Programming Multiple GPUs

Youngsok Kim Jaewon Lee Jangwoo Kim
High Performance Computing Lab., POSTECH

Motivation

- How to reduce the burden of multi-GPU programming?
- How to guarantee functional correctness on multi-GPU?
- How to achieve the optimal performance on multi-GPU?

Summary

- We analyze the burden of multi-GPU programming.
- We are working on static code analysis & architectural supports to achieve functional correctness & optimal performance.

Background

- **Higher throughput with multiple GPUs**

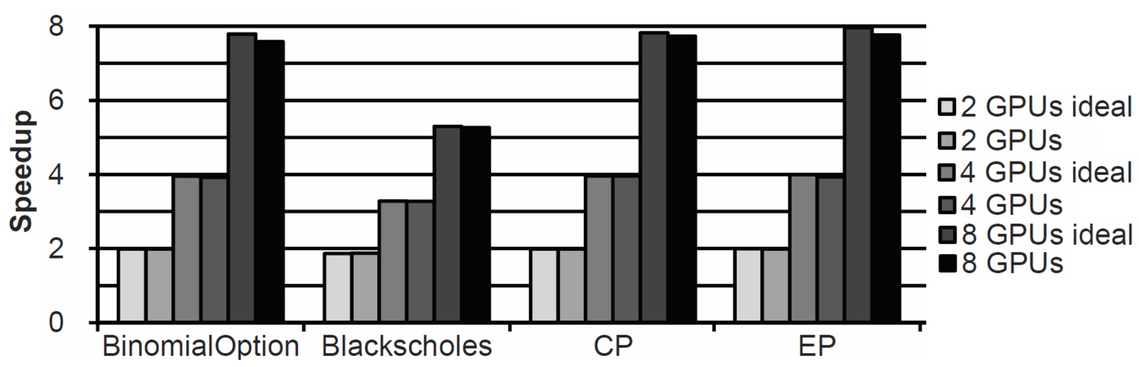


Figure 1. Speedup versus a single GPU [1]

- **Non-optimal performance with multi-GPU**

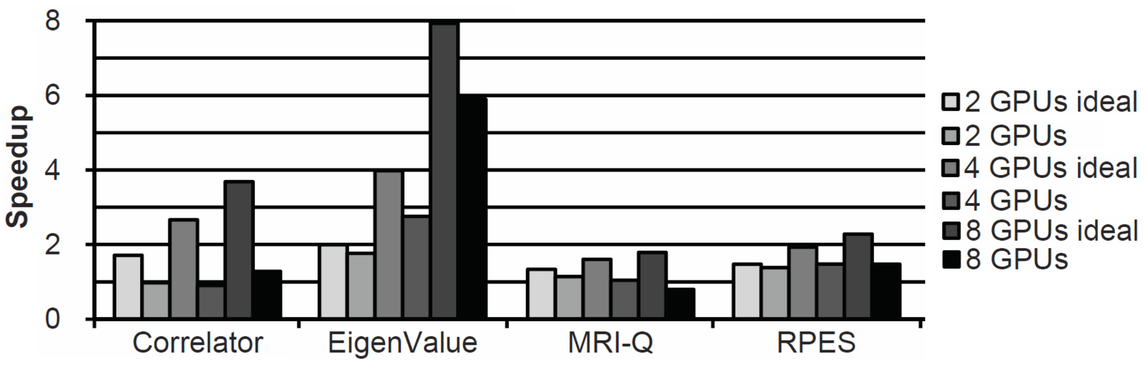


Figure 2. Speedup versus a single GPU [1]

Not all applications take advantage of multiple GPUs!

- **Burden of multi-GPU programming**

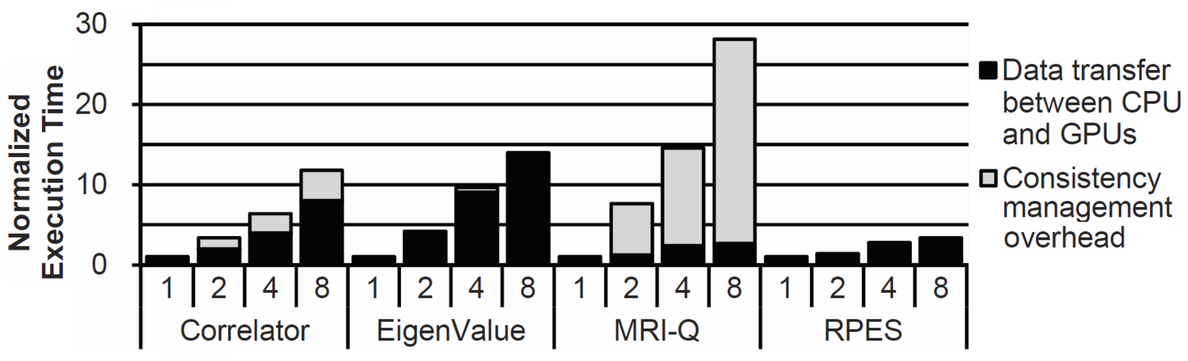


Figure 3. Execution time analysis [1]

> Keeping data consistency across GPUs degrades performance.
> Data and thread assignment cannot be changed during program execution.
> Lack of atomic operations forces programmers to modify their codes.

Problem:
> Programmers must redesign their algorithms and rewrite their codes for multi-GPUs.
> Programmers lose performance when porting programs to multiple GPUs.

Experimental Results

- **Existing multi-CPU programming model**

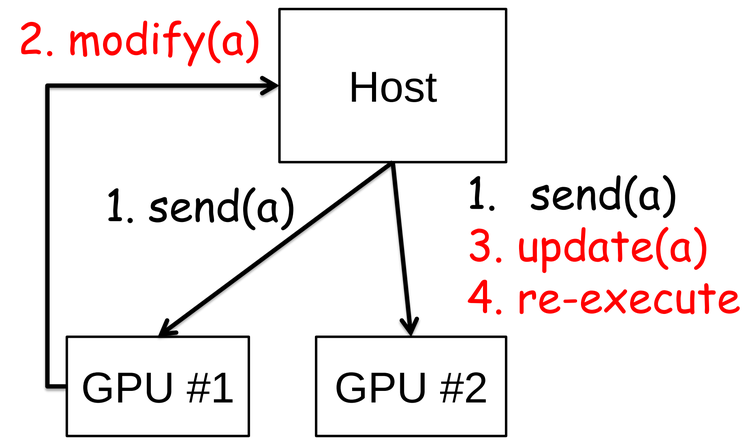
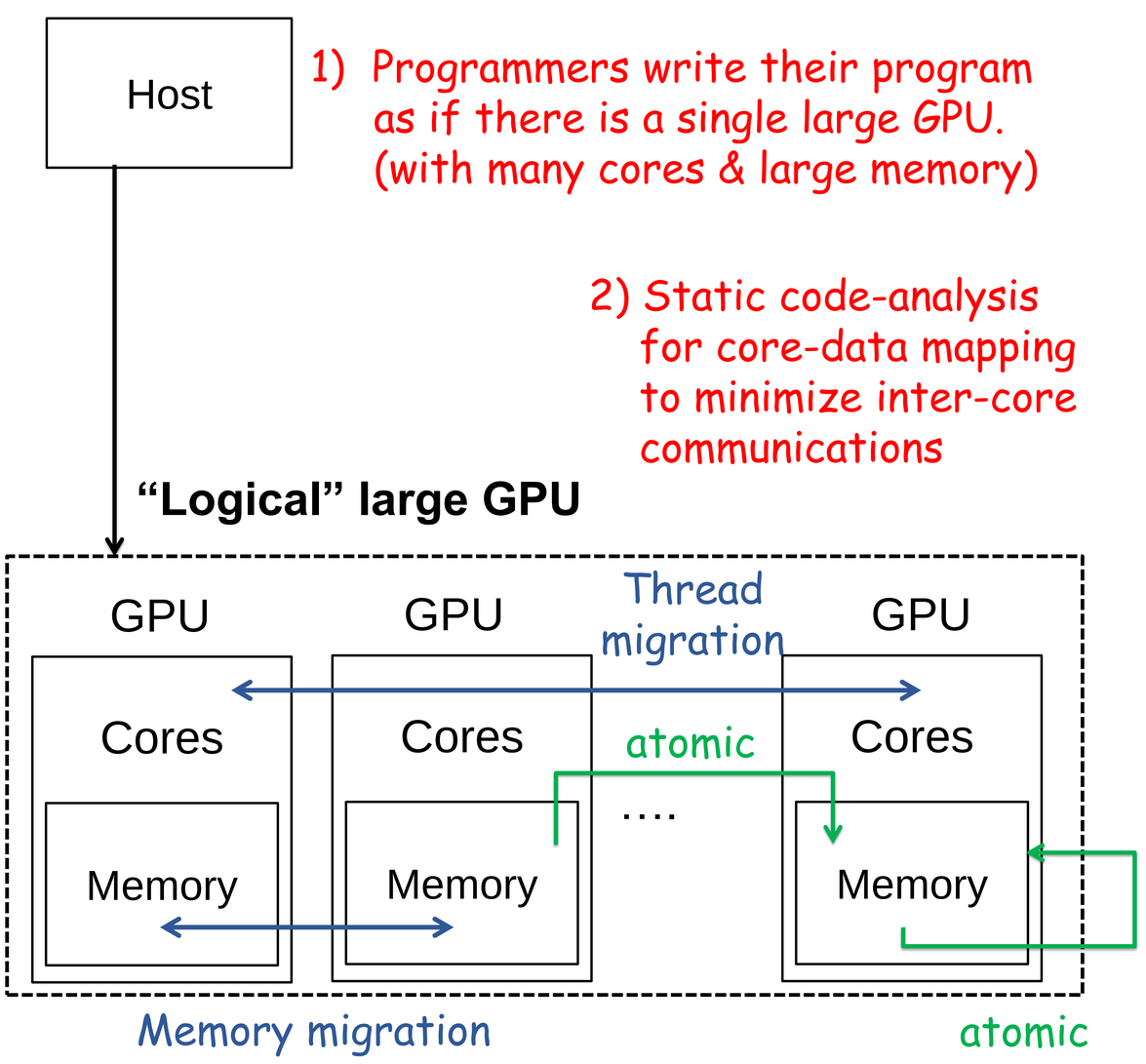


Figure 4. Inter-GPU communications through a host

- **New programming model**



3) Locality-aware run-time thread & memory migration for optimal performance

4) Guarantee atomic operations from cores to be "atomic"

Current Work

We are currently implementing static code analysis (front-end data and thread assignment) and architectural supports (inter-GPU synchronization, atomics, data replication) to guarantee functional correctness and optimal performance based on OpenCL [2].

References

[1] Jungwon Kim, Honggyu Kim, Joo Hwan Lee, and Jaejin Lee, "Achieving a Single Compute Device Image in OpenCL for Multiple GPUs," In *PPoPP'11*.
[2] Khronos Group, "*OpenCL: The Open Standard for Parallel Programming of Heterogeneous Systems,*" http://www.khronos.org/opencl/