Work in Progress: Increasing Schedulability via on-GPU Scheduling

Joshua Bakita and James H. Anderson

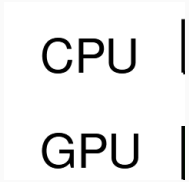
Department of Computer Science
University of North Carolina, Chapel Hill

Multiple tasks, one GPU

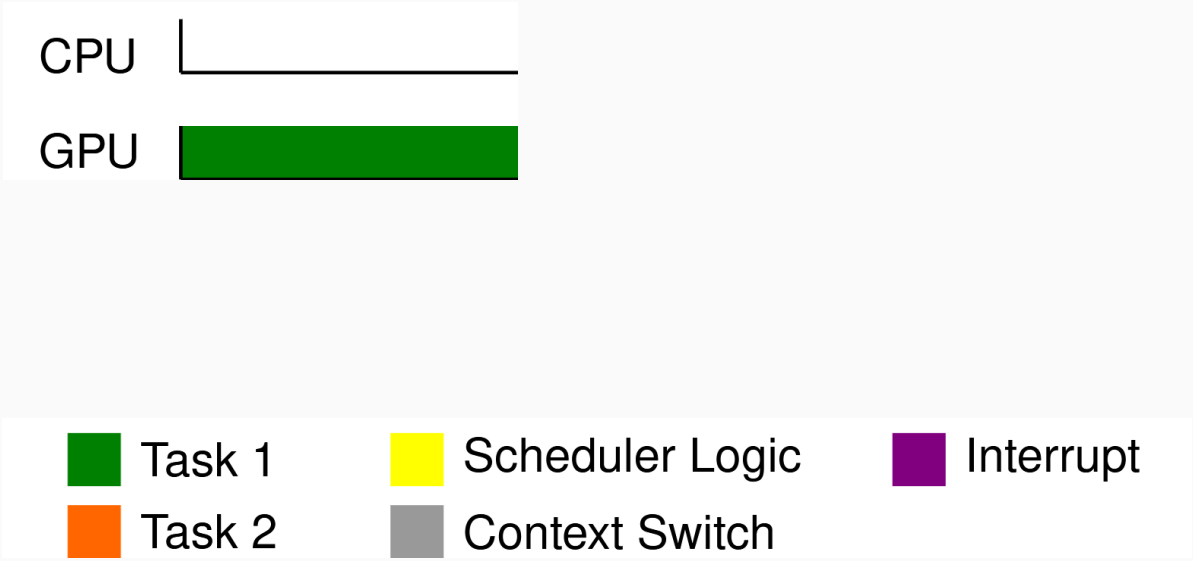
GPU Scheduling Assumptions Worth Revisiting…

Assumption: GPU scheduling overhead is negligible at runtime.

Assumption: GPU scheduling overhead is negligible at runtime.

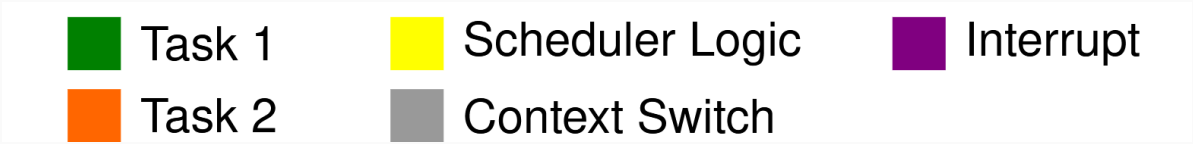


Assumption: GPU scheduling overhead is negligible at runtime.

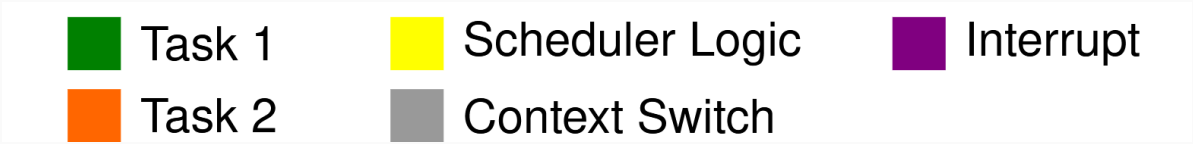


GPU Scheduling Assumptions Worth Revisiting…

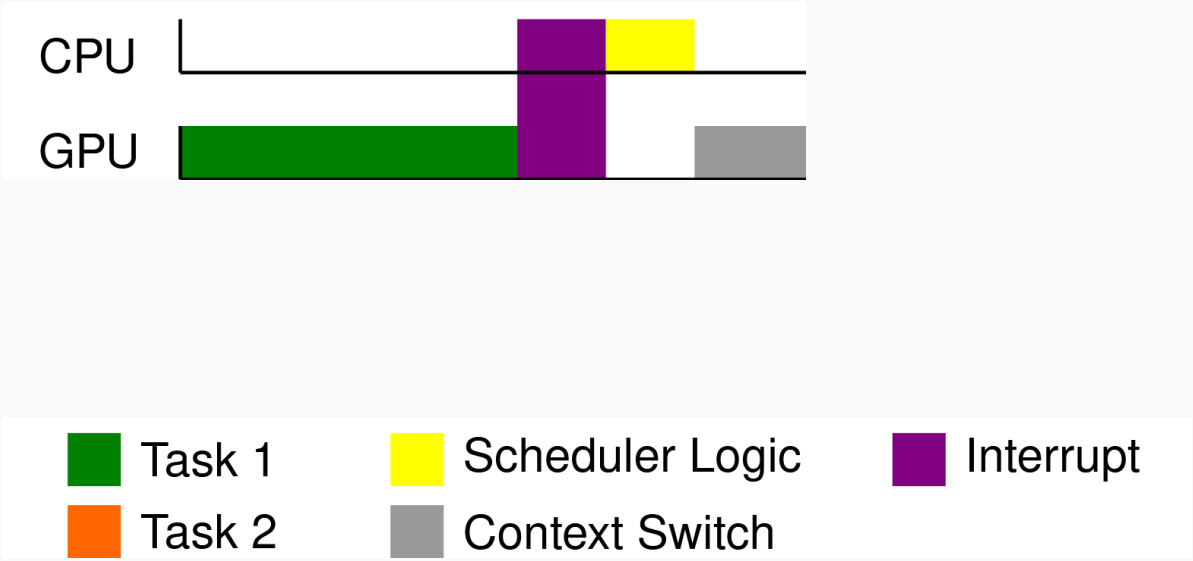
Assumption: GPU scheduling overhead is negligible at runtime.



Assumption: GPU scheduling overhead is negligible at runtime.

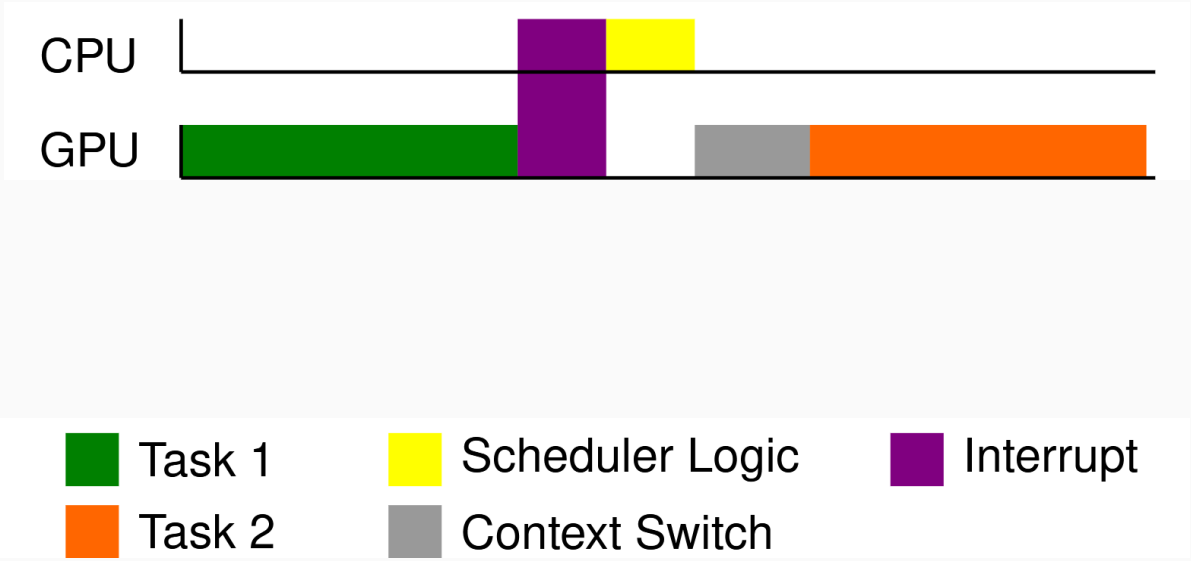


Assumption: GPU scheduling overhead is negligible at runtime.



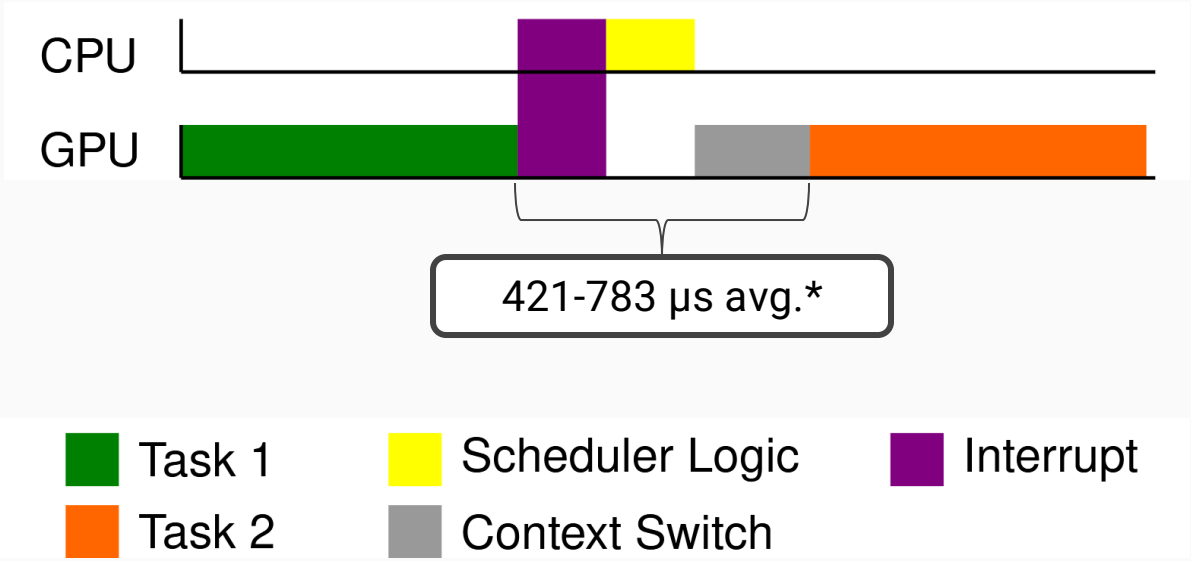
GPU Scheduling Assumptions Worth Revisiting…

Assumption: GPU scheduling overhead is negligible at runtime.



GPU Scheduling Assumptions Worth Revisiting…

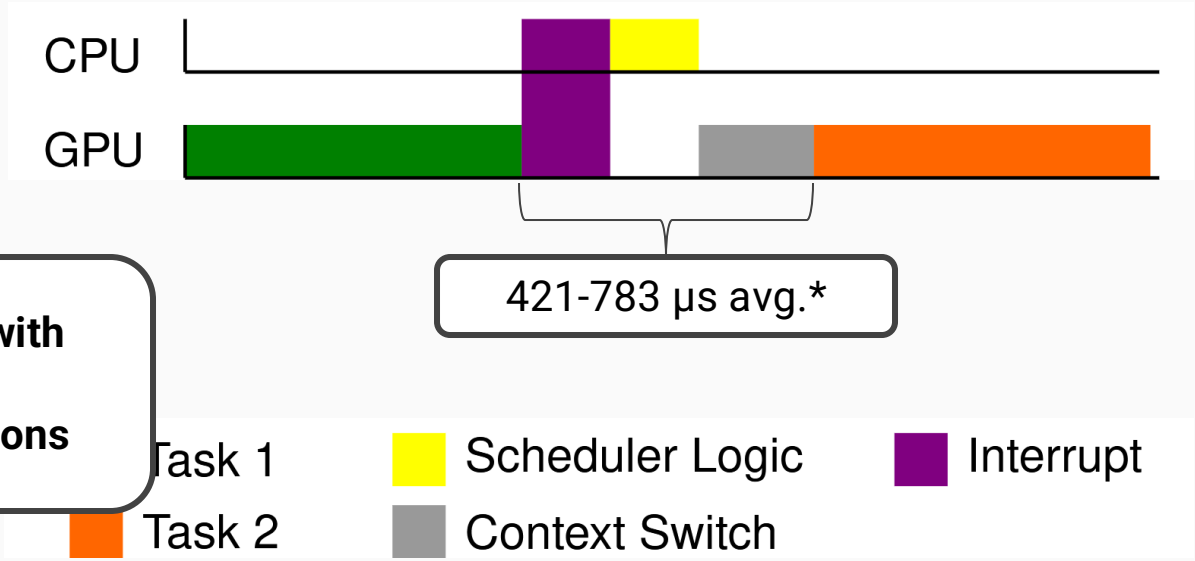
Assumption: GPU scheduling overhead is negligible at runtime.



*Y. Wang, *et al.*, "GCAPS: GPU Context-Aware Preemptive Priority-Based Scheduling for Real-Time Tasks", ECRTS'24

GPU Scheduling Assumptions Worth Revisiting…

Assumption: GPU scheduling overhead is negligible at runtime.



Reality: Not with current implementations

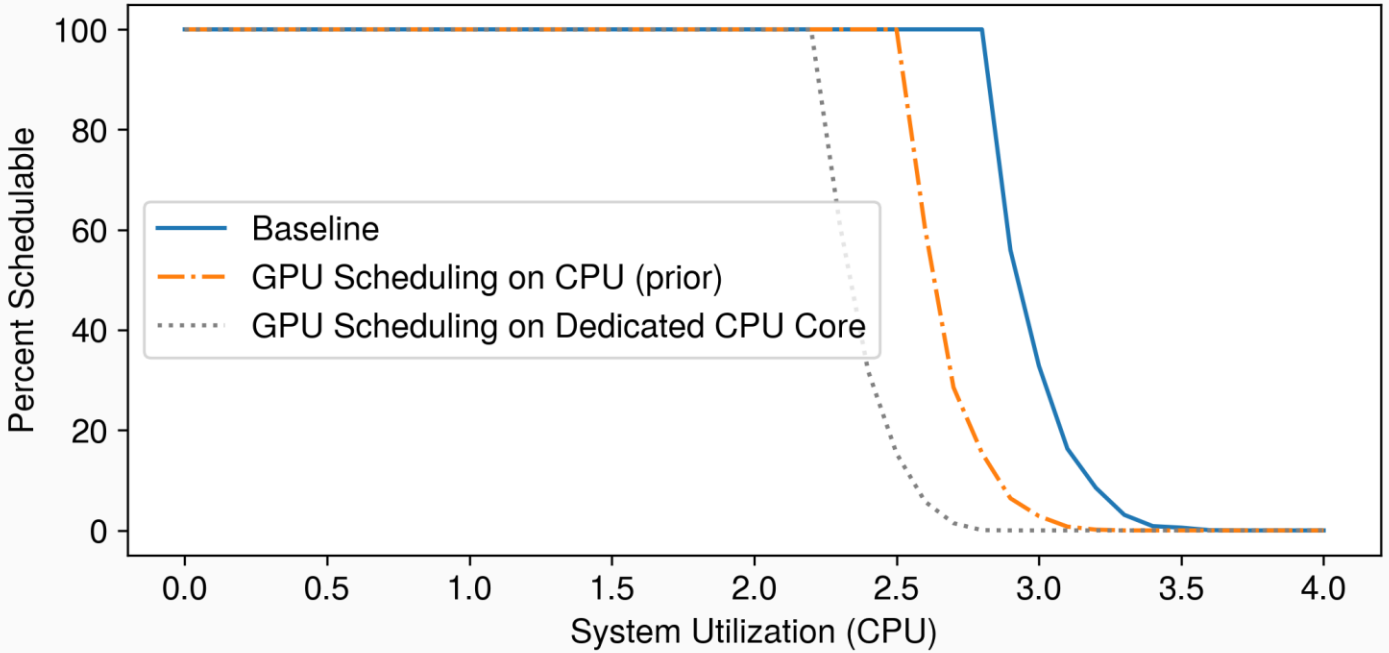
*Y. Wang, *et al.*, "GCAPS: GPU Context-Aware Preemptive Priority-Based Scheduling for Real-Time Tasks", ECRTS'24

GPU Scheduling Assumptions Worth Revisiting…

Assumption: GPU scheduling overhead is negligible analytically.

GPU Scheduling Assumptions Worth Revisiting…

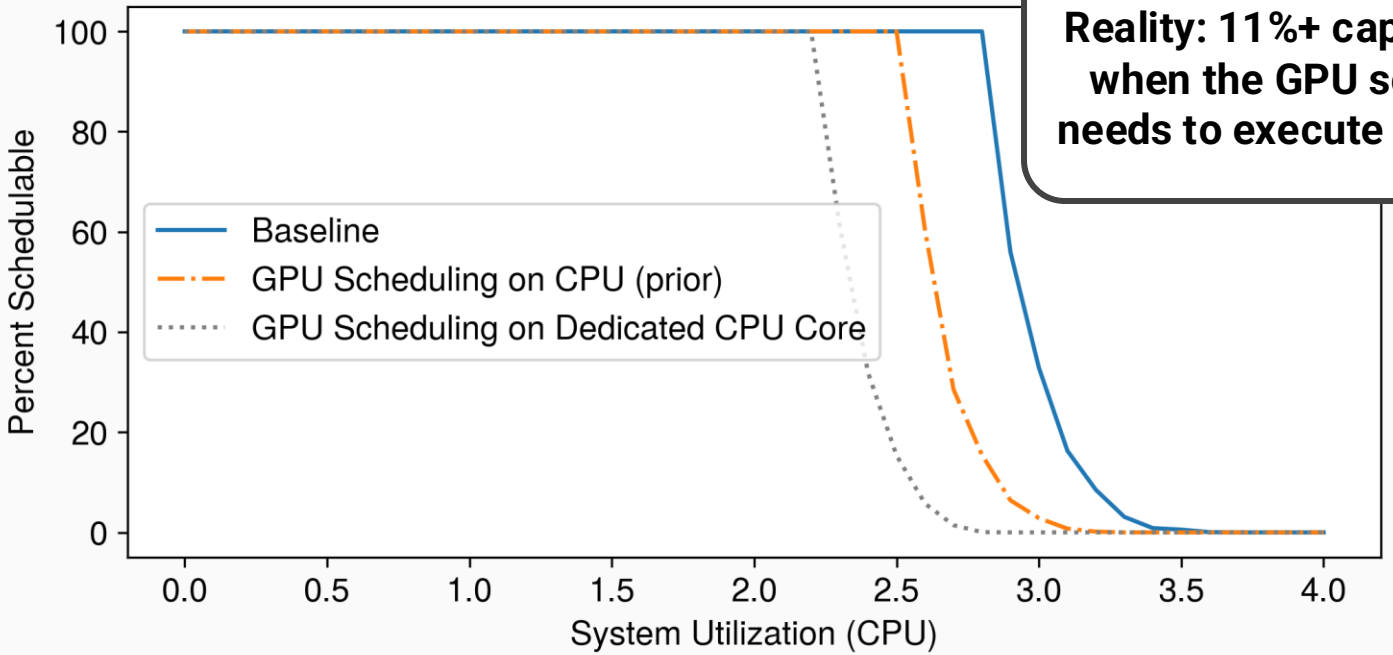
Assumption: GPU scheduling overhead is negligible analytically.



Quad-core system scheduled under G-EDF.

GPU Scheduling Assumptions Worth Revisiting…

Assumption: GPU scheduling overhead is negligible analytically

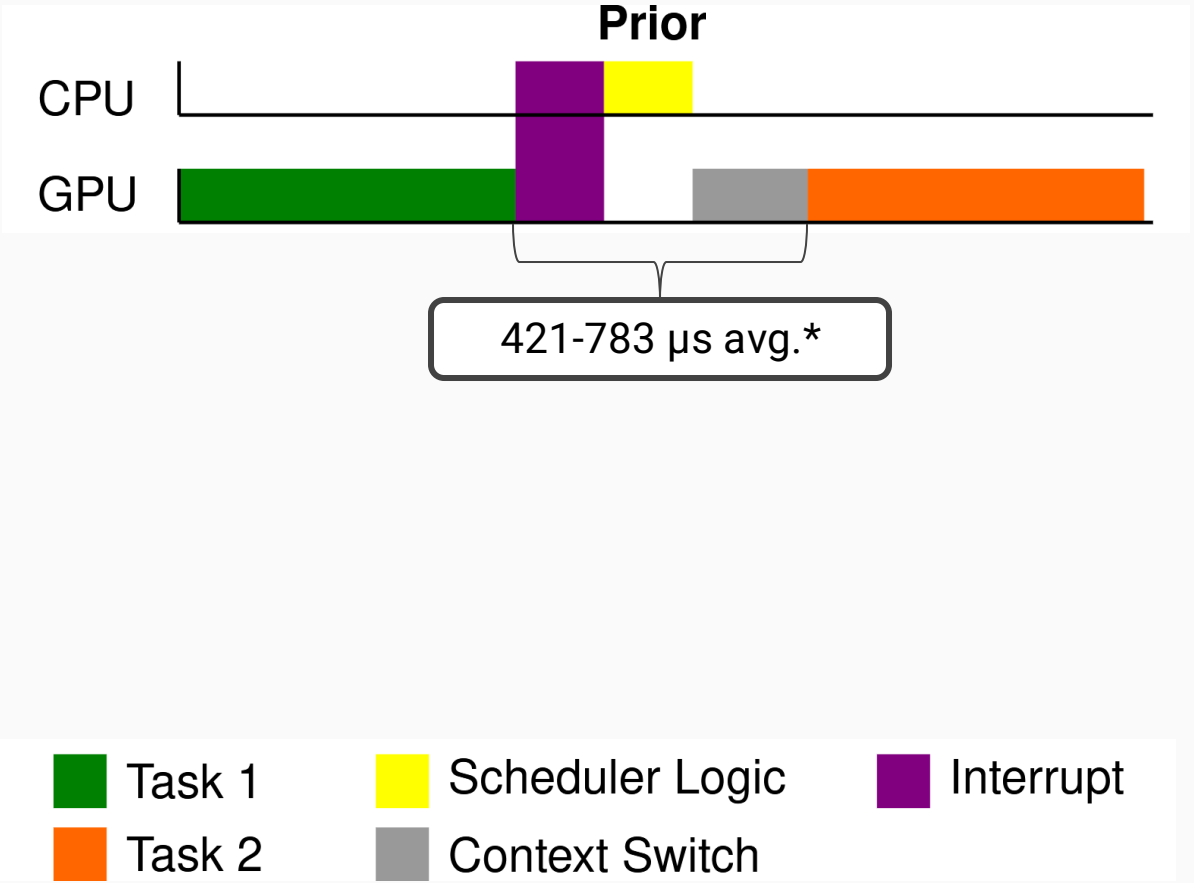


Reality: 11%+ capacity loss when the GPU scheduler needs to execute frequently

Quad-core system scheduled under G-EDF.

Solution

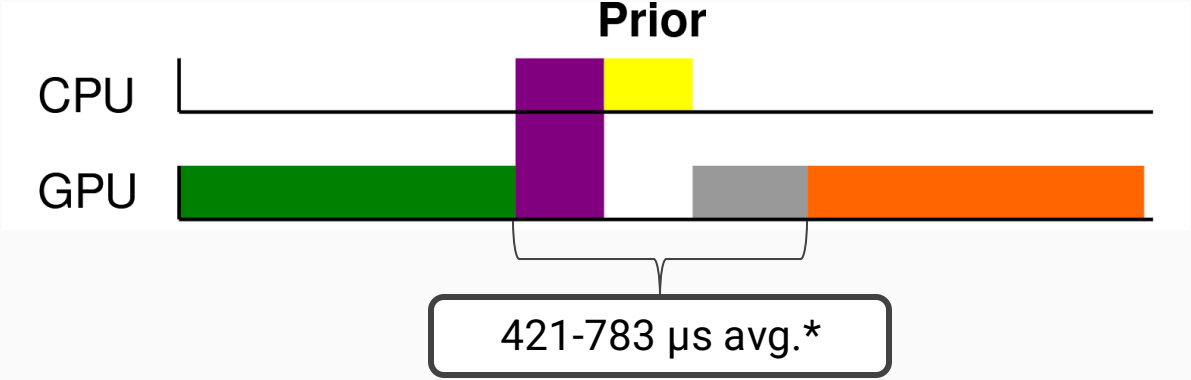
Scheduling On-GPU

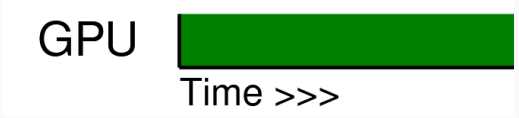


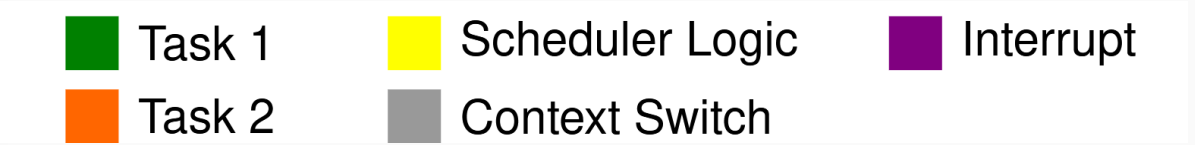
*Y. Wang, *et al.*, "GCAPS: GPU Context-Aware Preemptive Priority-Based Scheduling for Real-Time Tasks", ECRTS'24

Solution

Scheduling On-GPU



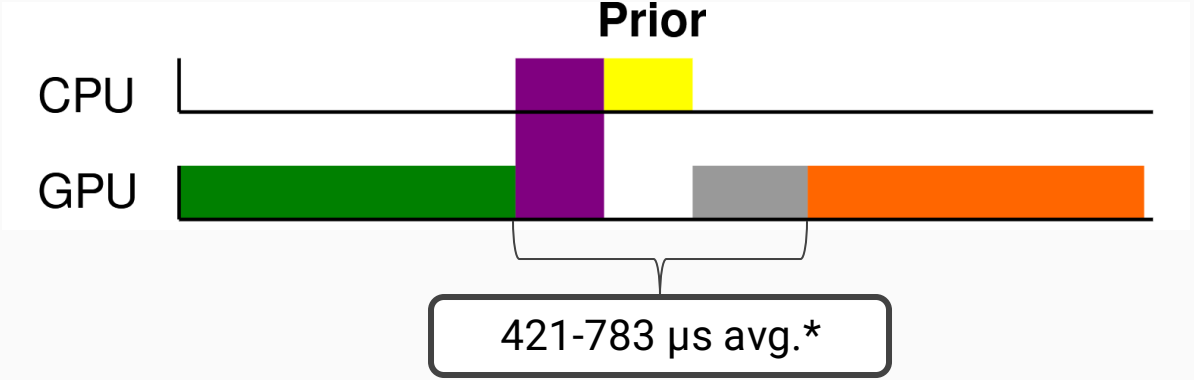




*Y. Wang, *et al.*, “GCAPS: GPU Context-Aware Preemptive Priority-Based Scheduling for Real-Time Tasks”, ECRTS’24

Solution

Scheduling On-GPU

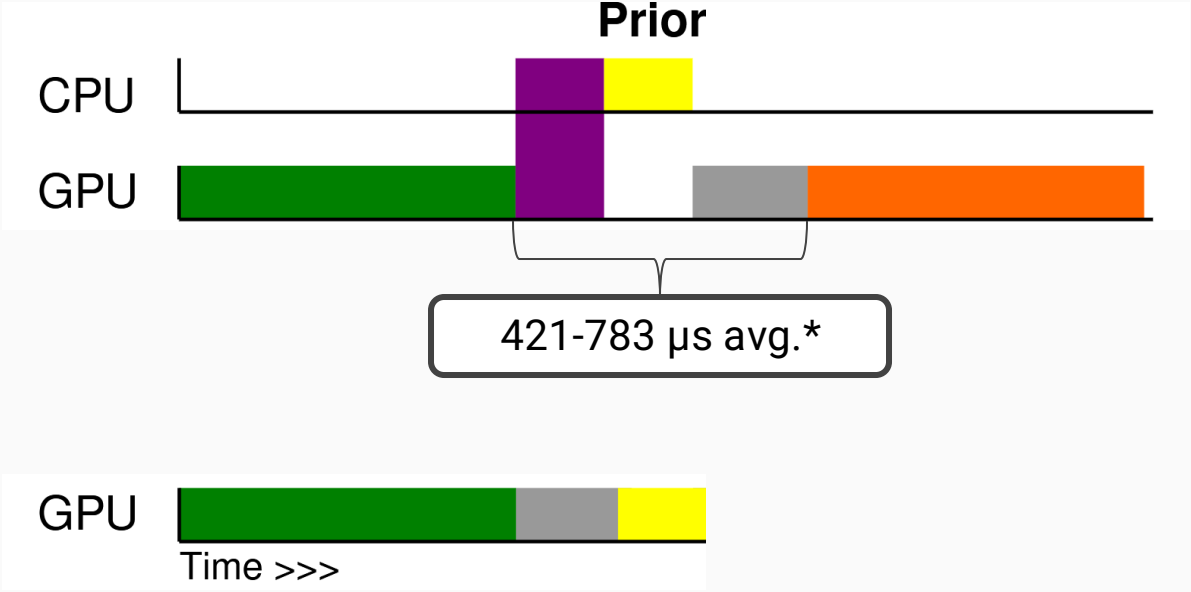


Task 1 | Scheduler Logic | Interrupt
Task 2 | Context Switch

*Y. Wang, *et al.*, “GCAPS: GPU Context-Aware Preemptive Priority-Based Scheduling for Real-Time Tasks”, ECRTS’24

Solution

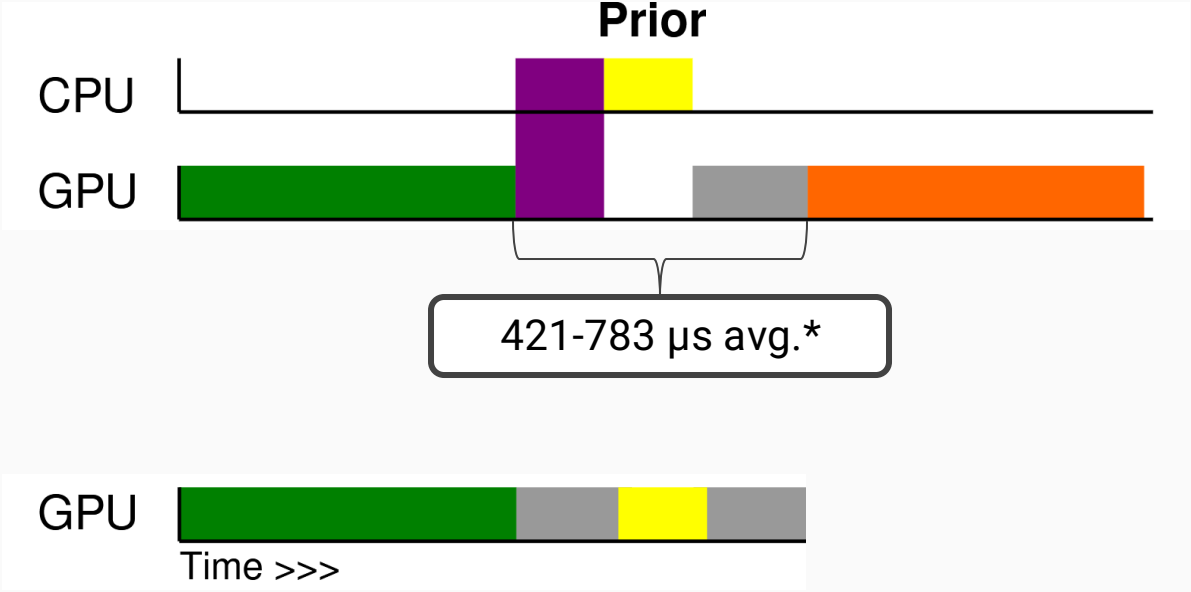
Scheduling On-GPU



*Y. Wang, *et al.*, “GCAPS: GPU Context-Aware Preemptive Priority-Based Scheduling for Real-Time Tasks”, ECRTS’24

Solution

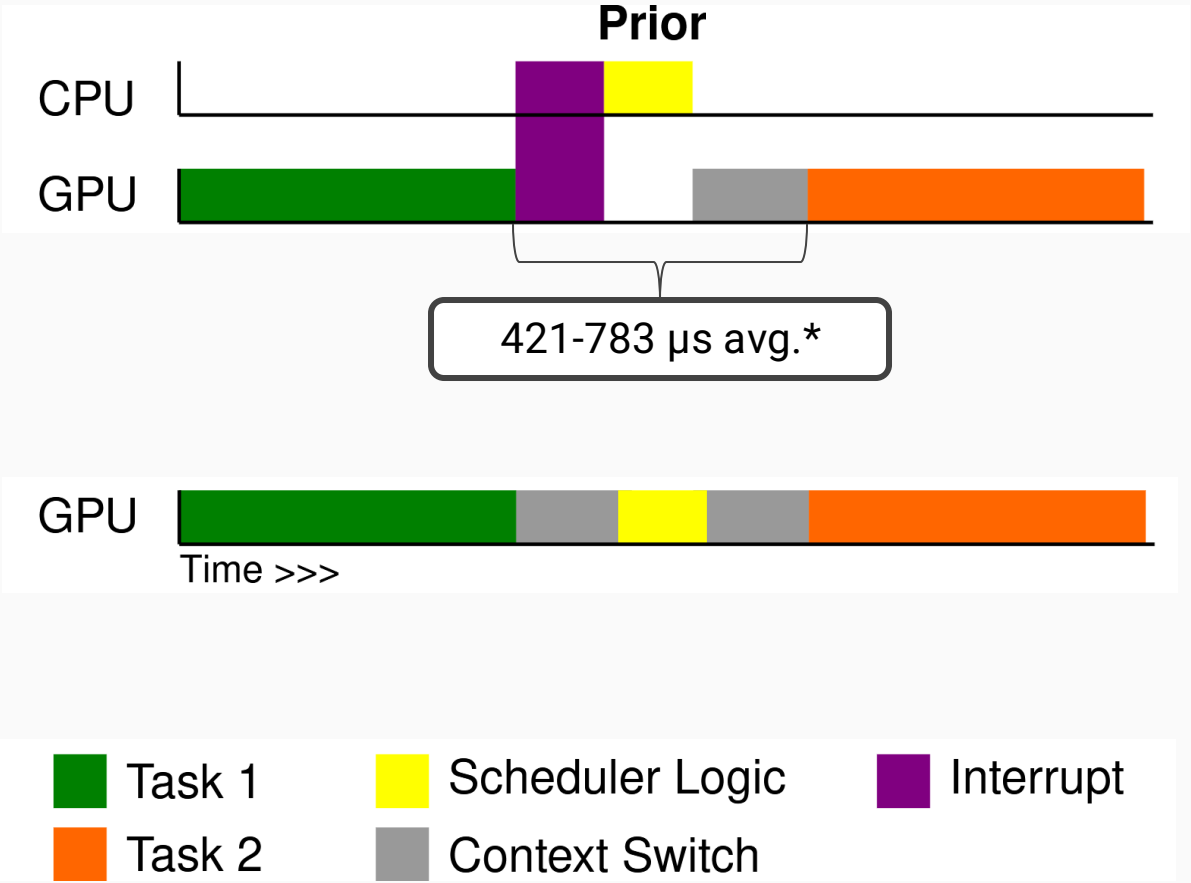
Scheduling On-GPU



*Y. Wang, *et al.*, "GCAPS: GPU Context-Aware Preemptive Priority-Based Scheduling for Real-Time Tasks", ECRTS'24

Solution

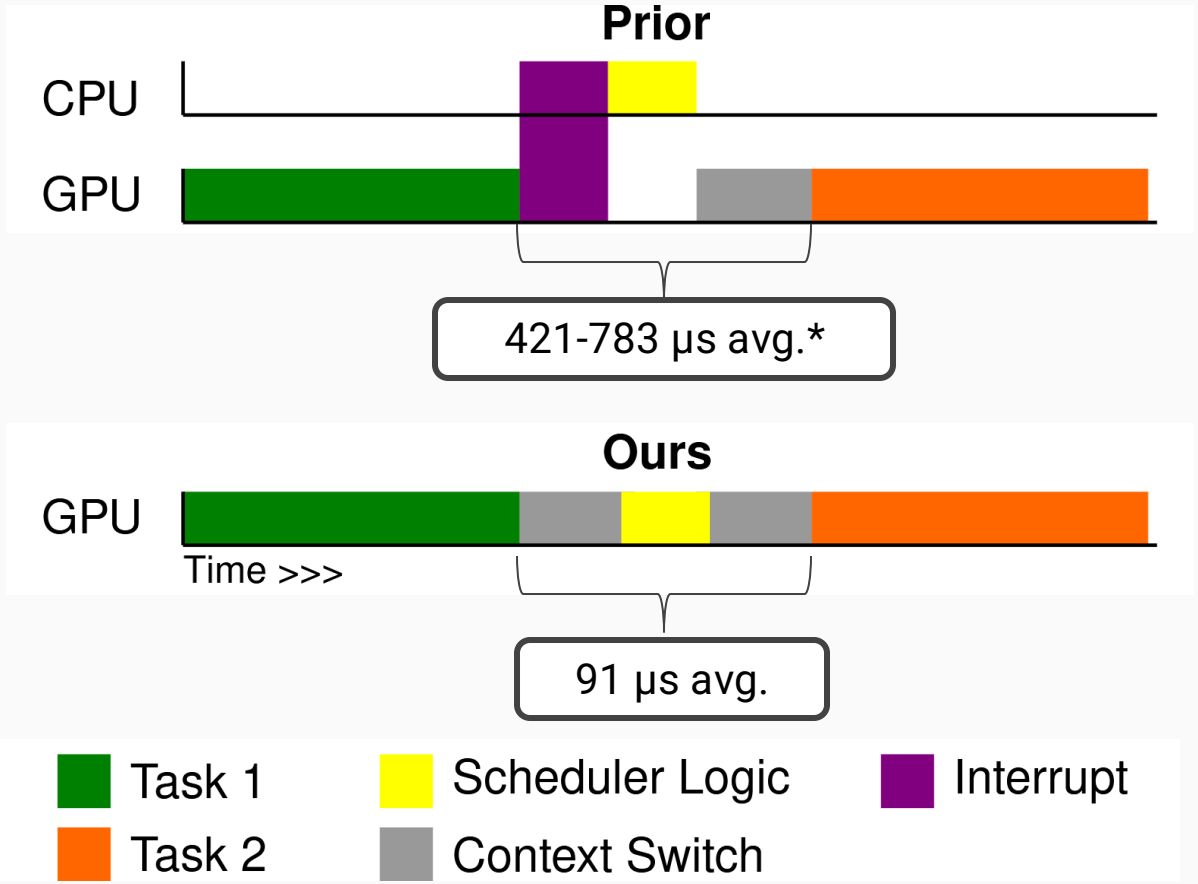
Scheduling On-GPU



*Y. Wang, *et al.*, “GCAPS: GPU Context-Aware Preemptive Priority-Based Scheduling for Real-Time Tasks”, ECRTS’24

Solution

Scheduling On-GPU

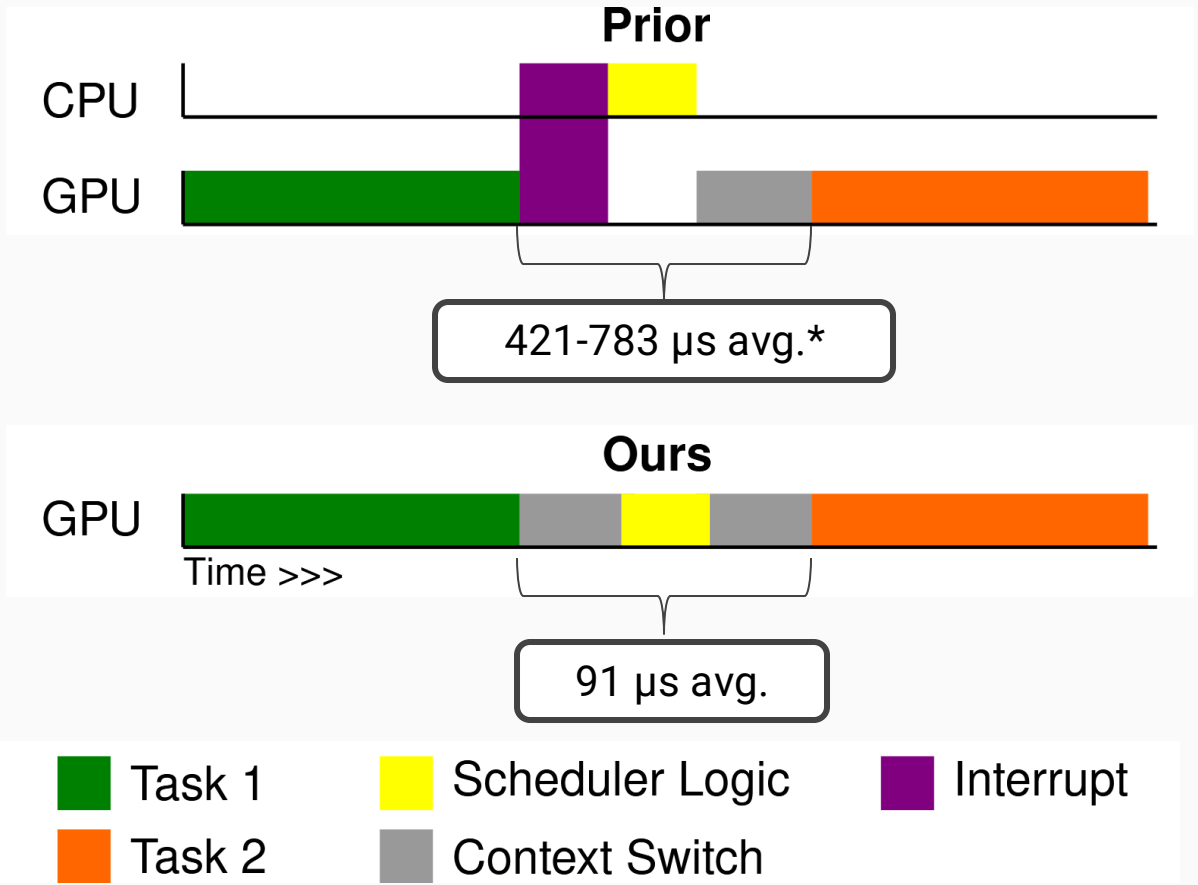


*Y. Wang, *et al.*, "GCAPS: GPU Context-Aware Preemptive Priority-Based Scheduling for Real-Time Tasks", ECRTS'24

Solution

Scheduling On-GPU

Key insight:
Scheduling from on-GPU cuts overhead from >20% to **<5%**



*Y. Wang, *et al.*, "GCAPS: GPU Context-Aware Preemptive Priority-Based Scheduling for Real-Time Tasks", ECRTS'24

Thank you!

Come visit the poster for questions.

Contact:
Email: jbakita@cs.unc.edu
X: @JJBakita
Web: https://cs.unc.edu/~jbakita

Old Well, University of North Carolina at Chapel Hill, Winter 2017