Chai: Collaborative Heterogeneous Applications for Integrated-architectures

CHAI

Juan Gómez-Luna[1], Izzat El Hajj[2], Li-Wen Chang[2], Víctor García-Flores[3,4], Simon Garcia de Gonzalo[2], Thomas B. Jablin[2,5], Antonio J. Peña[4], and Wen-mei Hwu[2]

[1]Universidad de Córdoba, [2]University of Illinois at Urbana-Champaign, [3]Universitat Politècnica de Catalunya, [4]Barcelona Supercomputing Center, [5]MulticoreWare, Inc.

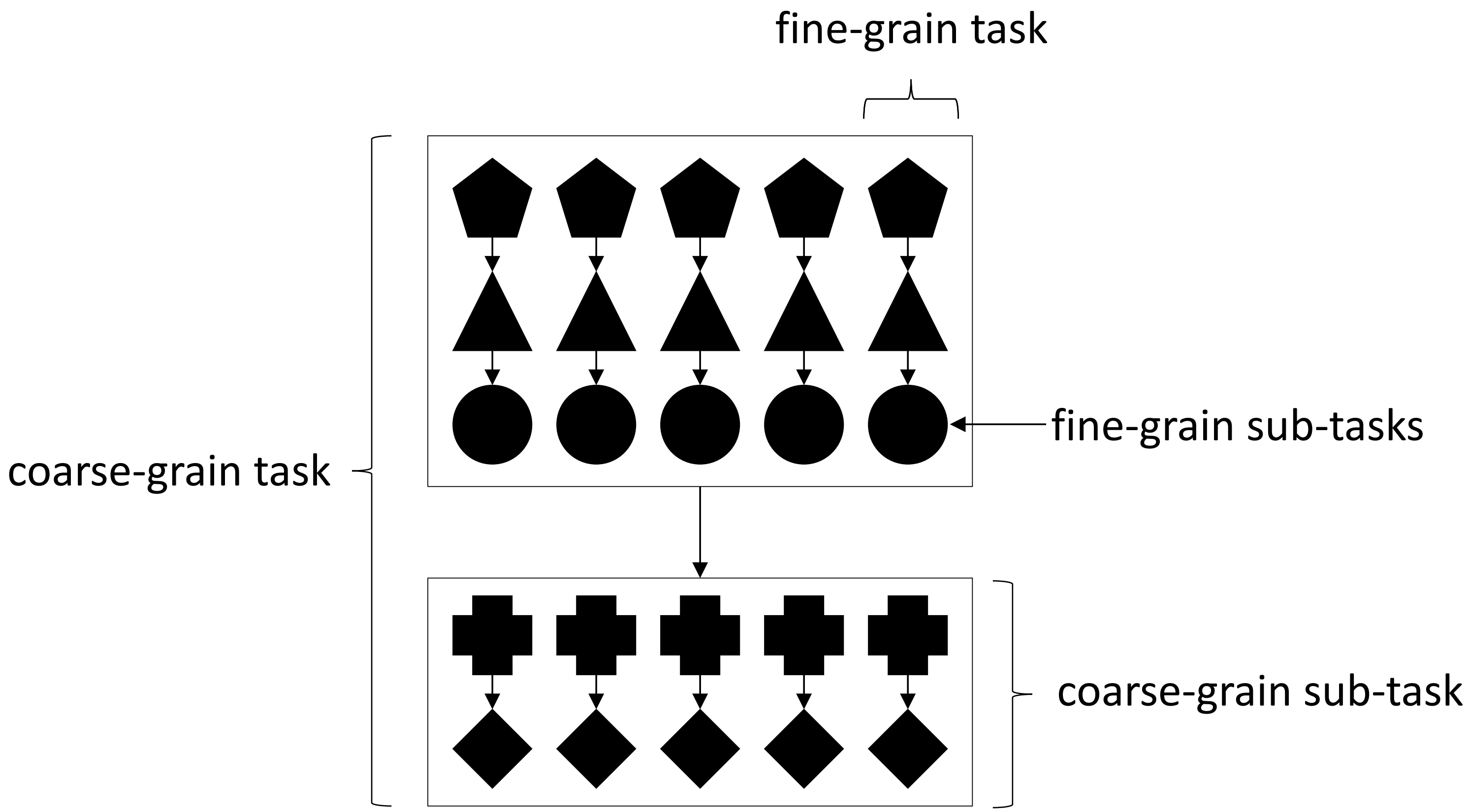
C-FAR
Hewlett Packard Enterprise
HUAWEI
JUNTA DE ANDALUCIA
MINISTERIO DE EDUCACIÓN
NVIDIA
Center for Exascale Simulation XPACC of Plasma-Coupled Combustion
UNIVERSIDAD D CÓRDOBA
ILLINOIS

Motivation

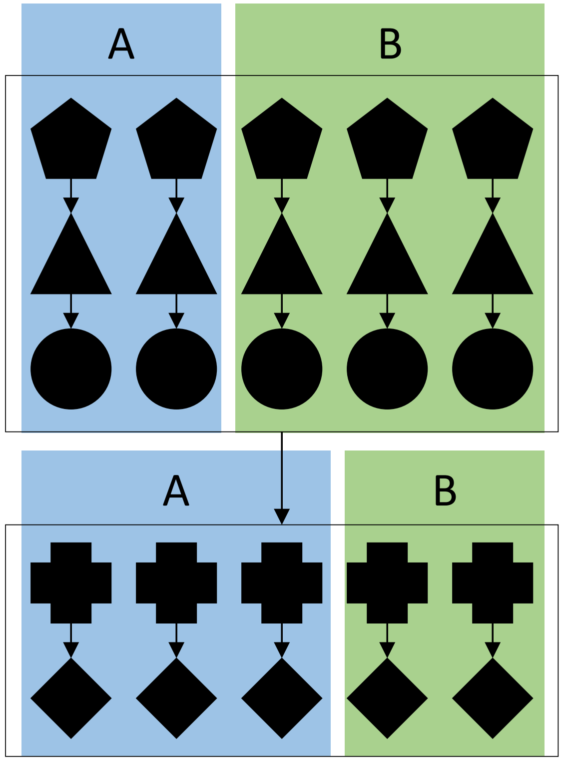
- Heterogeneous systems are moving towards tighter integration
  - Shared virtual memory, coherence, system-wide atomics
  - OpenCL 2.0, CUDA 8.0
- Benchmark suite is needed
  - Analyzing collaborative workloads
  - Evaluating new architecture features

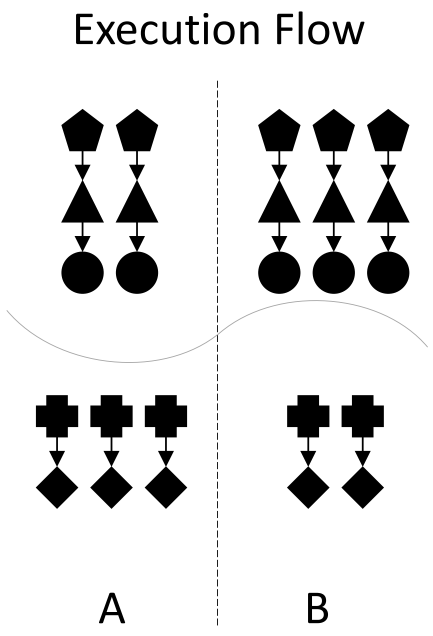
CHAI

Application Structure



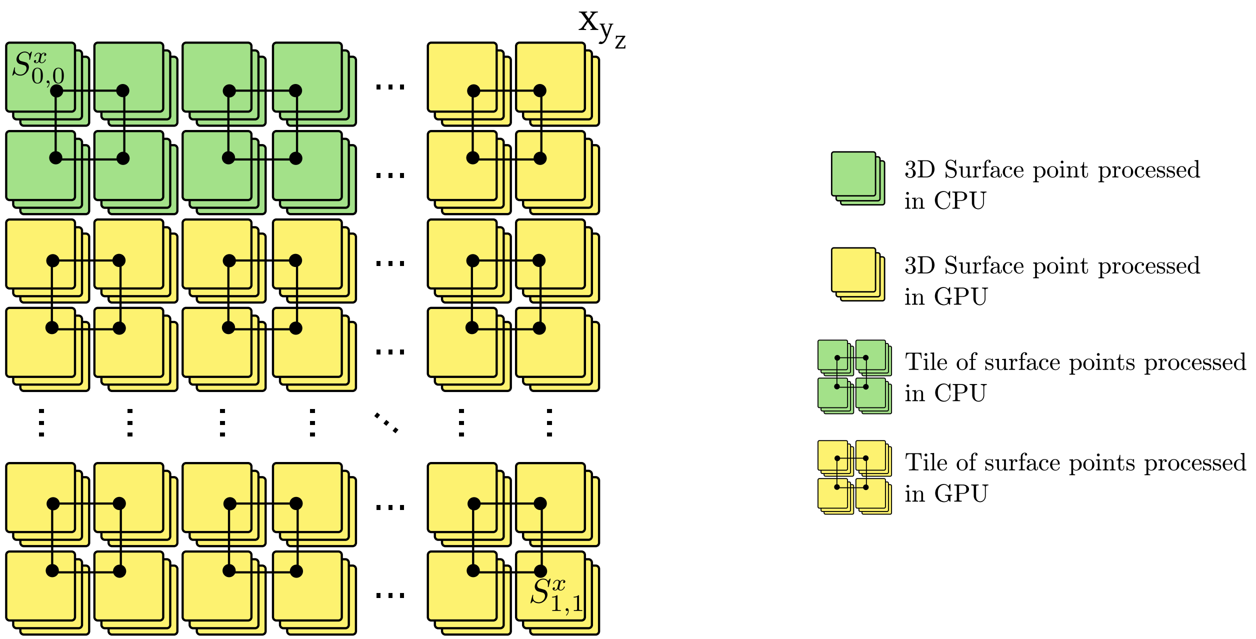
Data Partitioning





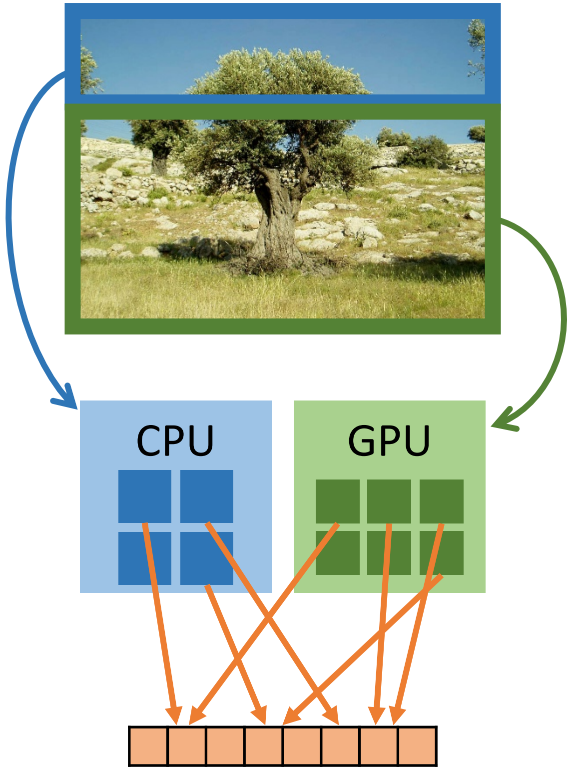
Data Partitioning: Bézier Surfaces

- Output surface points are distributed across devices

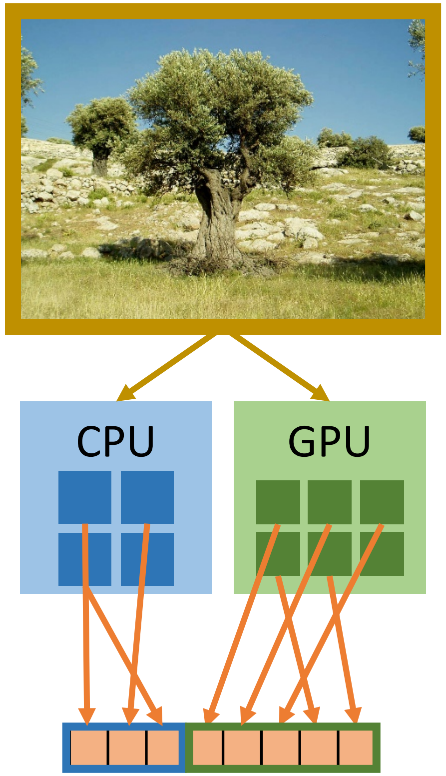


Data Partitioning: Image Histogram

Input pixels distributed across devices

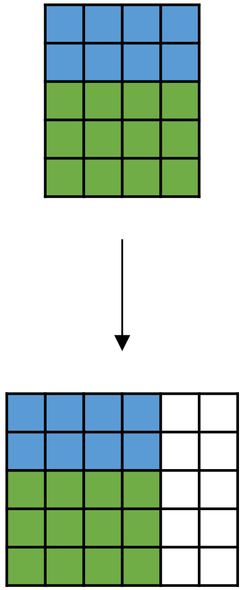


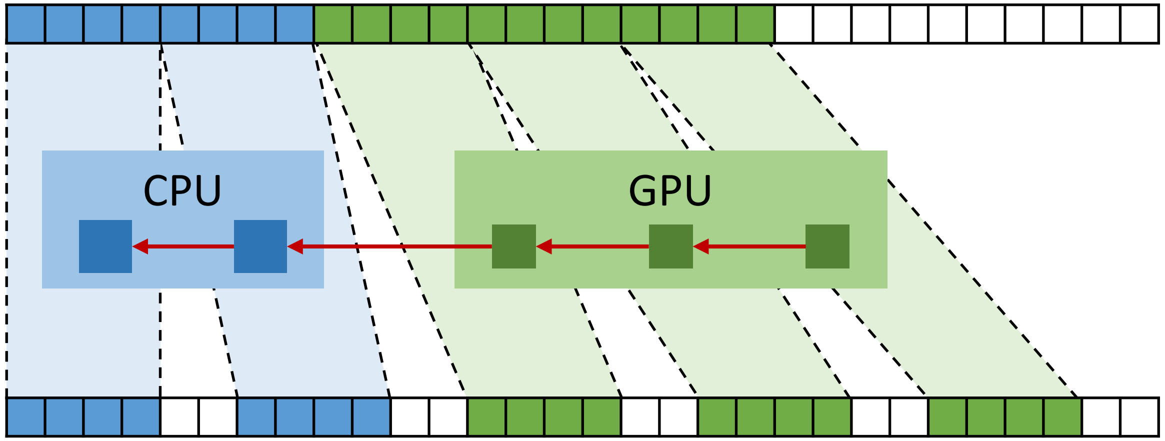
Output bins distributed across devices



Data Partitioning: Padding

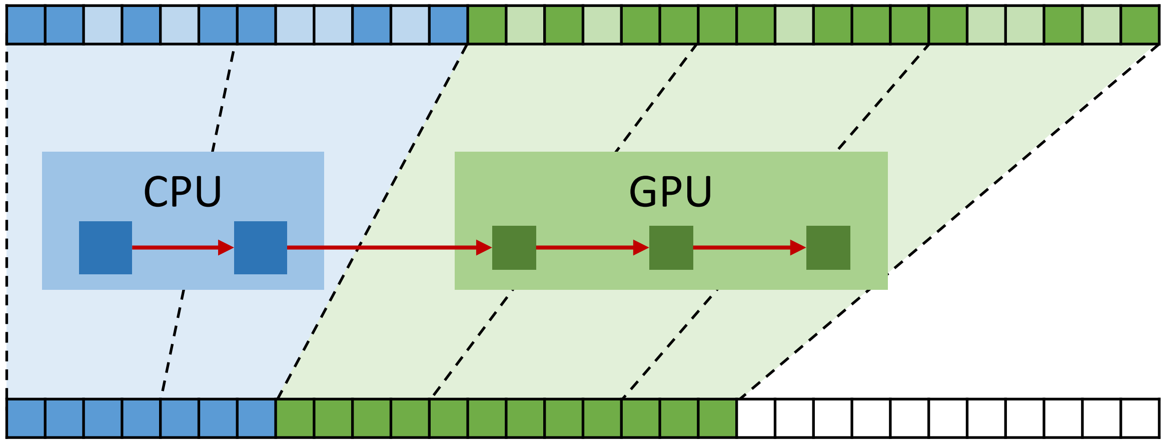
- Rows are distributed across devices
  - Challenge: in-place, required inter-worker synchronization





Data Partitioning: Stream Compaction

- Rows are distributed across devices
  - Like padding, but irregular and involves predicate computations



Data Partitioning: Other Benchmarks

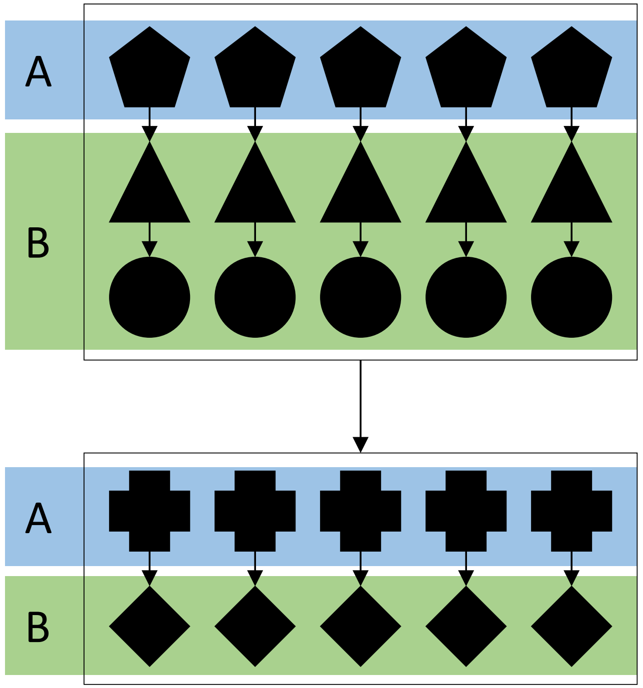
- Canny Edge Detection
  - Different devices process different images
- Random Sample Consensus
  - Workers on different devices process different models
- In-place Transposition
  - Workers on different devices follow different cycles

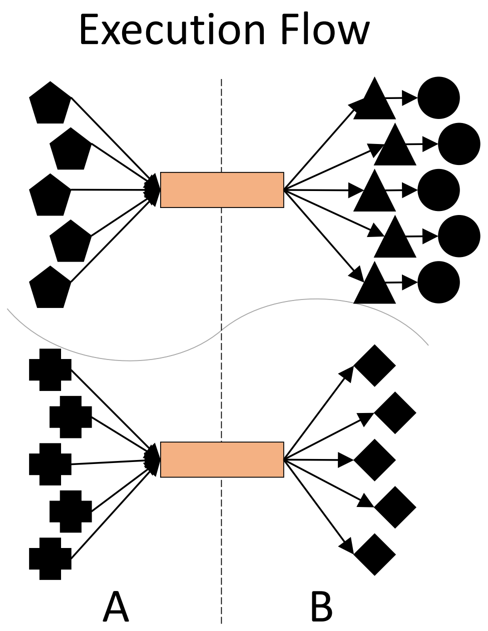
Types of data partitioning

- Partitioning strategy:
  - Static (fixed work for each device)
  - Dynamic (contend on shared worklist)
  - Flexible interface for defining partitioning schemes

- Partitioned data:
  - Input (e.g., Image Histogram)
  - Output (e.g., Bézier Surfaces)
  - Both (e.g., Padding)

Fine-grain Task Partitioning

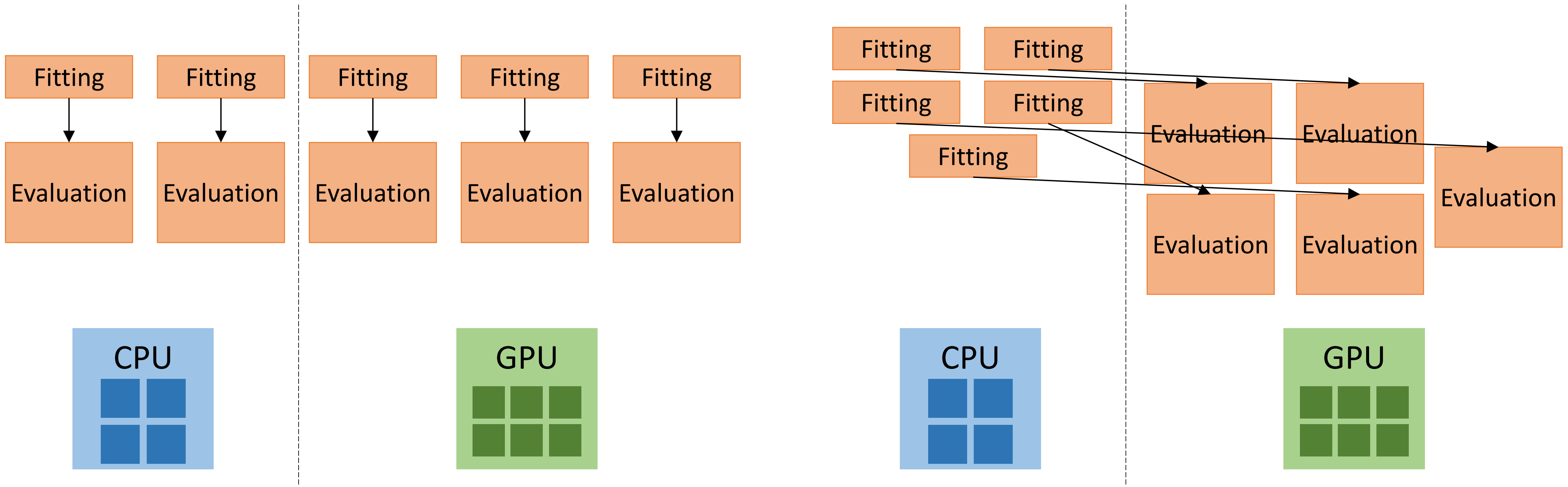




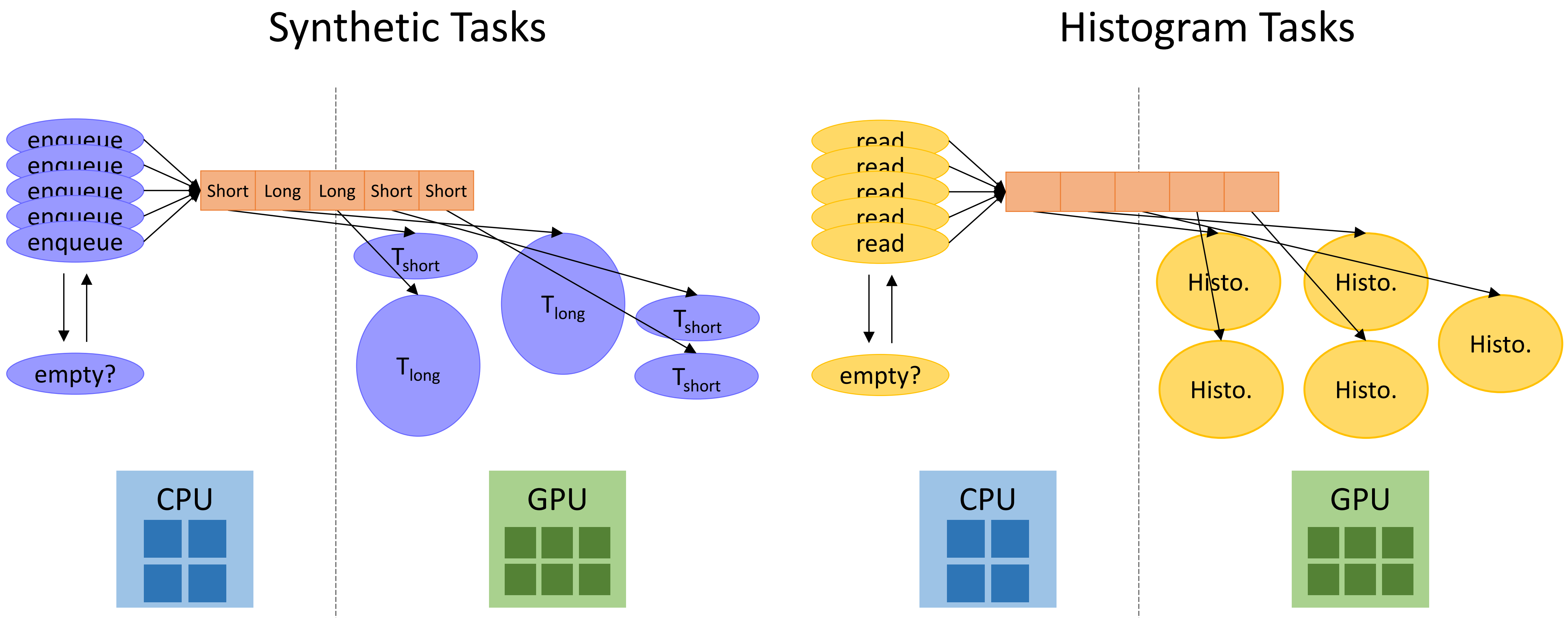
Fine-grain Task Partitioning: Random Sample Consensus

Data partitioning: models distributed across devices

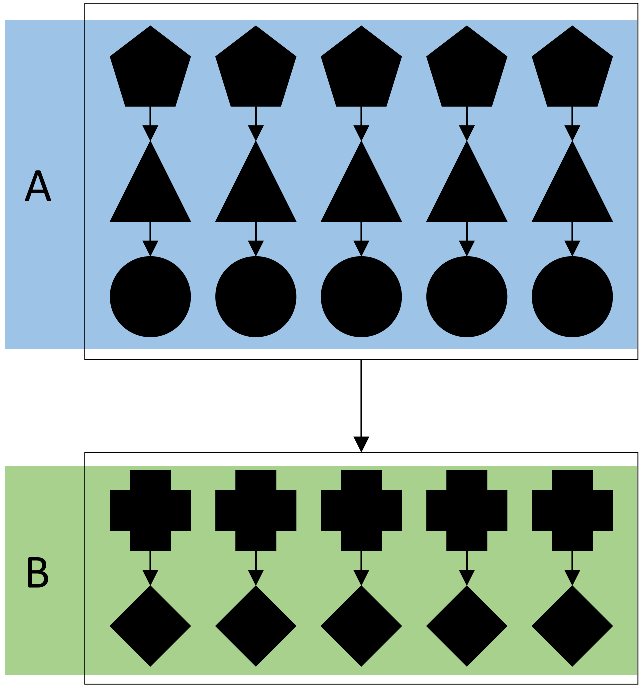
Task partitioning: model fitting on CPU and evaluation on GPU

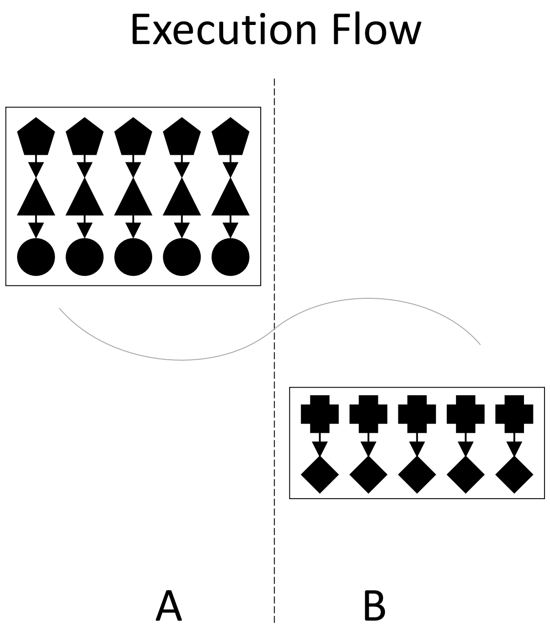


Fine-grain Task Partitioning: Task Queue System

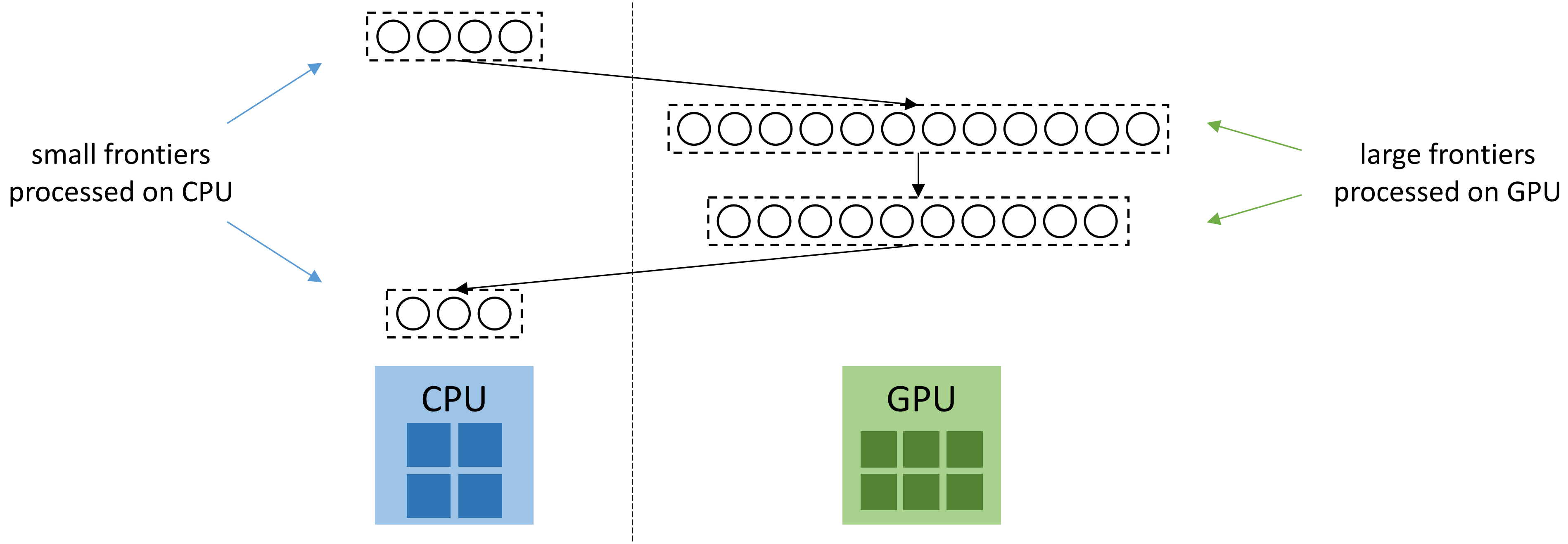


Coarse-grain Task Partitioning



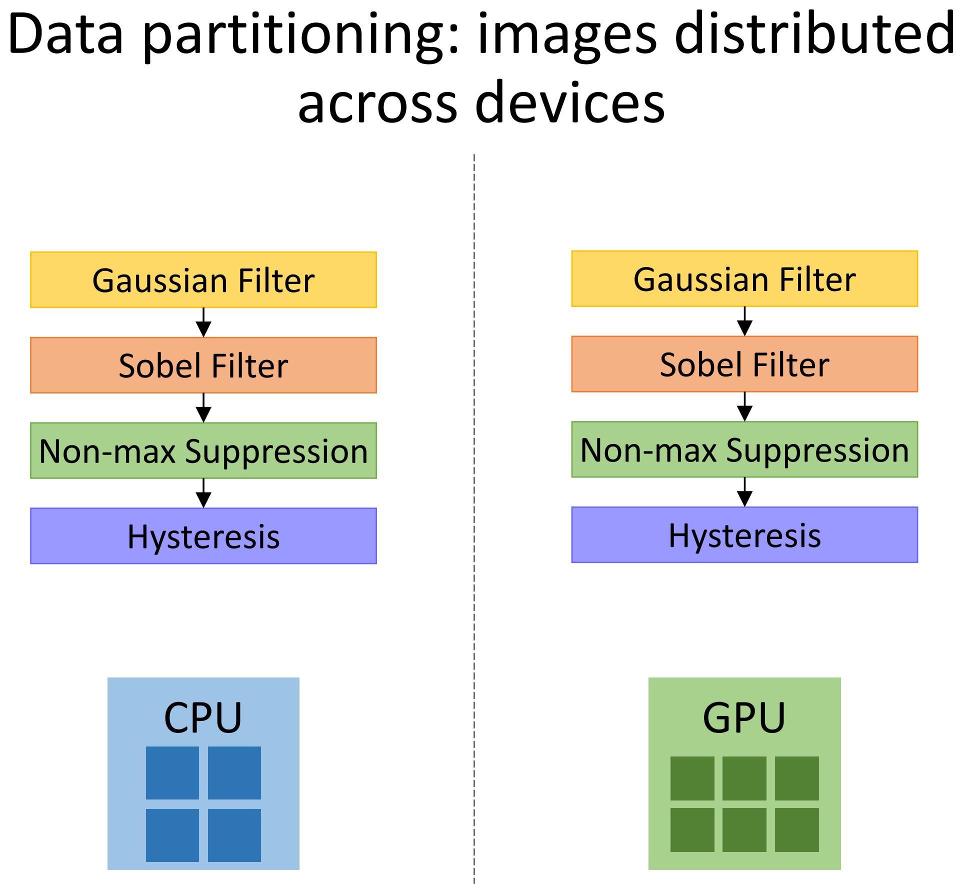


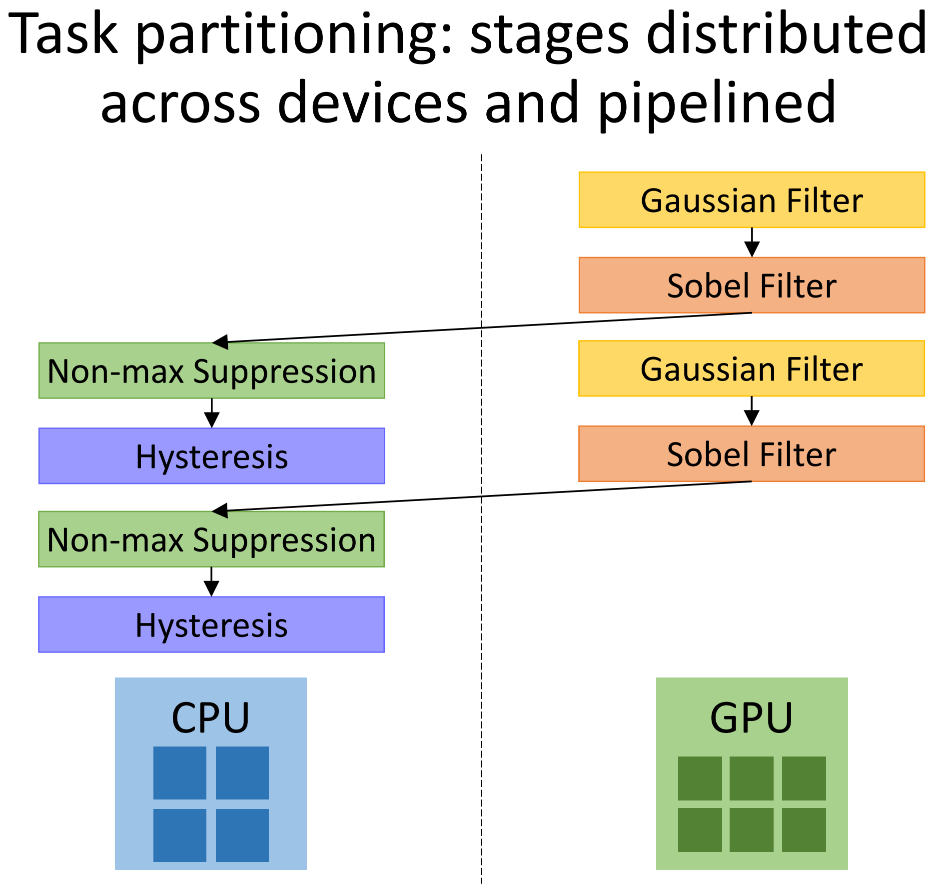
Coarse-grain Task Partitioning: Breadth First Search & Single Source Shortest Path



SSSP performs more computations than BFS which hides communication/memory latency

Coarse-grain Task Partitioning: Canny Edge Detection





Benchmarks and Implementations

Collaboration Pattern		Short Name	Benchmark
Data Partitioning		BS	Bézier Surface
		CEDD	Canny Edge Detection
		HSTI	Image Histogram (Input Partitioning)
		HSTO	Image Histogram (Output Partitioning)
		PAD	Padding
		RSCD	Random Sample Consensus
		SC	Stream Compaction
		TRNS	In-place Transposition
Task Partitioning	Fine-grain	RSCT	Random Sample Consensus
		TQ	Task Queue System (Synthetic)
		TQH	Task Queue System (Histogram)
	Coarse-grain	BFS	Breadth-First Search
		CEDT	Canny Edge Detection
		SSSP	Single-Source Shortest Path

Implementations:

- OpenCL-U
- OpenCL-D
- CUDA-U
- CUDA-D
- CUDA-U-Sim
- CUDA-D-Sim
- C++AMP

Benchmark Diversity

Data Partitioning

Benchmark	Partitioning Granularity	Partitioned Data	System-wide Atomics	Load Balance
BS	Fine	Output	None	Yes
CEDD	Coarse	Input, Output	None	Yes
HSTI	Fine	Input	Compute	No
HSTO	Fine	Output	None	No
PAD	Fine	Input, Output	Sync	Yes
RSCD	Medium	Output	Compute	Yes
SC	Fine	Input, Output	Sync	No
TRNS	Medium	Input, Output	Sync	No

Fine-grain Task Partitioning

Benchmark	System-wide Atomics	Load Balance
RSCT	Sync, Compute	Yes
TQ	Sync	No
TQH	Sync	No

Coarse-grain Task Partitioning

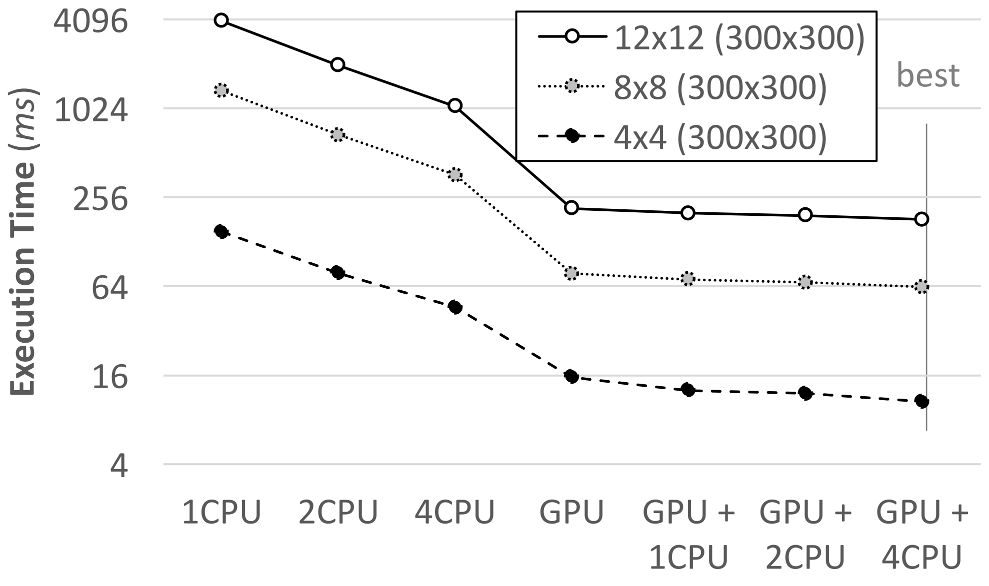
Benchmark	System-wide Atomics	Partitioning	Concurrency
BFS	Sync, Compute	Iterative	No
CEDT	Sync	Non-iterative	Yes
SSSP	Sync, Compute	Iterative	No

Evaluation Platform

- AMD Kaveri A10-7850K APU
  - 4 CPU cores
  - 8 GPU compute units
- AMD APP SDK 3.0
- Profiling:
  - CodeXL
  - gem5-gpu

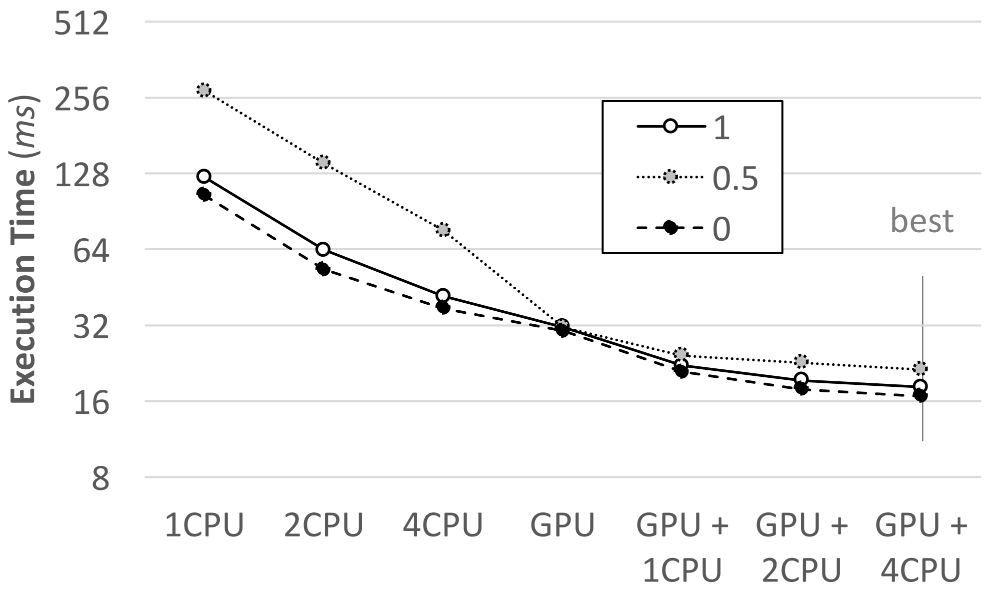
Benefits of Collaboration

- Collaborative execution improves performance



Bézier Surfaces

(up to 47% improvement over GPU only)

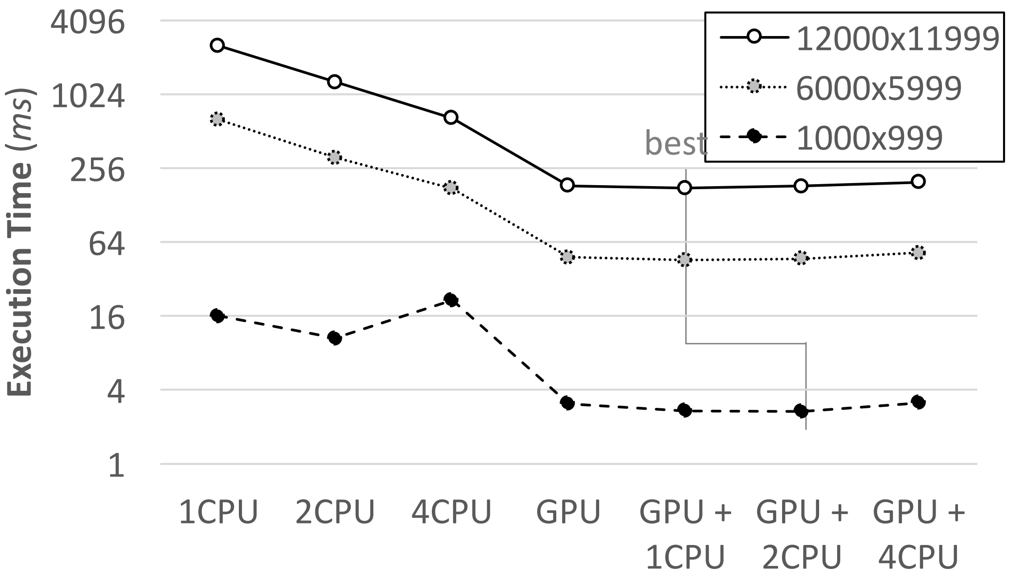


Stream Compaction

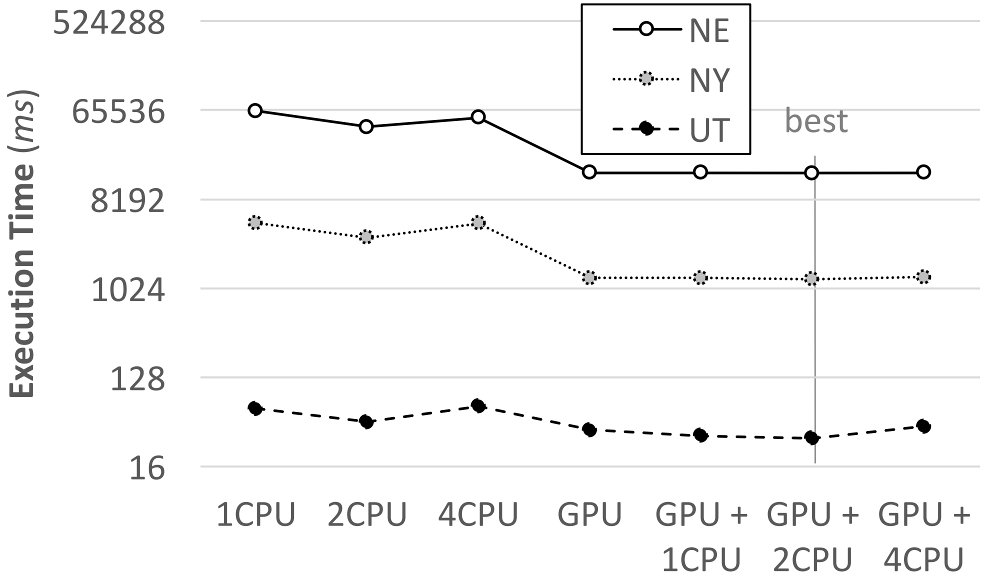
(up to 82% improvement over GPU only)

Benefits of Collaboration

- Optimal number of devices not always max and varies across datasets

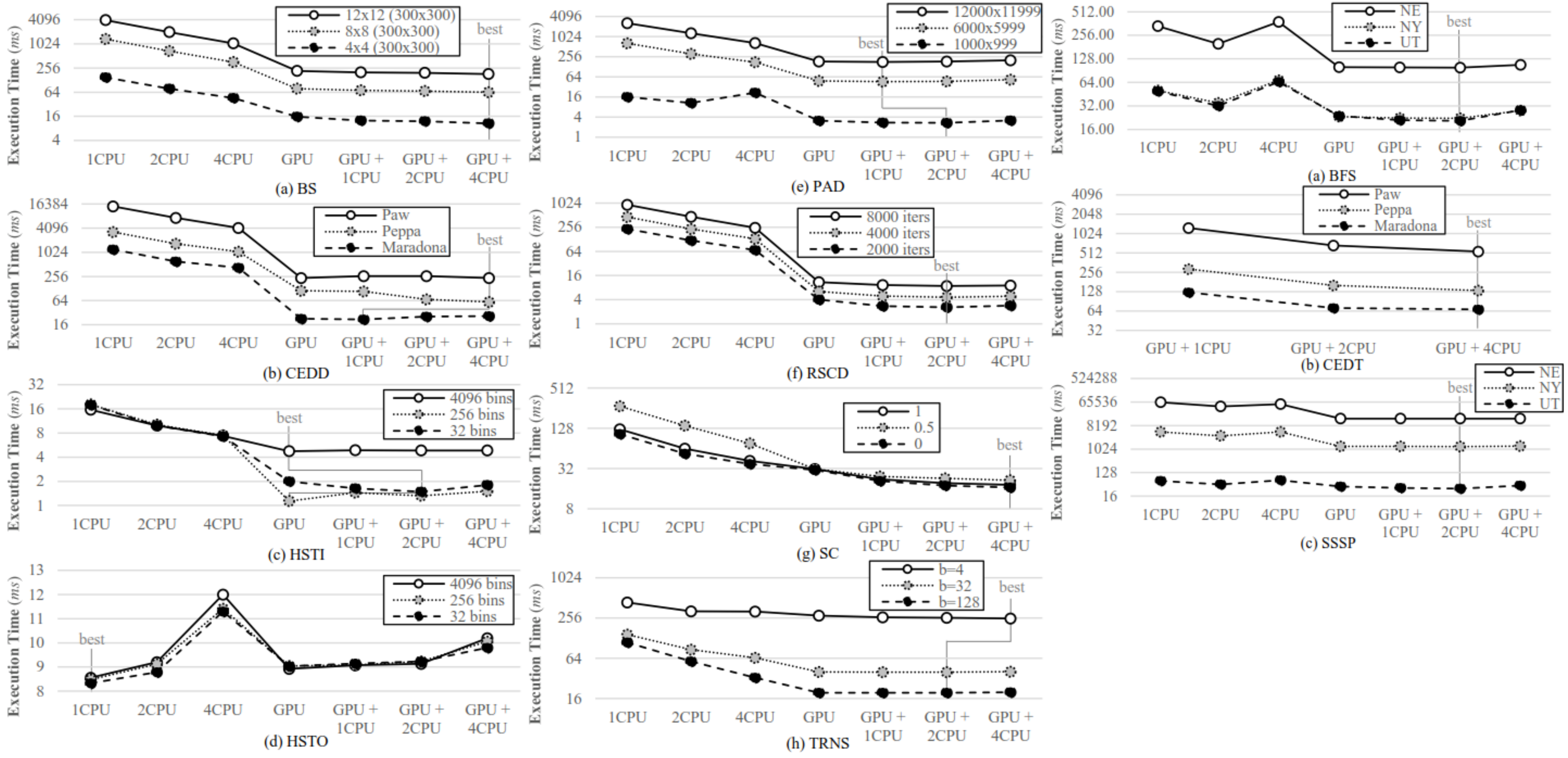


Padding
(up to 16% improvement over GPU only)

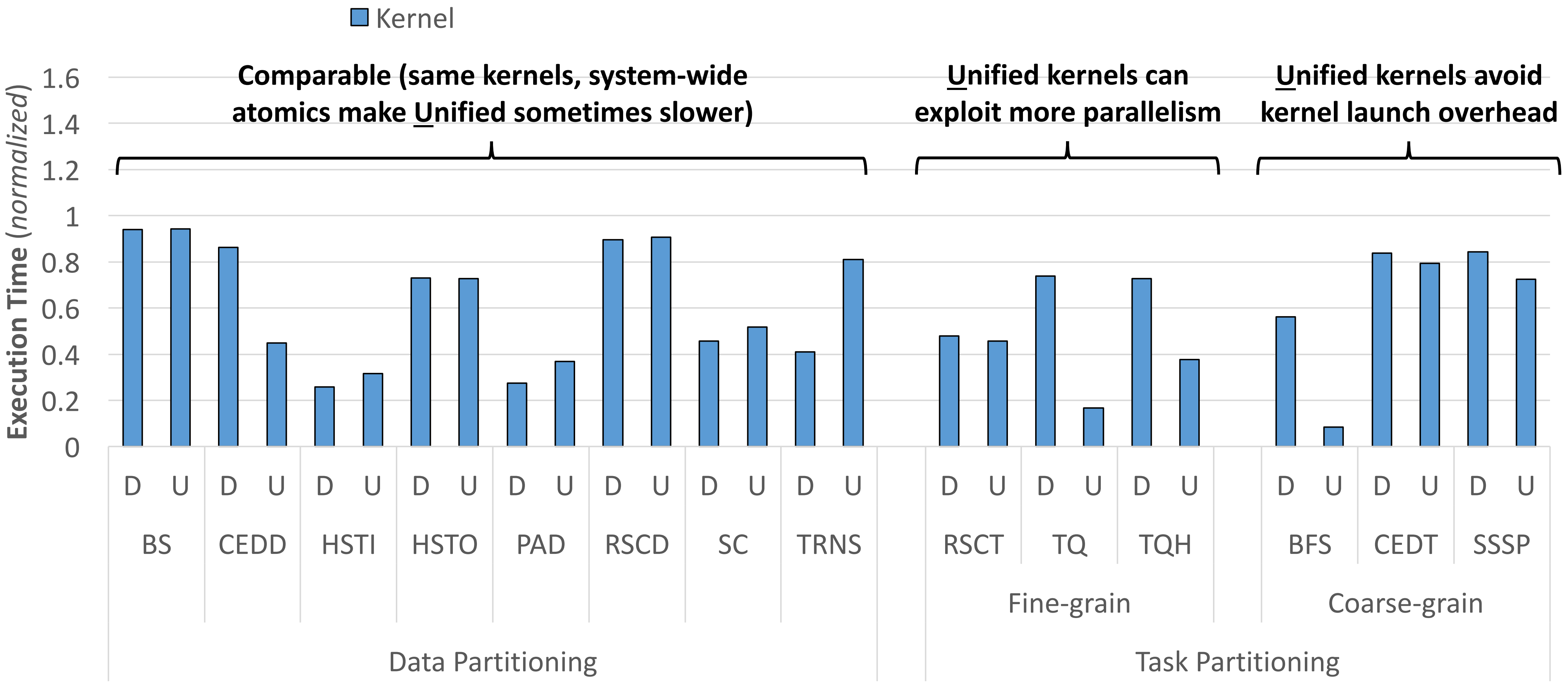


Single Source Shortest Path
(up to 22% improvement over GPU only)

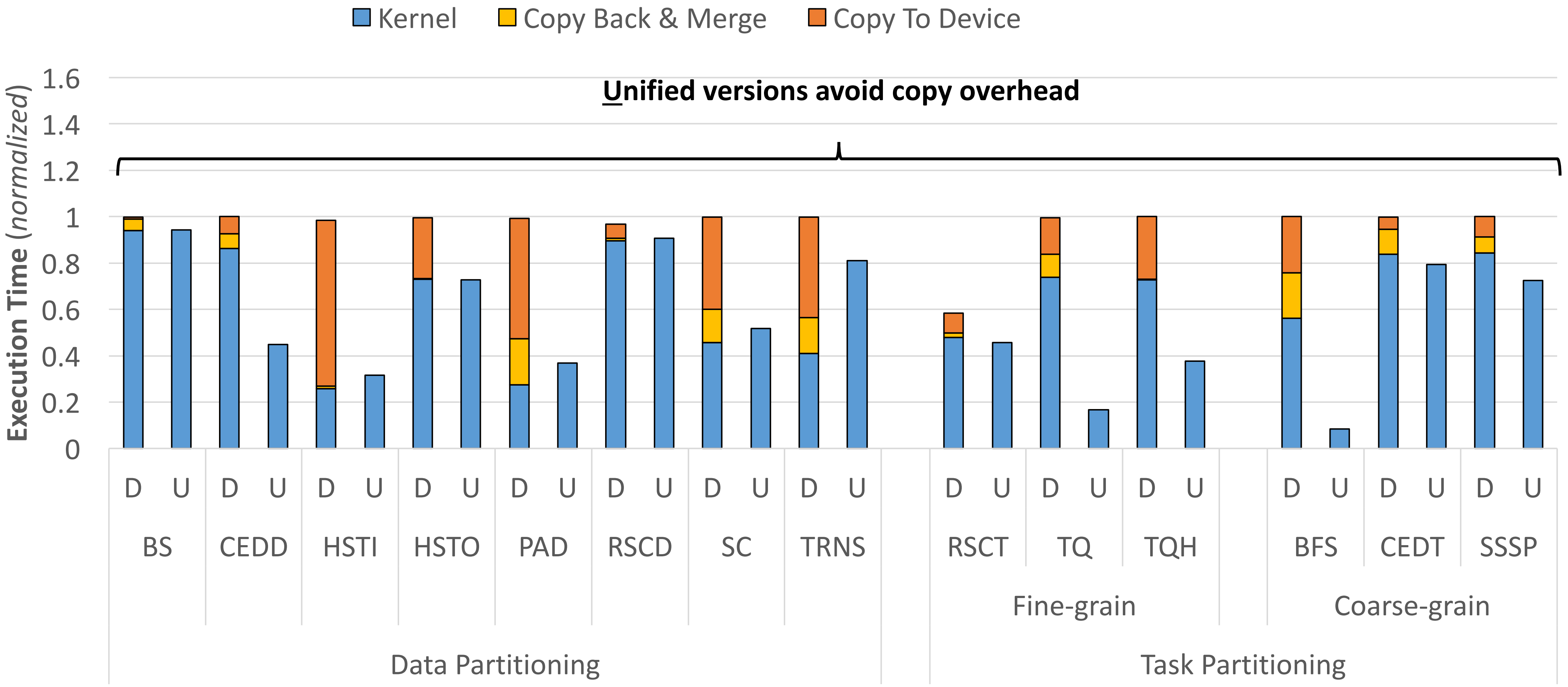
Benefits of Collaboration



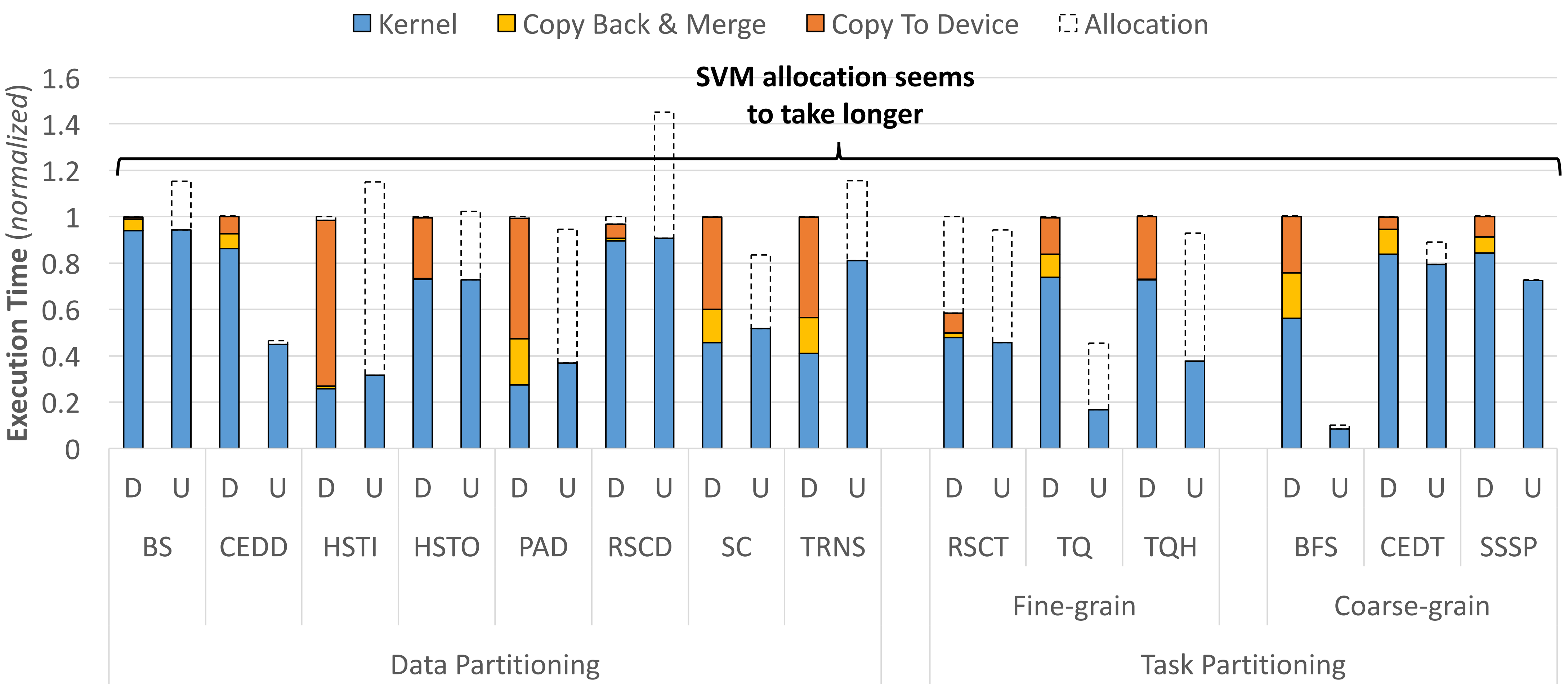
Benefits of Unified Memory



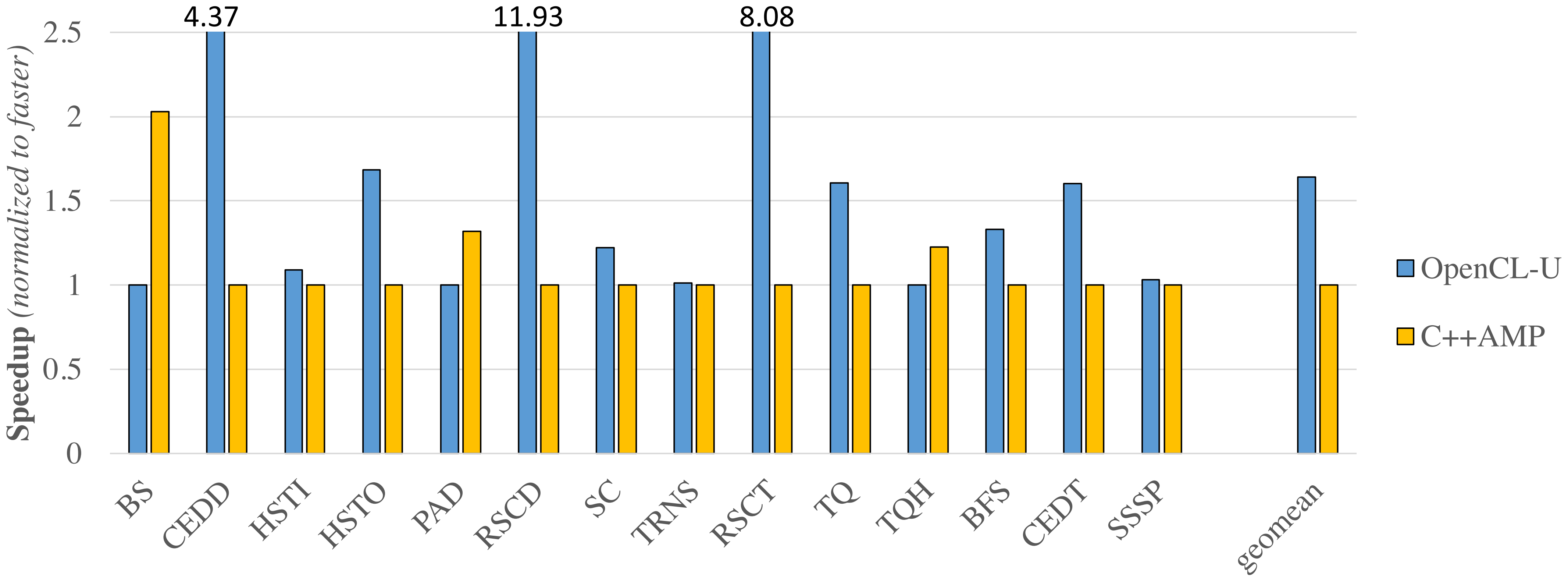
Benefits of Unified Memory



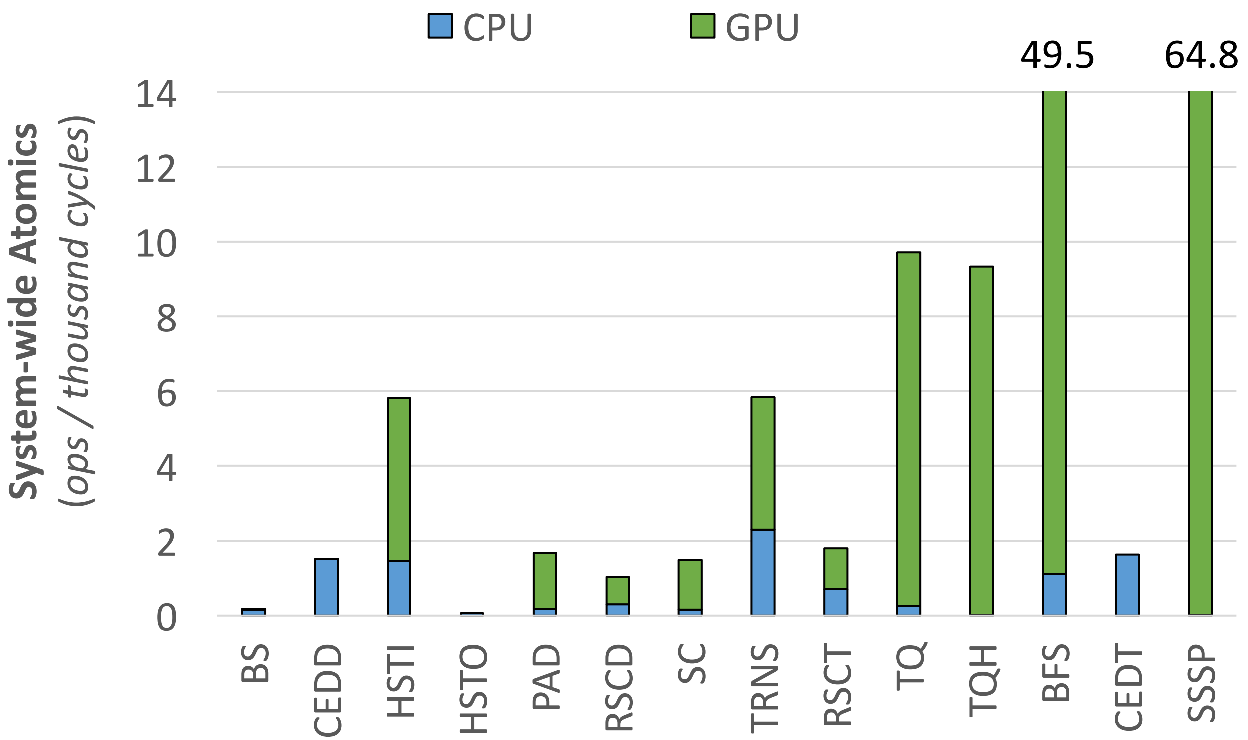
Benefits of Unified Memory



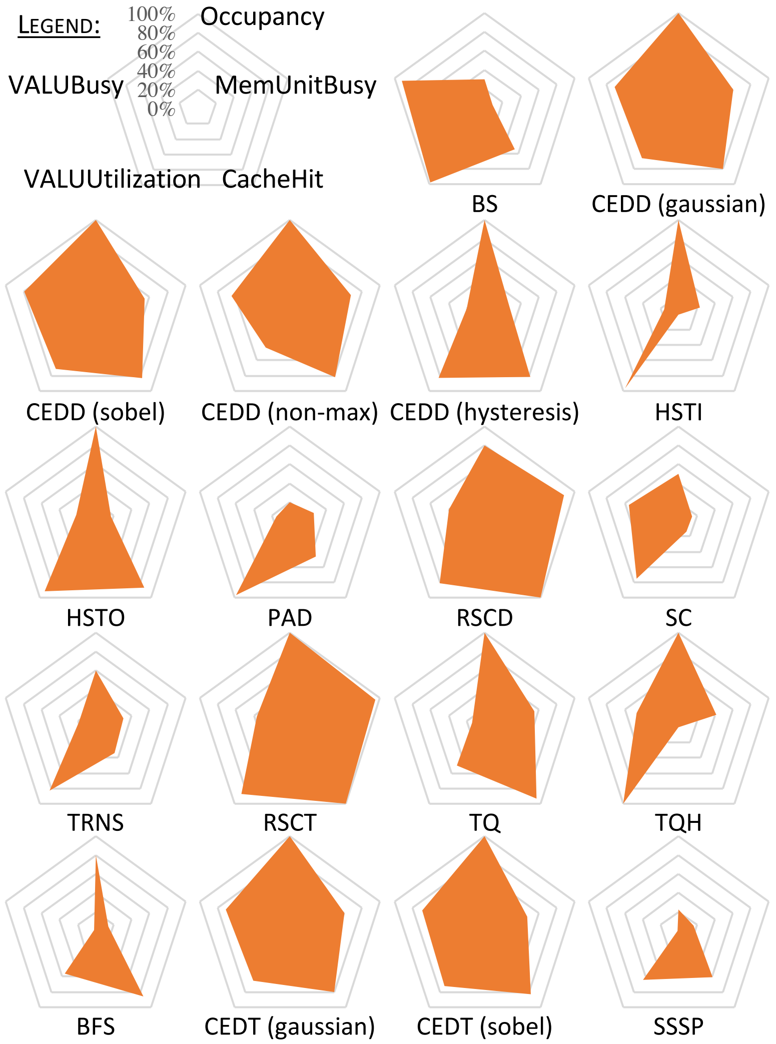
C++ AMP Performance Results



Benchmark Diversity



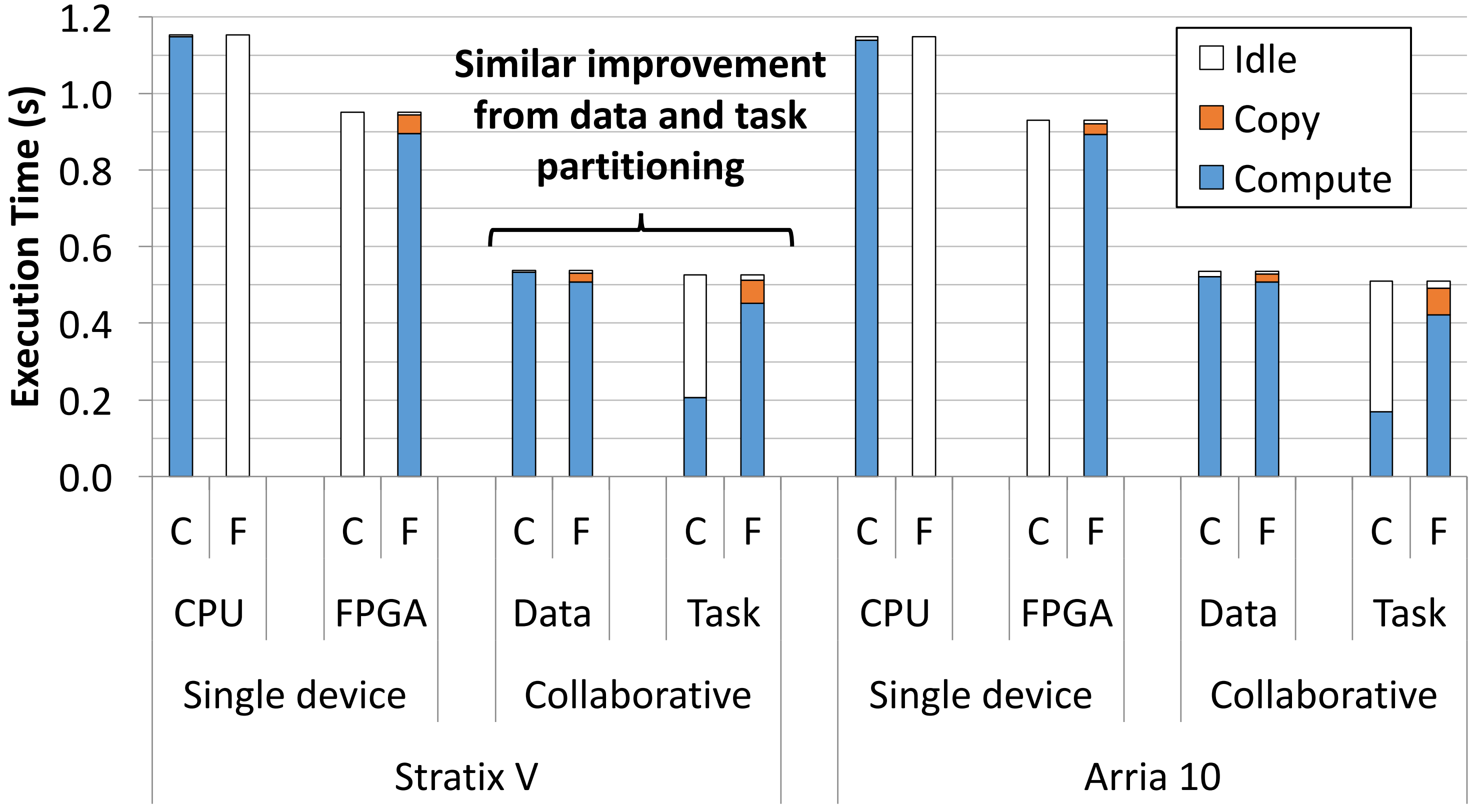
Varying intensity in use of system-wide atomics



Diverse execution profiles

Benefits of Collaboration on FPGA

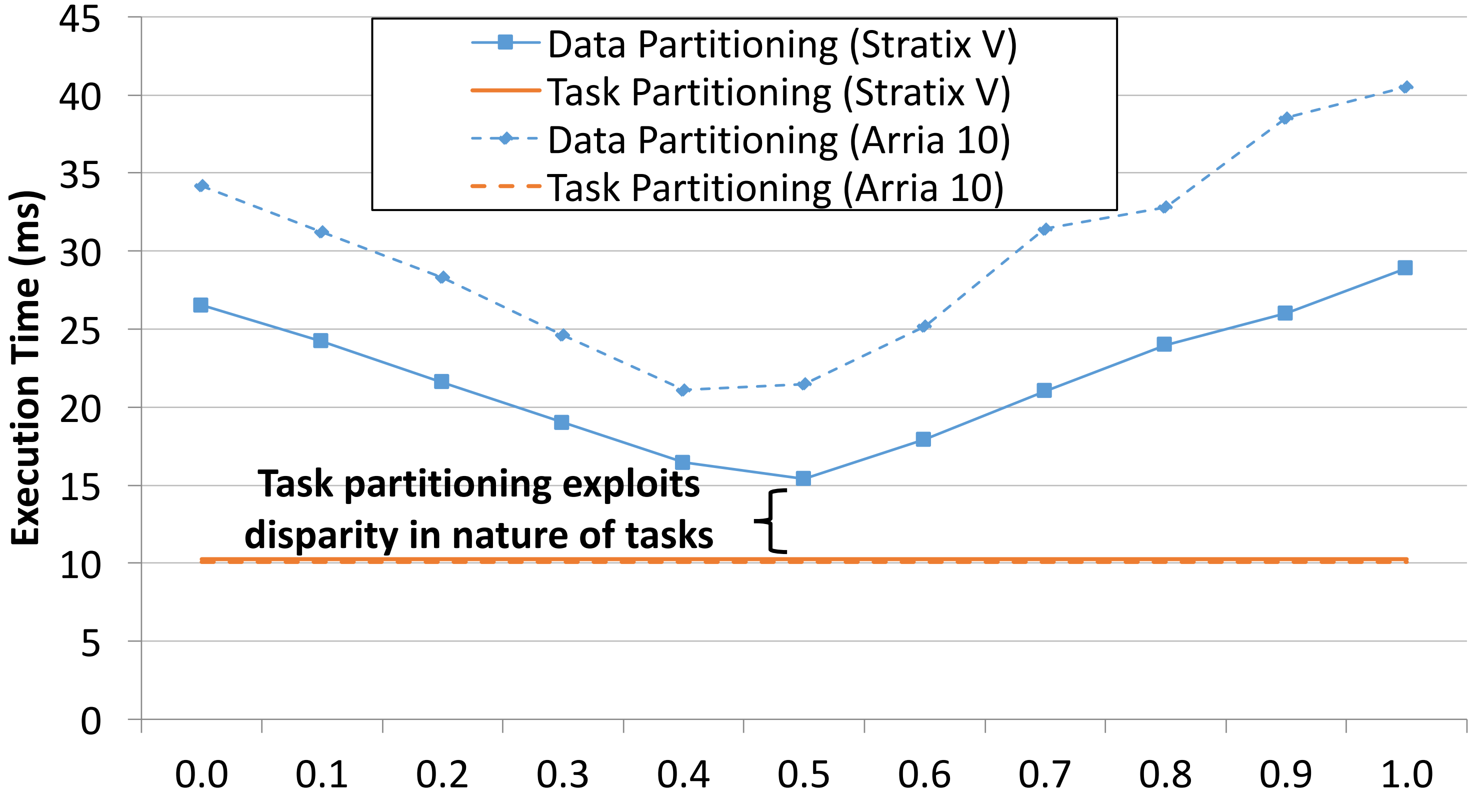
<u>Case Study:</u>
Canny Edge Detection



<u>Source:</u> Collaborative Computing for Heterogeneous Integrated Systems. *ICPE'17 Vision Track.*

Benefits of Collaboration on FPGA

<u>Case Study:</u>
Random Sample Consensus



<u>Source:</u> Collaborative Computing for Heterogeneous Integrated Systems. *ICPE'17 Vision Track.*

Released

- Website: chai-benchmarks.github.io
- Code: github.com/chai-benchmarks/chai
- Online Forum: groups.google.com/d/forum/chai-dev
- Papers:
  - **Chai: Collaborative Heterogeneous Applications for Integrated-architectures.** *ISPASS'17*.
  - **Collaborative Computing for Heterogeneous Integrated Systems.** *ICPE'17 Vision Track*.

CHAI

Chai: Collaborative Heterogeneous Applications for Integrated-architectures

Juan Gómez-Luna[1], Izzat El Hajj[2], Li-Wen Chang[2], Víctor García-Flores[3,4], Simon Garcia de Gonzalo[2], Thomas B. Jablin[2,5], Antonio J. Peña[4], and Wen-mei Hwu[2]

[1]Universidad de Córdoba, [2]University of Illinois at Urbana-Champaign, [3]Universitat Politècnica de Catalunya, [4]Barcelona Supercomputing Center, [5]MulticoreWare, Inc.

URL: chai-benchmarks.github.io

Thank You! ☺