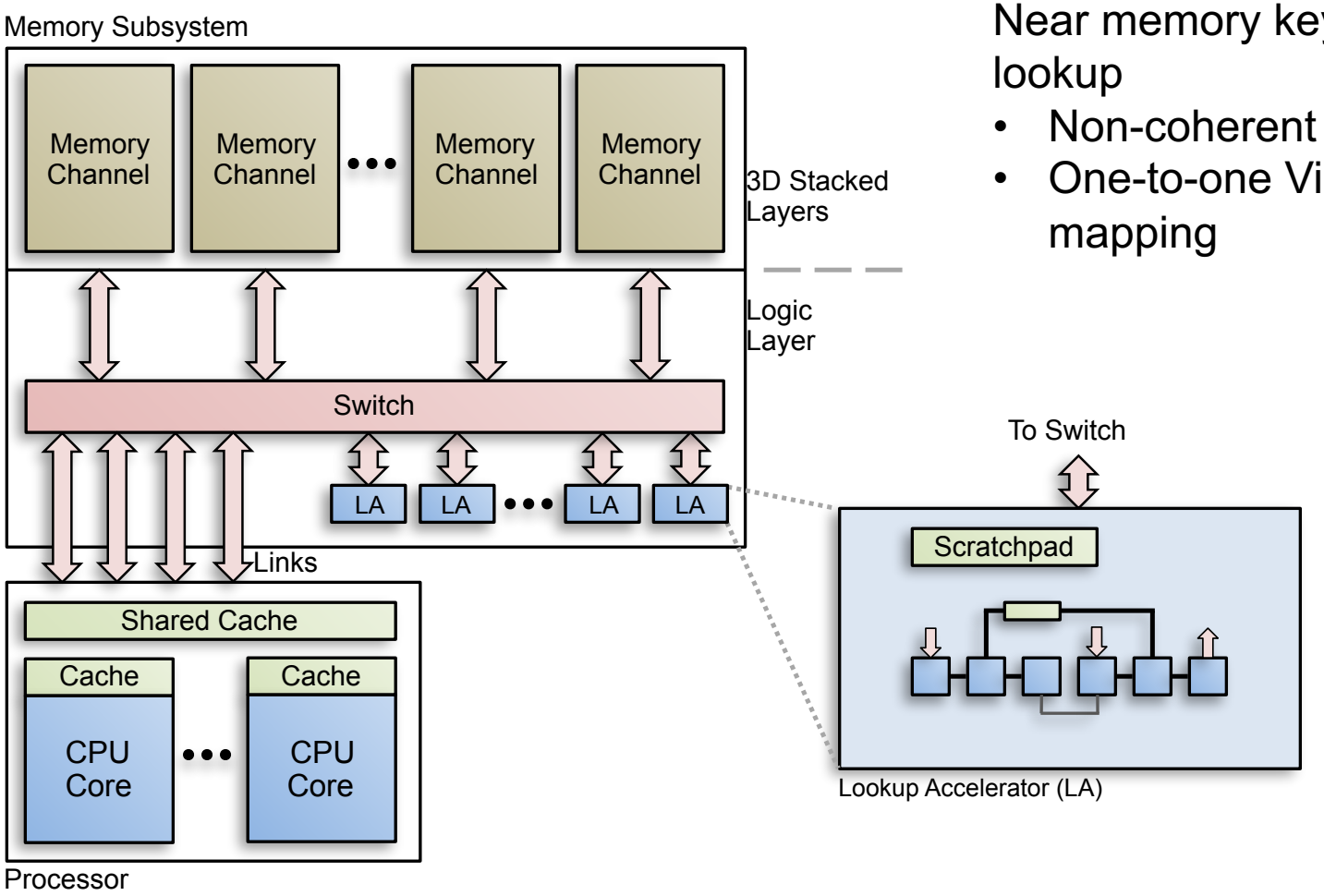
Using LiME+SST to evaluate a near memory accelerator

Joshua Landgraf*†, Scott Lloyd*, Maya Gokhale*
*LLNL †University of Texas at Austin

LLNL-PRES-741492
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

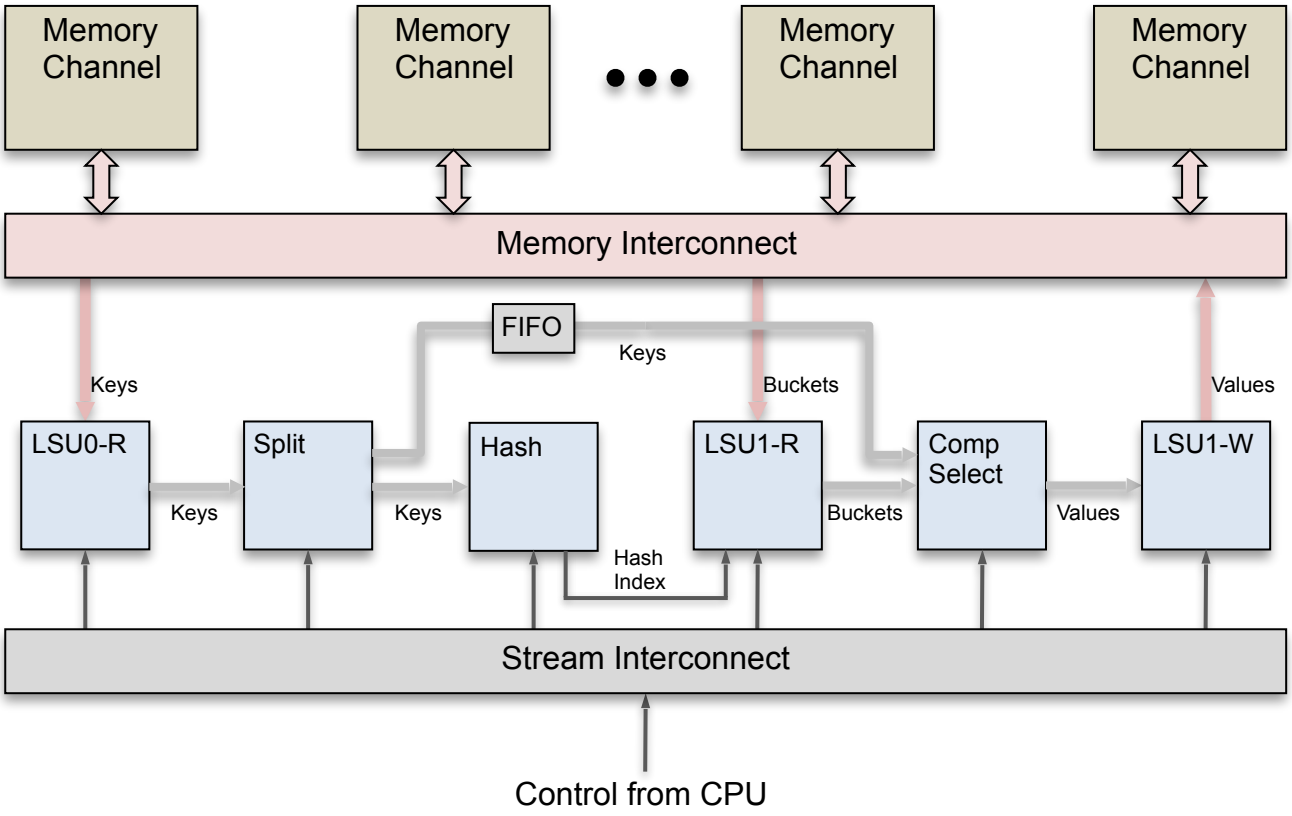
We started from an existing RTL design



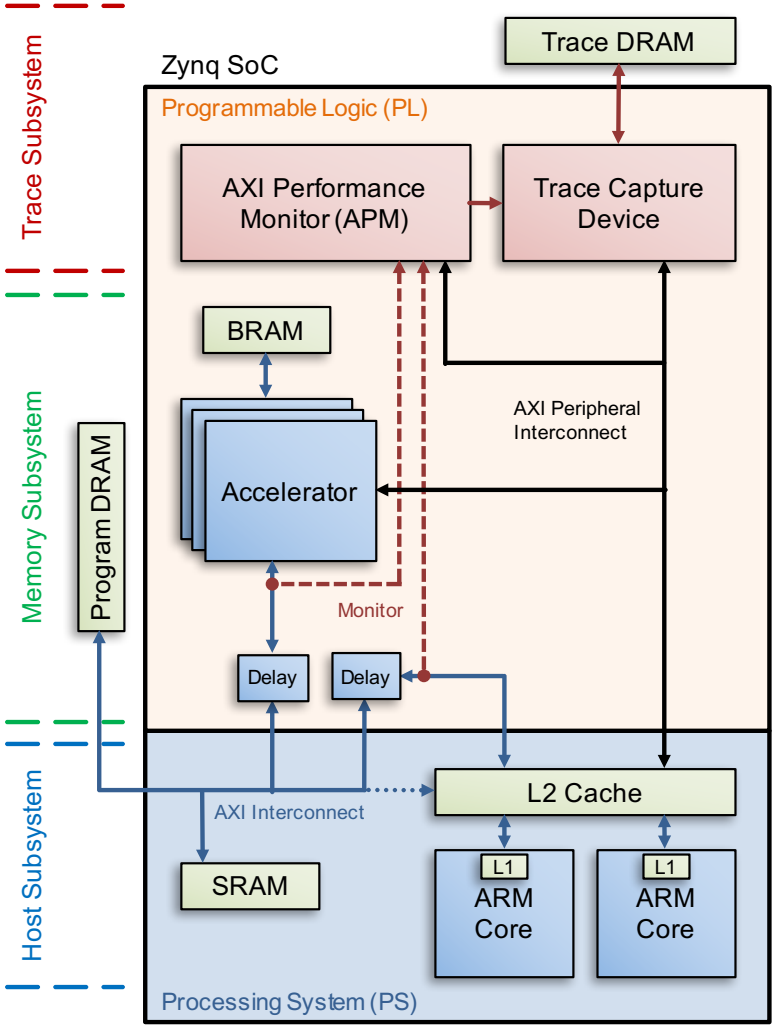
Near memory key/value store lookup

- Non-coherent
- One-to-one Virt to Phys mapping

Pipelined fixed function units, including gather/scatter engines



Logic in Memory Emulator (LiME)



- Advantages
  - Fast cycle-accurate emulation
  - Hard-core ARM CPU
  - Tunable memory latencies
    - Read / write handled separately
  - Full memory trace capture
- Limitations
  - Fixed CPU and cache hierarchy
  - Limited FPGA fabric
  - Memory bandwidth limit
  - Simple memory model – fixed latency – can't capture bank or switch conflicts

LiME + Simulation

- Capture memory access traces through LiME
- Use memory traces to determine model parameters
- Model accelerator, CPU, and LiME memory model in SST
- Use HMC-Sim to simulate a Hybrid Memory Cube (HMC)

HMC-Sim
HMC Memory Model

- Current versions implement the v2 HMC Specification
  - HMC-Sim 1.0 targets the original HMC spec
- Detailed HMC simulation
  - Vault packet routing
  - Bank conflicts
  - Memory access delays
    - Latency specified in cycles
    - Closed page policy
- Operates on packet granularity
  - FLIT size is implementation specific
  - Packet size does not affect routing or memory access latency

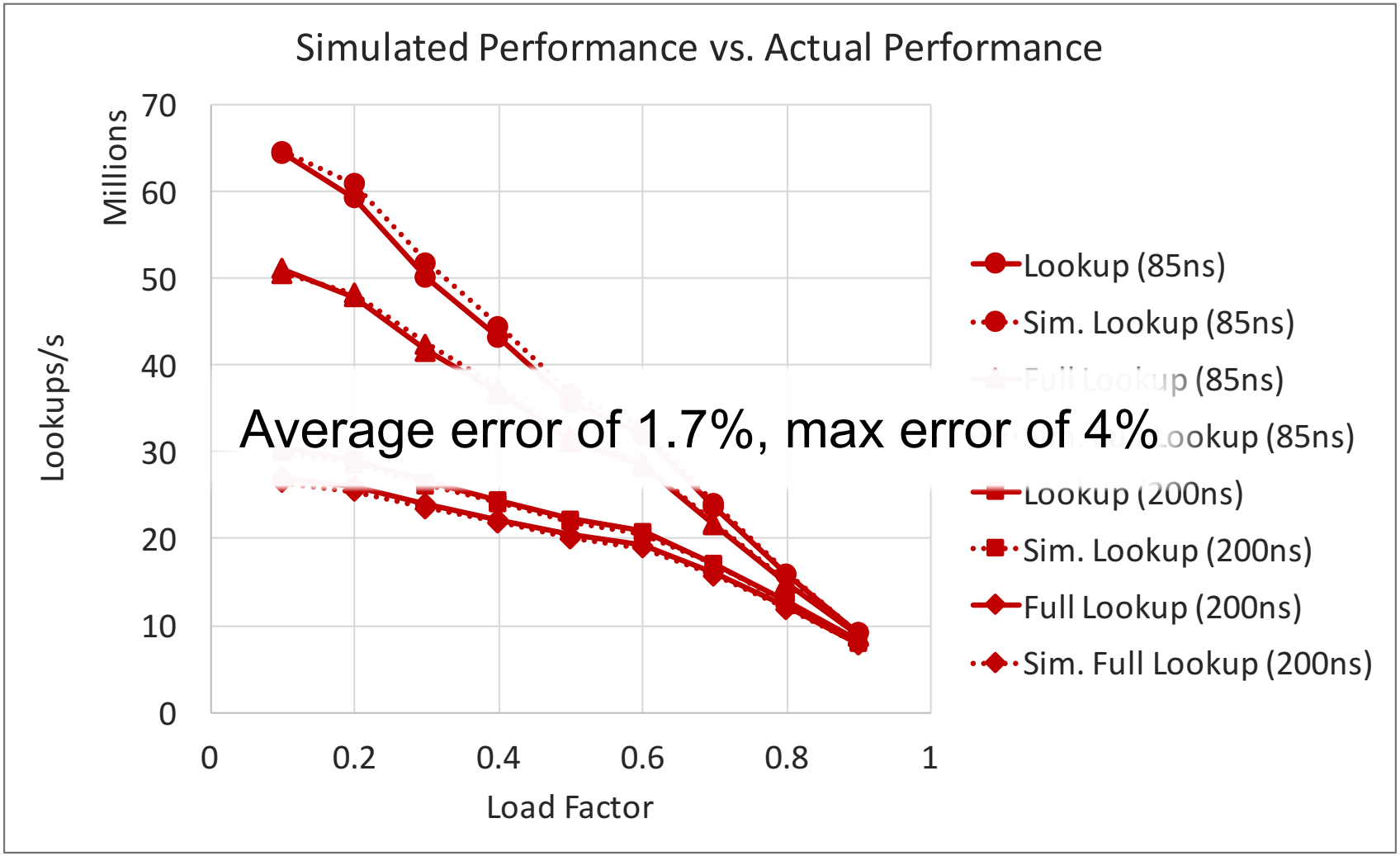
SST Advantages

- Can handle arbitrary number of CPUs and accelerators
- Easy to integrate new memory simulators in the future
- Flexible configuration system
  - Python interface allows for automation and easy scaling
- Powerful initialization system
  - Simplifies general simulation code

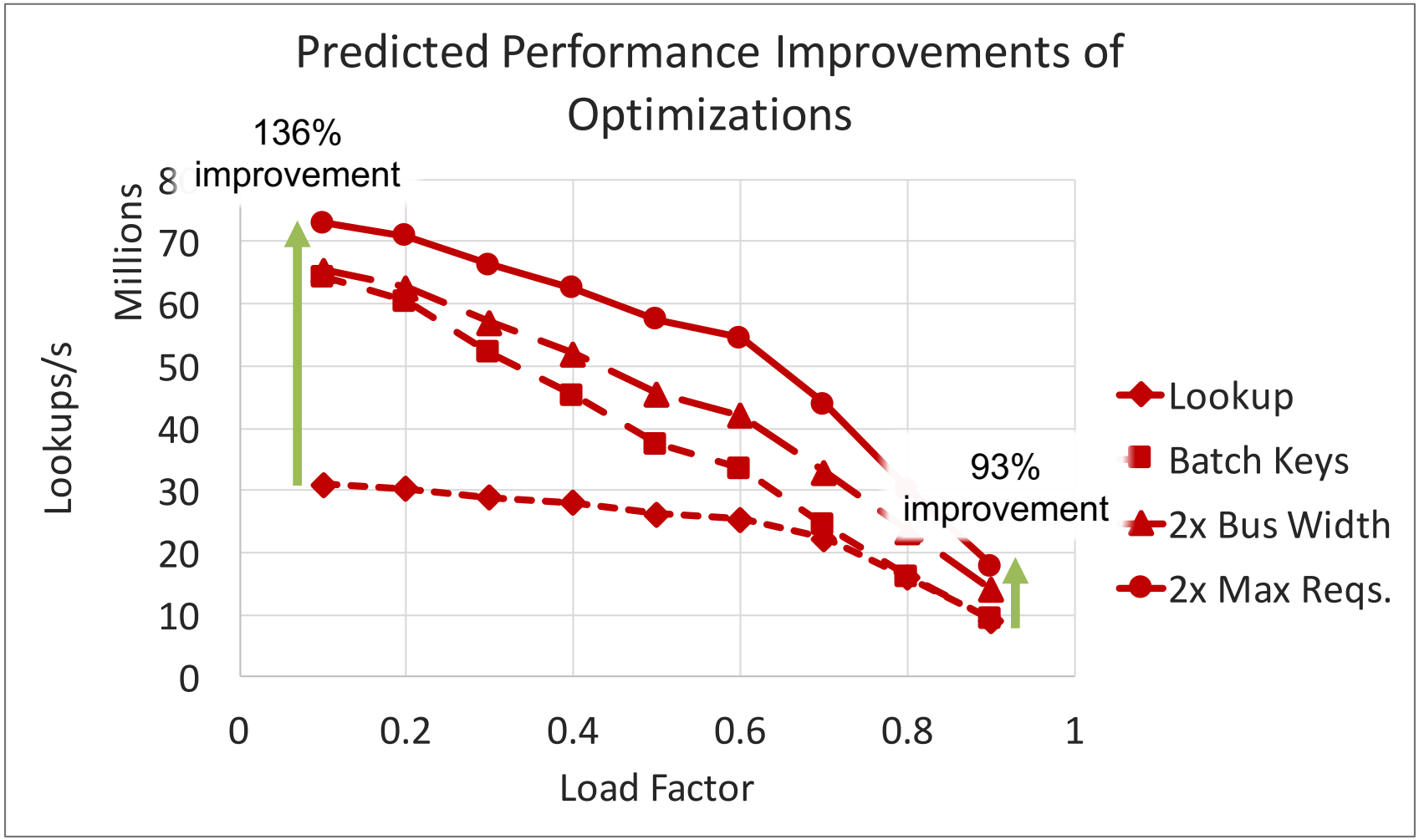
Experiments

- Verified simulated performance against LiME results
  - Used 85ns and 200ns latencies
  - LiME memory model
  - 10GB/s accelerator bandwidth
  - Reported results with and without CPU overhead
    - "Full Lookup" vs "Lookup"
- Predicted improvement from accelerator optimizations
  - Used 85ns latency with HMC-Sim
- Predicted scalability on a single HMC
  - Used 85ns latency with HMC-Sim

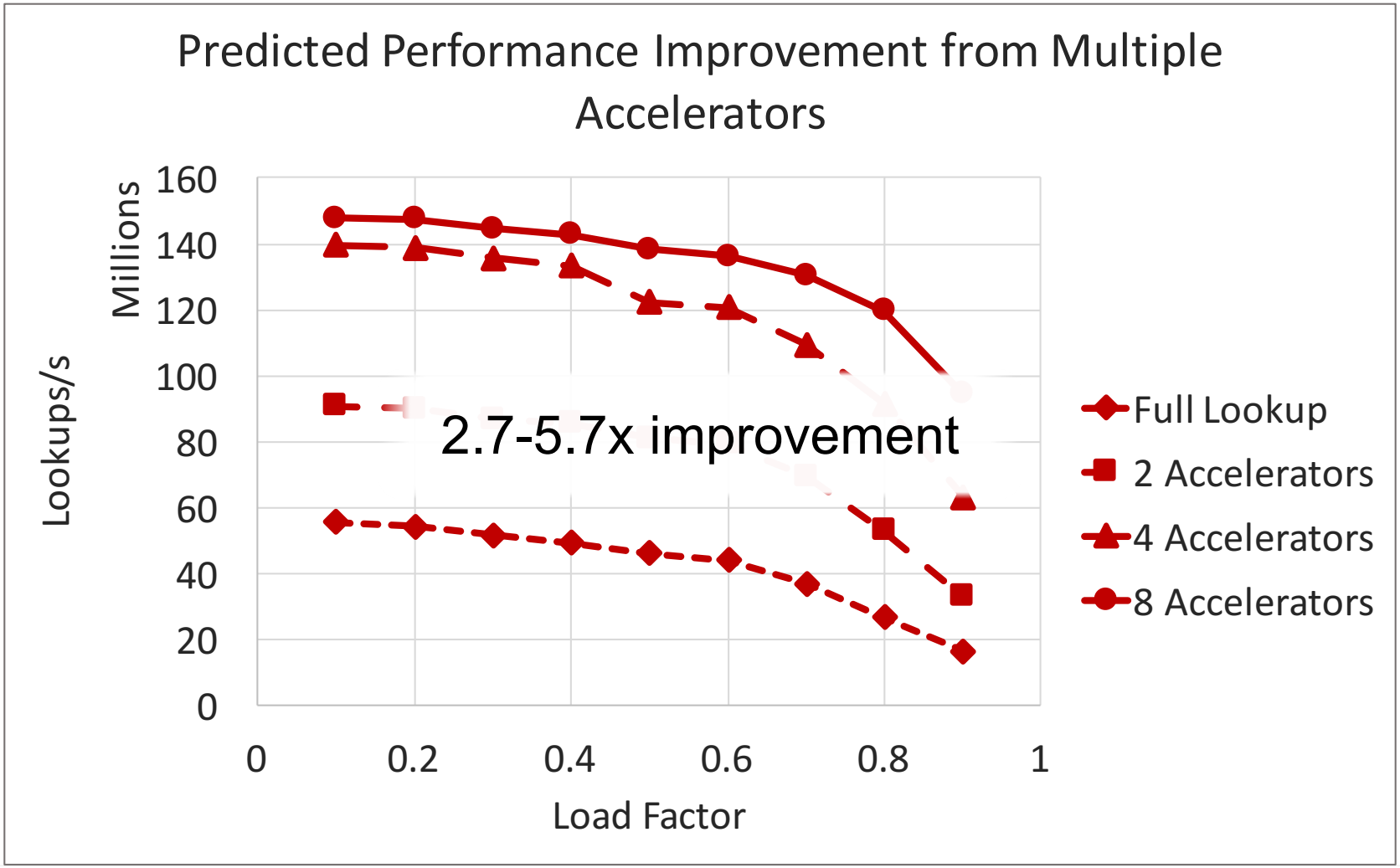
Verification Results



Optimization Predictions



Scaling Predictions



Summary and Conclusions

- Simulation was used to complement emulation
  - Simplified simulation model based on hardware design
  - Simulation model was verified against emulator
  - Iterative simulator model refinement achieved high accuracy
- Simulation enabled fast and scalable design exploration
  - Use of HMC-Sim enabled HMC-specific optimization of query stream
  - Scaling experiments were easily modeled in SST
  - Estimated 5-13x speedup from optimization + scaling
- Future work
  - Evaluate with different memory models
  - Explore additional optimizations
  - Expand to different types of accelerators

Lawrence Livermore
National Laboratory