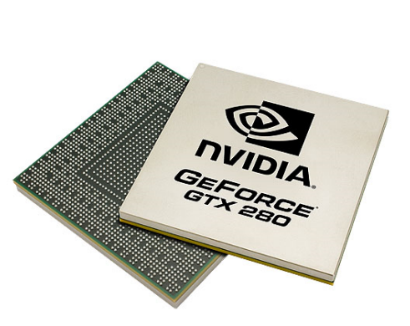
Computer Architecture and Systems Laboratory

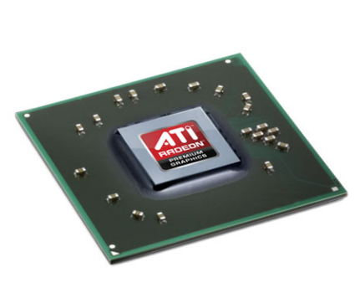
GPU Ocelot: Dynamic Compilation for PTX

Andrew Kerr, Gregory Diamos, Naila Farooqui, Jeff Young, and Sudhakar Yalamanchili
Georgia Institute of Technology
{arkerr@gatech.edu, gtg250v@mail.gatech.edu, naila@gatech.edu, jyoung9@gatech.edu sudha@ece.gatech.edu}

Project Goals



Efficient execution of data-parallel kernels on heterogeneous platforms



CUDA on multiple architectures: NVIDIA GPU, Multicore CPUs, Vector ISAs, and AMD GPUs

Performance scalability and portability

Developer Productivity

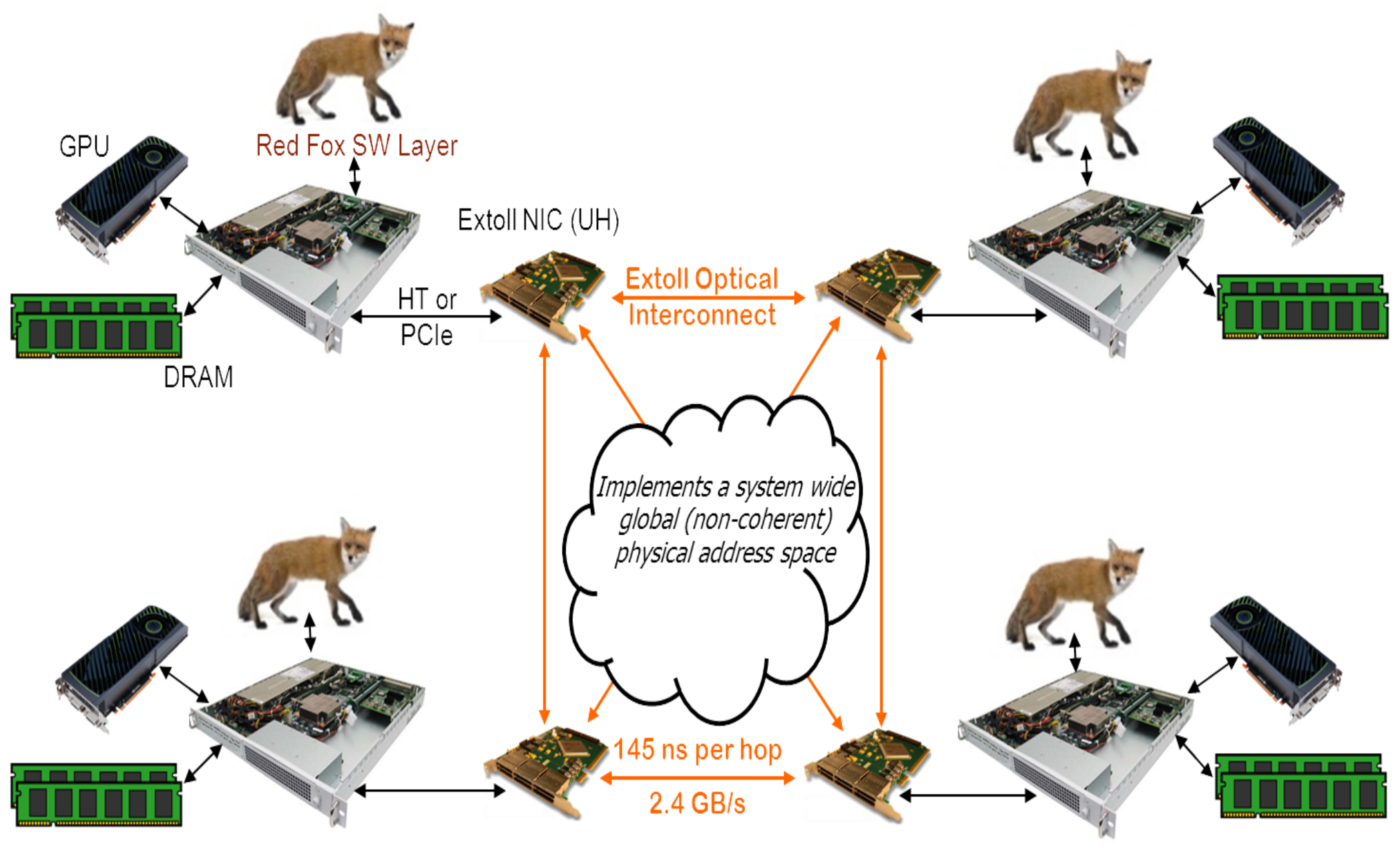
Sponsors: NSF, LogicBlox, NVIDIA

Oncilla Hardware Infrastructure

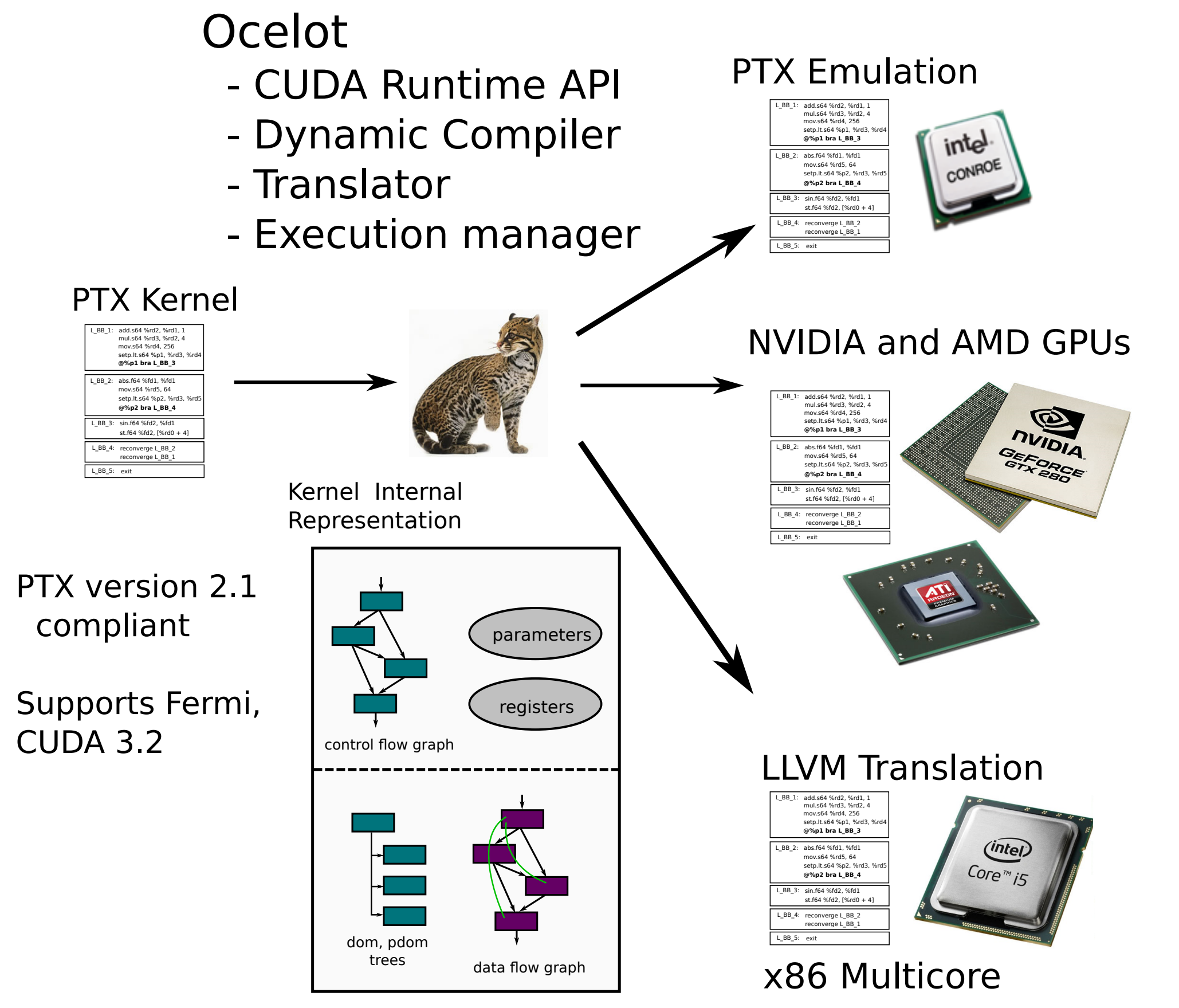
- Prototype hardware to support non-coherent Global Address Spaces for accelerated data warehousing applications

-Oncilla will support efficient data movement through low-latency put/get operations between nodes using HT and EXTOLL interconnects

-Collaboration with University of Heidelberg, Polytechnic University of Valencia, AIC Inc., LogicBlox Inc.
-Sponsors: NVIDIA


GPU
Red Fox SW Layer
Extoll NIC (UH)
HT or PCIe
DRAM
Extoll Optical Interconnect
Implements a system wide global (non-coherent) physical address space
145 ns per hop
2.4 GB/s

Ocelot Overview



Ocelot
- CUDA Runtime API
- Dynamic Compiler
- Translator
- Execution manager

PTX Kernel

Kernel Internal Representation

PTX version 2.1 compliant

Supports Fermi, CUDA 3.2

Translate PTX kernels to architectures beyond GPUs

Link with existing CUDA applications

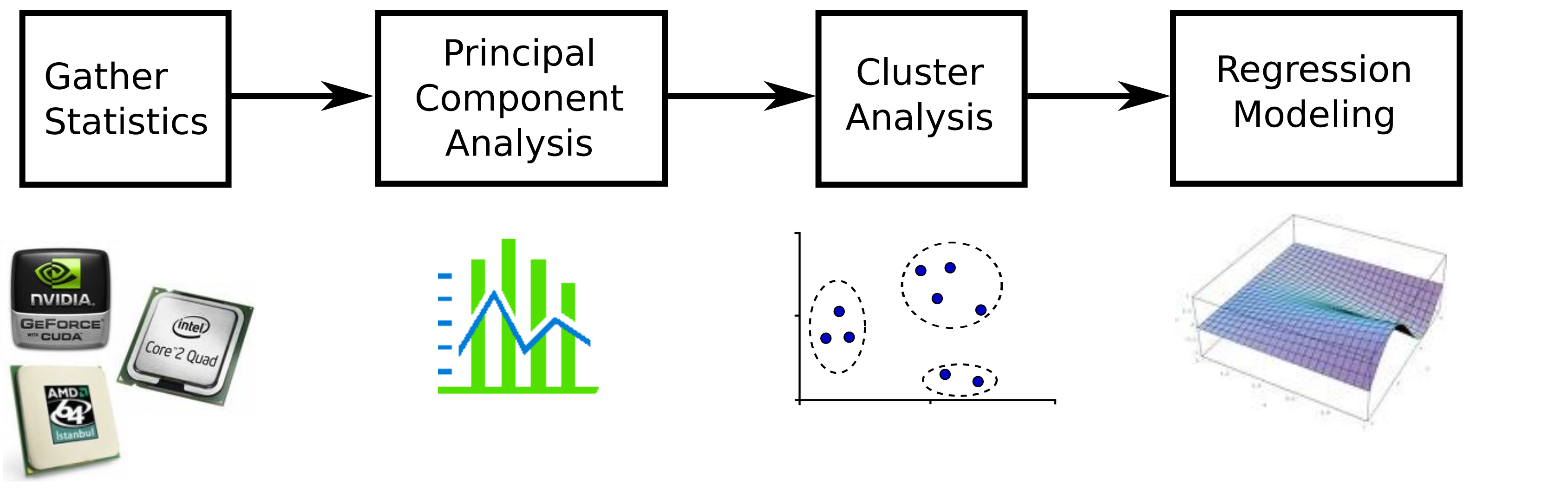
Execution on several architectures

Instrument and profile kernel execution

Explore novel GPU architectures

available on Google Code: **http://code.google.com/p/gpuocelot**

Performance Modeling [2]



Gather Statistics → Principal Component Analysis → Cluster Analysis → Regression Modeling

Application metrics collected via: static analysis, instrumentation, emulation

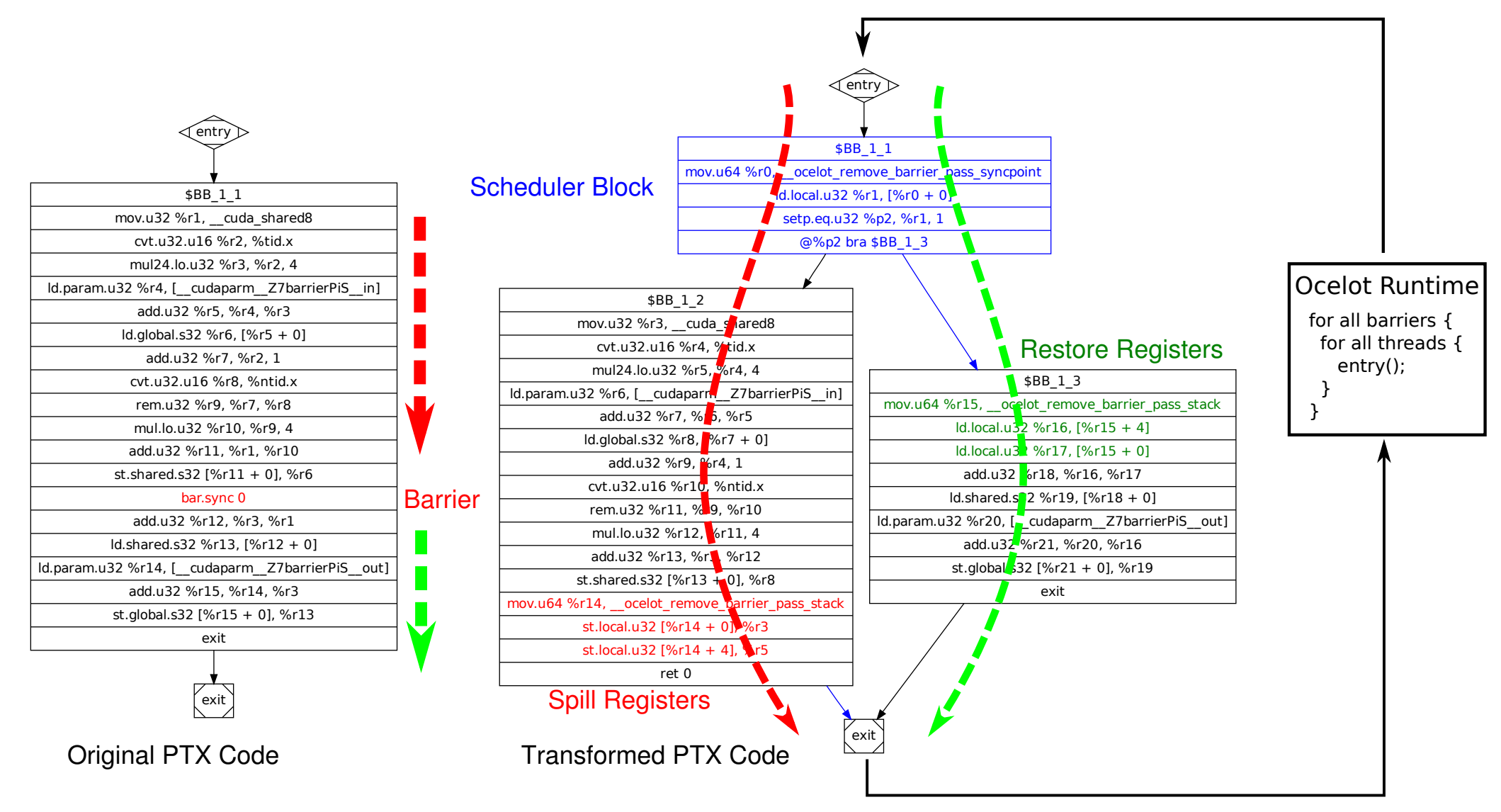
Correlated metrics detected via Principal Component Analysis

Clusters of applications and machine models identified

Statistical performance model: predicted performance within 10%

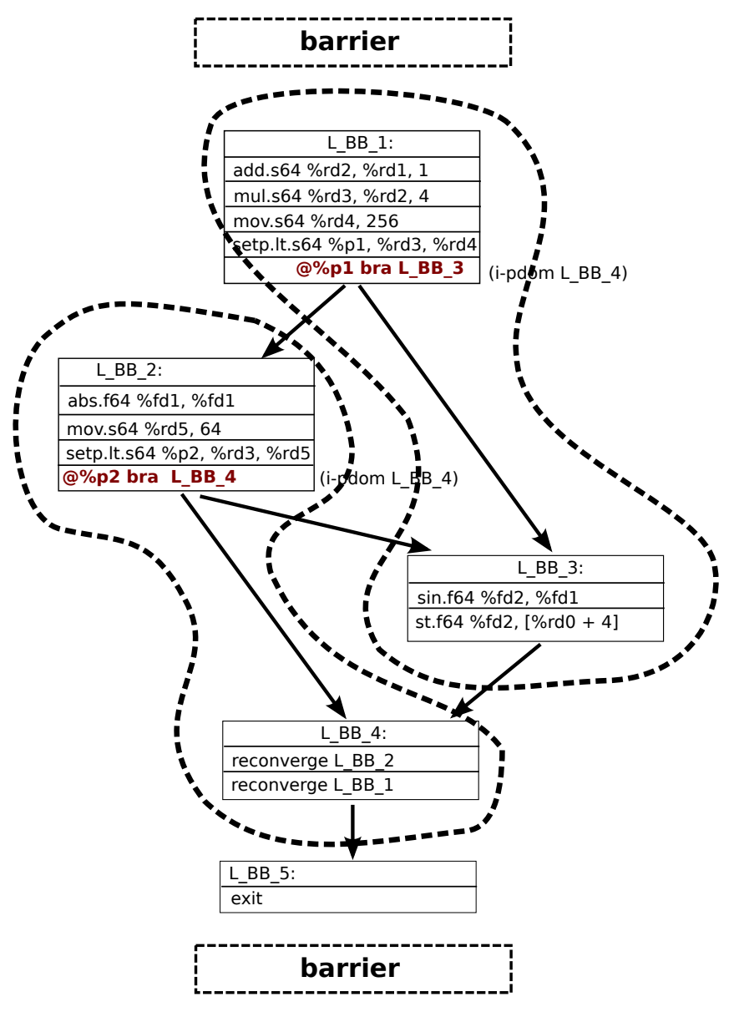
Non-GPU Execution Targets [3]

Multicore CPU [3]



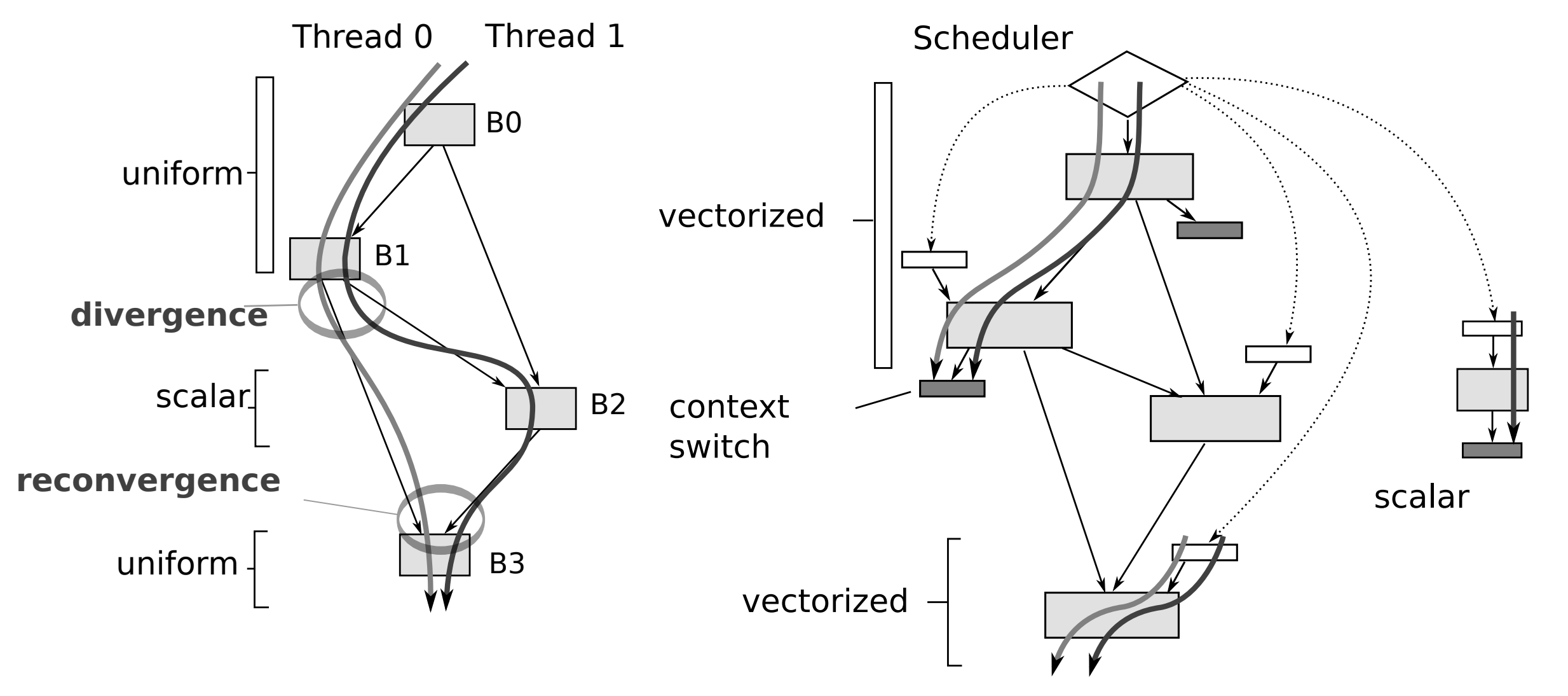
- Execute each CTA on a processore core
- Serialize threads within CTA, switch context at CTA-wide barriers
- Explore novel thread scheduling techniques

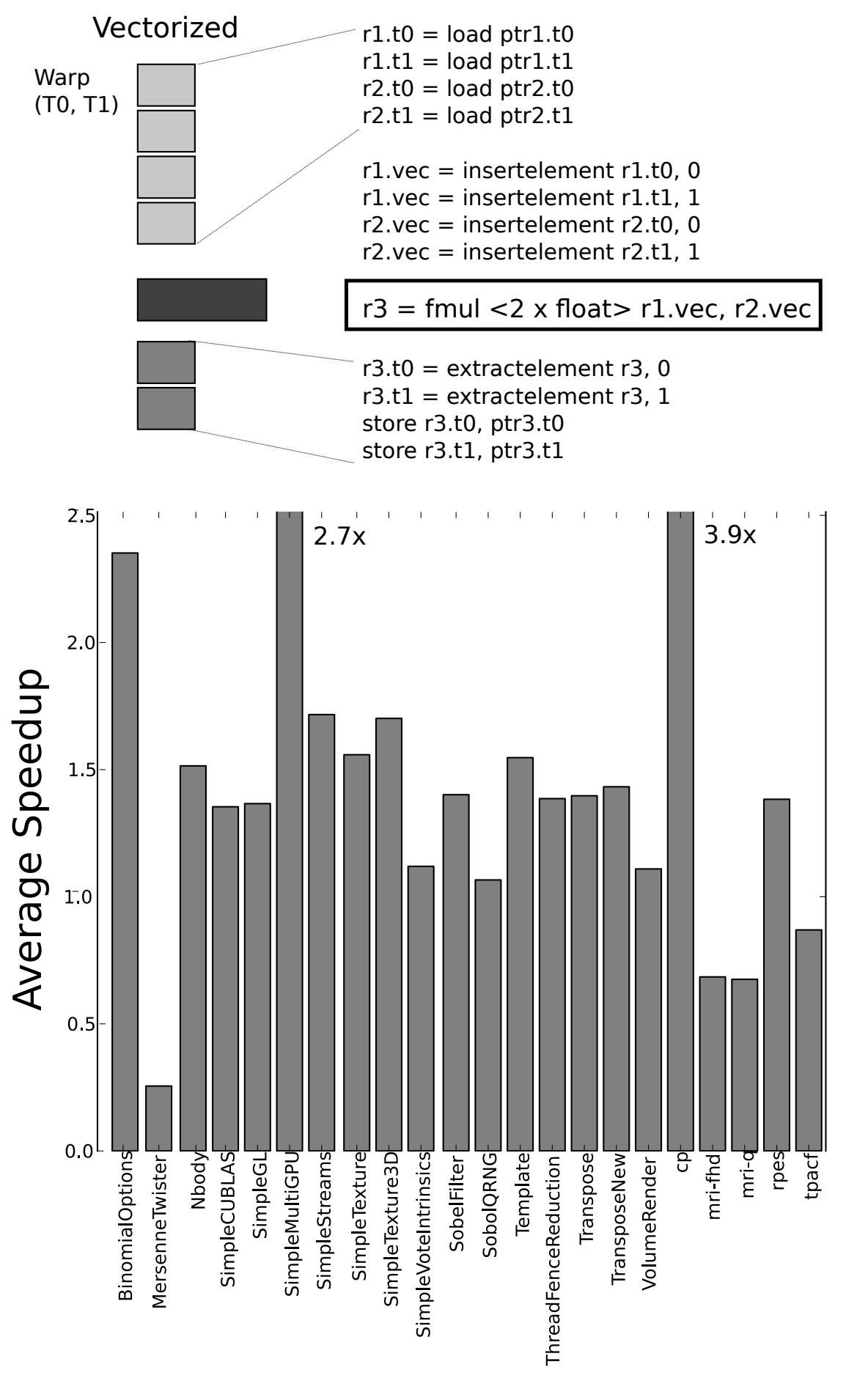
Subkernel Formation



- Partition kernels into subkernels
- Translate subkernels lazily
- Schedule subkernels on different processors or functional units

Vectorized Multicore Execution [6]



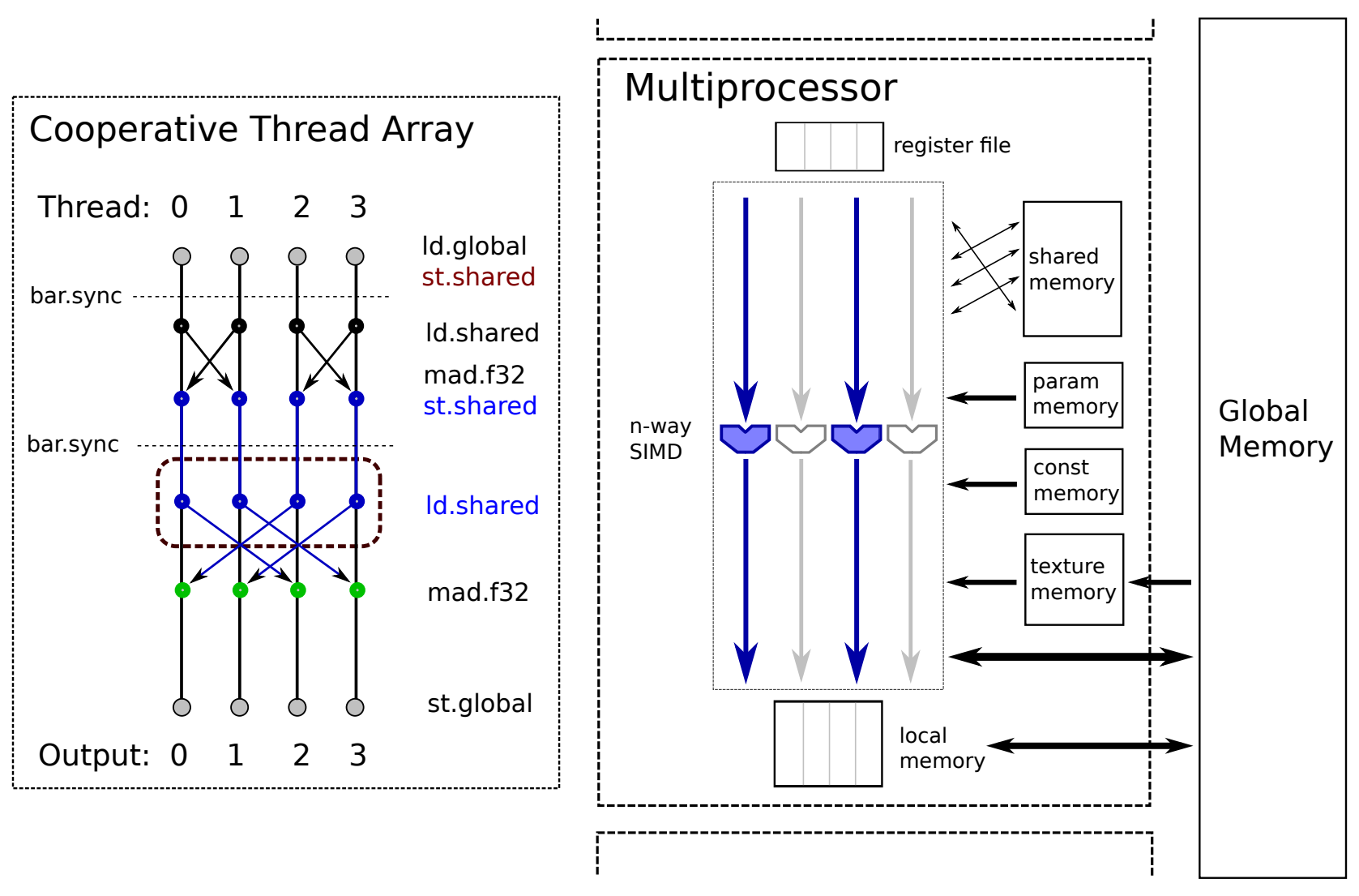


```
r1.t0 = load ptr1.t0
r1.t1 = load ptr1.t1
r2.t0 = load ptr2.t0
r2.t1 = load ptr2.t1

r1.vec = insertelement r1.t0, 0
r1.vec = insertelement r1.t1, 1
r2.vec = insertelement r2.t0, 0
r2.vec = insertelement r2.t1, 1

r3 = fmul <2 x float> r1.vec, r2.vec

r3.t0 = extractelement r3, 0
r3.t1 = extractelement r3, 1
store r3.t0, ptr3.t0
store r3.t1, ptr3.t1
```

- Transform scalar kernel into vectorized kernel
- Execution of a control path is logically equivalent to executing several PTX threads
- Detect control divergence and exit to execution manager

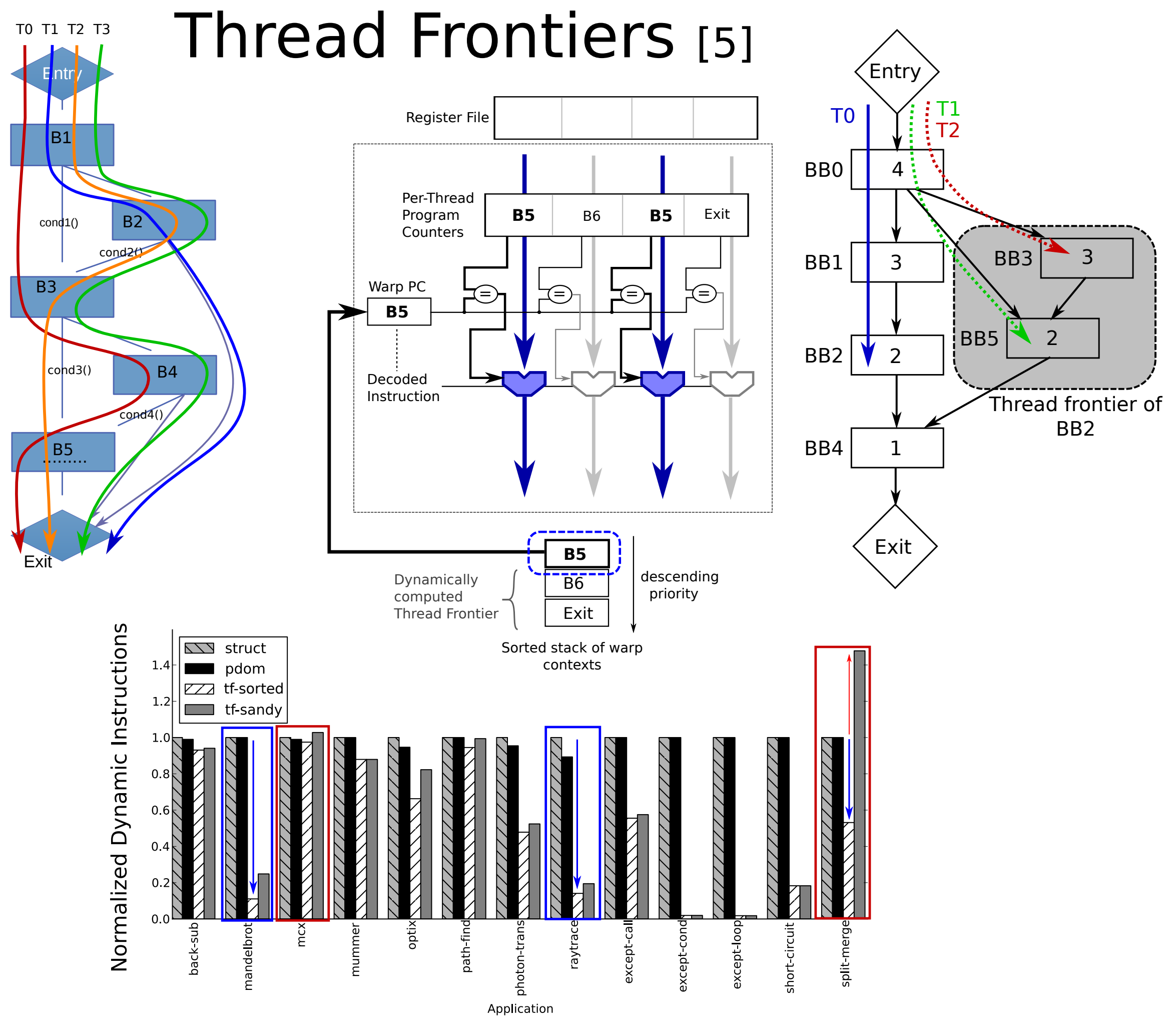
Workload Characteristics, Optimization, and Productivity Tools [1,4,5,7]

PTX Emulation [1,4]

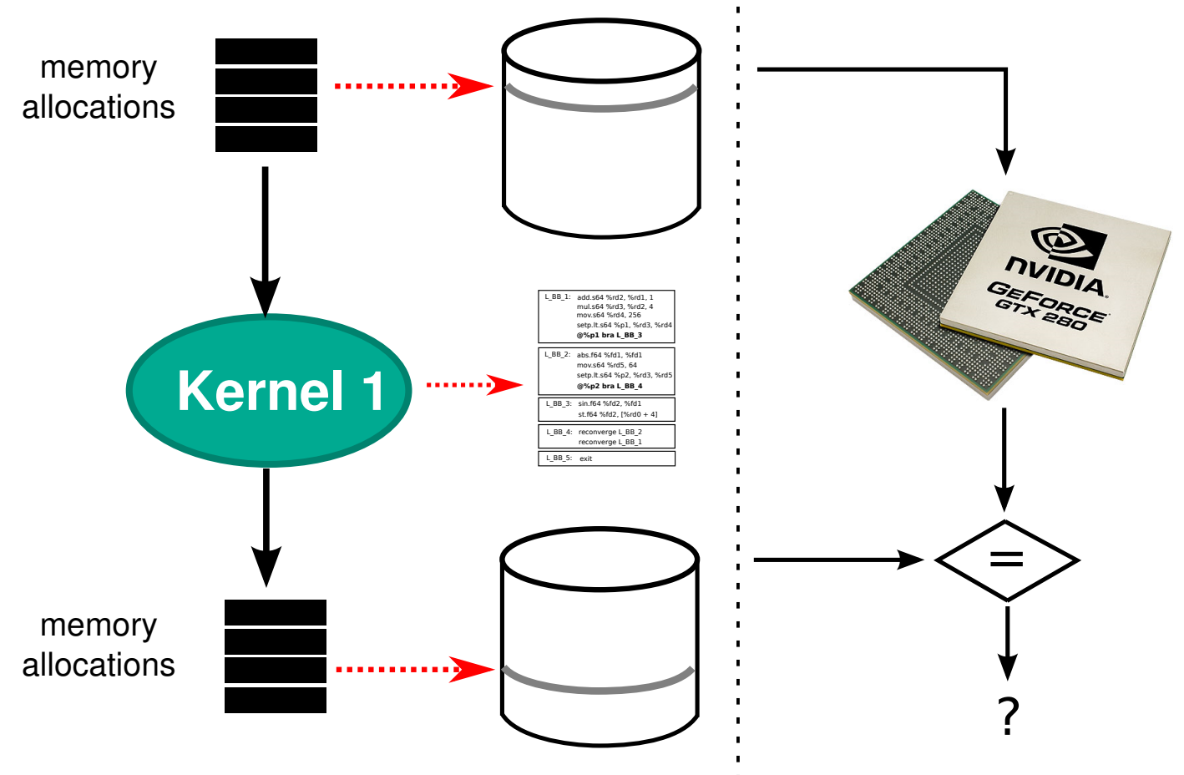


- Workload characterization
- Correctness Tools
- Performance Tuning
- Architecture research
- Cycle-accurate simulator driver

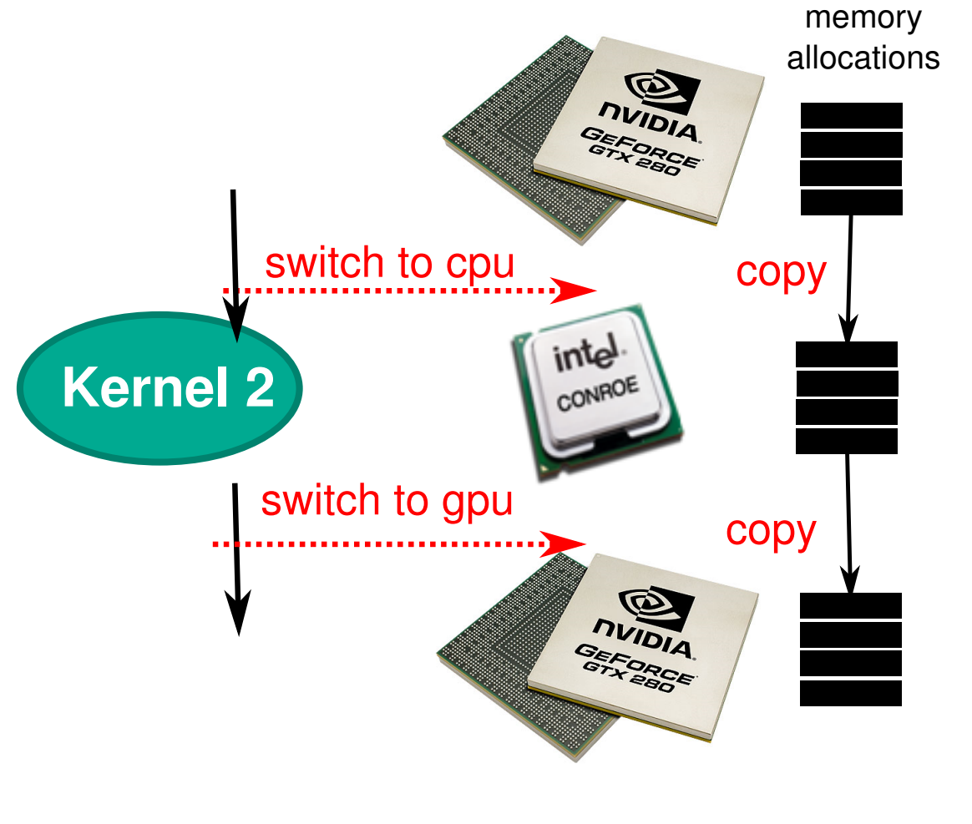
Thread Frontiers [5]



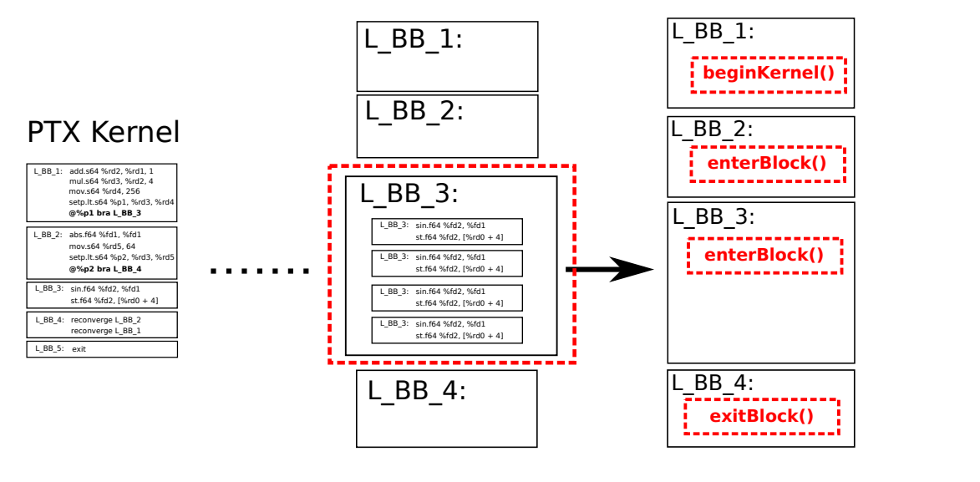
Checkpointing [5]



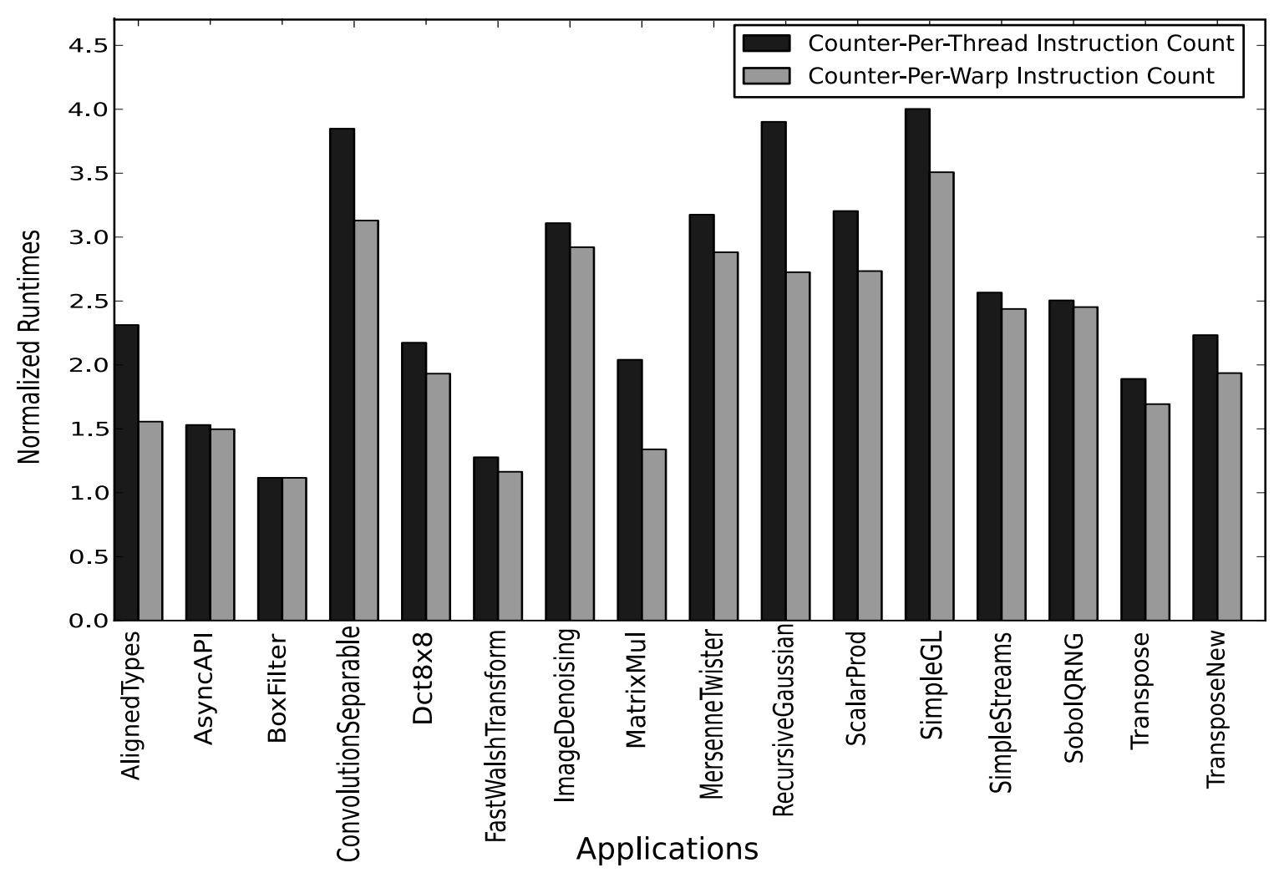
CUDA Device Switching



Kernel Instrumentation [7]



Dynamically transform and instrument PTX kernels



References

[1] Kerr, Diamos, Yalamanchili. "Workload Characterization of PTX Kernels." IISWC 2009

[2] Kerr, Diamos, Yalamanchili. "Modeling GPU-CPU Workloads." GPGPU-3 2010

[3] Diamos, Kerr, Yalamanchili. "Ocelot: A Dynamic Optimization Framework for Bulk-Synchronous Applications in Heterogeneous Systems." PACT 2010

[4] Kerr, Diamos, Yalamanchili. "GPU Application Development, Debugging, and Performance Tuning with GPU Ocelot" GPU Computing Gems vol. 2. 2011

[5] Diamos, A. Ashbaugh, S. Maiyuran, Kerr, Wu, Yalamanchili. "SIMD Re-convergence at Thread Frontiers" MICRO-44 2011.

[6] Kerr, Diamos, Yalamanchili. "Dynamic Compilation of Data-Parallel Kernels for Vector Processors" CGO 2012

[7] Farooqui, Kerr, Diamos, Yalamanchili, Schwan. "Lynx: Dynamic Instrumentation System for Data-Parallel Applications on GPU Architectures" ISPASS'12