Motivation: Big Data and Accelerators

- Big Data workloads can vary in size from a few TB/month up to PB/day
  - Accelerators like GPUs and Phi are playing a bigger role in processing Big Data
  - GPUs can provide significant speedups for applications like data warehousing [1]
  - However, existing resource aggregation techniques either ignore accelerators or are focused on HPC applications

System Model for Data Warehousing

- Current work focuses on data warehousing and data movement for large data sets between in-core (in-memory) data sets and accelerators
  - The Red Fox runtime takes data warehousing queries and translates them into optimized accelerator and CPU kernels using CUDA and OpenCL [2]
- Oncilla (on-see-yuh) provides high-performance memory aggregation and data movement using a “managed” Global Address Space (GAS) that allow for the aggregation of multiple memory chunks via existing interconnects like IB, Ethernet, and custom fabrics like EXTOLL [3]
  - Oncilla consists of a runtime for allocation and a library to provide high-performance remote memory aggregation and data movement

Control Path vs. Data Path

- Control path uses POSIX and TCP/IP messages to request remote allocations throughout the runtime
  - Oncilla library keeps track of local allocations while runtime daemon tracks system-wide allocations
  - Oncilla allocation overhead is consistent for two node case – 1.2 µs
- Data path relies on high-performance networking to move data between remote and local memory and accelerators
  - Oncilla API abstracts the semantics of underlying network stacks, allowing for portability and improved programmability

Oncilla Two-Node Networking Evaluation

- The Oncilla API allows for easy characterization of a two-node system with QDR InfiniBand and EXTOLL RMA
  - EXTOLL client setup/teardown outperforms IB at sizes up to 32 MB - 242 µs up to 95.47 ms for setup (64 B – 4 GB)
  - IB scales better for larger sizes of allocation and deallocation up to 4 GB - 1.3 ms up to 40.8 ms for setup (64 B – 4 GB)
  - EXTOLL bandwidth maxes out at 7.8 Gbps vs. ~20 Gbps for IB

TPC-H Micro-Benchmarks Evaluation

- TPC-H micro-benchmarks represent common patterns in the 22 TPC-H queries
  - IB cluster with Oncilla is 81% faster (80.9 s → 44.2 s)
  - Join on GPU takes up to 74 seconds!
  - EXTOLL cluster is 22% faster

Experimental Setup

Graph Applications and Future Work

- Oncilla two-node version of SHOC BFS [4] is 2x slower than single-node version
  - Most of the overhead is tied up in allocation and transfer due to small input size
  - Oncilla transfer performance matches MPI-1 implementation
  - Future work is focused on building and evaluating a true multi-node BFS algorithm with accelerator support
    - Large, in-core data sets are more suitable for use with the Oncilla runtime
    - Additional work will focus on integration with OpenCL memory management and runtimes like SnuCL [5]

References