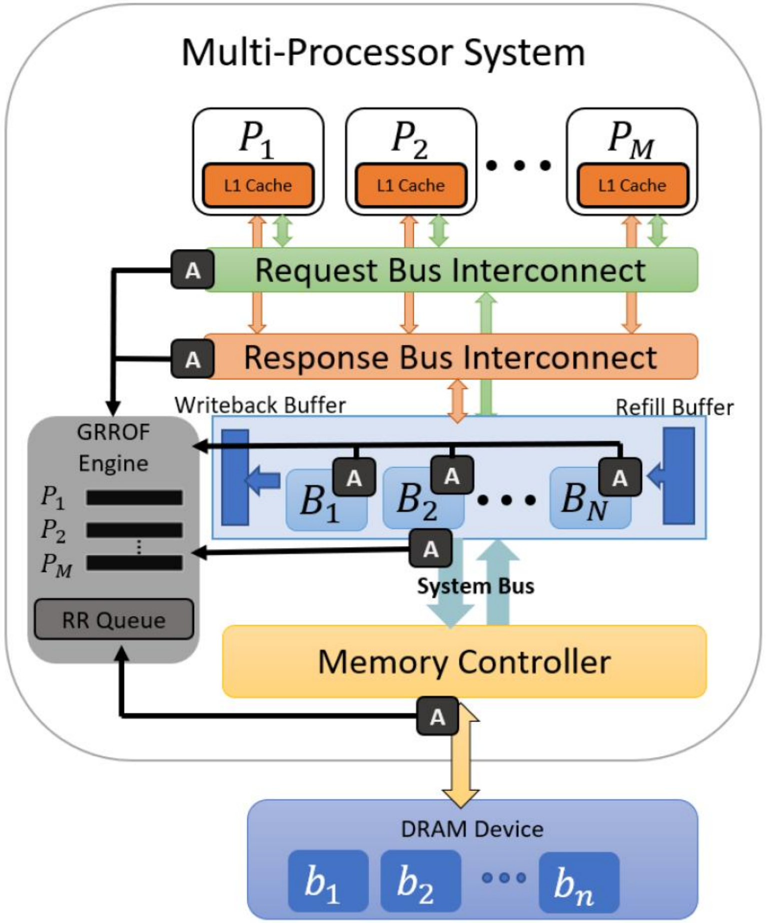
Octopus: A Gem5-Integrated Rapid Prototyping for Resource Contention Measurement and Control

UNIVERSITY OF WATERLOO

Yuying Lai
Guotong Miao
Shorouk Abdelhalim
Mohamed Hossam
Yazi Chen
Rodolfo Pellizzoni
Mohamed Hassan

McMaster University

- Modern memory systems are complex
  - Tens of features/optimizations
  - Several resources with parallelism and reorderings
- Resource arbitration is done using local information only with no request “global” view/semantics



Coordinating Memory Resources

Coordinating Memory Resources

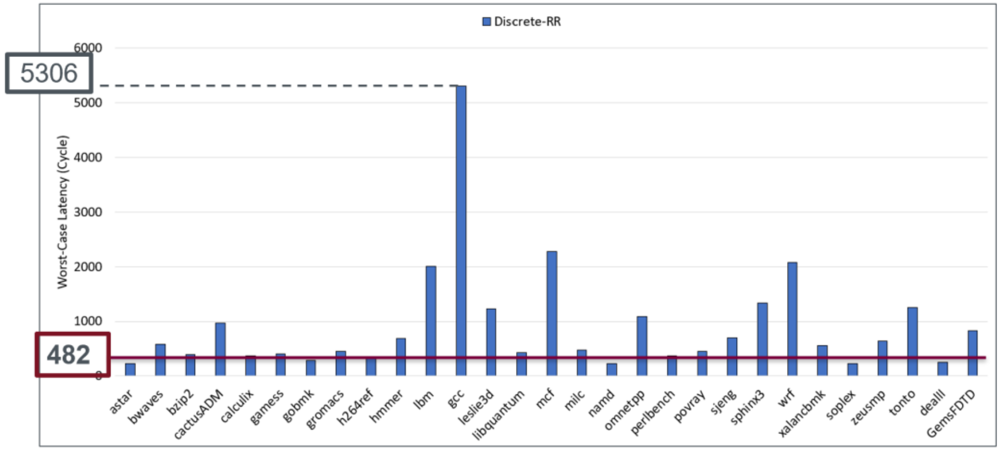
- Modern memory systems are complex
  - Tens of features/optimizations
  - Several resources with parallelism and reorderings
- Resource arbitration is done using local information only with no request “global” view/semantics

- Predictable resource arbitration and then sum things up is unsafe

Memory System of a Multi-Core Platform

WCL of the oldest requests (**Discrete-RR**)

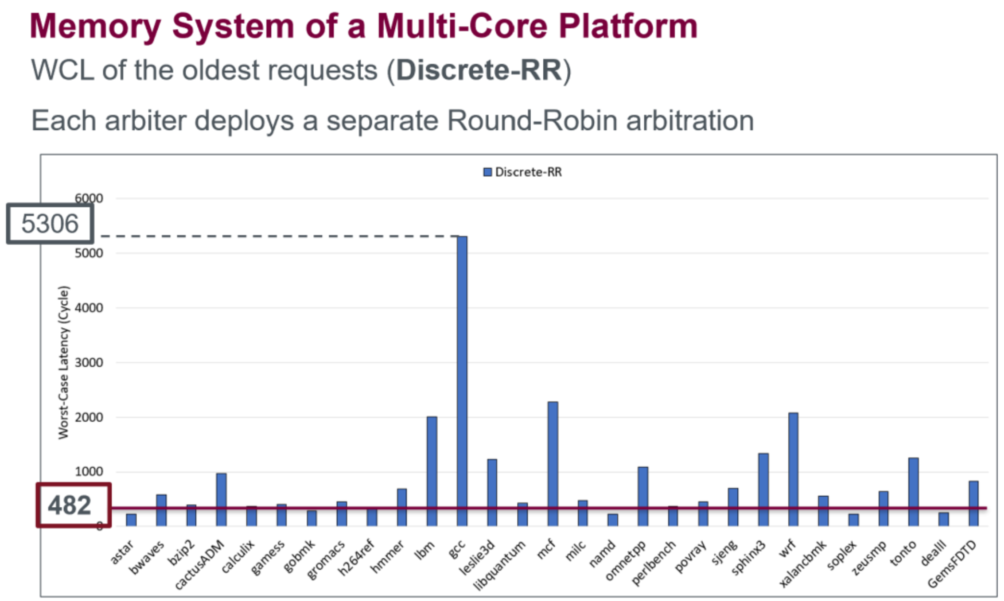
Each arbiter deploys a separate Round-Robin arbitration



Coordinating Memory Re

- Modern memory systems are complex
  - Tens of features/optimizations
  - Several resources with parallelism and reorderings
- Resource arbitration is done using local information only with no request “global” view/semantics

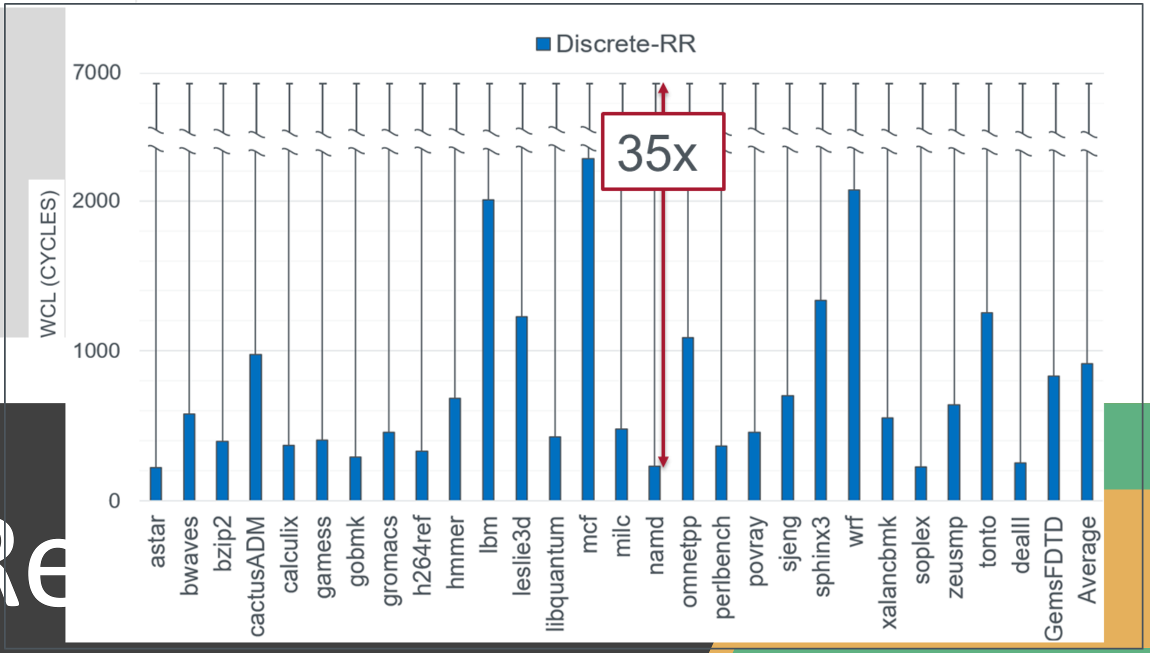
- Predictable resource arbitration and then sum things up is unsafe
  - Solution for this unsafety is drastically pessimistic

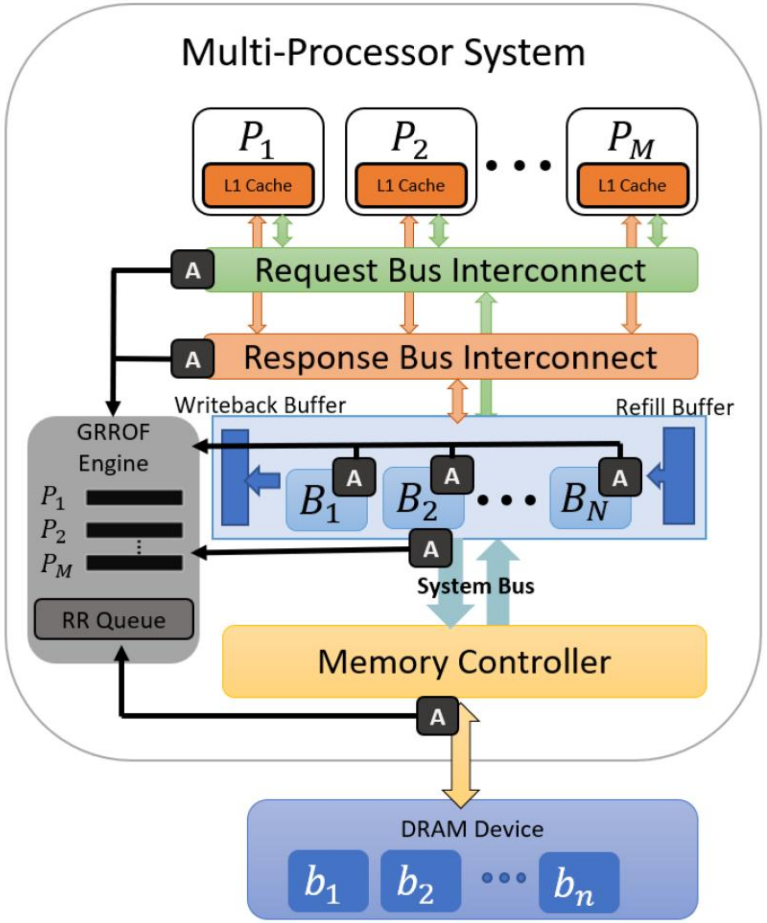


Memory System of a Multi-Core Platform

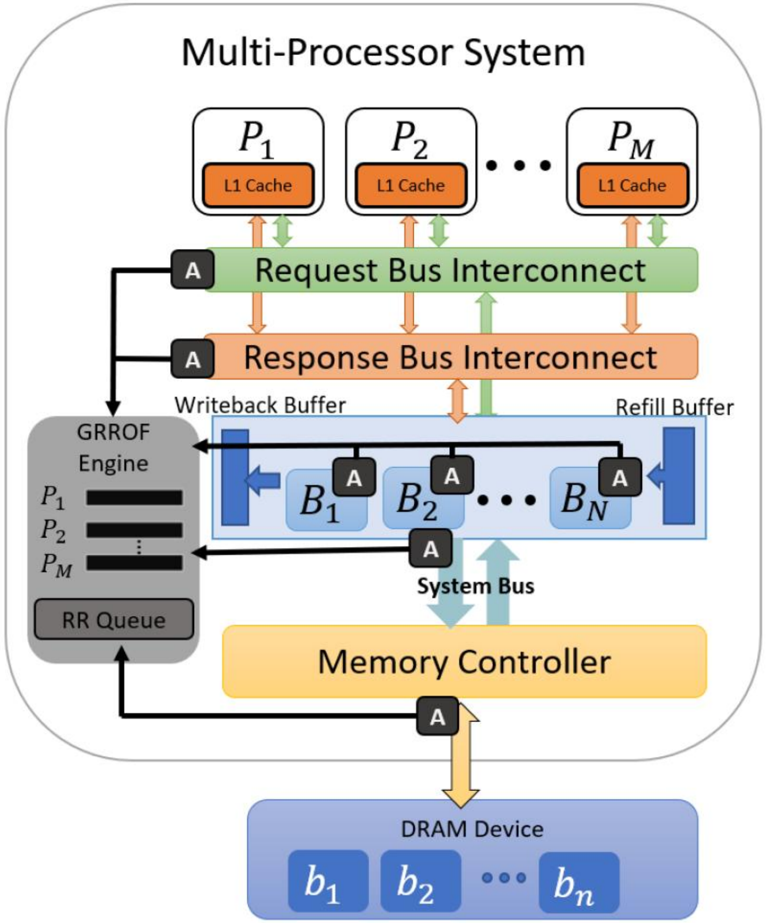
WCL of the oldest requests (**Discrete-RR**)

Each arbiter deploys a separate Round-Robin arbitration





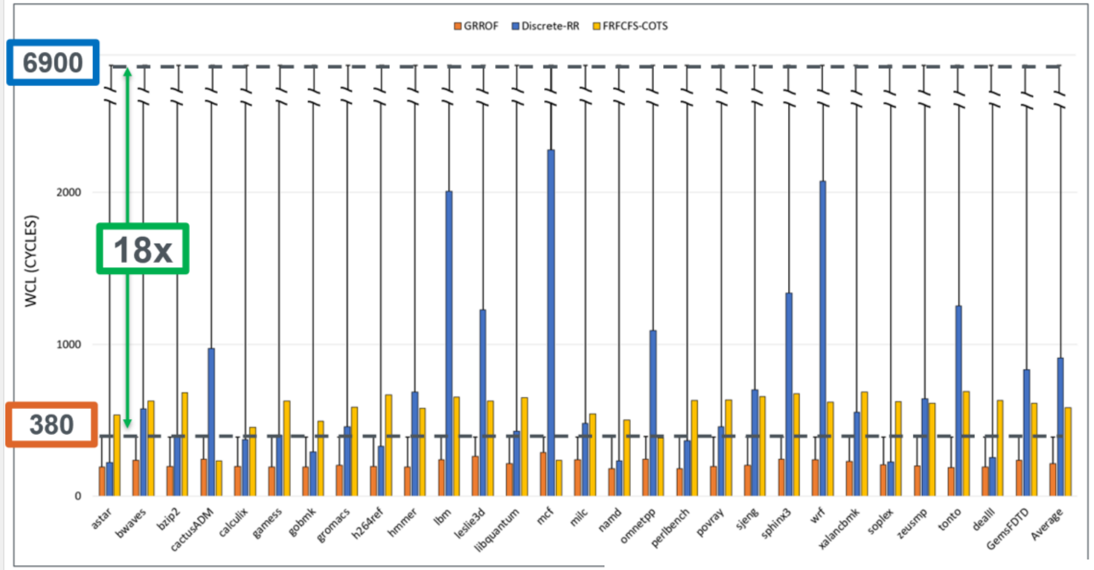
Coordinating Memory Resources



Evaluation

Per-Request WCL (SPEC BMs)

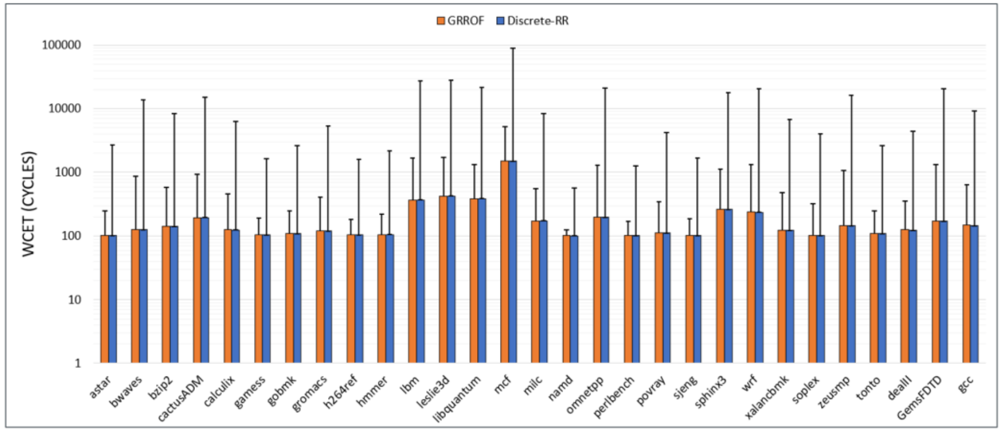
🟧 GRROF
🟦 Discrete RR
🟨 FRFCFS



Evaluation

Per-Task WCL (SPEC BMs)

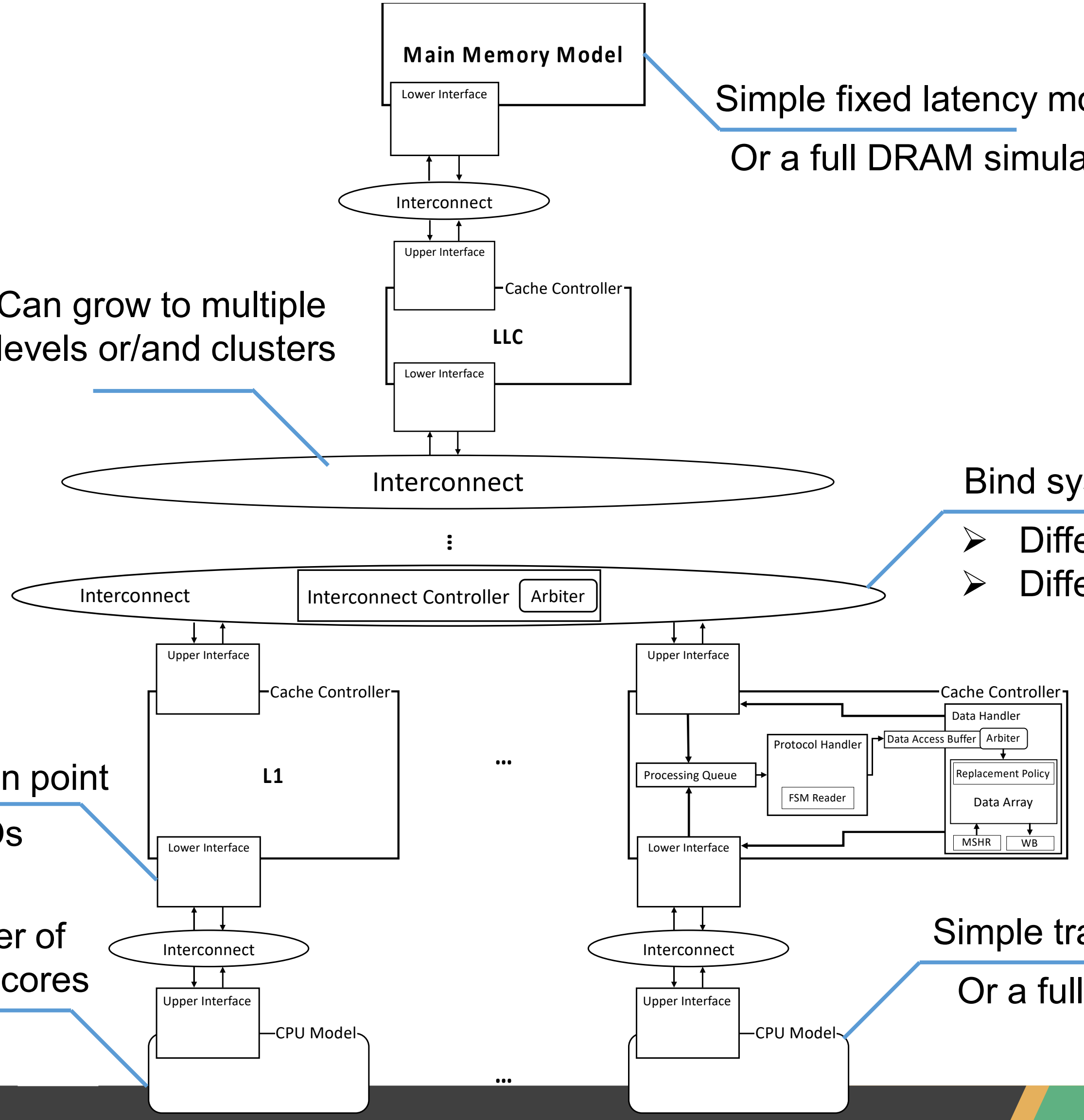
🟧 GRROF
🟦 Discrete RR



WCET (y-axis) is in logarithmic scale

Coordinating Memory Res

- Modularity: Plug and Play
- Extensibility
- Cycle-Accuracy
- Predictability/QoS Considerations



Simple fixed latency model

Or a full DRAM simulator

Can grow to multiple levels or/and clusters

Bind system components

- ➢ Different Topologies
- ➢ Different Arbiters

Connection point

FIFOs

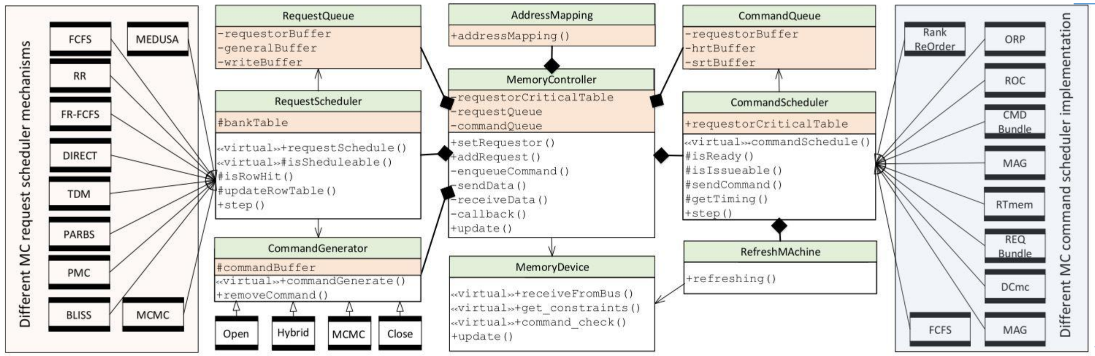
Any number of processing cores

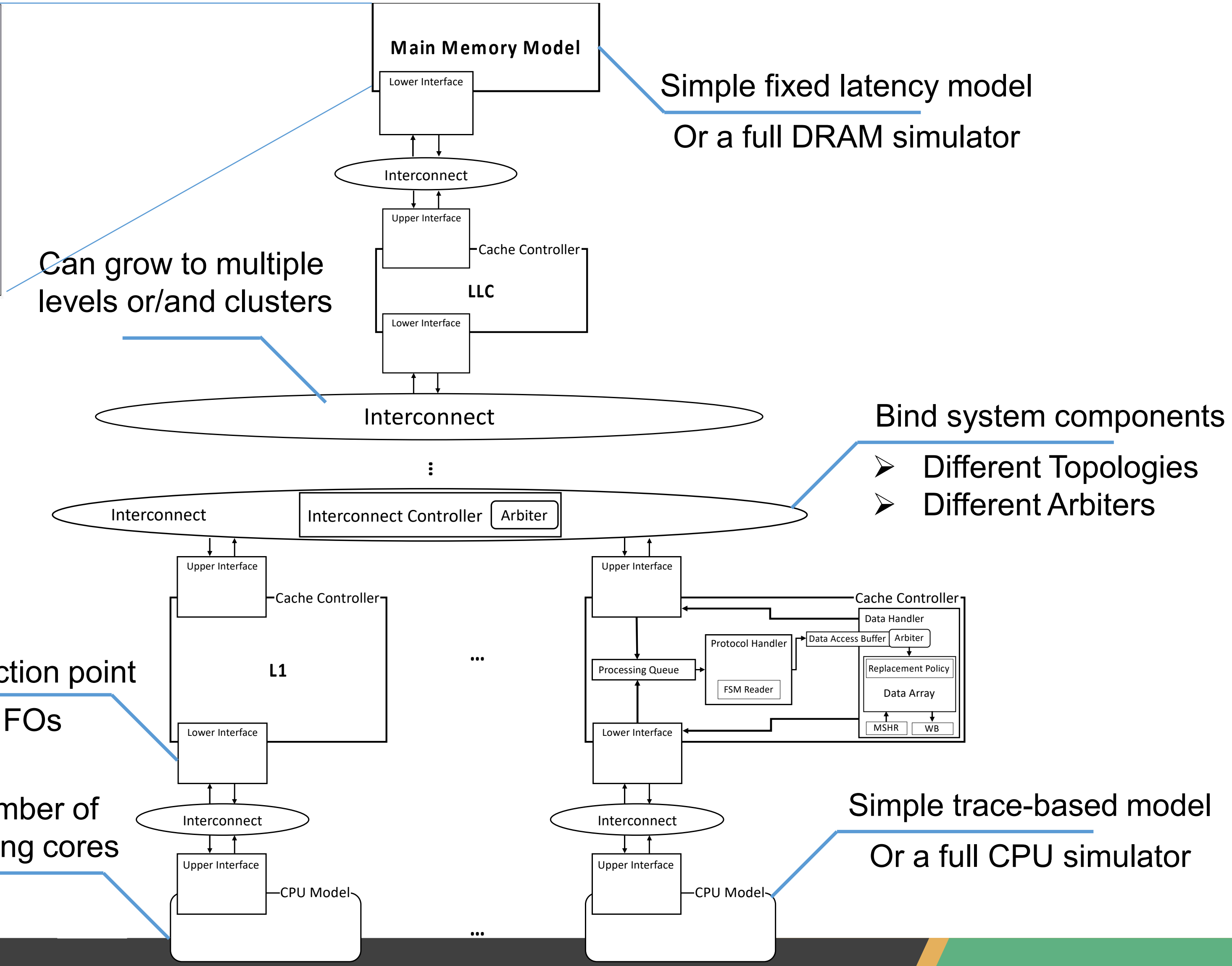
Simple trace-based model

Or a full CPU simulator

Octopus

Category	Supported Features
Coherence Protocols	MSI, MESI, MOESI; FSM-based with stable & transient states
Interconnects	Point-to-Point, Bus, Split Bus, Mesh, NoC; configurable controllers
Cache Hierarchy	Arbitrary levels; private/shared; multi-bank; MSHRs, write buffers, optimizations
Replacement Policies	LRU, FIFO, Random; extensible
Arbiters	FCFS, FR-FCFS, RR, Weighted RR, Harmonic RR, TDM, GRROF
Integration Modes	Standalone (trace-based) and full-system (gem5, MacSim, MCSim)
Monitoring Tools	Per-request latency tracking; CSV debug logs
Configurability	CSV/XML + CLI-based hierarchical configuration
Extensibility	Modular OO design; plug-in protocols/components
Real-Time Support	Set/bank partitioning; predictable coherence: PMSI, PISCOT, PCC, DUPECO, etc.





Simple fixed latency model

Or a full DRAM simulator

Can grow to multiple levels or/and clusters

Bind system components

- ➢ Different Topologies
- ➢ Different Arbiters

Connection point

FIFOs

Any number of processing cores

Simple trace-based model

Or a full CPU simulator

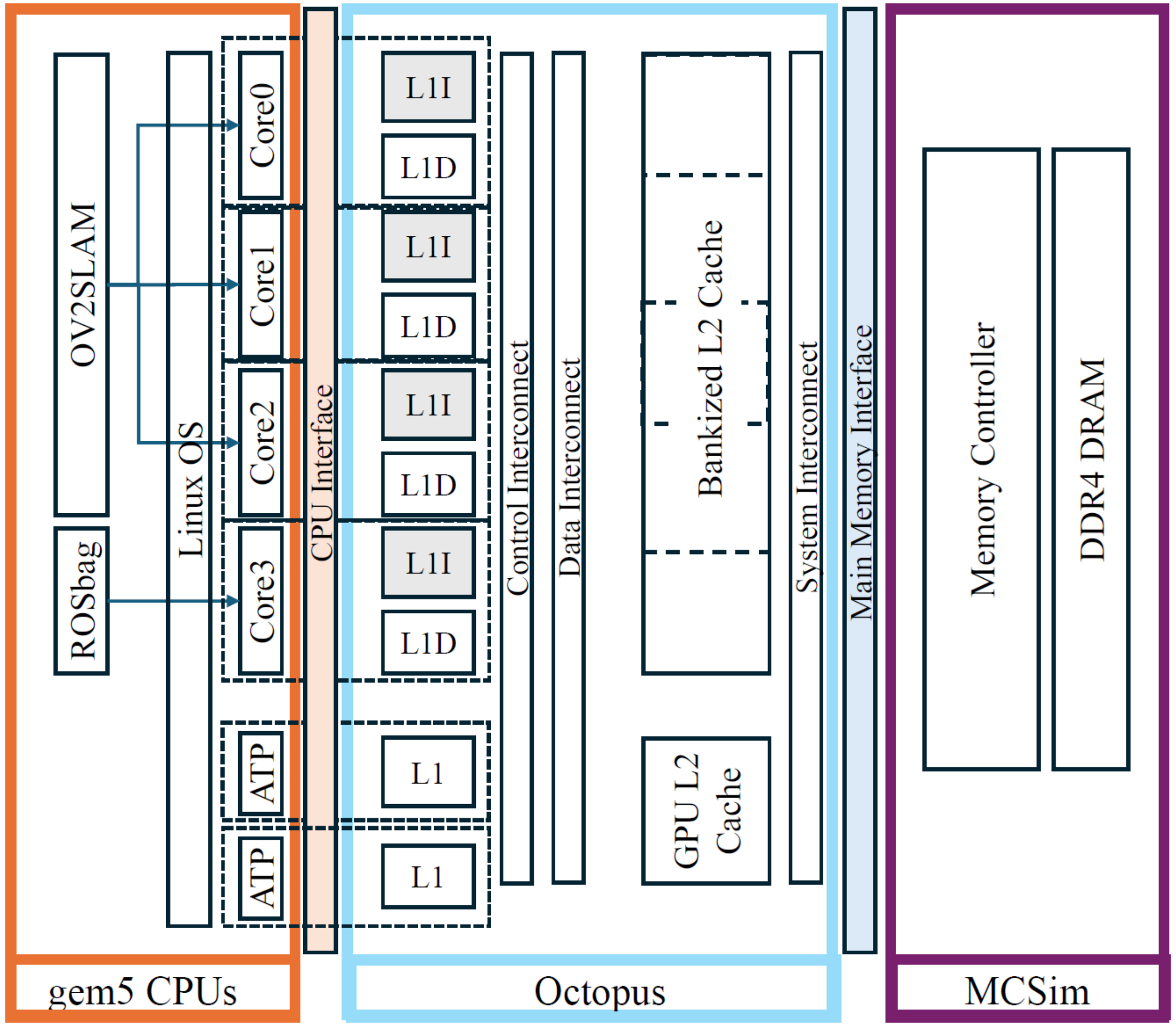
- Modularity: Plug and Play
- Extensibility
- Cycle-Accuracy
- Predictability/QoS Considerations

Octopus (Interconnect&Caches&Coherence) + MCSim (Main Memory)

MCSim

Category	Supported Features
Operation Modes	Trace-based or full-system integration (gem5, Octopus, MacSim)
Requestor Support	Per-requestor queues; criticality-aware scheduling
Memory Standards	DRAM standards via Ramulator; configurable DRAM hierarchy
Extensibility	New controllers added with ~133 LOC
Queue Configurability	Fully customizable request/command queues
Validation	Validated via MCXplore and simulator comparison
Implemented Controllers	High-Perf: FCFS, FR-FCFS, BLISS, PAR-BS; Predictable: REQBundle, ORP, PMC, etc.
Open Source	https://github.com/uwuser/MCsim

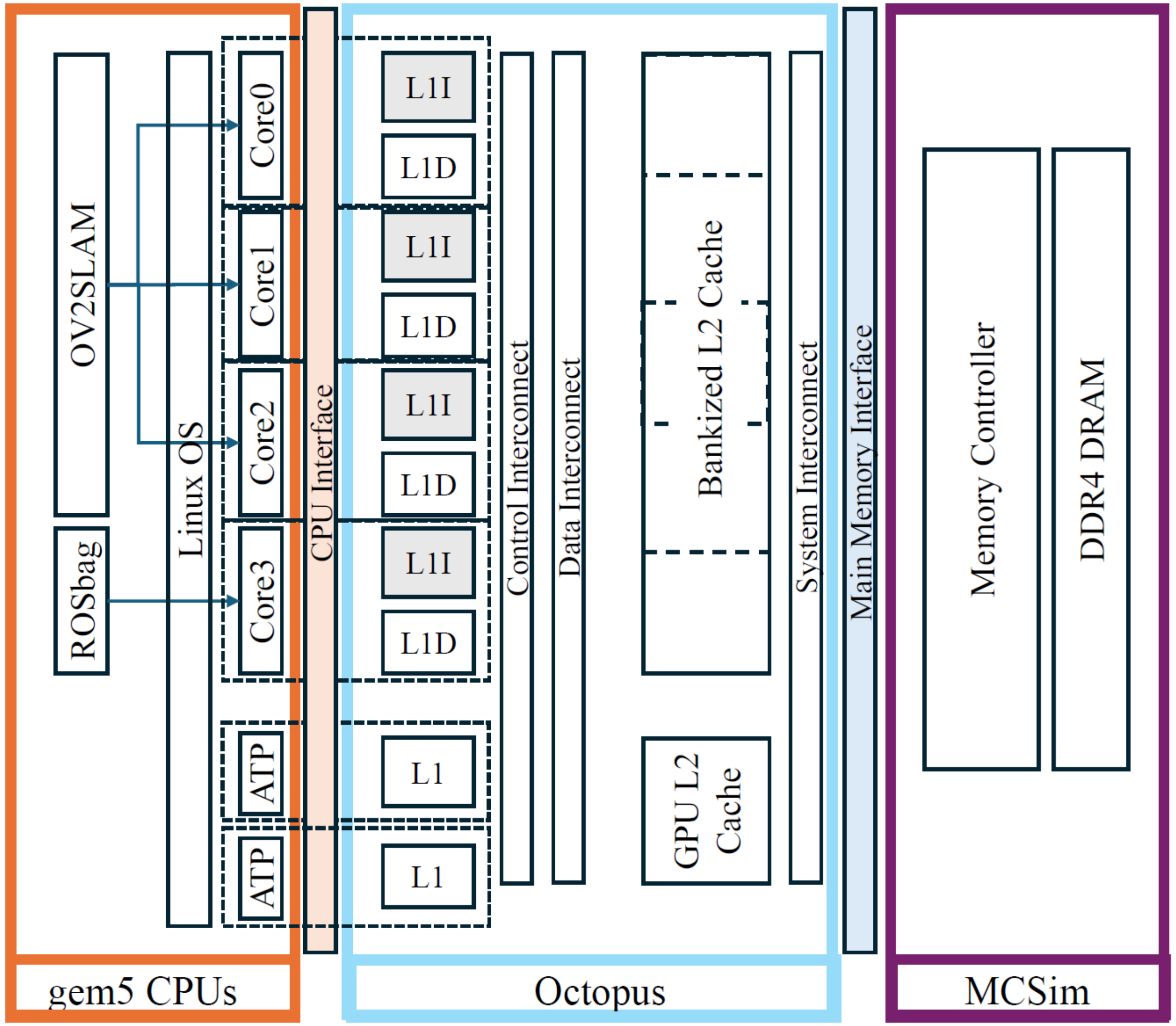
- We extended Octopus to enable gem5 full system simulation using ARM cores.
- We also implement additional machinery and tracking to fully support LL/SC instructions and the atomic memory operations (AMOs).



Octopus (Interconnect&Caches&Coherence) + MCSim (Main Memory) + Integrated into gem5

Industrial Challenge Use-case

- OV2SLAM:
  - Cores 0-2: OV2SLAM fast setting with 3 threads
  - Input images from the EuRoC dataset
  - The Machine Hall 01 scenario
  - The data has a frequency of 20Hz
- Rosbag:
  - Core 3
  - feeds images to OV2SLAM using
- GPU/DPU Interfering tasks:
  - Modeled using ARM's AMBA ATP



Evaluation Setup

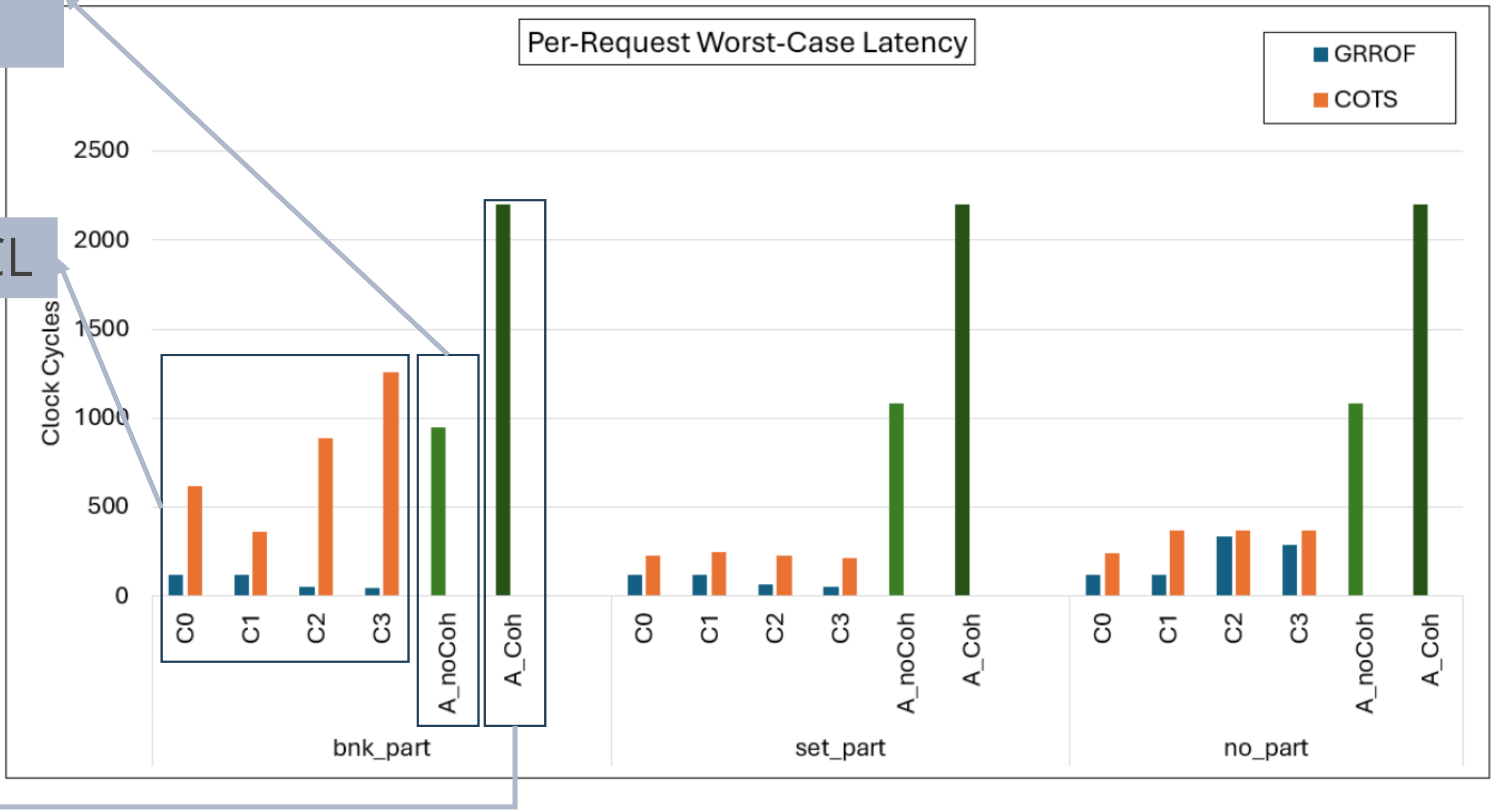
- Gem5 full system mode configured with Arm VExpress GEM5 V1
  - 4-cores O3CPU processor model.
- Modeled After Xavier platform:
  - Private 4-way 64KB L1 Split instruction and data caches
    - 16 MSHRs
  - Shared Cache Coherent 16-way 4MB L2 cache with 8 banks.
  - DDR4 2400U 8Gb x8 memory device.
- Operating System:
  - Ubuntu 18.04, Linux kernel 4.14, and ROS Melodic.
- We only monitor performance monitor and report statistics for the execution of OV2SLAM (region of interest) and not the Linux booting process.
- 6 Different setups:
  - Set, Bank, and no Partitioning (between interfering tasks and OV2SLAM)
  - COTS vs GRROF

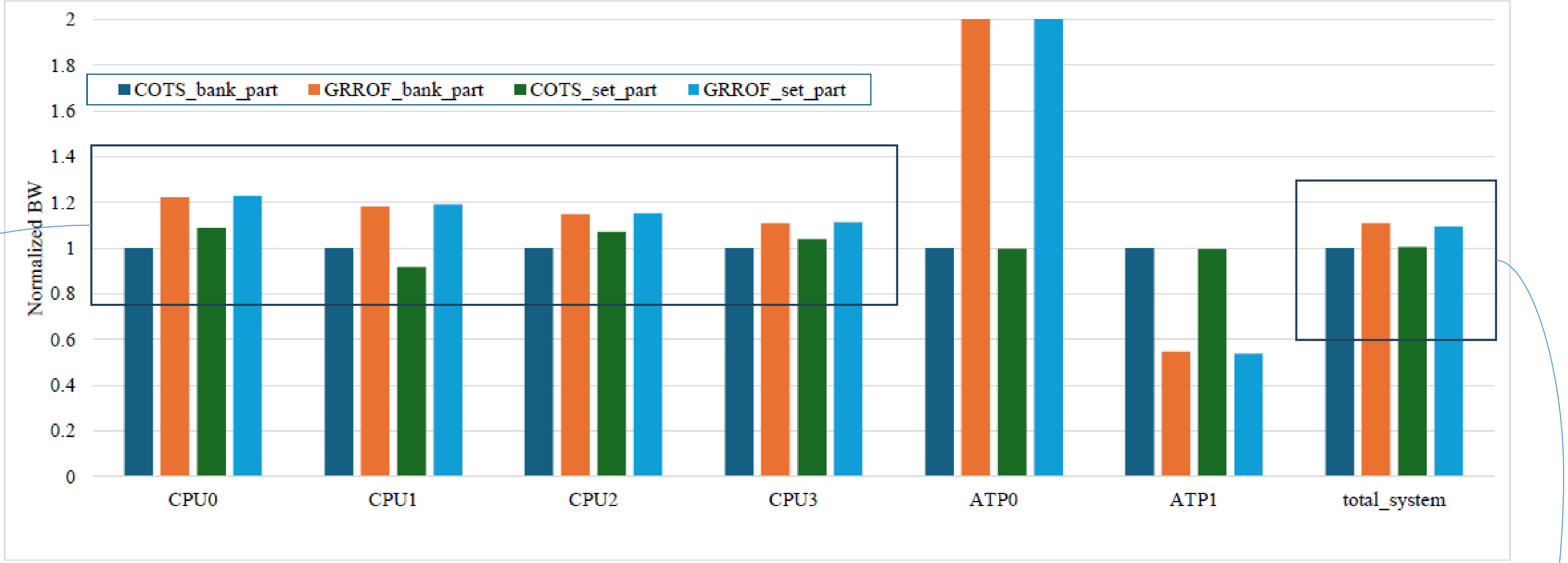
Per-Request WCL

Analytical WCL with no Coherence (ECRTS’23)

Observed Per-Req WCL

Analytical WCL with Coherence





Protecting critical task’s BW

Protecting critical task’s BW without comprising average system’s performance

Core’s and System BW

- Octopuses have both short-term and long-term memory powered by half a billion neurons (**Different Parallel Complex Memory Resources**)
- They can even enhance their short-term memory during times of stress when they need to learn quickly to survive. (**Real-Time**)
- Each of their eight legs has its own memory, giving them better reaction time to their surroundings. (**Agility, and Modularity**)

https://www.scientificamerican.com/article/the-mind-of-an-octopus/