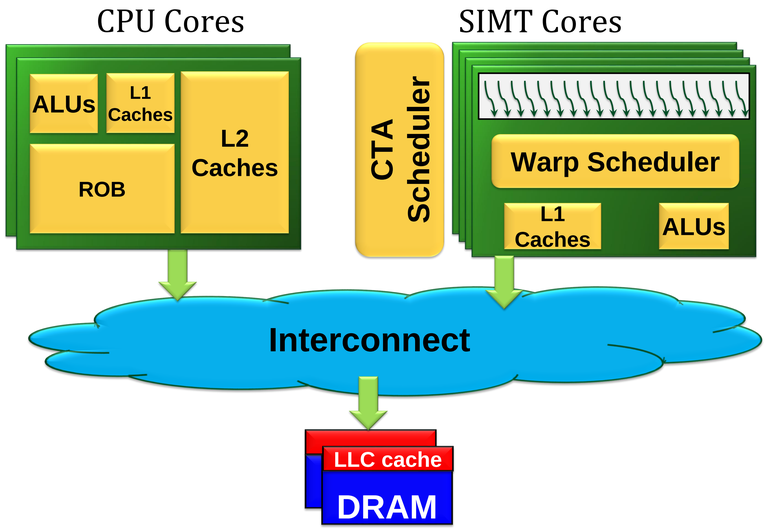
# Managing GPU Concurrency in Heterogeneous Architectures

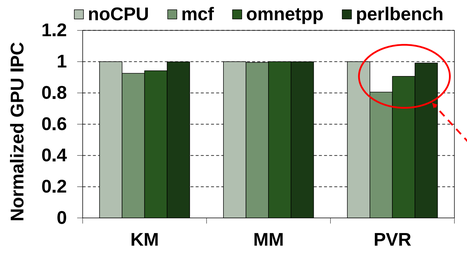
**Onur Kayıran**[1] Nachiappan CN[1] Adwait Jog[1] Rachata Ausavarungnirun[2]
Mahmut T. Kandemir[1] Gabriel H. Loh[3] Onur Mutlu[2] Chita R. Das[1]
[1]The Pennsylvania State University [2]Carnegie Mellon University [3]AMD Research

## Heterogeneous Architectures



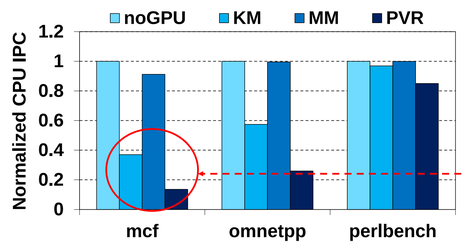

Latency optimized cores and throughput optimized cores share the memory hierarchy

## Effects of Application Interference




GPU applications are affected moderately due to CPU interference

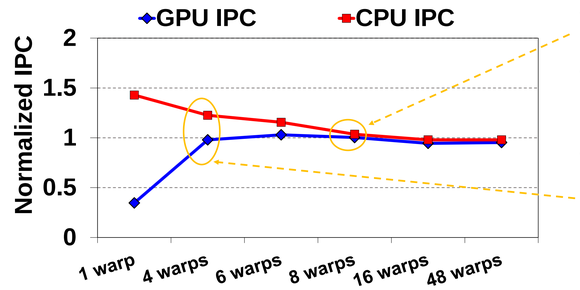
**Up to 20%**




CPU applications are affected significantly due to GPU interference

**Up to 85%**

## Latency Tolerance of CPUs vs. GPUs

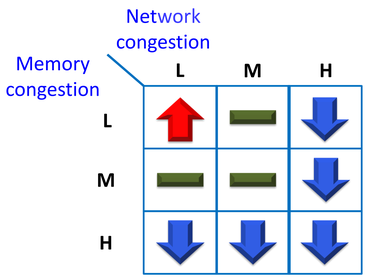



- Achieved by an existing GPU-based technique
- Effective for GPU performance
- 23% potential CPU improvements w/o significant performance loss for the GPU

- High GPU TLP causes memory and network congestion
- High memory congestion degrades CPU performance
- GPU cores can tolerate memory congestion due to multi-threading
- The optimal TLP for CPUs and GPUs might be different due to the disparity between latency tolerance of CPUs and GPUs

Problem: Existing GPU-based TLP management techniques for GPUs might not be effective in heterogeneous systems

## Scheme




### CPU-based Scheme: CM-CPU

- GPU TLP is reduced if memory or network congestion is high.
- Improves CPU performance.
- Might cause low latency tolerance for GPU cores.

GPU scheduler stalls can be high due to:

- High memory congestion
- Low latency tolerance due to low TLP

Increase # of warps | No change in # of warps | Decrease # of warps

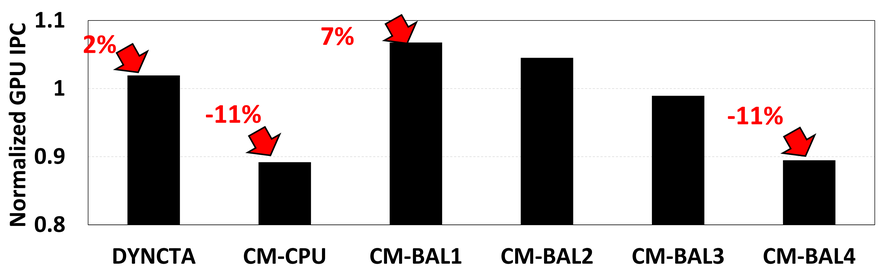
### CPU-GPU Balanced Scheme: CM-BAL

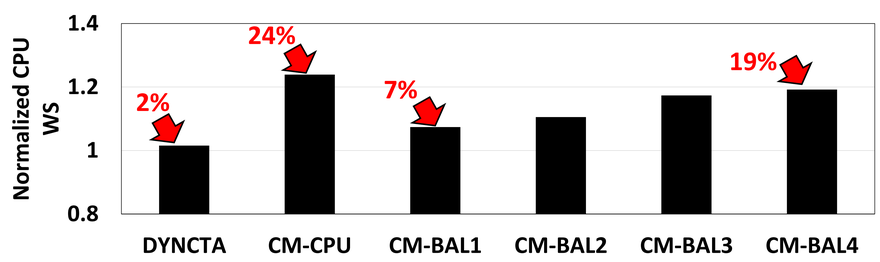
- GPU TLP is increased if GPU cores suffer from low latency tolerance.
- Provides balanced improvements.
- The CPU-GPU benefits trade-off can be controlled.

## Performance Benefits

CM-BAL1: Balanced improvements for both CPUs and GPUs

CM-BAL4: Tuned to favor CPU applications




## Summary

**Warp Scheduler Controls GPU Thread-Level Parallelism**

| | Improved GPU performance | Improved CPU performance |
|---|---|---|
| Existing Works | ☑ | × |
| CPU-based Scheme | × | ☑ |
| CPU-GPU Balanced Scheme | ☑ | ☑ |

+ control the trade-off