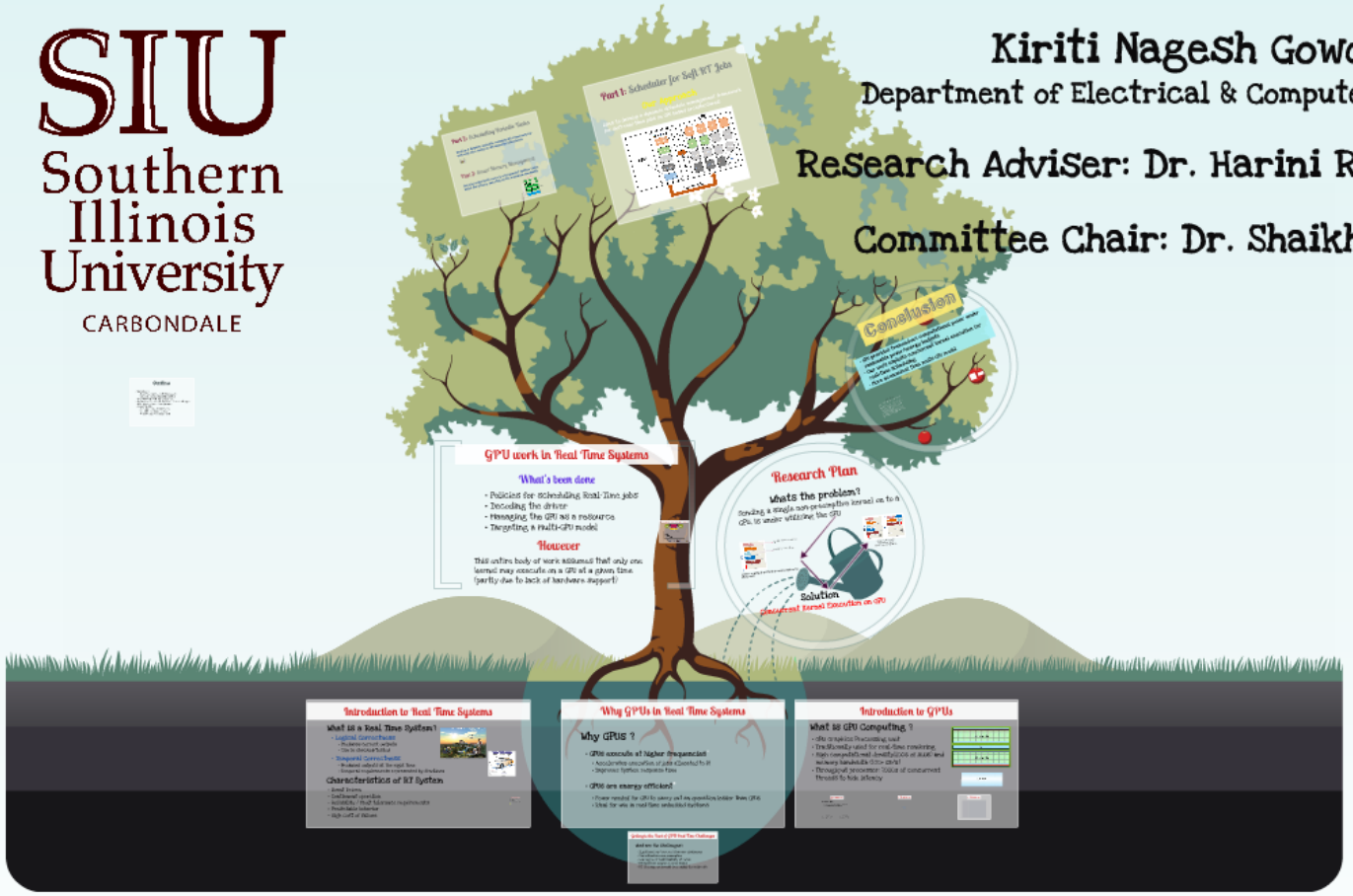
Real Time Execution On GPUs

SIU
Southern Illinois University
CARBONDALE

Kiriti Nagesh Gowda
Department of Electrical & Computer Engineering

Research Adviser: Dr. Harini Ramaprasad

Committee Chair: Dr. Shaikh Ahmed



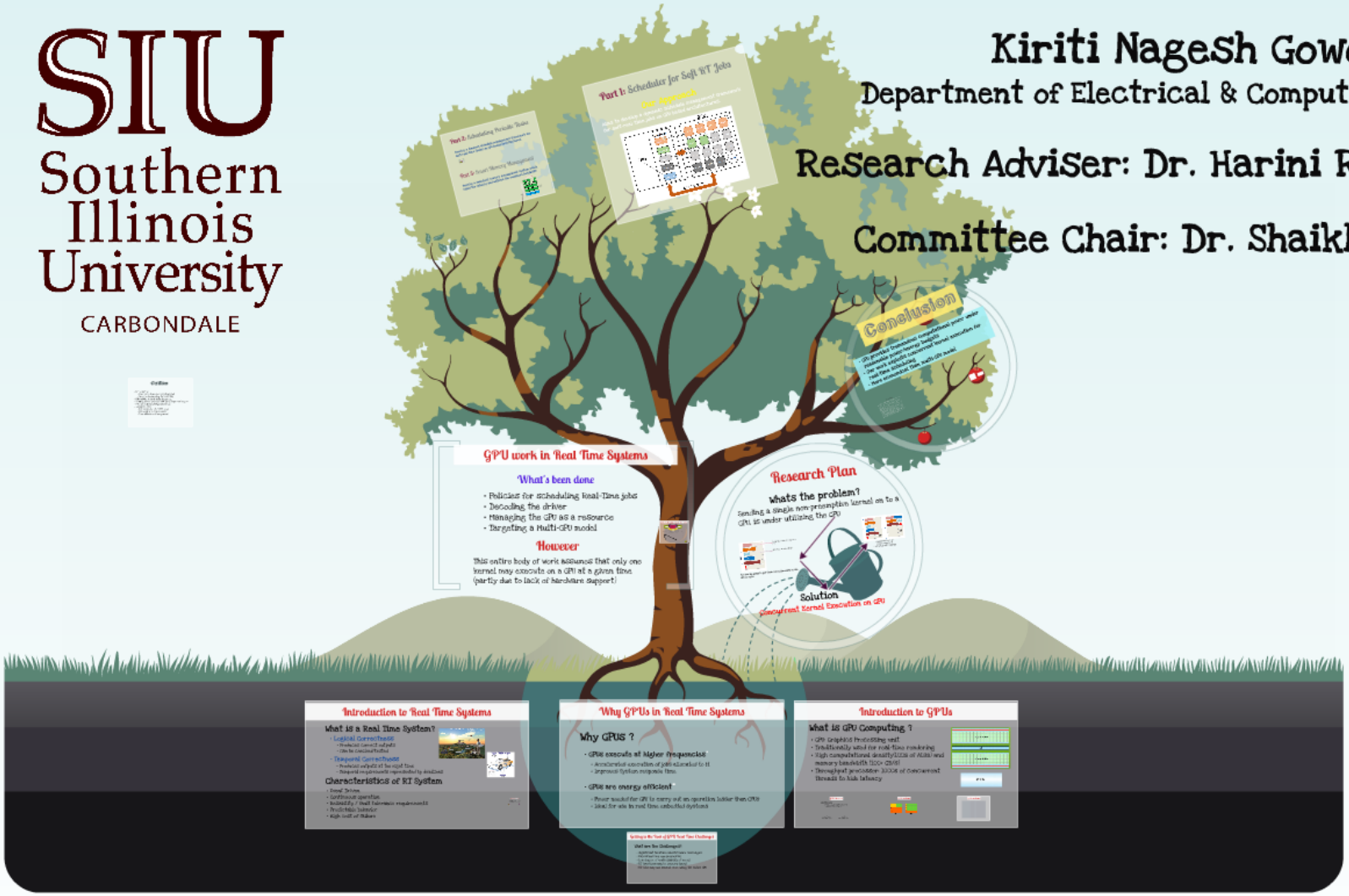
Real Time Execution On GPUs

SIU
Southern Illinois University
CARBONDALE

Kiriti NageSh Gowda
Department of Electrical & Computer Engineering

ReSearch AdviSer: Dr. Harini RamaprasSad

Committee Chair: Dr. Shaikh Ahmed

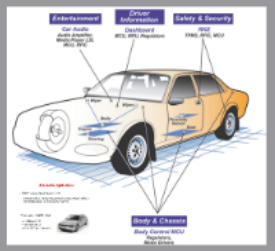


Outline

- Introduction
  - Real Time Systems (RT Systems)
  - Graphic Processing Units (GPUs)
- Why GPUs in Real Time Systems
- Getting to the root of GPU Real Time Challenges
- GPU work in Real Time Systems
- Research Plan
  - Scheduler for Soft RT Jobs
  - Scheduling Periodic Tasks
  - Smart Memory Management

Introduction to Real Time Systems

What is a Real Time System?

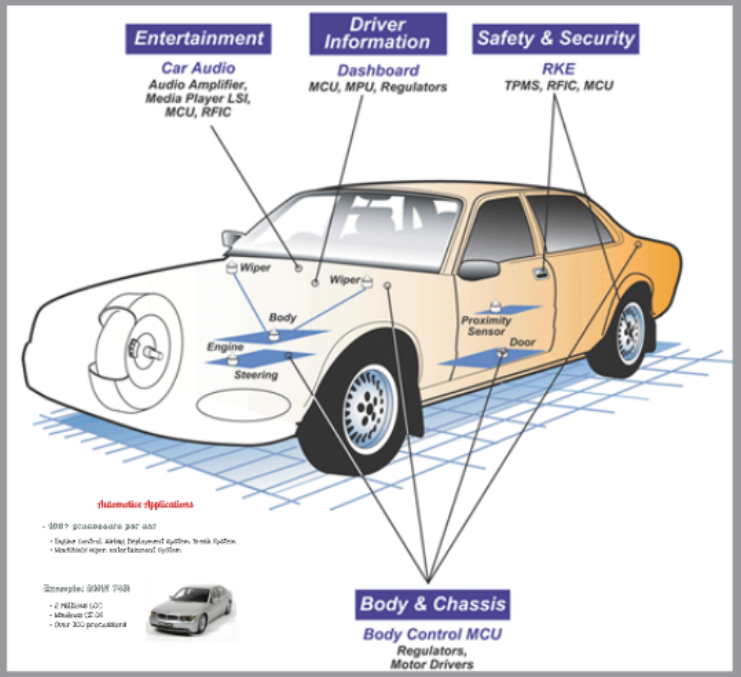


- Logical Correctness
  - Produces correct outputs
  - Can be checked/tested
- Temporal Correctness
  - Produces outputs at the right time
  - Temporal requirements represented by deadlines

Characteristics of RT System

- Event Driven
- Continuous operation
- Reliability / fault tolerance requirements
- Predictable behavior
- High cost of failure


Smart planes for safe air travel
Earthquake-proof building
Energy-efficient power grid
Structurally sound bridges
Assistive medical devices help patients enjoy independence
Smart appliances in energy-efficient homes
Safe highways
Smart car
Intelligent sensors for cleaner water resources
Robots at work and play
Smart navigational device so parents and children stay connected


Entertainment
Car Audio
Audio Amplifier, Media Player LSI, MCU, RFIC
Driver Information
Dashboard
MCU, MPU, Regulators
Safety & Security
RKE
TPMS, RFIC, MCU
Wiper
Wiper
Body
Engine
Steering
Proximity Sensor
Door
Automotive Applications
Body & Chassis
Body Control MCU
Regulators, Motor Drivers

Automotive Applications

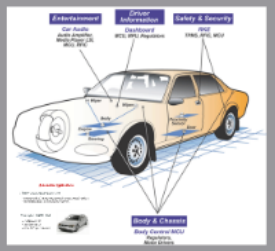
- 100+ processors per car
  - Engine control, Airbag Deployment System, Break System
  - WindShield wiper, entertainment System

Example: BMW 745i

- 2 Millions LOC
- Windows CE OS
- Over 100 processors

Introduction to Real Time Systems

What is a Real Time System?



- Logical Correctness
  - Produces correct outputs
  - Can be checked/tested
- Temporal Correctness
  - Produces outputs at the right time
  - Temporal requirements represented by deadlines

Characteristics of RT System

- Event Driven
- Continuous operation
- Reliability / fault tolerance requirements
- Predictable behavior
- High cost of failure

Misconceptions about Real-Time Systems

- Advancement in Supercomputing will take care of RT Requirements "Buy a faster processor"
- RT is performance engineering
- RT systems function in a static environment

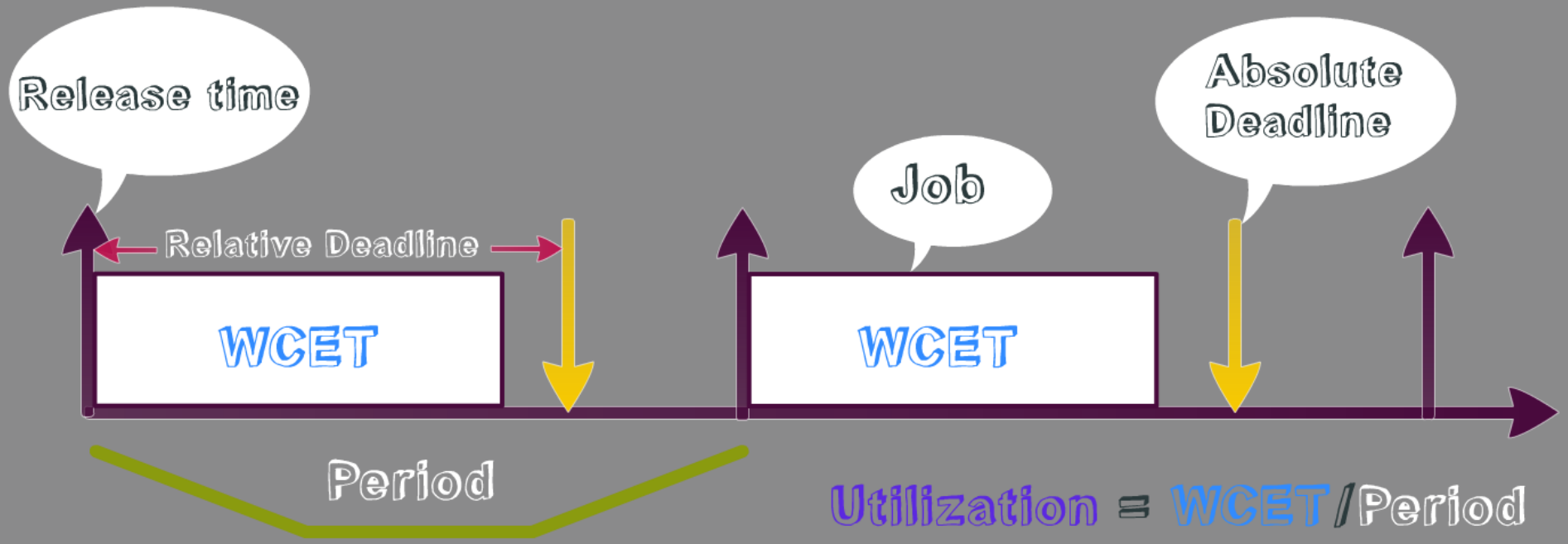
Categorization of Real Time Systems

Hard Real Time Systems

Soft Real Time Systems

Task Characteristics

- A Program is called a Task
- Job is an instance of a Task
- Tasks may be Periodic or Non-Periodic

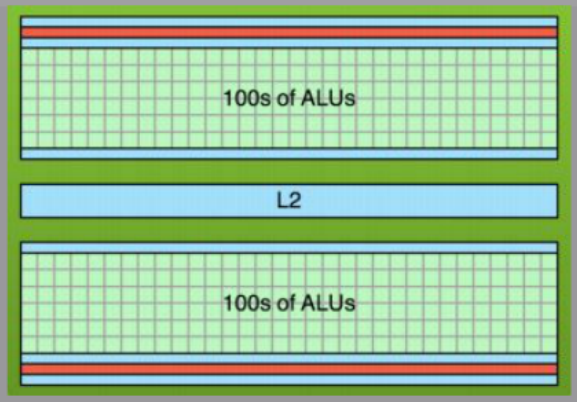


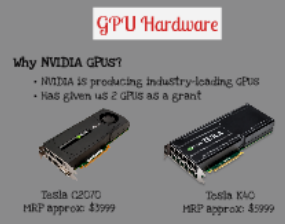
WCET - Worst Case Execution Time

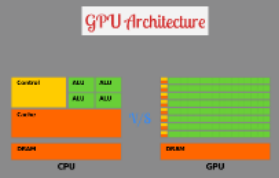
Introduction to GPUs

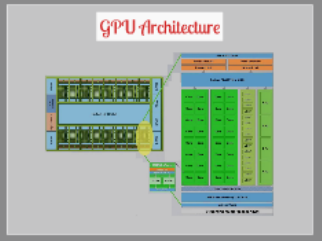
What is GPU Computing ?

- GPU: Graphics Processing unit
- Traditionally used for real-time rendering
- High computational density(100s of ALUs) and memory bandwidth (100+ GB/s)
- Throughput processor: 1000s of concurrent threads to hide latency









GPU Hardware

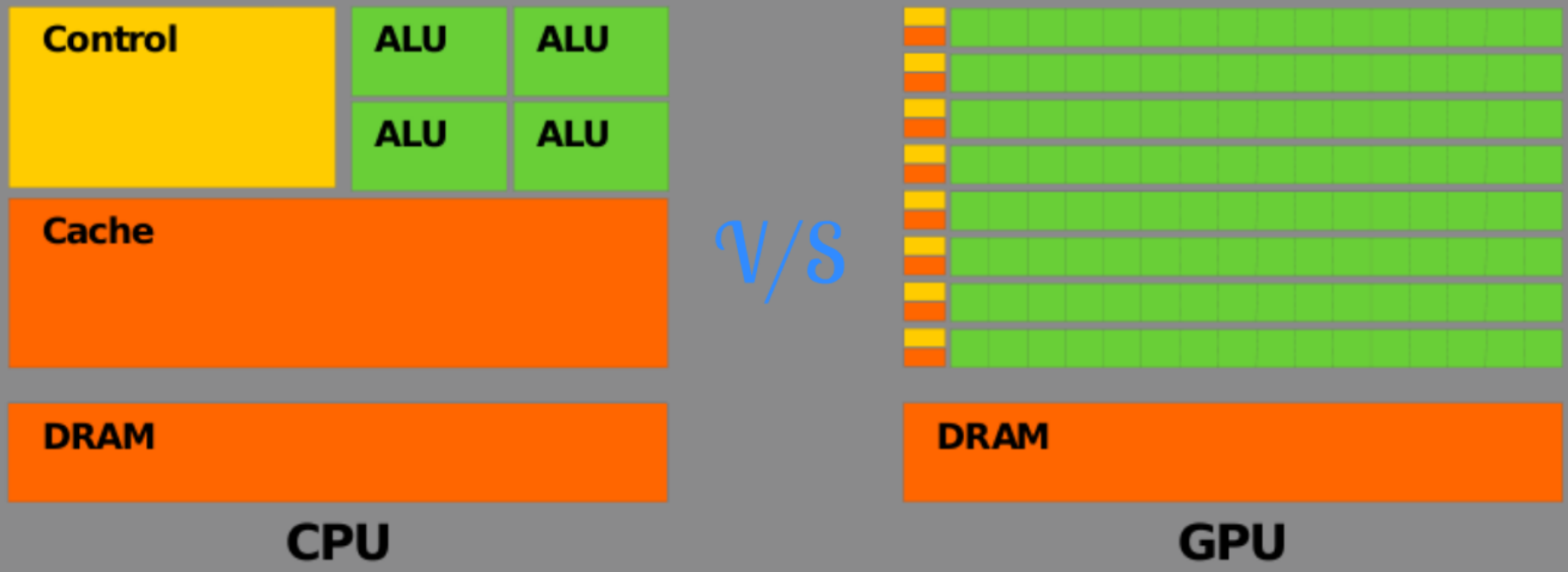
Why NVIDIA GPUs?

- NVIDIA is producing industry-leading GPUs
- Has given us 2 GPUs as a grant

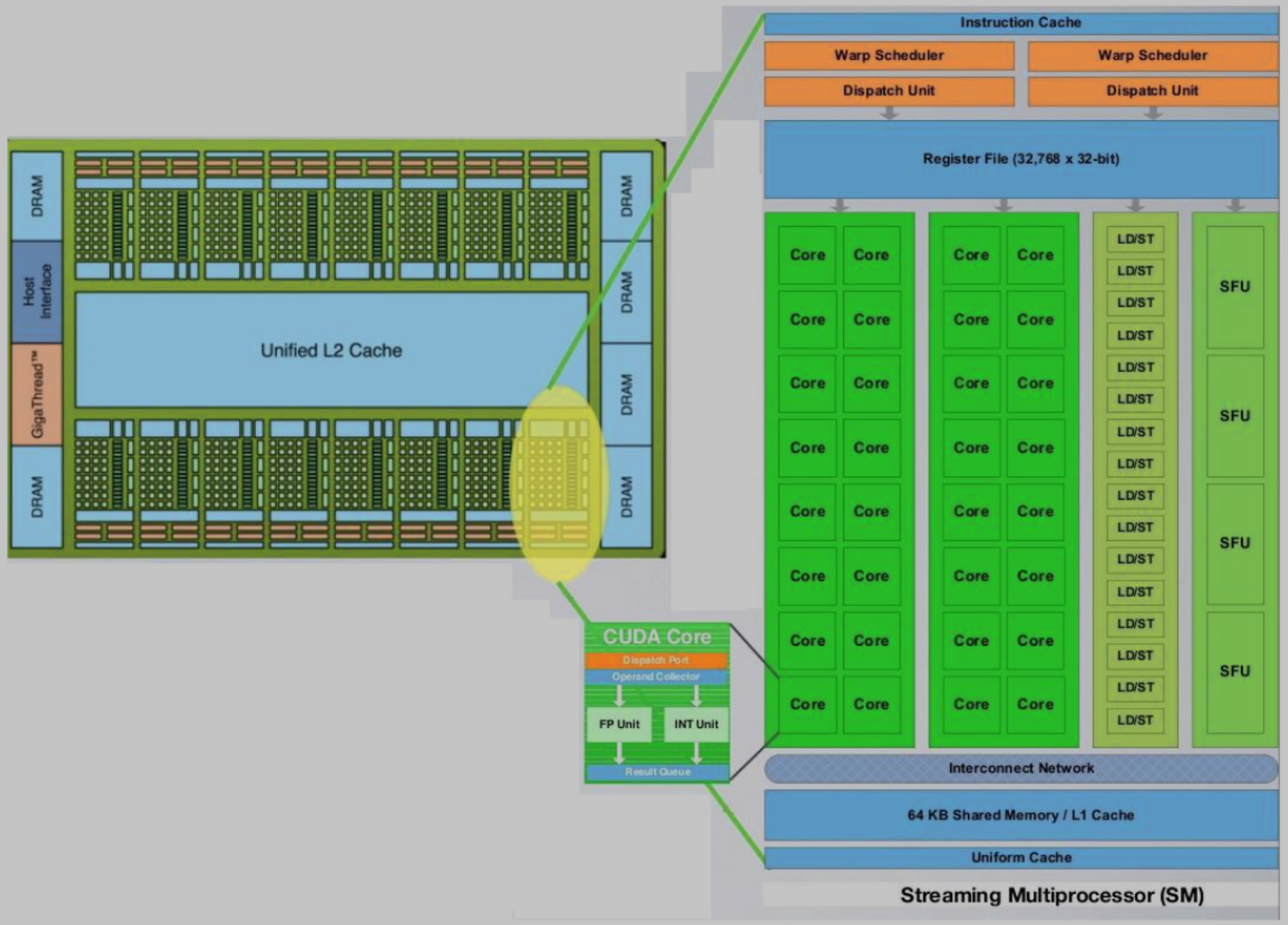
Tesla C2070
MRP approx: $3999

Tesla K40
MRP approx: $5999

GPU Architecture



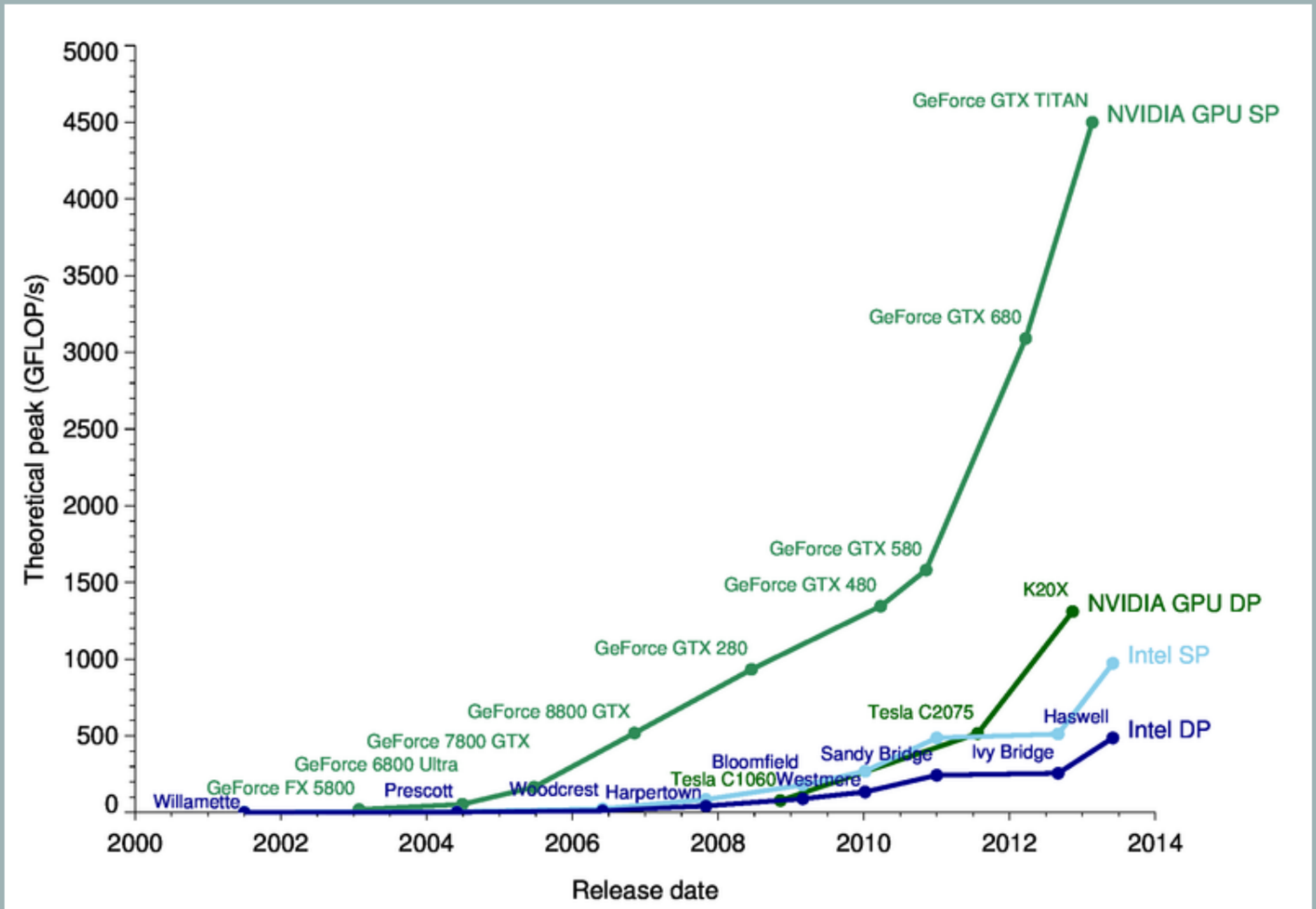
GPU Architecture



Why GPUs in Real Time Systems

Why GPUS ?

- GPUS execute at higher frequencies
  - Accelerates execution of jobs allocated to it
  - Improves System response time
- GPUS are energy efficient
  - Power needed for GPU to carry out an operation lesser than CPUS
  - Ideal for use in real time embedded systems


Theoretical peak (GFLOP/s)
Release date
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
2000
2002
2004
2006
2008
2010
2012
2014
GeForce GTX TITAN
NVIDIA GPU SP
GeForce GTX 680
GeForce GTX 580
GeForce GTX 480
GeForce GTX 280
GeForce 8800 GTX
GeForce 7800 GTX
GeForce 6800 Ultra
GeForce FX 5800
Willamette
Prescott
Woodcrest
Harpertown
Bloomfield
Tesla C1060
Westmere
Sandy Bridge
Ivy Bridge
Haswell
Tesla C2075
K20X
NVIDIA GPU DP
Intel SP
Intel DP

Why GPUs in Real Time Systems

Why GPUS ?

- GPUS execute at higher frequencies
  - Accelerates execution of jobs allocated to it
  - Improves System response time
- GPUS are energy efficient
  - Power needed for GPU to carry out an operation lesser than CPUS
  - Ideal for use in real time embedded systems

The Green500 List

Listed below are the November 2013 The Green500's energy-efficient supercomputers ranked from 1 to 10.

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	4,503.17	GSIC Center, Tokyo Institute of Technology	TSUBAME-KFC - LX 1U-4GPU/104Re-1G Cluster, Intel Xeon E5-2620v2 6C 2.100GHz, Infiniband FDR, NVIDIA K20x	27.78
2	3,631.86	Cambridge University	Wilkes - Dell T620 Cluster, Intel Xeon E5-2630v2 6C 2.600GHz, Infiniband FDR, NVIDIA K20	52.62
3	3,517.84	Center for Computational Sciences, University of Tsukuba	HA-PACS TCA - Cray 3623G4-SM Cluster, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband QDR, NVIDIA K20x	78.77
4	3,185.91	Swiss National Supercomputing Centre (CSCS)	Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Level 3 measurement data available	1,753.66
5	3,130.95	ROMEO HPC Center - Champagne-Ardenne	romeo - Bull R421-E3 Cluster, Intel Xeon E5-2650v2 8C 2.600GHz, Infiniband FDR, NVIDIA K20x	81.41
6	3,068.71	GSIC Center, Tokyo Institute of Technology	TSUBAME 2.5 - Cluster Platform SL390s G7, Xeon X5670 6C 2.930GHz, Infiniband QDR, NVIDIA K20x	922.54
7	2,702.16	University of Arizona	iDataPlex DX360M4, Intel Xeon E5-2650v2 8C 2.600GHz, Infiniband FDR14, NVIDIA K20x	53.62
8	2,629.10	Max-Planck-Gesellschaft MPI/IPP	iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x	269.94
9	2,629.10	Financial Institution	iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x	55.62
10	2,358.69	CSIRO	CSIRO GPU Cluster - Nitro G16 3GPU, Xeon E5-2650 8C 2.000GHz, Infiniband FDR, Nvidia K20m	71.01

* Performance data obtained from publicly available sources including TOP500

Getting to the Root of GPU Real Time Challenges

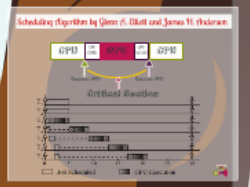
What are the Challenges?

- Significant hardware and firmware challenges
- Executions are non-preemptive
- Low degree of controllability of cores
- RT functions may be memory bound
- RT Jobs may not benefit from using the entire GPU

GPU work in Real Time Systems

What's been done

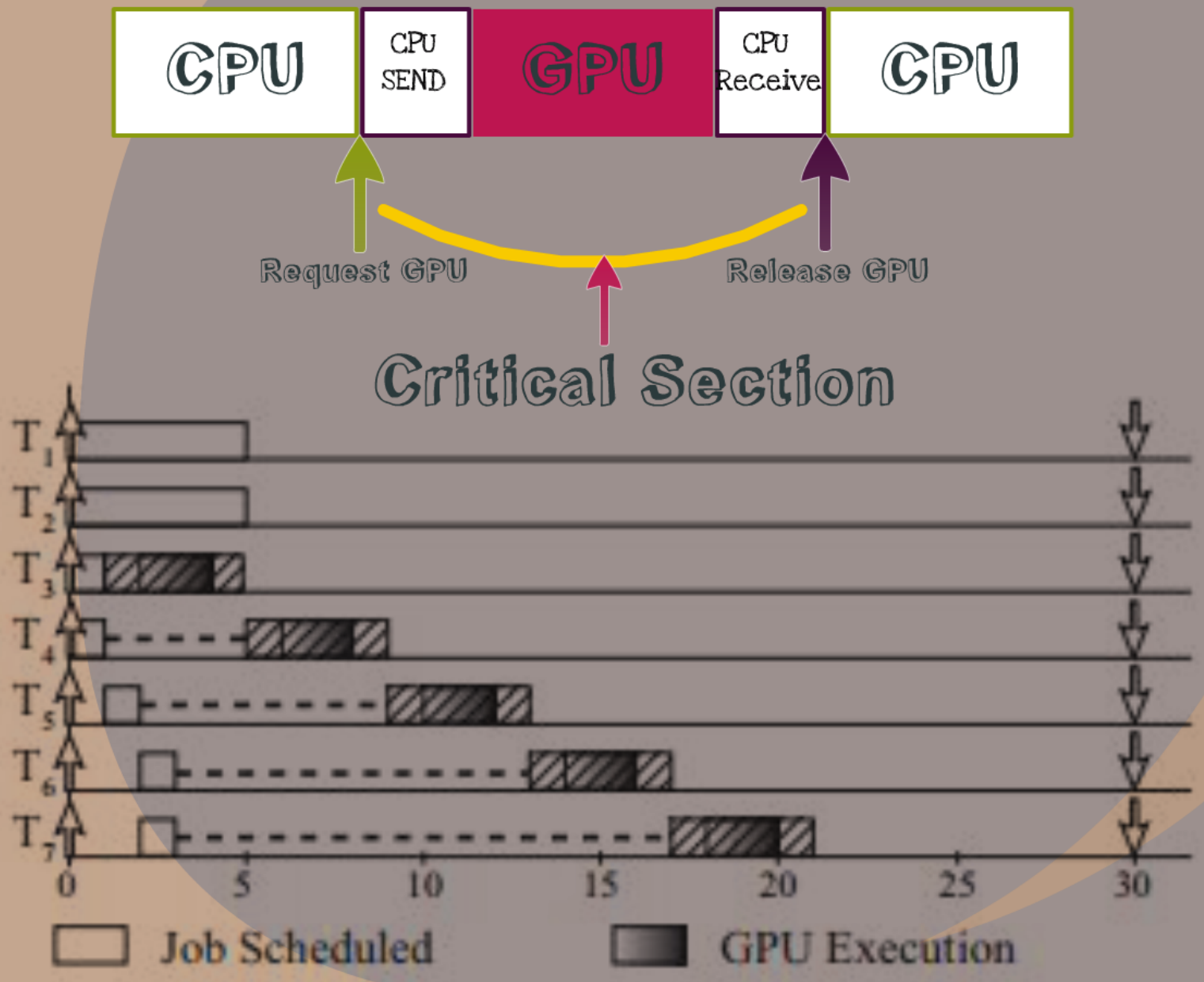
- Policies for scheduling Real-Time jobs
- Decoding the driver
- Managing the GPU as a resource
- Targeting a Multi-GPU model



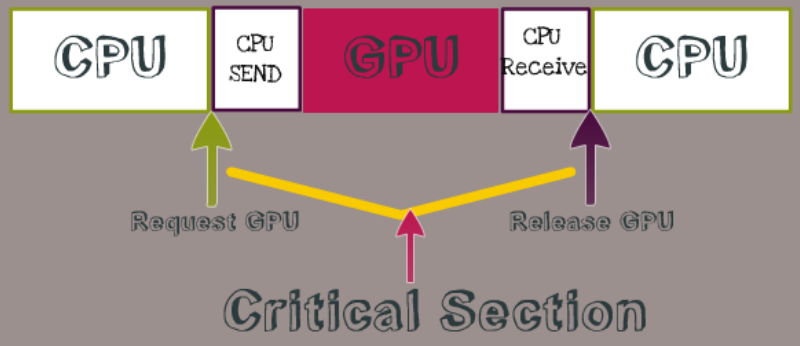
However

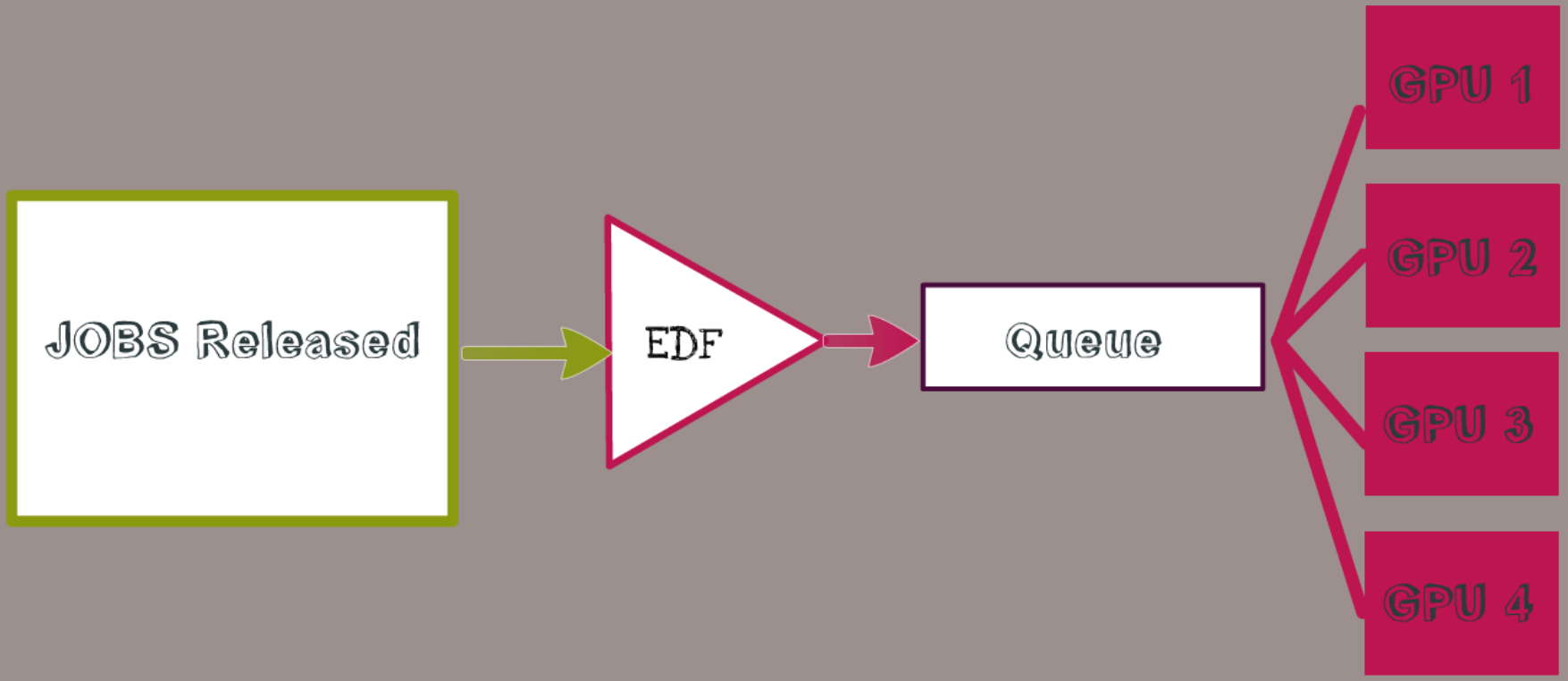
This entire body of work assumes that only one kernel may execute on a GPU at a given time (partly due to lack of hardware support)

Scheduling Algorithm by Glenn A. Elliott and James H. Anderson



Scheduling On Multi-GPU System

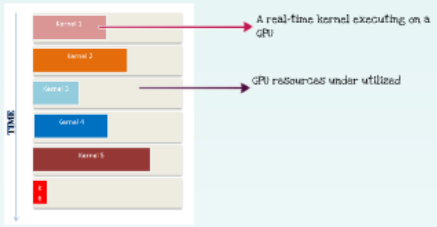




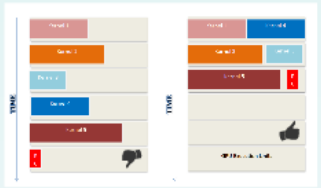
Research Plan

Whats the problem?

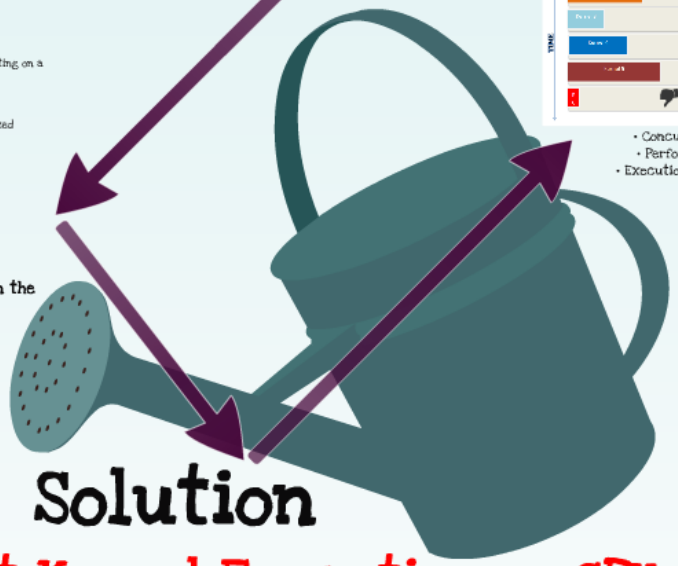
Sending a single non-preemptive kernel on to a GPU, is under utilizing the GPU



How are we going to put these execution units on the GPU to work?

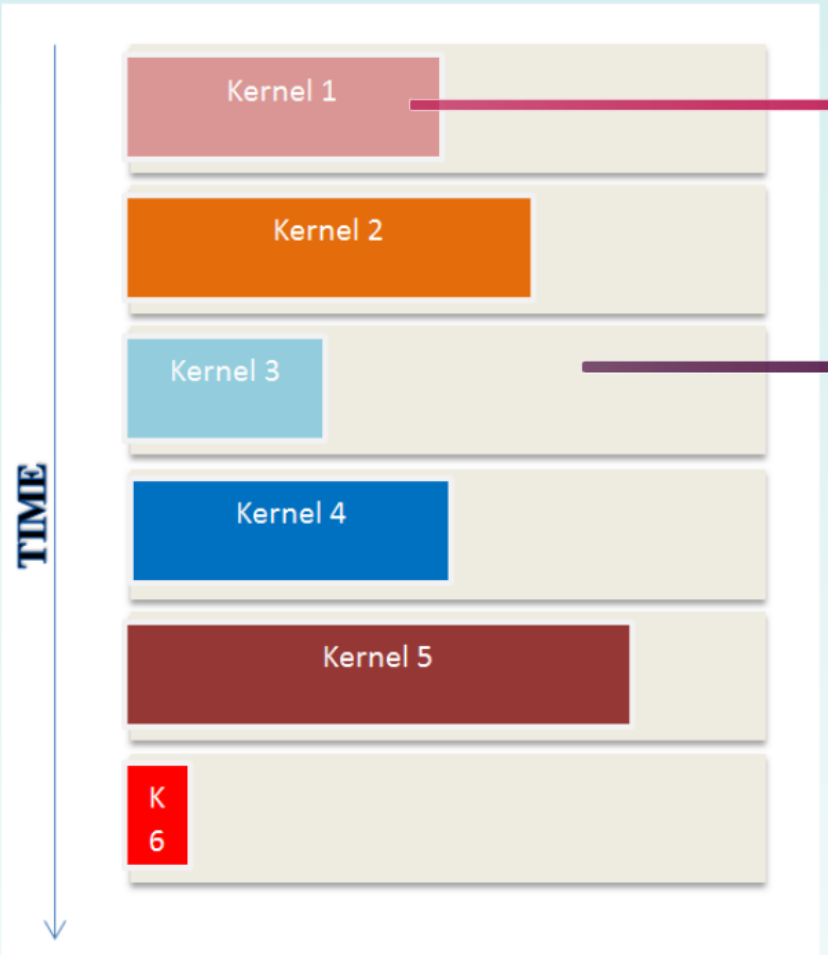


- Concurrent kernels
- Performance boost
- Execution units available



Solution

Concurrent Kernel Execution on GPU

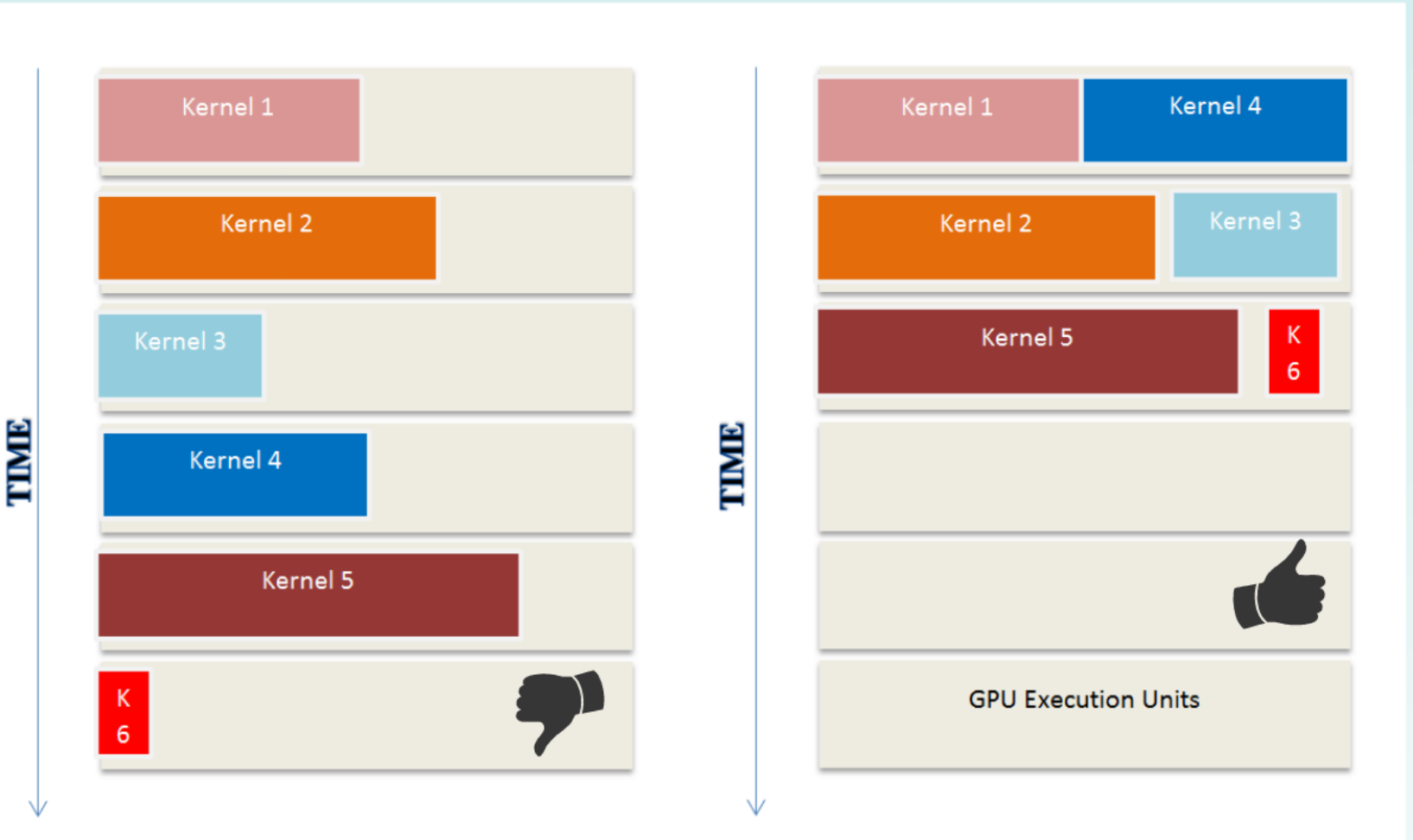


A real-time kernel executing on a GPU

GPU resources under utilized

How are we going to put these execution units on the GPU to work?

ing to put these execution units on the

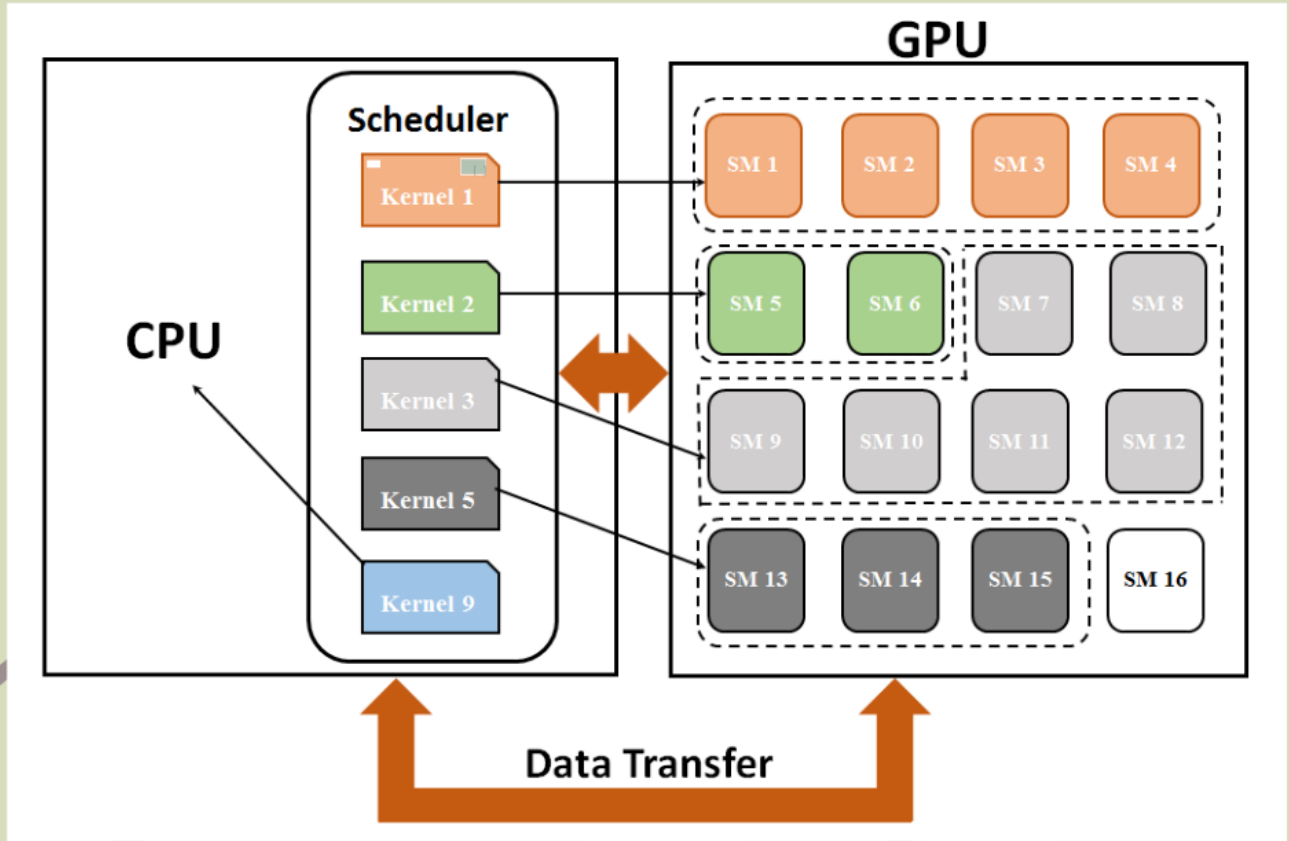


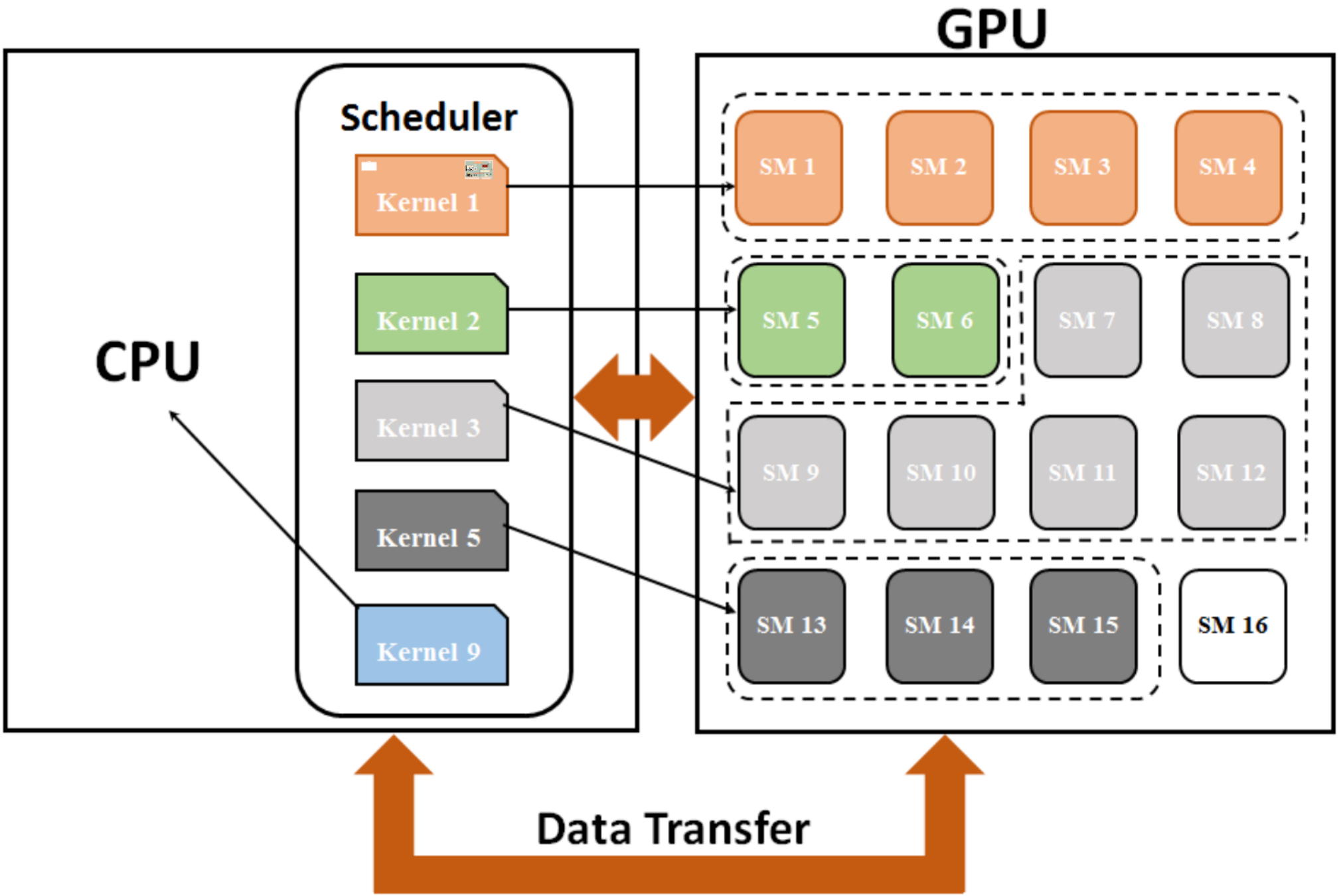
- Concurrent kernels
- Performance boost
- Execution units available

Part 1: Scheduler for Soft RT Jobs

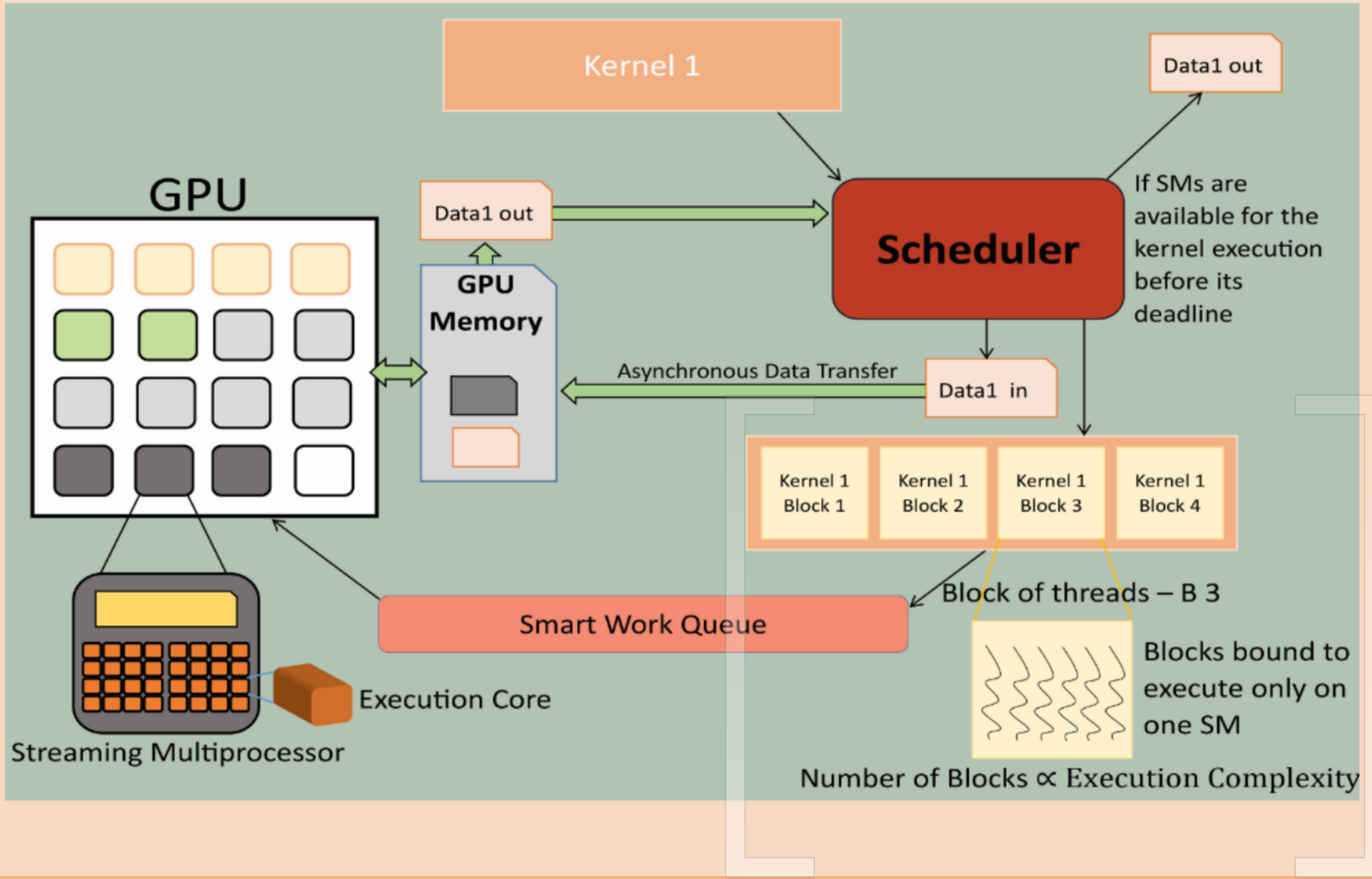
Our Approach

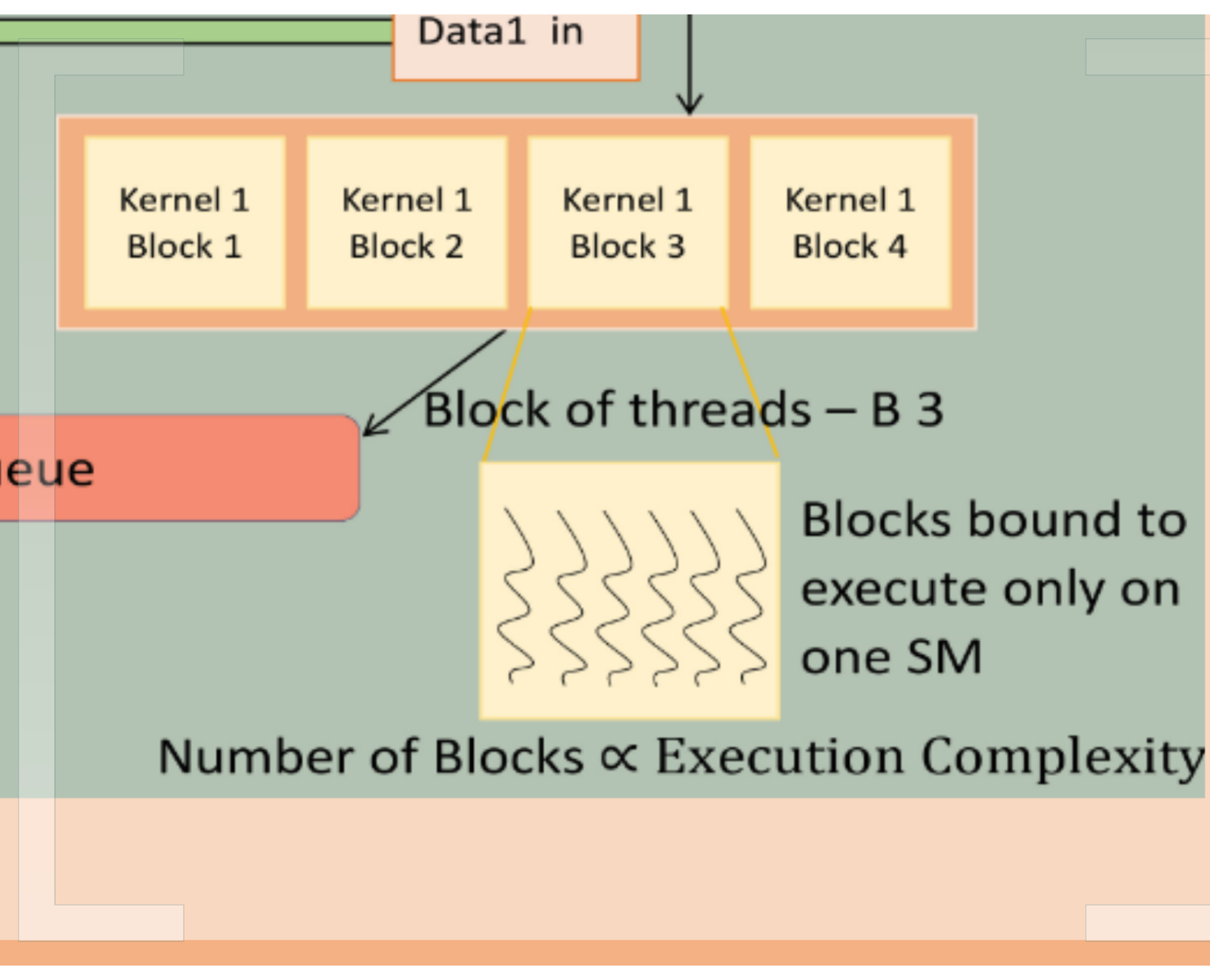
Aims to develop a dynamic schedule management framework for soft-real-time jobs on GPU based architectures.




GPU
Scheduler
Kernel 1
Kernel 2
Kernel 3
Kernel 5
Kernel 9
CPU
SM 1
SM 2
SM 3
SM 4
SM 5
SM 6
SM 7
SM 8
SM 9
SM 10
SM 11
SM 12
SM 13
SM 14
SM 15
SM 16
Data Transfer

Scheduler Video


Kernel 1
Data1 out
GPU
Data1 out
Scheduler
If SMs are available for the kernel execution before its deadline
GPU Memory
Asynchronous Data Transfer
Data1 in
Kernel 1 Block 1
Kernel 1 Block 2
Kernel 1 Block 3
Kernel 1 Block 4
Block of threads – B 3
Smart Work Queue
Blocks bound to execute only on one SM
Execution Core
Streaming Multiprocessor
Number of Blocks ∝ Execution Complexity

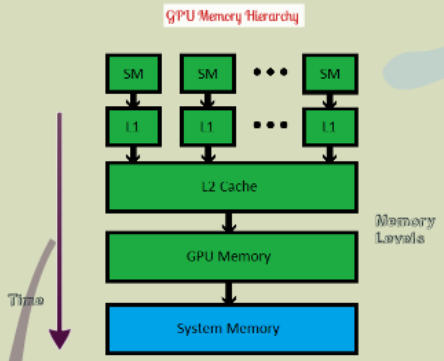

Data1 in
Kernel 1
Block 1
Kernel 1
Block 2
Kernel 1
Block 3
Kernel 1
Block 4
Block of threads – B 3
Blocks bound to execute only on one SM
Number of Blocks ∝ Execution Complexity

Part 2: Scheduling Periodic Tasks

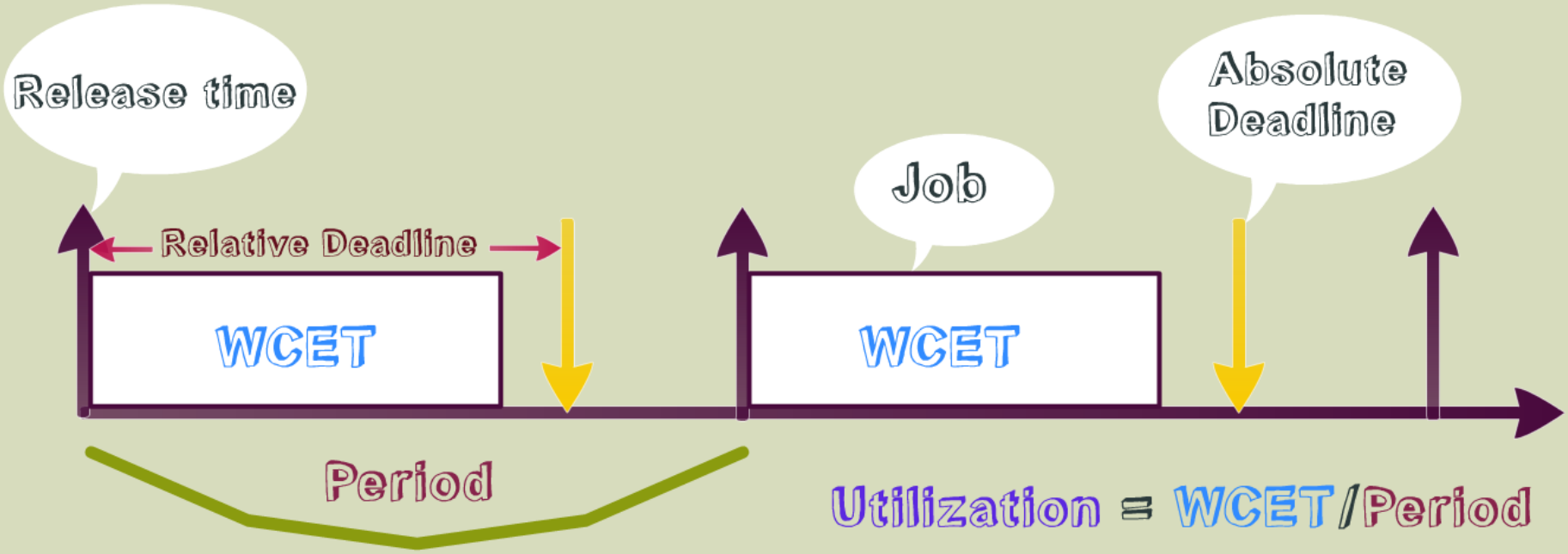
Develop a dynamic schedule management framework for soft-real-time Tasks on GPU based architectures.

Part 3: Smart Memory Management

Develop a seamless memory management system which hides the latency and utilizes the maximum bandwidth



Periodic Tasks



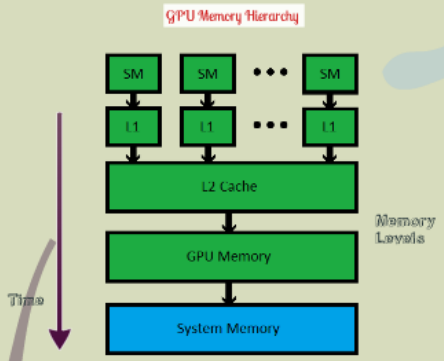
WCET - Worst Case Execution Time

Part 2: Scheduling Periodic Tasks

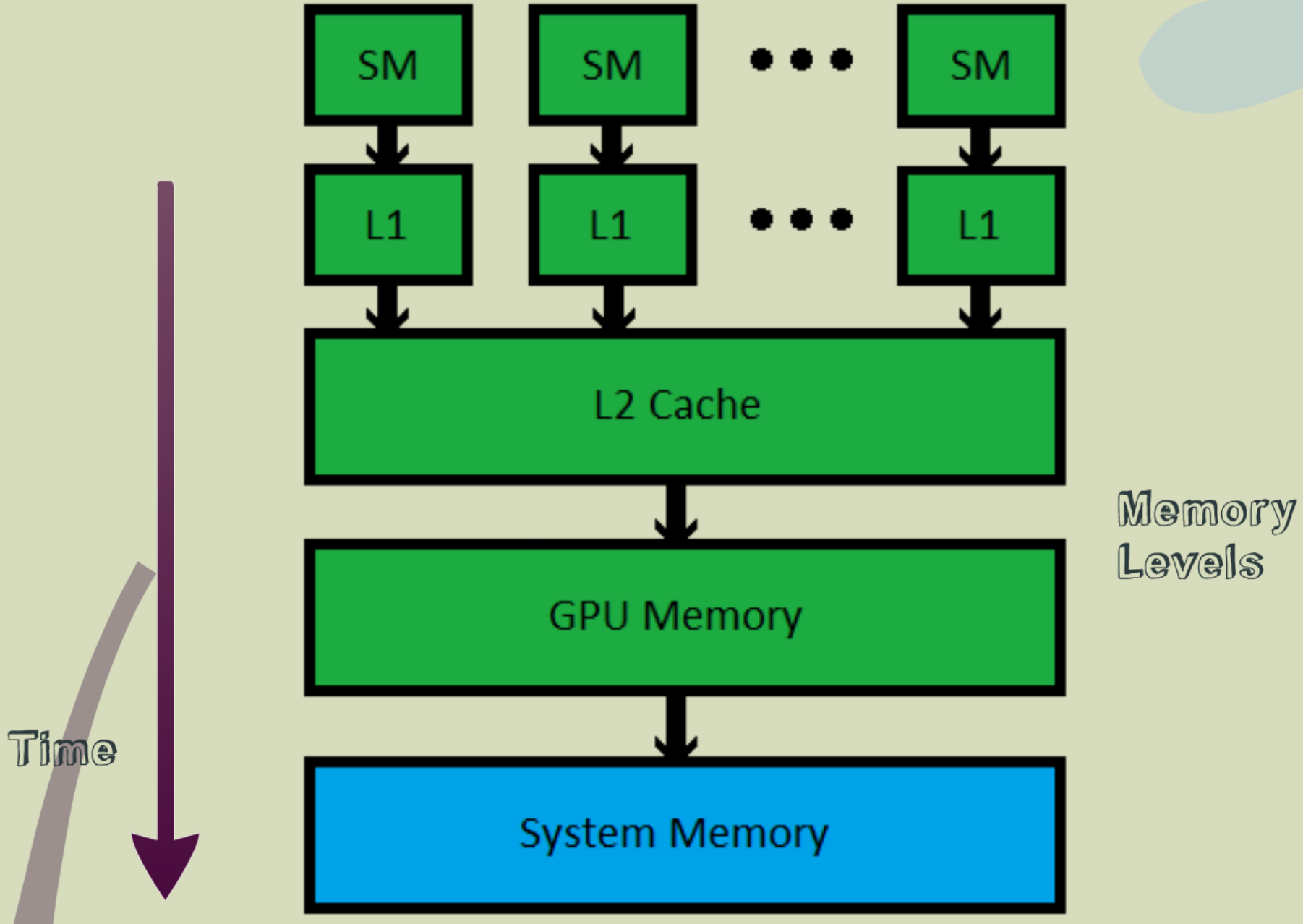
Develop a dynamic schedule management framework for soft-real-time Tasks on GPU based architectures.

Part 3: Smart Memory Management

Develop a seamless memory management system which hides the latency and utilizes the maximum bandwidth



GPU Memory Hierarchy



Conclusion

- GPU provides tremendous computational power under reasonable power/energy budgets
- Our work exploits concurrent kernel execution for real-time scheduling
- More economical than multi-GPU model

References

[1] G. A. Elliott and J. H. Anderson, "Real-World Constraints of GPUS in Real-Time Systems," in International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), 2011, pp. 48-54.

[2] P. N. Glaskowsky, "NVIDIA'S Fermi: the first complete GPU computing architecture," White paper, 2009.

[3] L. Wang, M. Huang, and T. El-Ghazawi, "Exploiting concurrent kernel execution on graphic processing units," in International Conference on High Performance Computing and Simulation (HPCS), 2011, pp. 24-32.

[4] G. A. Elliott and J. H. Anderson, "Globally Scheduled real-time multiprocessor systems with GPUS," Real-Time Systems, vol. 48, no. 1, pp. 34-74, 2012.

[5] S. Kato, K. Lakshmanan, R. Rajkumar, and Y. Ishikawa, "TimeGraph: GPU scheduling for real-time multi-tasking environments," in USENIX Annual Technical Conference (USENIX ATC), 2011, p. 17.

[6] G. A. Elliott, B. C. Ward, and J. H. Anderson, "GPUSync: A framework for real-time GPU management," in Real-Time Systems Symposium (RTSS), 2013, pp. 33-44.

[7] G. A. Elliott and J. H. Anderson, "An optimal k-exclusion real-time locking protocol motivated by multi-GPU systems," Real-Time Systems, vol. 49, no. 2, pp. 140-170, 2013.

[8] G. Elliott and J. Anderson, "Robust real-time multiprocessor interrupt handling motivated by GPUS," in Euromicro Conference on Real-Time Systems (ECRTS), 2012, pp. 267-276.

[9] S. Kato, K. Lakshmanan, A. Kumar, M. Kelkar, Y. Ishikawa, and R. Rajkumar, "RGEM: A responsive GPGPU execution model for runtime engines," in Real-Time Systems Symposium (RTSS), 2011, pp. 57-66.

[10] J. Kim, R. R. Rajkumar, and S. Kato, "Towards adaptive GPU resource management for embedded real-time systems," ACM SIGBED Review, vol. 10, no. 1, pp. 14-17, 2013.

&


Suggestions


Thank
You

Real Time Execution On GPUs

SIU
Southern Illinois University
CARBONDALE

Kiriti NageSh Gowda
Department of Electrical & Computer Engineering

ReSearch AdviSer: Dr. Harini RamaprasSad

Committee Chair: Dr. Shaikh Ahmed