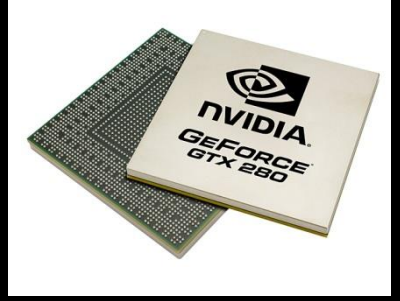
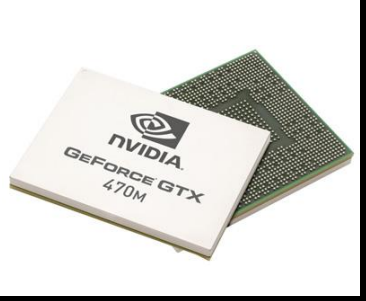





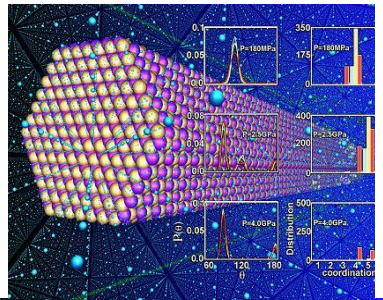
TimeGraph: GPU Scheduling for Real-Time Multi-Tasking Environments

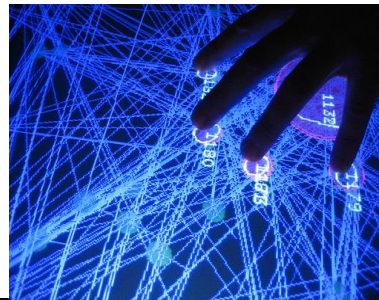
Shinpei Kato*, Karthik Lakshmanan*, Raj Rajkumar*, and Yutaka Ishikawa**

* Carnegie Mellon University ** The University of Tokyo



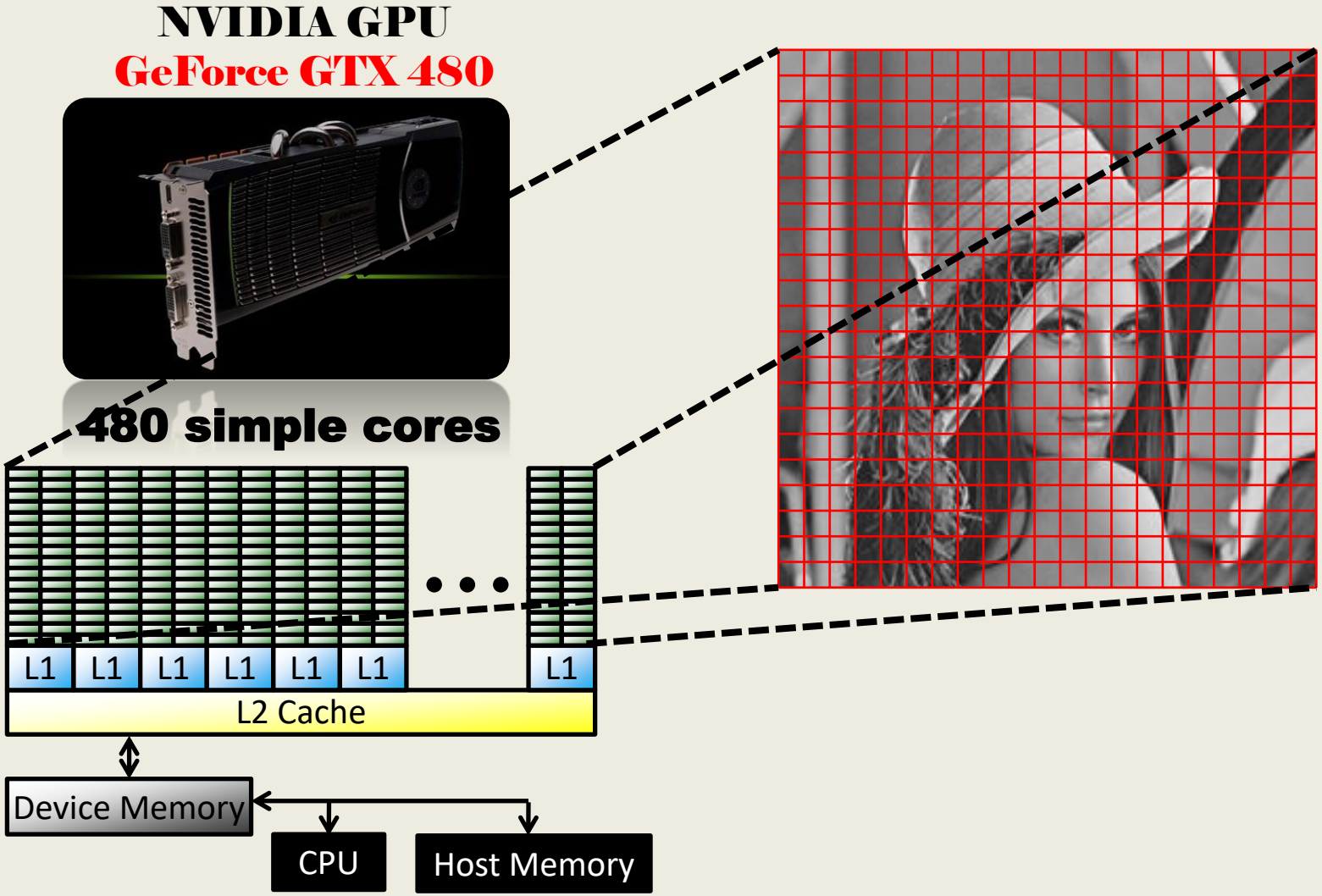


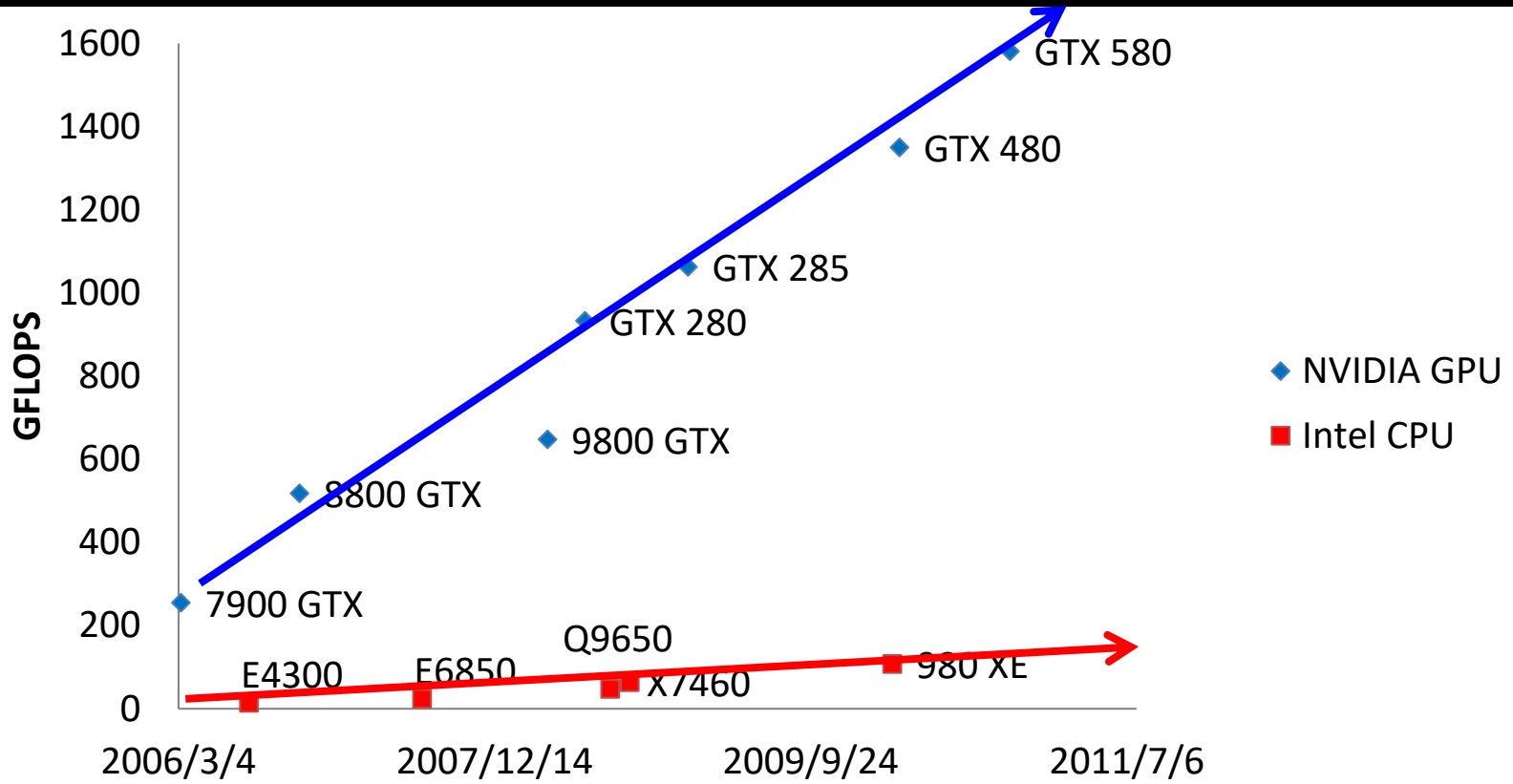


Graphics Applications

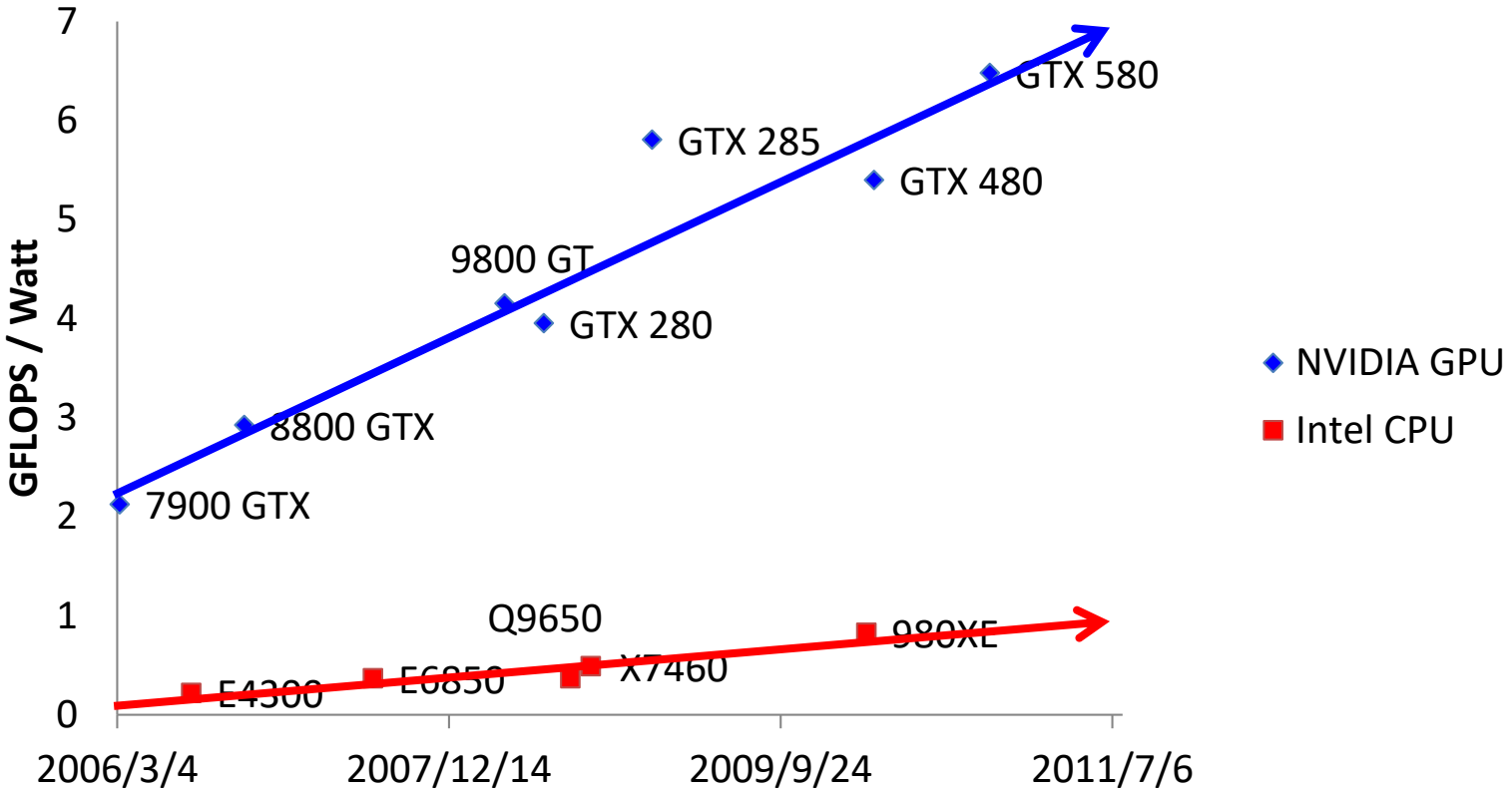
Graphics Processing Unit (GPU)



Peak Performance



Peak Performance "per Watt"



General-Purpose Computing on GPU (GPGPU)

3-D On-line Game

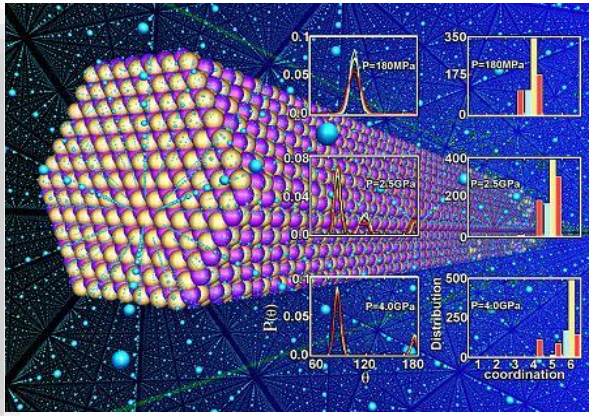
Autonomous Driving

Virtual Reality

3-D Interface

Computer Vision

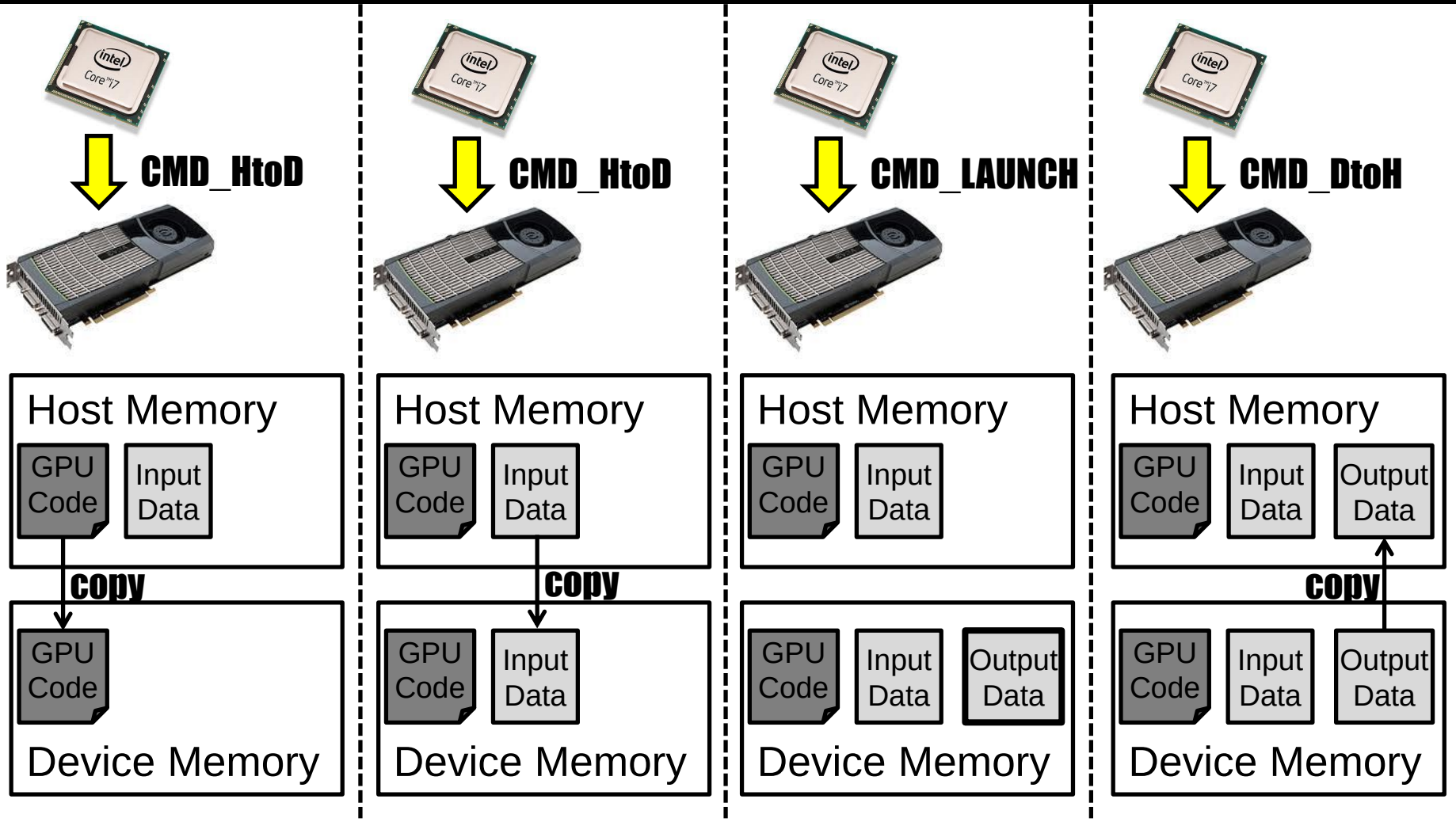
Scientific Simulation



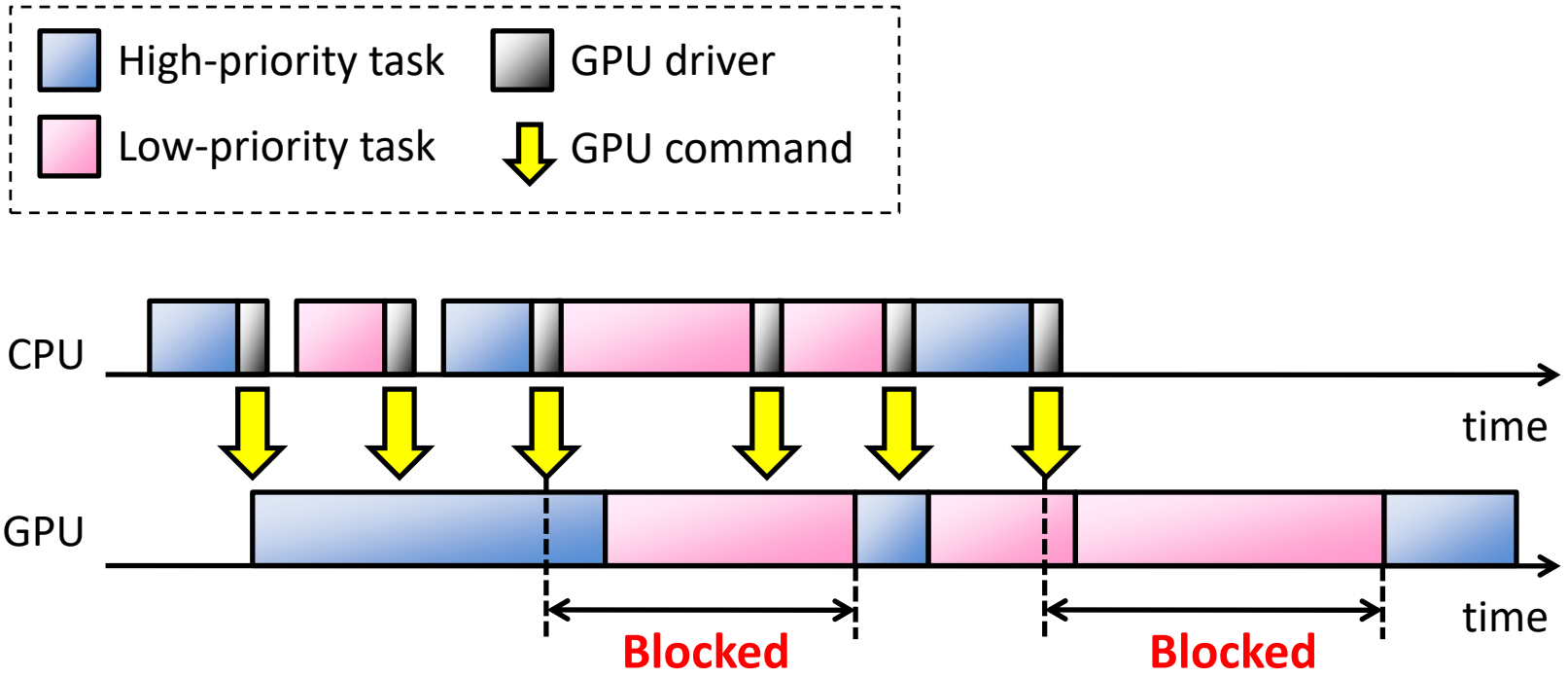
Outline

1. Introduction
2. What's Problem
3. Our Solution – "TimeGraph"
4. Evaluation
5. Summary

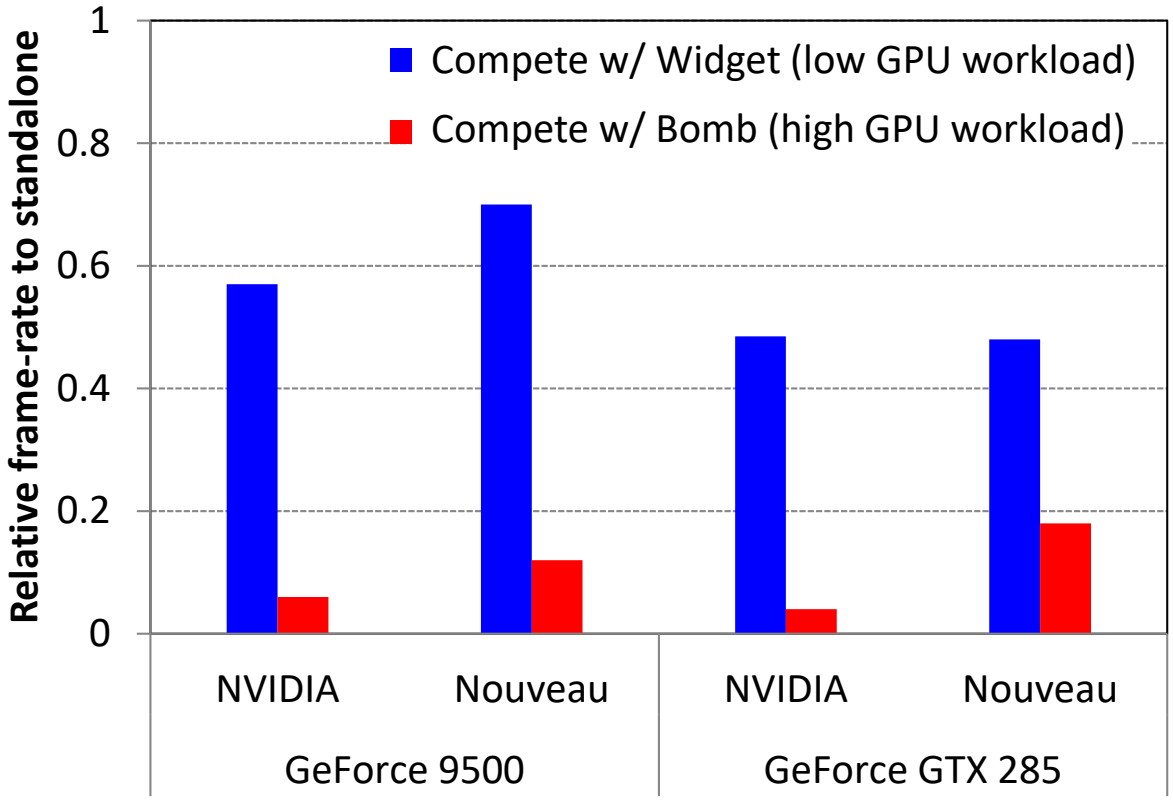
GPU Is Command-Driven



Multi-Tasking Problem



Impact of Interference



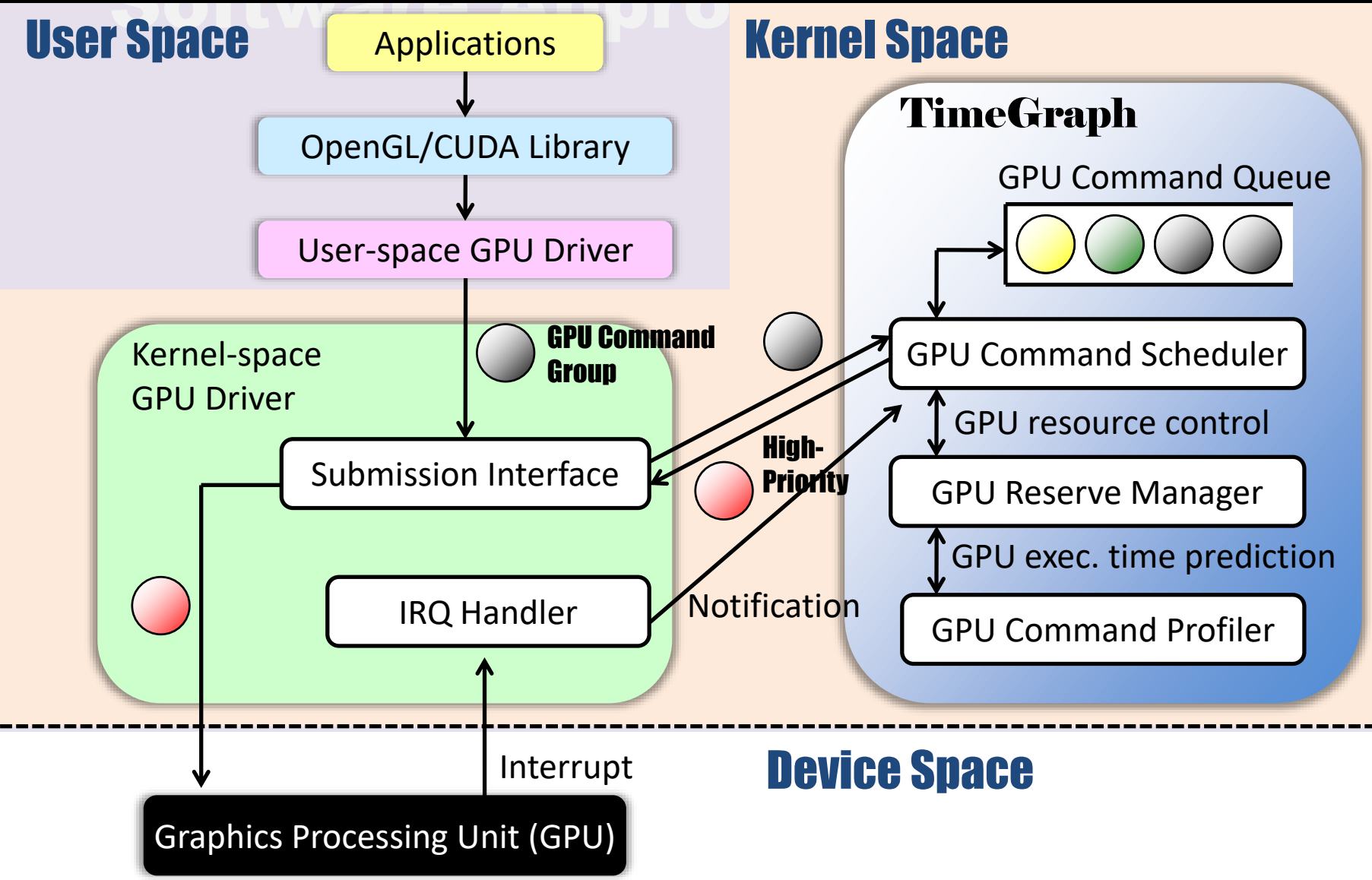
Observe Frame Rate of OpenArena (3-D Game) on Linux

NVIDIA proprietary driver
Nouveau open-source driver

Outline

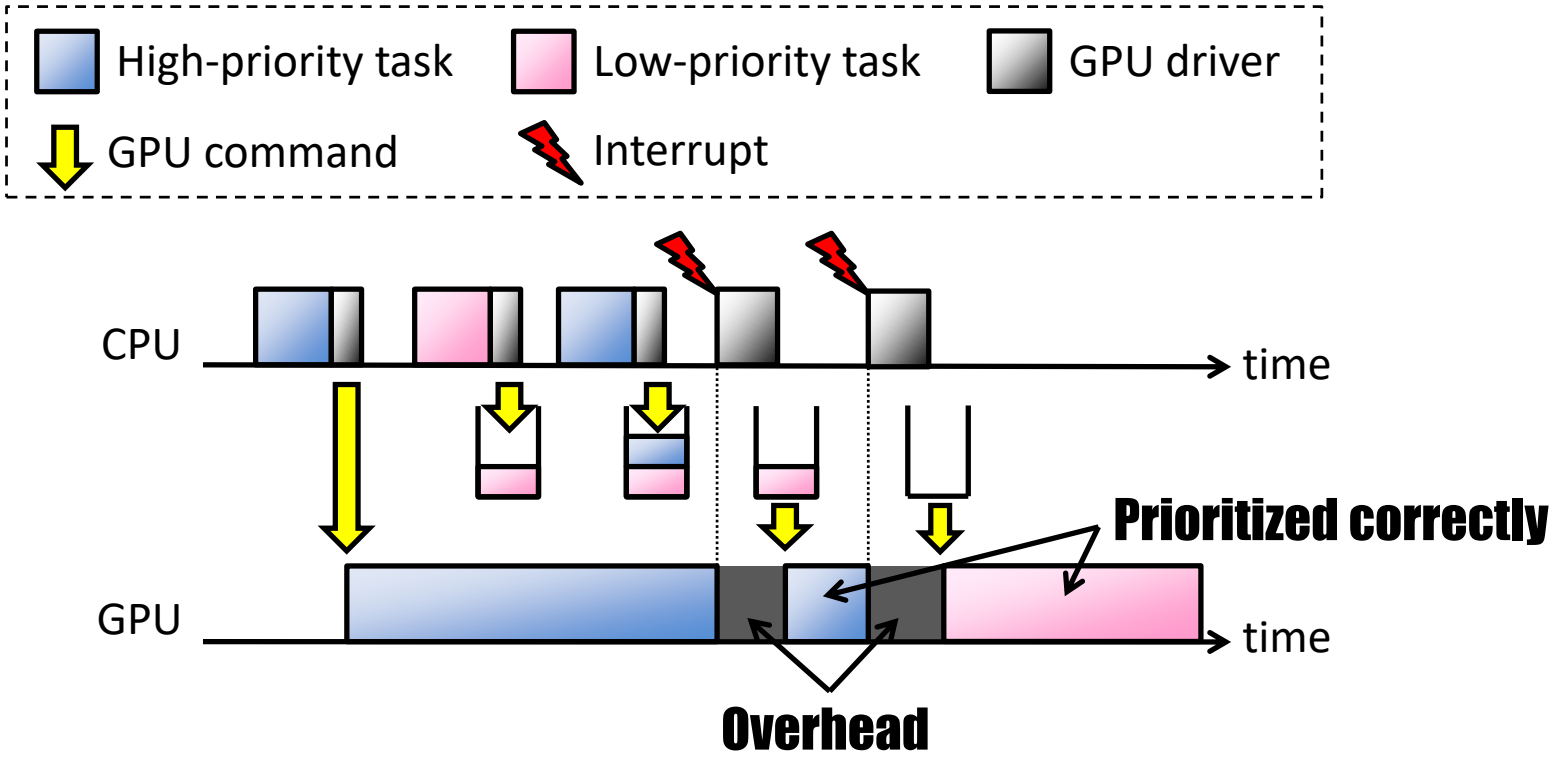
1. Introduction
2. What's Problem
3. Our Solution – "TimeGraph"
4. Evaluation
5. Summary

TimeGraph Architecture



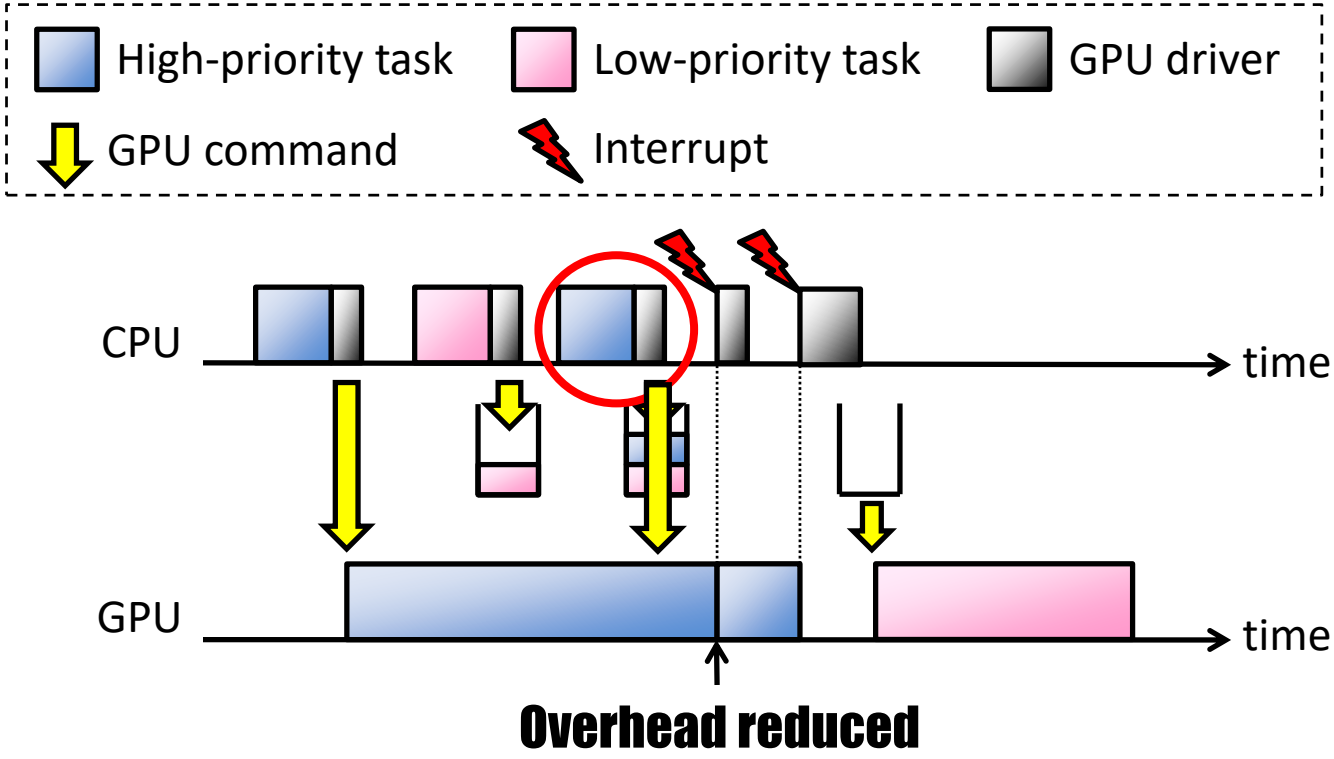
Priority Support – Predictable Response Time (PRT) Policy

- When GPU is not idle, GPU commands are queued
- When GPU gets idle, GPU commands are dispatched



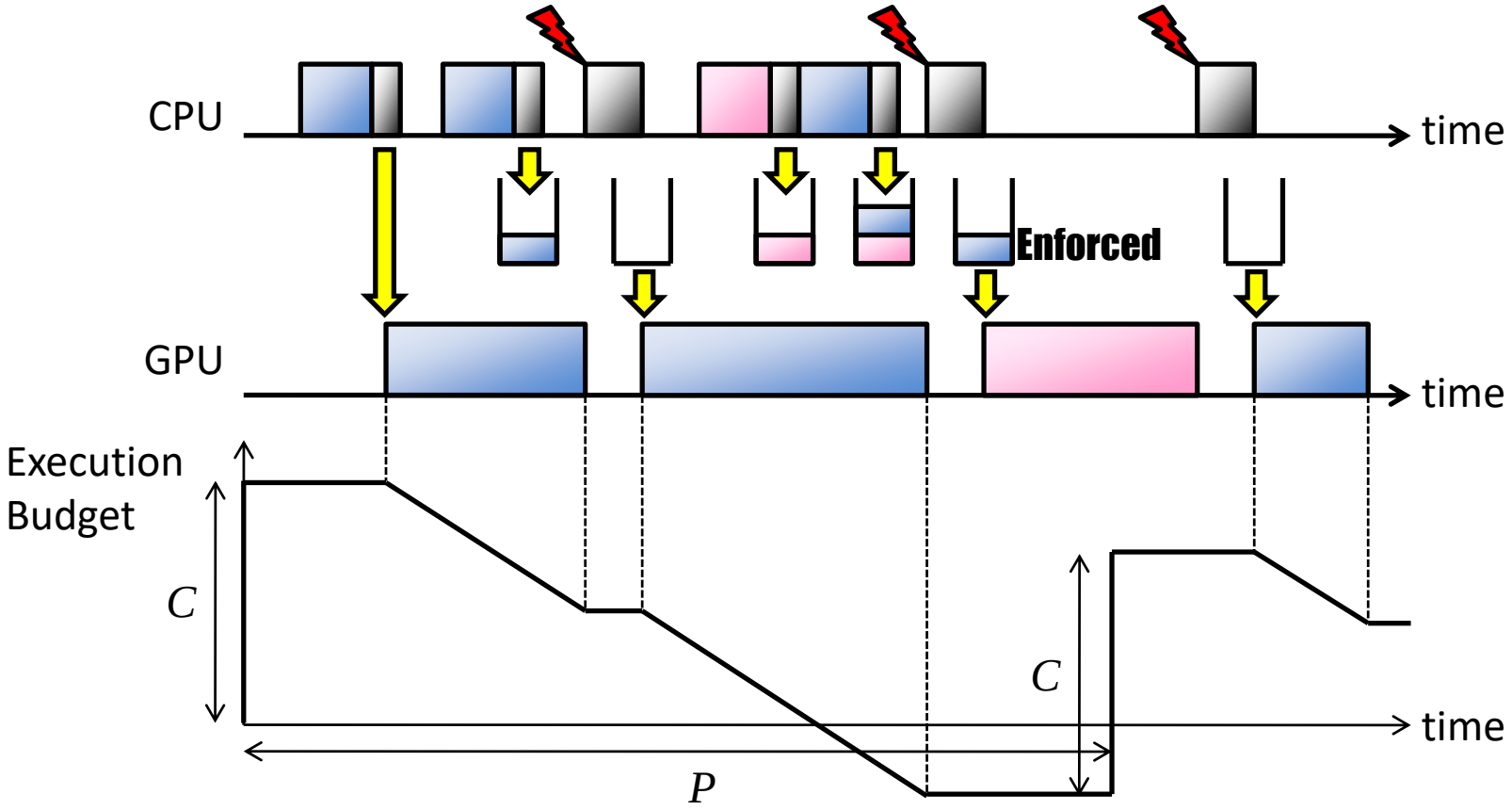
Priority Support – High Throughput (HT) Policy

- When GPU is not idle, GPU commands are queued, **only if priority is lower than current GPU context**
- When GPU gets idle, GPU commands are dispatched



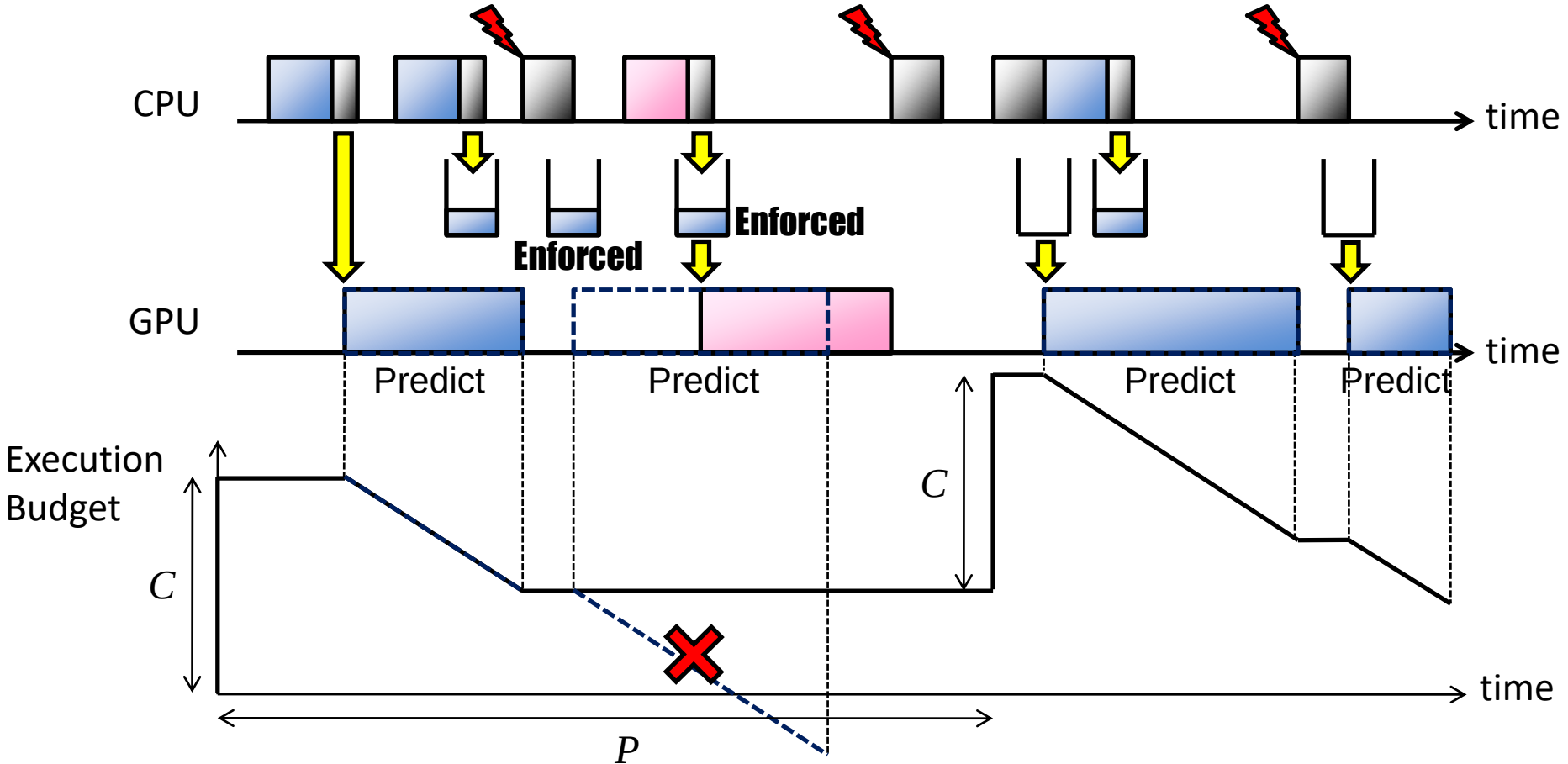
Reservation Support – Posterior Enforcement (PE) Policy

- Enforce GPU resource usage ***optimistically***
- Specify *capacity* (C) and *period* (P) per task (/proc/GPU/$*TASK*)



Reservation Support – Apriori Enforcement (AE) Policy

- Enforce GPU resource usage ***pessimistically***
- Specify *capacity* (C) and *period* (P) per task (/proc/GPU/$*TASK*)



GPU Execution Time Prediction

- History-based approach
  - Search **records of previous sequences of GPU commands** that **match** the **incoming sequences of GPU commands**
  - Works for **2-D** but needs investigation for **3-D** and **Compute**
- Please see the paper for the detail

Outline

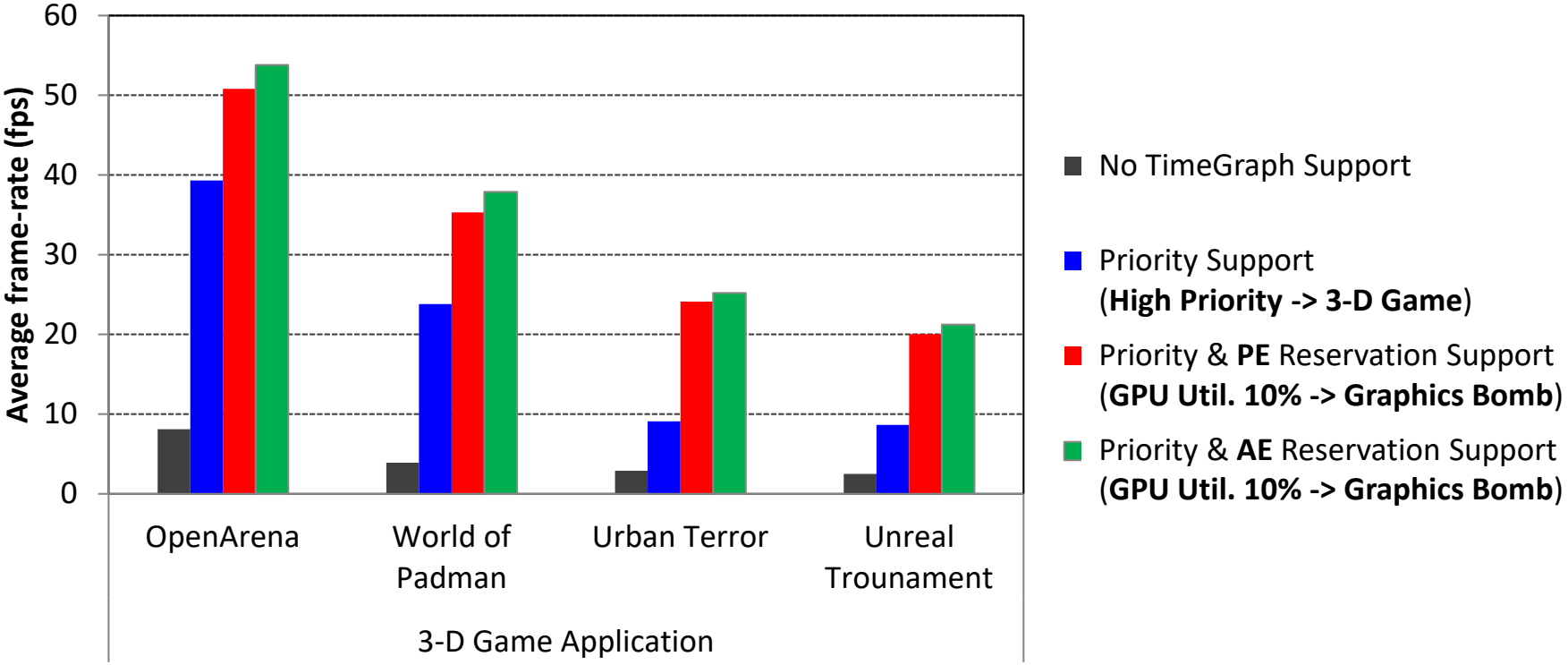
1. Introduction
2. What's Problem
3. Our Solution – "TimeGraph"
4. Evaluation
5. Summary

Experimental Setup

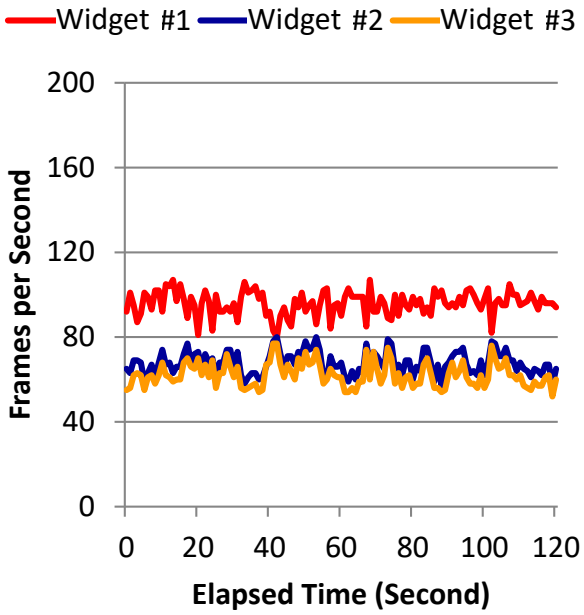
- **GPU**: NVIDIA GeForce 9800 GT
- **CPU**: Intel Xeon E5504
- **OS**: Linux Kernel 2.6.36
  - **Nouveau** open-source driver
- **Benchmark:**
  - Phoronix Test Suite http://www.phoronix-test-suite.com/
    - Including OpenGL 3-D game programs
  - Gallium3D Demo Suite http://www.mesa3d.org/
    - Including OpenGL 3-D widget and graphics-bomb programs

Performance Protection

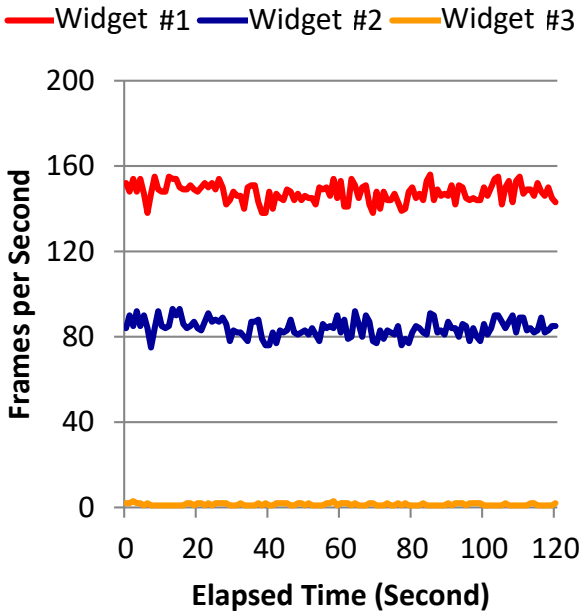
Frame Rate of 3-D Game competing with Graphics Bomb in background



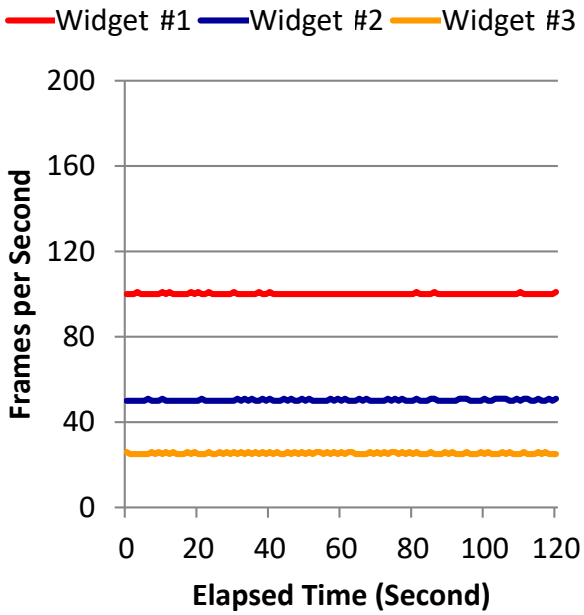
Interference on Time



No TimeGraph Support

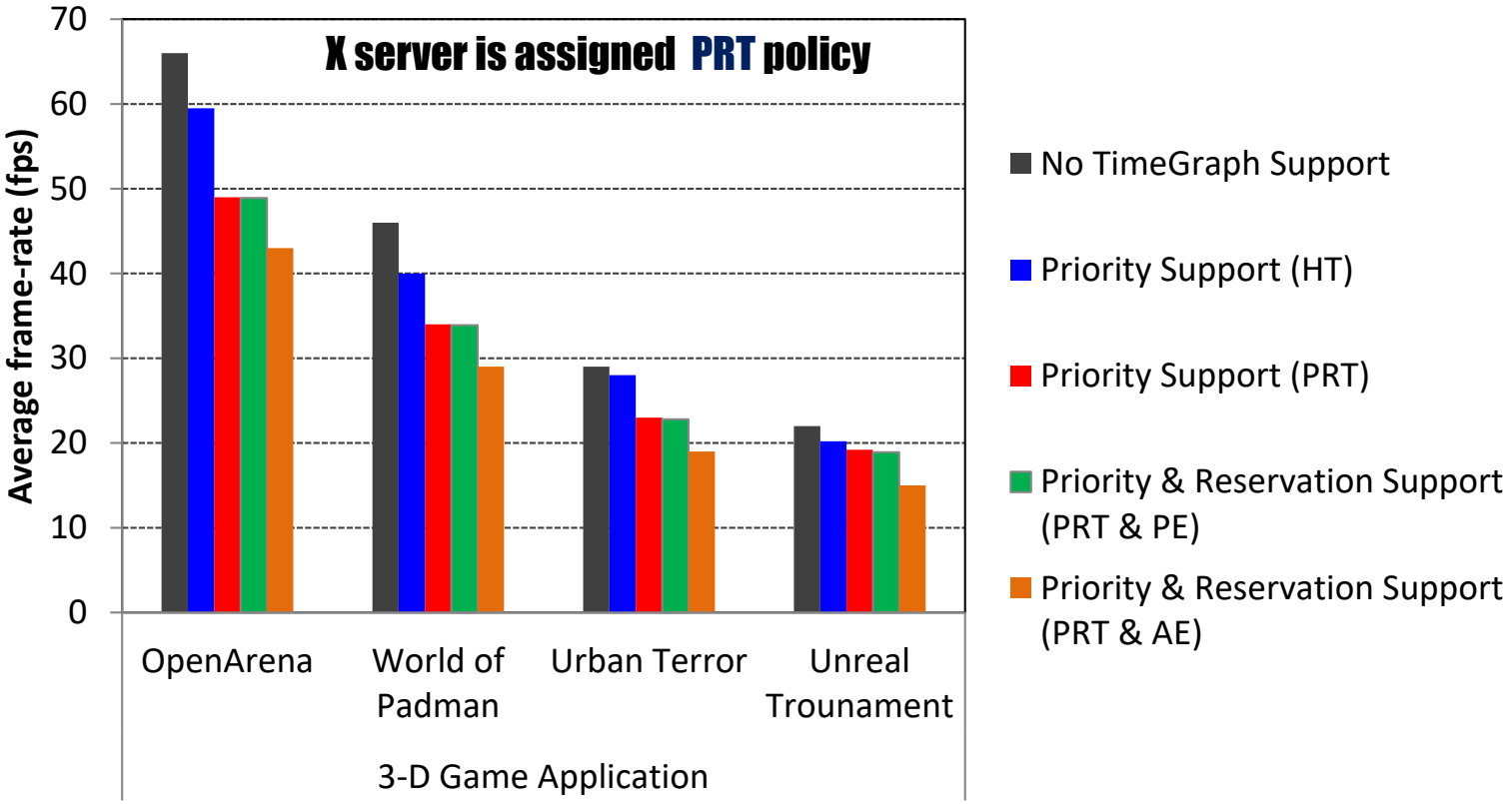


Priority Support (PRT)



Priority Support (PRT) + Reservation Support (PE)

Standalone Performance



Overhead is acceptable for protecting GPU

Outline

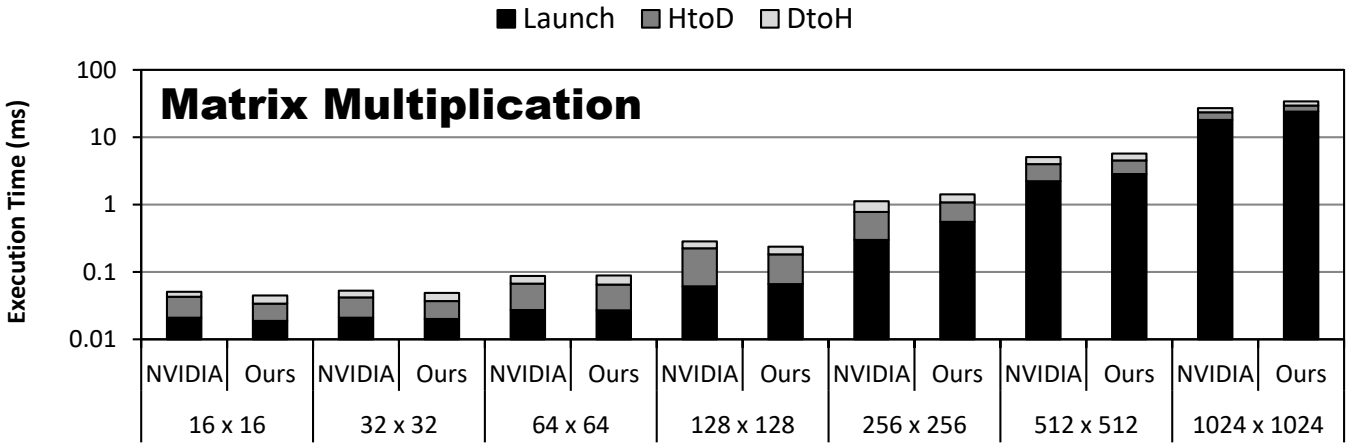
1. Introduction
2. What's Problem
3. Our Solution – "TimeGraph"
4. Evaluation
5. Summary

Concluding Remarks

- **TimeGraph** enables **prioritization** and **isolation** for **GPU** applications in multi-tasking environments
  - Device-driver solution: no modification to user-space
  - Scheduling of GPU commands
  - Reservation of GPU resource usage
- http://rtml.ece.cmu.edu/projects/timegraph/

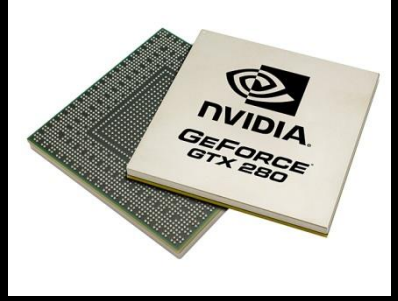
Current Status

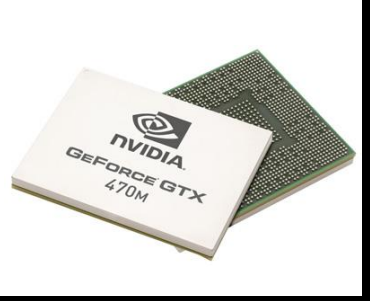
- **GPGPU** support (collaboration with *PathScale* Inc.)
  - Visit http://github.com/pathscale/pscnv
- Making open-source **fast** and **reliable**
  - It's getting competitive to the proprietary driver!
  - Some result from our OSPERT'11 paper (*) below:



NVIDIA GPU
GeForce GTX 480

* Available at http://www.contrib.andrew.cmu.edu/~shinpei/papers/ospert11.pdf





Thank you for your attention!

Questions?