Satisfying Data-Intensive Queries Using GPU Clusters

Haicheng Wu, Jeff Young
Sudhakar Yalamanchili

Computer Architecture and Systems Laboratory
Center for Experimental Research in Computer Systems
School of Electrical and Computer Engineering
Georgia Institute of Technology

Sponsors: AIC, AMD, LogicBlox Inc., National Science Foundation, NEC, NVIDIA
Application: Data Warehousing

- On-line and off-line analysis
  - Retail analysis
  - Forecasting
  - Pricing
  - Etc...

- Combination of relational data queries and computational kernels

- Current applications process 1 to 50 TBs of data [1]

- Techniques can be applied to other “Big Data” problems like irregular graphs, sorting

**Proposed System Model**

- **Red Fox**: Compilation and optimization of queries for GPUs
  - Remove need for application developer to optimize applications to run on GPUs
- **Oncilla**: Global Address Space (GAS) layer
  - Create an API to simplify data movement and scheduling
Red Fox Compilation Flow

Datalog Queries

LogicBlox Front-End

Primitives Library

RA-to-PTX (nvcc + RA-Lib)

Kernel Weaver

Runtime Manager

Language Front-End

Translation Layer

Back-End

Query Plan

PTX/Binary Kernel
Relational Algebra Primitives on GPUs

Raw Performance (NVIDIA C2050)

Fastest known for GPUs!

- Multi-stage algorithm
- Simple primitives are close to maximum performance
- More complex primitives could show better performance with newer implementations (in progress with NVIDIA Research)
Red Fox: TPC-H Q1 Results

- GPU computation scales well with problem size
- Improved primitives could lead to further 10x speedup
Oncilla: Fabrics for Accelerator Clouds

Companies: LogicBlox, NVIDIA, AIC

- **Goal**: Transparent, efficient host memory aggregation across node for accelerators
- **Solution**: Use Global Address Spaces (GAS) and commodity fabrics (HT, QPI, PCIe, 10GE, IB)
  - Support in-core databases using software from Red Fox project
Oncilla aims to combine support for multiple types of data transfer and CUDA-based optimizations under a simplified runtime.

- Ex: “oncilla_malloc(2 GB, node2, gpumem)”

- Enable application developers and schedulers to take advantage of high-performance GAS without needing to be experts in specialized hardware.

Oncilla: Efficient Data Movement
Questions?

For more information:

Red Fox:  
http://gpuocelot.gatech.edu/projects/compiler-projects/

Oncilla:  

J. Young, S. Yalamanchili, *Commodity Converged Fabrics for Global Address Spaces in Accelerator Clouds,”* HPCC, June, 2012