Teaching Big Data and Open Source Software
on Chameleon Cloud

Gregor von Laszewski, Geoffrey C. Fox

Indiana University, Smith Research Center, 2805 E 10th St, Bloomington, Indiana, 47408

laszewski@gmail.com

ABSTRACT

Project CH-818664, KVM: This paper reports on the experience that we gained form using chameleon cloud as part of a course on Big Data and Open Source Software. The course had 62 registered users and used 91% of its allocation with 18195 SUs. The course studies software used in many commercial activities related to Big Data. The backdrop for course contains more than 370 software subsystems from which the students can select. Chameleon cloud was used by the students to conduct a course project that included the creation of a reproducible big data computational infrastructure with DevOps as well as the execution of an application in this infrastructure. A rich variety of 31 projects were conducted. The result of the project with its 31 projects is published in an online class report. We report on success and challenges of chameleon cloud as used in classes. Chameleon Cloud provided the compute resources to make this class possible.

KEYWORDS

Cloudmesh, Chameleon Cloud, Big Data

1 INTRODUCTION

As part of a graduate course at Indiana University that was offered to online and residential students chameleon cloud was used as one of the major computer resources for the class. The course studied software used in many commercial activities related to Big Data. The backdrop for course contains more than 370 software subsystems that have been studied in part and collaboratively in class. The software architecture represented by this collection has been discussed and work towards identifying best practices to deploy, access and interface with them.

Topics of this class will included:

(1) The cloud computing architecture underlying open source big data software and frameworks and contrast of them to high performance computing.
(2) Analysis of software as part of an architecture with its different layers covering broad functionality and rationale for each layer.
(3) Identification on how to create and replicate software environments combining cloud and DevOps technologies.
(4) The main activity of the course included building a significant project using multiple complex subsystems to create a reproducible big data software stack combined with user code and data.

Topics taught in this class are therefore highly relevant for industry as it not only exposes students to theoretical concepts with the topic of big data, but also through practice while having access to clouds and making them usable through DevOps and collaborative

code experiences. Students were using cloudmesh client [1] making it possible to execute the code on multiple different clouds and enabling comparative studies. Figure 1 illustrates that we pursued the theoretical and practical components of this course in parallel.
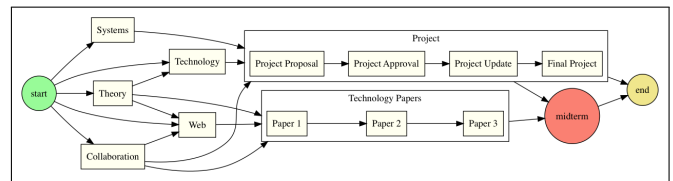

Figure 1: Course Tracks

2 CLOUD USAGE

The course had 62 registered users and used 91% of its allocation with 18195 SUs. From 31 project 27 projects completed while the other 4 are still being completed. As the chameleon cloud allocation for this project expires July 31, we will have to apply for an extension. Furthermore we will run out of compute time for the outstanding projects so we like to ask for an increase in the allocation.

In addition to Chameleon cloud we also used Jetstream and the Futuresystems cloud. Adding these clouds was essential to obtain comparisons of chameleon cloud to other clouds.

3 PROJECTS

The projects conducted as part of this class contained three major portions. First, students had to design a software stack that relates to big data and uses DevOps methods based on ansible to create a reproducible cyber infrastructure. This infrastructure had than to be replicated on other clouds if the project was conducted in a team. Chameleon cloud was recommended to be used as the default cloud. Only after the projects were executable on chameleon other clouds were allowed to be used.

A detailed report of all projects is available online [4]. To provide an overview of the richness of the projects we provide here a list of the titles:

- Remotely Deploying, Visualizing and Controlling a Robot Swarm with ROS and Cloudmesh
- Charge Detection Mass Spectrometry
- Automated Sharded MongoDB Deployment and Benchmarking for Big Data Analysis
- Music Predictive Analysis Project based on Lyrics
- Deploying CouchDB Cluster
- Analysis of USGS Earthquake Data
- Twitter sentiment analysis of the Affordable Care Act in 2017

- Detection of street signs in videos in a robot swarm
- Analysis of H-1B Temporary Employment-Based in Data Science Occupation
- On-line advertisement click prediction
- Flight Data Analysis Using Big Data Tools
- Real-time Analysis and Visualization of Twitter data
- Aviation Data Analysis Using Apache Pig
- Detecting Stop Signs in Images and Videos in a Robot Swarm
- Using Hadoop and Spark for Big Data Analytics: Predicting Readmission of Diabetic patients
- Analysis Of People Relationship Using Word2Vec on Wiki Data
- Amazon Web Services Cloudmesh Extension
- Deploying a spam message detection application using R over Docker and Kubernetes
- Big data Visualization with Apache Zeppelin
- Cloudmesh Docker Extension
- Deployment of Vehicle Detection application on Chameleon clouds
- Head Count Detection Using Apache Mesos
- Optical Character Recognition
- Weather Data Analysis
- Analysis of Airline delays data using Spark and HDFS
- Deployment and performance analysis of a Storm cluster on various cloud environments
- Predicting Customer Churn Using Apache Spark Machine Learning

4 EXPERIMENTAL CONFIGURATION

Conducting a class project with many students provides its unique challenges that are worth while to analyze as part of Chameleon Clouds mission to support educational projects. We will outline here some challenges and important contributions of chameleon cloud.

4.1 Resource Requirements

Based on our rich experience with other clouds such as FutureGrid and Futuresystems we knew that the class projects needed to be limited in order to stay within 20K SUs. Thus we limited each project team to only running 3 virtual machines as part of their project and making sure that the application and data sizes were small in nature while limiting the possible analysis. Furthermore, one of the requirements was to be able to start and stop virtual machines while not encountering too much penalties as this supported the goal of teaching DevOps which not only requires a "perfect" application to be executed, but to develop a software stack that can be reproduced and potentially executed elsewhere. Hence the charge rate of starting and stopping VMs must be small.

Lessons learned: In contrast to jetstream chameleon cloud provided a reasonable charge rate that did not impact the overall allocation adversely. However, we found that it would be useful to have a more clear and prominent description of the charge formula published in order to educate the students better of prudent usage. Jetstream did also not have this information published and after our allocation on jetstream (which was much smaller) expired, we found we weer charged on jetstream a full hour every time a

students started a VM. However as they did this several times per hour on jetstream we ran out of SUs quickly and our allocation was renewed and the accounting was switched off to enable continued use. We hope that chameleon cloud will continue their chrge model and make sure that start of VMS is not taxed at all.

4.2 Capability Requirements

Base don experience we found that offering bare-metal as part of a class such as the one described is not feasible due to the lack of experience of most of the students. The VM model is much more suitable. At time we noticed that chameleon cloud was too busy so it woudl have helped to have access to a queuing system that allows scheduling of vm execution. However, during development waiting is naturally not a good idea and most students recognized when chameleon was too busy or had other interruptions.

4.3 Monitoring Requirements

Due to the many users in this project it is not sufficient to just provide the overall charge. It is important that the project managere can see the individual charges of all users. This service is conveniently offered by Futuresystems and we propose reintegration into chameleon cloud.

4.4 Features offered by Chameleon Cloud

As part of this class we used direct access to the REST interface for OpenStack using virtual machines. The horizon interface was used by some students to just make sure that the VMs were started and had the correct security groups associated. However, the overwhelming majority of the students used cloudmesh client that conveniently provides this information not just for one cloud but many clouds from the commandline. Furthermore, almost all vm management was conducted via the cloudmesh client. This convenient mechanism was superior to the native client offered by OpenStack. GUIs were inconvenient including iPlant and horizon as they do not offer scriptability.

4.5 New software created

As part of this class we improved the cloudmesh client software [1][2] [3] that was essential to the success of the class.

In addition many devops deployments for deploying virtual clusters based on hadoop, docker swarm, kubernetes and others have been developed and are available as part of the open source class repository including the applications and documentation on how to deploy and run them.

4.6 User Management

It is clear that the class has a significant user management challenge. We found that the management for large project could be improved and inspiration from FutureSystems/Grid could be utilized to improve the easy in which project membership could be managed.

The biggest issue however we had with the user management provided across different clouds and high performance compute infrastructure such as XSEDE. As it turns out a broken process is offered to users that have accounts on one or more resources that

provide user management through the centralized user management service for XSEDE, chameleon and jetstream. In certain cases where users have existing accounts on one or more resources it lead to the situation where some users could not access one ore more of these systems. Some students could not identify this issue at all and only after a significant time not being able to login they reported issues. Initially they just thought the process takes time. However, as we found out the process was broken.

Based on our experience with other clouds and infrastructure, students have no issue remembering multiple account names and passwords for different services (this is naturally the nature of the cloud, they have Google, Microsoft, box, ...), but they have an issue of a SSO service offered that does not 100% work. The students gave the recommendation that Chameleon cloud fixes this issue.

4.7 Performance Comparison

We have conducted detailed performance studies on applications comparing chameleon cloud, jetstream and Futuresystems. While Futuresystems is an older cloud and can not compete on a vm level on speed, it still provides a reliable basis for running VMs. The speed between Chameleon Cloud and jetstream VMs is very similar. We ran in both clouds via cloudmesh. Jetstream was accessed in friendly user mode and experienced at time network outages which were in that mode to be expected. IN general chameleon cloud was a very reliable cloud and due to the staff support at University of Chicago we were able to overcome the reported issues.

5 CONCLUSION

We were able to support 62 students as part of the class using chameleon cloud as the main resource. The experience was so good that we in=tend to apply again for vm usage of another class in the fall. A significant amount of software was been able to be developed as part of this class while using DEvOps and making the results freely available. Cloudmesh allowed the class to easily switch between chameleon jetstream and Futuresystems, thus in a case of an outage they could easily continue their work elsewhere. It also allowed performance studies between the clouds. A full report with all completed projects is available online. The biggest issue we had were the untransparant bugs in the SSO framework offered between XSEDE, chameleon cloud and jetstream.

REFERENCES

[1] Gregor von Laszewski. 2017. Cloudmesh Client Framework. Github. (2017). https://github.com/cloudmesh/client
[2] Gregor von Laszewski. 2017. Cloudmesh CMD5. Github. (2017). https://github.com/cloudmesh/cloudmesh.cmd5
[3] Gregor von Laszewski. 2017. Cloudmesh REST Framework. Github. (2017). https://github.com/cloudmesh/cloudmesh.rest
[4] Gregor von Laszewski. 2017. *Projects in Big Data Software and Applications*. Class Proceedings. Indiana University. https://github.com/cloudmesh/sp17-i524/raw/master/project/projects.pdf