Accelerating Big Data Processing in the Cloud with Scalable Communication and I/O Schemes

Shashank Gugnani and Dhabaleswar K. Panda (Advisor)
Department of Computer Science and Engineering, The Ohio State University
{gugnani.2, panda.2}@osu.edu

ABSTRACT

With the advent of cloud computing, the field of Big Data has seen rapid growth. Most cloud providers provide hardware resources such as NVMe SSDs, large memory nodes, and SR-IOV. This opens up the possibility of large-scale high-performance data analytics and provides opportunities to use these resources to develop new designs. Cloud computing provides flexibility, security, and reliability, which are important requirements for Big Data frameworks. However, several important requirements are missing, such as performance, scalability, consistency, and quality of service (QoS). The focus of this work revolves around developing communication and I/O designs and concepts which can provide these requirements to Big Data frameworks. Specifically, we explore new ways to provide QoS and consistency in cloud storage systems, and provide scalable and high-performance communication frameworks.

1 INTRODUCTION

The cloud computing paradigm has motivated a large number of users to move their applications to the cloud or build private clouds within their organizations. However, performance degradation in virtualized environments is one of the primary hindrances limiting this migration. Most cloud and big data processing middleware use traditional TCP/IP sockets based communication which has known performance issues. Modern high-performance interconnects such as InfiniBand [4] and RoCE [1] offer advanced features like Remote Direct Memory Access (RDMA) and provide high-bandwidth low-latency communication.

In order to run data-intensive applications in the cloud efficiently, enabling cloud-based distributed storage solutions is a must. Existing solutions forgo consistency for high-availability and performance. Eventual consistency is the most popular model provided by these systems. While this consistency model provides the opportunity to implement highly available systems, it is unsuitable for running concurrent large-scale enterprise workloads. This highlights the need for stronger consistency in cloud storage systems.

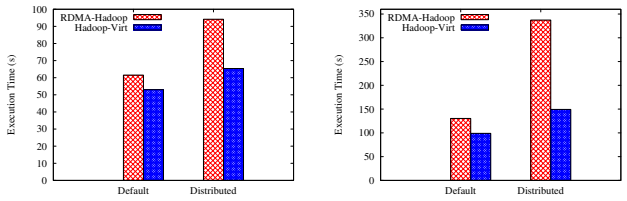
In multi-tenant cloud environments enabling service guarantees is essential. Prior work has focused mostly on software-based QoS. In this work, we focus on hardware-assisted QoS and providing application-oblivious service guarantees for cloud storage systems.

All these issues lead us to the following broad challenge: *How can we design efficient communication and I/O schemes to accelerate big data in the cloud?* To address these challenges, we propose designs for high-performance communication middleware and scalable and consistent cloud storage to enable efficient big data processing in the cloud.

2 PROPOSED DESIGNS

In this section, we briefly discuss our proposed designs.

2.1 Topology-aware communication

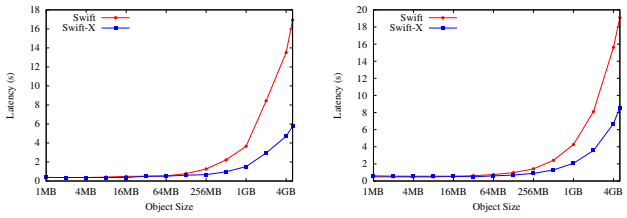


(a) CloudBurst (b) Self-join

Figure 1: Application Evaluation with Proposed Design (Hadoop-Virt) and RDMA-Hadoop

Existing designs in Hadoop are not aware of the topology of virtual clusters. We propose an **automatic topology detection module** and **virtualization-aware designs** in Hadoop to fully take the advantage of virtualized environments. Our topology detection module is based on the scalable MapReduce framework and can detect topology changes during runtime. We also enable **topology and locality-aware communication** by maximizing communication between co-located VMs [2]. This includes changes to the container allocation and map task scheduling policies in Hadoop. Our experimental evaluations (Fig. 1) show that our design delivers upto 34% better performance in the default execution mode and upto 52.6% better performance in the distributed mode as compared to the default RDMA-Hadoop for SR-IOV-enabled virtualized clusters.

2.2 Scalable Cloud Storage



(a) Get Latency (b) Put Latency

Figure 2: Get and Put Latency Evaluation with Proposed Design (Swift-X) and OpenStack Swift

To improve the scalability and performance of OpenStack Swift, we propose high-performance scalable designs. We propose changes to the Swift architecture and operation design. Our design uses the

proxy server only as a metadata server and uses client-based replication for scalability, bypassing the proxy server. Thus, the proxy server is no longer a bottleneck. We also propose **high-performance** implementations of network communication and I/O modules based on **RDMA** to provide the fast object transfer and **maximum overlap** between communication and I/O [3]. Our evaluation (Fig. 2) reveals that our designs can deliver up to 47% and 66% improvement in put and get latencies, respectively.

2.3 POSIX-like consistent Cloud Storage

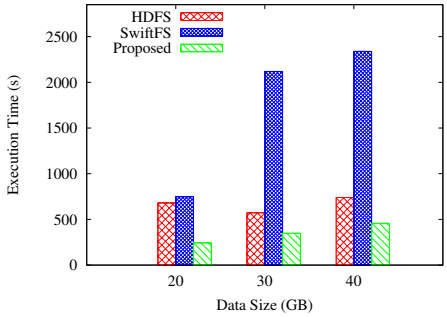


Figure 3: Evaluation of WordCount Execution Time with Proposed Design, SwiftFS, and HDFS

For providing a POSIX-like consistent cloud storage system, we propose the use of atomic write operations. These are implemented using a Two-Phase Commit (2PC) for atomicity and global timestamp server for providing proper order of operations. This not only provides **POSIX-like consistency**, but also **fault-tolerance** and **fine-grained concurrency control**. This allows any application requiring strong consistency to run directly on cloud storage. Experimental evaluation (Fig. 3) shows that our design can provide up to 64% improvement compared to HDFS and 83% improvement compared to SwiftFS for read-intensive workloads.

2.4 QoS-aware Storage Runtime

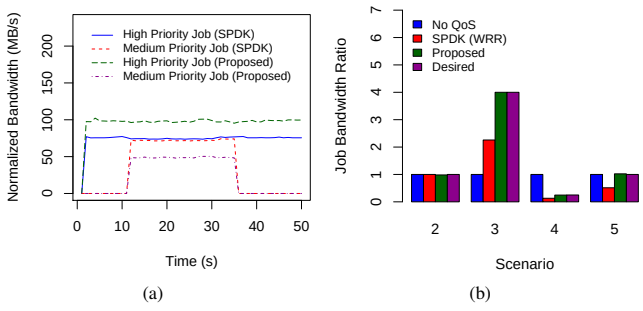


Figure 4: Evaluation with Synthetic Application Scenarios: (a) Bandwidth over time with Scenario1, (b) Job bandwidth ratio for Scenarios 2-5

For providing service guarantees, we use **hardware-based arbitration** provided by NVMe SSDs. To enable application-oblivious QoS provisioning, we use the Linux I/O priority sytem to transparently pass the service level to the underlying runtime. This provides a storage runtime which can provide **accurate bandwidth guarantees** to applications *without* modifying them. We evaluated our design and compared it with Intel SPDK (Fig. 4) , which uses weighted round robbin arbitration. Results show that our design provides accurate bandwidth guarantees, close to the desired values.

3 CONCLUSION

In this work, we presented scalable and high-performance communication and I/O schemes for efficient big data processing in the cloud. We proposed topology and locality-aware communication policies for Hadoop. For improving the scalability and performance of cloud storage, we discussed high-performance designs which we implemented on OpenStack Swift. For enabling applications requiring consistency guarantees, we used atomicity as a way to guarantee consistency. For service guarantees, we propose a QoS-aware storage runtime using hardware-based arbitration.

REFERENCES

[1] Infiniband Trade Association et al. 2010. Supplement to Infiniband Architecture Specification Volume 1, Release 1.2. 1: Annex A16: RDMA over Converged Ethernet (RoCE). (2010).

[2] Shashank Gugnani, Xiaoyi Lu, and Dhabaleswar K Panda. 2016. Designing Virtualization-Aware and Automatic Topology Detection Schemes for Accelerating Hadoop on SR-IOV-Enabled Clouds. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 152–159.

[3] Shashank Gugnani, Xiaoyi Lu, and Dhabaleswar K Panda. 2017. Swift-X: Accelerating OpenStack Swift with RDMA for Building an Efficient HPC Cloud. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. IEEE Press, 238–247.

[4] InfiniBand Trade Association. 1999. http://www.infinibandta.com (Oct. 2018). (1999).