# Oncilla – A GAS Run-time for Efficient Resource Partitioning in Data Centers

Jeffrey Young, Alex Merritt, Sudhakar Yalamanchili (Georgia Tech)

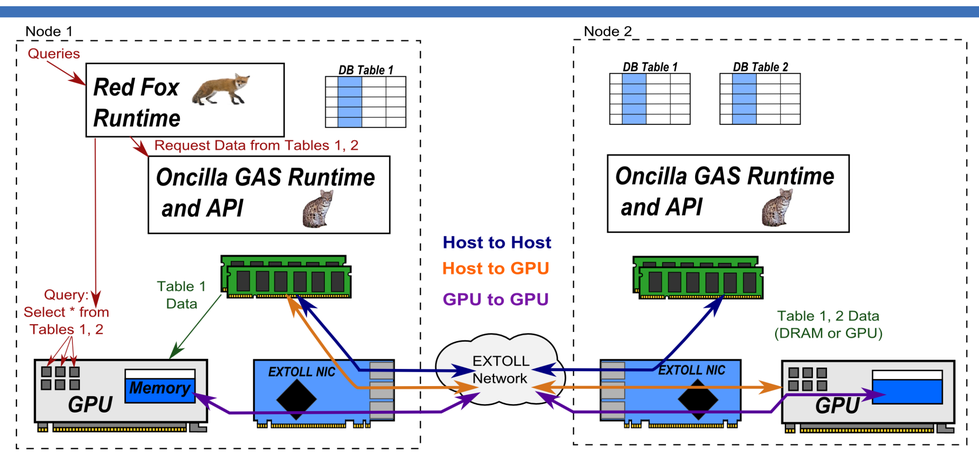
## Application Space: Data Warehousing

Walmart amazon.com and you're done. 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

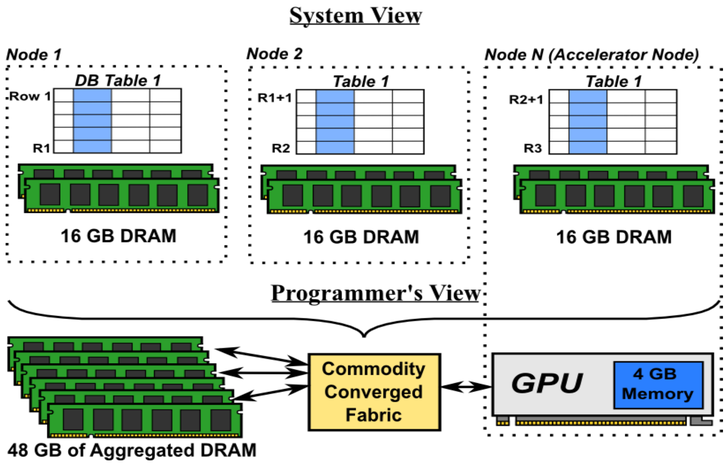
```
……
LargeQty(p) <- Qty(q),
q > 1000.
……
```

## Oncilla: Data Path



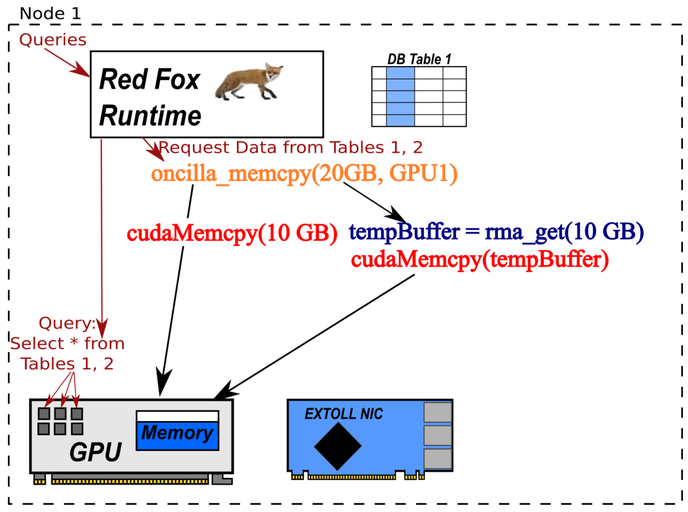

- Oncilla aims to combine support for multiple types of data transfer and CUDA-based optimizations under a simplified run-time.
- Uses EXTOLL NIC to enable high-performance data transfers
  - Ex: **oncilla_malloc(2 GB, node2, gpumem)**

## Oncilla Motivation



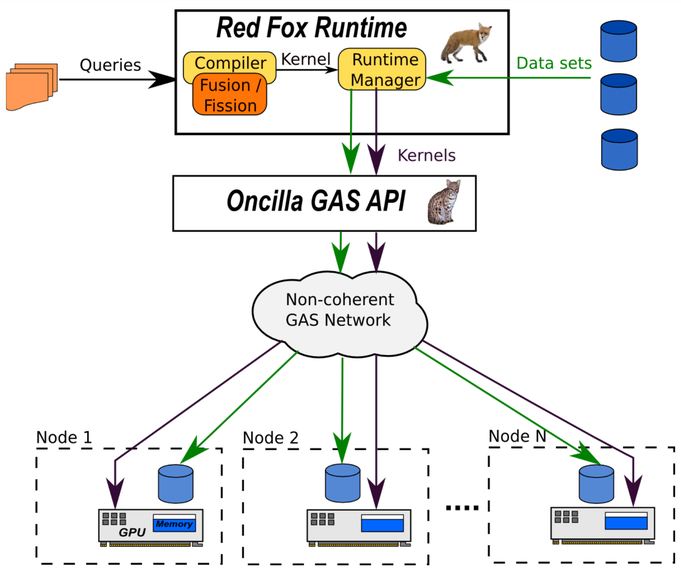

- Problem: How can resources (host and accelerator memory) be efficiently managed and aggregated for data warehousing or other Big Data applications?
- *Solution:* *Commodity-based Global Address Spaces (GAS)* can be used to better manage data center resources through more efficient data movement between CPUs, DRAM, and GPUs.

## Oncilla: Remote memcpy example



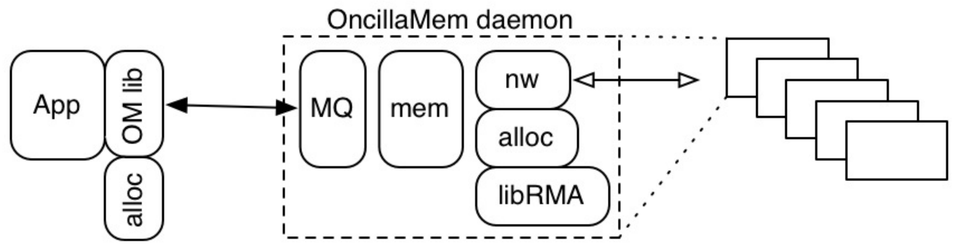

- Operations can be "opaque" or "transparent" depending on level of control developer desires.
  - Opaque operations abstract away the complexity of remote operations from application developers and can encapsulate remote and local copy operations.

## System Model for Data Warehousing



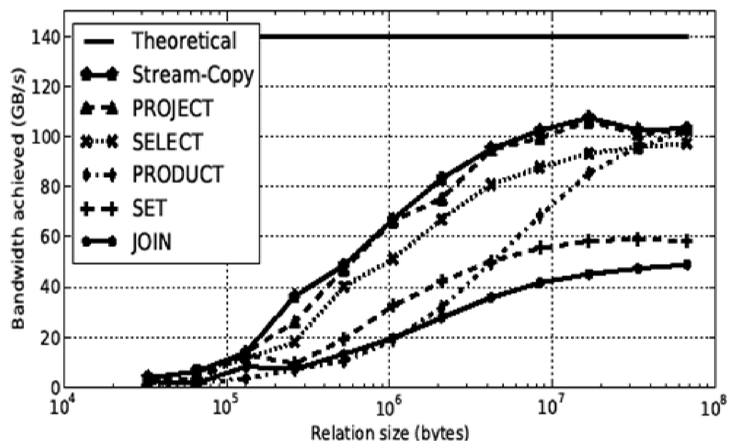

- Red Fox: Compilation and optimization of queries for GPUs [3]
  - Remove need for application developer to optimize applications to run on GPUs
- Oncilla: Global Address Space (GAS) layer built around HT, QPI, 10GE, IB, EXTOLL [4]
  - Create an API to simplify data movement and scheduling
  - Collaborative effort with University of Heidelberg, University of Valencia, AIC, Inc.

## Oncilla: Control Path




- OncillaMem library asks daemon for free memory
  - Local allocations are handled through the local library
- Daemon checks with a master node for free memory and then coordinates remote allocation on each node
- Local library uses returned descriptor to perform remote transfers

## Preliminary Results – GPU Primitives




- Multi-stage algorithm for GPU primitives run on NVIDIA C2050
- Simple primitives (Project, Select, Product) are close to maximum performance (fastest known for GPU)

## 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. *Kernelweaver: Automatically fusing database primitives for efficient GPU computation.* MICRO, 2012.
[4] H. Fröning, et al. *A case for FPGA based accelerated communication*. ICN, 2010.

For more information on Oncilla:
-S. Yalamanchili et al. *Oncilla - Optimizing accelerator clouds for data warehousing applications*. 2012.
-Oncilla project website at:
http://gpuocelot.gatech.edu/projects/compiler-projects/

Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University

Georgia Tech intel PRINCETON UNIVERSITY UC Berkeley