Warped-MC:
An Efficient Memory Controller Scheme for Massively Parallel Processors

Jong Hyun Jeong*, Myung Kuk Yoon†, Yunho Oh*, Gunjae Koo*

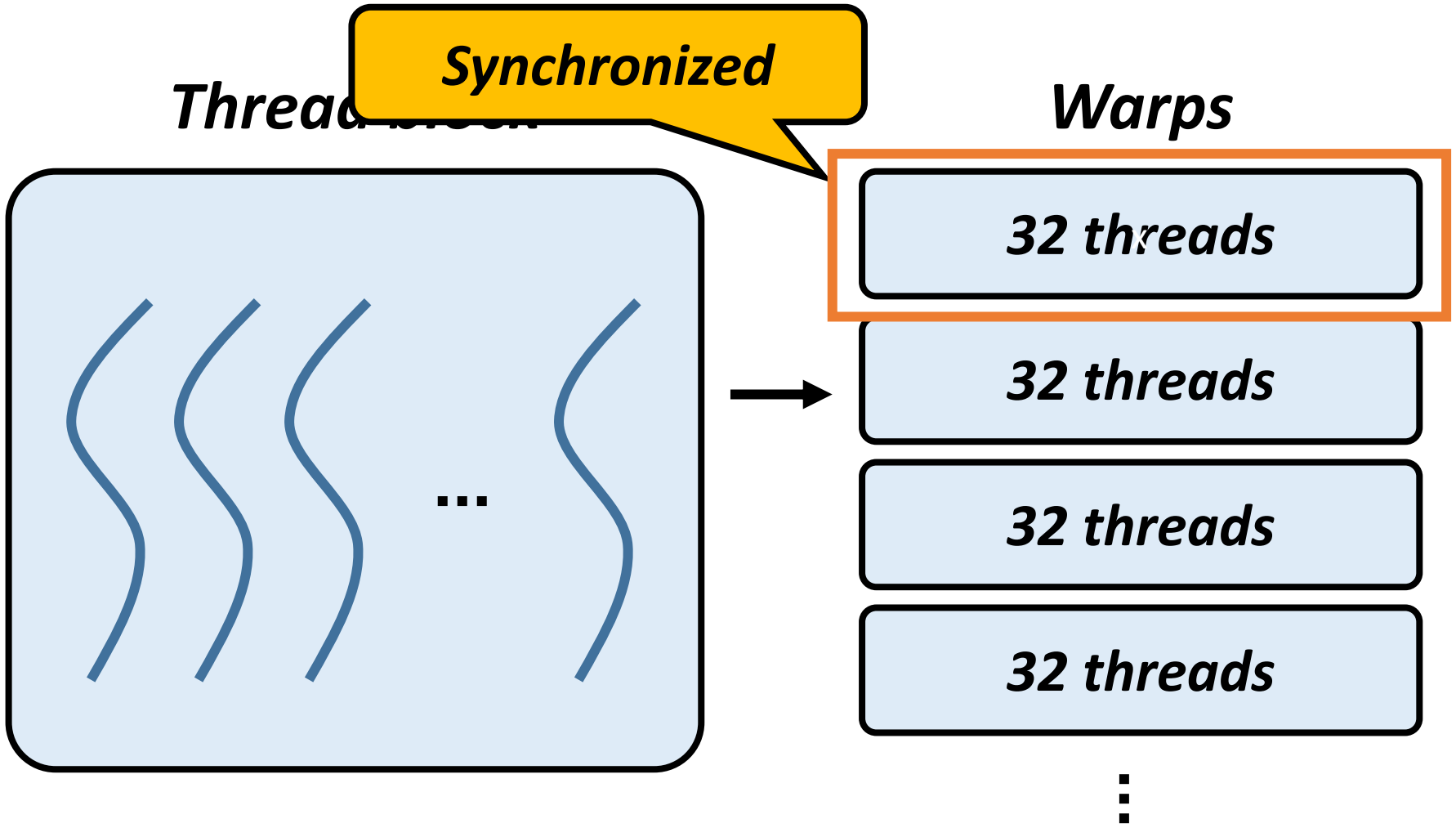
*Korea University

†Ewha Womans University

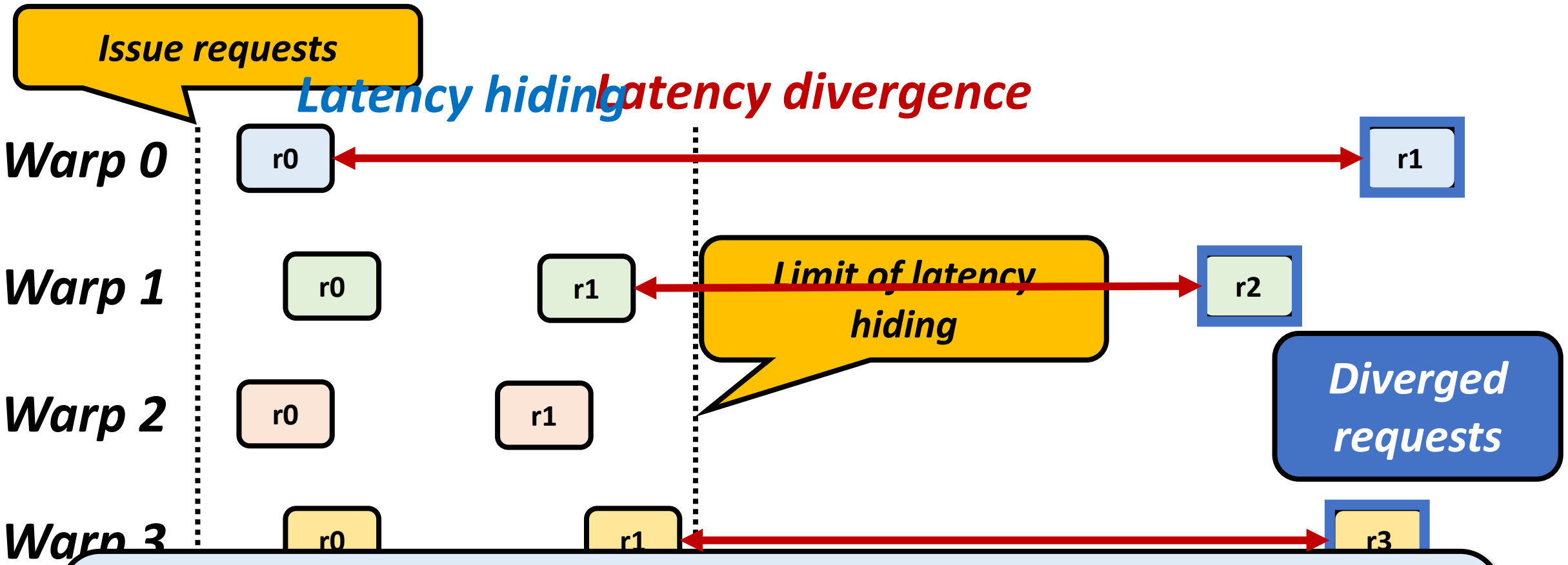
Outline

- *Background & Motivation*
- *Warped-MC*
- *Evaluation*
- *Conclusion*

Warp Execution



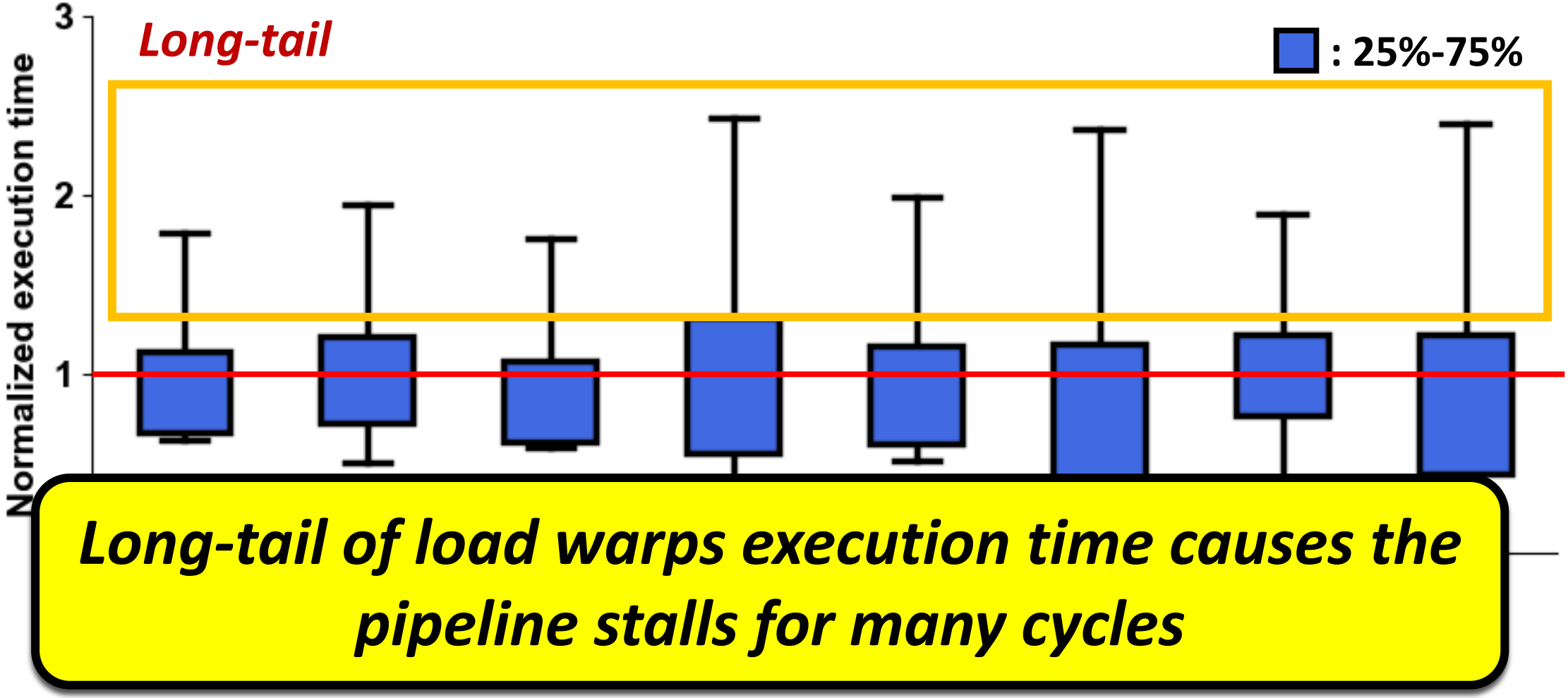
Memory Latency Divergence



Latency divergence within a warp

- GPU can not hide long memory latency
- Disturb the lock-step executions of warps

Diverged Load Warp Execution Time



Long-tail of load warps execution time causes the pipeline stalls for many cycles

Source of Latency Divergence

Ex-controller

- Failure in cache reservation
- Congestion in interconnection network

In-controller

- Timing behavior of memory
- Diverged memory scheduling

Over 80% of latency divergence is provoked by memory controllers

Diverged Memory Scheduling

Conventional GPU memory controller

- First-ready scheduling for maximizing memory bandwidth
- Does not consider architectural features of GPU
- Cannot prevent diverged requests

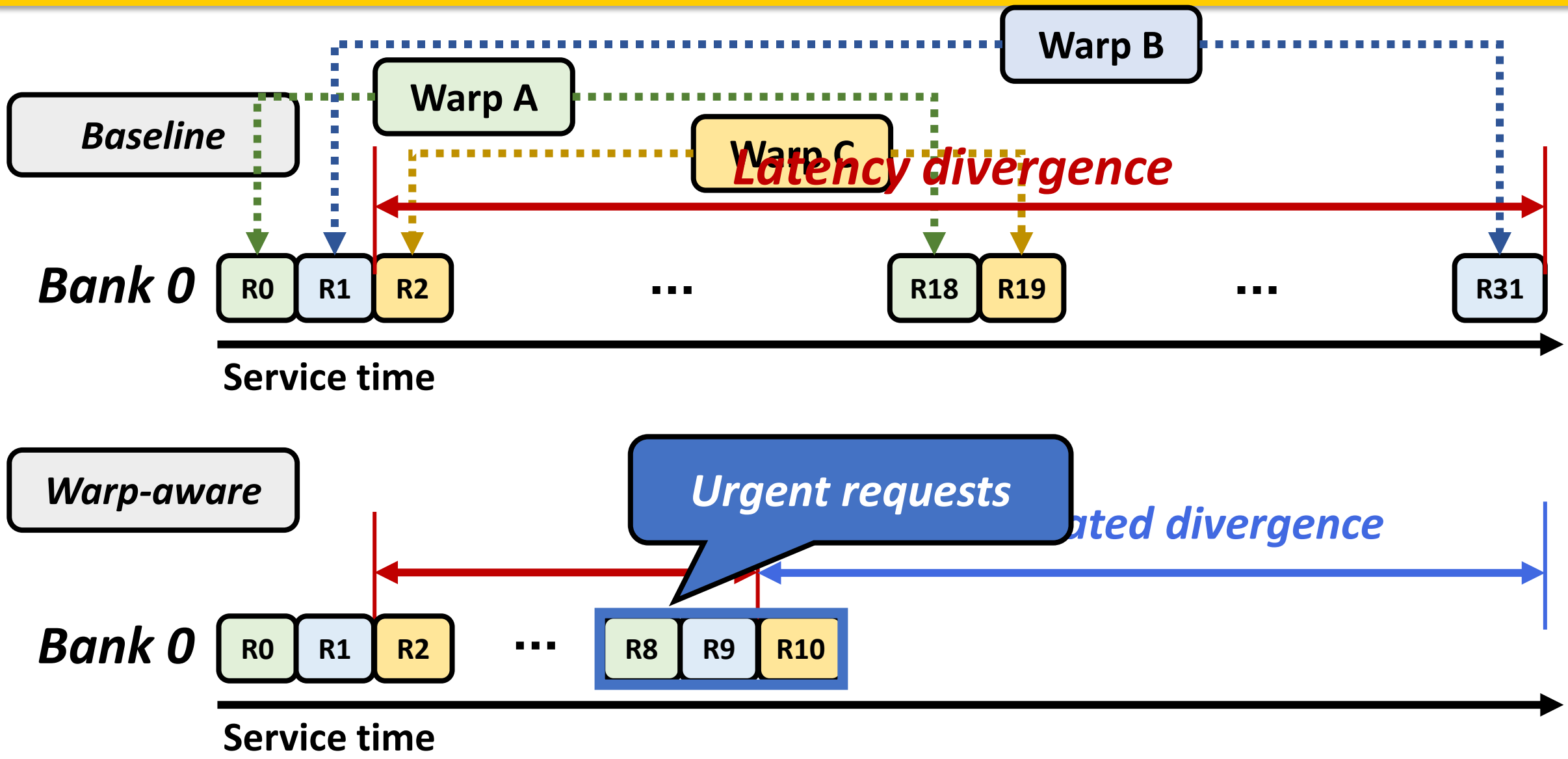
GPU needs a warp-aware memory controller scheme

Solution: Warp-Aware Memory Scheduling

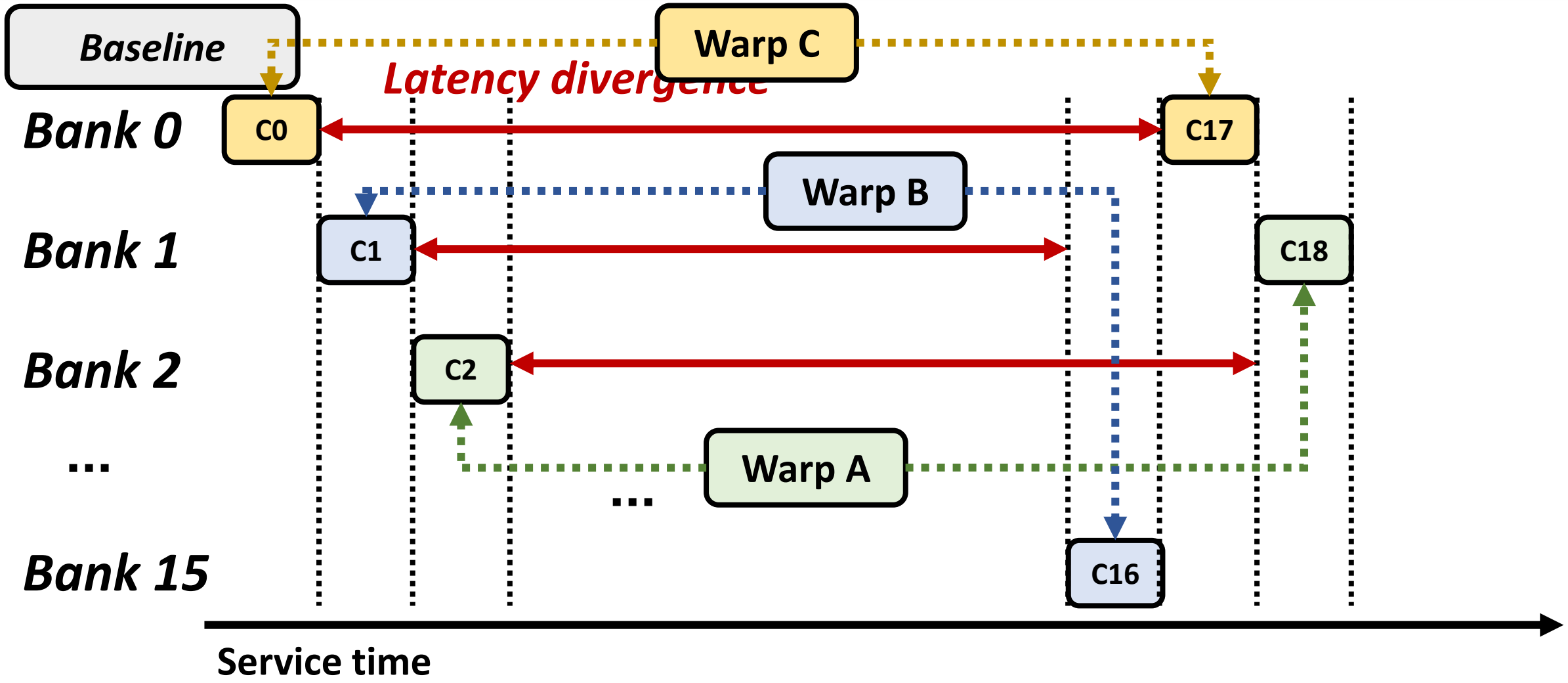
Warp-aware memory scheduling

- Defines an urgent request (the ***slowest*** request of a warp)
- Identifies an urgent request
- Prioritizes an urgent request
- ***Maintains high memory throughput***

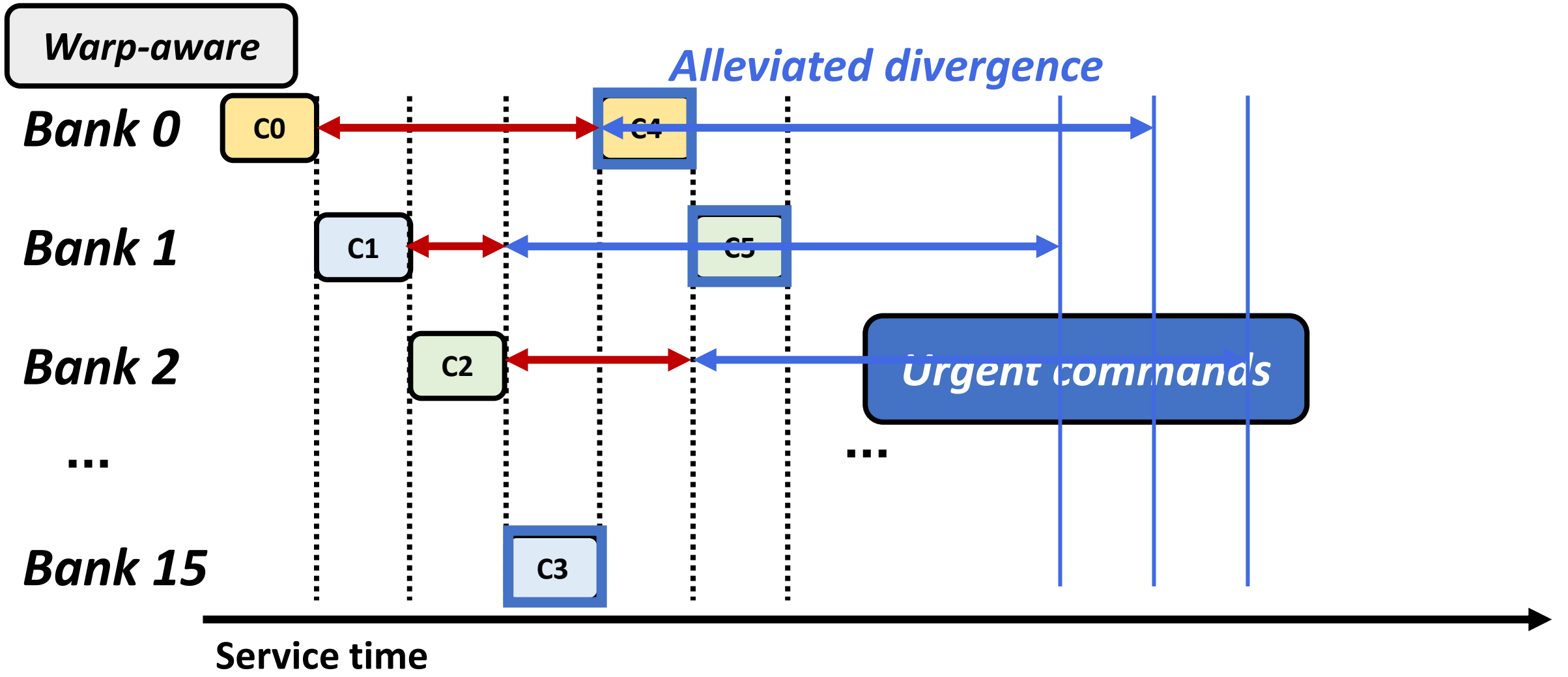
Warp-Aware Memory Scheduling Scheme: Request



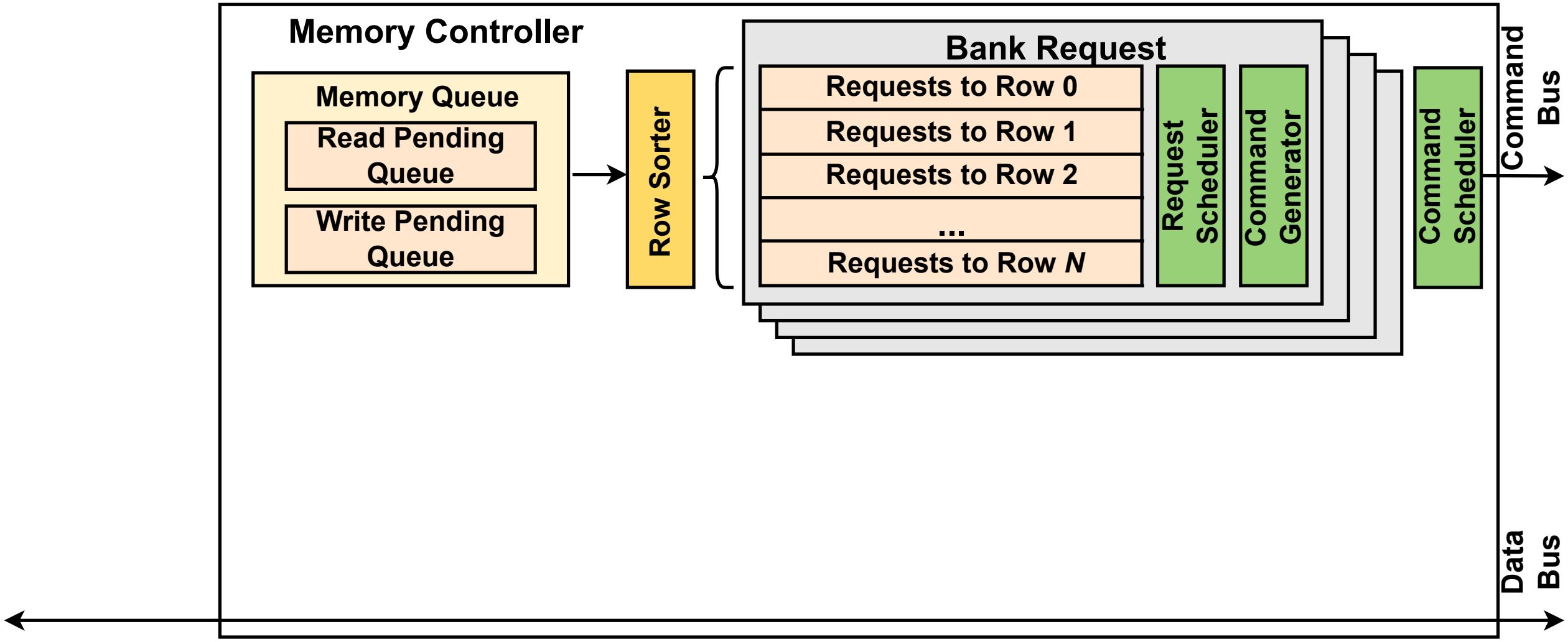
Warp-Aware Memory Scheduling Scheme: Command



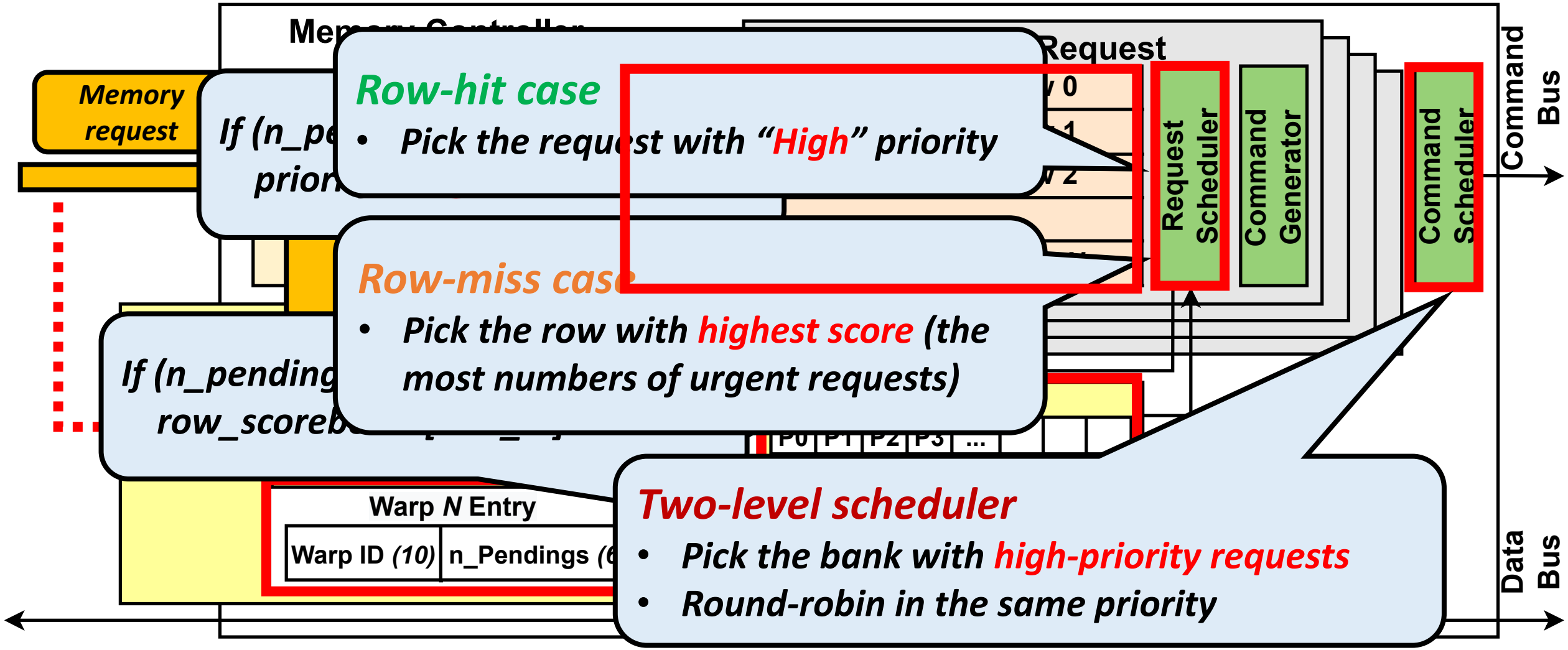
Warp-Aware Memory Scheduling Scheme: Command



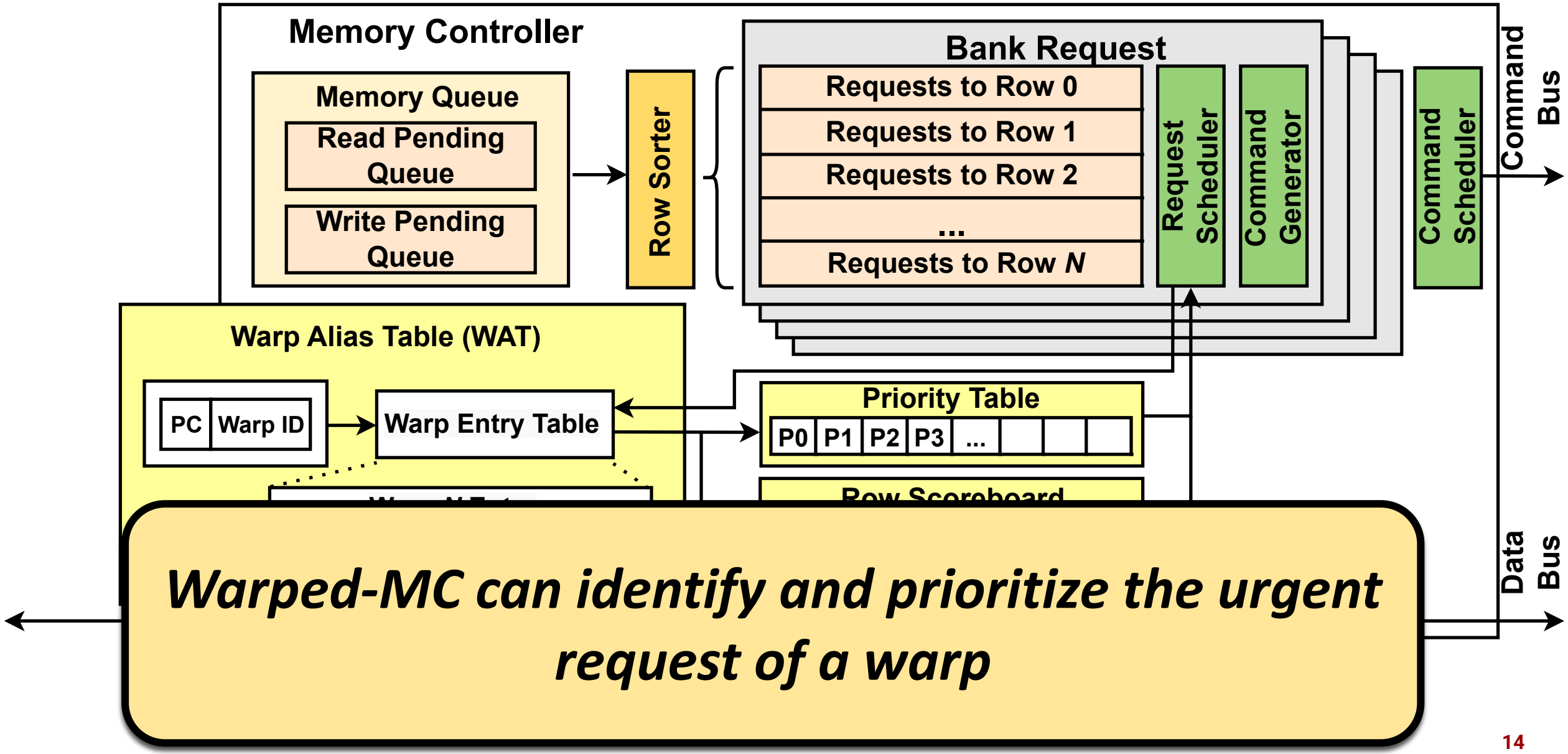
Warped-MC Architecture



Warped-MC Architecture



Warped-MC Architecture



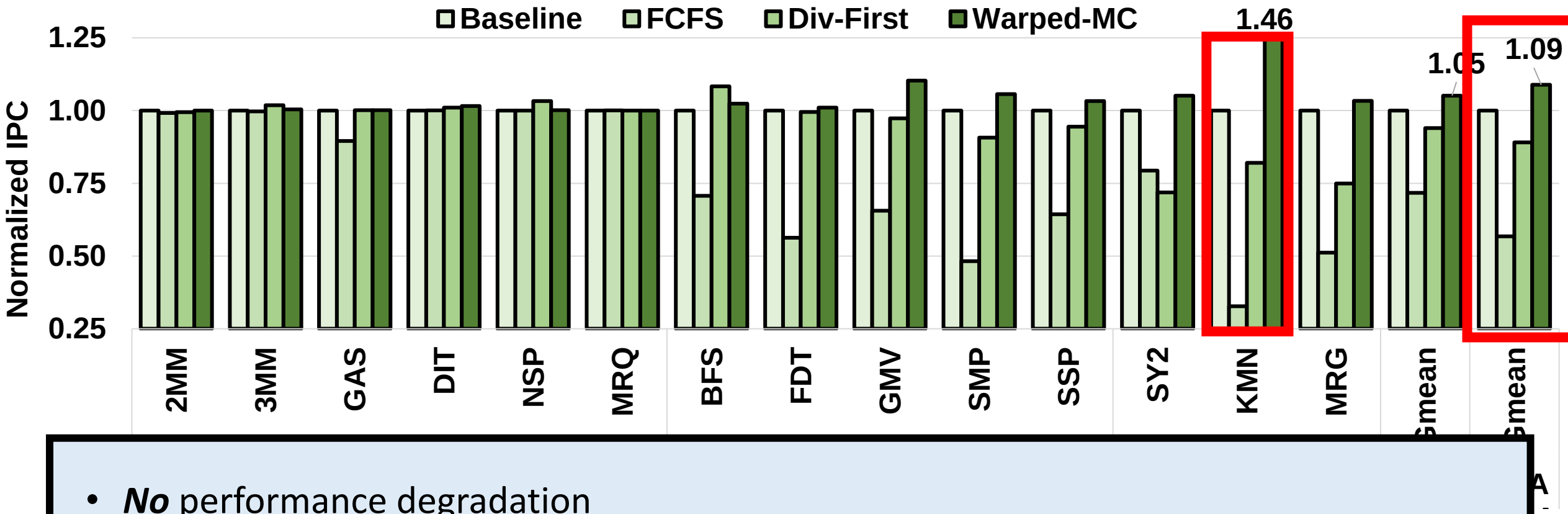
Warped-MC can identify and prioritize the urgent request of a warp

Evaluation

- GPGPU-Sim v4.2, CUDA 10.1
- Configuration: NVIDIA RTX2060 Super
- Baseline: FR-FCFS request scheduling + round-robin command scheduling
- 14 GPGPU applications, 3 types by the number of off-chip requests per a warp

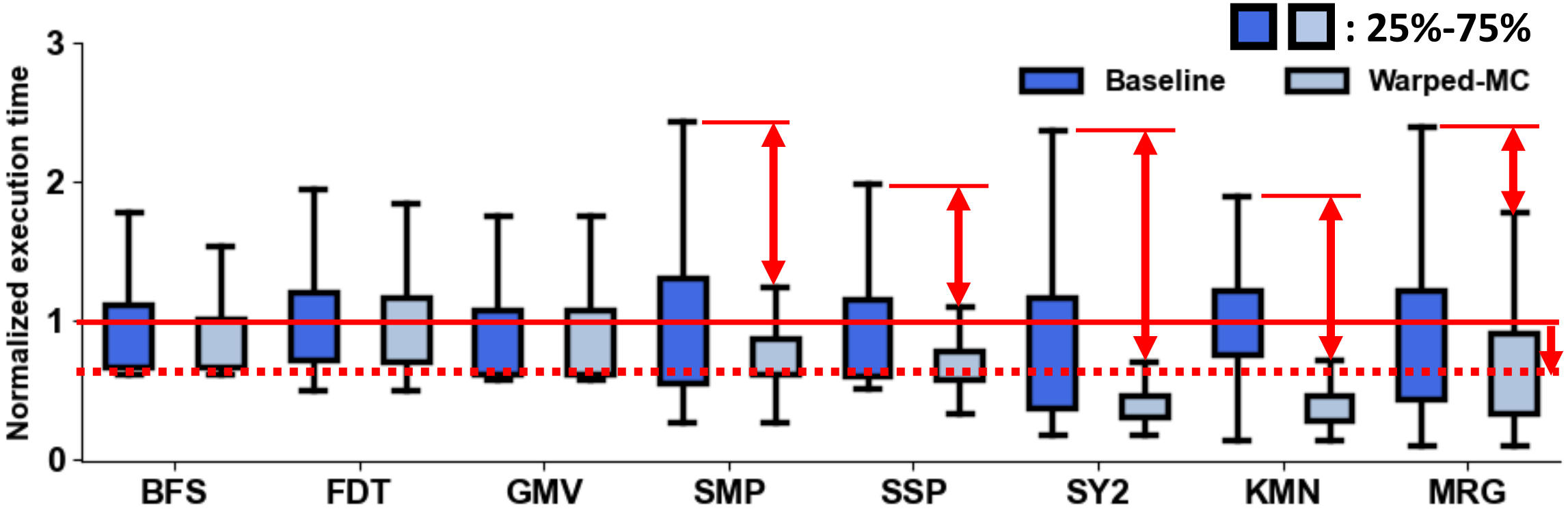
Parameter	Configuration
Core	32 SMs, 64 CUDA cores / SM @ 1905 MHz
Warp	32 Warps / SM
Warp scheduler	LRR, 4 schedulers / SM
CTA	32 CTAs / SM
Register file	256 KB / SM
L1 data cache	64 KB / SM, 128B line, 4 sector / line, LRU, 16 way
L2 cache	4 MB, 128B line, LRU, 16 way
Dram	GDDR6, 384 bit bus @3500 MHz, 16 channels, 16 banks
GDDR timing	tRP=20, tRC=62, tRAS=50, tRCD=20, tRRD=10, tCCD=4

Result: Performance



- ***No*** performance degradation
- Improve performance by ***8.9%*** on average

Result: Load Warp Execution Time



- Decrease execution time of load warp ***19.3%*** on average with a maximum ***39.7%***

Conclusion

Latency divergence

- Critical to warp executions
- Mainly provoked by memory controllers

Warped-MC

- Warp-aware memory controller scheme
- Identify and prioritize urgent requests of warp

Key results

- ***8.9%*** performance improvement for applications with many off-chip accesses
- Decrease ***19.3%*** of load warp execution time and mitigate long-tail of load warp execution

Thank you

Warped-MC: An Efficient Memory Controller Scheme for Massively Parallel Processors

Jong Hyun Jeong, Myung Kuk Yoon, Yunho Oh, Gunjae Koo

✉ dida1245@korea.ac.kr