Can High-Performance Interconnects Benefit Memcached and Hadoop?

D. K. Panda and Sayantan Sur

Network-Based Computing Laboratory
Department of Computer Science and Engineering
The Ohio State University, USA

OpenFabrics Monterey Workshop (April '11)

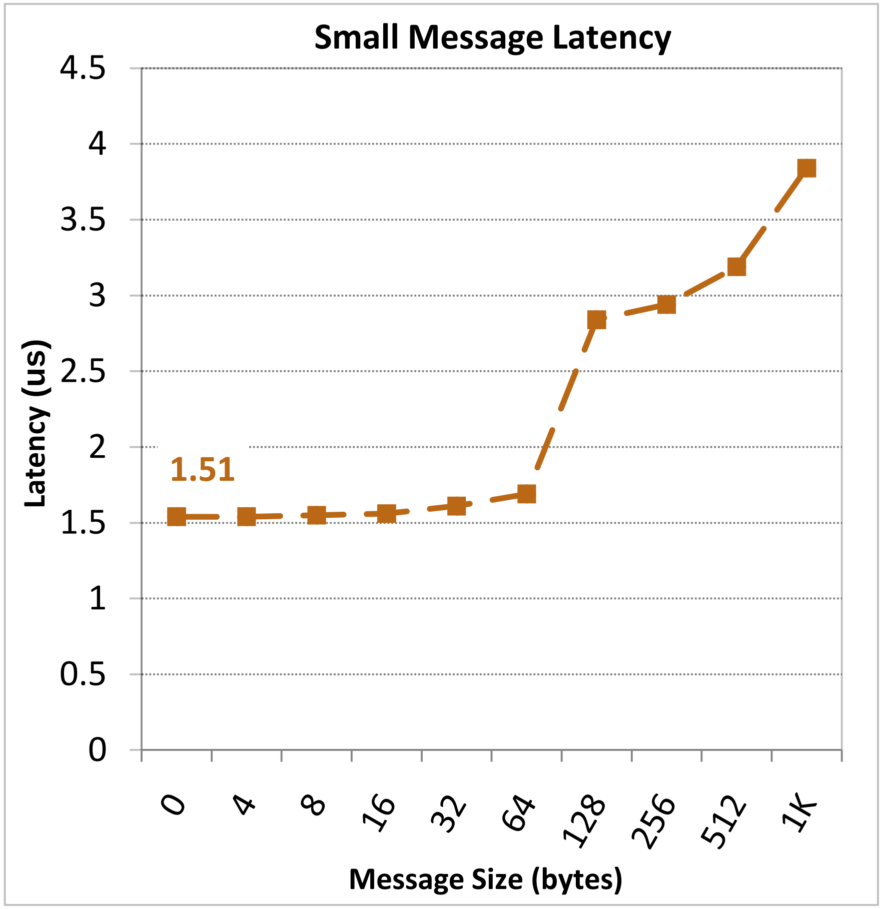
Outline

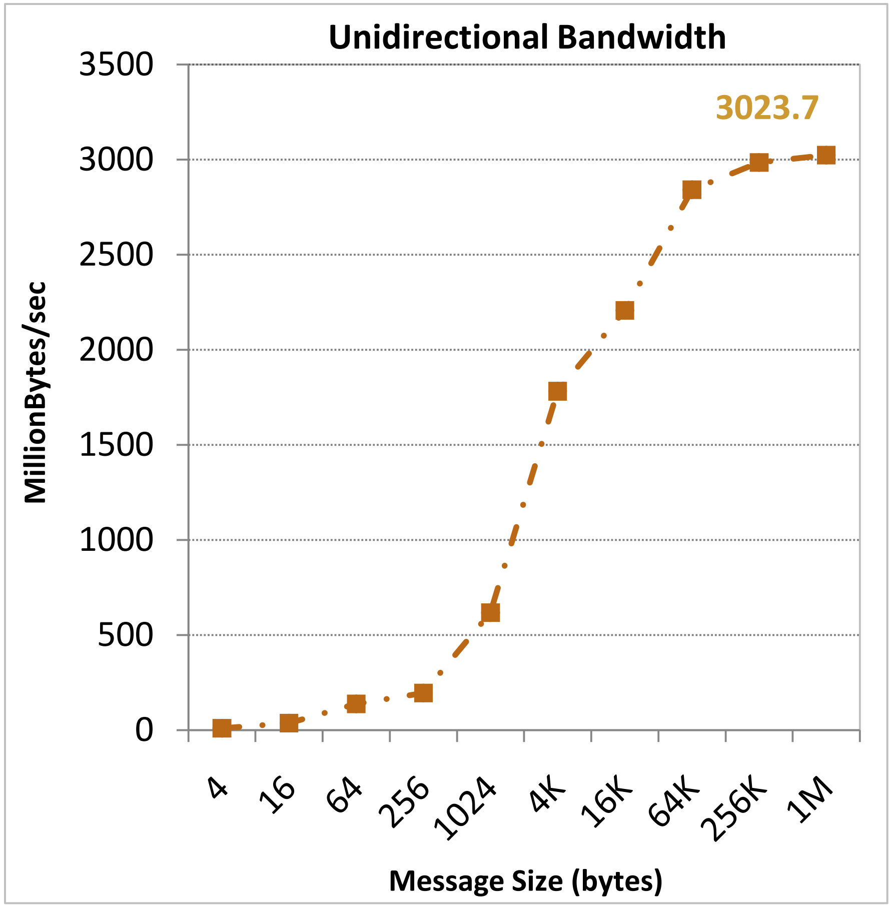
- Introduction
- Overview of Memcached and Hadoop
- A new approach towards OFA in Cloud
- Experimental Results & Analysis
- Summary

Introduction

- High-Performance Computing (HPC) has adopted advanced interconnects (e.g. InfiniBand, 10 Gigabit Ethernet)
  - Low latency (few micro seconds), High Bandwidth (40 Gb/s)
  - Low CPU overhead
- OpenFabrics has been quite successful in the HPC domain
- Many machines in Top500 list
- Beginning to draw interest from the enterprise domain
  - Google keynote shows interest in IB for improving RPC cost
  - Oracle has used IB in Exadata
- Performance in the enterprise domain remains a concern
  - Google keynote also highlighted this

MPI (MVAPICH2) Performance over IB



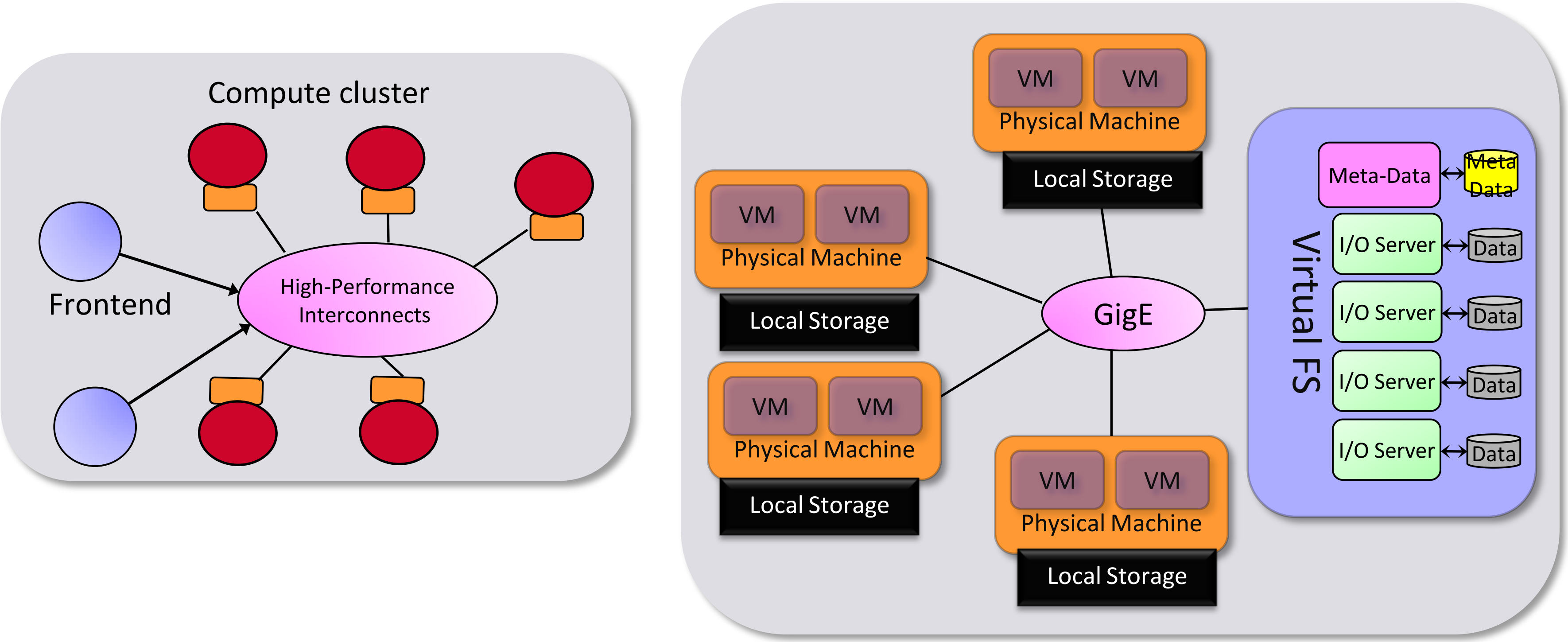


2.4 GHz Quad-core (Nehalem) Intel serves with Mellanox ConnectX-2 QDR adapters and IB switch

Software Ecosystem in Cloud and Upcoming Challenges

- Memcached – scalable distributed caching
  - Widely adopted caching frontend to MySQL and other DBs
- MapReduce – scalable model to process Petabytes of data
  - Hadoop MapReduce framework widely adopted
- Both Memcached and Hadoop designed with Sockets
  - Sockets API itself was designed decades ago
- At the same time SSDs have improved I/O characteristics
  - Google keynote also highlighted that I/O costs are coming down a lot
  - Communication cost will dominate in the future
- Can OFA help cloud computing software performance?

Typical HPC and Cloud Computing Deployments

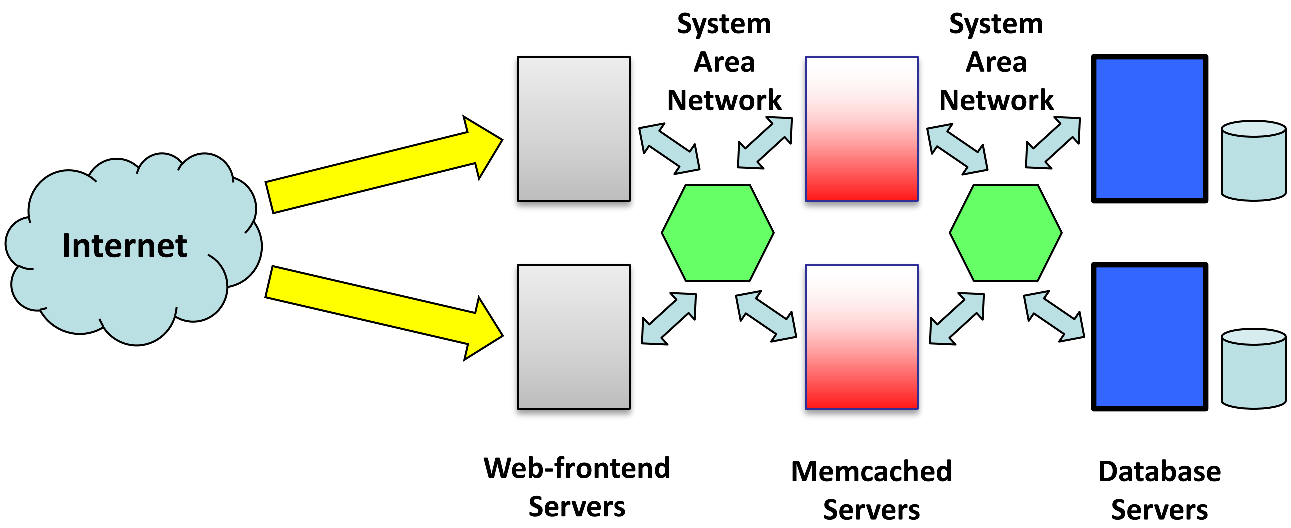


- HPC system design is interconnect centric
- Cloud computing environment has complex software and historically relied on Sockets and Ethernet

Outline

- Introduction
- Overview of Memcached and Hadoop
- A new approach towards OFA in Cloud
- Experimental Results & Analysis
- Summary

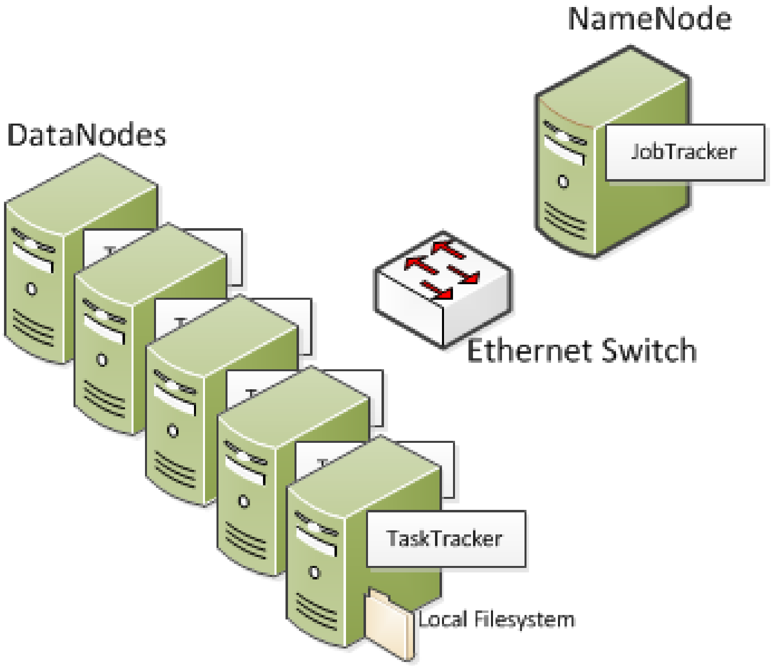
Memcached Architecture



- Distributed Caching Layer
  - Allows to aggregate spare memory from multiple nodes
  - General purpose
- Typically used to cache database queries, results of API calls
- Scalable model, but typical usage very network intensive

Hadoop Architecture

- Underlying Hadoop Distributed File System (HDFS)
- Fault-tolerance by replicating data blocks
- NameNode: stores information on data blocks
- DataNodes: store blocks and host Map-reduce computation
- JobTracker: track jobs and detect failure
- Model scales but high amount of communication during intermediate phases



Outline

- Introduction
- Overview of Memcached and Hadoop
- A new approach towards OFA in Cloud
- Experimental Results & Analysis
- Summary

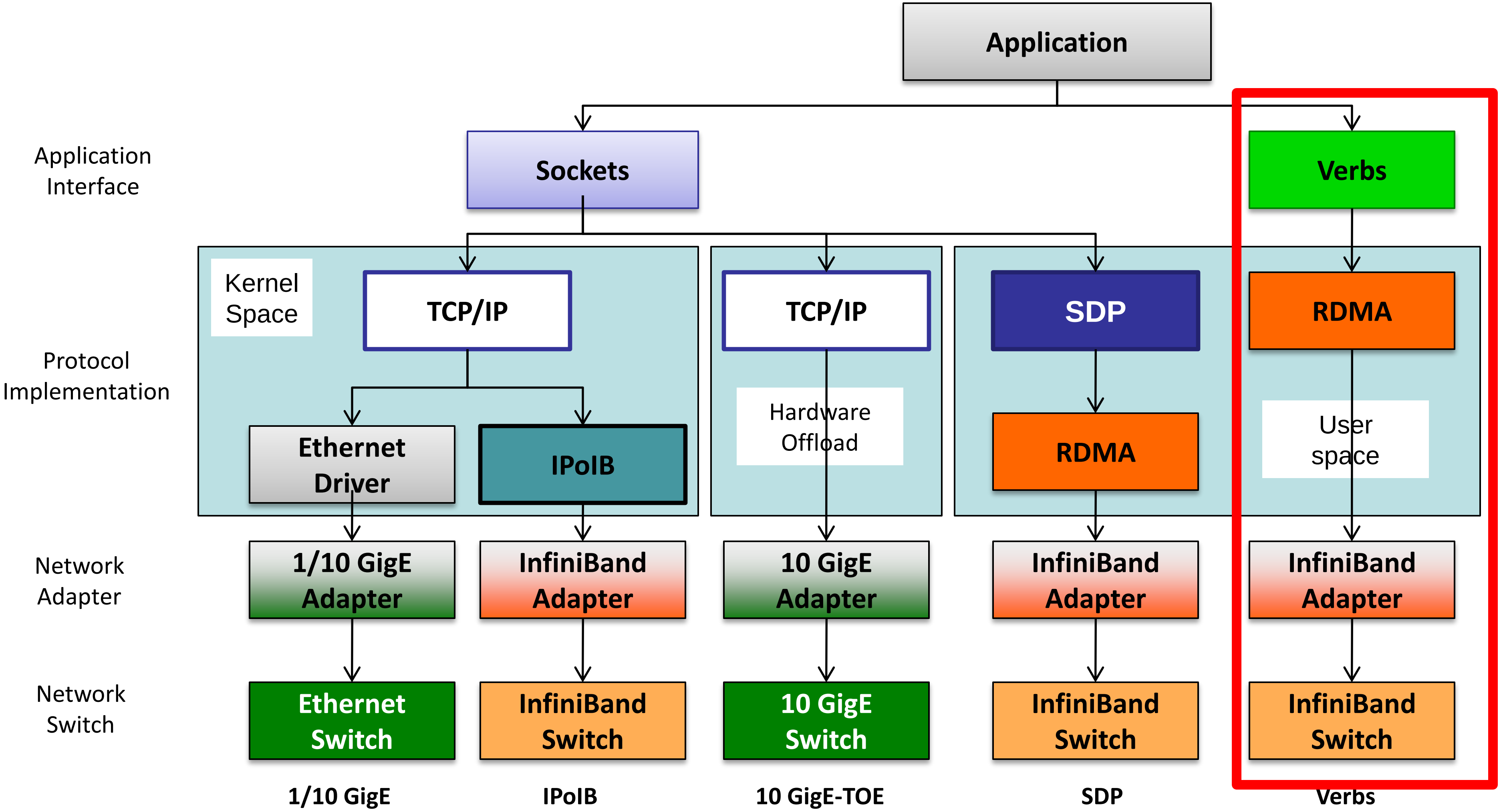
InfiniBand and 10 Gigabit Ethernet

- InfiniBand is an industry standard packet switched network
- Has been increasingly adopted in HPC systems
- User-level networking with OS-bypass (verbs)

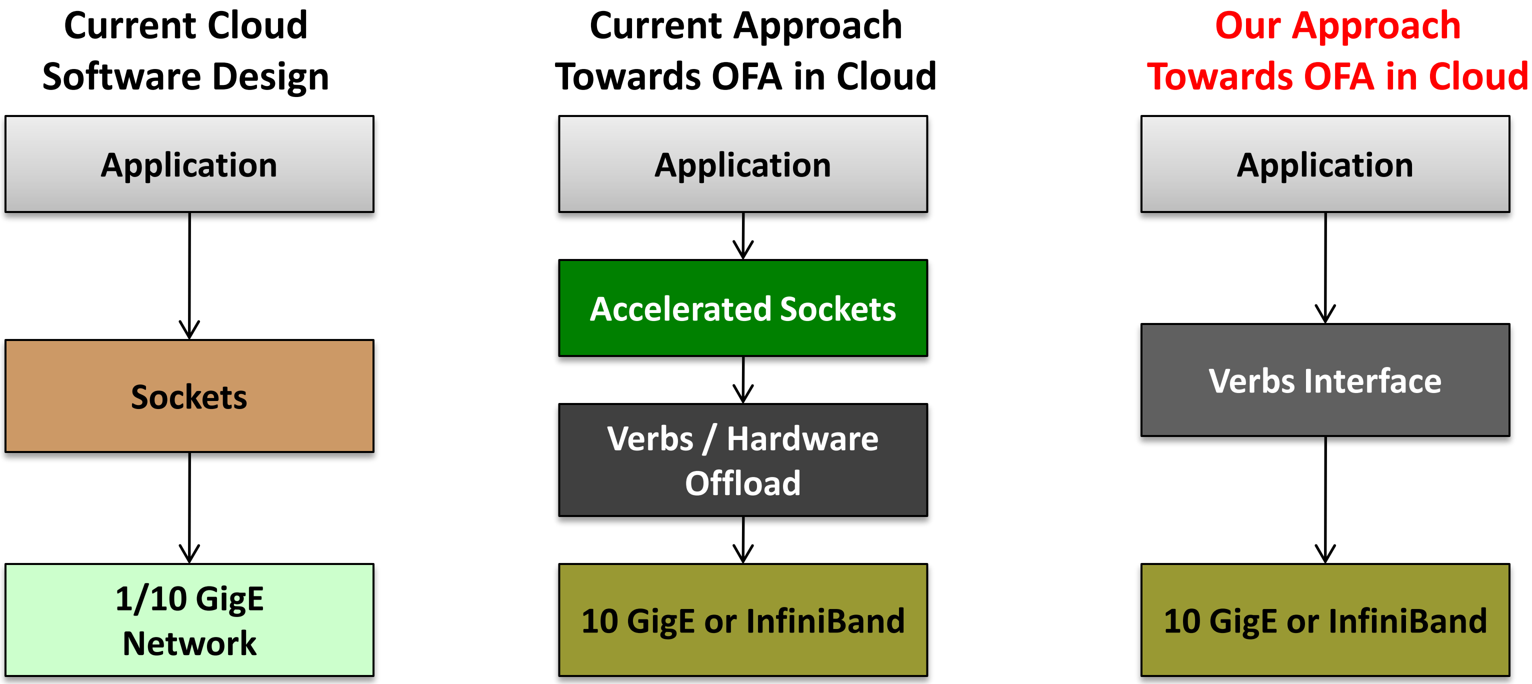
- 10 Gigabit Ethernet follow up to Gigabit Ethernet
- Provides user-level networking with OS-bypass (iWARP)
- Some vendors have accelerated TCP/IP by putting it on the network card (hardware offload)

- **Convergence**: possible to use both through OpenFabrics
  - Same software, different networks

Modern Interconnects and Protocols

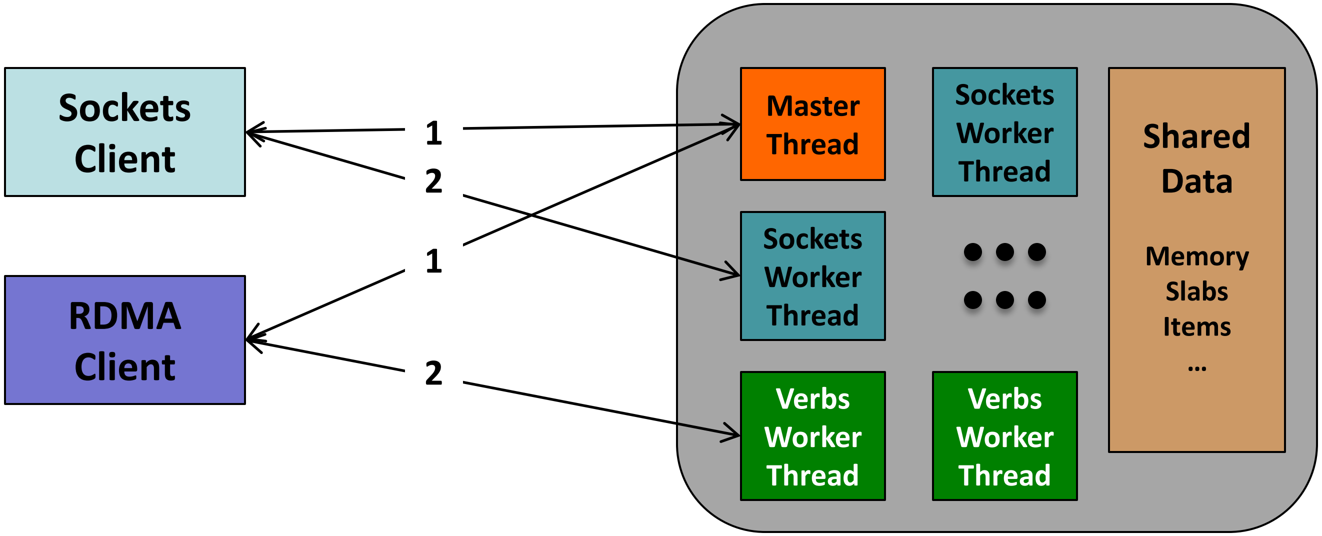


A New Approach towards OFA in Cloud



- Sockets not designed for high-performance
  - Stream semantics often mismatch for upper layers (Memcached, Hadoop)
  - Zero-copy not available for non-blocking sockets (Memcached)
- Significant consolidation in cloud system software
  - Hadoop and Memcached are developer facing APIs, not sockets
  - Improving Hadoop and Memcached will benefit many applications immediately!

Memcached Design Using Verbs



- Server and client perform a negotiation protocol
  - Master thread assigns clients to appropriate worker thread
- Once a client is assigned a verbs worker thread, it can communicate directly and is “bound” to that thread
- All other Memcached data structures are shared among RDMA and Sockets worker threads

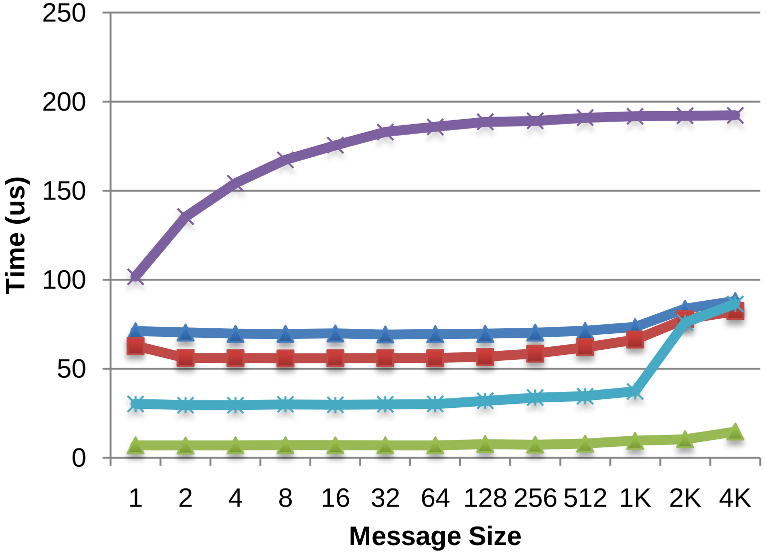
Outline

- Introduction
- Overview of Memcached and Hadoop
- A new approach towards OFA in Cloud
- Experimental Results & Analysis
- Summary

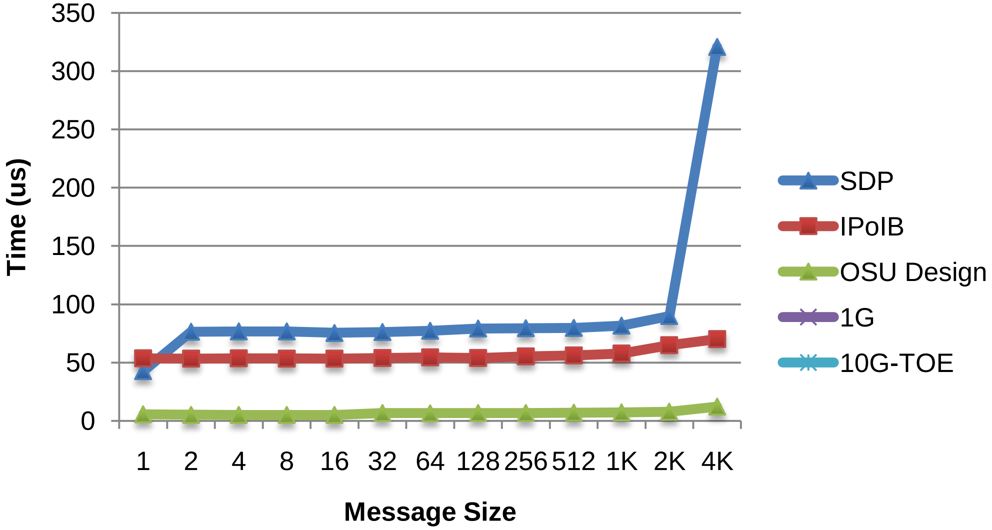
Experimental Setup

- Memcached Experiments
  - Intel Clovertown 2.33GHz, 6GB RAM, InfiniBand DDR, Chelsio T320
  - Intel Westmere 2.67GHz, 12GB RAM, InfiniBand QDR
  - Memcached server: 1.4.5 Memcached Client (libmemcached) 0.45
- Hadoop Experiments
  - Intel Clovertown 2.33GHz, 6GB RAM, InfiniBand DDR, Chelsio T320
  - Intel X-25E 64GB SSD and 250GB HDD
  - Hadoop version 0.20.2, Sun/Oracle Java 1.6.0
  - Dedicated NameServer and JobTracker
  - Number of Datanodes used: 2, 4, and 8
- We used unmodified Hadoop for our experiments
  - OFA used through Sockets

Memcached Get Latency



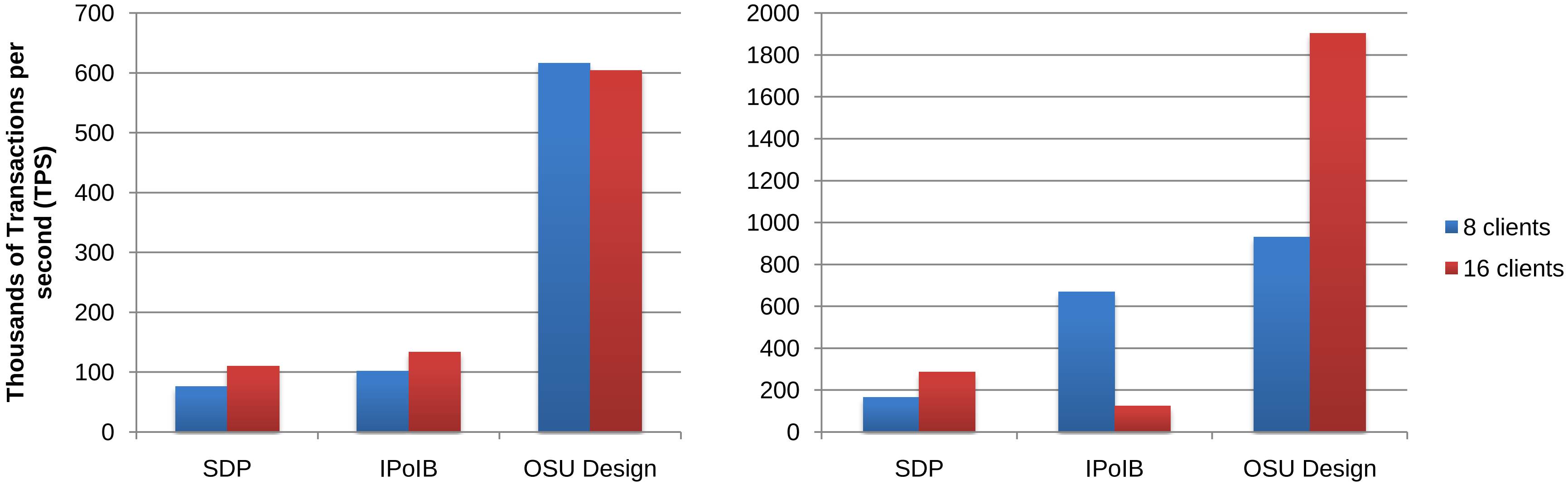
Intel Clovertown Cluster (IB: DDR)



Intel Clovertown Cluster (IB: QDR)

- Memcached Get latency
  - 4 bytes – DDR: 6 us; QDR: 5 us
  - 4K bytes -- DDR: 20 us; QDR:12 us
- Almost factor of *four* improvement over 10GE (TOE) for 4K
- We are in the process of evaluating iWARP on 10GE

Memcached Get TPS

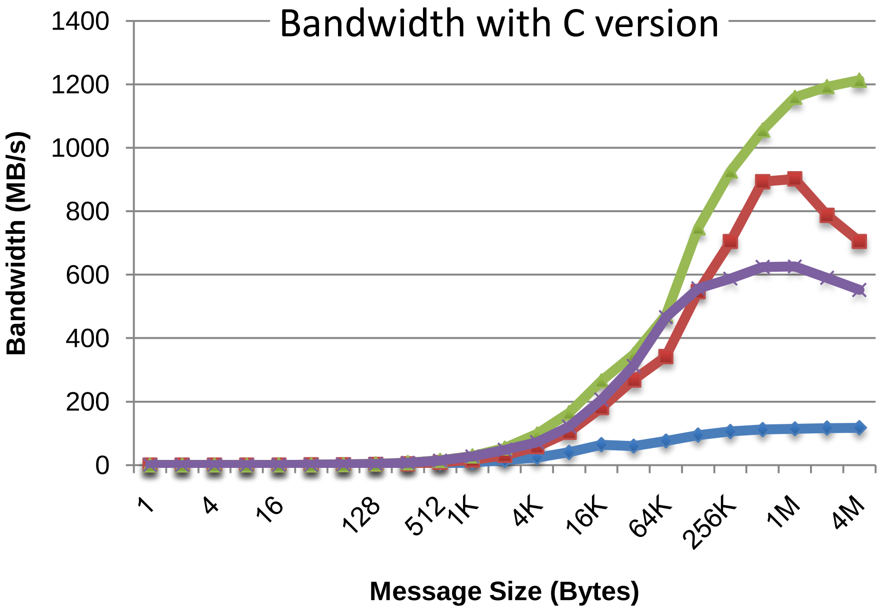


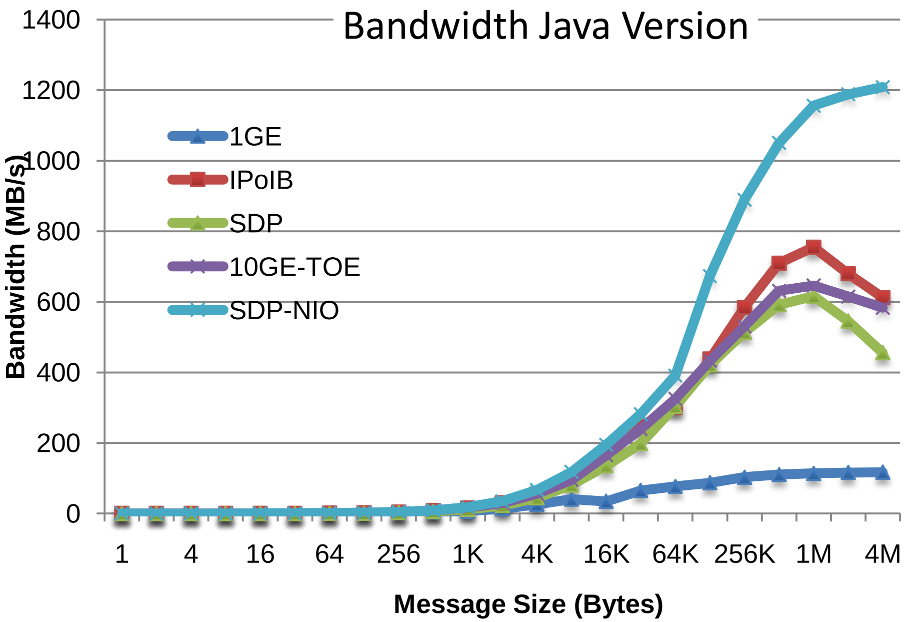
Intel Clovertown Cluster (IB: DDR)

Intel Clovertown Cluster (IB: QDR)

- Memcached Get transactions per second for 4 bytes
  - On IB DDR about 700K/s for 16 clients
  - On IB QDR 1.9M/s for 16 clients
- Almost factor of *six* improvement over SDP
- We are in the process of evaluating iWARP on 10GE

Hadoop: Java Communication Benchmark

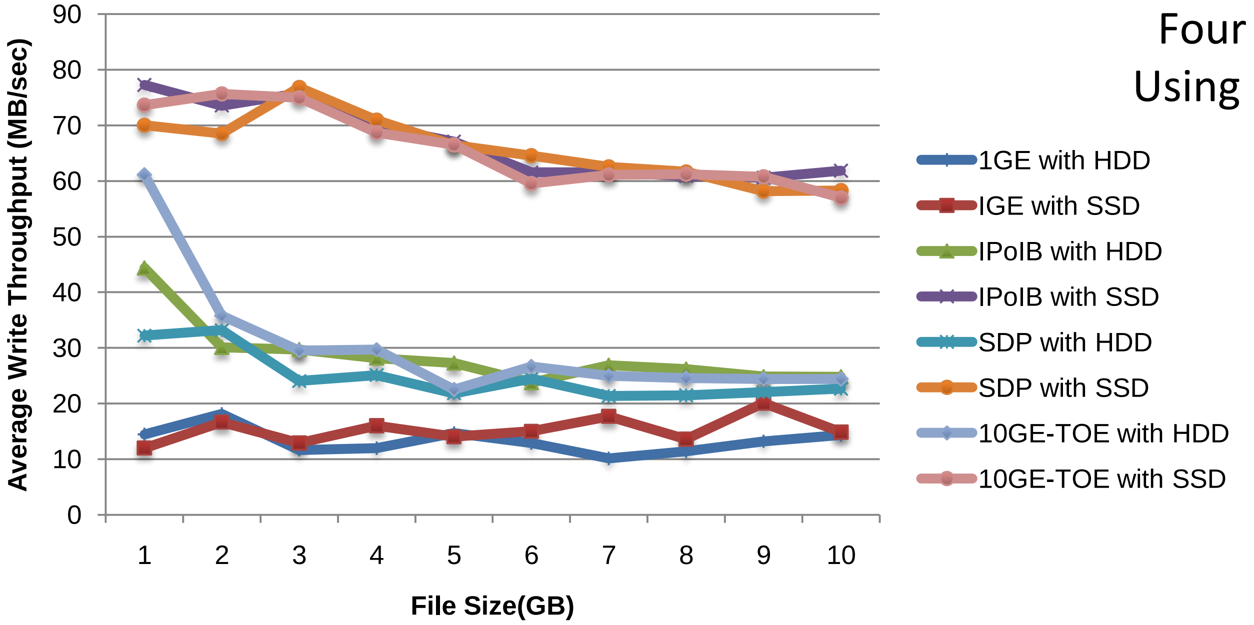




- Sockets level ping-pong bandwidth test
- Java performance depends on usage of NIO (allocateDirect)
- C and Java versions of the benchmark have similar performance
- HDFS does not use direct allocated blocks or NIO on DataNode

S. Sur, H. Wang, J. Huang, X. Ouyang and D. K. Panda ***"Can High-Performance Interconnects Benefit Hadoop Distributed File System?"*****, MASVDC '10 in conjunction with MICRO 2010, Atlanta, GA.**

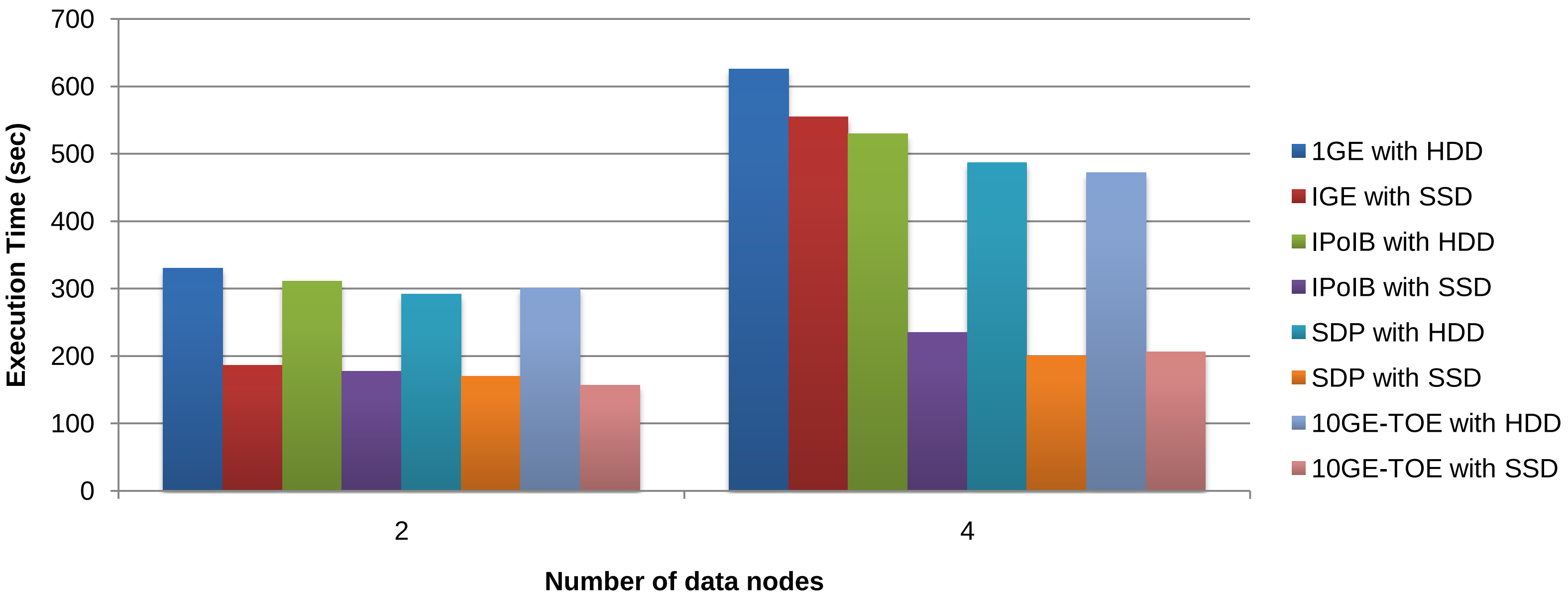
Hadoop: DFS IO Write Performance



Four Data Nodes
Using HDD and SSD

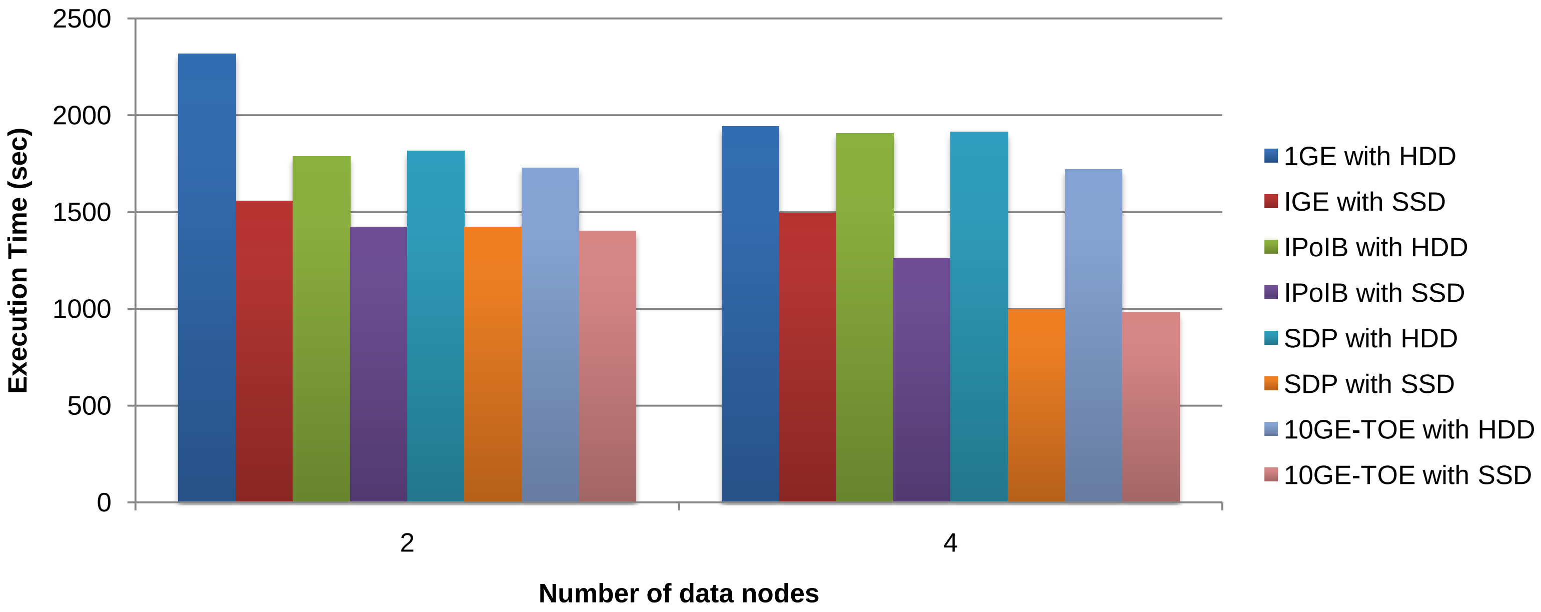
- DFS IO included in Hadoop, measures sequential access throughput
- We have two map tasks each writing to a file of increasing size (1-10GB)
- Significant improvement with IPoIB, SDP and 10GigE
- With SSD, performance improvement is almost seven or eight fold!
- SSD benefits not seen without using high-performance interconnect!
  - In-line with comment on Google keynote about I/O performance

Hadoop: RandomWriter Performance



- Each map generates 1GB of random binary data and writes to HDFS
- SSD improves execution time by 50% with 1GigE for two DataNodes
- For four DataNodes, benefits are observed only with HPC interconnect
- IPoIB, SDP and 10GigE can improve performance by 59% on four DataNodes

Hadoop Sort Benchmark



- Sort: baseline benchmark for Hadoop
- Sort phase: I/O bound; Reduce phase: communication bound
- SSD improves performance by 28% using 1GigE with two DataNodes
- Benefit of 50% on four DataNodes using SDP, IPoIB or 10GigE

Summary

- OpenFabrics has come a long way in HPC adoption
- Facing new frontiers in the Cloud computing domain
- Previous attempts at OpenFabrics adoption in Cloud focused on Sockets
- Even using OpenFabrics through Sockets good gains can be observed
  - 50% faster sorting when OFA used in conjunction with SSDs
- There is a vast performance gap between Sockets and Verbs level performance
  - Factor of four improvement in Memcached get latency (4K bytes)
  - Factor of six improvement in Memcached get transactions/s (4 bytes)
- Native Verbs-level designs will benefit cloud computing domain
- We are currently working on Verbs-level designs of HDFS and Hbase

Thank You!

{panda, surs}@cse.ohio-state.edu

Network-Based Computing Laboratory

http://nowlab.cse.ohio-state.edu/

MVAPICH Web Page

http://mvapich.cse.ohio-state.edu/