Intel Cloud Orchestration Networking

Fall Progress Report

Matthew Johnson, Cody Malick, and Garrett Smith

Team 51, Cloud Orchestra

Abstract

This document outlines the progress of the Cloud Orchestration Networking project over the entirety of the fall term. It contains a short description of the project's purposes and goals, current progress, current issues, and any solutions to those issues. It also contains a week by week retrospective for all ten weeks of fall term.

CONTENTS

I Project Goals 2

II Purpose 2

III Current Progress 2

IV Week by Week Reports 3

IV-A Background . . . 3

IV-B Weeks Zero Through Two . . . 3

IV-C Week Three . . . 3

IV-D Week Four . . . 4

IV-E Week Five . . . 4

IV-F Week Six . . . 4

IV-G Week Seven . . . 4

IV-H Week Eight . . . 5

IV-I Week Nine . . . 5

IV-J Week Ten . . . 5

V Fall Term Retrospective 6

I. Project Goals

Our project is to first switch the Linux-created GRE tunnel implementation in Ciao to use GRE tunnels created by Open vSwitch. From that point we will switch the actual tunneling implementation from GRE to VxLAN/nvGRE based on performance measurements of each on data center networking cards. After this is completed, a stretch goal is to replace Linux bridges with Open vSwitch switch instances.

II. Purpose

The current implementation of Ciao tightly integrates software defined networking principles to leverage a limited local awareness of just enough of the global cloud's state. Tenant overlay networks are used to overcome traditional hardware networking challenges by using a distributed, stateless, self-configuring network topology running over dedicated network software appliances. This design is achieved using Linux-native Global Routing Encapsulation (GRE) tunnels and Linux bridges, and scales well in an environment of a few hundred nodes.

While this initial network implementation in Ciao satisfies current simple networking needs in Ciao, all innovation around software defined networks has shifted to the Open vSwitch (OVS) framework. Moving Ciao to OVS will allow leverage of packet acceleration frameworks like the Data Plane Development Kit (DPDK) as well as provide support for multiple tunneling protocols such as VxLAN and nvGRE. VxLAN and nvGRE are equal cost multipath routing (ECMP) friendly, which could increase network performance overall.

III. Current Progress

At present, the project is moving along smoothly. Our testing environment has been set up and is networked appropriately. Each Intel NUC (Next Unit of Computing) has Clear Linux installed. Come Winter term, we will get Ciao set up on each machine and begin development on Ciao. Software development on the project has yet to begin as we have just wrapped up the design phase.

Designing has been quite helpful in developing our understanding of the project, its goals, and purpose. Because this is a small component of a very complicated system, taking the time to investigate what a software defined network is, why it's being used, and why we are implementing the piece that we are has been quite beneficial.

While in the design phase, we found an extremely useful library for interfacing with Open vSwitch, libovsdb. Libovsdb is a library written in the Go programming language that allows for simple and efficient calls to the OVS Database Management Protocol. Interfacing with OVS is going to be a very large portion of the project for us, so finding the library is quite the boon. Here is a quick example of how this library functions:

Listing 1. Example insert operation using libovsdb

```
// simple insert operation
insertOp := libovsdb.Operation{
```

```
    Op:  "insert",
    Table:  "Bridge",
    Row:    bridge,
    UUIDName: namedUUID,
}
```

The above example can be reused for all major operations in the OVS Database Management Protocol. Other example operations include select, delete, and update. Using just the operations listed here, we can accomplish most of the needed configuration changes within Open vSwitch.

IV. Week by Week Reports

A. Background

Over the Summer, 2016, Matthew worked as a Software Engineering Intern for the Advanced Systems Engineering (ASE) group within the Open Source Technology Center (OTC) that is in turn within the Software Services Group (SSG) at Intel. Matthew's coworkers had a need to integrate Open vSwitch in their cloud orchestration software (Cloud Integrated Advanced Orchestrator, or Ciao) but did not have the man hours to contribute time to it. Matthew worked with the team to propose the project as a Senior Capstone project at Oregon State University.

Matthew identified two other students, Cody Malick and Garrett Smith, as intelligent hard workers who would benefit the project. Because of this Robert Nesius, the Intel Engineering Manager serving as our client, requested Matthew, Cody, and Garrett specifically for this project.

B. Weeks Zero Through Two

During the first week of class, we all visited Intel in Hillsboro. The principal engineer in charge of Ciao networking, Manohar Castelino, gave us all an explanation of Ciao, how the networking works, and what he expects us to accomplish.

It was also during this time that Rob provided us with five Intel NUCs that would serve as our local cluster. We found out that we needed to register the MAC addresses with the university, and communicated this need to Todd Shechter, the Oregon State University Director of Information Technology.

C. Week Three

During week three we attempted to install Clear Linux OS for Intel Architecture [**?**] on all five Intel NUCs. We were unsuccessful because the installer requires a network connection to download the packaging. At this point, network access had yet to be approved by OSU IT.

Network access is required to install Clear Linux, Ciao, and access the machines remotely. Since Clear Linux is a datacenter OS, not a desktop OS, it does not support wireless internet connections. Because ethernet is required our hardware must be

registered with the university. If we were unable to obtain network access for the hardware on OSU's network we would have needed to find somewhere else to house it.

We also wrote our problem statement in week three, earlier than most groups were able to, since we were ahead of schedule with regard to meeting our team and choosing a project.

D. Week Four

During week four Matthew installed Clear Linux on the Intel NUCs. Since there the networking issues had still not been solved he brought them to his house to use the wired connection there. At this point our hardware had been registered with the university, but for unknown reasons our NUCs were not connecting to the network. Todd Shechter was devoting a lot of his time to help us debug, but we were not yet successful. He set us up with two five-port switches in Kevin McGrath's lab, but they were not receiving IP addresses from the network.

We had our problem statement reviewed and were waiting for feedback by the end of the week. We also started working on the client requirements document, though much of our contributions were simple outline and templating work.

E. Week Five

This week we spent most of our time writing the rough draft for the requirements document. We turned it in by the end of the week and were satisfied with our progress. The final draft of the requirements document was due the next week, week six. This week we also turned in our signed copy of the problem statement.

This week our networking issues were resolved. We had another email conversation with Todd Shechter, who was at a loss as to why we could not access the network from the NUCs. He granted us access to an HP switch we had successfully connected via in the past. On Thursday, we went to move all our NUCs to the new switch, but tried out the network on the mini switches one last time. This time they all worked. Our hardware was now set up and ready to go.

F. Week Six

This week we finished writing the requirements document that was due at the end of the week. After the rough draft we turned in the previous week we continued working on it ourselves until Tuesday. On Wednesday Frank emailed us some suggestions and we addressed those right away. We got the document signed that afternoon by Rob Nesius, our client at Intel.

G. Week Seven

This week we started working on the technical review document due the following Monday. This document outlines nine different components of our system. For each component we explored three different technologies that could be used to implement the component. Since our project is implementing a component of a larger system, it was difficult for us to come

up with nine components and three technologies each (twenty-seven different options). We spent much of our week working together to figure out how to split the project up.

H. Week Eight

This week we submitted the tech review document. It was a lot of work, but we got it in on time unlike many other capstone groups. Garrett researched software switch options, network latency tools, and network throughput tools. Matthew dealt with high-level language, testing, and logging tools. Cody handled the network-specific implementation pieces, such as packet protocols, network virtualization implementation, and bridge implementation.

I. Week Nine

This was a short week with Thursday and Friday given over to the Thanksgiving holiday. We started working on the design document but did not make much headway before breaking for the holiday weekend.

Our team also got together and talked about how we were going to execute the final presentation video for a few minutes this week.

J. Week Ten

This week we focused on the design document due Friday. We spent time researching design strategies and writing up our plan to execute. During this research we found a very useful Go library that interfaces with Open vSwitch. This library will simplify our implementation, allowing us more time to do network performance testing, which the client is very interested in. Our client signed the document with a half-hour to spare before the deadline, and the technical advisors for the project, Manohar Castelino and Tim Pepper, indicated they were impressed with the design we had outlined.

V. Fall Term Retrospective

Week	Positives	Deltas	Actions
0-2	Met the Intel team, studied project goals, purposes	Write project abstract	Research project, write project abstract
3	Started testing hardware setup, problem statement first draft submitted	Resolve networking issues	Contact Todd to get NUC network authorization, write final draft of problem statement
4	Completed hardware setup	Get problem statement signed	Email project owners and get problem statement approved
5	Problem statement submitted, completed first draft of requirements document	Submit final draft of requirements document	Update requirements document, email project owners, get approval via signature
6	Final draft of requirements document submitted	Begin work on tech review	Research technologies for tech review
7	First draft of tech review completed	Finalize tech review	Update and submit tech review
8	Submitted tech review	Begin design document	Research project design steps and implementation details
9	Began work on design document	Complete design document	Fill out the rest of the design document over Thanksgiving weekend
10	Completed design document, began work on final report and final presentation	Complete final report, complete final presentation	Over the weekend, complete final report, create slides for final presentation