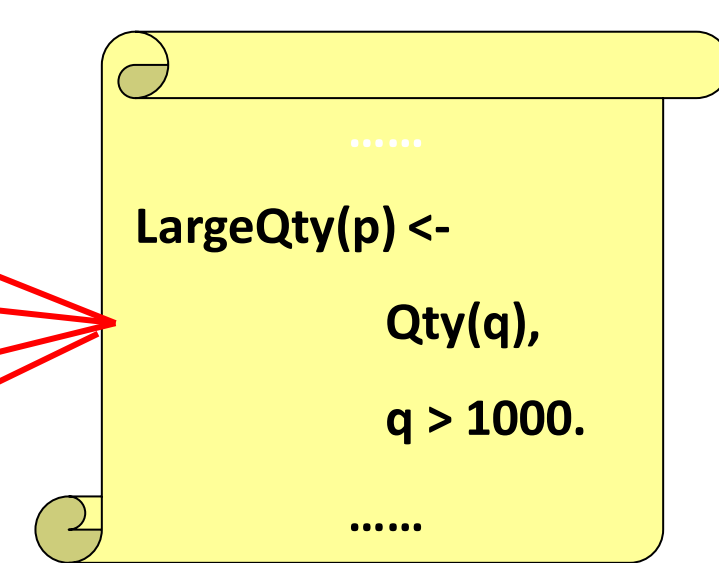# Exploration of Data Warehousing and Graph Applications with GPUs

Ifrah Saeed, Se Hoon Shon, Haicheng Wu, Jeffrey Young, Sudhakar Yalamanchili

School of Electrical and Computer Engineering, Georgia Institute of Technology

Application Space: Data Warehousing

LargeQty(p) <-
Qty(q),
q > 1000.
......

Applications: Retail analysis, forecasting, pricing, etc…

Combination of relational data queries and compute kernels

- Current applications process 1 to 50 TBs of data [1]
- Not a traditional domain for GPU acceleration, but parallel queries experience good speedup on GPUs [2]

Base Primitives and Data Structures

- Primitives

| Relational Algebra	Math	Built-in	Others
• PROJECT	• Arithmetic: + - * /	• String	• Sort
• PRODUCT	• Aggregation	• Datetime	• Unique
• SELECT			
• JOIN			
• SET			

Red: Thrust library
Blue: ModernGPU library

- Data Structure: Key-Value Store
  – weakly sorted arrays of densely packed tuples
  – Support up to 1024 bit tuple
  – Support int, float, string, date

id   price   padding   tax   zeros
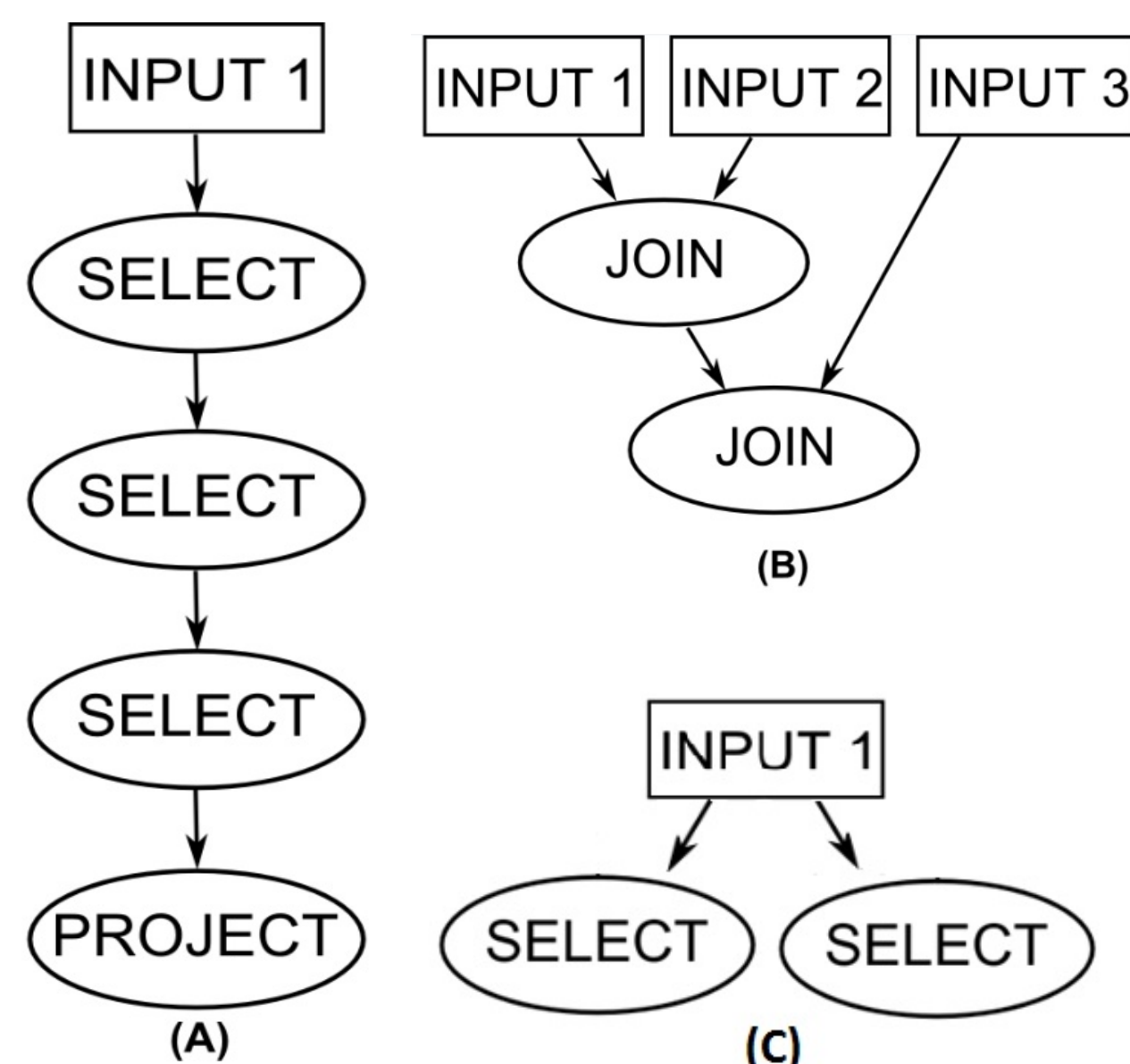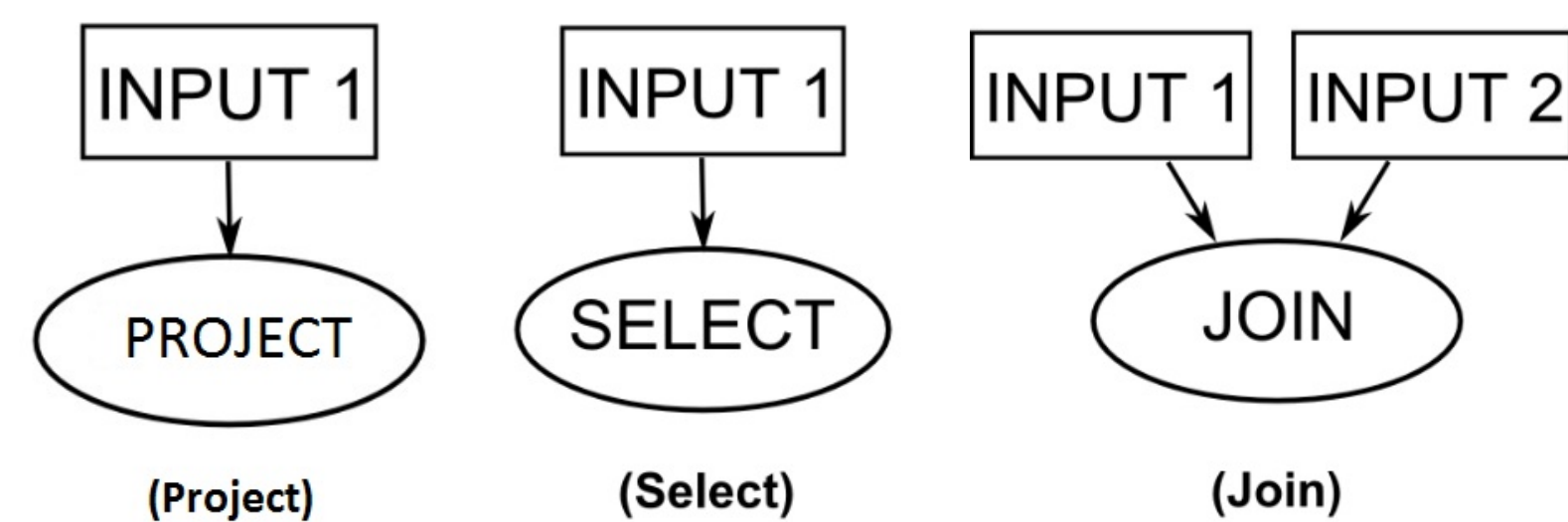4 bytes
8 bytes
16 bytes
Key   Value
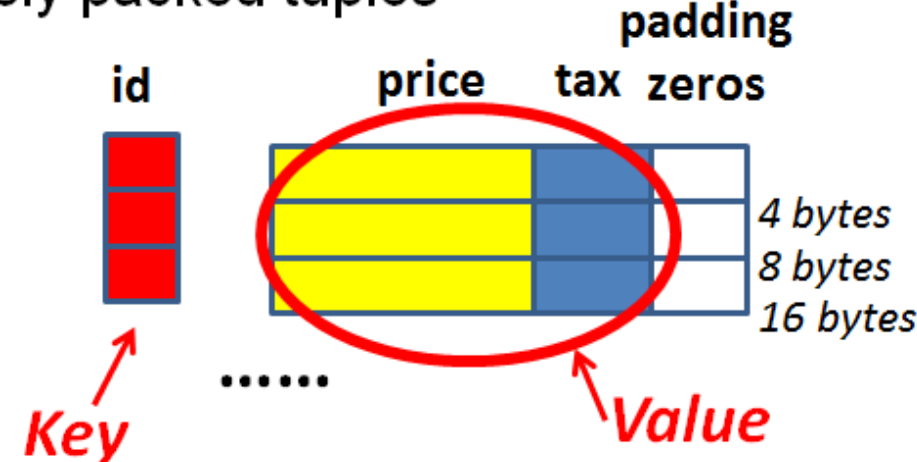
OpenCL Backend for Red Fox

- Port of Red Fox to new accelerator platforms using OpenCL
- Initial performance is evaluated using 16-bit key-value store and TPC-H micro-benchmarks

TPC-H Micro-benchmarks

Micro-benchmarks represent frequently occurring patterns of operators from the 22 queries in TPC-H [3].

INPUT 1 → PROJECT (Project)
INPUT 1 → SELECT (Select)
INPUT 1, INPUT 2 → JOIN (Join)

INPUT 1 → SELECT → SELECT → SELECT → PROJECT (A)
INPUT 1, INPUT 2, INPUT 3 → JOIN → JOIN (B)
INPUT 1 → SELECT, SELECT (C)

Graph Applications – BFS (Ongoing Work)

- Breadth First Search (BFS) is important for large-scale analysis of social networks and linked datasets such as Wikipedia
  - Limited PCIe bandwidth makes it difficult to map this algorithm to clusters of accelerators
    - Exchange of edge lists requires low-latency transfer
- Current work involves the design of an efficient partitioning scheme that maps across a cluster
  - Building on single-node work done by Merrill [4] as well as CPU-based Graph500 implementations
  - Optimized OpenCL and CUDA versions will allow for high performance with different accelerators
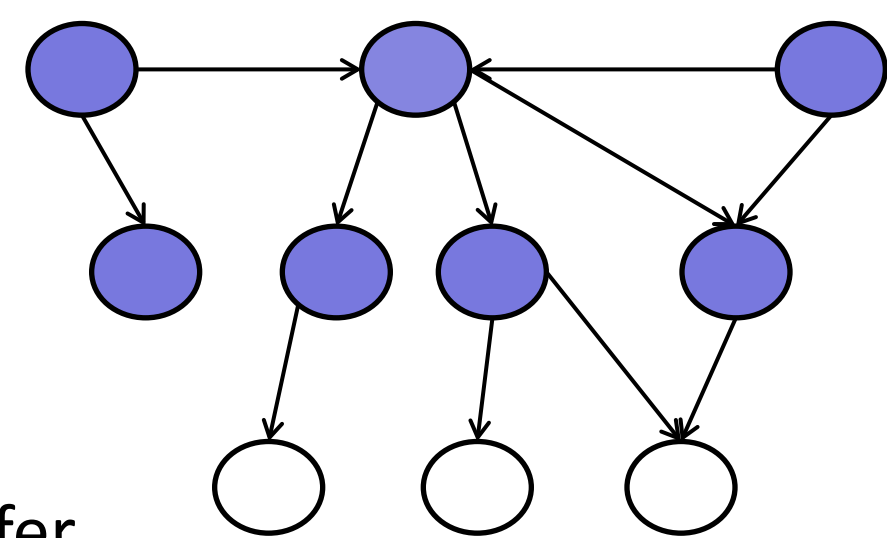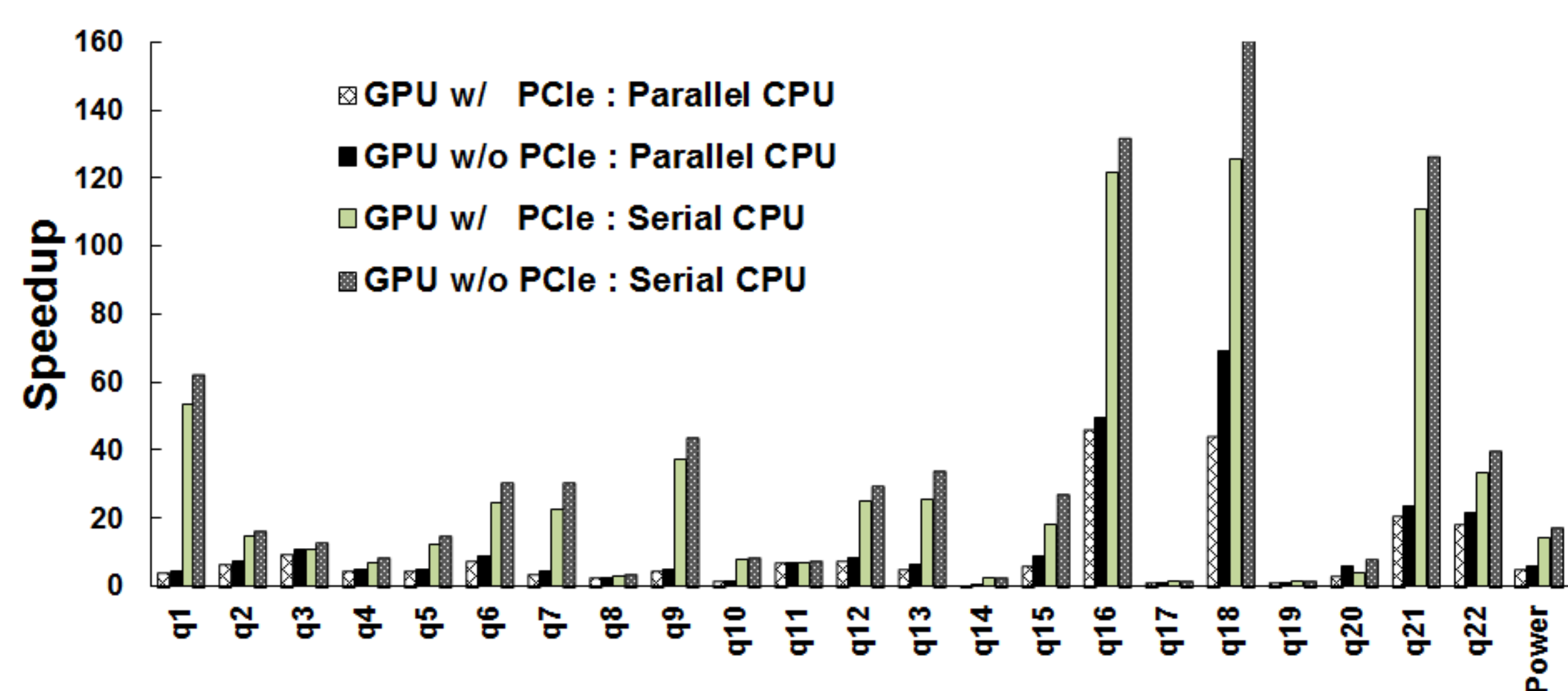

Our Approach: Red Fox Compiler and Runtime
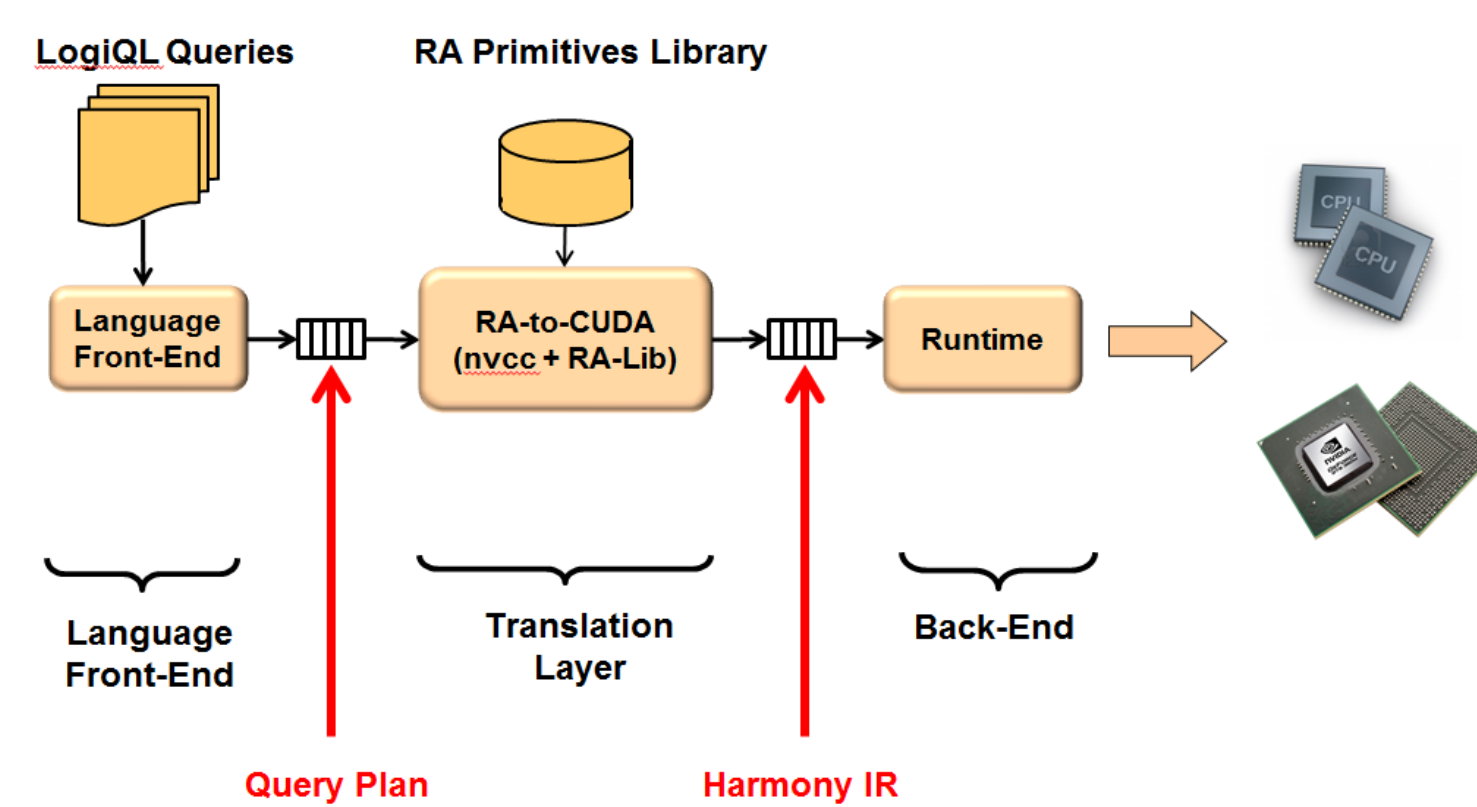
Collaboration with LogicBlox Inc.

- Relational queries are translated to optimized query plans and GPU primitives via the Red Fox compilation and runtime framework [3]
- Initial work used CUDA-based primitives; current work focuses on OpenCL-based primitives

Red Fox CPU/GPU Comparison for TPC-H*

On average (geo mean)

GPU w/ PCIe : Parallel CPU	= 4.92x
GPU w/o PCIe : Parallel CPU	= 5.96x
GPU w/ PCIe : Serial CPU	= 14.30x
GPU w/o PCIe : Serial CPU	= 17.31x

Experimental Setup

CPU	Intel i7-4771 @ 3.50GHz
GPU	GeForce GTX Titan
PCIe	3.0 x 16
OS	Ubuntu 12.04
G++/GCC	4.6
NVCC	5.5
Thrust	1.7

*CPU version runs LogicBlox 4.0 on Amazon EC2 instance cr1.8xlarge. Scale Factor for TPC-H = 1. CUDA-based implementation.

OpenCL Backend Preliminary Results

Experimental Setup

CPU
- Intel® Core i5-3470 CPU @ 3.20GHz
- Intel Xeon Phi (*TBD*)

GPUs
- Intel® HD Graphics 2500 - 6 Compute Units
- AMD Fused GPU (HD7660D) - 6 Compute Units
- AMD Discrete GPU (HD5800) - 20 Compute Units

- Operating System - Windows 7
- Framework - OpenCL 1.2

Legend:
- Serial Implementation on AMD CPU
- AMD Discrete GPU
- AMD Fused GPU
- Serial Implementation on Intel CPU
- Intel CPU
- Intel Fused GPU

Project - Execution Time | Select - Execution Time | Join - Execution Time
A - Execution Time | B - Execution Time | C - Execution Time
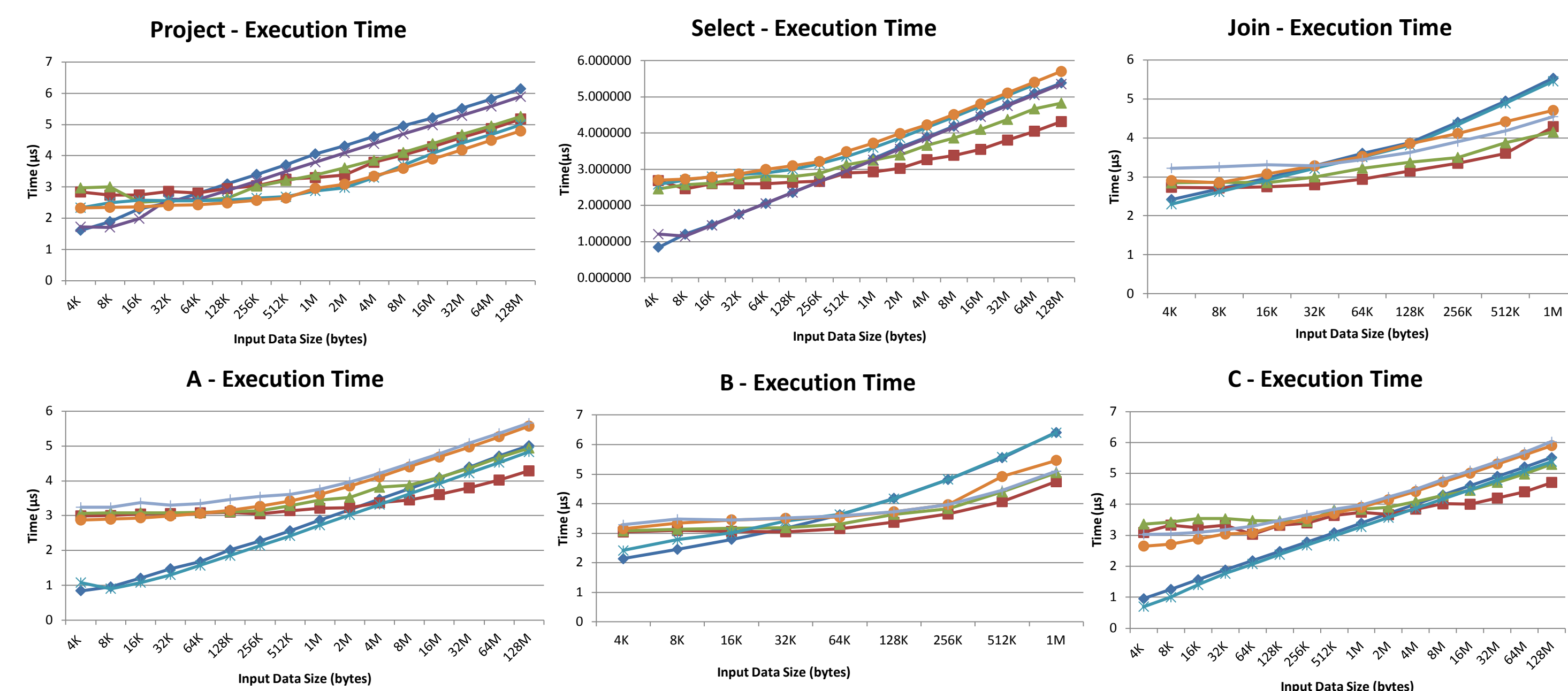
References

[1] IND. Oracle Users Group. *A New Dimension to Data Warehousing: 2011 IOUG Data Warehousing Survey.*

[2] B. He, *et al. Relational query co-processing on graphics processors*. TODS, 2009.

[3] Wu, H., Diamos, G., Cadambi, S., Yalamanchili, S., *Kernel Weaver: Automatically Fusing Database Primitives for Efficient GPU Computation,* MICRO 2012

[4] Merrill, Duane, et al., "Scalable GPU Graph Traversal", PPoPP '12