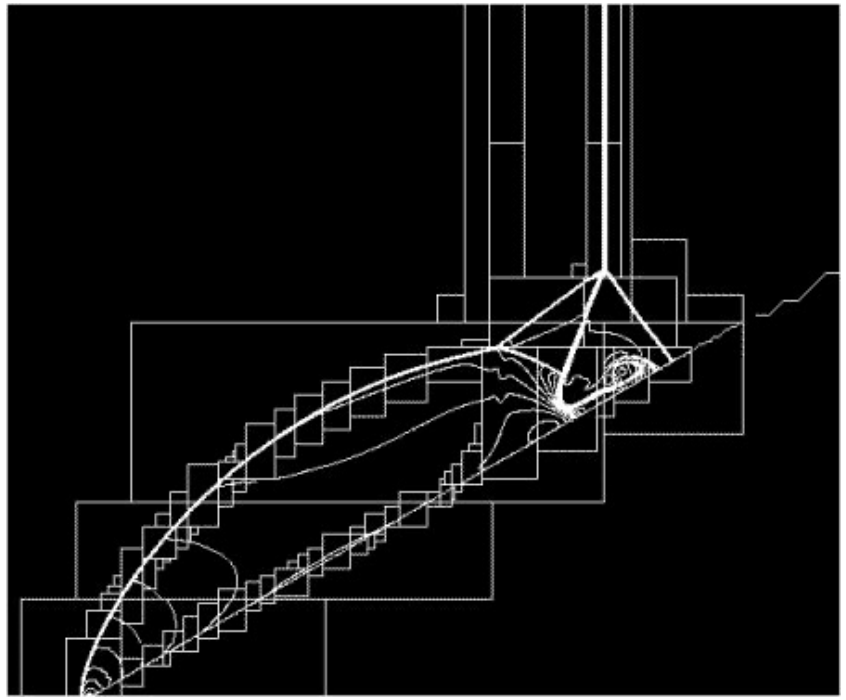
Sandia National Laboratories

Georgia Tech

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & 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.

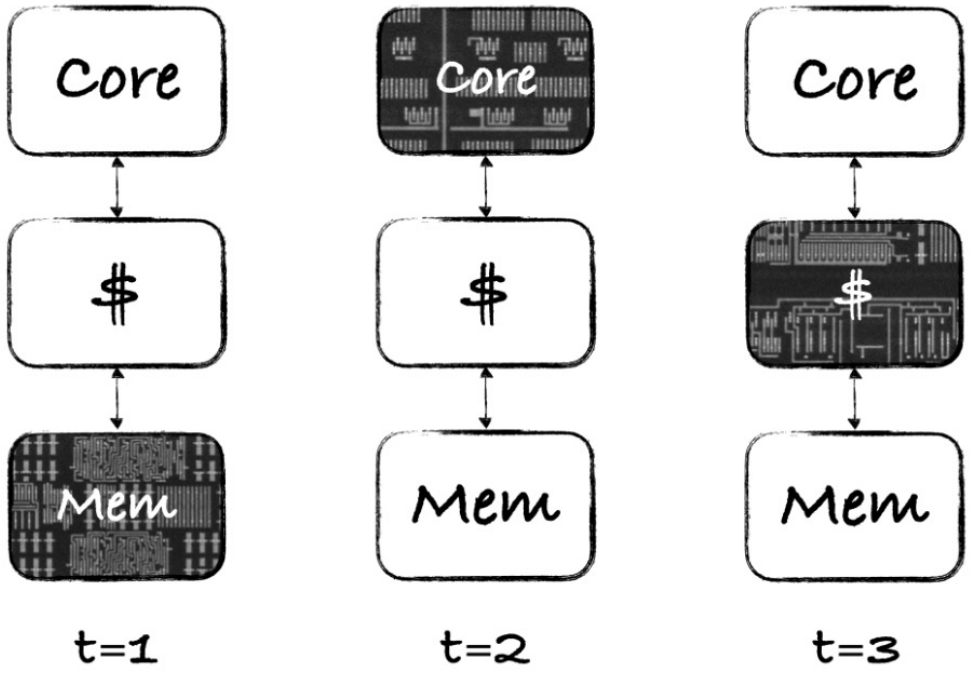
Multifidelity Simulation

- Physical simulation is often adapted to different levels of detail when more or less is required
  - E.g. Adaptive mesh refinement
- We want to harness this technique to speed up architectural simulation
- Phase detection lets us break up computation into simpler pieces
  - A DGEMM-heavy region should be easier to model than sparse computation



Related Work

- Online Model Swapping for Architectural Simulation [CF ’21]
  - Showed how to create a Markov model for each phase and swap during simulation
- Basic-block centric multifidelity methods
  - Simulate a BB only when it is first encountered
- Sampled Simulation
  - Uses offline clustering to find representative phases
- Statistical Sampling
  - Use sampling theory to estimate the performance from a sample of the program
- Design-space exploration
  - Simulate only a portion of the design-space, estimate the rest of the space using analytical or ML techniques

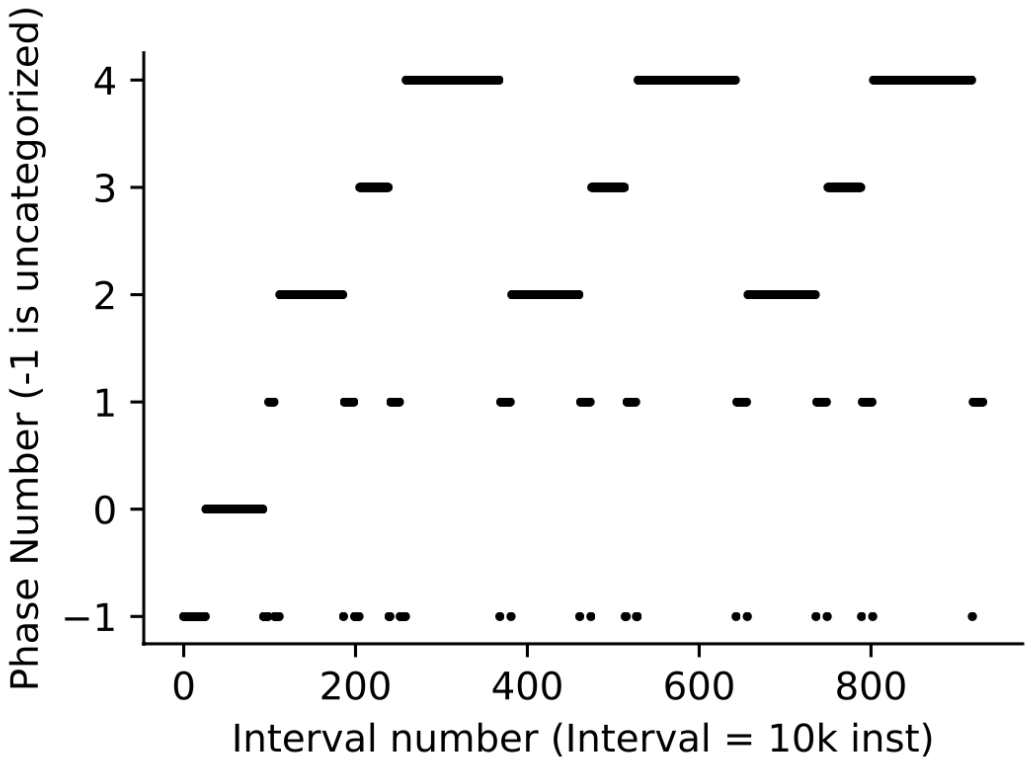


We want an online, component-centric approach

Georgia Tech

Phase Analysis Models

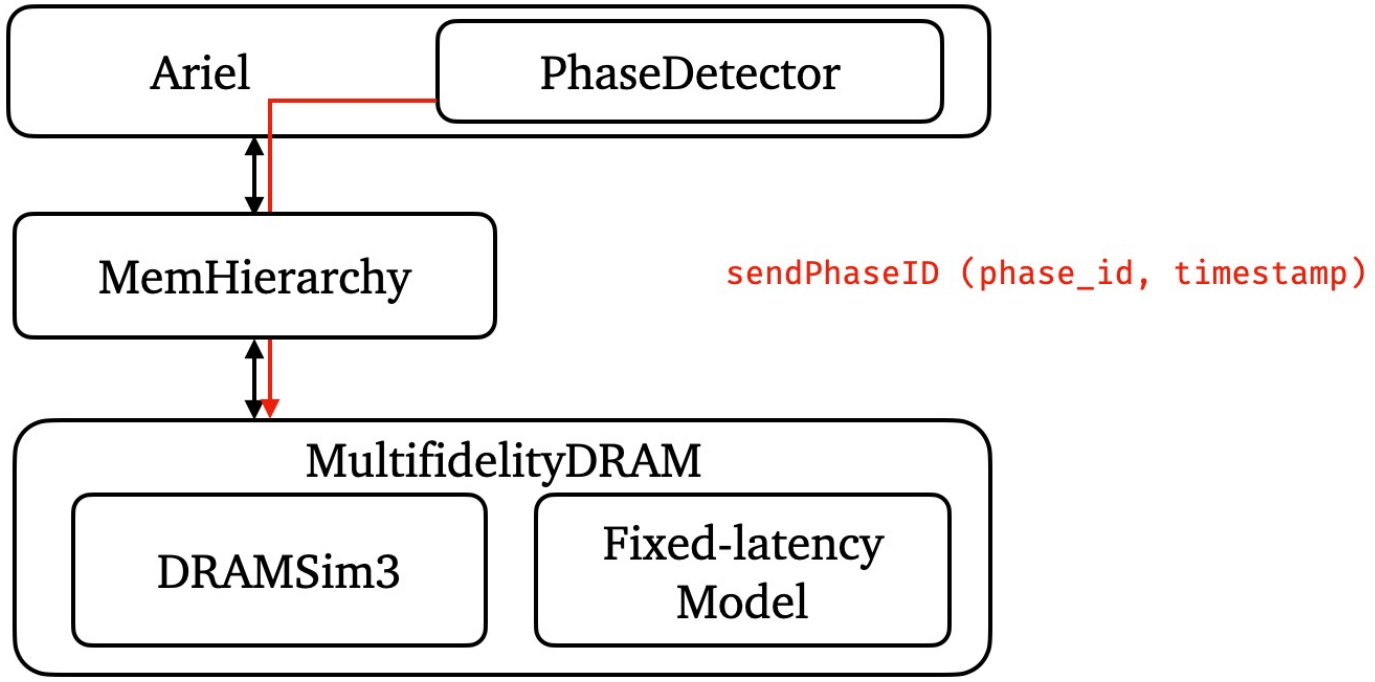
- We use an online, working set-based phase detector
  - Hash each instruction pointer into a bit vector
  - Compare bit vectors of each interval to find phases
- We extended DRAMSim3 to create a fixed-latency model for each phase
  - We hope to extend it to use other simple models
  - When a phase is encountered for a second time, we can use the average latency instead of simulating it



Georgia Tech

Integration of Mixed Fidelity DRAM with SST

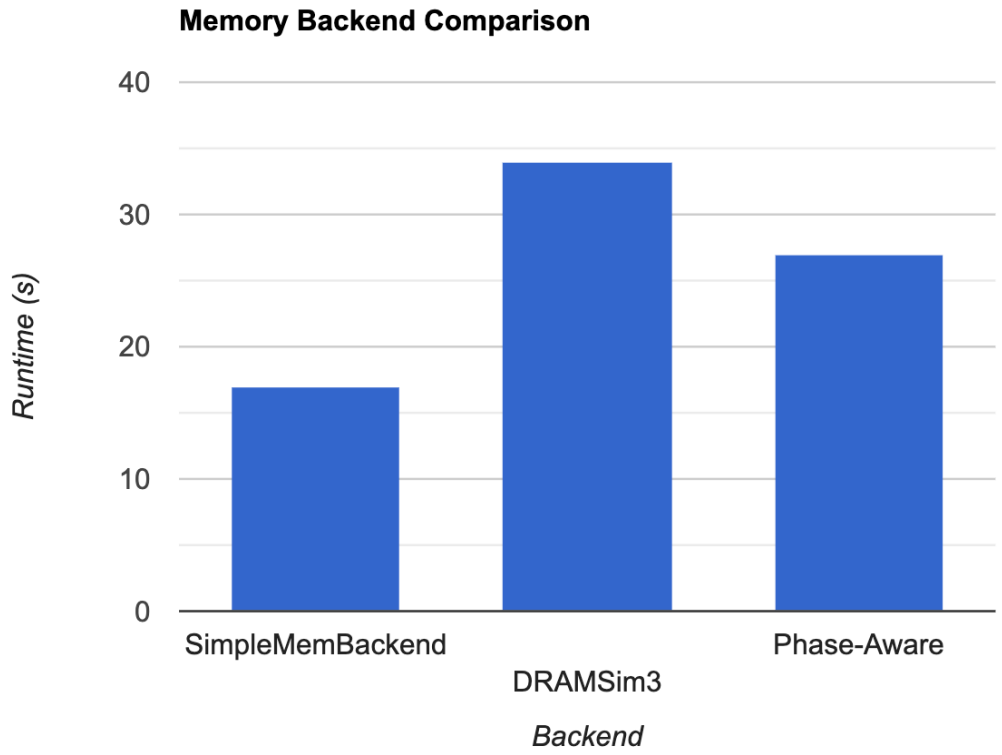
- Currently, the phase detector is tightly integrated with Ariel
- Ariel traces a running process and passes the memory references to memHierarchy, a memory system simulator
- Phase information is passed through memHierarchy to the memory backend, DRAMSim3
- We use DRAMSim3 to find the average latency for a phase the first time it is run, and create a fixed-latency model for each phase



Georgia Tech

Results and Next Steps

- Status
  - DRAMSim3 slows down the simulation by 2x vs. fixed-latency
  - Our phase-aware model has some overhead, and can't quite match this
  - Getting high accuracy will require some tuning
- Current work
  - Optimize and tune implementation
    - Phase parameters need to be chosen carefully
  - Error bounds for the surrogate models
    - Bootstrap method
  - SST interface for phase detection component



Simple hello-world simulation with small cache. Phase-aware represents the current best possible runtime

Georgia Tech