CSE 222A

Traffic Optimization for Hadoop Based on SDN

Checkpoint 2 Report

Siyuan Liu, Jiaxu Zhu, Xinyi Wang

Background

At previous project checkpoint, we have designed the system structure (it is shown as a picture in the checkpoint 1 report), which consists of 1) Hadoop cluster network, 2) auxiliary program attached to each node in the network, 3) SDN control system. Also we had tried to build the integral system of these three components. Our team used vagrant to build the hadoop cluster network with VM nodes, and tried to use Mininet and OpenFlow to implement a SDN control system above the hadoop network. With the work around the previous checkpoint, we got an insight about a Hadoop application system based on SDN. Then we planned to complete the building of the entire system and make experiment on the system. The rest of the report is all about the work and new achievement before the checkpoint 2.

Achievement

System

SDN Control System: Flow Allocator

We've designed the SDN Control System to be a Flow Allocator. And the Flow Allocator implements both routing and flow allocation algorithms.

For the routing part in the Flow Allocator, we compute k-shortest paths among all pairs of nodes in the network. And the implementation uses hop counts as the distance metric. For the flow allocation part, we use information collected from auxiliary programs attached to nodes in the network. Now we just have 0-1 flow for one path, then we use OpenDayLight (OpenFlow) to update flows on the network and allocate flow path for packets.

Hadoop Cluster System

The system building and deployment can be divided into two parts and are almost finished.

1. Set up Hadoop cluster using vagrant
    a. Set up virtual machines using Vagrant.

        First we set up 8 virtual machines (1 name node with 8GB memory, 7 data node with 2GB memory). Vagrant is used to eliminate redundant work by setting up, booting and provision all virtual machines together with Vagrantfile (/vagrant/Vagrantfile). For the system, we choose ubuntu server 14.04 and only install the GUI on the name node for further cluster and hadoop jobs tracking.

    b. Deploy Hadoop using Cloudera Manager

        To deploy Hadoop on those 8 virtual nodes, we take the advantage of Cloudera Manager, which provide an integrated and fast way to install Hadoop. Also, we only active the core Hadoop service (HDFS, YARN, etc.) because our virtual nodes are much less powerful than physical nodes and our benchmark is simple sorting job.

2. Migrate the cluster into the gns3 virtual network

    When we first install Hadoop on those virtual nodes, they are all attached to the network adapter of the physical node and can communicate with the Internet. While doing the later SDN experiment, we need to build a virtual network that can trace network flow and optimize routing behaviour.

    Thus, gns3, the network simulator, is a great choice for us. Unlike Mininet, It can load virtual nodes generated by VirtualBox and VMWare, also it can simulate all kinds of routers as long as you give it the router image (ex. Cisco IOS file) and do the correct configuration.

Specifically, to parallelized this project by separating build SDN algorithms from Hadoop platform testing. We first migrate the cluster into a virtual network with virtual Cisco routers and then finally replace these routers with our own routers.

a. Router configuration

   Once you load the Cisco router image into gns3, you need to first configure it as DHCP server that can assign IP addresses to virtual nodes. What's more, Hadoop cluster relies on fixed IP addresses. We let the IP addresses assigned within gns3 be the same as those given by Vagrant, otherwise we need to reinstall all Hadoop service, which is horrible.

b. Cluster Migration

   Once the router is configured, we load virtual nodes into gns3 either. Then we test the clustering by running a simple word count jobs. also we can track the network flow using wireshack integrated in gns3.

Experiment

Our results should consist of several groups of experiments:

1. X: time, Y: Completion rate of Hadoop jobs. So the goal is to observe the performance of Hadoop with SDN. There should be more analysis with these observation.
2. X: Hadoop Benchmarks Y: Hadoop completion time for both of Normal Network and SDN. So it is like bar charts for comparing Hadoop performance of using SDN or not.
3. X: different adjustment of SDN controller, Y: Hadoop completion time. The goal is to come out optimal SDN strategy details for Hadoop.

We may design more experiments and graphs later.