# Building Application-Aware Network Environments Using SDN for Optimizing Hadoop Applications

Shuai Zhao[1], Ali Sydney[2], Deep Medhi[1]
[1]University of Missouri-Kansas City, [2]Raytheon BBN Technologies
(shuai.zhao, dmedhi)@umkc.edu, asydney@bbn.com

## ABSTRACT

Hadoop has become the de facto standard for Big Data analytics, especially for workloads that use the MapReduce (M/R) framework. However, the lack of network awareness of the default MapReduce resource manager in Hadoop can cause unbalanced job scheduling, network bottleneck, and eventually increase the Hadoop run time if Hadoop nodes are clustered in several geographically distributed locations. In this paper, we present an application-aware network approach using software-defined networking (SDN) for distributed Hadoop clusters. We develop the SDN applications for this environment that consider network topology discovery, traffic monitoring, and flow rerouting in addition to loop avoidance mechanisms.

## Keywords

Application-Aware Networking; Hadoop; Software-Defined Networking

## 1. INTRODUCTION

Hadoop M/R is a scalable framework that allows dedicated, seemingly unbound number of servers to participate in big and complex data analytics' processes. Consider the scenario where a site does not have sufficient Hadoop nodes and resources to run a large application. This could be due to a private cloud that may not have enough resources that may use a public cloud as a supplement for Hadoop computation. Generalizing this further, we consider running Hadoop applications that need to be run on Hadoop nodes distributed over multiple geographic locations. Hadoop's lack of network awareness becomes an issue. While the compute and disk I/O requirements can be scaled with the number of servers, a distributed environment leads to increased network traffic between Hadoop nodes. The bandwidth-intensive shuffle phase of M/R contributes significantly to the overall response time. Furthermore, the response time is exacerbated if communication patterns are heavily skewed, as is very common in many M/R workloads when Hadoop nodes are distributed in multiple geographic locations. These issues associated with the M/R program exist as the default Hadoop resource manager only considers compute resources such as CPU, disk, and memory, but is unaware of the underlying network that connects multiple locations.

We propose an application-aware networking (AAN) approach to manage such distributed Hadoop network traffic. AAN provides the ability for an intelligent network to maintain current information about applications that connect to it and as a result, optimize the operation of these applications. The information maintained includes the application state and resource requirements. We use SDN to provide a software-based implementation of AAN for Hadoop distributed clusters, which enables a fine-grained way of controlling individual applications and network devices. SDN removes the control plane from the network hardware and implements it in software instead. AAN benefits from SDN in three ways: (1) by enabling dynamic control, configuration, and allocation of resources; (2) by running network control in a separate server from the traffic forwarding device; and (3) by enabling innovation of experimenting new network protocol implementations and tests. We have now prototyped SDN applications for AAN that include network topology discovery, traffic monitoring, and flow rerouting in addition to loop avoidance mechanisms.

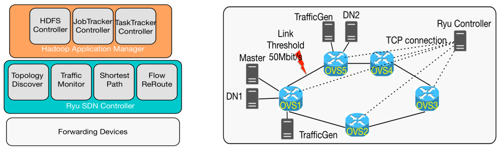
Heuristic task schedule algorithms were proposed to achieve data locality and increase network bandwidth utilization [4, 5, 7]. However, these approaches can be improved if we consider the integration of applications



*SIGCOMM '16, August 22–26, 2016, Florianopolis, Brazil*



DOI: http://dx.doi.org/10.1145/2934872.2959059




(a) (b)

**Figure 1: *(a) Architecture, (b) Topology***

with the network. Other research has considered a runtime job scheduling platform but simulations were not conducted to validate this approach [6]. A well-defined control API for SDN was introduced in [2]. This research was leveraged in the design of our SDN application framework. Corral [3] proposed an offline Network-Aware scheduling framework using SDN.

Subsequent sections of the paper are organized as follows. Section 2 describes our proposed AAN architecture. In Section 3, preliminary experimentation results are presented. We conclude with a summary and future work in Section 4.

# 2. AAN ARCHITECTURE

Figure. 1(a) shows our proposed architecture. This has two components: (1) an SDN environment that includes packet forwarding devices, an SDN controller that provides SDN applications such as topology discovery, traffic monitoring, shortest path, and flow rerouting, in addition to an ARP solver mechanism for network-loop avoidance; (2) a Hadoop Application Manager that includes three components: a Hadoop File System (HDFS), JobTracker, and TaskTracker Controller.

The Hadoop application manager controls any traffic movement inside of the Hadoop cluster. Three main components are implemented using a default Hadoop configuration port number. The HDFS controller enables SSH access among clusters, and it also supports Hadoop file operations such as read and write functions. The JobTracker Controller and TaskTracker Controller support the allocation of Hadoop resources and communications between master and data nodes as well as the data nodes themselves.

# 3. PRELIMINARY EXPERIMENTATION

The testbed is created using virtual machines (VMs) in the Global Environment for Network Innovations (GENI) [1] platform. Fig. 1(b) shows our topology setup. We created a ring topology using five OpenVSwitches (OVSs) as forwarding devices. To illustrate the impact, we consider a Hadoop cluster with one Hadoop master and two data nodes (DN) at two different locations, along with two iPerf traffic generators

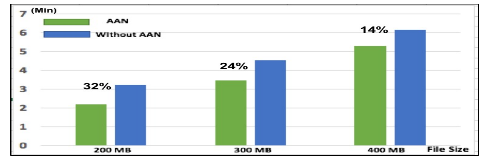



**Figure 2: Hadoop M/R Run Time Comparison**

for background traffic. All of the VMs have one core X5650@2.67GHz, 880MB memory. The bandwidth between the OVSs and host was set to 100 Mbps. The link threshold was set to 50 Mbps between OVS1 and OVS5.

In this environment, we tested Hadoop M/R performance by running the WordCount program with the following file sizes: 200 MB, 300 MB, and 400 MB. Fig. 2 shows that Hadoop WordCount run time decreases by an average of 23% when using our AAN platform.

# 4. CONCLUSION AND FUTURE WORK

In this paper, we presented an application-aware networking environment using SDN for Hadoop M/R applications running in a distributed environment. We conducted a proof-of-concept experiment to show that the AAN approach reduces the compute time. The flow re-route results show that the SDN controller is CPU intensive. Rigorous load testing, using a high-end SDN controller in a complex topology and M/R scheduler optimizations are planned in our future work. Our goal is to schedule M/R jobs in a more efficient and manageable way using our AAN environment.

# 5. REFERENCES


[1] C. Elliott. Geni-global environment for network innovations. In *LCN*, 2008.

[2] Ferguson et al. Participatory networking: An API for application control of SDNs. In *ACM SIGCOMM*, 2013.

[3] Jalaparti et al. Network-aware scheduling for data-parallel jobs: plan when you can. In *ACM SIGCOMM Comp. Comm. Rev.* ACM, 2015.

[4] Jin et al. Bar: An efficient data locality driven task scheduling algorithm for cloud computing. In *Proc. 11th IEEE/ACM CCGrid*, 2011.

[5] Tan et al. Coupling task progress for mapreduce resource-aware scheduling. In *INFOCOM*, 2013.

[6] Wang et al. Programming your network at run-time for big data applications. In *SDN Hot Topics.* ACM, 2012.

[7] Zaharia et al. Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. In *Proc. of 5th Euro. Conf. on Comp. Systems*, 2010.