Deadline-aware Offloading for High Throughput Accelerators

Tsung Tai Yeh, Matthew D. Sinclair,
Bradford M. Beckmann, Timothy G. Rogers

國立交通大學
National Chiao Tung University

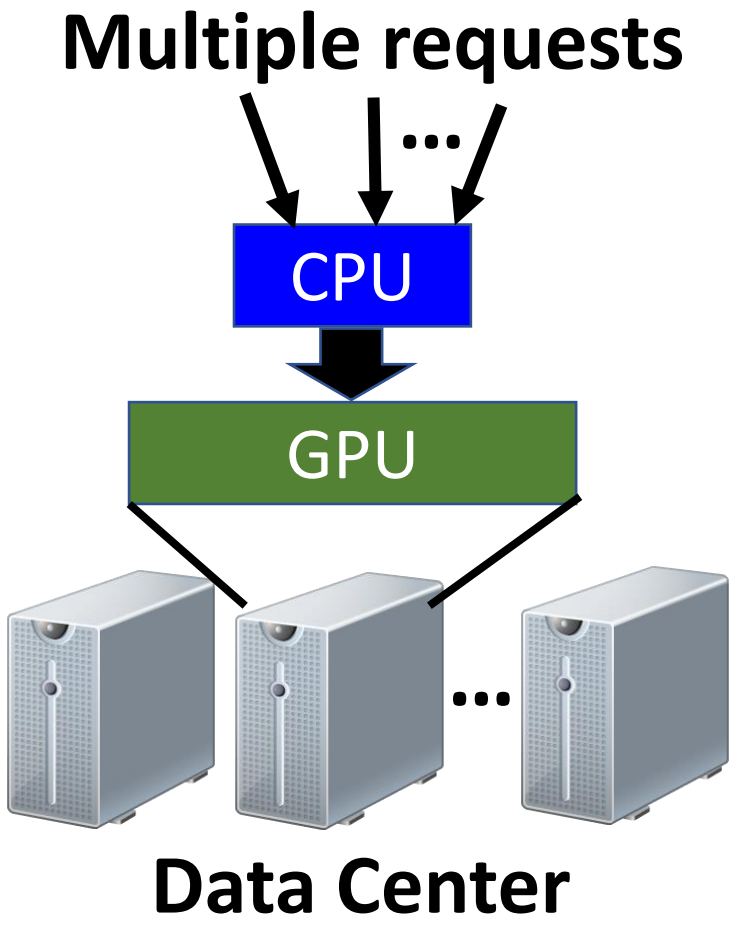
WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

AMD

PURDUE
UNIVERSITY

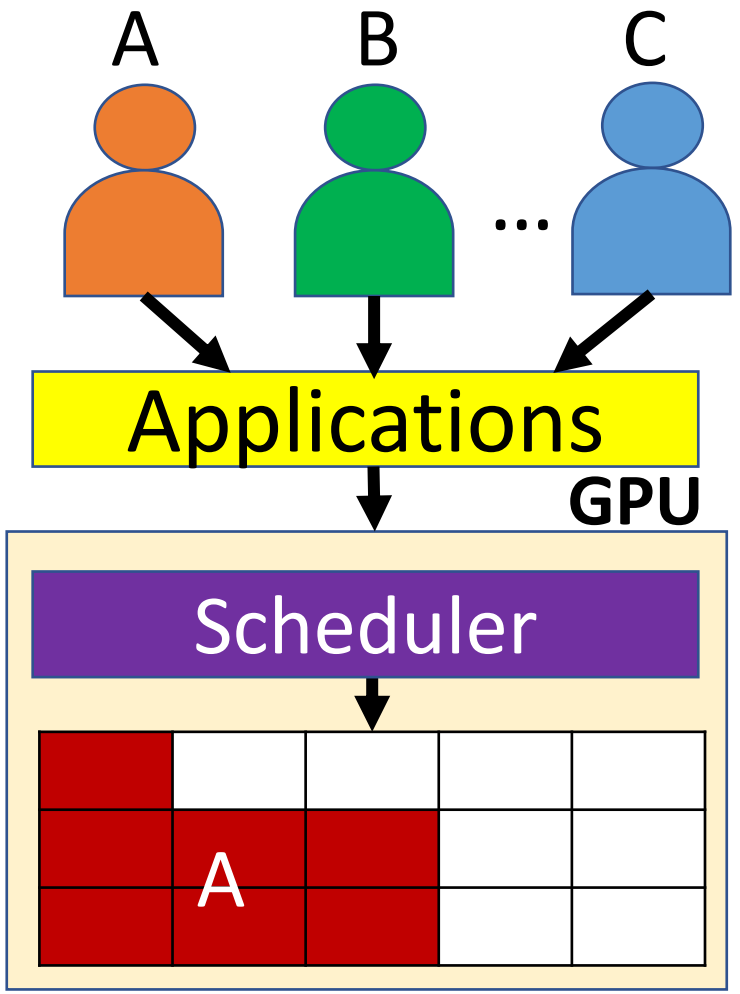
Motivation

- **Emerging data center workloads**
  - Compute-intensive
  - Highly data parallel
  - Have tight deadlines
  - GPUs increasingly used at data centers
- **Applications**
  - Network processing
  - DNN inference and others
- **GPU streams**
  - Concurrent kernel execution
  - Improves occupancy but difficult to meet different deadlines

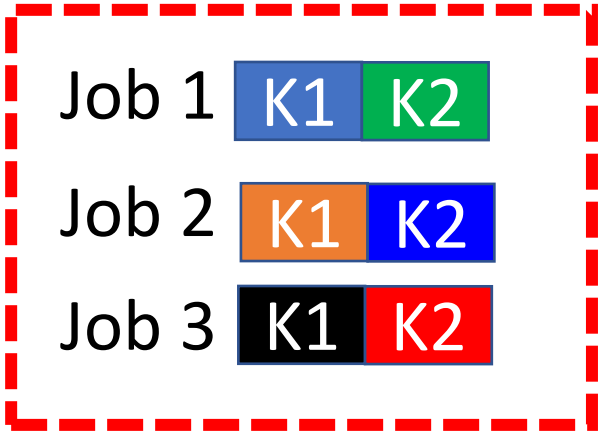


Motivation

- **Medium parallelism**
  - A single job cannot fully utilize entire GPU
- **GPU inefficient for latency-driven workloads**
  - High host scheduling overhead
  - Static priority assigned by programmers
- **Requirement**
  - Need to carefully co-schedule requests

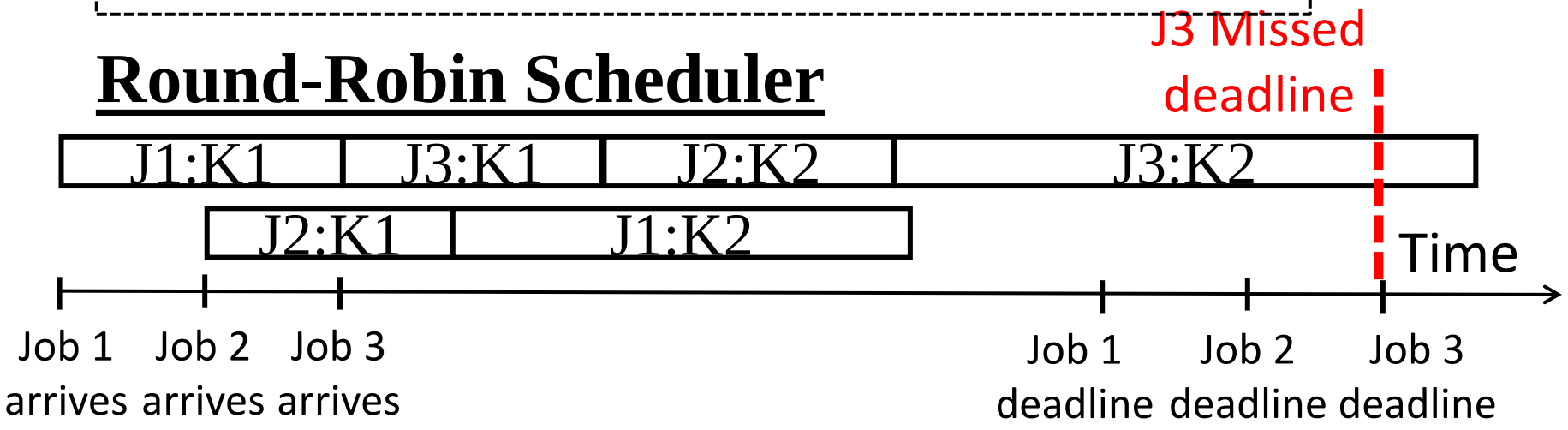


Key Challenge 1



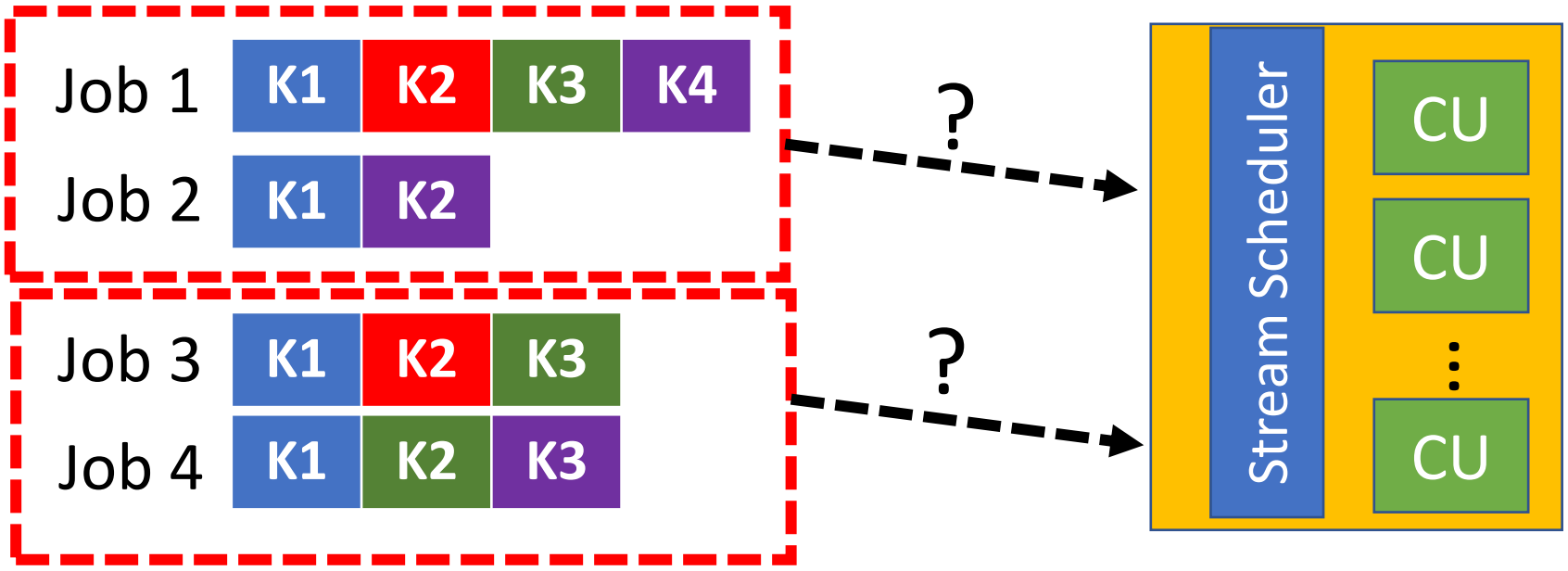
- How to decide job priorities?
  - QoS constraints for laxity-sensitive applications
  - Multiple jobs contend for GPU resources
  - Static priorities can be overly conservative

3 jobs, each with 2 kernels
Assume GPU can execute 2 kernels simultaneously



Key Challenge 2

- How to avoid oversubscribing the GPU?
  - Slow system response makes it difficult to meet real-time deadlines
  - **Challenge 2A:** How many jobs should be picked?
  - **Challenge 2B:** Which job should be chosen?



Our Goal

Minimize the number of jobs that miss their deadlines while **maximizing** the GPU utilization

We don’t explain how LAX works here and encourage viewers to watch the longer talk and read the paper to learn more.

Evaluation Methodology

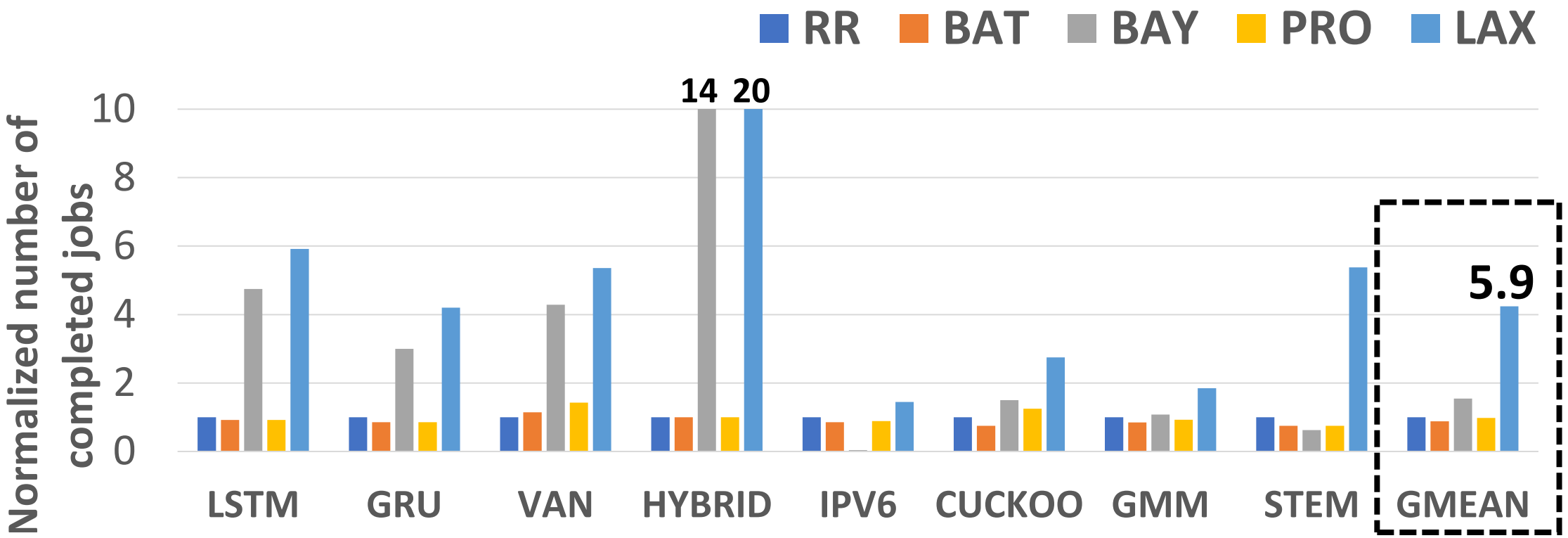
- **Simulator:** gem5-APU
  - 8 CUs, 4 SIMD units per CU
  - 128 compute queues
  - Up to 10 wavefronts per CU
  - **Extensive comparisons** to 10 other job schedulers
- **Workloads:**
  - DeepBench RNNs (Vanilla, GRU, LSTM, Hybrid)
  - G-Opt (Networking: CUCKOO, IPV6)
  - Lucida (IPA: GMM, Stemmer)
  - Each application has different real-time deadlines
  - High, medium, and low arrival rates (exponential distribution)

Extensive comparisons to other Scheduling

- **CPU-side Scheduling**
  - BatchMaker **(BAT)** [Gao et. al., EuroSys '18]
  - Baymax **(BAY)** [Chen et. al., ASPLOS, '16]
  - Prophet **(PRO)** [Chen et. al., ASPLOS '17]
- **CP-extension Scheduling**
  - Multi-Level Feedback Queue **(MLFQ)**
  - Shortest-Job First **(SJF)**
  - Shortest Remaining Time Job First **(SRF)**
  - Longest-Job First **(LJF)**
  - Earliest Deadline First **(EDF)**
  - **PREMA** [Choi et. al. HPCA '20]

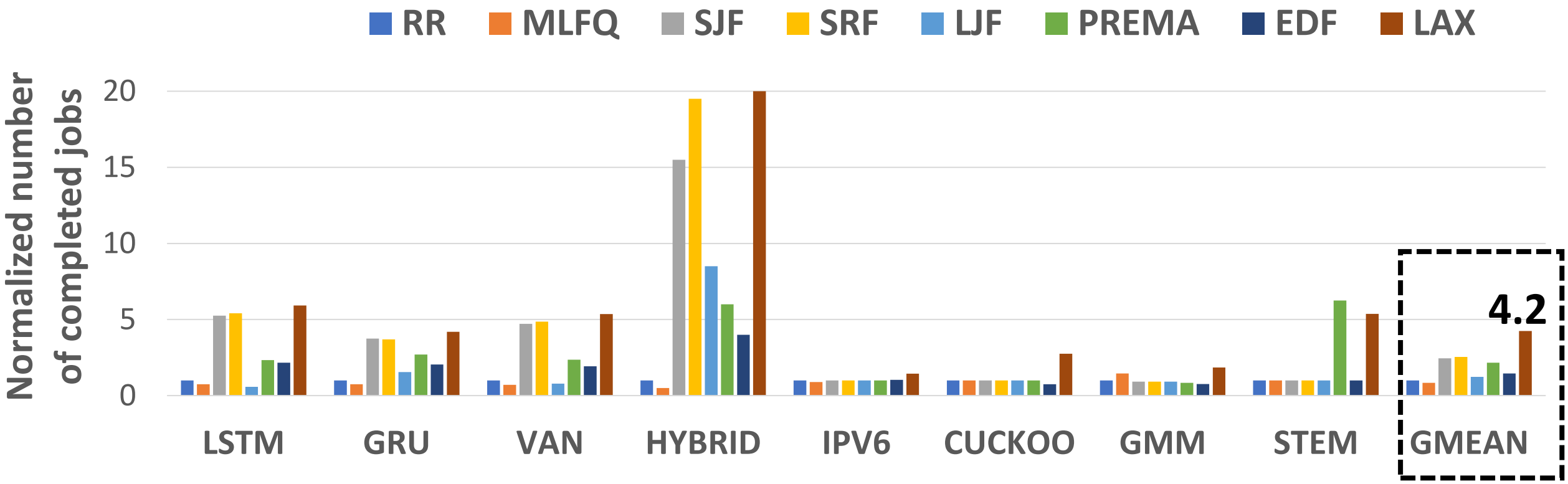
- **CPU-side scheduling** return to CPU to schedule jobs
- **CP-extension scheduling** extend the GPUs Command Processor (CP)

CPU-side Scheduling Performance



LAX up to **5.9X** geomean better than CPU-side schedulers at the high job arrival rate

CP-extension Scheduling Performance



LAX up to **4.2X** geomean better than other schedulers that extend CP at the high job arrival rate

Conclusion

- **Emerging GPU applications have different characteristics**
  - Real-time constraints, medium amount of parallelism
- **Opportunity**
  - Using stream scheduler to execute jobs simultaneously
- **Problems:**
  - How to decide the priority of jobs?
  - How many jobs should be offloaded?
- **More intelligent scheduler:** Laxity-aware scheduling
  - Predict job completion time and queuing delay
  - Dynamically change job priorities based on their laxity
- **Results:** Complete 1.7X – 5.9X more jobs by their deadlines

Copyright Disclosure

Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information


AMD