

CSE 222A PROJECT - GROUP 6

Stanislav Mushits, Sreejith Unnikrishnan, Ruby Pai, Amit Borase, Ritvik Jaiswal

November 24 2015

Coflow scheduler performance verification

Milestone report 2

1 Progress so far

As part of this milestone, we have completed the following tasks.

- Implemented the actor program to create connections between mappers and reducers and start the simulation
- Deploying the master and slave programs on individual mininet hosts, and writing a script to automate this task
- Running varys simulation on a single Mininet[1] host and traditional (without running varys code) simulation on multiple mininet hosts

The current state of the project is as follows-

- Current simulation involves sending fake data from mapper to reducer.
- The Master is the controller which creates connections between mappers and reducers, and starts the simulation.
- The Slaves are either mappers or reducers.
- The Master reads the simulation file which contains the location of the hosts and task files and a field to specify whether the simulation is a traditional one or a varys simulation, and the task file which contains information about all the mappers and reducers and how much data each mapper sends to a reducer.
- Based on this, the connections between mappers and reducers is established, and the simulation is started.

1.1 Varys simulation

- A single mininet host runs the Master and Slaves along with the varys[2] scheduler running on the controller (Master).
- The Master parses the file *simulation.json* to get the location of the *taskFile* (which contains information about all the tasks that need to be performed in the simulation, i.e, all the mapper and reducer indices, and how much data is to be sent from a particular mapper to a particular reducer), and the *hostsFile* (which lists the number of hosts on the first line followed by the IP address of each host).
- *simulation.json* also specifies if the simulation uses varys scheduler or not.
- The Master then sends the tasks to all mappers and all reducers, which then perform the simulation.
- In the case of Varys simulation, the Master initially registers the coflow, i.e, obtain a coflow ID, before sending tasks to the mappers and reducers.

1.2 Traditional simulation

- As opposed to the previous case, the Master and Slaves run on different mininet hosts.
- Since this is a traditional simulation, there is no varys scheduler running, which means the coflows are not registered.
- The Master simply forwards the task file to all the mappers and reducers which proceed with the simulation.

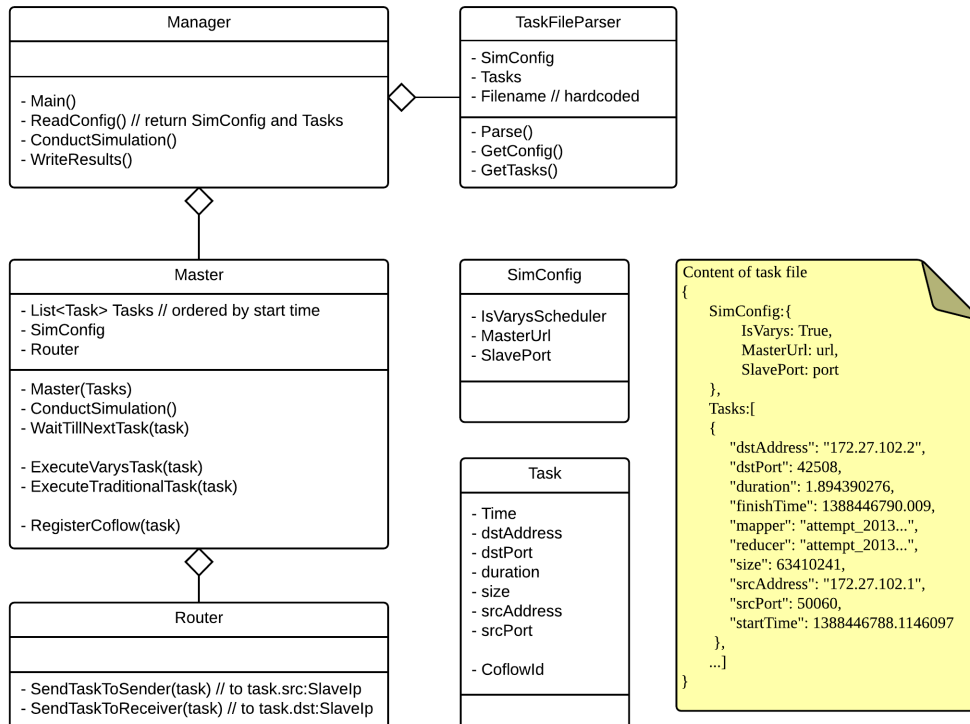
1.3 Challenges Faced

Currently the program runs only on a single mininet host when using the varys scheduler. This is because, running the controller on a separate host to the mappers and reducers results in the coflows not getting registered. Thus Varys simulation is not being able to be run on multiple mininet hosts.

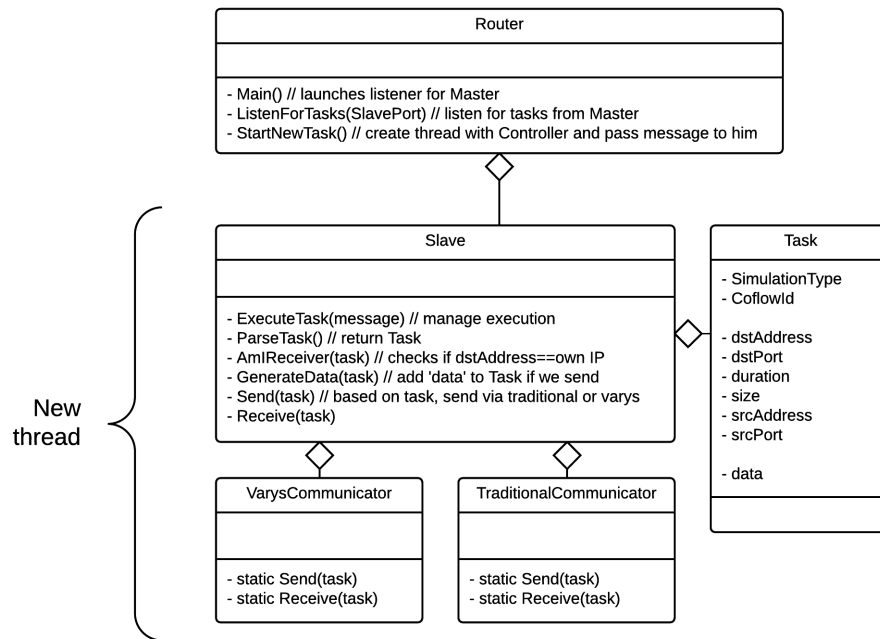
Another challenge has been finding reliable and usable datacenter traces. Traces found on the internet, even if authentic, need to be massaged before they can be used in our project.

2 Class Diagrams

2.1 Master



2.2 Slave



3 Next Steps

Below are the set of pending tasks for the project:

- To get varies scheduling running on multiple mininet hosts, with the controller, mappers and reducers all running on different hosts.
- To run simulations using a realistic datacenter trace.

References

- [1] Mininet: An Instant Virtual Network *mininet.org*
- [2] Efficient Coflow Scheduling with Varys, Mosharaf Chowdhury, Yuan Zhong, Ion Stoica, ACM SIGCOMM, 2014.
- [3] Team 06 Github Repo <https://github.com/ucsdcs222a/group6>