Poster: A Reconfigurable Control Platform for Hadoop using SDN

Jason Anderson  
School of Computing  
Clemson University  
jwa2@clemson.edu

Amy Apon (Advisor)  
School of Computing  
Clemson University  
aapon@clemson.edu

Kuang-Ching Wang (Advisor)  
Department of Electrical and Computer Engineering  
Clemson University  
kwang@clemson.edu

ABSTRACT
This poster describes the motivation, design, and construction of a hardware platform to facilitate research in using SDN to observe and control Hadoop MapReduce traffic. OpenFlow-enabled hardware switches, multiple network interfaces, and Open vSwitch were utilized to enable users of the system to reconfigure the logical network topology with a software controller. The system is currently being used to research application-aware mechanisms for network reconfiguration toward the optimization of Hadoop shuffle traffic.

Categories and Subject Descriptors
C.2.1 [Computer-Communication Networks]: Network Architecture and Design

General Terms
Management, Performance, Experimentation

Keywords
OpenFlow, Software Defined Networking, Hadoop

1. INTRODUCTION
Software Defined Networking (SDN) is emerging as a paradigm that has applications in many areas of computing. SDN allows control of network switches from a remote server. OpenFlow [1], an SDN control protocol, provides a means for controllers such as Floodlight [2] to manage the network.

One area of intense research is to use SDN as a way to implement application-aware network control. Our lab is exploring methods of using OpenFlow to optimize network characteristics for traffic generated by Hadoop [3] MapReduce jobs. To enable research in this area, we decided to build a cluster to support Hadoop and the OpenFlow SDN paradigm.

2. PHASE I
2.1 Design
The SDN paradigm of separating the control and data planes was the basis for designing our cluster’s network. We used physically separate control and data networks in star topologies with separate switches.

Figure 1. Control and data star-topology networks.

To provide a mechanism for monitoring traffic, Open vSwitch (OVS) [4] was planned for abstracting the data link in each compute node. Additionally, we required traffic control capabilities in central locations. Therefore, we planned to use OpenFlow-enabled physical switches.

2.2 Construction
We procured six computers that were being decommissioned from one of our student labs. The eight-year-old machines were sufficient for our SDN testbed thanks to their dual network interfaces. A seventh machine acted as the head node and hosted the OpenFlow and Hadoop control software.

We were provided with two OpenFlow-enabled switches, which formed the hubs of our independent control and data networks. Floodlight was used to manage traffic on OVS as well as the physical switches.

2.3 Results
The SDN cluster was a success, as it allowed our team to develop tools to monitor, regulate, and visualize Hadoop traffic. It has also proven useful in the first steps of exploring the relationship between shuffle traffic and job run time.
3. PHASE II

3.1 Requirements
While the Phase I cluster worked well, our end goal was to use SDN to create an application-aware control mechanism to dynamically reconfigure the network at runtime. The topologies we wanted to explore required the addition of more compute nodes and internode links.

![Figure 2. A 2-D Torus network, requiring 4 links per node.](image)

3.2 Design
We planned on a total of four data interfaces per compute node, all connected to an SDN switch, and managed by OVS. We planned to use OpenFlow to define a logical layer-2 topology with flows installed on the switches.

![Figure 3. Final physical topology with 4 data links per node.](image)

3.3 Construction
We obtained six more machines for a total of twelve identical compute nodes. On each machine, we installed two consumer-grade dual-port NICs to provide a total of six interfaces. The new interfaces from each machine were connected to an SDN switch, which was again controlled by Floodlight on the head node.

Using Floodlight’s RESTful interface, we wrote software to logically reconfigure the topology to a set of predefined layouts, as well as implementing a simple spanning protocol for each topology.

4. FUTURE WORK
Our current research using the cluster is to explore how different topologies and types of data affect the shuffle time of a Hadoop job. Using correlations we obtain from this data, we plan to test an adaptive mechanism to reconfigure the network as the job type changes.

Another research project that will utilize the cluster an investigation into using SDN as a proxy mechanism for distributed services, employing information encoded in IP headers to alter destination addresses.

5. ACKNOWLEDGMENTS
[TBD, departmental and funding acknowledgments]

6. REFERENCES