

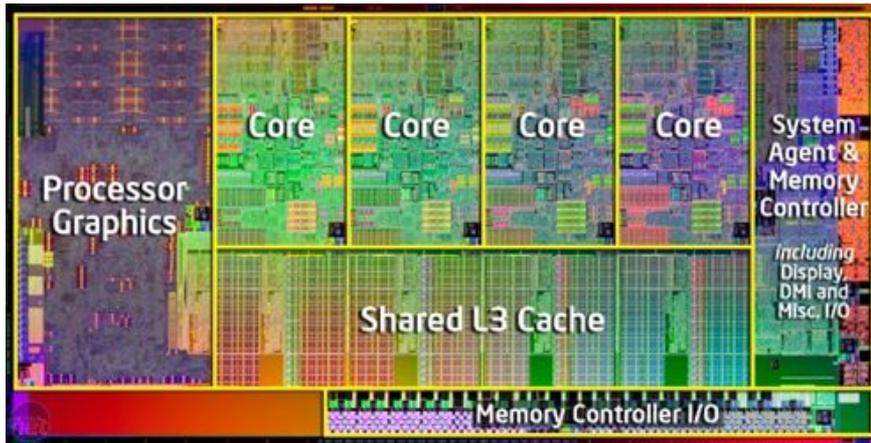
BRU: Bandwidth Regulation Unit for Real-Time Multicore Processors

Farzad Farshchi[§], Qijing Huang[¶], Heechul Yun[§]
[§]University of Kansas, [¶]University of California, Berkeley
RTAS 2020



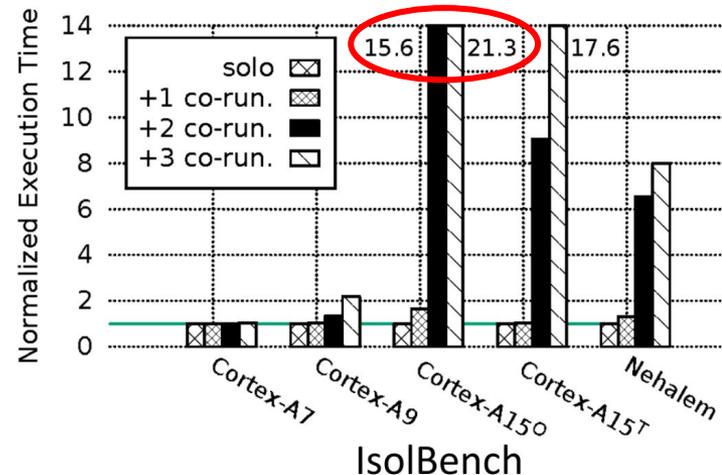
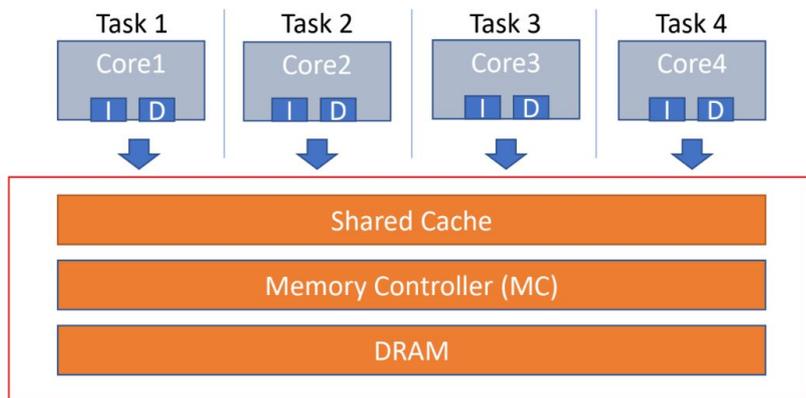
Multicore Processors in Real-time Systems

- Provide high computing **performance** needed for intelligent real-time systems
- Allow **consolidation** reducing cost, size, weight, and power



Challenge: Inter-core Memory Interference

- Memory system is shared between the cores
- Memory performance varies widely due to **memory interference**
- Task WCET can be **extremely pessimistic**: >10x or >100x



Software Solutions

- To **bound memory interference**: MemGuard¹, PALLOC², etc.
- Usually implemented in OS or hypervisor
- Use COTS processors features (performance counters, MMU, etc.)

✗ Fundamentally limited due to lack of full control over hardware

✗ Treat hardware as a black box

✗ Overhead. E.g. interrupt-handler overhead

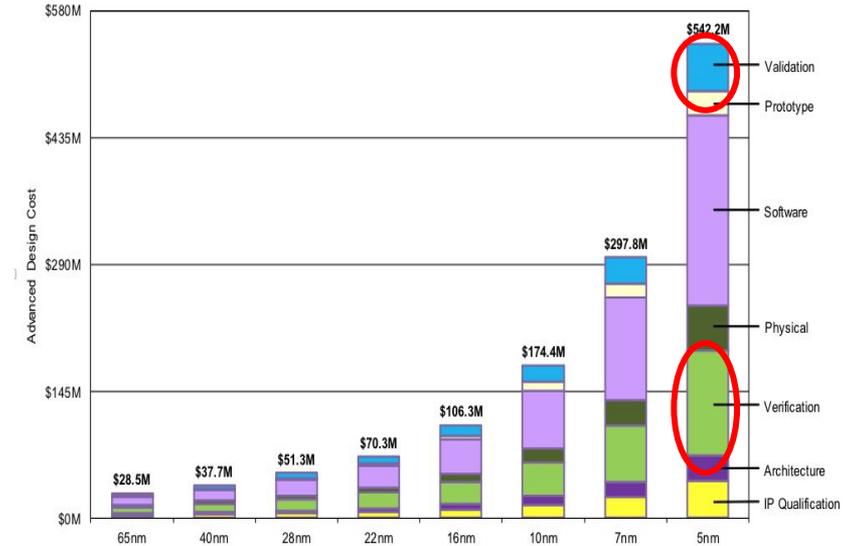
¹ H. Yun et al. "Memguard: Memory bandwidth reservation system for efficient performance isolation in multi-core platforms" RTAS'13

² H. Yun et al. "PALLOC: DRAM bank-aware memory allocator for performance isolation on multicore platforms" RTAS'14

Hardware Solutions

- Real-time architectures: T-CREST¹, MERASA²
- Priority-aware memory components: LLC³, DRAM controller⁴
- ✗ Low average performance
- ✗ Verifying a new IP is costly
- ✗ Hard to justify commercially

Cost of Developing a New Chip



<https://www.extremetech.com/computing/272096-3nm-process-node>

¹ M. Schoeberl et al. "T-CREST Time-predictable multi-core architecture for embedded systems" Journal of Systems Architecture 2015

² T. Ungerer et al. "MERASA: Multicore execution of hard real-time applications supporting analyzability" Micro'10

³ J. Yan et al. "Time-predictable L2 cache design for high performance real-time systems" RTCSA'10

⁴ F. Farshchi et al. "Deterministic memory abstraction and supporting multicore system architecture" ECRTS'18

Outline

- Motivation
- **BRU**
 - Access Regulation
 - Writeback Regulation
- Implementation
- Evaluation

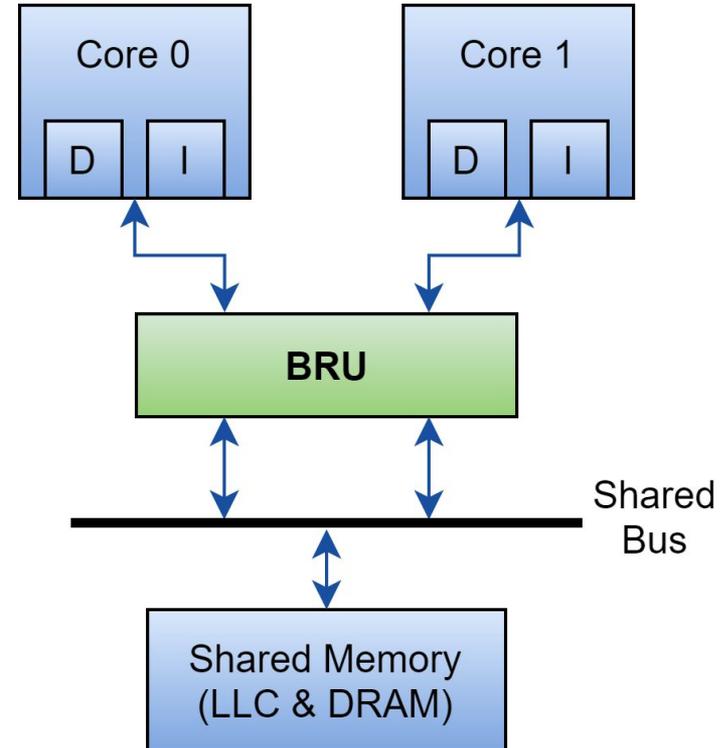
BRU: Bandwidth Regulation Unit

- BRU is a hardware IP
 - ✓ Drop-in module, less intrusive
 - ✓ No runtime overhead (e.g. interrupt handling)

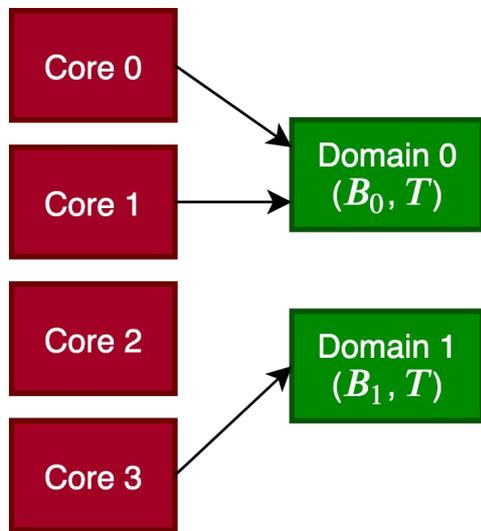
- BRU enables
 - ✓ Fine-grained regulation period
 - ✓ Group-regulation for multiple cores

Bird's Eye View of BRU Architecture

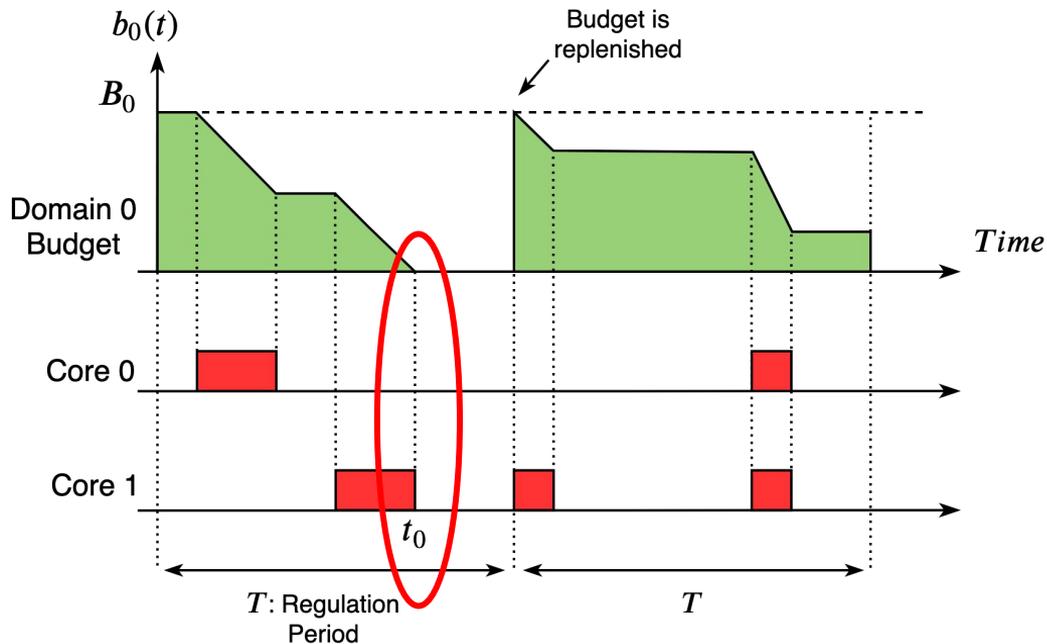
- Located between private caches and the shared memory
- Regulates bandwidth by **throttling private caches misses and writebacks**
- **Low logic complexity** due to direct connection to private caches
- Can throttle each core independently without interfering with the other cores
- No LLC metadata to store core ID



Access (Cache Miss) Regulation



Multiple cores can be assigned to a **domain**. B/W is **regulated collectively** for these cores.



t_0 : Access to shared memory by cores 0~1 is throttled

Domain budget is decremented when a private cache miss causes access to shared memory.

Bandwidth Budget Equation

$$B/W = \frac{B}{T} \cdot LS \cdot f_{clk}$$

Shared memory is accessed at the granularity of a cache line

LS : Cache line size

f_{clk} : System clock frequency

Writeback Regulation

- Cause and effect relationship between **cache misses** and **writebacks**:

Cache miss → cache conflict → dirty line eviction → writeback

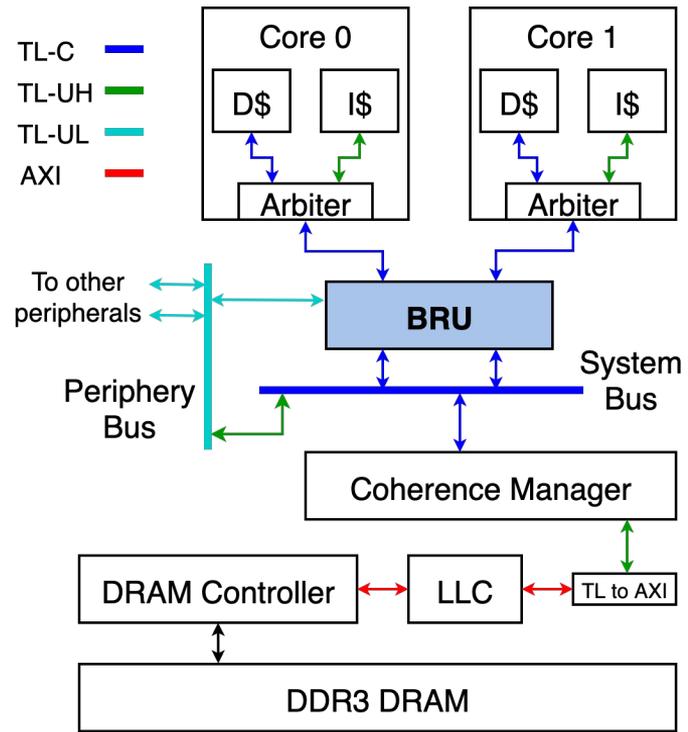
- With access bandwidth set to **X MB/s**, the writeback bandwidth is also limited to **X MB/s**
- Writes contend more severely in shared memory [1]. We want to set a lower budget for writebacks
- Add a **writeback budget** to each domain. When writeback budget depletes, throttle writebacks

Outline

- Motivation
- BRU
 - Access Regulation
 - Writeback Regulation
- **Implementation**
- Evaluation

Rocket Chip SoC¹

- An open-source system on chip
- Can be configured with BOOM² out-of-order processor
- Uses **TileLink** cache-coherent protocol for on-chip communication and accessing memory

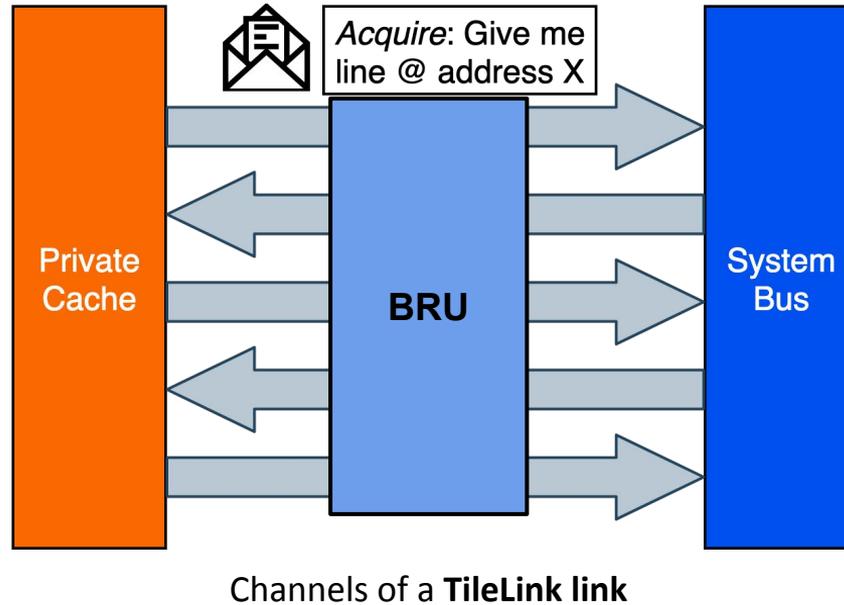


¹ K. Asanovic et al. "The Rocket Chip Generator" UC Berkeley Tech. Rep. 2016

² C. Celio et al. "The Berkeley Out-of-Order Machine (BOOM): An Industry-Competitive, Synthesizable, Parameterized RISC-V Processor" UC Berkeley Tech. Rep. 2015

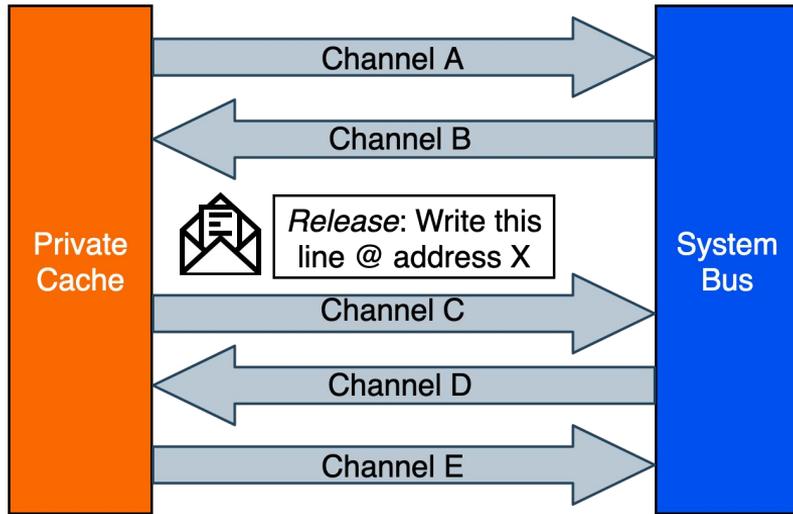
Rocket Chip augmented with BRU

Access Regulation Implementation

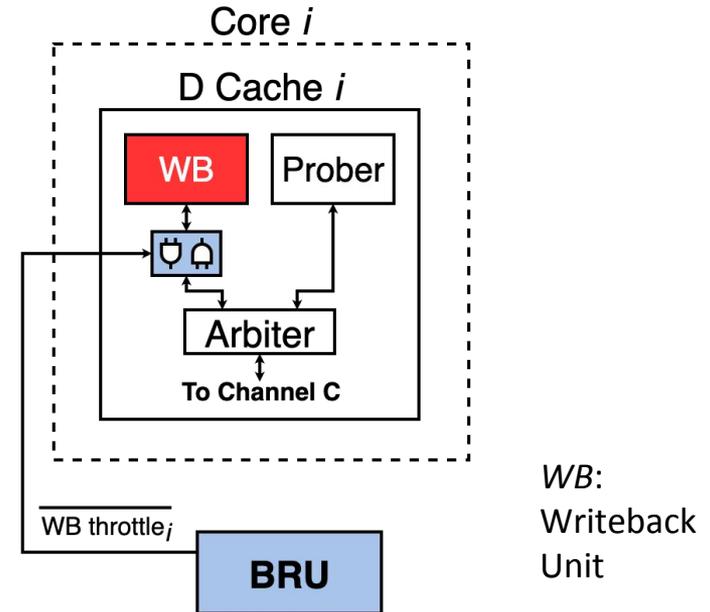


- On a cache miss, an *Acquire* message is transferred over Channel A
- BRU counts *Acquires* and when the budget depletes, throttles Channel A

Writeback Regulation Implementation



- On a writeback, a *Release* message is transferred over Channel C
- Cannot throttle Channel C due to other messages (*Probe responses*) going through this channel



- A special throttle logic inserted after WB unit (only two AND gates)
- BRU sends a signal to D cache to throttle writebacks

Outline

- Motivation
- BRU
 - Access Regulation
 - Writeback Regulation
- Implementation
- **Evaluation**

Evaluation

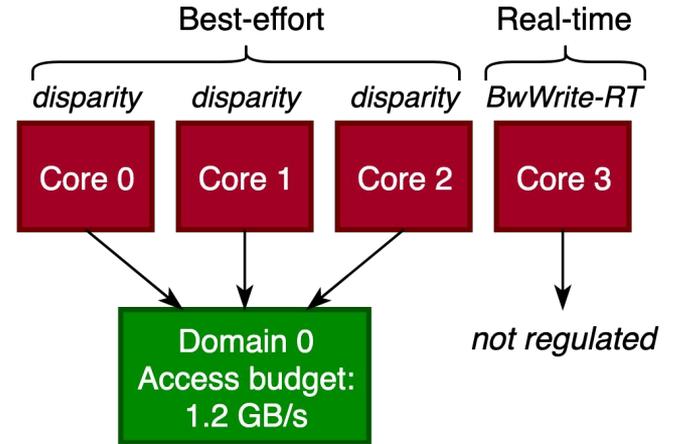
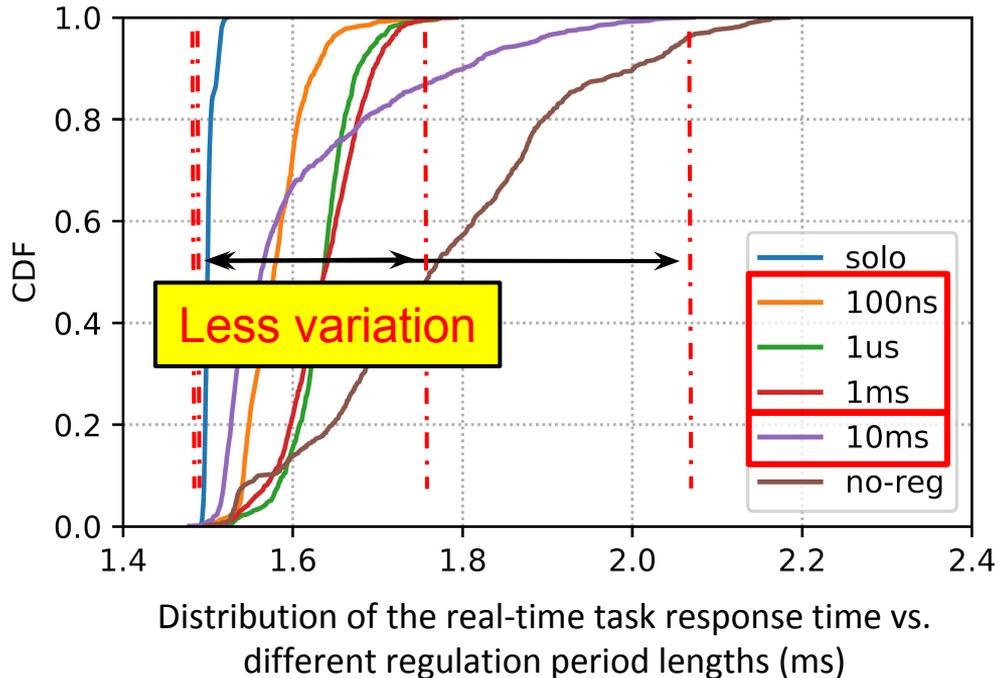
- FireSim FPGA-accelerated simulator
 - Directly derived from RTL
 - Runs on FPGAs in Amazon cloud
 - **Fast, highly accurate**
- Setup
 - Quad-core out-of-order (RISC-V ISA) 2.13 GHz
 - Caches: 64-byte lines, Private L1-I/D: 16/16 KiB, Shared LLC: 2MiB
 - DDR3-2133, 1 rank, 8 banks, FR-FCFS
- Workloads
 - SD-VBS¹, IsolBench² (synthetic)



¹ S. K. Venkata et al. "SD-VBS: The san diego vision benchmark suite" IISWC'09

² <https://github.com/CSL-KU/IsolBench>

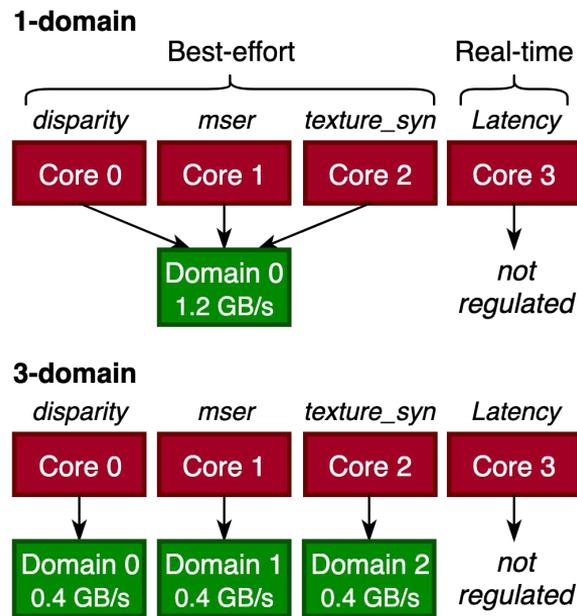
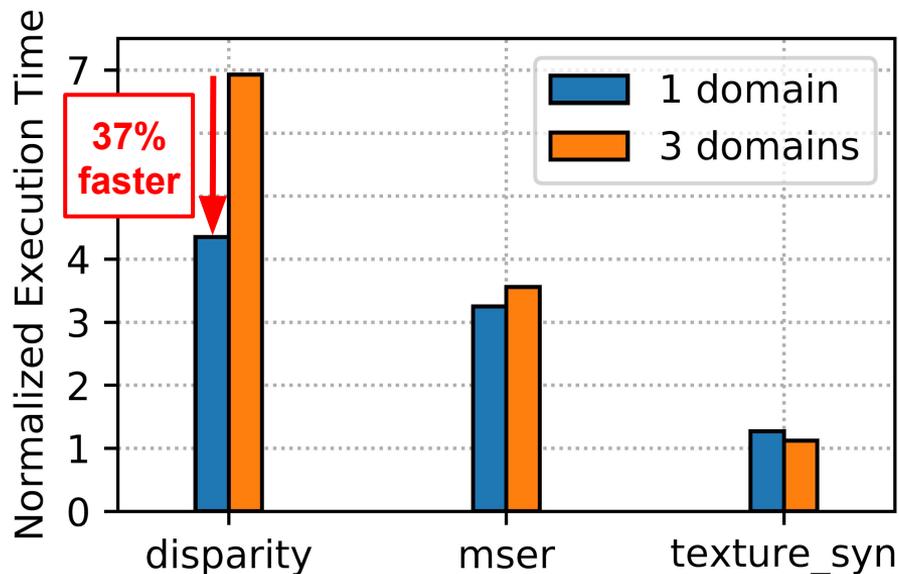
Effect of Regulation Period Length



Real-time task **WCET: 1.5ms** in isolation, run for 1k periods

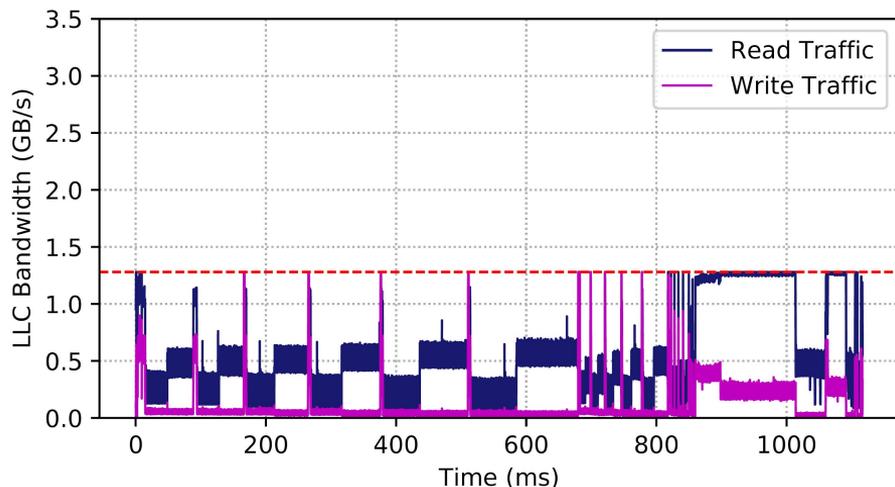
- Regulation period **shorter than task WCET** reduces **response time variation**

Effect of Group Bandwidth Regulation

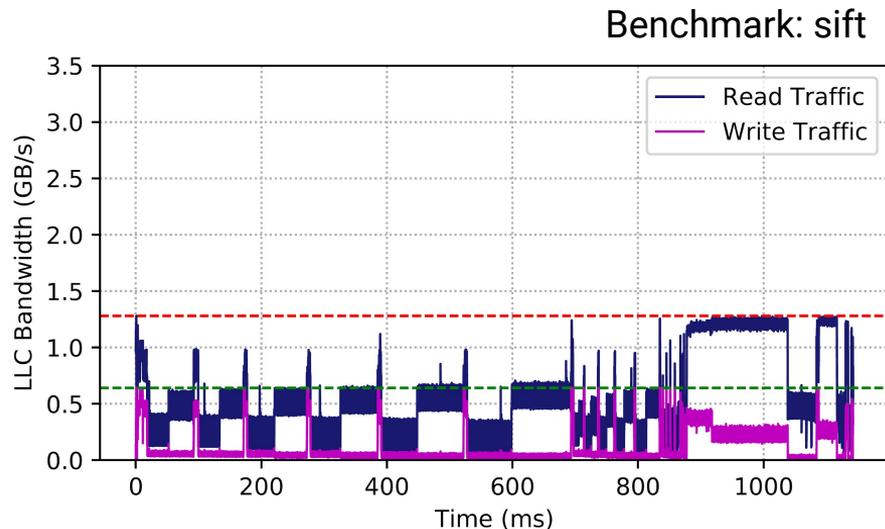


- Memory intensity: $\text{disparity} > \text{msr} > \text{texture_syn}$
- Group bandwidth regulation of best-effort tasks improves utilization

Effect of Writeback Regulation



Writeback regulation: **disabled**
Access budget: 1.28 GB/s



Writeback budget: **0.64 GB/s**
Access budget: 1.28 GB/s

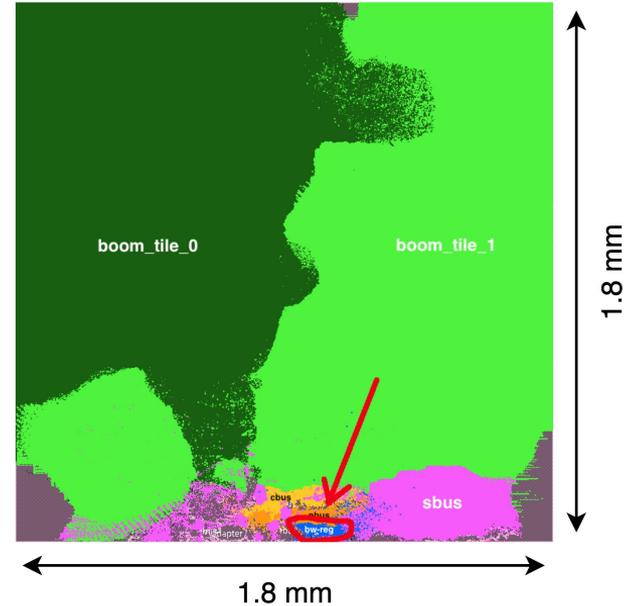
- Access regulation limits writeback bandwidth
- Writeback regulation allows setting a lower budget for writebacks

Hardware Overhead

- Synthesis and place and route for 7nm
- The area overhead is negligible: < **0.3%**
- < **2%** impact on max clock frequency

BOOM PROCESSORS AREA BREAKDOWN (mm^2)

	Dual-core	Quad-core	Octa-core
BRU	0.005 (0.19%)	0.007 (0.17%)	0.023 (0.28%)
BOOM Cores	2.310 (92.41%)	4.072 (95.13%)	8.144 (96.99%)
Others (Buses, Manager, etc.)	0.185 (7.40%)	0.201 (4.70%)	0.230 (2.74%)
Total	2.499	4.280	8.397



A dual-core processor chip layout with BRU circled in red

Conclusion

- BRU enables **bounding the memory interference** with minimal changes to the hardware
- Single drop-in module; **less intrusive** than other hardware solutions
- No runtime overhead; **reduces response time variation** and **improves utilization**
- Negligible hardware overhead

Thank you for listening!

Acknowledgement:

This research is supported in part by NSF CNS 1718880 and CNS 1815959, NSA Science of Security initiative contract #H98230-18-D-0009, and AWS Cloud Credits for Research.