Red Fox: An Execution Environment for Relational Query Processing on GPUs

Haicheng Wu¹, Gregory Diamos², Tim Sheard³, Molham Aref⁴, Sean Baxter², Michael Garland², Sudhakar Yalamanchili¹

1. Georgia Institute of Technology
2. NVIDIA
3. Portland State University
4. LogicBlox Inc.
System Diversity Today

Amazon EC2 GPU Instances

Mobile Platforms (DSP, GPUs)

**Hardware Diversity is Mainstream**

Keeneland System (GPUs)

Cray Titan (GPUs)
Relational Queries on Modern GPUs

The Opportunity

- Significant potential data parallelism
- If data fits in GPU memory, 2x—27x speedup has been shown.

The Problem

- Need to process 1-50 TBs of data.
- Fine grained computation
- 15–90% of the total time spent in moving data between CPU and GPU.

---


The Challenge

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

Candidate Application Domains

Walmart

amazon.com

NASDAQ

facebook

Large Graphs

Relational Computations Over Massive Unstructured Data Sets: Sustain 10X – 100X throughput over multicore
Goal and Strategy

**GOAL**

- Build a compilation chain to bridge the semantic gap between *Relational Queries* and *GPU* execution models
  - 10x-100X speedup for relational queries over multicore

**Strategy**

1. Optimized Primitive Design
   - Fastest published GPU RA primitive implementations (PPoPP2013)
2. Minimize Data Movement Cost (MICRO2012)
   - Between CPU and GPU
   - Between GPU Cores and GPU Memory
3. Query level compilation and optimizations (CGO2014)
The Big Picture

LogicBlox RT parcels out work units and manages out-of-core data.

Red Fox extends LogicBlox environment to support GPUs.
LogicBlox Domain Decomposition Policy

- **Sand, Not Boxes**
  - Fitting boxes into a shipping container => hard (NP-Complete)
  - Pouring sand into a dump truck => dead easy

- Large query is partitioned into very fine grained work units
  - Work unit size should fit GPU memory
  - GPU work unit size will be larger than CPU size
  - Still many problems ahead, e.g. caching data in GPU

- **Red Fox**: Make the GPU(s) look like very high performance cores!
Domain Specific Compilation: Red Fox

First thing first, **mapping the computation** to GPU

Source Language: LogiQL

- LogiQL is based on Datalog
  - A declarative programming language
  - Extended Datalog with aggregations, arithmetic, etc.

- Find more in [http://www.logicblox.com/technology.html](http://www.logicblox.com/technology.html)

- Example
  
  \[
  \text{ancestor}(x,y) \leftarrow \text{parent}(x,y).
  \]

  \[
  \text{ancestor}(x,y) \leftarrow \text{ancestor}(x,t), \text{ancestor}(t,y).
  \]

  recursive definition
Language Front-end

Front-End Compilation Flow

LogicBlox Parser

- Parsing
- Type Checking
- AST Optimization

LogicBlox Flow

RA Translation

- common (sub)expression elimination
- dead code elimination
- more optimizations are needed

Query Plan

Pass Manager

Red Fox

Red Fox Compilation Flow: Translating LogiQL Queries to Relational Algebra (RA)

Industry strength optimization
Structure of the Two IRs:

Query Plan

- **Module**
  - **Variable**
    - **Types**
    - **Data**
  - **Basic Block**
    - **Operator**
      - **Input**
      - **Output**

Harmony IR

- **Module**
  - **Variable**
    - **Types**
    - **Data**
  - **Basic Block**
    - **Operator**
      - **Input**
      - **Output**
      - **CUDA**

RA Primitives

RA-to-GPU Compiler (nvcc + RA-Lib)
Two IRs Enable More Choices

LogiQL Queries

SQL Queries

Design Supports Extensions to
- Other Language Front-Ends
- Other Back-ends
Primitive Library: Data Structures

- **Key-Value Store**
  - Arrays of densely packed tuples
  - Support for up to 1024 bit tuples
  - Support int, float, string, date

![Diagram of Key-Value Store](image-url)
Primitive Library: When Storing Strings

Key

Value

String Table (Len = 32)

String Table (Len = 64)

String Table (Len = 128)
Primitive Library: Performance

Stores the GPU implementation of following primitives

- **Relational Algebra**
  - PROJECT
  - PRODUCT
  - SELECT
  - JOIN
  - SET

- **Math**
  - Arithmetic: + - * /
  - Aggregation

- **Built-in**
  - String
  - Datetime

- **Others**
  - Sort
  - Unique

RA performance on GPU (PPoPP 2013)*


Measured on Tesla C2050
Random Integers as inputs
Forward Compatibility: Primitive Library Today

- Use best implementations from the state of the art
- Easily integrate improved algorithms designed by 3rd parties

<table>
<thead>
<tr>
<th>Relational Algebra</th>
<th>Math</th>
<th>Built-in</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PROJECT</td>
<td>• Arithmetic: + - * /</td>
<td>• String</td>
<td>• Merge Sort</td>
</tr>
<tr>
<td>• PRODUCT</td>
<td>• Aggregation</td>
<td>• Datetime</td>
<td>• Radix Sort</td>
</tr>
<tr>
<td>• SELECT</td>
<td></td>
<td></td>
<td>• Unique</td>
</tr>
<tr>
<td>• JOIN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• SET</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Red: Thrust library
Green: ModernGPU library\(^1\)
   - Merge Sort
   - Sort-Merge Join
Purple: Back40Computing\(^2\)
Black: Red Fox Library

Harmony Runtime

Schedule GPU Commands on available GPUs

Current scheduling method attempts to minimize memory footprint

Allocate \( j_1 \)

Complex Scheduling such as speculative execution* is also possible

Allocate \( p_1 \)
Free \( j_1 \)

Benchmarks: TPC-H Queries

- A popular decision making benchmark suite

- Comprised of 22 queries analyzing data from 6 big tables and 2 small tables

- Scale Factor parameter to control database size
  - SF=1 corresponds to a 1GB database

Courtesy: O'Neil, O'Neil, Chen. Star Schema Benchmark.
## Experimental Environment

### Red Fox

<table>
<thead>
<tr>
<th>Component</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel i7-4771 @ 3.50GHz</td>
</tr>
<tr>
<td>GPU</td>
<td>Geforce GTX Titan (2688 cores, $1000 USD)</td>
</tr>
<tr>
<td>PCIe</td>
<td>3.0 x 16</td>
</tr>
<tr>
<td>OS</td>
<td>Ubuntu 12.04</td>
</tr>
<tr>
<td>G++/GCC</td>
<td>4.6</td>
</tr>
<tr>
<td>NVCC</td>
<td>5.5</td>
</tr>
<tr>
<td>Thrust</td>
<td>1.7</td>
</tr>
</tbody>
</table>

### LogicBlox 4.0

Amazon EC2 instance cr1.8xlarge
- 32 threads run on 16 cores
- CPU cost - $3000 USD
Red Fox TPC-H (SF=1) Comparison with CPU

>10x Faster with 1/3 Price

On average (geo mean)
- GPU w/ PCIe : Parallel CPU = 11x
- GPU w/o PCIe : Parallel CPU = 15x

This performance is viewed as lower bound - more improvements are coming

Find latest performance and query plans in
Red Fox TPC-H (SF=1) Comparison with CPU

Highest Speedup: string centric queries
- LogicBlox uses string library
- Red Fox re-implements string ops
  - 1 thread manages 1 string
  - Performance depends on string contents
  - Branch/Memory Divergence

Lowest Speedup: poor query plan
Performance of Primitives

Solutions:
• Better order of primitives
• New join algorithms, e.g. hash join, multi-predicate join
• More optimizations, e.g. kernel fusion, better runtime scheduling method
Next Steps: Running Faster, Smarter, Bigger…..

- Running Faster
  - Additional query optimizations
  - Improved RA algorithms
  - Improved run-time load distribution

- Running Smarter:
  - Extension to single node multi-GPU
  - Extension to multi-node multi-GPU

- Running Bigger
  - From in-core to out-of-core processing
The Future is Acceleration

Thank You