UCLH PEACH and NHS Open Source: OpenEHR Architecture and Analytics

Team 38: Report 9

Sandipan Ganguly
Mengyang Wu
Desislava Koleva

Overview of progress :
During the last couple of weeks, our main focus was to deploy the entire workflow that we were previously working on, onto DC/OS on Microsoft Azure. This means that our redeployment of the PEACH system is nearly complete. We completed this step by step starting with Apache Kafka since it takes up the majority of the allocated space in DC/OS. After that we quickly installed Spark. NiFi was a bit more complicated since the UI was directly not accessible from DC/OS.

We also looked into the possibilities for a UI for Kafka and found that Elastic Stack is a visualisation tool that can be used to monitor Kafka logs.

Problems Faced:
The first problem during the deployment of our architecture on DC/OS is installing all the components in one cluster. Although DC/OS provided a well-designed user interface, we confronted countless failures in maintaining all the three services. There are two main reasons for that, one is the capacity of the virtual machines we chose, the other is the configuration of the DC/OS services. We redeployed several times to figure out all the problems and made documents for future reference.

Secondly, we faced the difficulty connecting to the Apache Nifi UI page remotely. We assumed that the Nifi service should work as it did locally, while the fact is we cannot access it with the same way. The URL of Nifi service is hidden by a private IP address, which means we can only access it from internal network. Our solution to this problem is X11 forwarding. We used ssh to login the master virtual machine and installed a browser in the remote server. After that the terminal forwarded the UI of the browser through the ssh tunnel. We also installed a file management UI tools for understanding the file structure.

The last problem occurred before we successfully implemented the Nifi-Kafka flow. The getFile processor in Nifi cannot get file normally because the file system of DC/OS is different from our local machine. To solve that, we hosted another remote FTP server as a data hub.

Successes:

- Fully deployment of a 5-agent DC/OS on Microsoft Azure
- Successful installation of Nifi, Kafka(3 brokers) and Spark
- Access to Nifi UI page through ssh X11 forwarding
- FTP server host for data storage
- Remote workflow from FTP server - Nifi - Kafka

Plan for next two weeks:

No.	Task
1	Finish up with the redeployment of the entire workflow onto DC/OS
2	Install and experiment with Elastic Stack to configure a visualisation tool for Kafka
3	Test our DC/OS workflow with different input data files
4	Go over our progress with Dr Ramachandran and use the feedback for any changes/improvements

Summary of meetings held:

Meeting Date	Who attended	What we did
3/3/17	Sandipan Mengyang	● Attempted to install the individual services onto DC/OS to replicate the workflow from our local machines but failed due to storage issues ● We deployed a larger DC/OS with a larger CPU allocation since the Kafka Brokers used up a lot of the CPU.

		- Discussed the possibilities of accessing NiFi from within DC/OS
6/3/17	Sandipan Mengyang Desislava	- Started work on the slides for the progress check demo - Discussed about inclusion of Druid and Elastic Stack onto our workflow
8/3/17	Sandipan Mengyang Desislava	- Met with our supervisor Dr Harry Strange and discussed our current progress together and the future steps for our project
10/3/17	Sandipan Mengyang Desislava	- Finished up replicating our workflow onto DC/OS with NiFi and Spark integrated.

Individual Contributions:

Sandipan Ganguly

During the last few days, my main contribution was on replicating the workflow that we had on our local workflows onto DC/OS. Specifically, I installed Kafka couple of times to sort out the amount of CPU it uses and realised that the brokers take up most of the space after going through the logs. I also looked into the possibilities for a visualisation tool to manage Kafka and found that Elastic Stack looks pretty viable. In the next couple of weeks, I am going to attempt to install and configure it for our workflow.

I also created the set of slides for the progress check scheduled for the lab on 10th march.

Mengyang Wu

In the last two weeks, I continued working on the deployment of DC/OS on Azure. Fortunately, a significant progress was made based on previous achievements. Overcoming the problems faced before, I successfully installed Kafka, NiFi and Spark on the same cluster and ensured them running continuously. After that I replicated the connecting methods which we used locally to connect the three components in the remote server. During this period I also implemented series technologies to solve new problems. In the following two weeks I will make a summary of our recent progress and discuss with our client about the detailed plan for future.

Desislava Koleva

Over the past two weeks I have mostly been focused on more research-based work related to our project. In particular, although I have not been able to implement Kafka Streams onto my local machine yet, I have continued to read up on how to build distributed stream processing applications with the library. As we are now steadily but surely approaching our goal, it is exciting to see how all the pieces are coming together. Hopefully in the next few weeks we will be able to successfully implement the entire workflow on the DC/OS server. As soon as this task is complete, I will shift my focus onto helping to improve the Random Data Generator UI.