OpenFlow Enabled Hadoop Over Local and Wide Area Clusters

Sandhya Narayan and Stuart Bailey
InfoBlox Inc.
Santa Clara, CA
snarayan@infoblox.com
sbailey@infoblox.com

Matthew Greenway, Robert Grossman,
Allison Heath and Ray Powell
Computation Institute
University of Chicago
mgreenway@uchicago.edu
robert.grossman@uchicago.edu
rpowell1@uchicago.edu
aheath@uchicago.edu

Anand Daga
Dept. of Computer Science
University of Houston
adaga2@cs.uh.edu

Abstract— **Hadoop has emerged as an important platform for data intensive computing. The shuffle and sort phases of a MapReduce computation often saturate top of the rack switches, as well as switches that aggregate multiple racks. In addition, MapReduce computations often have "hot spots" in which the computation is lengthened due to inadequate bandwidth to some of the nodes. In principle, OpenFlow enables an application to adjust the network topology as required by the computation, providing additional network bandwidth to those resources requiring it. We describe Hadoop-OFE, which is an OpenFlow enabled version of Hadoop that dynamically modifies the network topology in order to improve the performance of Hadoop.**

Keywords: Hadoop, OpenFlow, OpenFlow over Ethernet, data intensive computing, MapReduce

I. Introduction

A. Hadoop-OFE

In the recent years, data intensive programming using Hadoop and MapReduce has become increasing important. As normally deployed, Hadoop's implementation of MapReduce in a multi-rack cluster is dependent upon the top of the rack switches and of the aggregator switches connecting multiple racks.

We are currently developing Hadoop-OFE[1]. Hadoop-OFE is described in more detail below, but the basic idea is to combine OpenFlow (OF) enabled switches and a modified JobTracker within Hadoop that is OpenFlow aware in order to improve the performance of Hadoop. Over time our hope is that Hadoop-OFE over standard Ethernet can provide superior performance to Hadoop over specialized interconnects, such as InfiniBand.

To quantify the performance improvements offered by Hadoop-OFE, we are performing experimental studies using: i) the MalStone Benchmark [1]; and, ii) an open source Hadoop-based application called Matsu [2] for processing satellite images to detect floods and other phenomena.

B. Relevance to the HPC community.

Data intensive computing is now widely recognized as important to the HPC community. There is relatively little work that is public on building OpenFlow enabled clusters to support data intensive computing. One of our goals with this SCinet Sandbox demonstration is to highlight the importance of the applicability of OpenFlow to this class of problems and to encourage interest in working on Hadoop-OFE.

Data intensive computing has traditional been restricted to clusters within a data center. Another goal of this project is to show that wide data intensive computing is possible with the appropriate high performance wide area networks and the ability of applications to suitably dynamically modify the network topology.

II. Hadoop-OFE

For many Hadoop-based applications, the network requirements for the Map and Reduce phases of the computations are quite different. Also, many Hadoop applications are iterative in nature, with different network requirements for different phases of the iteration. In principle, if the network topology of the cluster can be adjusted as required to support these requirements, greater efficiency could be achieved when processing data with Hadoop.

[1] Note that OFE is an abbreviation for both OpenFlow over Ethernet and OpenFlow Enabled.

Figures 1 and 2 show a Hadoop cluster with and without Openflow networking. To illustrate the benefits of Openflow consider the following example.

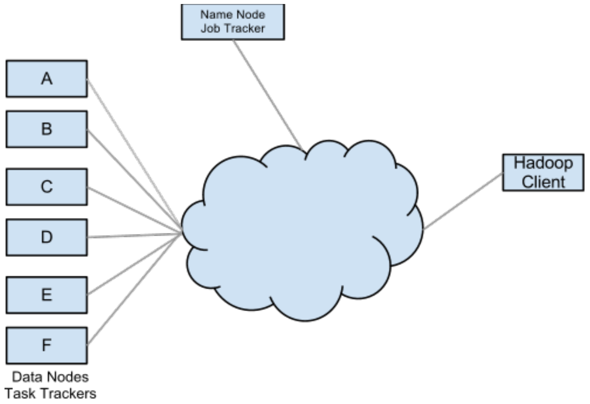

Figure 1. A Hadoop cluster

The JobTracker in Figure 2 is modified to get the OpenFlow Controller to change the properties of flow paths dynamically, depending on the execution stage of a job. During a Map job, the flow-path between systems A, B and system F (which holds input data) can be given higher priority for passing the data needed for the job. Likewise, during a Reduce job the flow-path between systems A, B and E (which performs Reduce) gets higher priority.

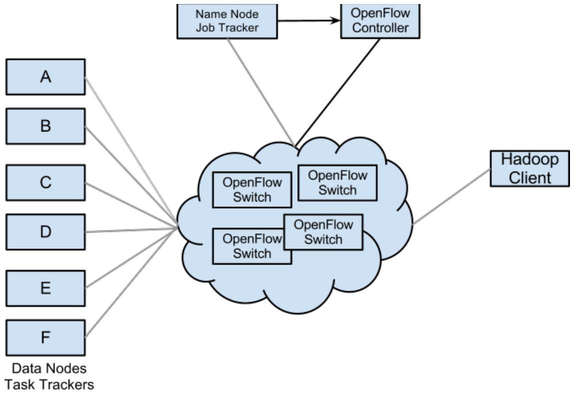

Figure 2. A Hadoop cluster with OpenFlow enabled interconnectivity.

III. Hadoop-OFE Testbed

A. SCinet Sandbox

At SC12, we ran Hadoop-OFE in a local cluster that was part of the Open Cloud Consortium Research Booth at SC 12. The local cluster used 10G networking to connect the nodes in the cluster.

B. Wide Area Testbed

We are putting in place a wide area OpenFlow testbed, that includes an OCC Hadoop cluster at a data center in Chicago, an OCC Hadoop cluster at a data center at the LVOC, and a OCC Hadoop cluster at a data center in Miami. The three data centers are connected by 10G networks. See Figure 3.

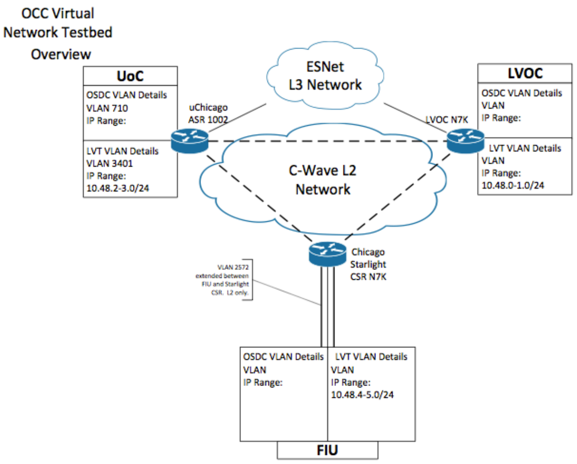

Figure 3. Wide area OpenFlow testbed for Hadoop-OFE.

C. Nature of the experiments

MalStone is a data intensive analytic benchmark that represents a typical component of a computation required to build statistical models [1]. Matsu is an Open Cloud Consortium Project (matsu.opencloudconsortium.org) that uses a cloud-based infrastructure to process the data for NASA's EO-1 satellite to produce various data products, applies analytics to the data products, and makes the data products and results of the analytics available to the public [2].

D. Desired outcomes

We have two desired outcomes for the experimental studies.

The first desired outcome is to quantify the performance improvement gained when running applications over Hadoop-OFE compared to running over standard Hadoop. This will be done using the Hadoop and Hadoop-OFE Clusters in the Open Cloud Consortium SC 12 Research Booth. In the preliminary results described in Section V below, we have demonstrated that OF can improve the performance of simple MapReduce operations. The goal of the demonstrations is to show that similar or better improvements are obtainable for actual applications.

The second desired outcome is to quantify the performance of wide area implementation of Hadoop-OFE compared to a wide area implementation of standard Hadoop.

We plan to quantify the performance improvement of Hadoop-OFE using the MalStone Benchmark as well as the Matsu application.

E. Vendor Collaborations

The SCinet Sandbox will be a joint project between Infoblox (a primary contributor of Hadoop-OFE), University of Chicago (one of the developers of Matsu software stack), and the Open Cloud Consortium (which manages and operates the Matsu Project).

IV. Preliminary Results

We set up a 10 node Hadoop cluster running the Cloudera distribution of Hadoop on 2 physical systems (xenovs2 and xenovs3) connected with a physical switch. The Hadoop nodes run in virtual machines (VMs) in a XenServer virtualized environment [8]. The Open VSwitch (OVS) [9] is the default network stack for the XenServer. Open Vswitch is a multi-layer software switch that supports OpenFlow standards [10]. Flows in the OpenVswitch can be setup by an OpenFlow Controller. We used BigSwitch's opensource Apache-licensed, Java-based OpenFlow Controller to setup flows. The Cloudera Manager (CM) runs on a VM in the cluster and manages the Hadoop cluster.

For the experiment, we used the sort benchmark to run as the job under test. Hadoop comes with a MapReduce program that does a partial sort of its input. It is very useful for benchmarking the whole MapReduce system, as the full input dataset is transferred through the shuffle. The three steps are: generate some random data, perform the sort, then validate the results [6].

We set up two queues in the Open vSwitch (OVS-OF) on the system xenovs3 and setup different priorities for the queues. Next we added a special flow entry for iperf traffic to flow entries on OVS-OF and assigned it to the queue with lower priority. With this, the iperf-traffic had lower priority over other traffic. In our cluster, only other traffic of significance is due to the Hadoop job. We ran the Hadoop sort job with network congestion in two cases: a) without OpenFlow enabled queues and b) With OpenFlow enabled queues. Figure 4 shows the results of the two cases. It shows that the Hadoop job performs better when iPerf traffic has lower priority over other traffic, with the traffic priority controlled by OpenFlow.

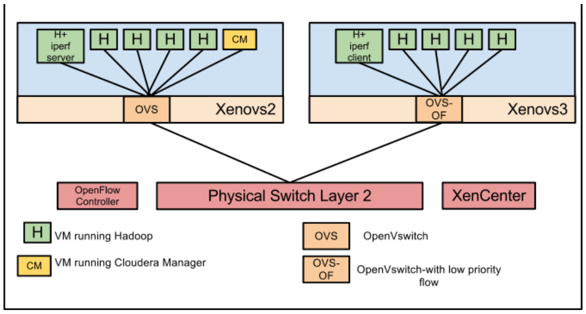

Figure 4. The experimental setup of our preliminary experiments.

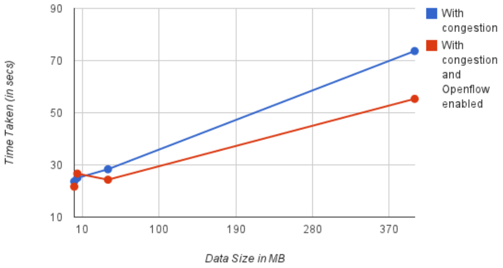

Figure 5. The effect of OpenFlow queues on Hadoop performance in a congested network.

V. Related Work

Several different approaches have been taken to accelerate Hadoop. Using high performance interconnects, overlapping the execution phases of Hadoop, providing early results from jobs, changing network topology based on application are some approaches presented in the research outlined below.

Topology switching [3] is a method that provides control to individual applications for deciding best how to route data among their nodes. Topology switching formalizes the simultaneous use of multiple routing mechanisms in a data center, allowing applications to define multiple routing systems and deploy individualized routing tasks at small time scales.

Hadoop-A [4] is an acceleration framework that optimizes Hadoop with plugin components implemented in C++ for fast data movement. The performance improvement is derived from three methods. First, a new method is used to merge the input to the reduce phase. Instead of repeated merge and store to disk, a reducer fetches only the keys from mappers, and merges them. It fetches the value only at the end, when all the merges are completed. Since the size of the key is small compared to that of the value, the entire input for merge can be stored in memory, avoiding going to the disk. Next, a pipeline is designed to overlap the shuffle, merge and reduce phases. Finally, it makes use of RDMA interconnects such as InfiniBand.

Hadoop Online [5] presents a modified version of the Hadoop MapReduce framework that supports online aggregation, which allows users to see "early returns" from a job as it is being computed. The Hadoop Online Prototype (HOP) also supports continuous queries, which enable MapReduce programs to be written for applications such as event monitoring and stream processing

Hadoop-OFE's approach to acceleration is orthogonal to the methods discussed above. Its goal is to improve the performance of MapReduce in Hadoop by utilizing OpenFlow as the interconnect between Hadoop nodes. One strategy is to make use of the QoS abilities of OpenFlow, which allows control over egress queues in an OpenFlow switch. This makes it possible for different flows to have different priorities over the bandwidth, and allows an application to control this priority setting. Thus applications can dynamically set different priorities to flows. In the case of Hadoop MapReduce there are distinct phases of execution that can benefit by prioritizing traffic on the network.

We have described Hadoop-OFE and several experimental studies that are underway to quantify the performance advantages of a version of Hadoop that uses OpenFlow to dynamically adjust the network topology of local and wide area Hadoop clusters. One of the experimental studies uses MalStone, a benchmark typical of the computations required when building statistical models over big data. The other experimental study uses Matsu, an application for processing satellite images. Both these Hadoop applications puts pressure on the network switches due to the amount of data transported.

VI. Summary and Conclusion

We have described Hadoop-OFE and several experimental studies that are underway to quantify the performance advantages of a version of Hadoop that uses OpenFlow to dynamically adjust the network topology of local and wide area Hadoop clusters. One of the experimental studies uses MalStone, a benchmark typical of the computations required when building statistical models over big data. The other experimental study uses Matsu, an application for processing satellite images. Both these Hadoop applications put pressure on the network switches due to the amount of data transported.

VII. References

[1] Collin Bennett, Robert L. Grossman, David Locke, Jonathan Seidman and Steve Vejcik, MalStone: Towards a Benchmark for Analytics on Large Data Clouds, The 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010), ACM, 2010.

[2] Daniel Mandl, Stuart Frye, Pat Cappelaere, Robert Sohlberg, Matthew Handy, and Robert Grossman, The Namibia Early Flood Warning System, A CEOS Pilot Project, IGARSS 2012.

[3] K. Webb, A. Snoeren, and K. Yocum. "Topology Switching for Data Center Networks". Workshop on Hot Topics in Management of Internet, Cloud and Enterprise Networks and Services (Hot-ICE), March 2011.

[4] Yandong Wang, Xinyu Que, Weikuan Yu, Dror Goldenberg, Dhiraj Sehgal, Liran Liss. Hadoop Acceleration Through Network Levitated Merge, SC11, Seattle, WA.

[5] Tyson Condie, Neil Conway, Peter Alvaro, Joseph M. Hellerstein, Khaled Elmeleegy, Russell Sears: MapReduce Online. NSDI 2010: 313-328.

[6] Tom White, Hadoop : The Definitive Guide, 2nd edn., O'Reilly, Sebastopol, CA, 2011.

[7] Iperf, http://code.google.com/p/iperf/

[8] XenServer, http://blogs.citrix.com/2011/09/30/xenserver-6-0-is-here/

[9] Open Vswitch, http://openvswitch.org/

[10] OpenFlow, https://www.opennetworking.org/

[11] FloodLight, http://floodlight.openflowhub.org/