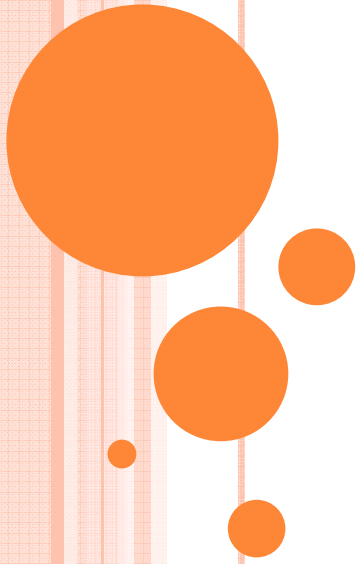


Effective Kernel Mapping for OpenCL Applications in Heterogeneous Platforms

Omer Erdil Albayrak, Ismail Akturk, Ozcan Ozturk

Bilkent University, Ankara, Turkey

Outline

- Introduction
- System Overview
- Our Approach
- Results
- Future Work
- Conclusion

Introduction

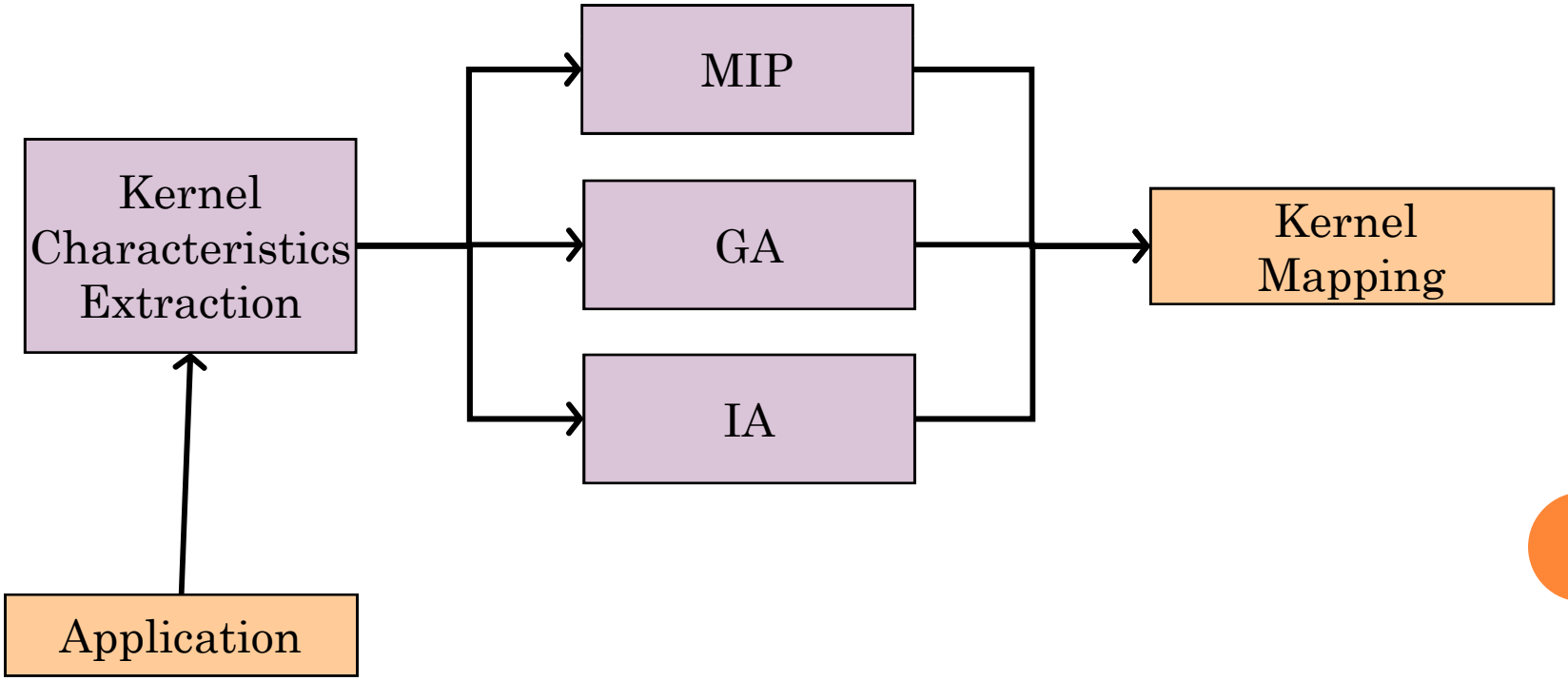
- Parallel computing systems consist of different types of accelerators
- Applications that are running on these systems have also different needs
  - Processing
  - Memory
  - Communication
  - Storage
- Use resources effectively
  - Application performance
  - Resource utilization

INTRODUCTION

- Applications composed of multiple exclusive regions
  - Kernels
  - Variation in characteristics and requirements
- Our Approach
  - Profiling-based kernel mapping algorithm
  - Multi-kernel applications
  - Heterogeneous platforms

System Overview

- Collect statistics about the kernels
- Select the best device for each kernel
- Execute the kernel on the chosen device

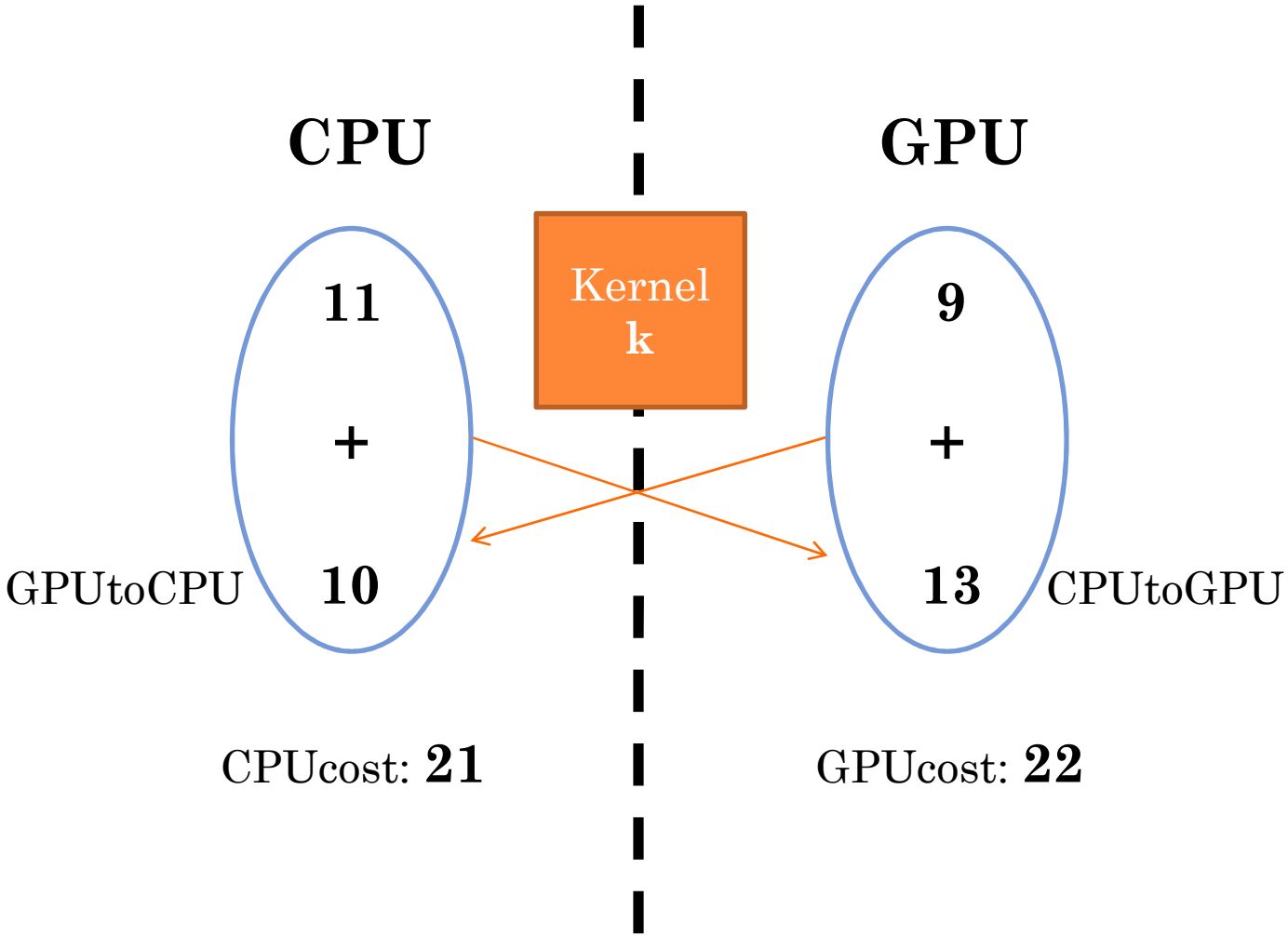


Greedy Algorithm

- Find the best device for each kernel starting from the first one

$$CPUcost_k = CPUrunningtime_k + \sum_{d=1}^{n} DeviceToHost \times InDevice_d \times Required_{k,d} \times size_d.$$

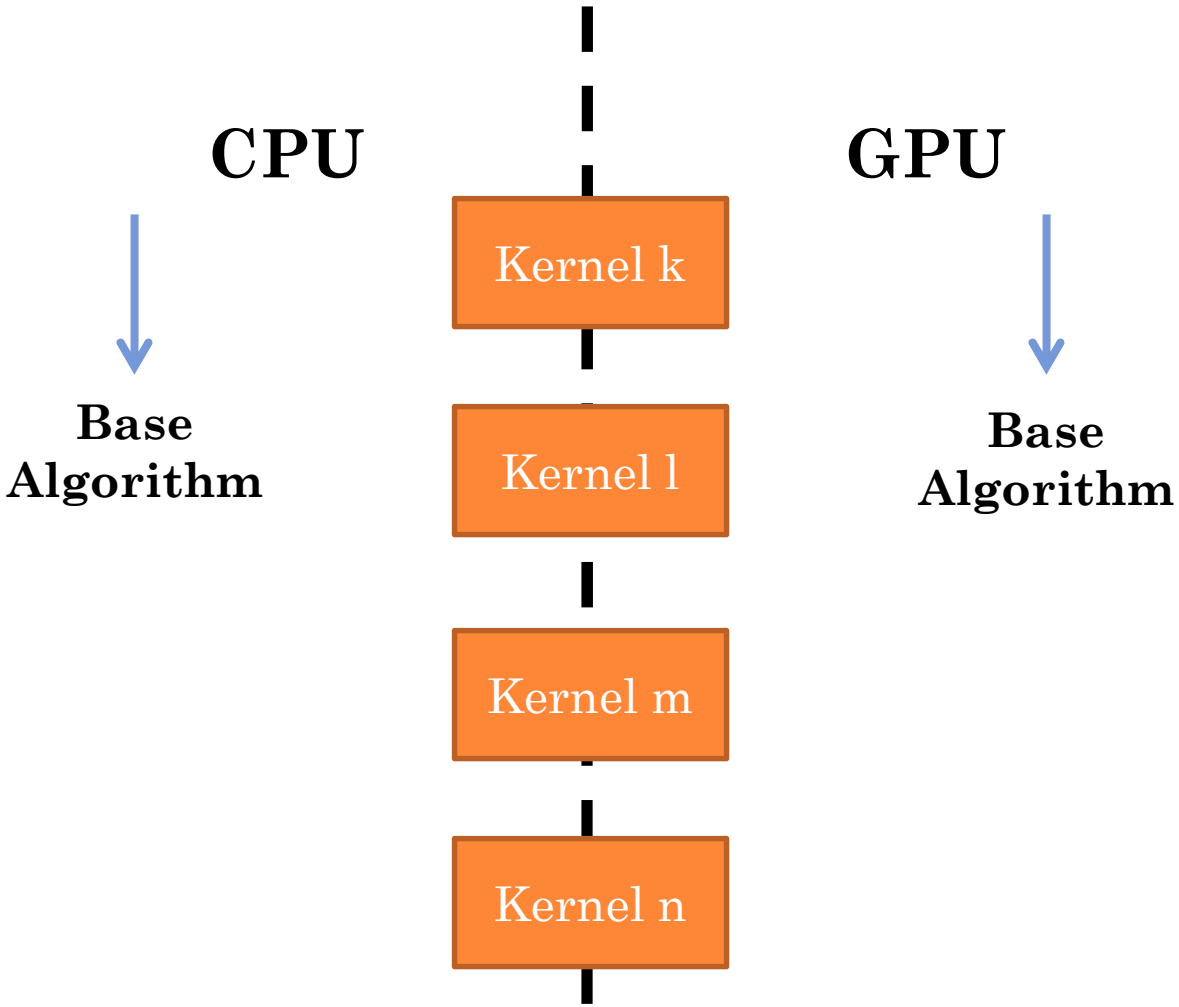
GreedyAlgorithm



Improved Algorithm

- For each kernel, force to explore both CPU and GPU
  - Run Base Algorithm by taking CPU path
  - Run Base Algorithm by taking GPU path
  - Compare the overall performances
  - Execute kernel k on the corresponding device

Improved Algorithm



Experimental Setup

- System

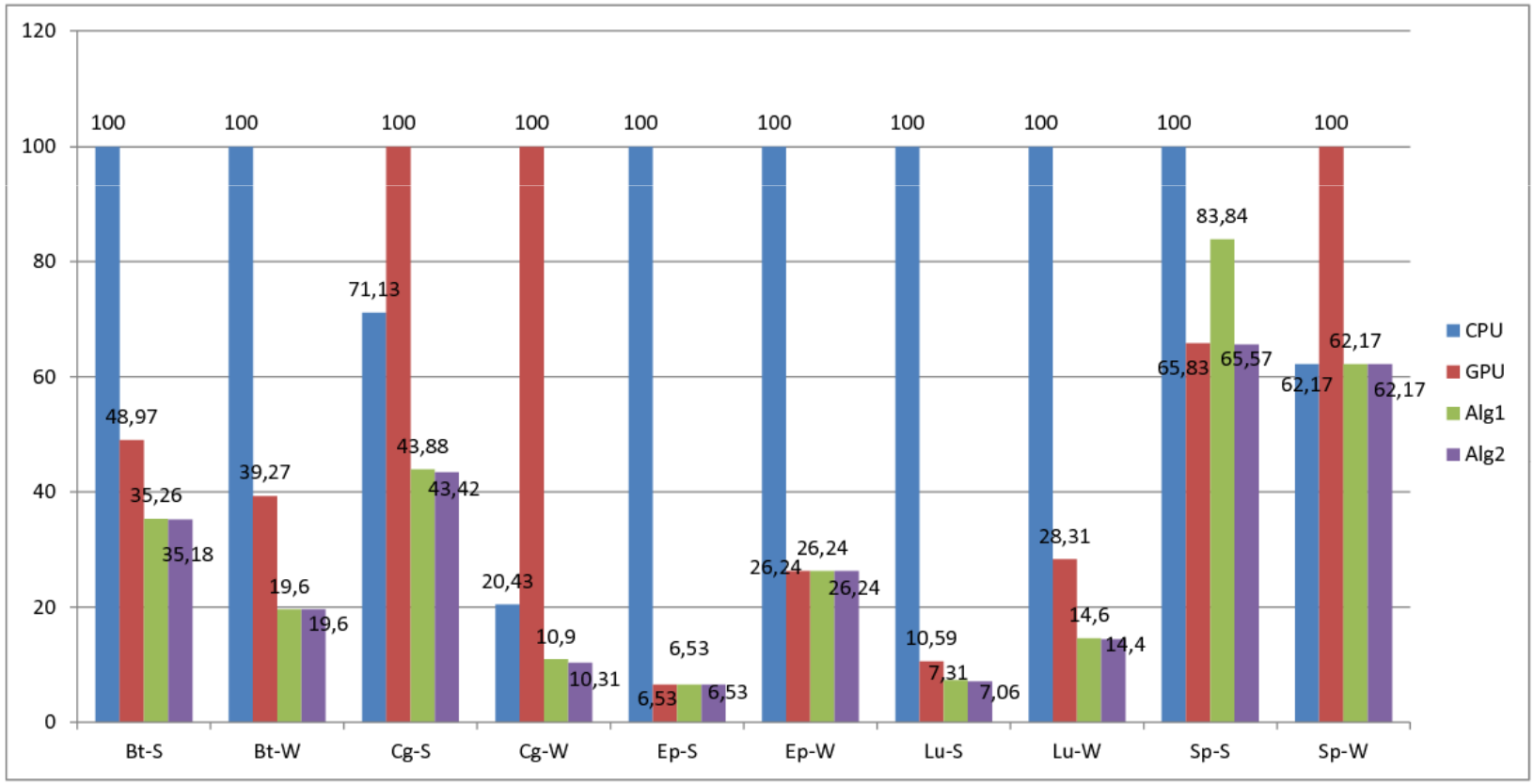
CPU	GPU
AMD Phenom II X6 1055T	GeForce GTX 460
AMD APP SDK v2.6	NVIDIA OpenCL SDK 4.0
4GB	1GB

- NAS Benchmarks [1]

Bench.	Description
BT	Solves non-diagonally dominant, block-tridiagonal equations.
CG	Computes an approximation for sparse, symmetrically positive definite matrix using a conjugate gradient method.
EP	Evaluates an integral by means of pseudo random trials.
LU	A regular-sparse, lower/upper triangular system solution.
SP	Solves non-diagonally dominant, pentadiagonal equations.

Experimental Results

- Except for EP-S, EP-W, and SP-W, all benchmarks use both CPU and GPU resources



Related Work

- Profiling
  - Luk et al.[2]
    Qilin: predict kernel execution times offline
  - Grewe et al.[3]
    A static machine learning-based task mapping algorithm
- Mapping
  - Augonnet et al.[4]
    Starpu: Runtime system tool that provides an interface unifying execution on accelerator technologies as well as multicore processors.

Future Work

- Memory controller is not considered in current implementation
- Check for independent kernels and allocate them onto devices in a parallel fashion
- Use multiple GPUs

Conclusion

- Effective kernel mapping for heterogeneous systems
  - Using CPUs and GPUs
  - Heuristic
  - ILP
- Our algorithm effectively assigns the kernels to the devices

REFERENCES

- [1]"NASA, NAS parallel benchmarks http://www.nas.nasa.gov/publications/npb.html
- [2] C.-K. Luk, S. Hong, and H. Kim, "Qilin: exploiting parallelism on heterogeneous multiprocessors with adaptive mapping,"
- [3] D. Grewe and M. F. P. O'Boyle, "A static task partitioning approach for heterogeneous systems using opencl,"
- [4] C. Augonnet, S. Thibault, R. Namyst, and P.-A. Wacrenier, "Starpu: a unified platform for task scheduling on heterogeneous multicore architectures,"

Thanks!