Improving GPU Performance Prediction with Data Transfer Modeling

Michael Boyer
University of Virginia
boyer@cs.virginia.edu

Jiayuan Meng and Kalyan Kumaran
Leadership Computing Facility
Argonne National Laboratory

GPU: Potential

	CPU	GPU	Increase
Throughput (TFLOPS)	0.2	4	5.6x
Memory bandwidth (GB/s)	51	288	20.7x
Power consumption (W)	150	250	1.7x

Potential for massive performance increases

GPU: Challenges

- *Porting* an application to the GPU is non-trivial
  - How to parallelize is not always obvious
  - Lots of boilerplate code required (memory allocation, data transfers, etc.)
- *Optimizing* an application on the GPU is challenging
  - Requires intimate knowledge of the hardware
  - Large space of potential optimizations

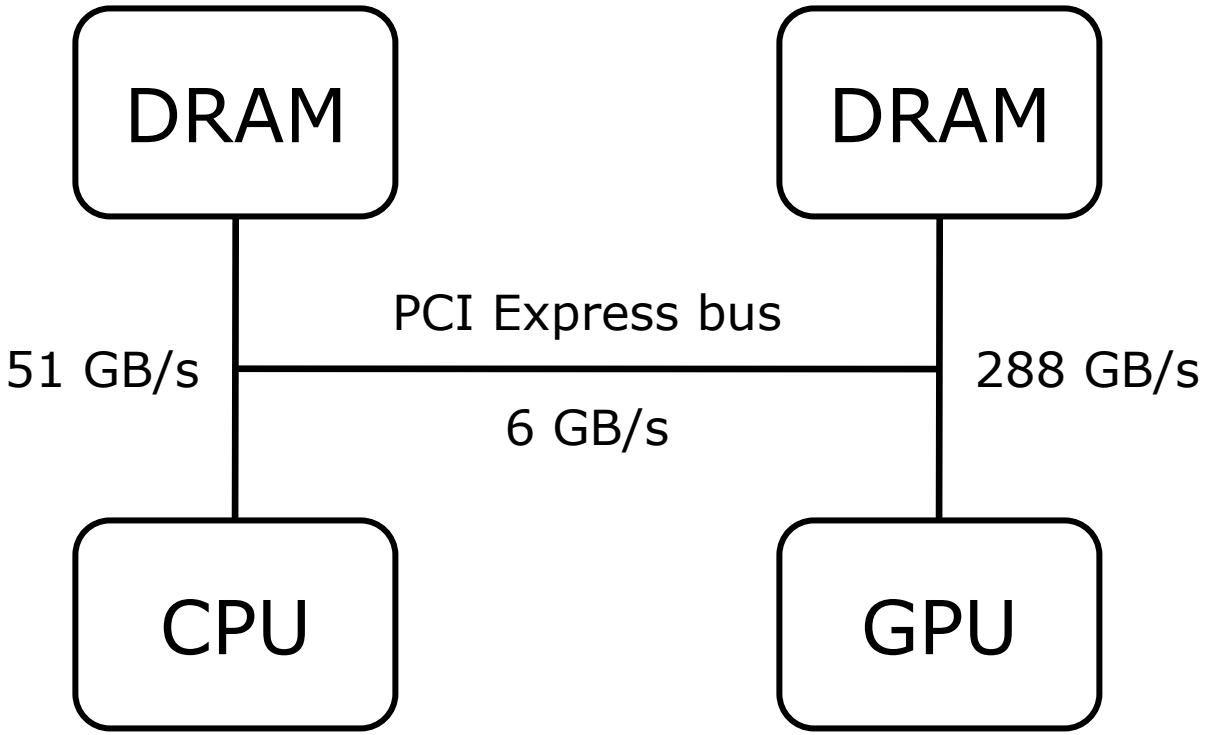
Question:

Should I port my application for execution on a GPU?

GROPHECY

- Input: code skeleton
- Output:
  - Projected performance achievable with a GPU
  - Transformations/optimizations necessary to achieve that performance

System Architecture



Example: Vector Addition

- Problem: add together two 1GB arrays

- Compute time:
  - CPU: 59 ms
  - GPU: 10 ms
  - GPU 6x faster!

- CPU-GPU transfer time: 500 ms
- Total time:
  - CPU: 59 ms
  - GPU: 10 + 500 = 510 ms
  - CPU 9x faster!

Goal: augment GROPHECY to account for data transfers

Data Transfer Model

- Input: Code skeleton
- Output: Predicted data transfer time

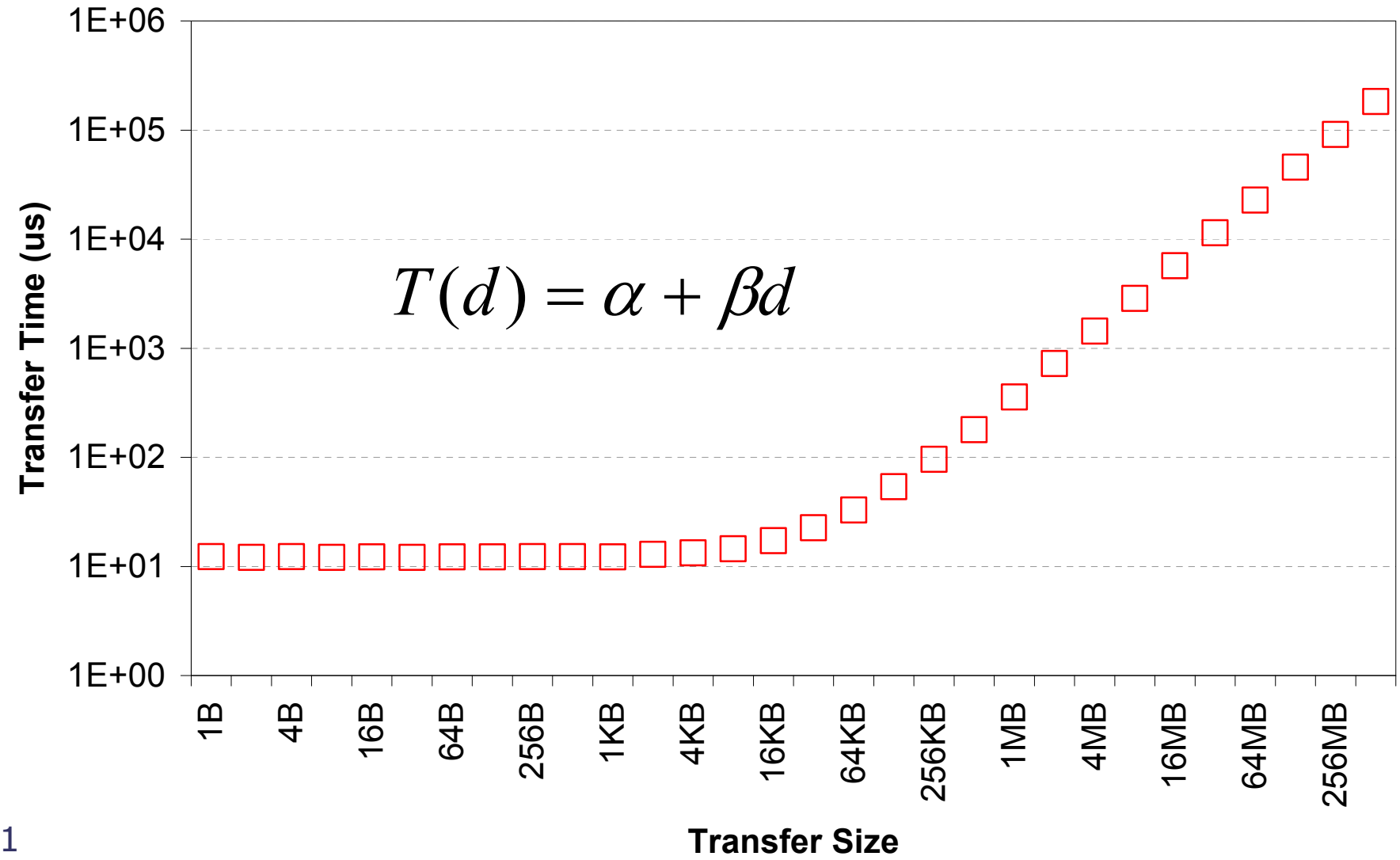
- Steps:
  1. Determine how much data needs to be transferred
  2. Predict how long the data transfer(s) will take

Determining Transfer Size

- Use Bounded Regular Sections (BRS) to track the range of array elements read and written by the kernel

- BRS is unknown:
  - Assume entire array is accessed
- BRS is read:
  - Input data (transferred to the GPU)
- BRS is written:
  - Output data (transferred back to the CPU)

Transfer Time

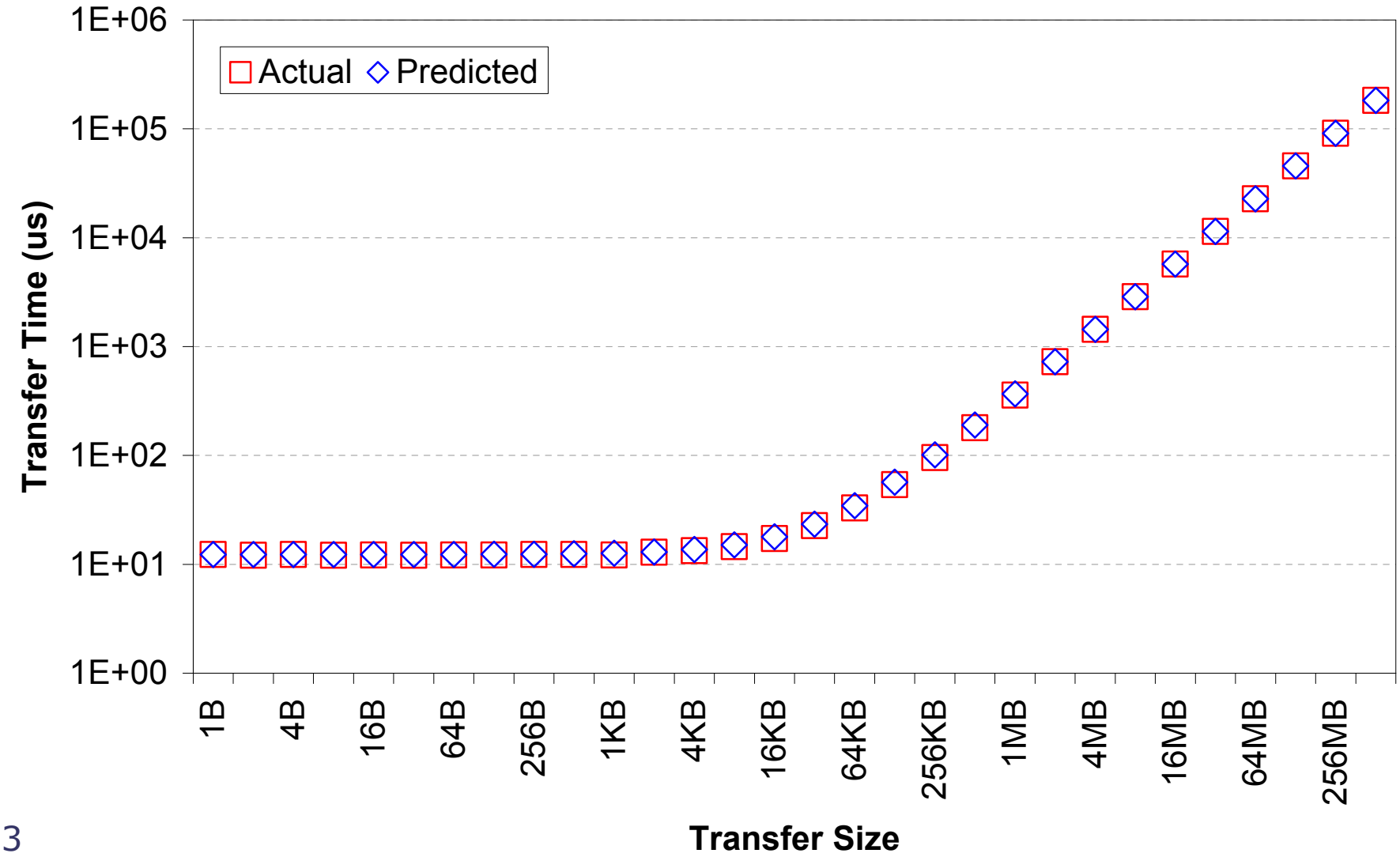


Building the Transfer Time Model

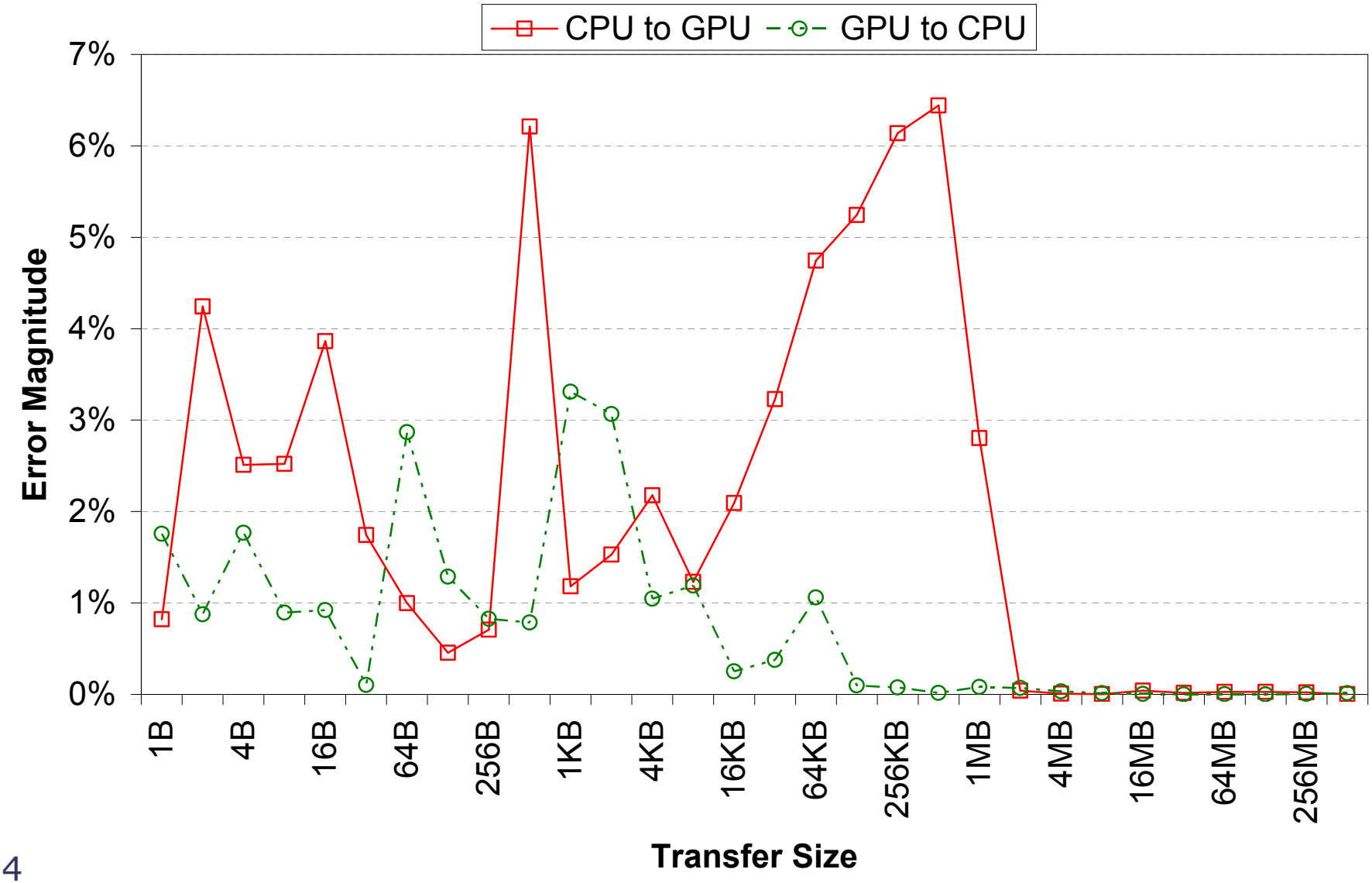
$$T(d) = \alpha + \beta d$$

- Measuring alpha:
  - t_S = transfer time of a single byte
  - $\alpha = t_s$
- Measuring beta:
  - s_L = transfer size (512 MB)
  - t_L = transfer time
  - β = tL / 512 MB

Validation: Synthetic Benchmark



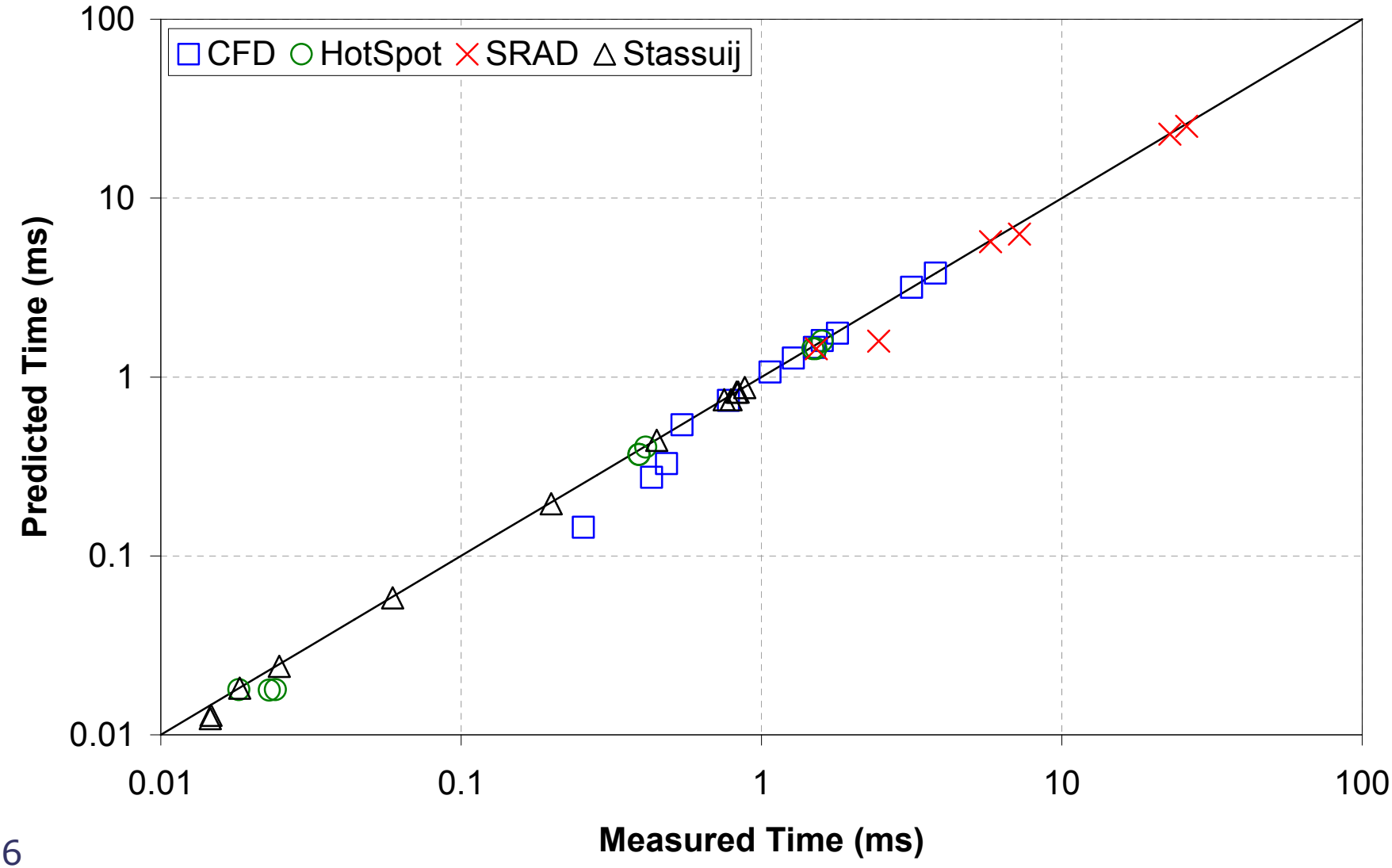
Validation: Synthetic Benchmark



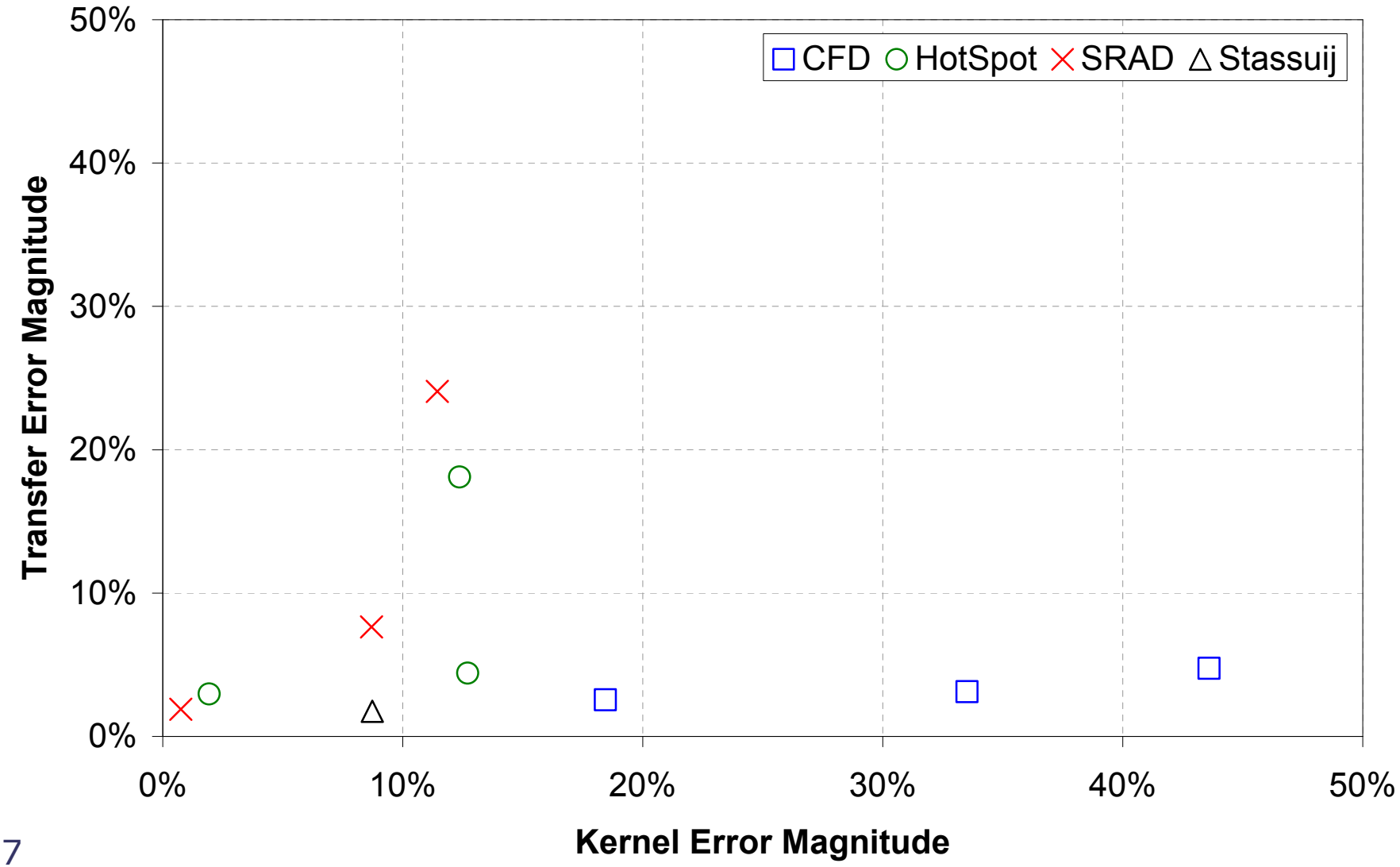
Experimental Setup

- Applications:
  - CFD (Computational Fluid Dynamics)
  - HotSpot
  - SRAD (Speckle Reducing Anisotropic Diffusion)
  - Stassuij
- Hardware:
  - CPU: Intel Xeon E5405
  - GPU: NVIDIA Quadro FX 5600 (PCIe v1)

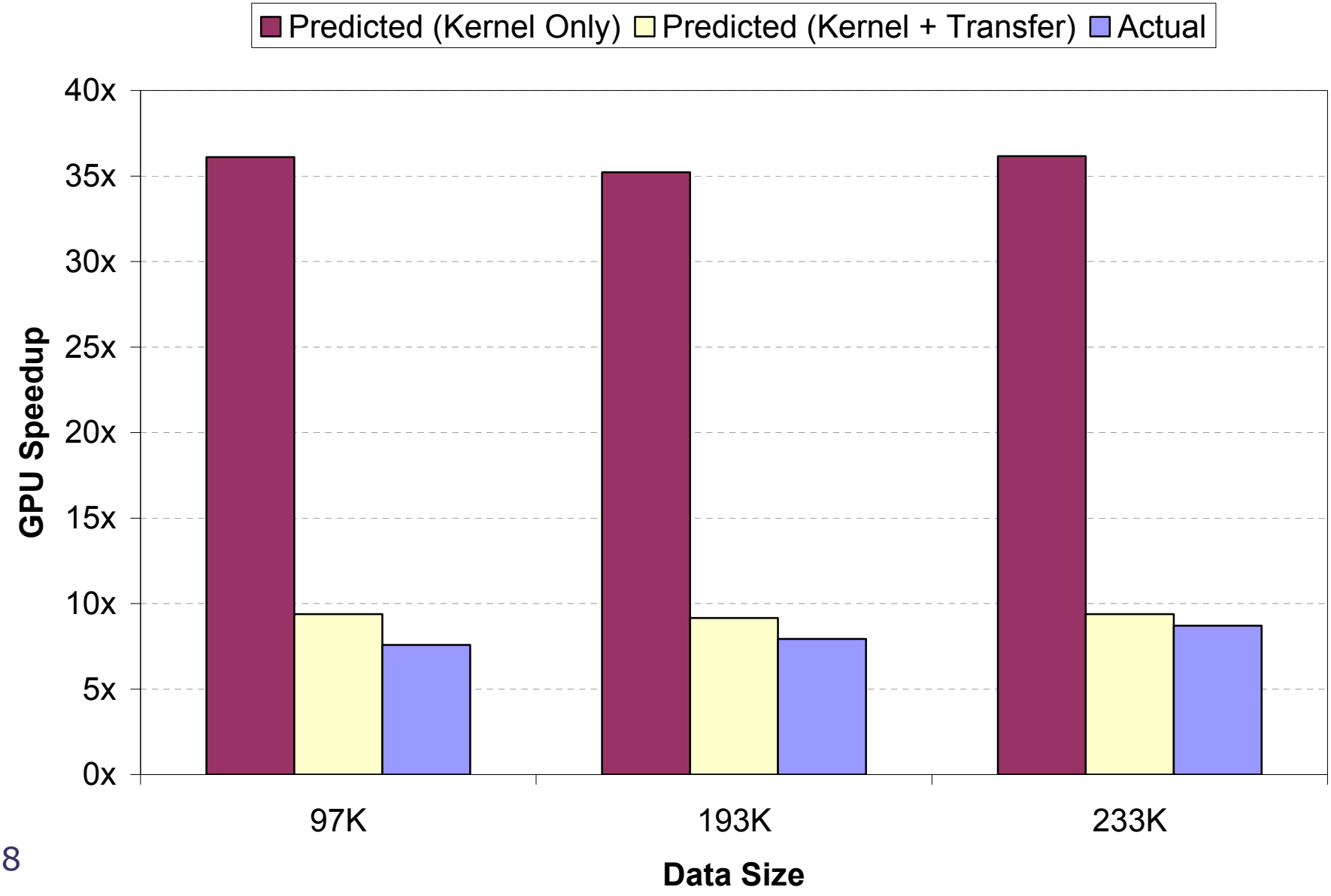
Validation: Applications



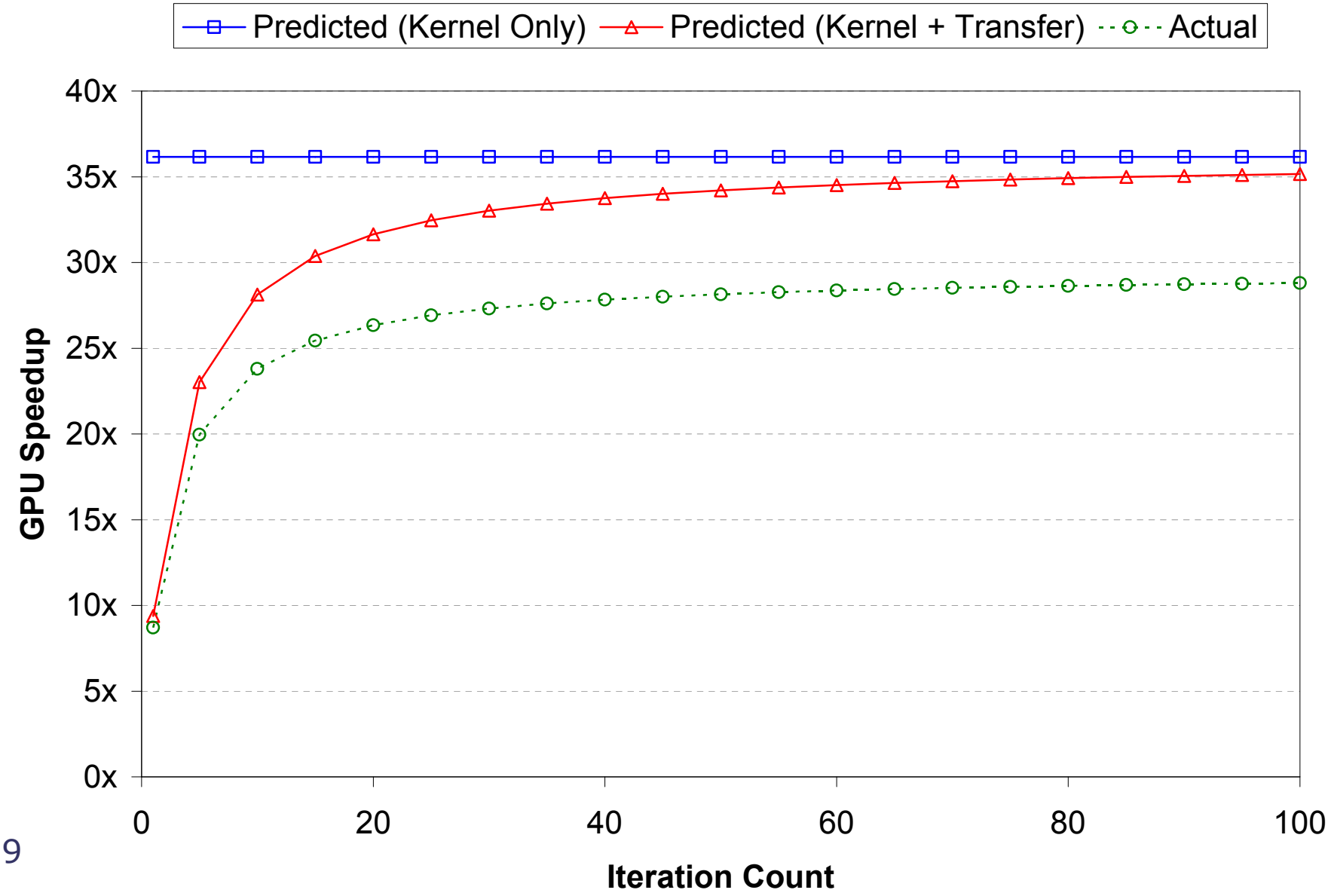
Prediction Errors



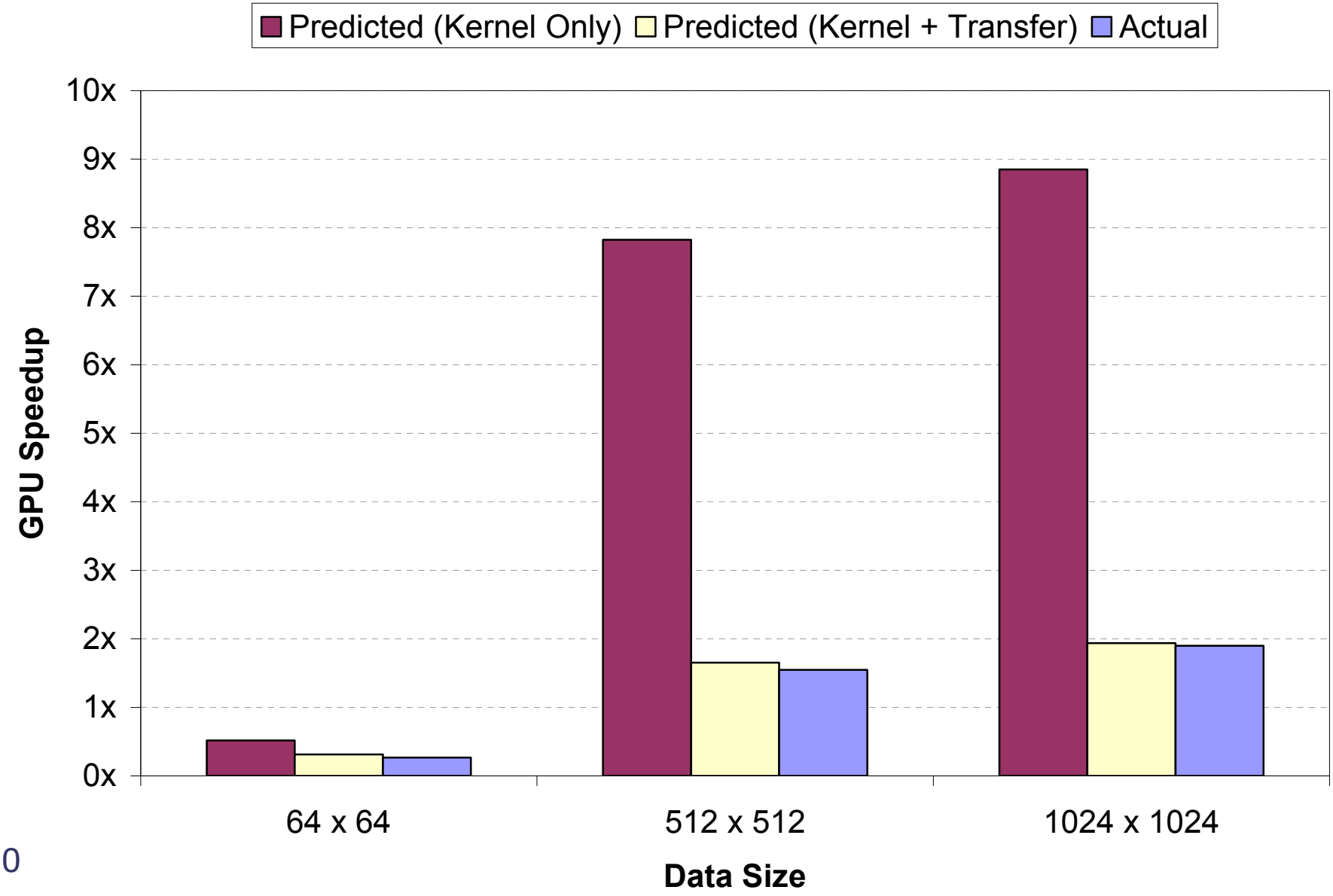
CFD


Predicted (Kernel Only)
Predicted (Kernel + Transfer)
Actual
40x
35x
30x
25x
20x
15x
10x
5x
0x
GPU Speedup
97K
193K
233K
Data Size

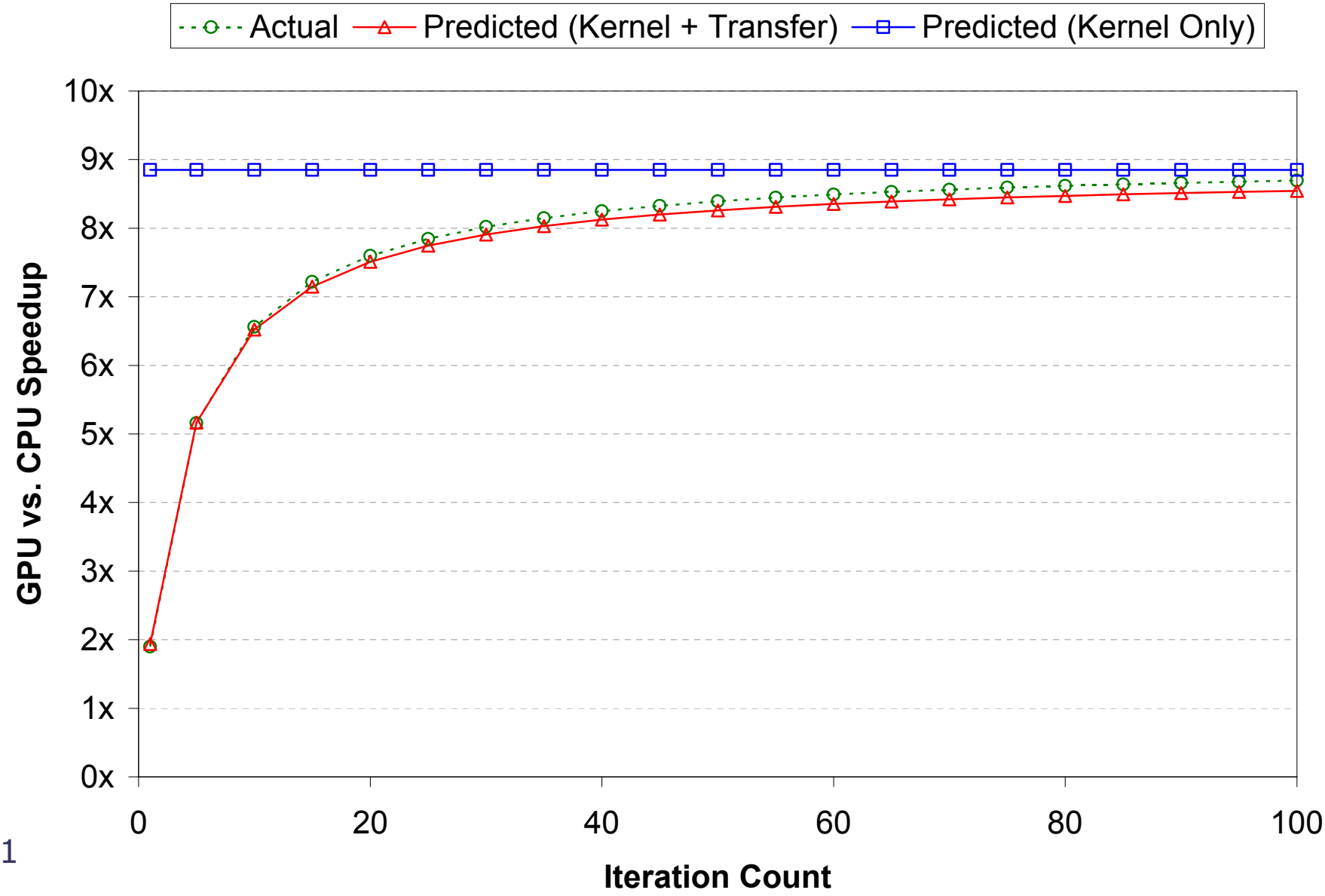
CFD


Predicted (Kernel Only)
Predicted (Kernel + Transfer)
Actual
40x
35x
30x
25x
20x
15x
10x
5x
0x
GPU Speedup
0
20
40
60
80
100
Iteration Count

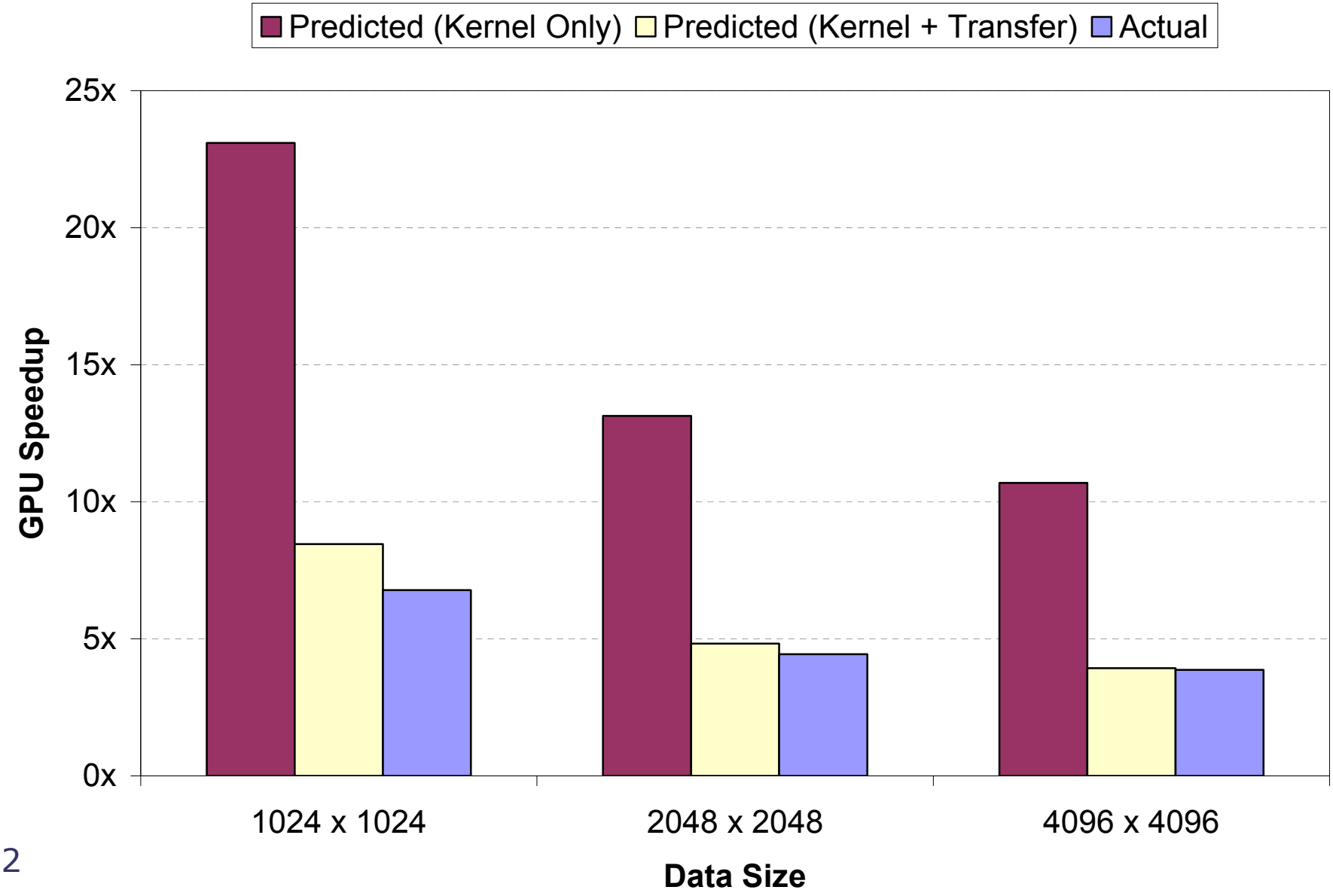
HotSpot


Predicted (Kernel Only)
Predicted (Kernel + Transfer)
Actual
10x
9x
8x
7x
6x
5x
4x
3x
2x
1x
0x
GPU Speedup
64 x 64
512 x 512
1024 x 1024
Data Size

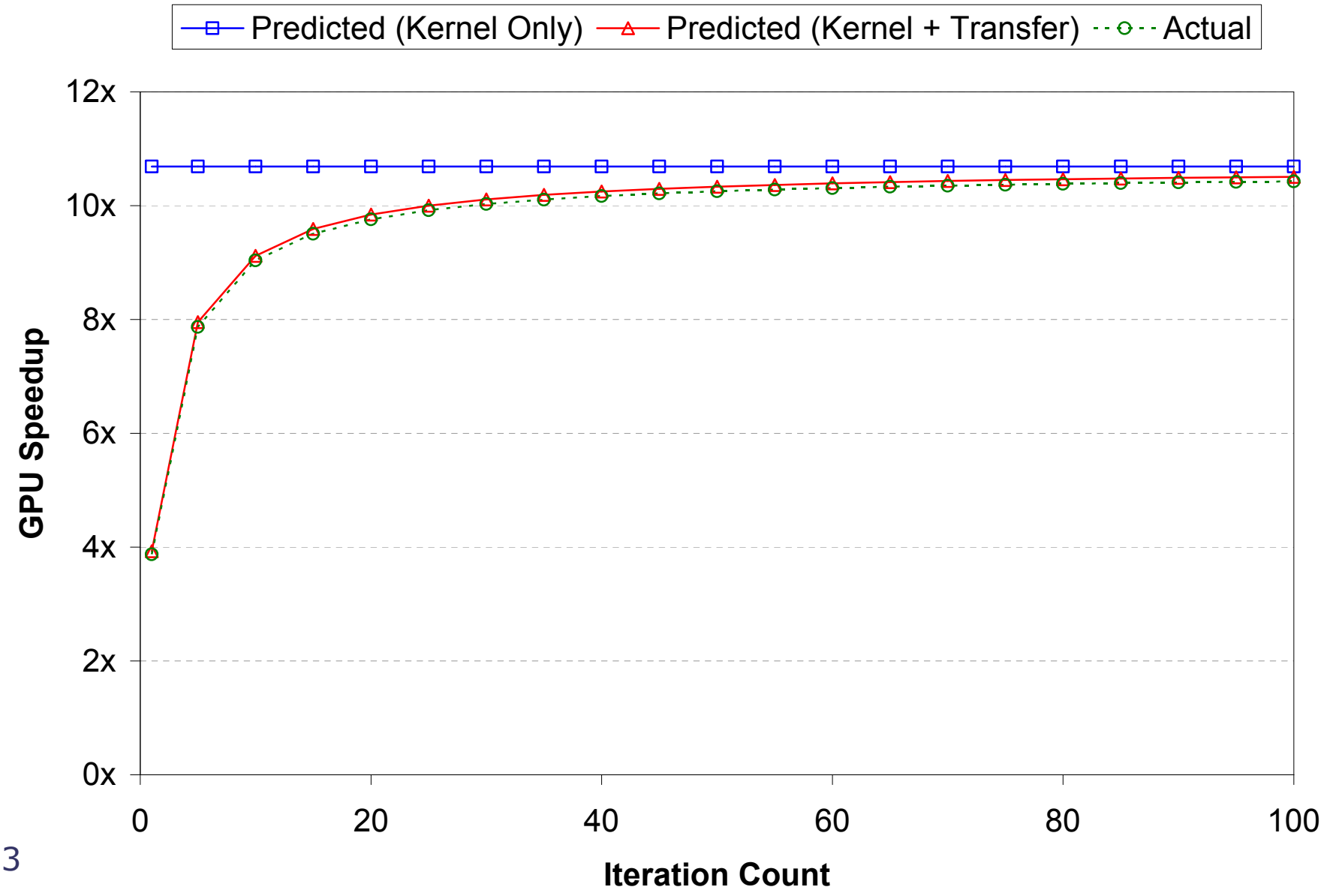
HotSpot


Actual
Predicted (Kernel + Transfer)
Predicted (Kernel Only)
10x
9x
8x
7x
6x
5x
4x
3x
2x
1x
0x
GPU vs. CPU Speedup
0
20
40
60
80
100
Iteration Count

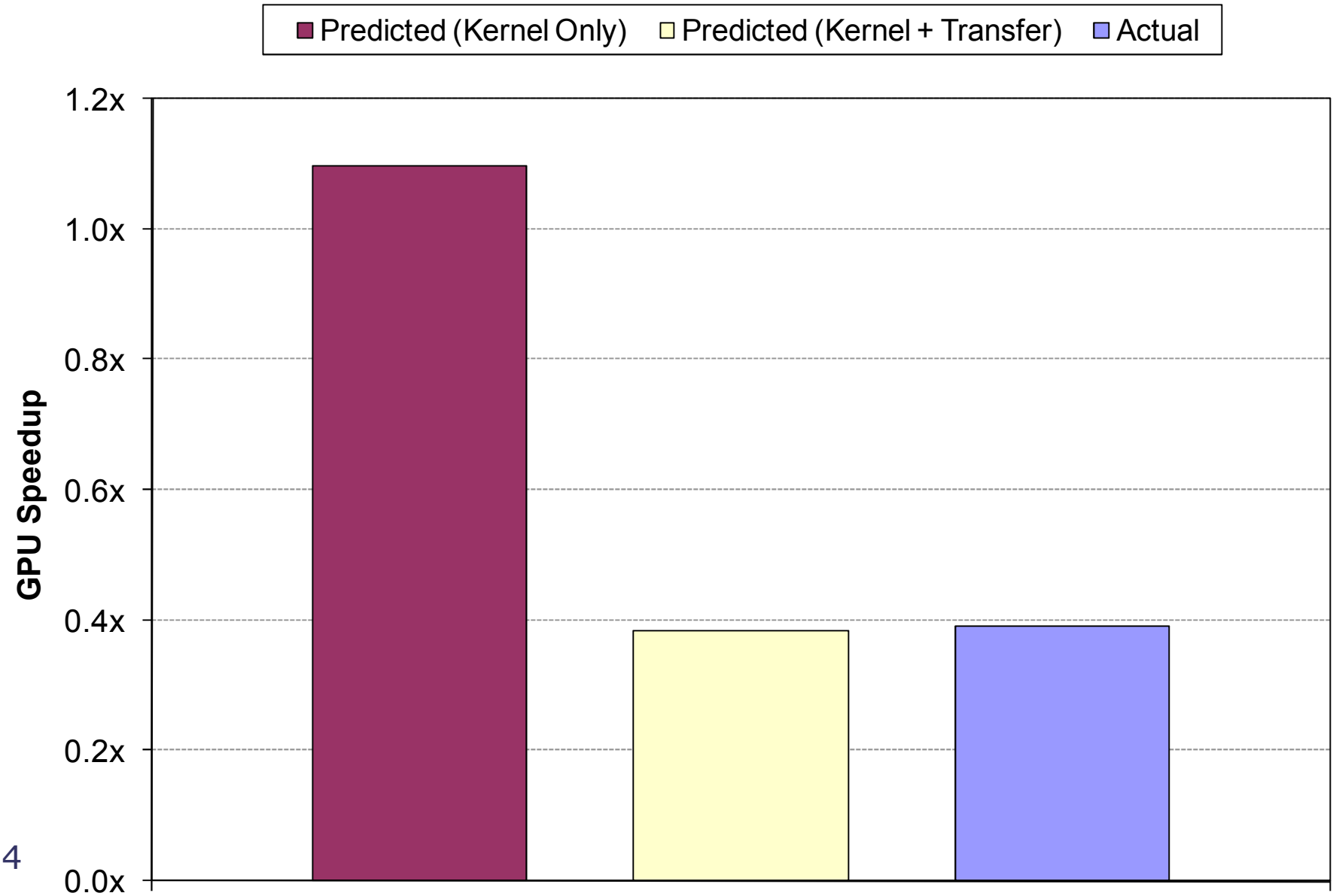
SRAD


Predicted (Kernel Only)
Predicted (Kernel + Transfer)
Actual
25x
20x
15x
10x
5x
0x
GPU Speedup
1024 x 1024
2048 x 2048
4096 x 4096
Data Size

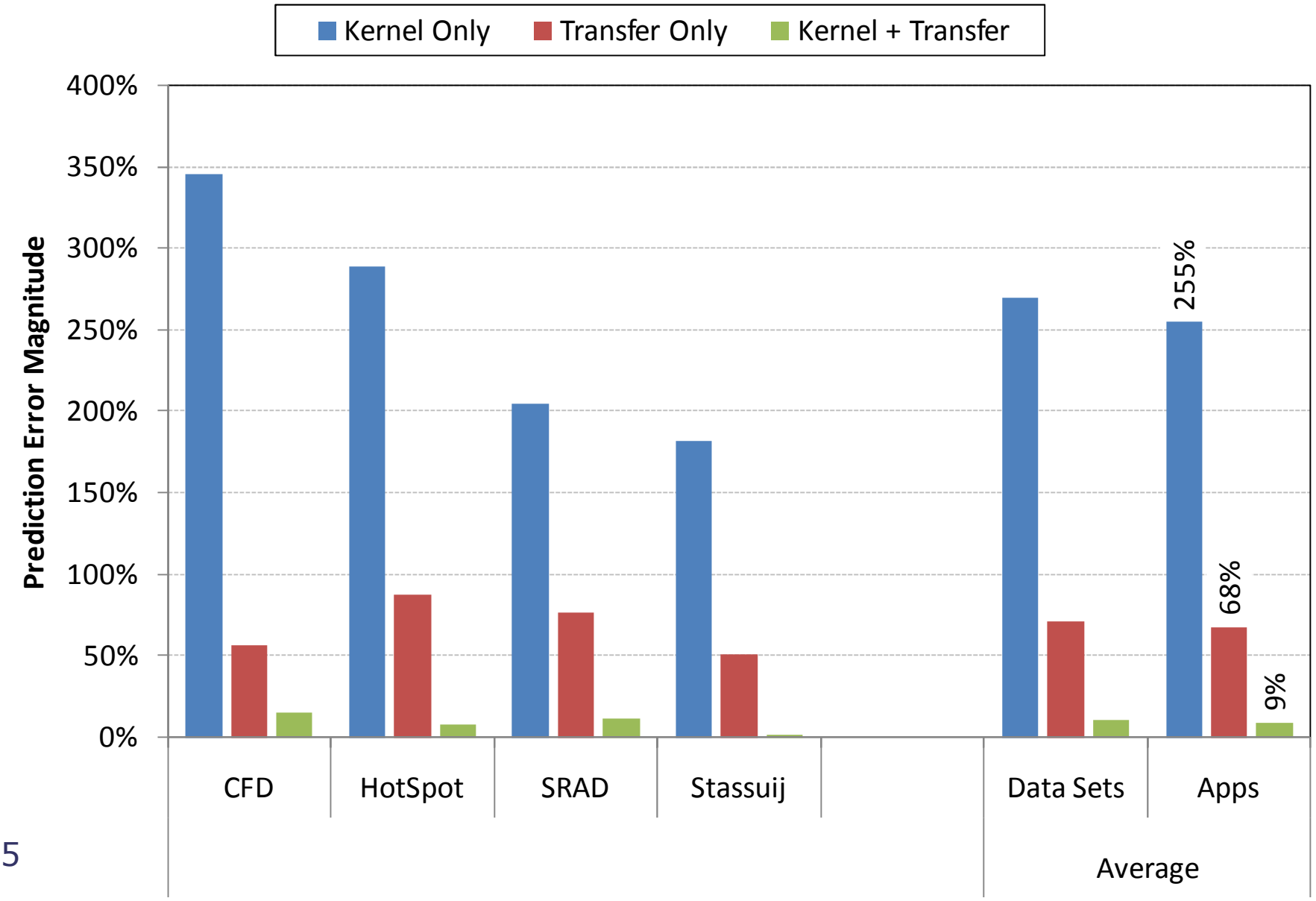
SRAD


Predicted (Kernel Only)
Predicted (Kernel + Transfer)
Actual
12x
10x
8x
6x
4x
2x
0x
GPU Speedup
0
20
40
60
80
100
Iteration Count

Stassuij


Predicted (Kernel Only)
Predicted (Kernel + Transfer)
Actual
GPU Speedup
1.2x
1.0x
0.8x
0.6x
0.4x
0.2x
0.0x

Predicted GPU Speedup Accuracy



Conclusions

- Data transfer overhead can be a bottleneck for GPU applications
- We need to account for this overhead when considering porting to a GPU
- We extend GROPHECY with a data transfer model, which determines:
  - The amount of data to be transferred
  - The time to transfer the data
- Our model:
  - Predicts transfer time with an average error of 8%
  - Reduces error in the predicted GPU speedup from 255% to 9%

Future Work

- Validate the model across a wider range of systems and applications:
  - More GPUs
  - PCIe v2 and v3
- Consider tradeoffs of different data transfer methods:
  - Pinned vs. pageable memory
  - Zero copy memory
  - Overlapping data transfer and kernel execution

Questions?