Satisfying Data-Intensive Queries Using GPU Clusters

Haicheng Wu, Jeffrey Young, and Sudhakar Yalamanchili

Georgia Institute of Technology

Application Space: Data Warehousing

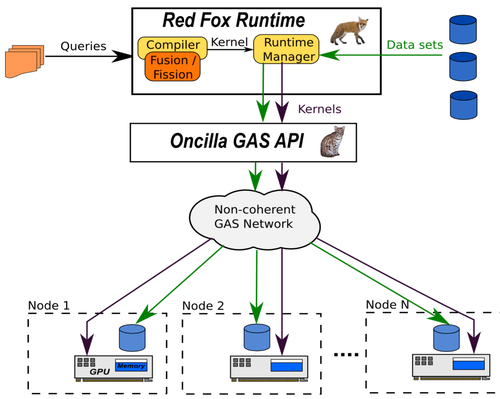
Walmart, amazon.com, NASDAQ, facebook

- 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]
- Not a traditional domain for GPU acceleration, but:
  - Parallel queries experience good speedup on GPUs [2]
  - GPU-related techniques can be applied to other "Big Data" problems like irregular graphs, sorting

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

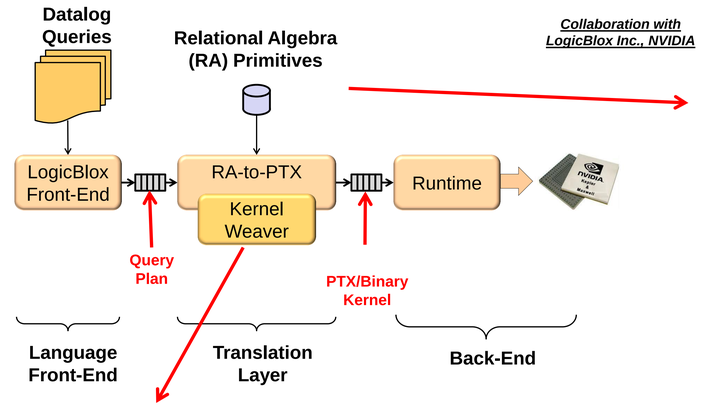
Our Solution: *Red Fox & Oncilla*

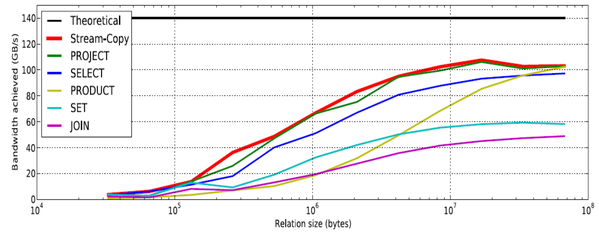
Combine Global Physical Address & GPUs



- *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)
  - Commodity Interconnects(HT, QPI, IB, PCIe)
  - HW/SW for global address space
  - Support for large in-core database
  - SW layer for optimized data movements

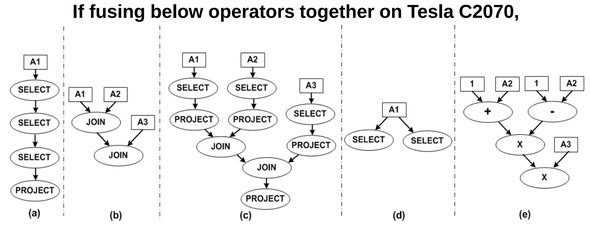
Red Fox: Execution Environment for the Enterprise





Kernel Weaver: Automatically Perform *Kernel Fusion* Optimization to reduce data movements [4,5]

If fusing below operators together on Tesla C2070,

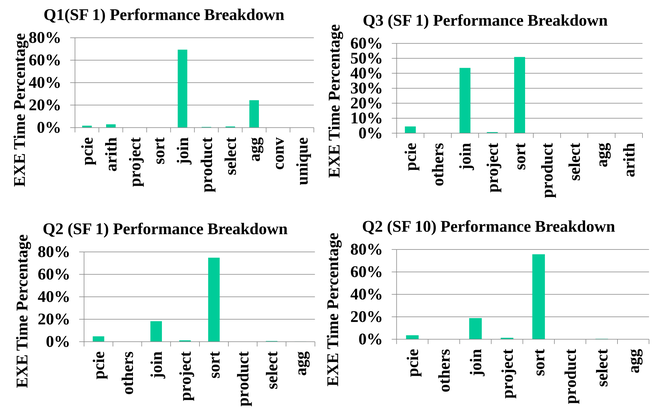


Then speedup will be 2.89x on average.

Red Fox TPC-H Benchmark Performance (Tesla C2075)

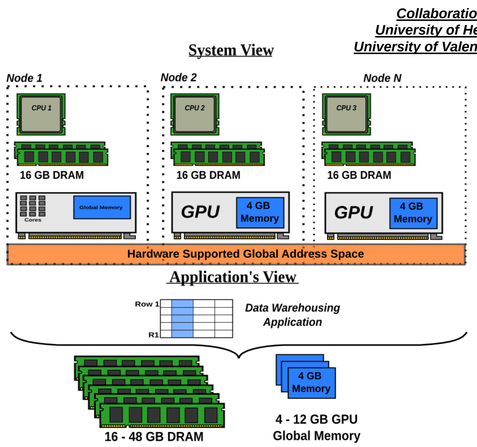
- **Execution time = PCIe + GPU Computation**
- **Problem size is restricted by GPU memory capacity**
- **No data movement optimizations**
- **Unoptimized query plan**

	Scale Factor	Execution Time (second)	Input Size (MB)	#Operators	#CUDA Kernels
Query 1	1	7.32	528	33	146
Query 2	1	0.28	38	48	164
Query 2	10	2.56	413	48	164
Query 3	1	1.62	340	34	95
Query 4	1	0.51	264	18	56
Query 5	1	1.23	277	42	111
Query 6	1	2.52	384	28	75
Query 7	1	1.16	349	53	124
Query 8	1	3.01	334	61	162
Query 9	1	19.50	437	29	93
Query 10	1	3.54	300	46	143
Query 11	1	3.48	192	21	53
Query 12	1	0.66	438	33	107



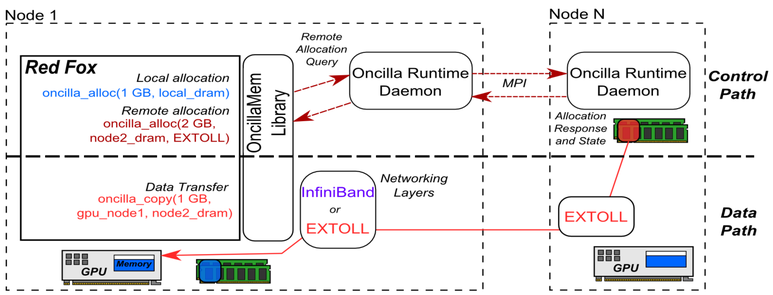
Oncilla: GAS GPU Cluster Model

Collaboration with University of Heidelberg, University of Valencia, AIC, Inc.



- Oncilla supports multiple types of data transfer and CUDA-based optimizations under a simplified runtime.
- Uses EXTOLL NIC to enable high-performance data transfers [6].

Oncilla Runtime and API Example



- Runtime handles allocation of remote memory and keeps track of connection state for different network layers.
- Oncilla API has a concept of "opaque" and "transparent" calls that allow for either simplified or more granular control of data movement between host and accelerator memories.

References

[1] Independent Oracle Users Group. "A New Dimension to Data Warehousing: 2011 IOUG Data Warehousing Survey."
[2] He, Lu, Yang, Fang, Govindaraju, Luo, Sander. "Relational query co-processing on graphics processors." TODS, 2009.
[3] Diamos, Wu, Wang, Lele, Yalamanchili. "Relational Algorithms for Multi-Bulk-Synchronous Processors." PPoPP 2013.
[4] Wu, Diamos, Wang, Cadambi, Yalamanchili, "Optimizing Data Warehousing Applications for GPUs Using Kernel Fusion/Fission." PLC 2012.
[5] Wu, Diamos, Cadambi, Yalamanchili, "Kernel Weaver: Automatically Fusing Database Primitives for Efficient GPU Computation." MICRO 2012.
[6] Fröning, Nüssle, Litz, Leber, Brüning. "On Achieving High Message Rates." CCGRID 2013.