CSE 222A PROJECT - GROUP 6

Sreejith Unnikrishnan, Stanislav Mushits, Amit Borase, Ritvik Jaiswal

November 4 2015

Milestone report

1 Project Topic

We are working towards verification of varys scheduler[1] and associated performance improvement in a data center environment. As a part of this goal, we have completed the following tasks.

2 Setting up Virtual Machines

The first step we performed was to setup the allotted virtual machine. We analyzed the required programming environment, tools and dependencies. Based on the analysis we ensured that the required programs and API‘s are installed on the machine. A shell script was written to automate this task, under utils directory of our repository. This makes the process of adding a new VM to our setup easier.

3 Deciding the network topology

Datacenter environment is unique in terms of traffic patterns and topology and because of that we gave special care while designing our network topology. Given that the topology will affect the data transmission efficiency of different coflows, we decided to work on a fat-tree based network setup. It consists of hosts, edge switches, aggregation layer switches and core switches. We use the open source library 'Fast Network Simulation Setup (FNSS)' [2] for topology development, which then later on is used to deploy mininets. We made sure that the important topology parameters can be easily configured and the software structure flexible enough to accommodate future changes.

4 Topology development

We created topology generator code using Python, namely topo_gen.py [3], using the FNSS framework. The topology generator requires 'network.config' file as an input. It parses the file to get the K value for fat-tree topology, the link speeds, link delay etc. By changing these values we can simulate different network environments as required without altering the code. Once the topology is created using FNSS, we port the topology into an XML file, so that it could be reused by other programs. We then convert the FNSS topology into mininet topology which can be deployed in the VM. Finally we ensure all the hosts are working by doing a ping test.

5 Tracefile generator

6 Trace segregator

7 Host programs and controller

8 Next steps

References

[1] Efficient Coflow Scheduling with Varys, Mosharaf Chowdhury, Yuan Zhong, Ion Stoica, ACM SIGCOMM, 2014.

[2] Fast Network Simulation Setup *https://github.com/fnss/fnss/*

[3] Team 06 Github Repo *https://github.com/ucsdcse222a/group6*