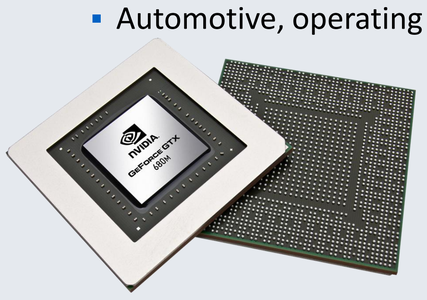
A Run-time Adaptive Framework for the Soft Real-time GPGPU Platform

Haeseung Lee, Mohammad Abdullah Al Faruque
Department of Electrical Engineering and Computer Science
University of California, Irvine
{haeseunl, alfaruqu}@uci.edu

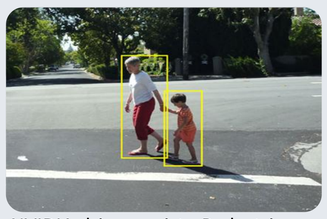
Introduction

- Both academia and industry use **multi/many core architectures** in **real-time systems**
  - GPU is one of the most well-known architecture
  - GPU is deployed in the critical path of the real-time applications
    - Automotive, operating theatre, and so on



The GeForce GTX 680M Mobile GPU (from Nvidia.com)

The Audi navigation system with live Google Earth is powered by NVIDIA (from Nvidia.com)

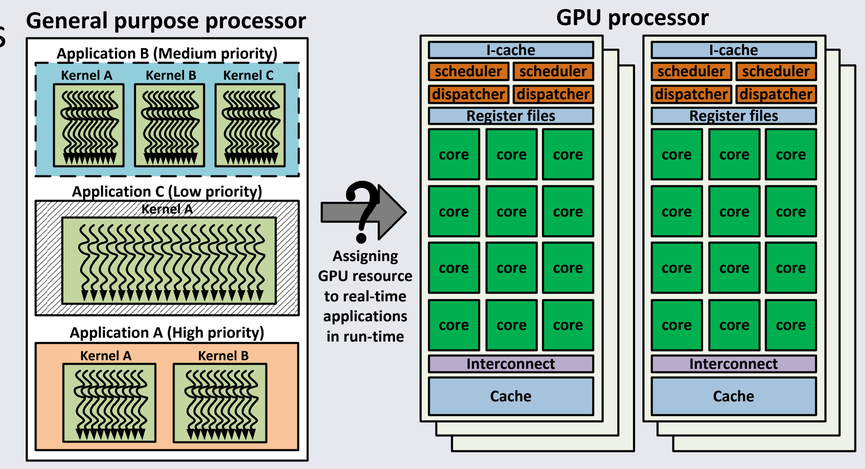


NVIDIA driver assist - Pedestrian detection (from Nvidia.com)

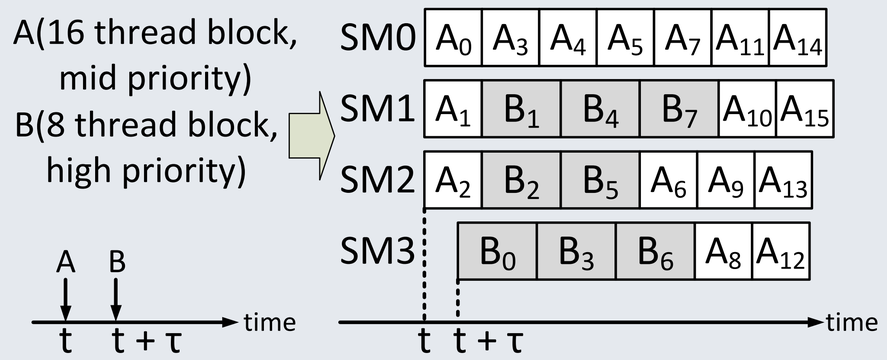
Leona M. and Harry B. Helmsley Surgical Suite, NewYork-Presbyterian Hospital, 2010

Motivation

- A GPU has several limitations to support real-time system
  - Implicitly processes its **workload in sequential way**
    - Parallelly processes workload only when there is enough resource
  - Implicitly allocates **as many GPU resource as possible**
    - Performance is main objective
  - No preemption support



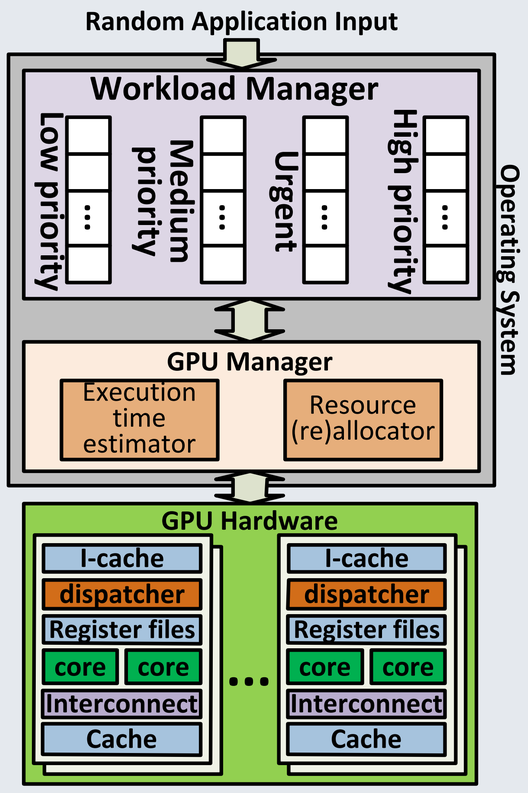
Desired behavior of GPGPU platform for real-time system



- Dynamically (re)allocates GPU resources
  - Assigns more GPU resources to the higher priority application
- Estimates the amount of GPU resources to meet deadline
  - Maximum performance is not mandatory

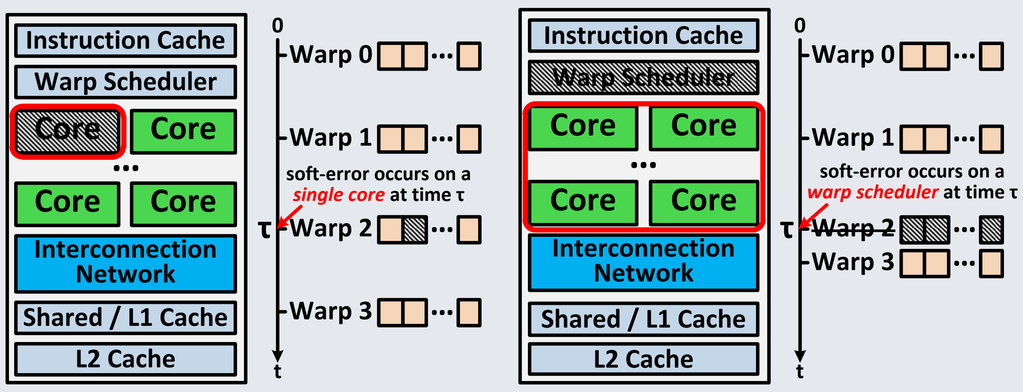
Preemption mechanism

- **Two levels of preemption** is proposed
  - **Temporal preemption**: Decides **when** the application is submitted into the GPU
  - **Spatial preemption**: Decides the **amount** of the GPU resources for the application
- Two modules implement each preemption
  - Workload manager: Temporal preemption
  - GPU manager: Spatial preemption
- Timing model is proposed to estimate the amount of GPU resources to meet deadline



Reliability (Soft-error)

- The **reliability** of the processor architecture is becoming an important issue as the technology scales
- **Soft-Error** has been considered as one of the most important reliability problem
  - Mainly caused by Cosmic Ray induced neutrons & Alpha particles
- Motivation
  - Since the GPU has multiple level of architectural hierarchy, effect of soft-error is different based on spatial distribution

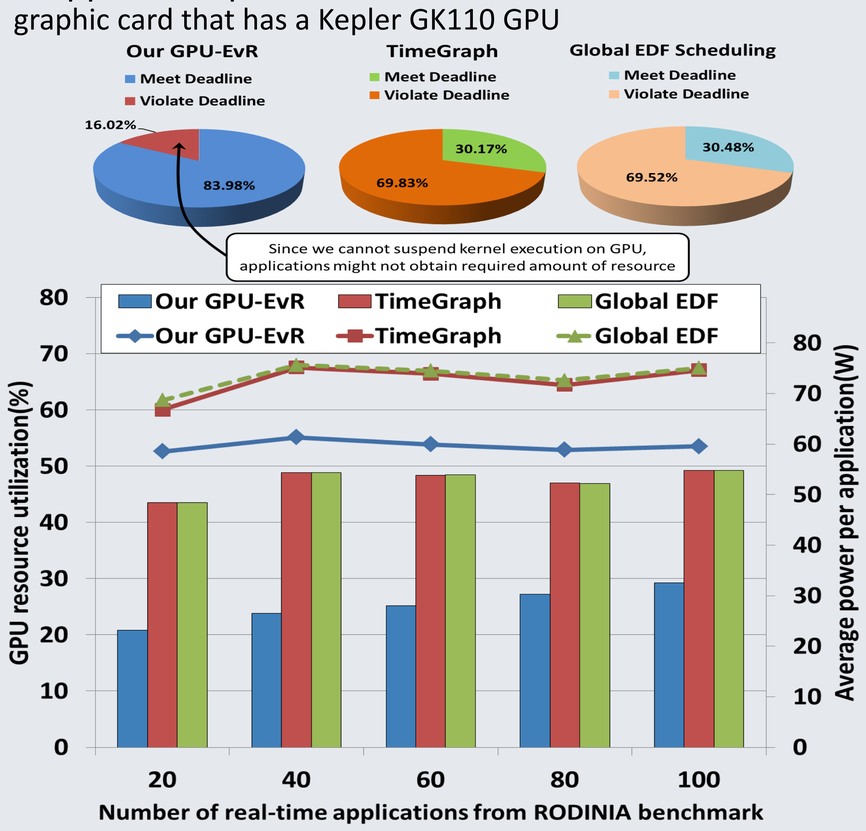


(a) A soft-error in CUDA cores

(b) A soft-error in warp scheduler

Results

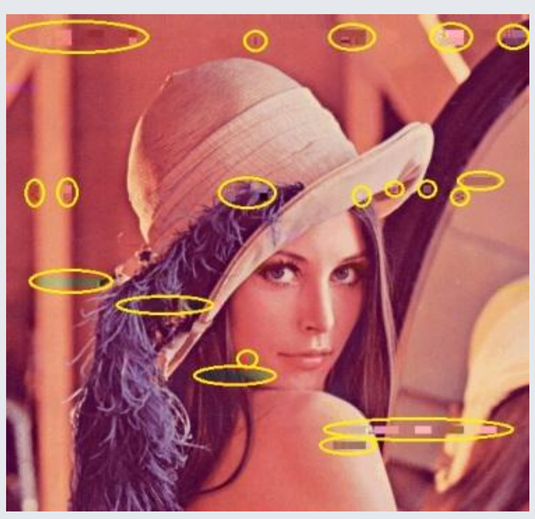
- Applications are randomly selected and injected into simulator in random time
  - Application profile is obtained from Nvidia's Tesla K20m graphic card that has a Kepler GK110 GPU



Preliminary results

- Based on the **spatial distribution** of the soft-error, the soft-error may cause **varying amount of effect**
  - Soft-error on the shared component may cause more error
- By using GPGPU-sim simulator, varying soft-error effect based on spatial distribution is simulated
- While executing jpeg encoding application, 60 soft-errors are injected in two components
  - In warp scheduler
  - In a single CUDA cores

(a) 60 soft-errors in CUDA cores



(b) 60 soft-errors in warp scheduler

Presented @ 2014 51st ACM/EDAC/IEEE Design Automation Conference