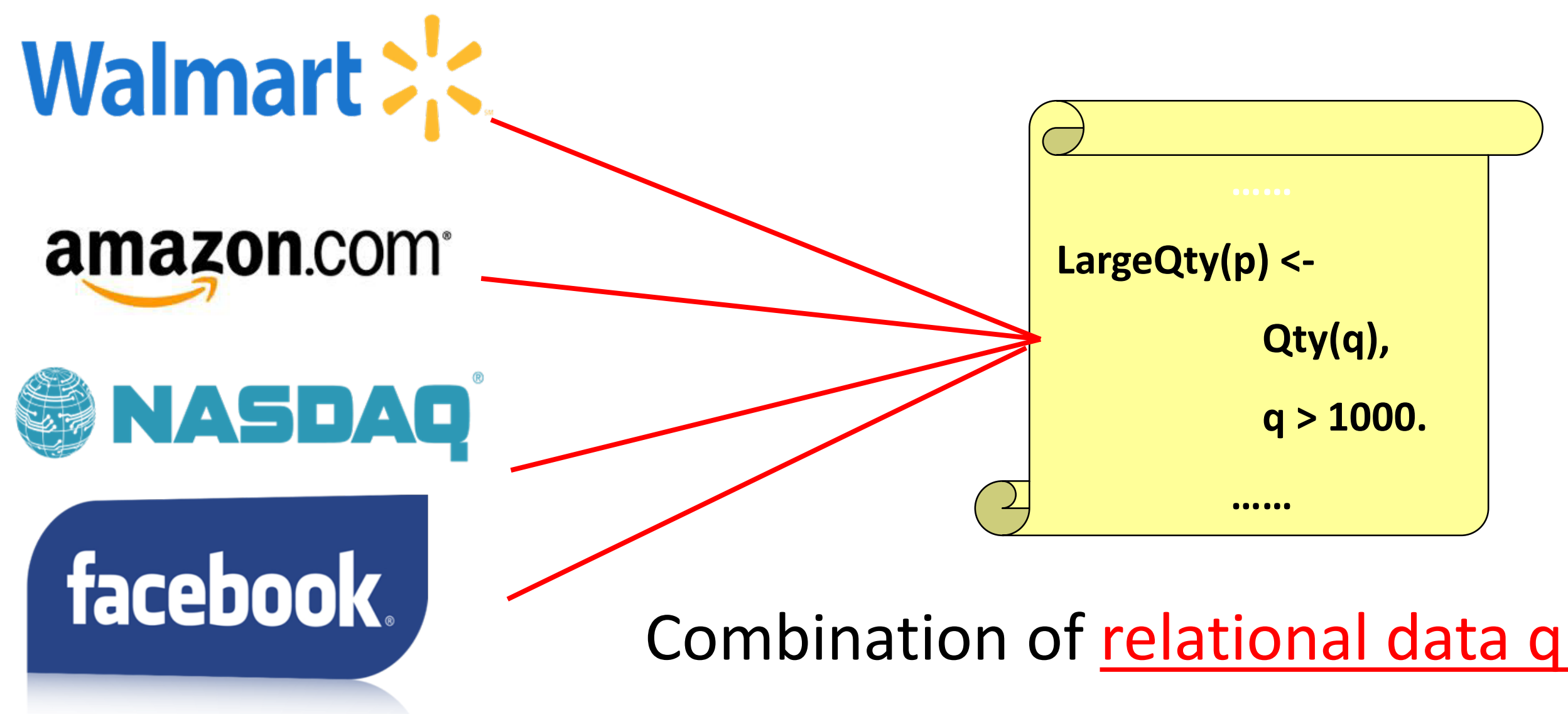
Optimizing Relational Computing Performance on Heterogeneous Processors

H. Wu, I. Saeed, J. Young, C. Kersey, and S. Yalamanchili
School of Electrical and Computer Engineering
Georgia Institute of Technology

High Performance Relational Computing



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
- Goal: **10X-100X** application speedup over multicore processors

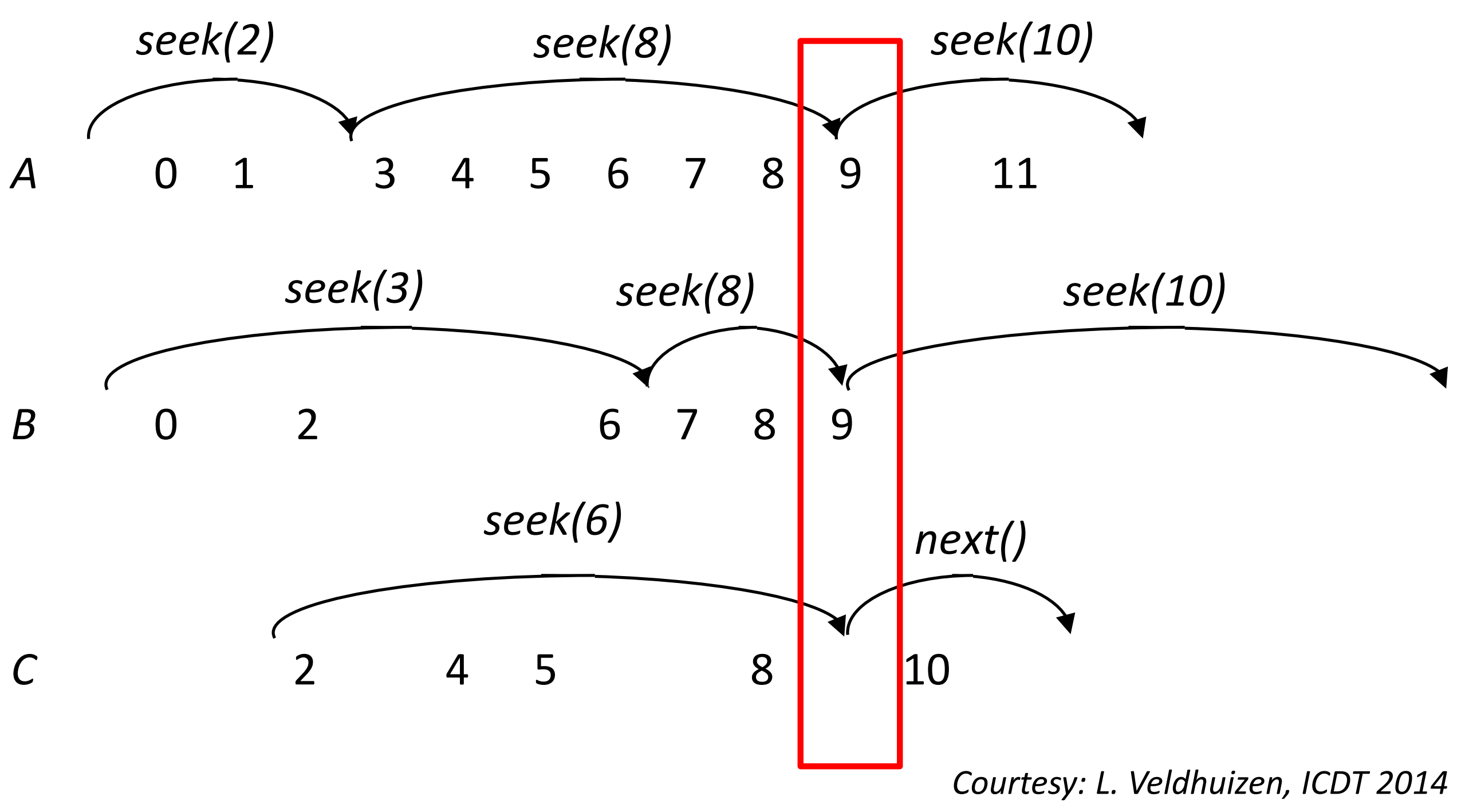
Multi-Predicate Join for SIMD Accelerators

- Implementation of Leapfrog Triejoin (LFTJ) on GPUs
- A worst-case optimal multi-predicate join algorithm
- CPU-version - T. L. Veldhuizen, *ICDT 2014*

Key idea: *Leap* over large segments to seek a specific (e.g., join) value

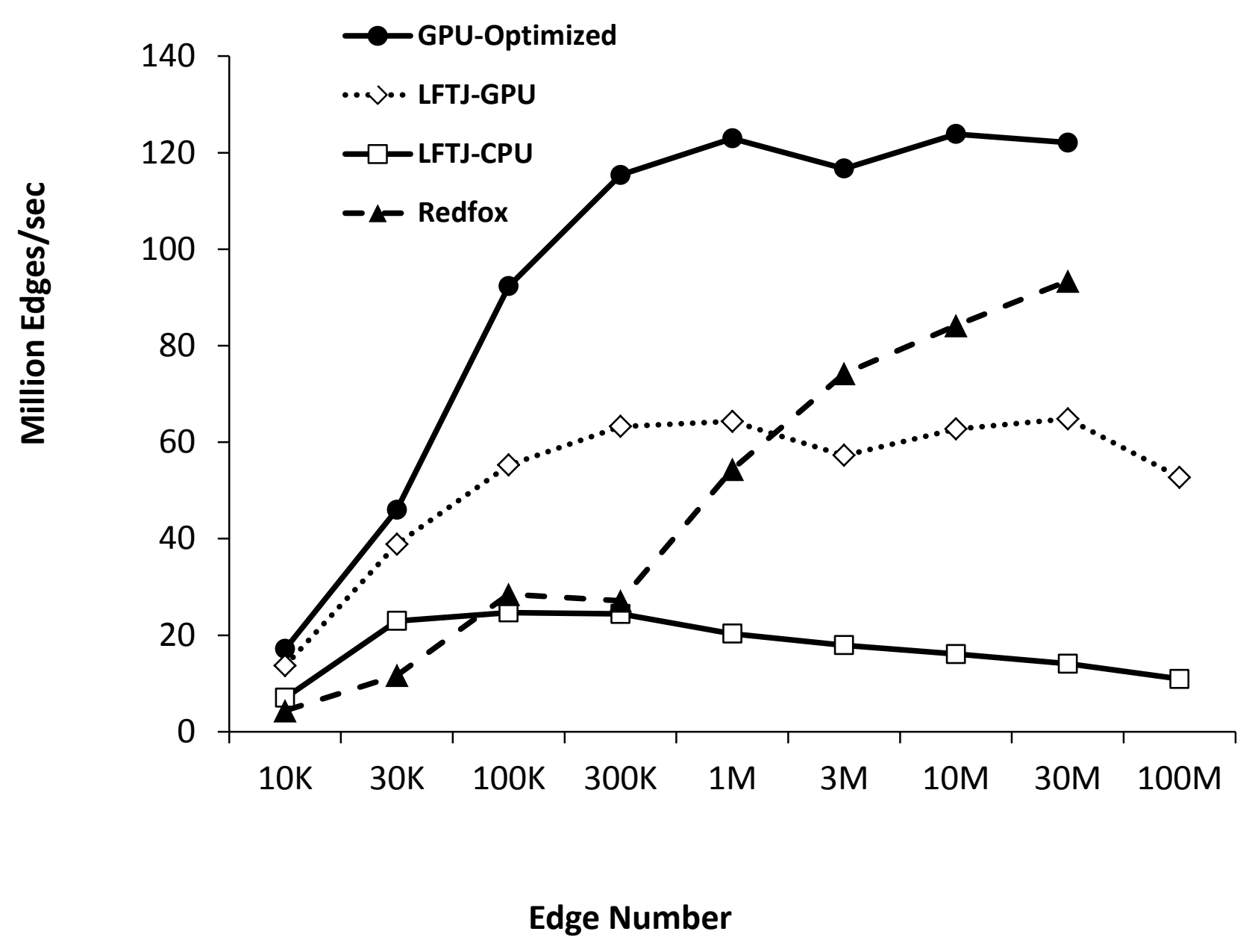
Primitives next() and seek()

Example using a linear iterator

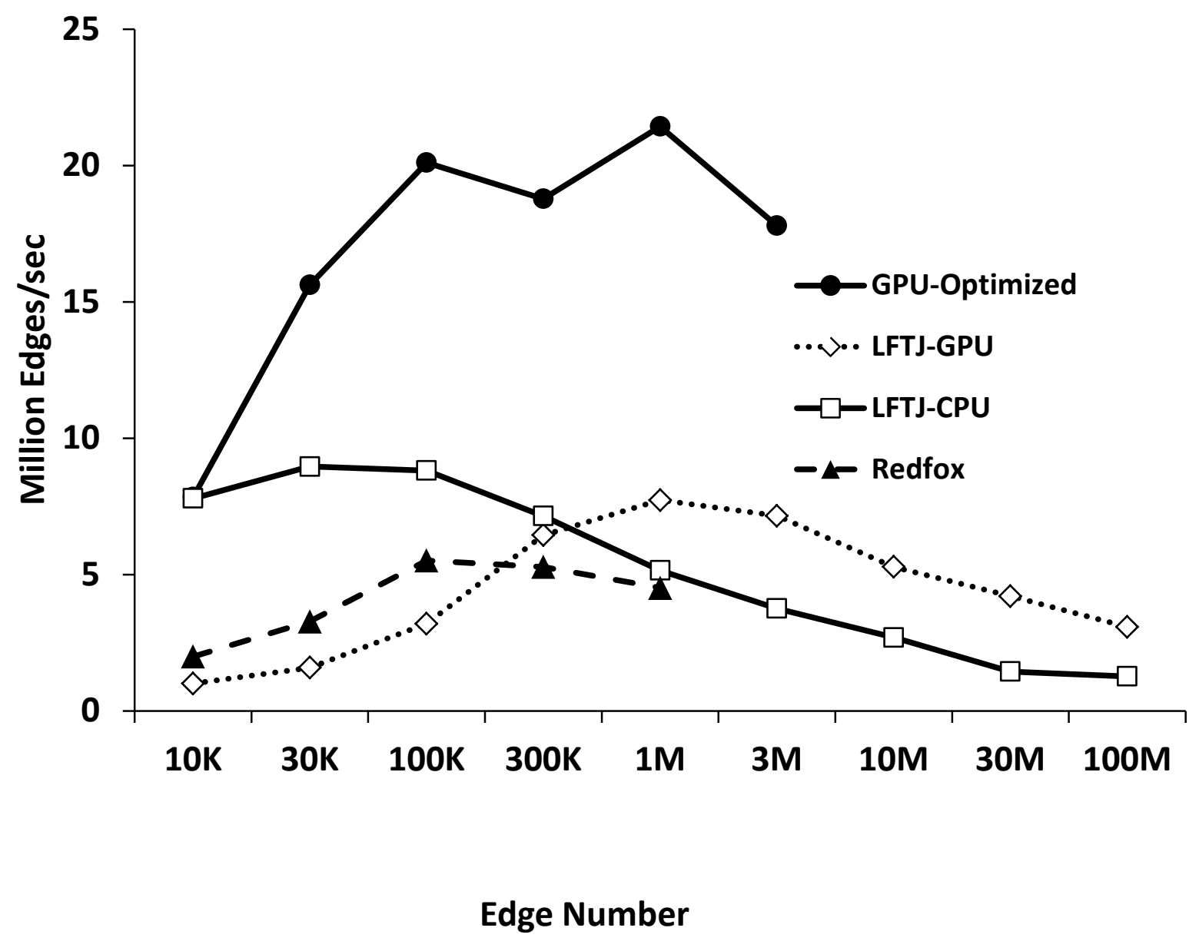


Courtesy: L. Veldhuizen, ICDT 2014

- Benefits
  - Smaller memory footprint for temporary results
  - No data reconstruction, e.g. sorting or hash table construction
- Worst-case optimal multi-predicate join in a SIMD accelerator[4]



Three-clique problem

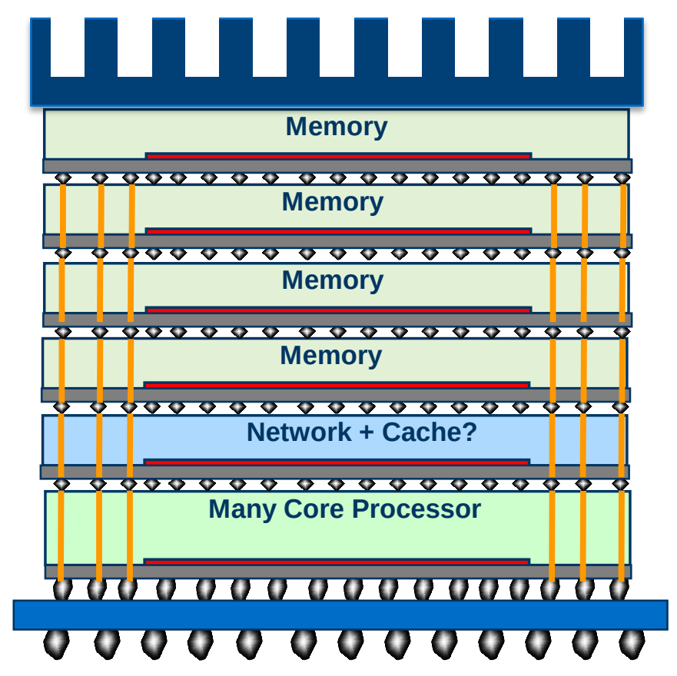


Four-clique problem

Near Memory Data Intensive Computing

Moving the compute primitives for data analytics into the memory system

- Parametric C++ processor synthesis environment
- HARP family of data parallel processors
- RISC processor core
- Assembler/emulator tools
- OpenCL compiler (in progress)

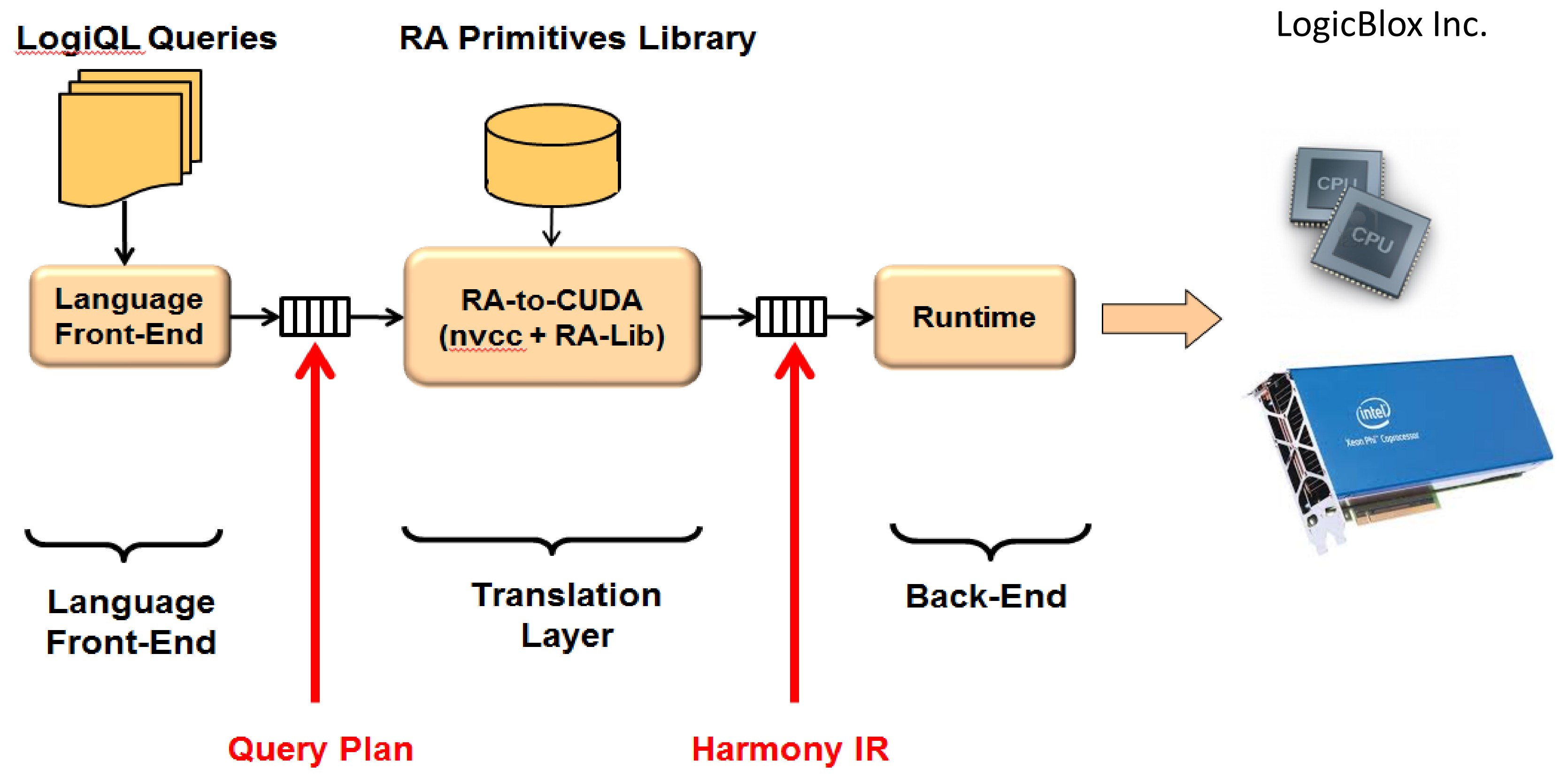


- Prototypes in execution in FPGAs
- Candidate Primitives
  - Relational Algebra, MemcacheD,
  - Sorting, search, encryption
  - Sparse matrix operations
  - Video and Text search

Intel Science & Technology
Center for Cloud Computing

Our Approach: Red Fox Tool Chain

Collaboration with LogicBlox Inc.



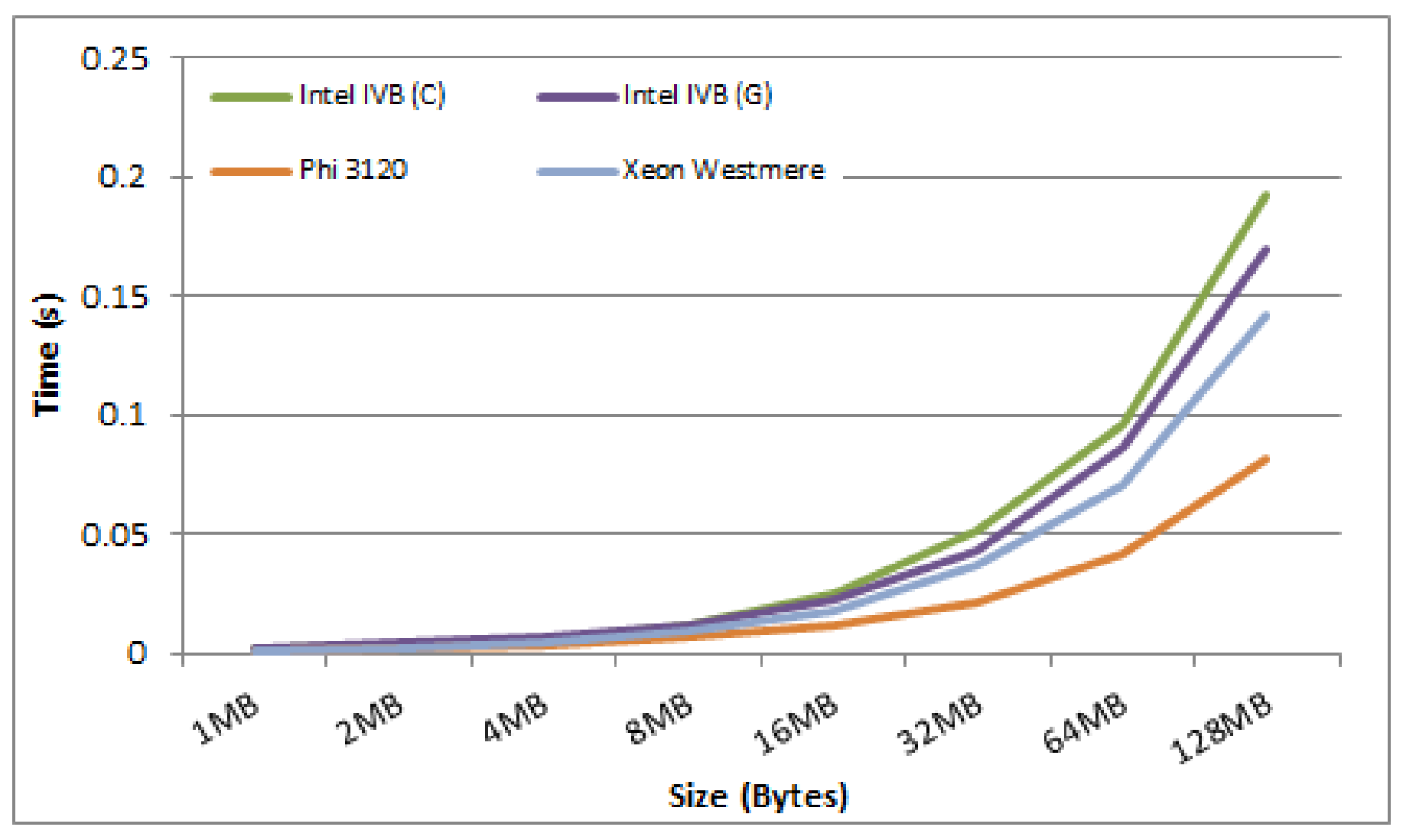
- Relational queries are translated to optimized query plans and GPU primitives via the Red Fox compilation and runtime framework [3]
- Early work used CUDA-based primitives; current work focuses on OpenCL-based primitives
- Execution on integrated and attached accelerators

SHOC Benchmark Implementations

http://keeneland.gatech.edu/software/keeneland/shoc

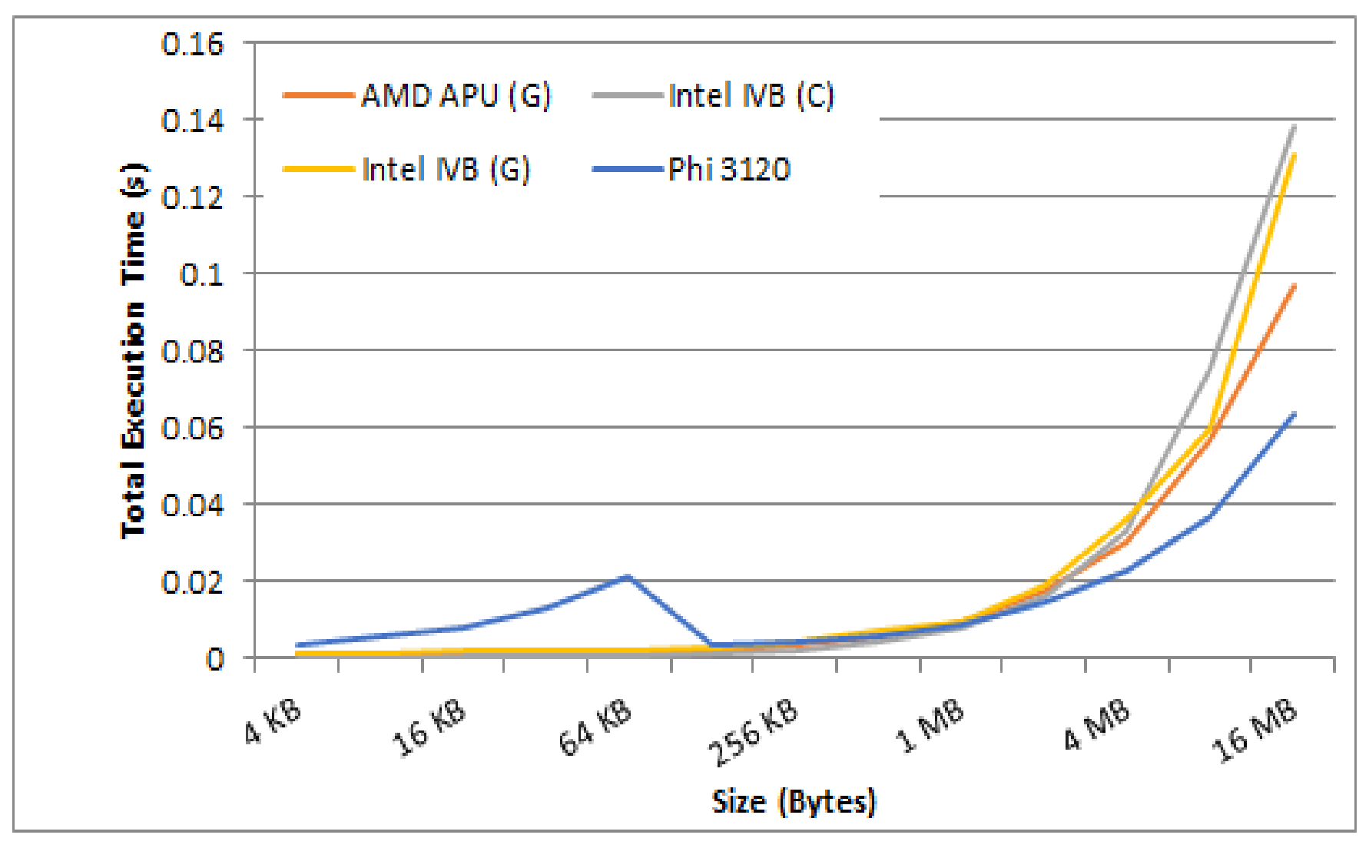
- SHOC - Scalable Heterogeneous Computing Benchmark Suite
- Standardized benchmark suite across languages and platforms
- Full support for OpenCL, CUDA. OpenACC and Phi Offload in progress.
  - Relational Algebra Primitives and TPC-H microbenchmarks
  - TPC-H queries
- Current Effort: RA implementations on Intel Phi and Gen

Select (OpenCL)



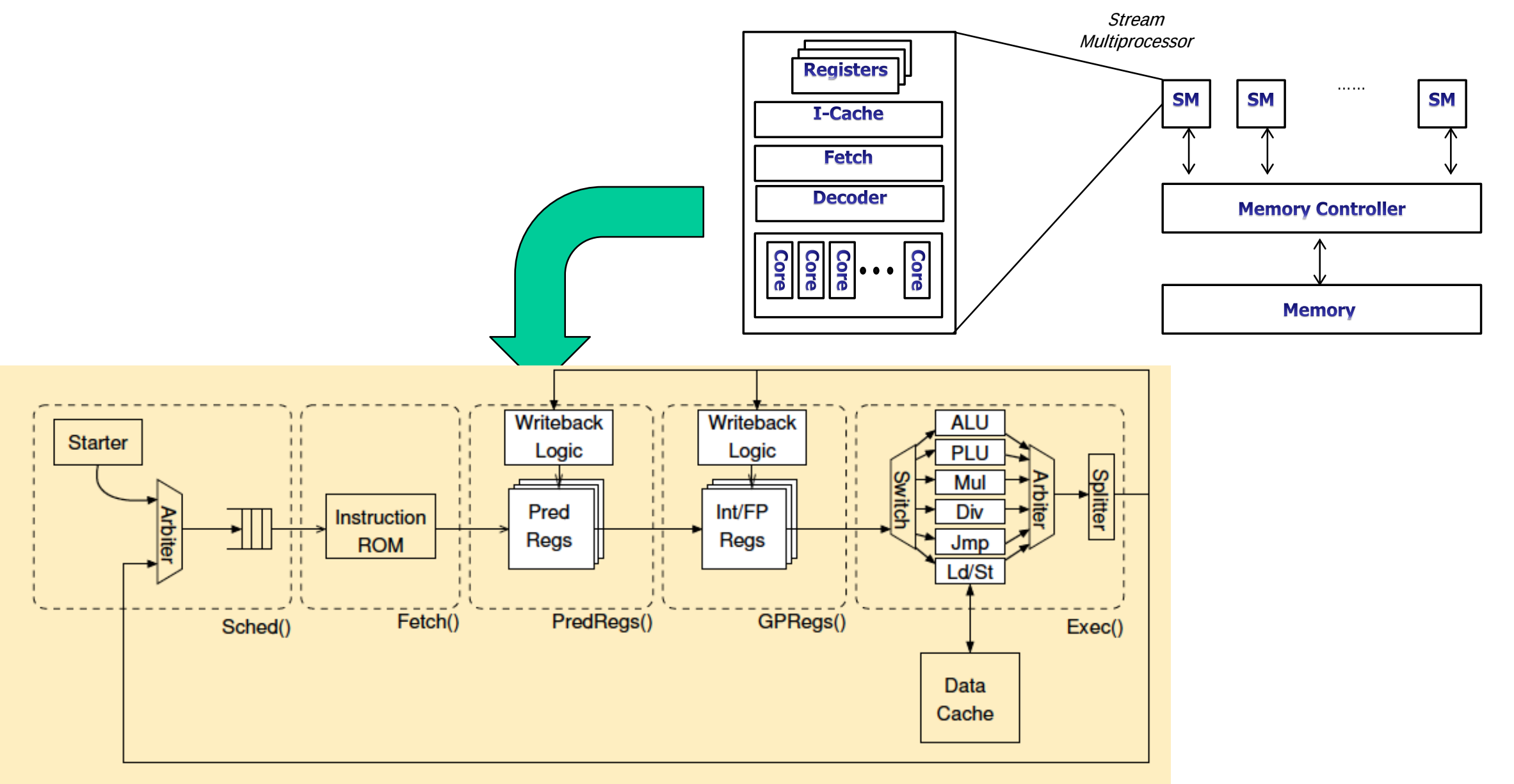
Xeon Phi excels at highly parallel primitives

Microbenchmark B (Seq. Joins)



Output size limits performance of joins

HARP Architecture V2



- Configurable
  - Registers per thread, number of lanes, data path width
  - Choice of functional units
- Small- ~1500 lines of C++

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] H. Wu, *et al. Red Fox: An Execution Environment for Relational Query Processing on GPUs,* CGO 2014
[4] H. Wu, *et al. Multipredicate Join Algorithms for Accelerating Relational Graph Processing on GPUs,* ADMS 2014

Carnegie Mellon University Georgia Tech intel PRINCETON UNIVERSITY UC Berkeley UNIVERSITY of WASHINGTON