MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

# Designing Scalable Communication and I/O Schemes for Accelerating Big Data Processing in the Cloud

**Shashank Gugnani**

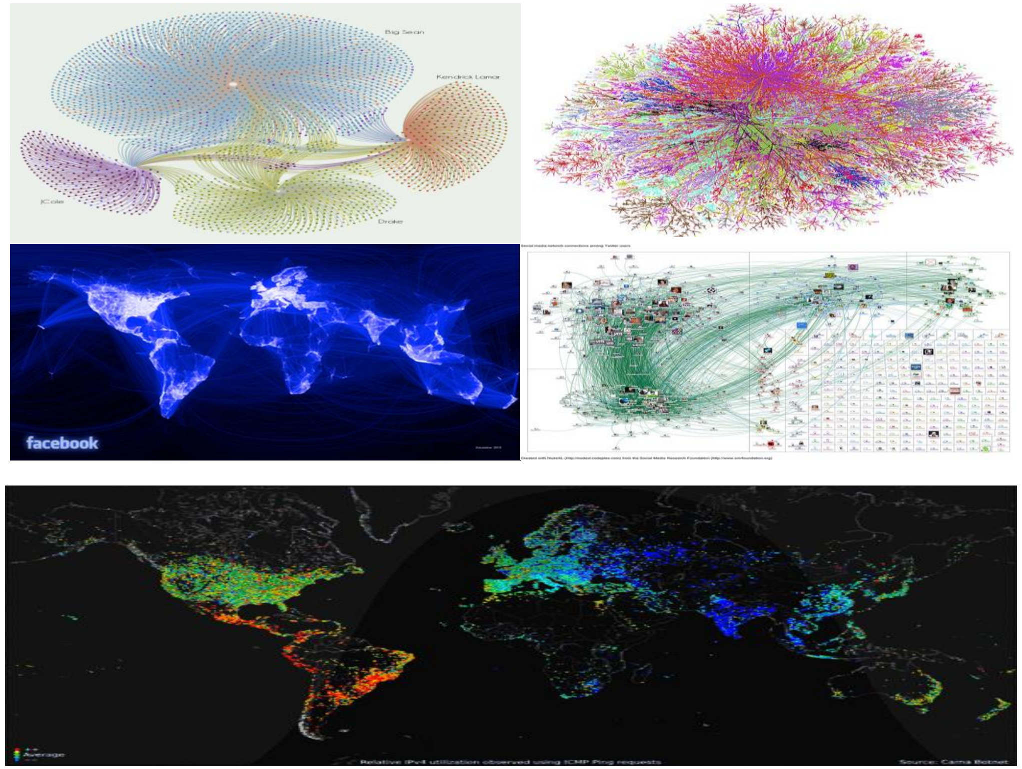
The Ohio State University

E-mail: gugnani.2@osu.edu

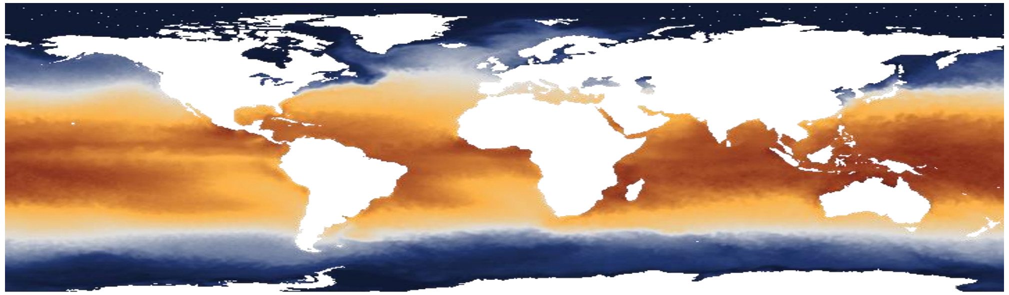
http://web.cse.ohio-state.edu/~gugnani/

# Introduction to Big Data Analytics and Trends

- Big Data has changed the way people understand and harness the power of data, both in the business and research domains
- Big Data has become one of the most important elements in business analytics
- Big Data and High Performance Computing (HPC) are converging to meet large scale data processing challenges
- Running High Performance Data Analysis (HPDA) workloads in the cloud is gaining popularity
  - According to the latest OpenStack survey, 27% of cloud deployments are running HPDA workloads




http://www.coolinfographics.com/blog/tag/data?currentPage=3



http://www.climatecentral.org/news/white-house-brings-together-big-data-and-climate-change-17194

# Drivers of Modern HPC Cloud Architectures


Multi-core Processors

High Performance Interconnects – InfiniBand (with SR-IOV)
<1usec latency, 200Gbps Bandwidth>

SSDs, Object Storage Clusters

Large memory nodes
(Upto 2 TB)

- Multi-core/many-core technologies
- Large memory nodes
- Remote Direct Memory Access (RDMA)-enabled networking (InfiniBand and RoCE)
- Single Root I/O Virtualization (SR-IOV)
- Solid State Drives (SSDs), Object Storage Clusters

Microsoft Azure | amazon web services EC2 | Grid'5000 | ORACLE Cloud | Chameleon Cloud

# Summary of HPC Cloud Resources

- High-Performance Cloud systems have adopted advanced interconnects and protocols
  - InfiniBand, 40 Gigabit Ethernet/iWARP, RDMA over Converged Enhanced Ethernet (RoCE)
  - Low latency (few micro seconds), High Bandwidth (200 Gb/s with HDR InfiniBand)
  - SR-IOV for hardware-based I/O virtualization
- Vast installations of Object Storage systems (e.g. Swift, Ceph)
  - Total capacity is in the PB range
  - Offer high availability and fault-tolerance
  - Performance and scalability is still a problem
- Large memory per node for in-memory processing

# Big Data in the Cloud: Challenges and Opportunities

- Scalability requirements significantly increased
- Explosion of data
- How do we handle huge amounts of this data?
- Requirement for more efficient and faster processing of Data
- Advancements in computing technology
  - RDMA, SR-IOV, byte addressable NVM, NVMe
- *How can we leverage the advanced hardware?*

# Our Goal

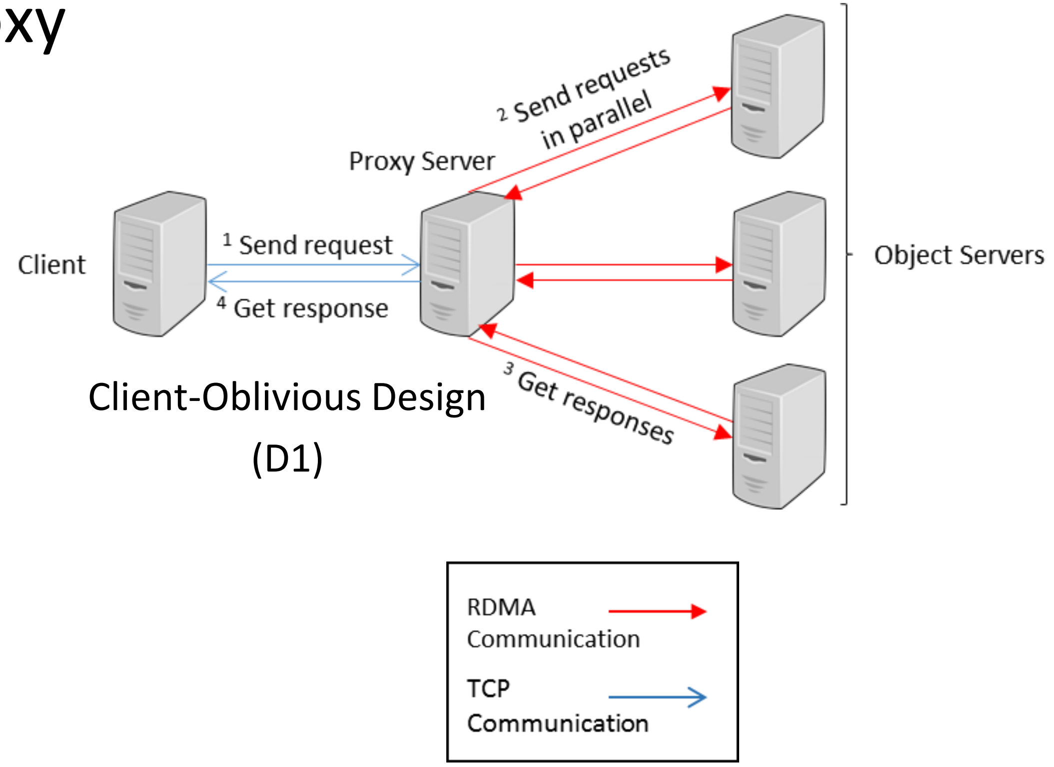
- Scalable Cloud Storage
  - Quality of Service and Consistency paramount
  - Need for newer algorithms and protocols
- Performant Communication Middleware
  - Use high-performance networking
  - Topology-aware communication

# Scalable Cloud Storage

- Re-designed Swift architecture for improved scalability and performance
- Two proposed designs:
  - **Client-Oblivious Design**: No changes required on the client side
  - **Metadata Server-based Design**: Direct communication between client and object servers; bypass proxy server
- RDMA-based communication framework for accelerating networking performance
- High-performance I/O framework to provide maximum overlap between communication and I/O
- New consistency model to enable legacy applications to run on cloud storage

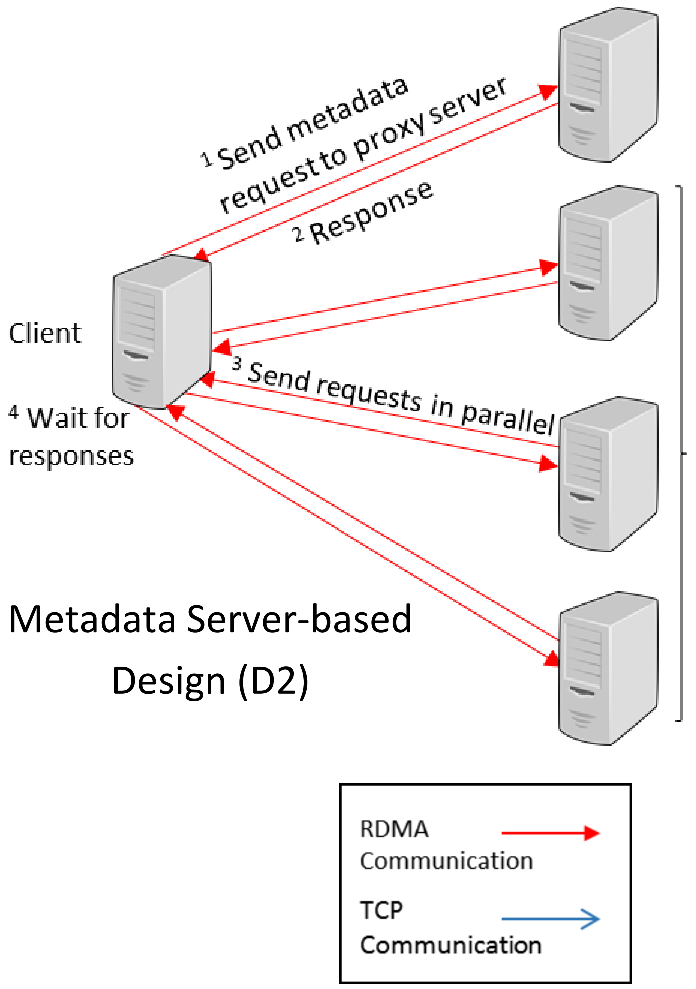
# Client-Oblivious Design

- No change required on the client side
- Communication between client and proxy server using conventional TCP sockets networking
- Communication between proxy server using high-performance RDMA-based networking
- Proxy Server is still the bottleneck!

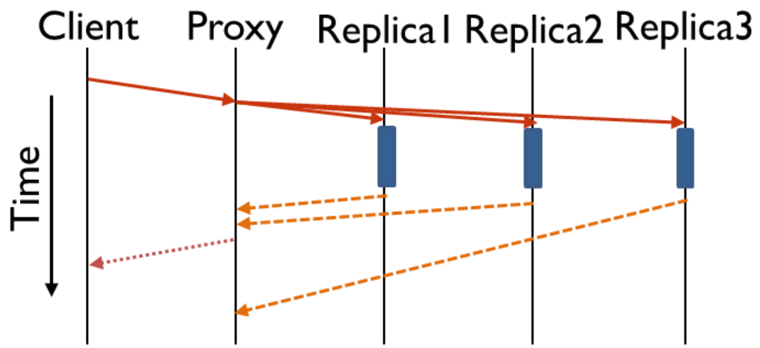


# Metadata Server-based Design

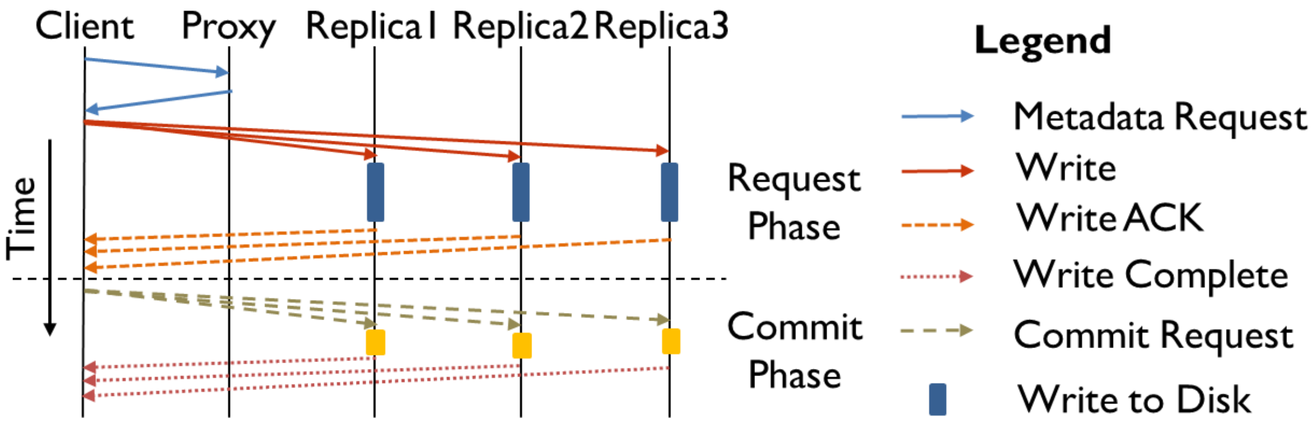
- Re-designed architecture for improved scalability
- Client-based replication for reduced latency and high-performance
- All communication using high-performance RDMA-based networking
- Proxy Server no longer the bottleneck!



# POSIX-like Consistent Cloud Storage



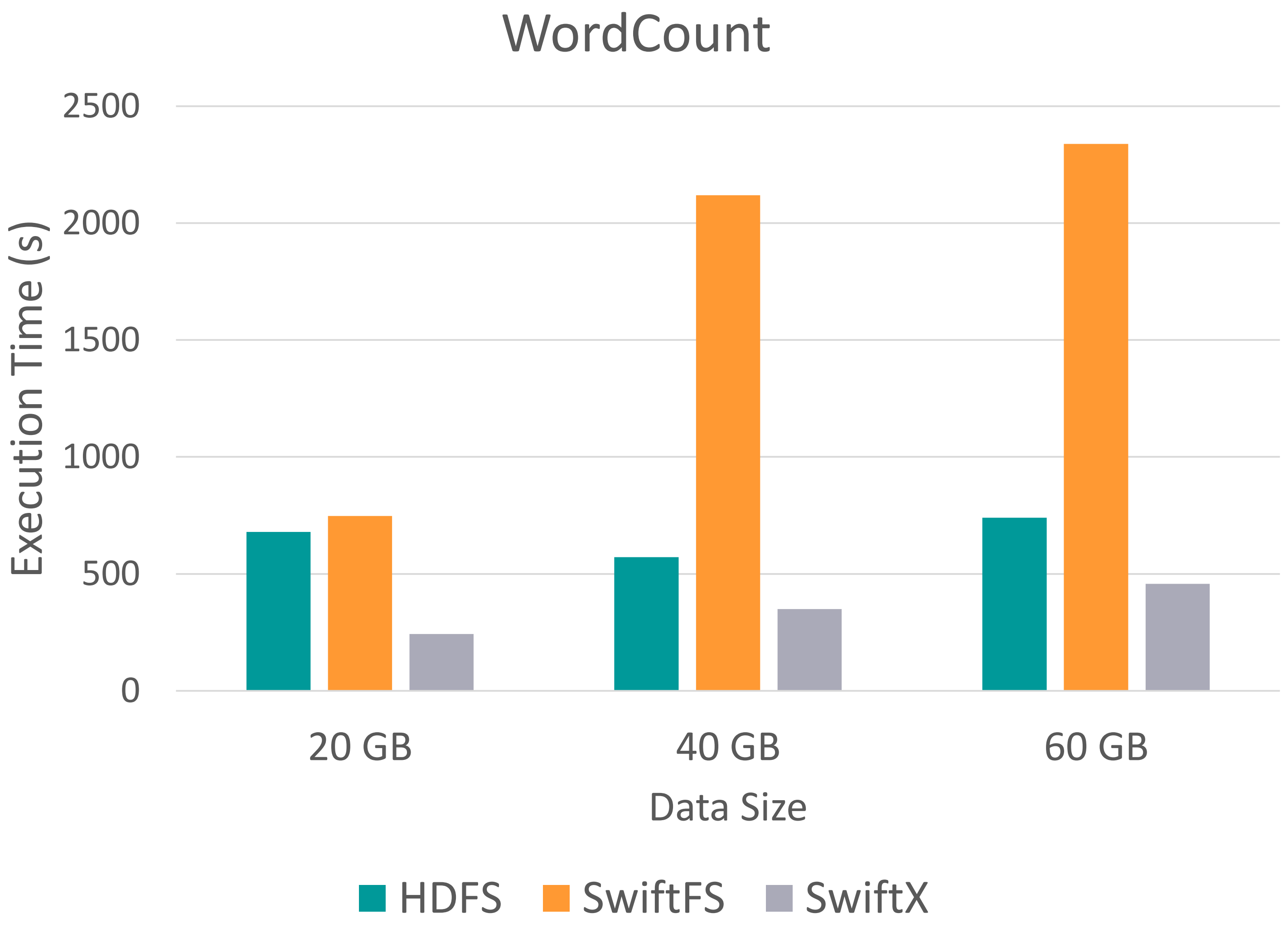

Default Write Design
Consistency not guaranteed




Proposed Write Design
Consistency guaranteed

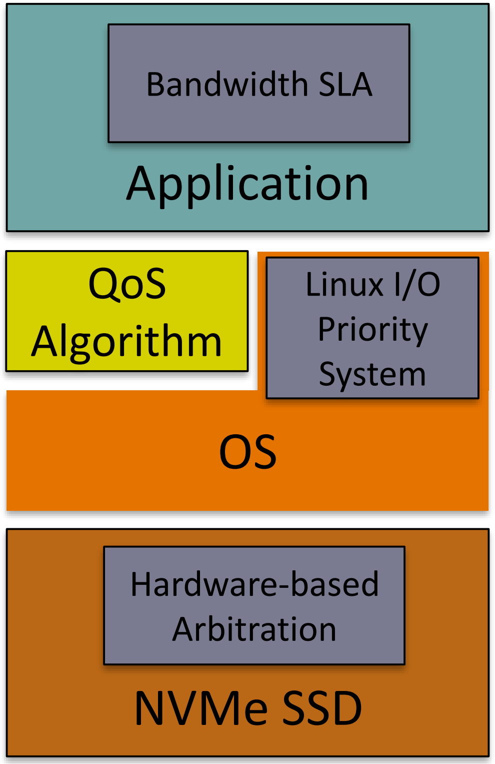
# Evaluation with WordCount

- Up to 83% improvement over SwiftFS
- Up to 64% improvement over HDFS
- With HDFS, data needs to be copied to/from Swift



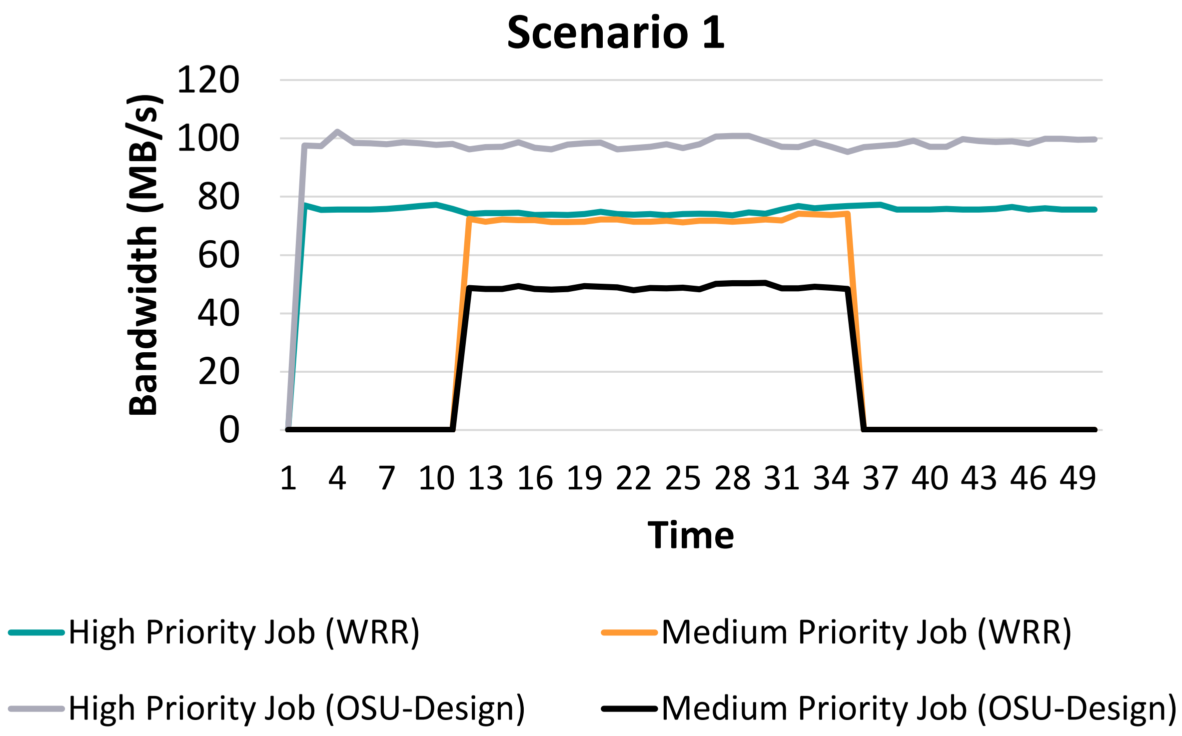
# QoS-Aware Storage

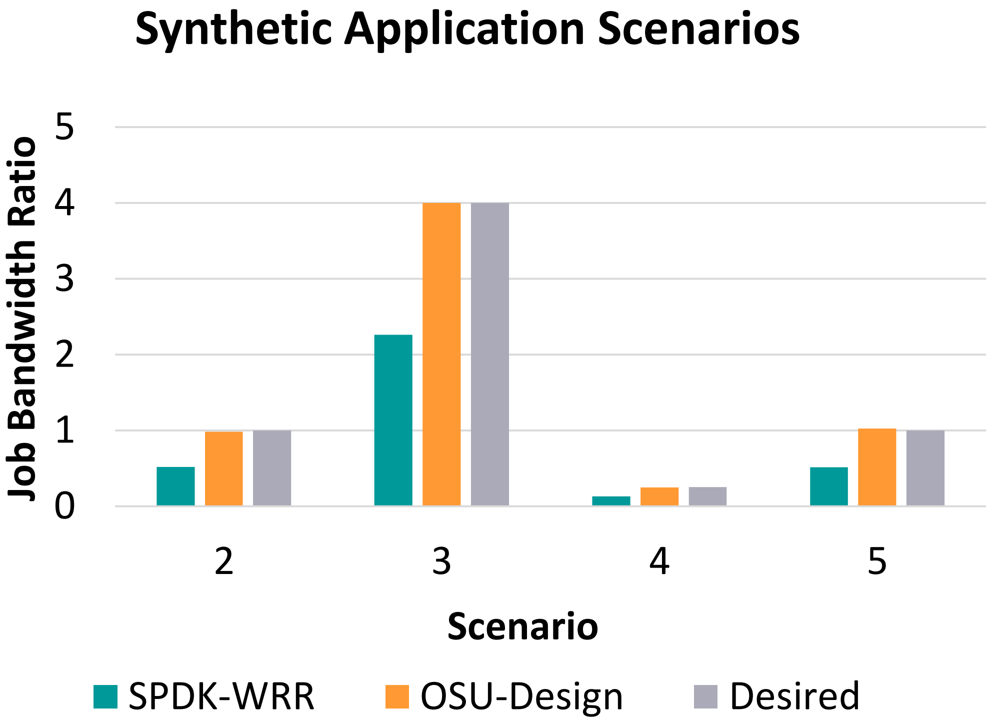
- Linux I/O priority system to transfer priority information to underlying runtime
- Hardware-based NVMe request arbitration
- Mechanisms to provide I/O bandwidth SLAs
- Request-size agnostic QoS algorithm




QoS-aware Storage Stack

# QoS-aware Storage

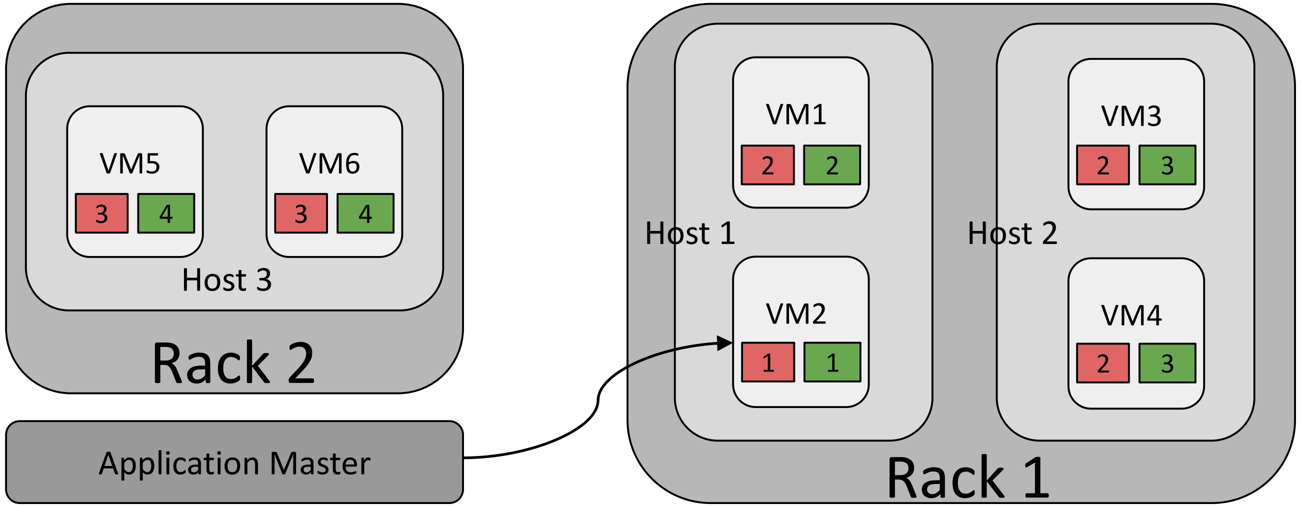



- Synthetic application scenarios with different QoS requirements
  - Comparison using SPDK with Weighted Round Robbin NVMe arbitration
- Near desired job bandwidth ratios
- Stable and consistent bandwidth

S. Gugnani, X. Lu, and D. K. Panda, Analyzing, Modeling, and Provisioning QoS for NVMe SSDs, UCC'18

# Topology-aware Communication: Map Task Scheduling

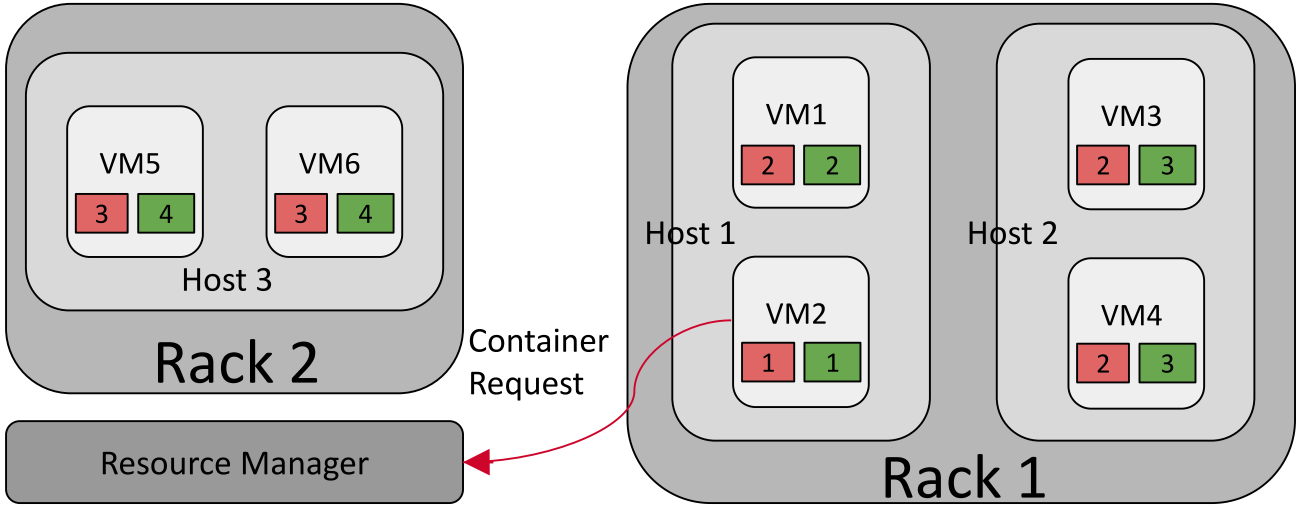



**Default Hadoop Policy**
1. Node local
2. Rack local
3. Off-rack

**Hadoop-Virt Policy**
1. Node local
2. Host local
3. Rack local
4. Off-rack

- Co-located VMs can communicate using loopback, without having to go through the network switch
- Maximize communication between co-located VMs
- Allocate Map tasks on a co-located VM before considering rack-local nodes or off-rack nodes
- Reduces inter-node network traffic through locality-aware communication

# Topology-aware Communication: Container Allocation




**Default Hadoop Policy**

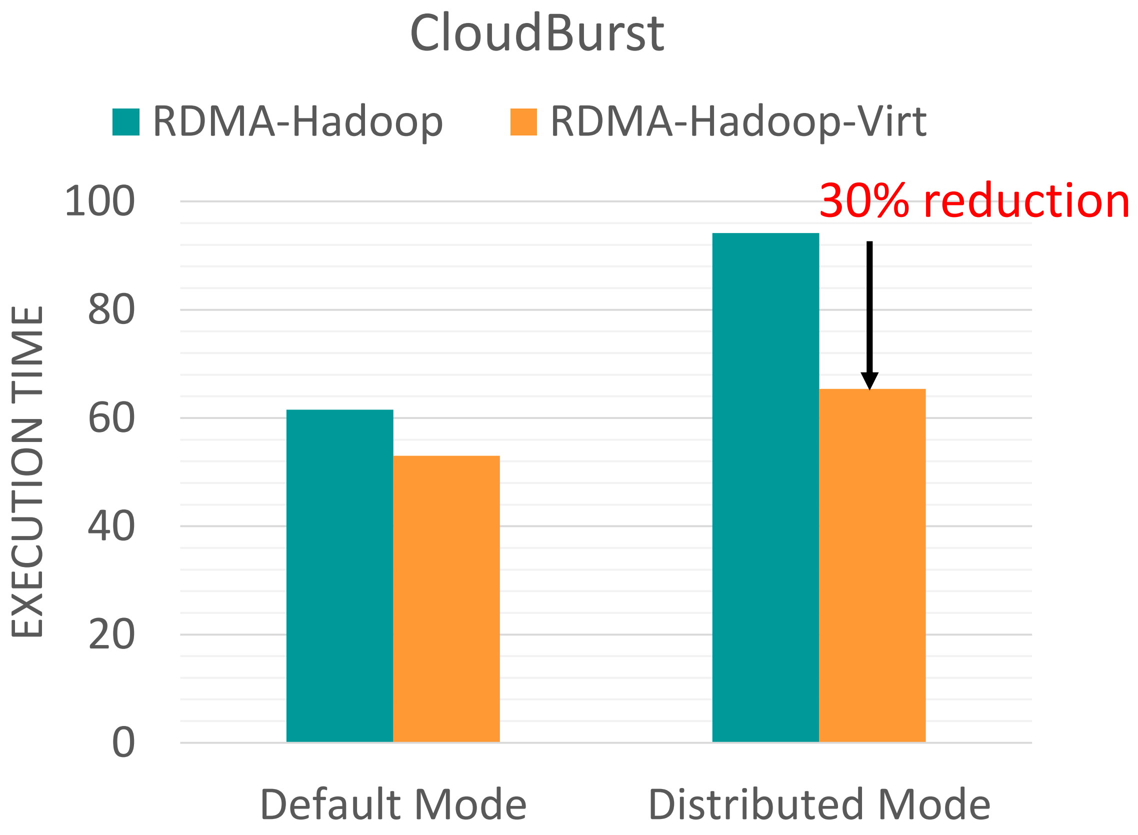
1. Node local
2. Rack local
3. Off-rack

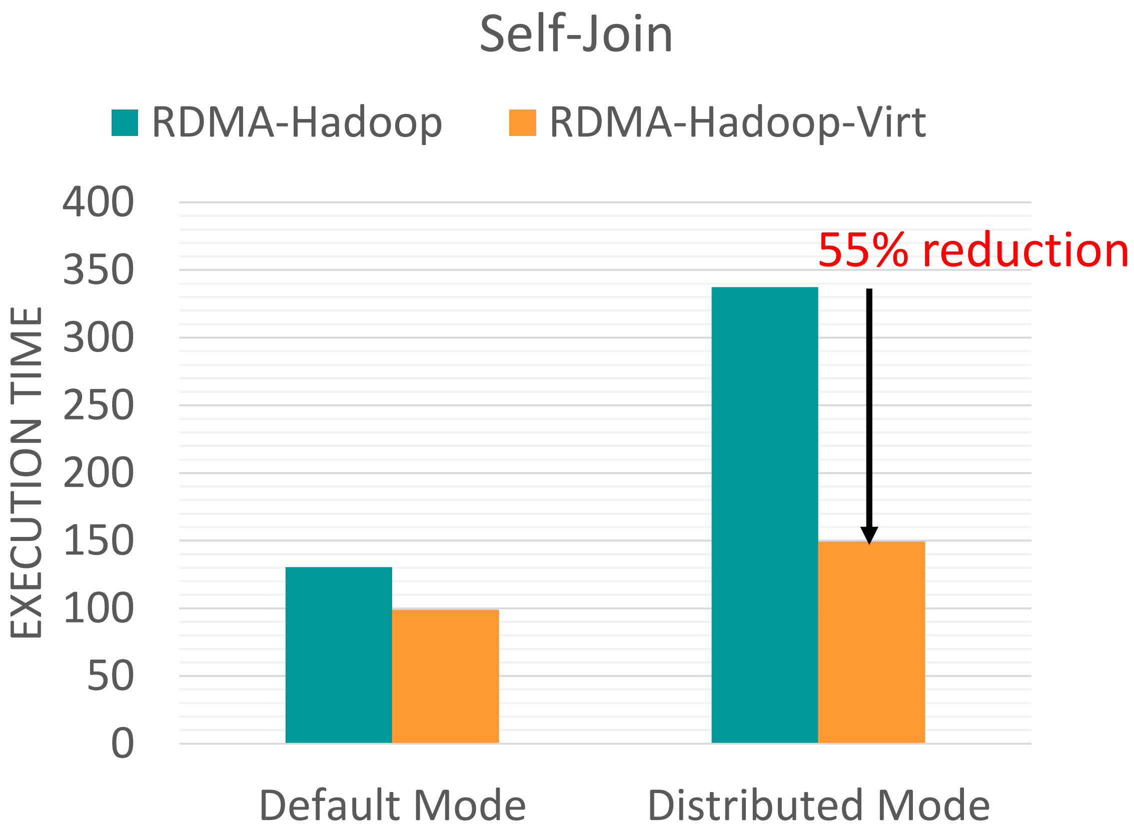
**Hadoop-Virt Policy**

1. Node local
2. Host local
3. Rack local
4. Off-rack

- Co-located VMs can communicate using loopback, without having to go through the network switch
- Maximize communication between co-located VMs
- Allocate Containers on a co-located VM before considering rack-local nodes or off-rack nodes
- Reduces inter-node network traffic through locality-aware communication

# Evaluation with Applications




– 14% and 24% improvement with Default Mode for CloudBurst and Self-Join

– 30% and 55% improvement with Distributed Mode for CloudBurst and Self-Join

# Conclusion

- Preliminary work to design Cloud-aware Storage and Communication Middleware
- QoS, Consistency, Scalability, and Performance as design goals
- Experimental results on working prototype are encouraging
- Future work
  - More work along storage direction
  - Use of NVMe, NVM, etc.
  - Additional design goals: Fault-tolerance and Availability

# Thanks!

gugnani.2@osu.edu

Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/

The High Performance Big Data Project (HiBD)

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