



Big Data Performance and Benchmarking

Sreev Doddabalapur and Eyal Gutkind

WBDB Big Data Workshop (Poona) | December 2012



Why Better Network Technology is Required?

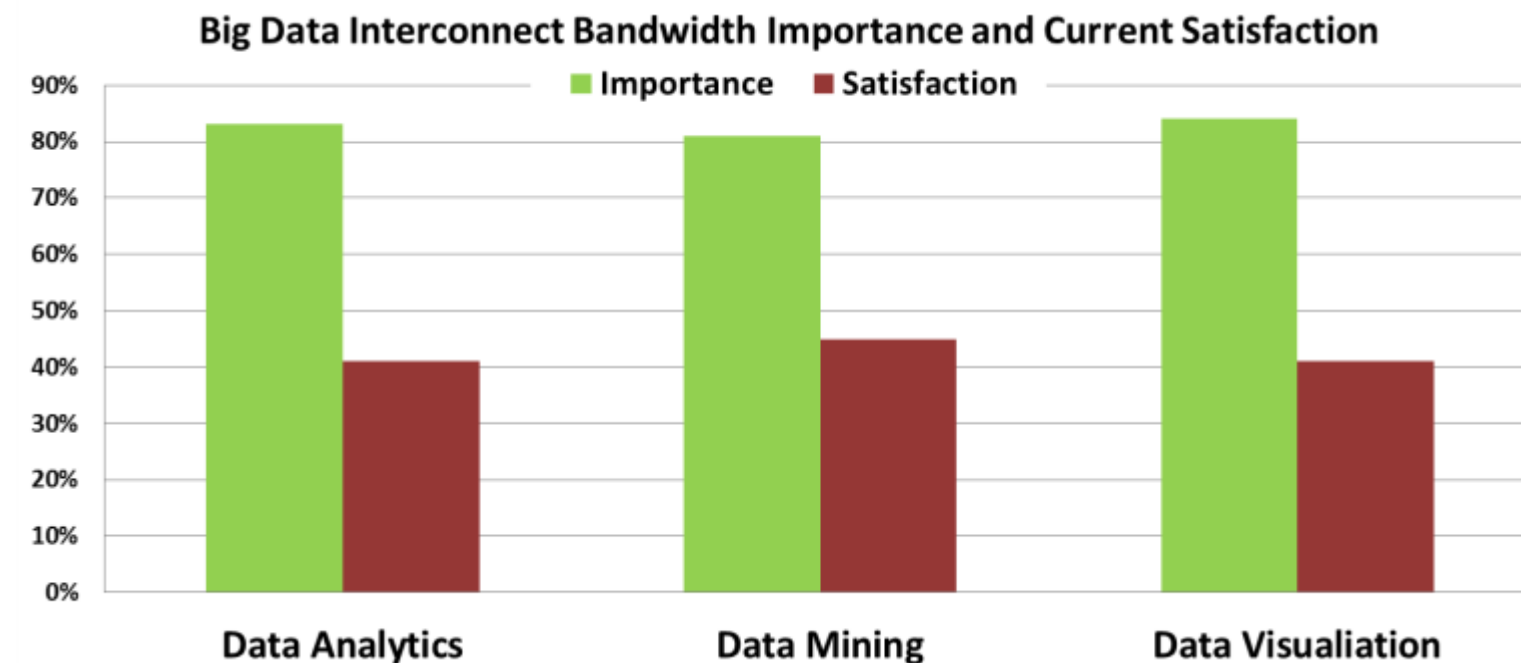
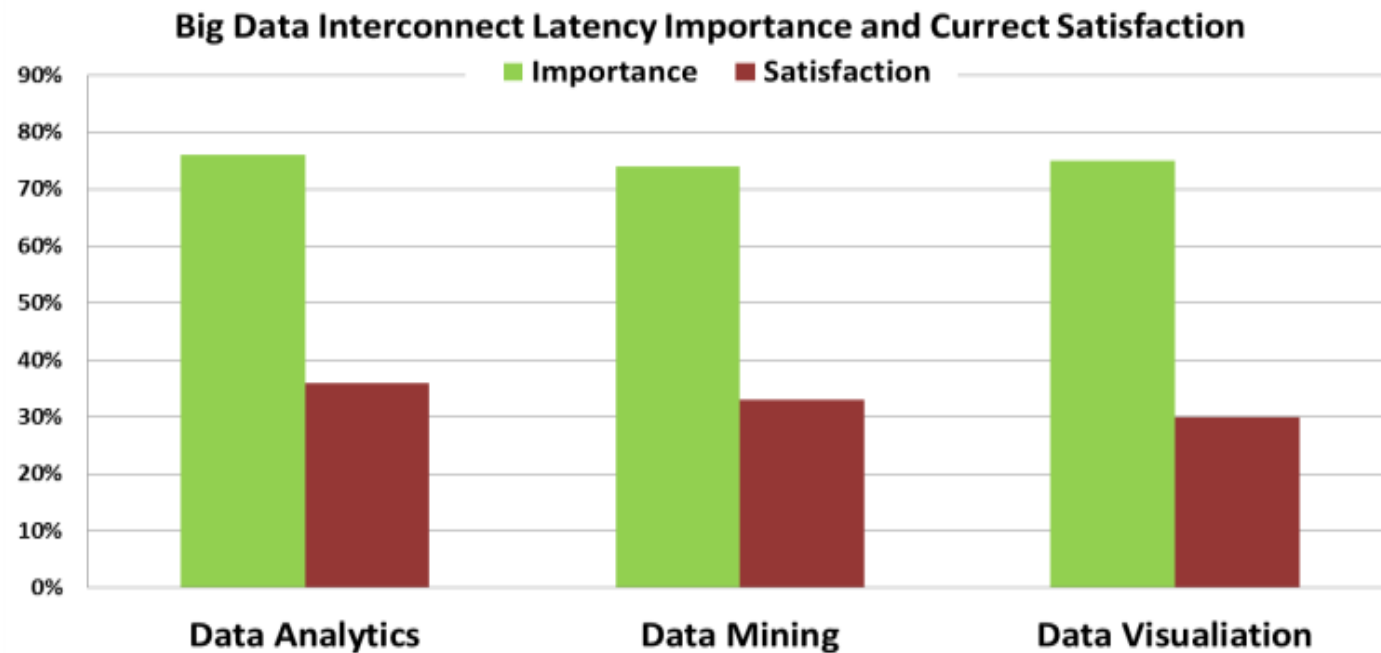
- Servers drive data at 50x faster than 1GbE or 5x faster than 10GbE
 - PCIe 3 Bus @ 52Gbps
- Single Hard Drive Throughput is 170MB/s
 - Systems with 4 and more HDDs are ubiquitous
- Solid State Drive
 - PCI Bus data rates at ~27Gbps per card!
 - Single SAS SSD provides 6Gbps throughput
- Using Lower Grade Network Introduce Bottlenecks
 - Resulting in higher purchasing and operational Cost!



Efficient Scalability is Imperative

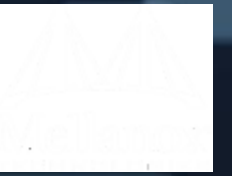
Big Data Pain Points

- Capabilities are Determined by the weakest component in the system
- Different approaches in Big Data marketplace – Same needs
 - Better Throughput and Latency
 - Scalable, Faster data Movement



Big Data Applications Require High Bandwidth and Low Latency Interconnect

* Data Source: Intersect360 Research, 2012, IT and Data scientists survey



What do current implementations have ?

- **HDFS Implementation**

- Sequential I/O operations
- Not suited for Low latency requirements

- **MapReduce Implementation**

- More disk I/O dependency
- Memory usage and tuning requires a constant change for different workloads

Big Data Performance and Scalability ?

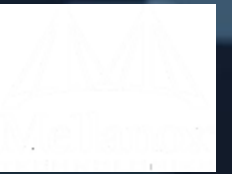
What's driving the change ?

- More knowledge of cluster capabilities
- Use cases coming closer to real-time requirements
- Data center footprint is growing at a phenomenal rate
- The “do more with less becoming the norm”
- Changing workloads becoming a challenge
- More departments getting added to the cluster

A Day in the Life of Databert



Big Data Performance and Scalability ?



How are we addressing this ?

- Simplification – Reduce complexity get more work done

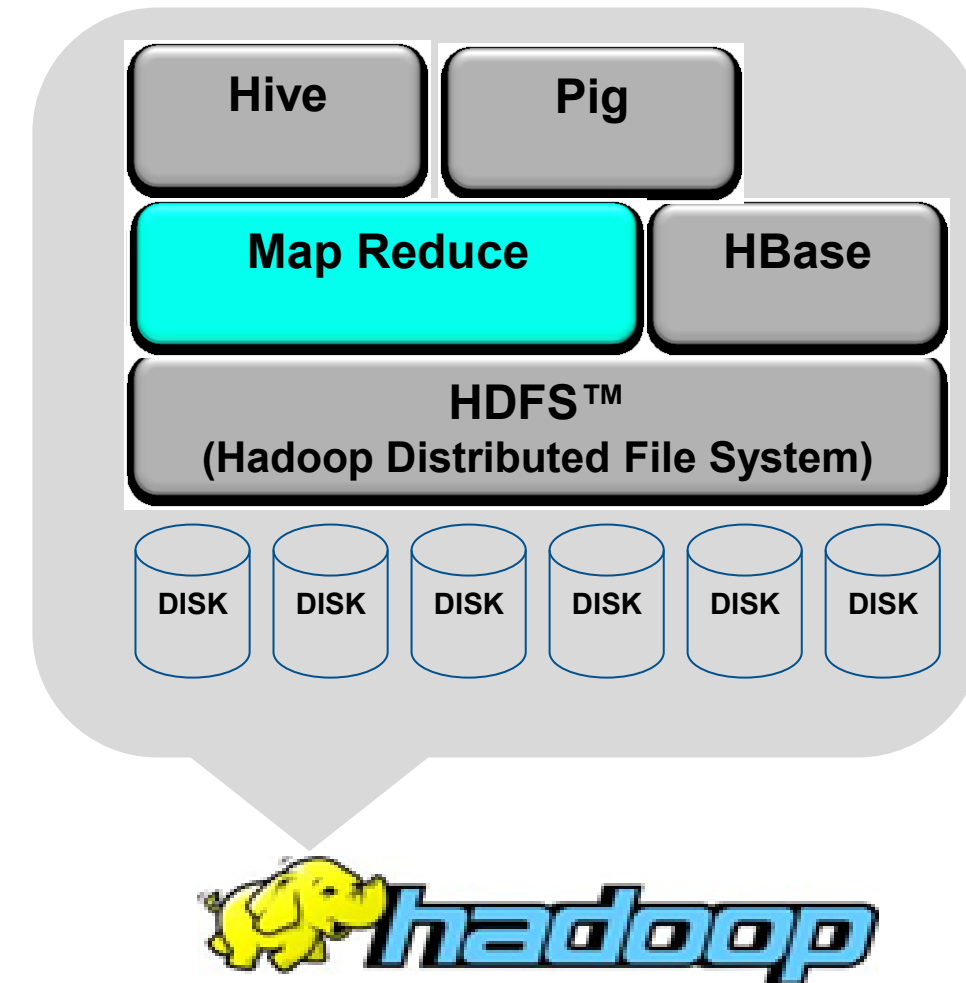
- HDFS Implementation
 - Move Sequential I/O operations to parallel
 - Use RDMA to speed up the HDFS acceleration

- MapReduce Implementation
 - Reduce / eliminate disk I/O dependency
 - Making memory tuning a thing of the past by maximizing the memory footprint of the cluster

- Driving workload balance for appliances

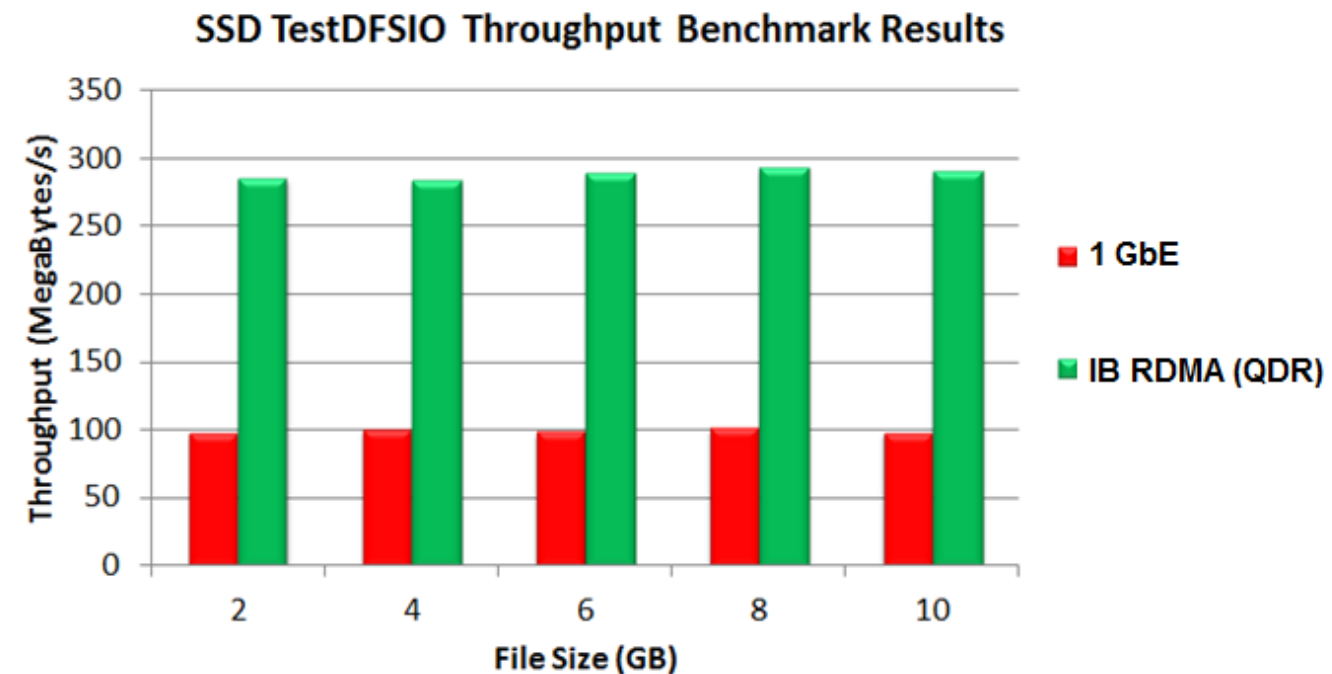
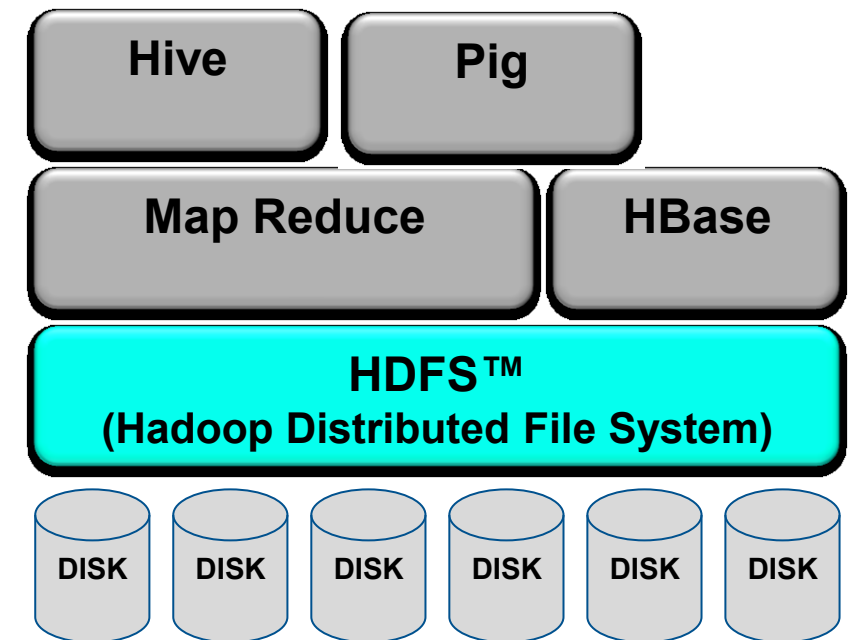
Unstructured Data Accelerator - UDA

- Plug-in architecture
 - Open-source
- Accelerates Map Reduce Jobs
 - Accelerated merge sort
- Efficient Shuffle Provider
 - Data transfer over RDMA
 - Supports InfiniBand and Ethernet
- Supported Hadoop Distributions
 - Apache Hadoop 1.0.x ; 1.1.x, 3.0.0
 - Cloudera Distribution Hadoop 3 update 4 (CDH3u4)
 - Hortonworks HDP 1.1
- Supported Hardware
 - ConnectX®-3 VPI
 - SwitchX-2 based systems
- Supported Operating Systems
 - MLNX_OFED 1.5.3 Linux based supported Operating Systems

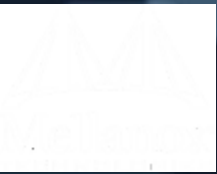


HDFS Acceleration; Joint Project With Ohio State University

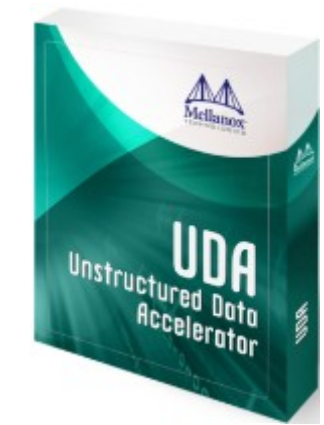
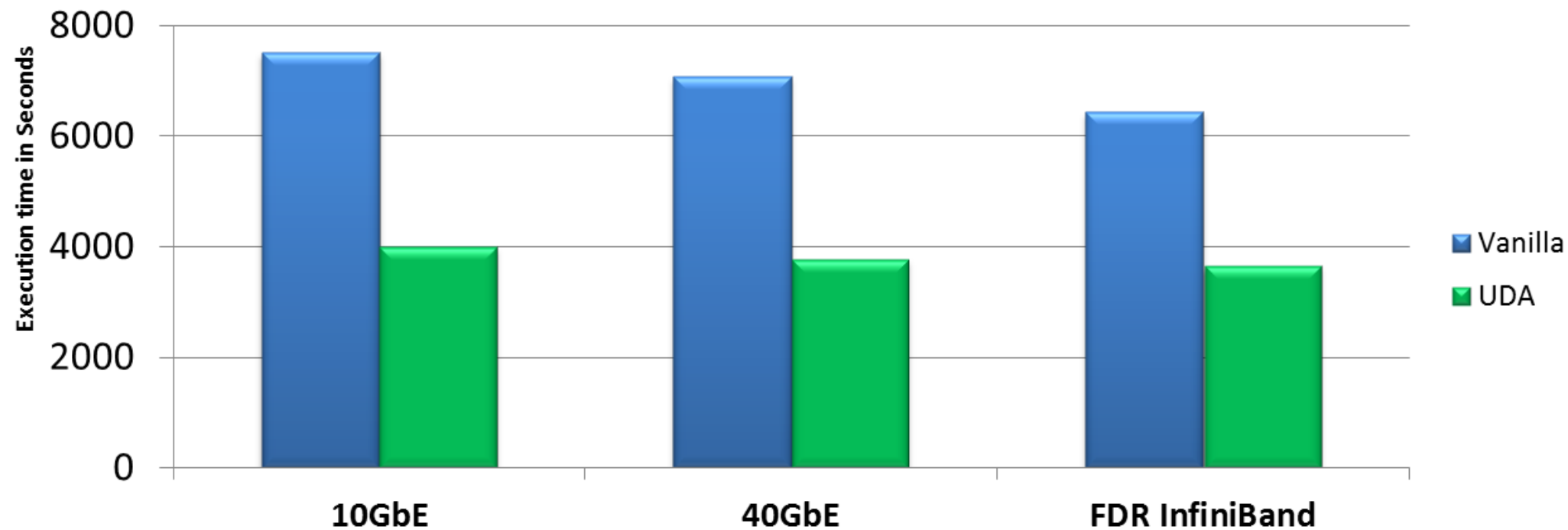
- HDFS is the Hadoop File System
 - The underlying File system for HBase and other NoSQL Data Bases
- More Drives, Higher Throughput is Needed
- SSDs Solutions Must use Higher Throughput
 - Bounded by 1GbE and 10GbE



Double Hadoop Performance with UDA



TeraSort^{*} Results Comparing Over Different Fabrics^{}**



Disk Writes

40%

Disk Reads

15%

CPU Efficiency

2.5X

*TeraSort is a popular benchmark used to measure the performance of Hadoop cluster

**1TB Data Set, 5x E5-2680 Machines, 4x HDD Base; Vanilla → Apache Hadoop 1.0.3; UDA → Apache Hadoop 1.0.3+UDA

~2X Faster Job Completion! Increase the Value of Data!

Key Benefits

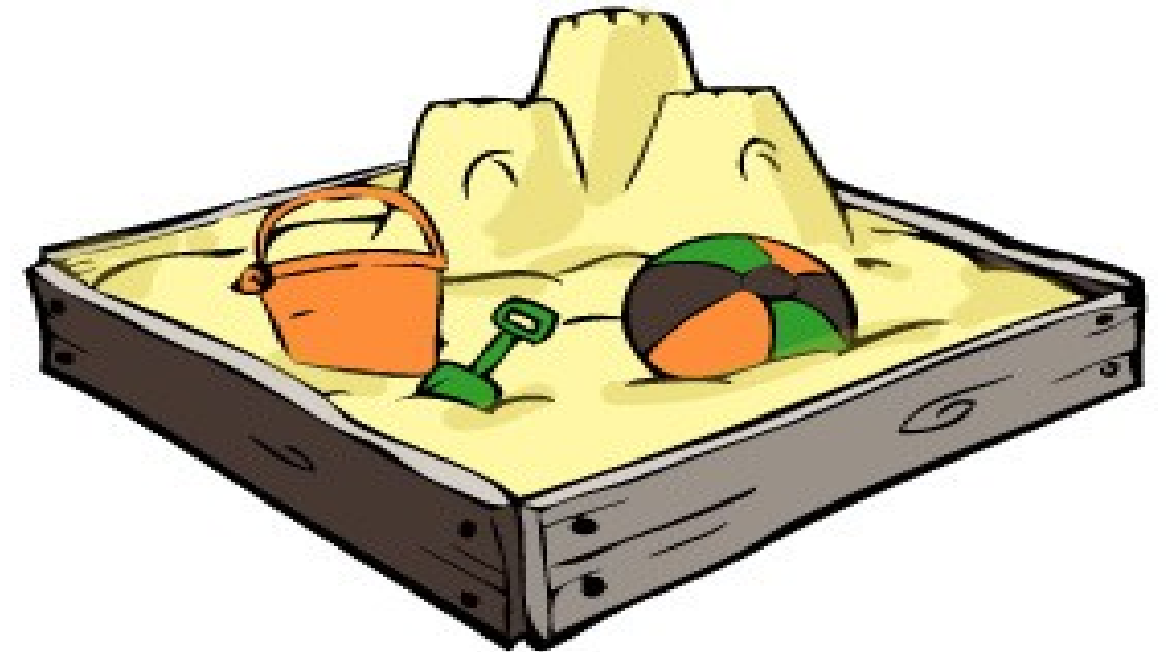
- Performance
- Fully maximize RDMA features for Hadoop
 - Helps in the Map Reduce part of the framework
- Simplification for varying work loads
- Achieve twice the performance
- Large / Complex datasets
- You need close to native performance of the cluster

Mellanox Sandbox, Invite your Customer to Use

Several clusters to try and learn about RDMA and Big Data

- ✓ Sandy Bridge Systems
- ✓ ConnectX®-3 adapter cards
- ✓ InfiniBand and 10Gb/40Gb Ethernet Setups
- ✓ Available now!

Email bigdata@mellanox.com for more information



Thank You

Email : bigdata@mellanox.com

