CXL-ClusterSim: Modeling CXL-based Disaggregated Memory Cluster for Pooling and Sharing using gem5 and SST

Kaustav Goswami, Maryam Babaie, Hoa Nguyen, Venkatesh Akella and Jason Lowe-Power

UC DAVIS
UNIVERSITY OF CALIFORNIA

Introduction

Memory Stranding

"Despite the high cost of memory, a significant portion of data center memory for AI and HPC applications often goes unused due to a phenomenon known as memory stranding."

H. Li *et al.*, Pond: CXL-Based Memory Pooling Systems for Cloud Platforms, ASPLOS 23

Introduction

Memory Stranding

"Despite the high cost of memory, a significant portion of data center memory for AI and HPC applications often goes unused due to a phenomenon known as memory stranding."

Compute Express Link (CXL)

Introduction

Compute Express Link (CXL)

Disaggregated:

Decouples compute and memory.

K. Lim *et al.*, System-level implications of disaggregated memory, HPCA 2012

Introduction

Compute Express Link (CXL)

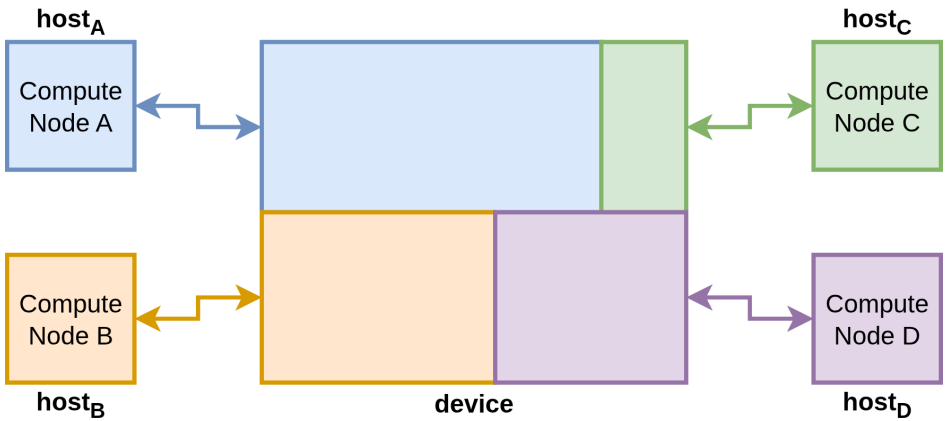
Disaggregated:

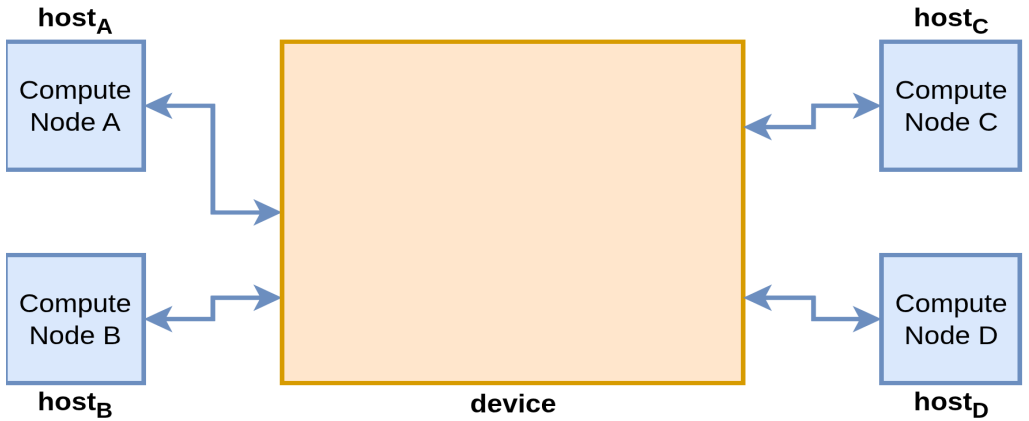
- Decouples compute and memory.
- Hosts are independent systems with their own OS and hardware.

Introduction

Compute Express Link (CXL)

Supports **pooling** and **sharing** memory.



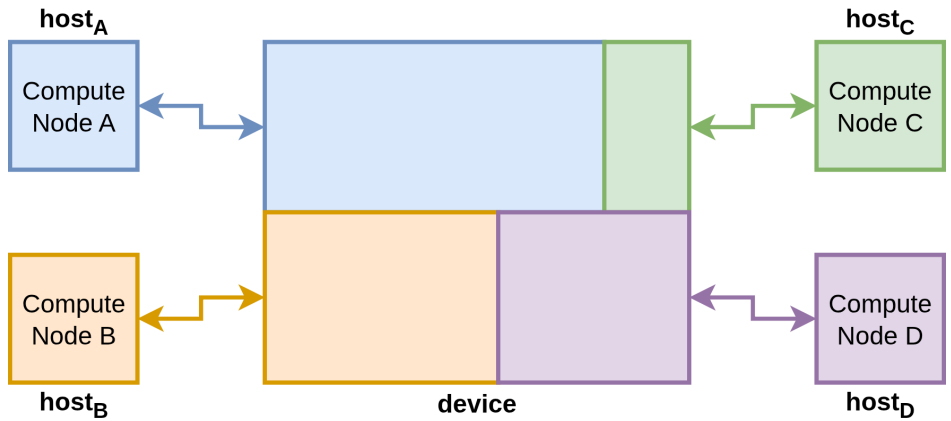


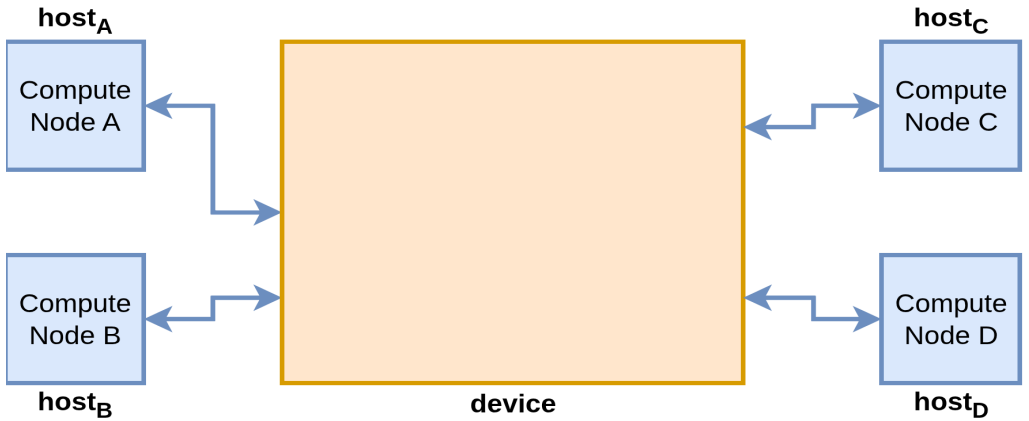
Introduction

Compute Express Link (CXL)

We need a framework to simulate these!

Supports **pooling** and **sharing** memory.





Introduction

Compute Express Link (CXL)

We need a framework to simulate these!

Supports **pooling** and **sharing** memory.

How can we simulate several hosts simultaneously?

Motivation

Simulating a Full-System CXL cluster

FPGA: high fidelity but cannot be scaled

gem5: high fidelity but single-threaded

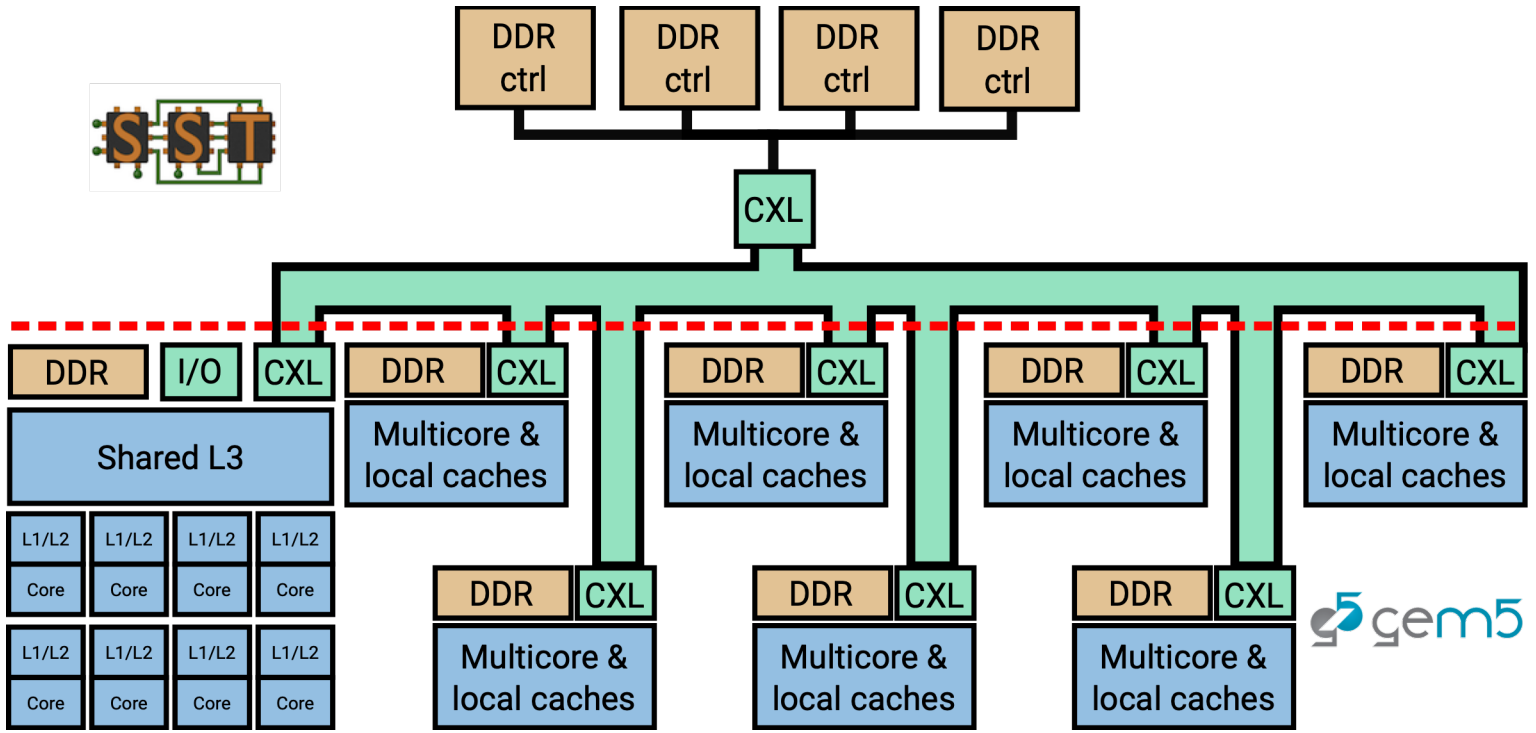
SST: parallelize and network models

Prior gem5-SST bridge?

J. Lowe-Power *et al.*, The gem5 simulator: Version 20.0+, ArXiv 2020
A. F. Rodrigues *et al.*, The structural simulation toolkit, SIGMETRICS 2011
M. Hsieh *et al.*, SST + gem5 = a scalable simulation infrastructure for high performance computing, SIMUTOOLS 2012
H. Nguyen *et al.*, gem5/sst integration 2021: Scaling full-system simulations.

Enter CXL-ClusterSim

Enter CXL-ClusterSim



J. Lowe-Power *et al.*, The gem5 simulator: Version 20.0+, ArXiv 2020
A. F. Rodrigues *et al.*, The structural simulation toolkit, SIGMETRICS 2011
M. Hsieh *et al.*, SST + gem5 = a scalable simulation infrastructure for high performance computing, SIMUTOOLS 2012
H. Nguyen *et al.*, gem5/sst integration 2021: Scaling full-system simulations.

Challenges

To boot an unmodified OS

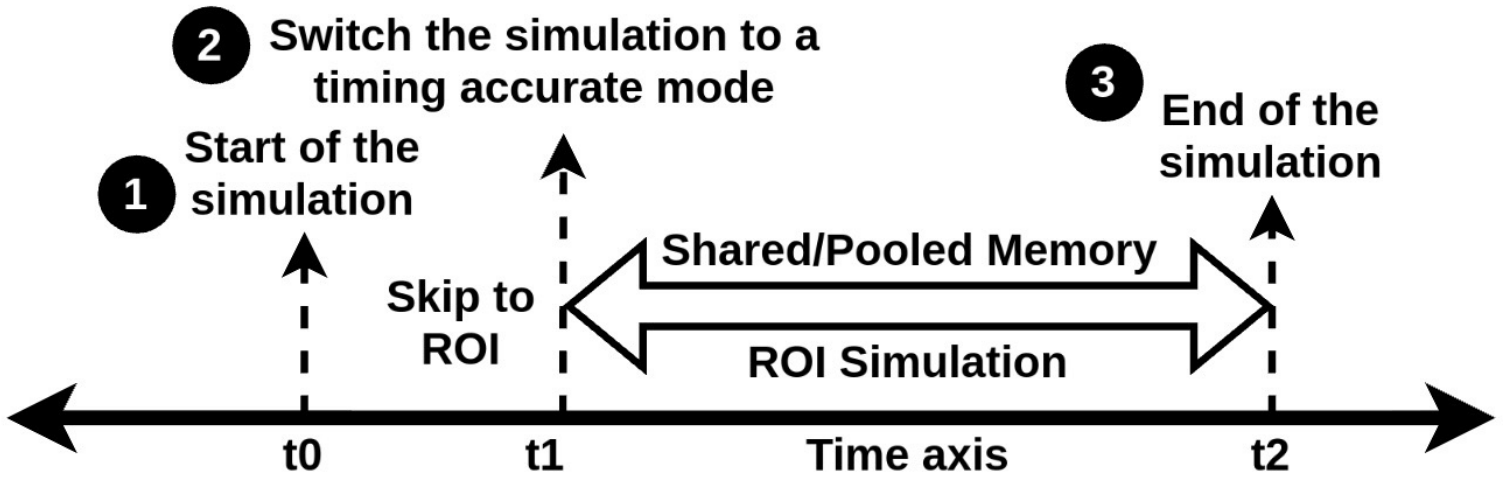
- NUMA and DAX

Access memory across MPI ranks

- Checkpointing and functional accesses

Challenges

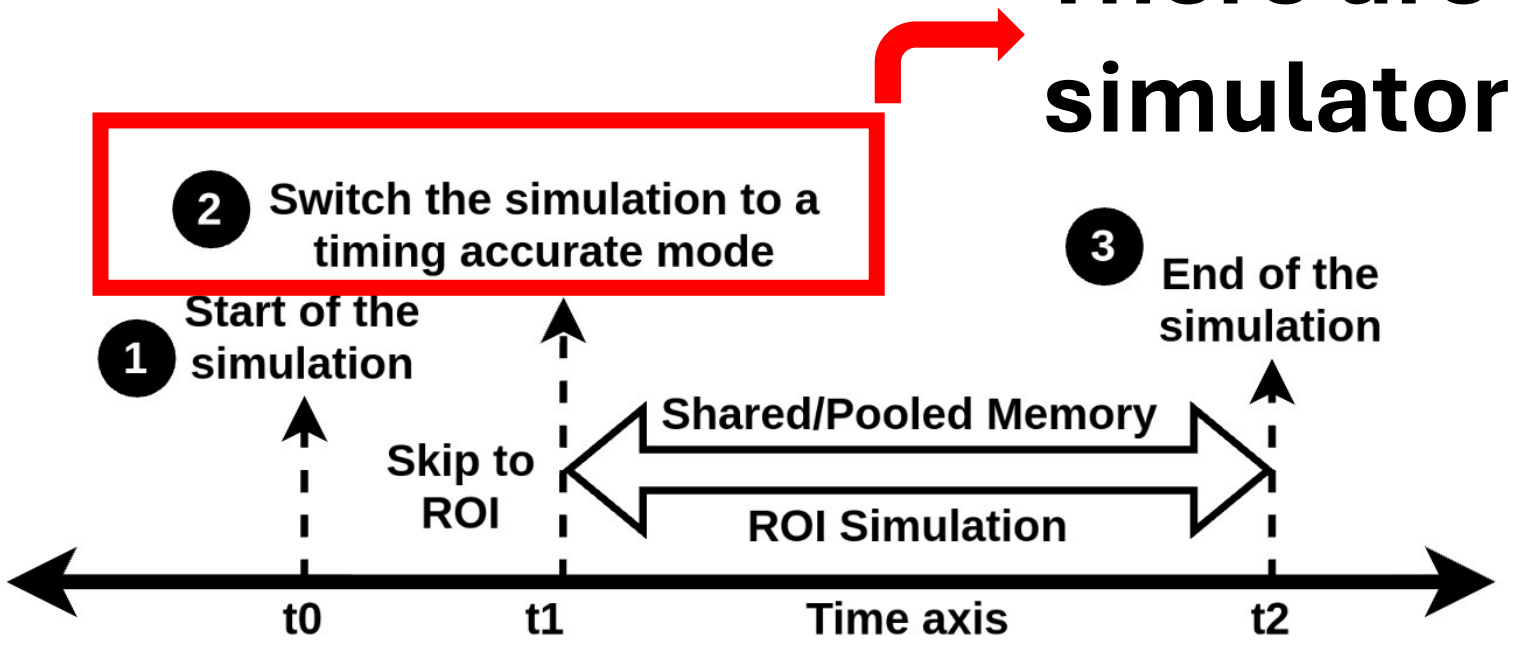
Ideal simulation timeline



Challenges

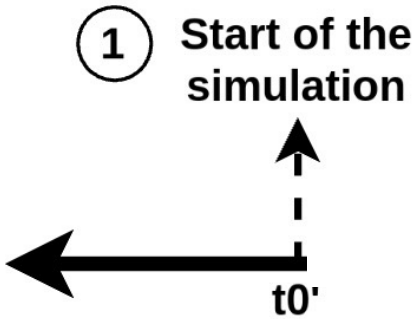
Ideal simulation timeline

There are two simulators now!



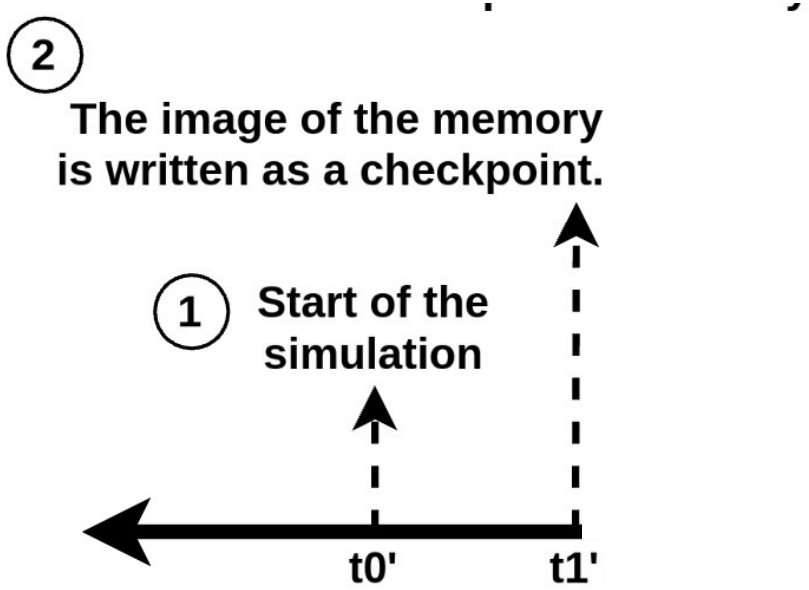
Solution

Decouple initialization from the ROI



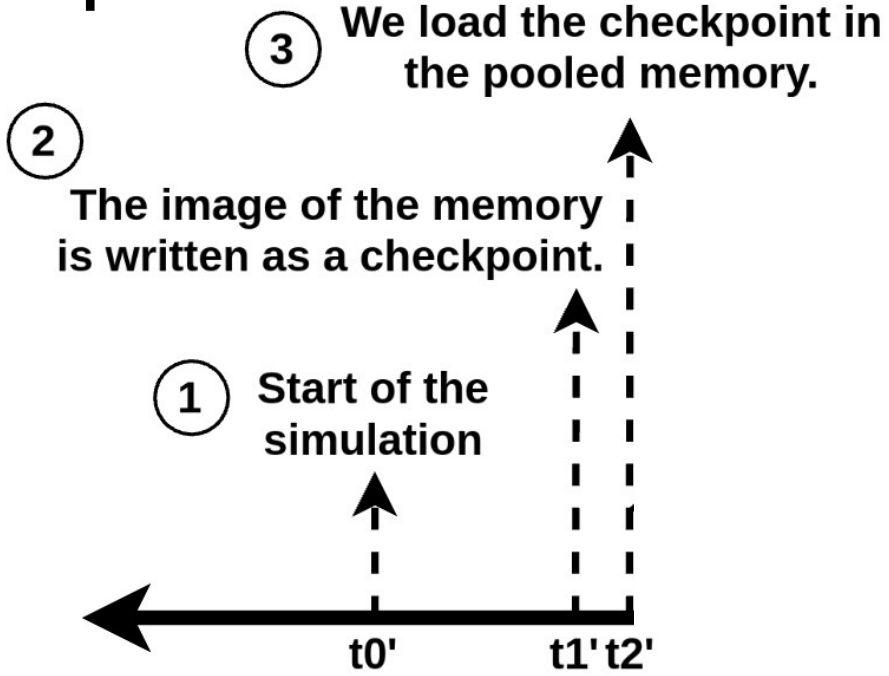
Solution

Decouple initialization from the ROI



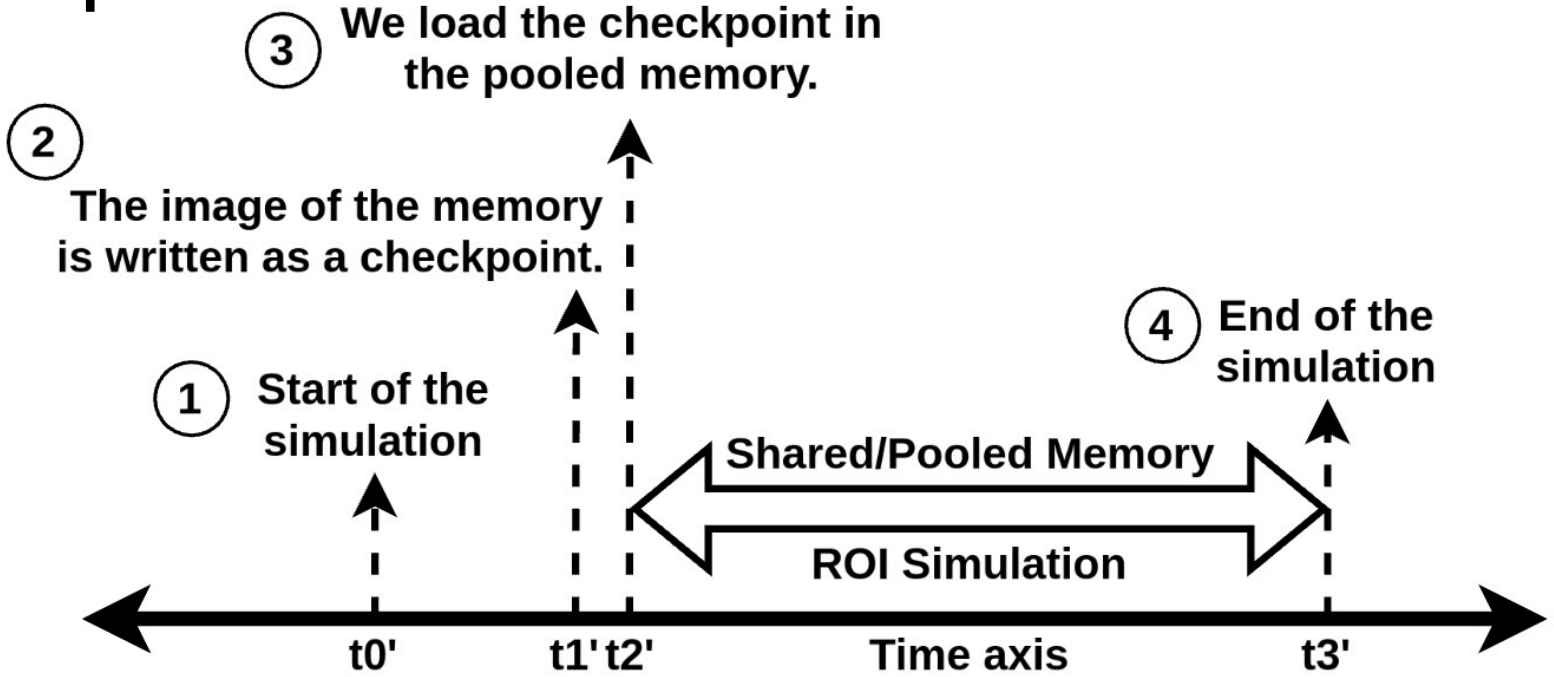
Solution

Decouple initialization from the ROI



Solution

Decouple initialization from the ROI

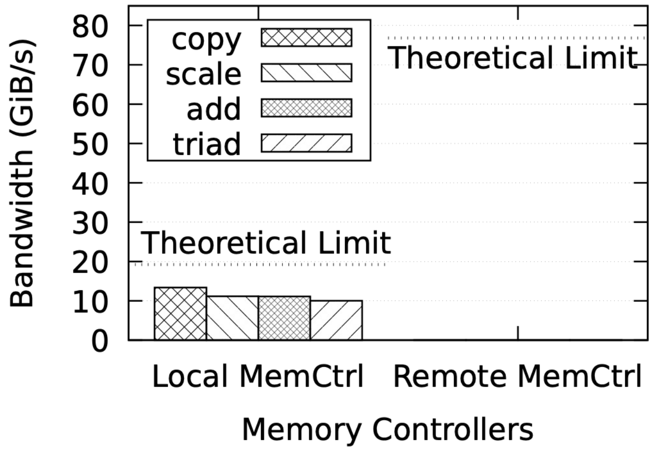


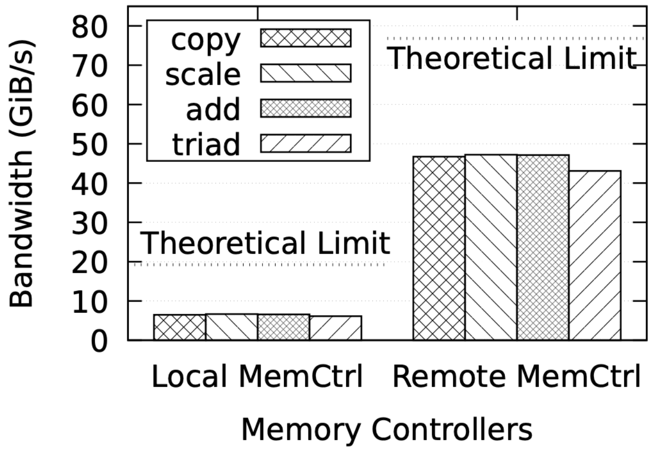
Results

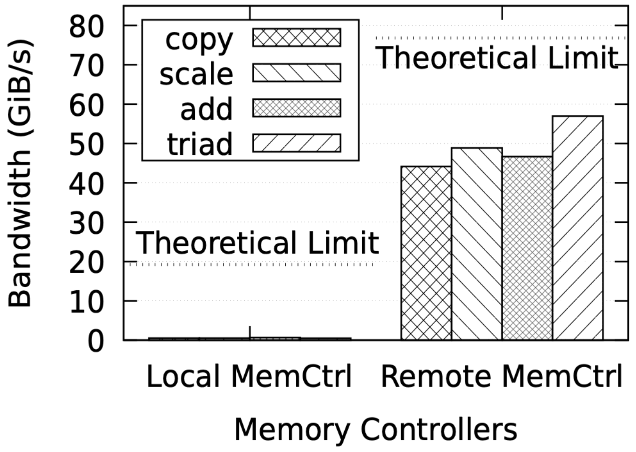
ARM Ubuntu 22.04 with *numactl*

Validated.

Bandwidth is within 0.99% of the baseline

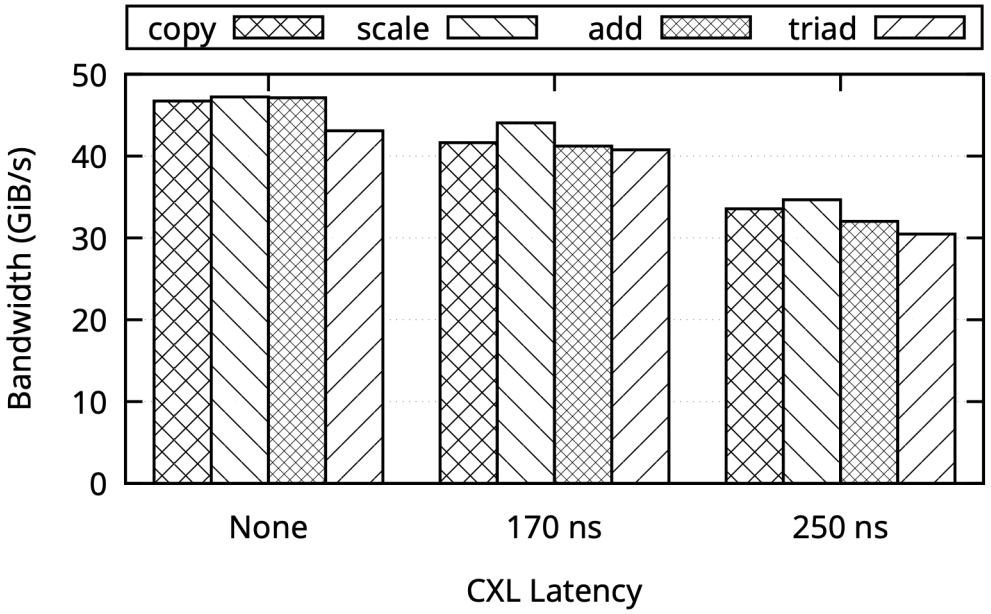




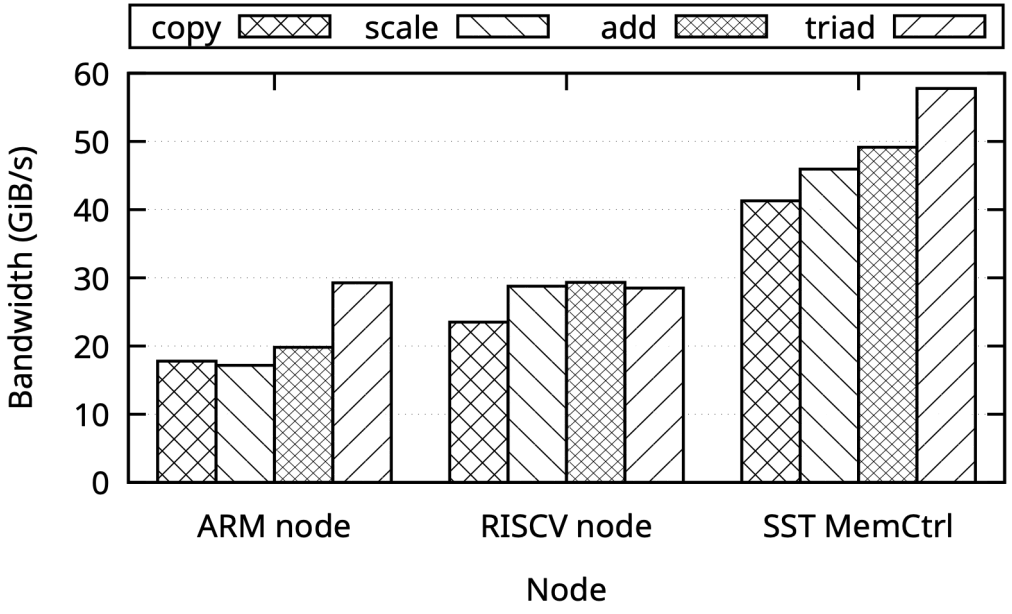


Results

CXL Latency

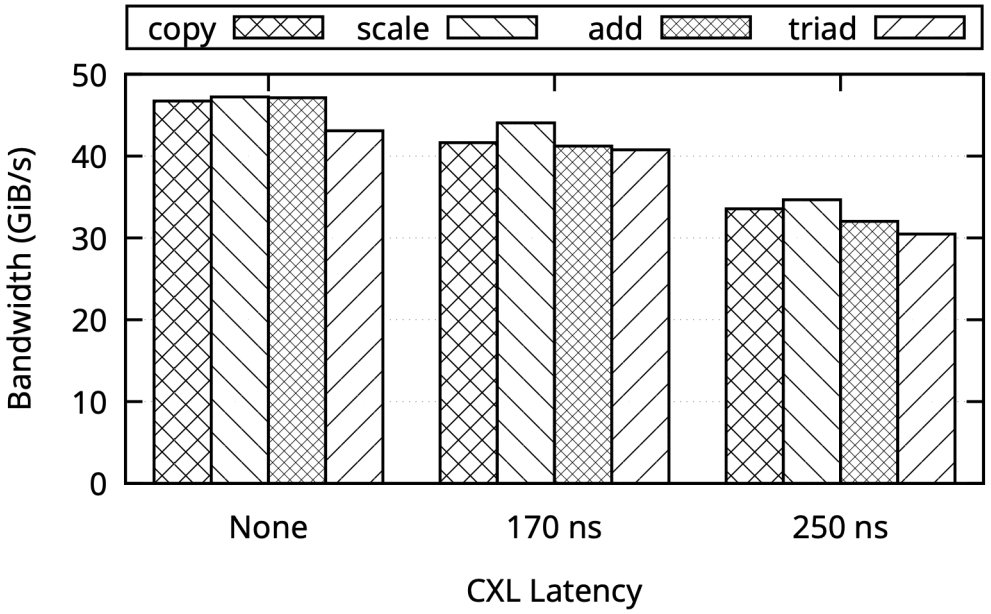


Devices are ISA-agnostic

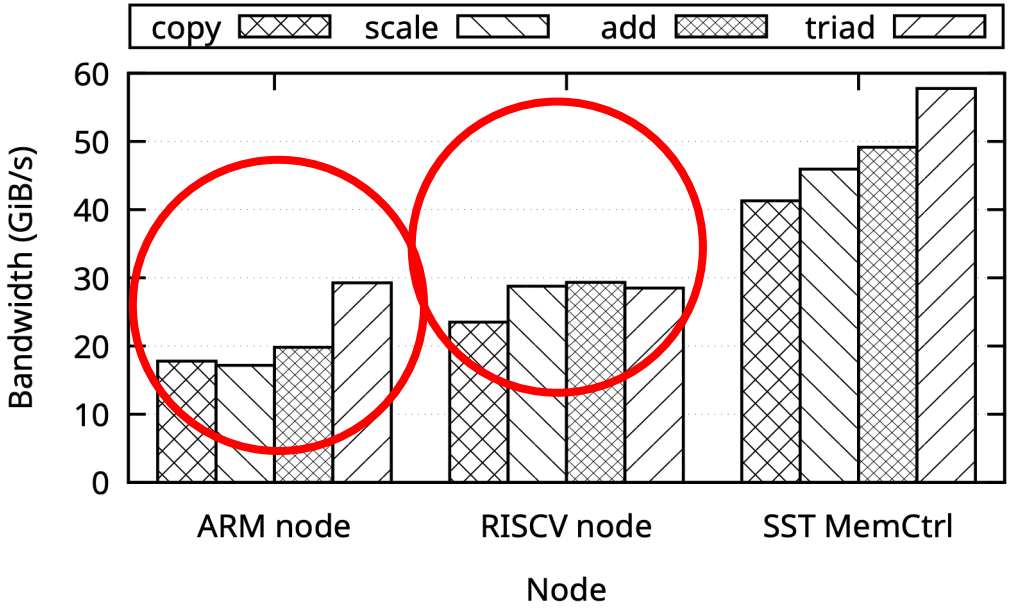


Results

CXL Latency



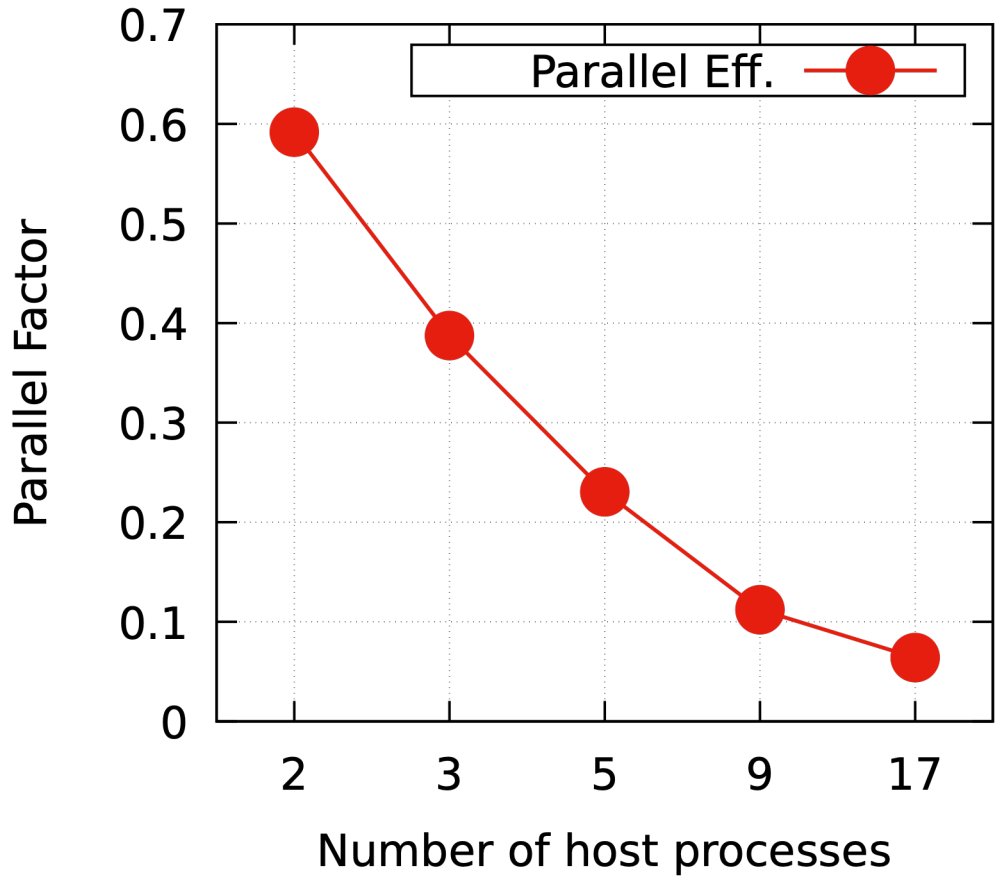
Devices are ISA-agnostic



Simulation Statistics

Parallel Efficiency (PE)

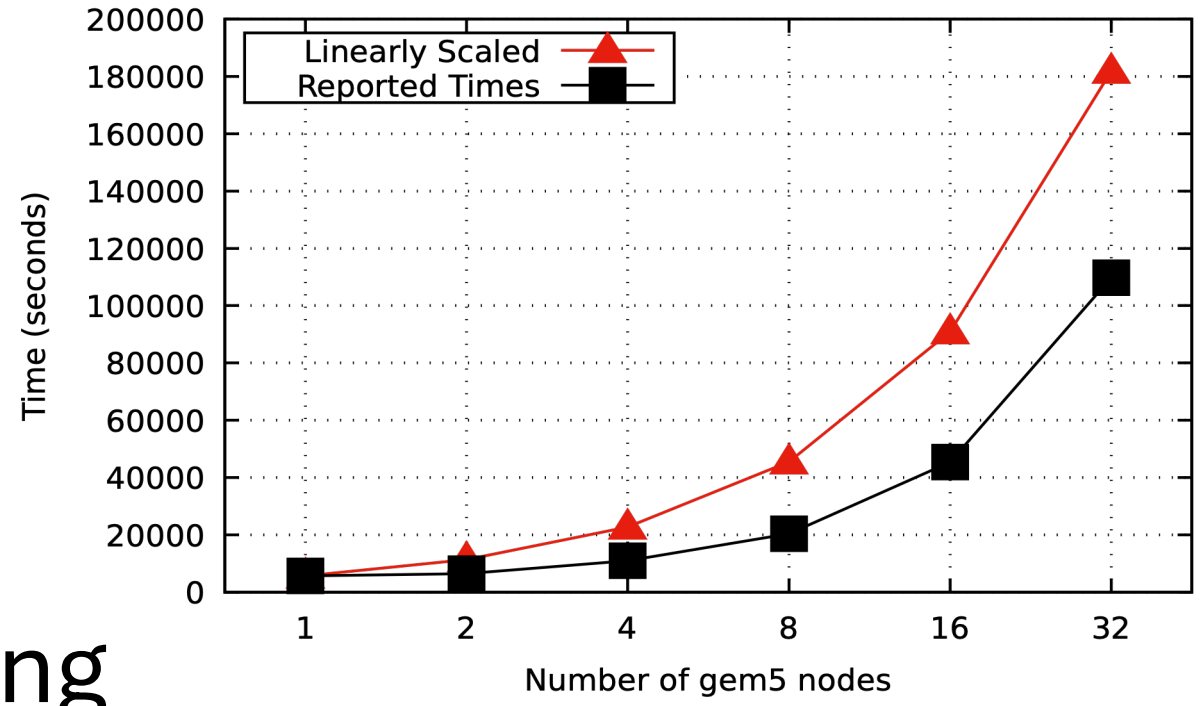
$$PE = \frac{1}{NUM_{processes}} \times \frac{TIME_{gem5\ only}}{TIME_{CXL\text{-}ClusterSim}}$$



Simulation Statistics

Parallel Efficiency (PE)

$$PE = \frac{1}{NUM_{processes}} \times \frac{TIME_{gem5\ only}}{TIME_{CXL\text{-}ClusterSim}}$$

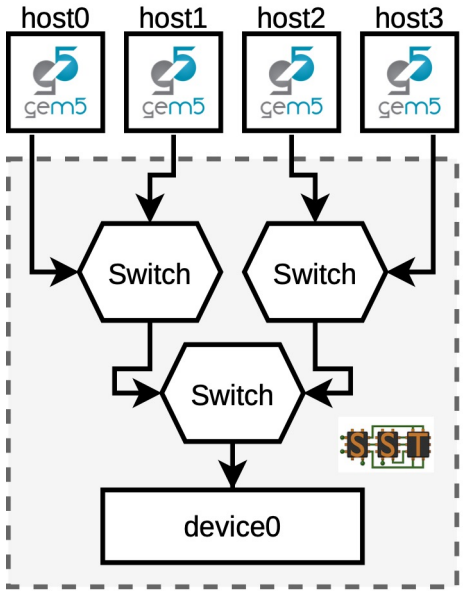


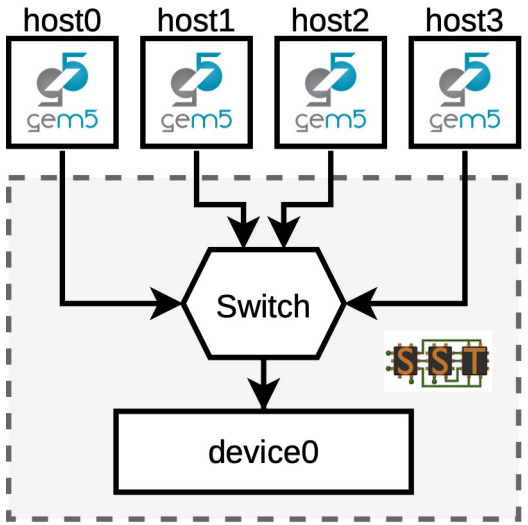
Faster than linear scaling

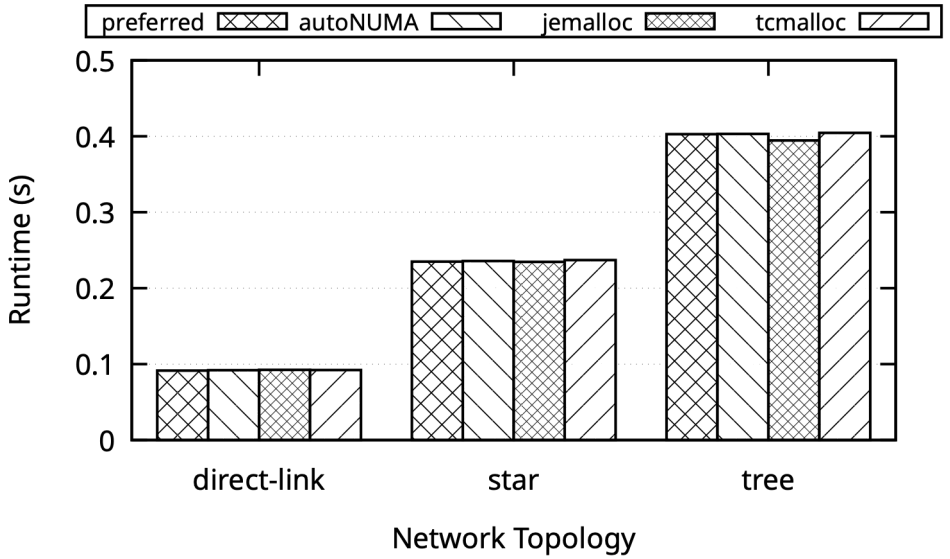
Network Topology Exploration

We enable network topology exploration

We show direct links, star and tree







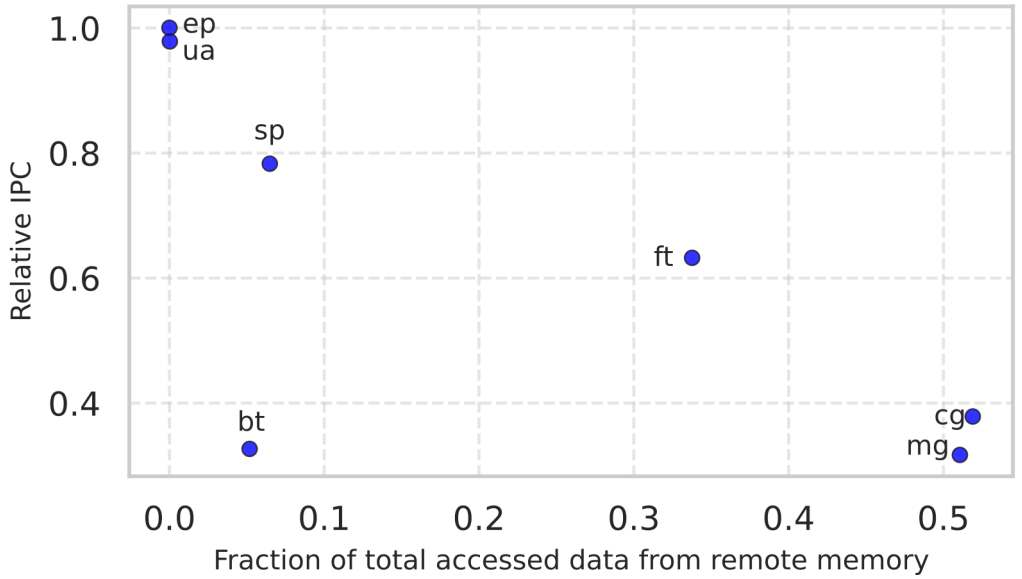
Memory Pooling

NAS Parallel Benchmarks

Baseline is a single system with 128 GiB of local memory.

CXL-ClusterSim hosts (ARM) have 8 GiB of local memory.

NUMA --*preferred*.



D. H. Bailey *et al.*, The NAS Parallel Benchmarks, J. of Supercomputing Applications 1991

Memory Sharing

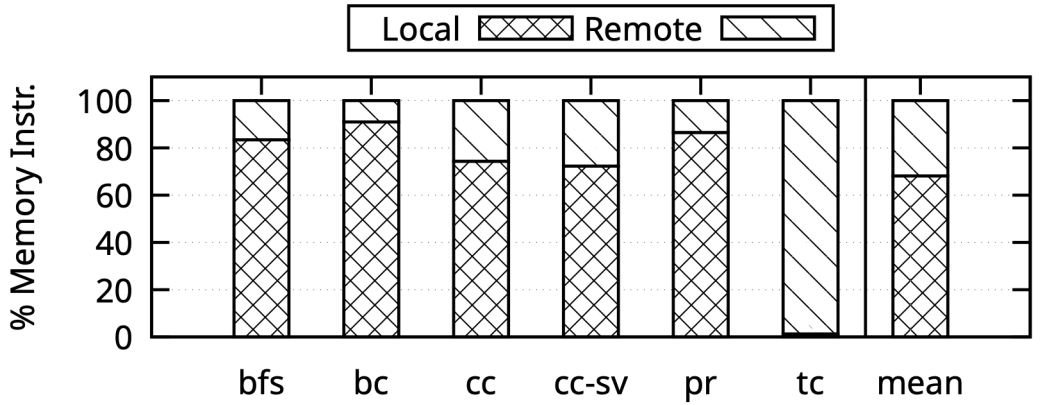
GAP Benchmarks

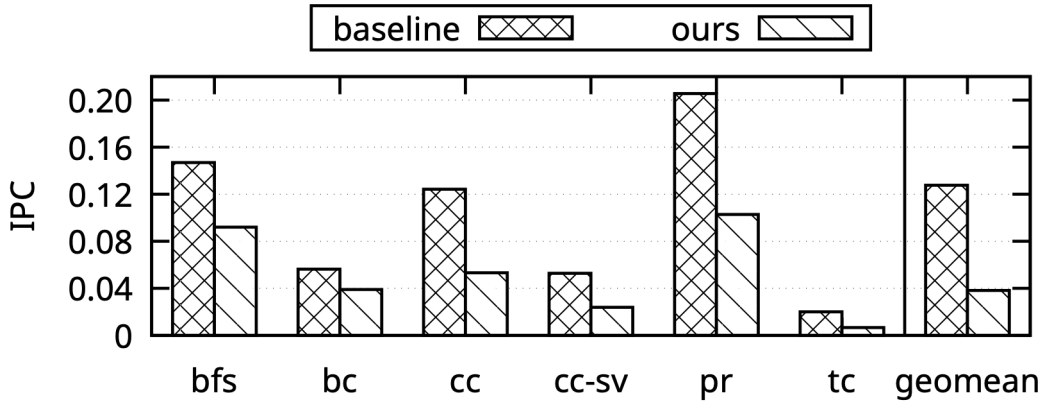
Kernels share the graph

Each host (x86) runs a kernel

New decoupled workloads

- Back-Invalidate support
- Explore memory consistency
- CXL fabric latency





S. Beamer *et al.*, The gap benchmark suite, ArXiv 2017

Using the Source Code

gem5 SimObjects or SST Component

Describe the cluster as a structured JSON

```
{
  "0": {
    "cpus": { "ff-core": "kvm", "roi-core": "o3"},
    "cache": "l1l2l3",
    "local-memory": "2GiB",
    "remote-memory": { "start": "0x100000000", "end": "0x180000000" }
  }
}
```

Configure the CXL fabric in SST/Python

Simulate!

Conclusion

In this work, we presented CXL-ClusterSim

- CXL-ClusterSim allows cycle-level simulation of a CXL cluster with several hosts and devices.
- Hosts are modeled in gem5 and devices are modeled in SST.
- We show both memory pooling and sharing.

We plan on improving the simulation speed and add support for CXL.io and CXL.cache in the future.

The code is released with an open-source license available at: https://github.com/darchr/cxl-clustersim

Thank You.

Full paper and source available at: