Literature seminar – Data engineering

Data engineering, 7.5 HP
1TD075

Junjie Chu
Ying Peng
Mandus Hjelm

Uppsala Universitet

2nd of May 2021

1 Question 1

What is the meaning and importance of contextualization and orchestration, and what are currently available tools and services in this area?

In cloud computing and in the use of a cloud service, some significant features are flexibility and adaptability and for the virtual machine (VM) to have the right settings and configurations. The process where this is done to the VM is called contextualization. [1]. Contextualization is mainly done on a low level, on a specific machine, which means that problems can occur with dynamic applications that need to be handled manually or by an orchestration tool. The authors of the article *Towards a Contextualization Solution for Cloud Platform Services* argue that contextualization has 3 main problems. One problem is that the contextualization tools implemented by the cloud providers are not cross-platform which means no end-to-end automatizing [2]. Here is where orchestration tools like Ubuntu Juju, Puppet, Chef, Kubernetes, Ansible helps to automate that process [1] [3] [4]. For example, Kubernetes: a container orchestration tool that not only is capable of handling containers and seamlessly integrate multiple containers and vertical scale. But also connect multiple nodes to make application easy and be able to easily scale horizontally [4].

2 Question 4

Explain the role and importance of model serving and CI/CD in today's data engineering world. Write down the names and briefly explain four frameworks (two for model serving and two for CI/CD) that offer model serving and CI/CD capabilities.

2.1 The role and importance of model serving and CI/CD

Machine learning with large data sets has a high demand of hardware nowadays. With MLaaS(model serving or machinle learning as a service), developers get access to sophisticated pre-built models and algorithms which would otherwise take an immense amount of time, skill, and resources to build. This means they are able to devote more time to building and focusing on the important parts of each project.

Also, getting a team of engineers and developers with the required skill and knowledge to build machine learning models costs a lot. The ease and the efficacy of MLaaS setups, with the obvious revenue spike they will provide, is a major allure for businesses.[5]

CI can solve the problem of conflicts caused by too many application branches in one development. ”CD” in CI/CD refers to continuous delivery and/or continuous deployment. Continuous delivery helps to solve the problem of poor visibility and communication between development and operations teams.The purpose of continuous delivery is to ensure that the effort required to deploy new code is minimized. Continuous deployment is mainly to solve the problem that manual process slows down the application delivery speed, which overloads the operation and maintenance team. Continuous deployment is based on the advantages of continuous delivery and realizes the automation of the subsequent stages of the pipeline.[6]

2.2 The frameworks of model serving and CI/CD

Model serving:
TensorFlow Serving is a flexible, high-performance serving system for machine learning models, designed for production environments. TensorFlow Serving makes it easy to deploy new algorithms and experiments, while keeping the same server architecture and APIs.
Model Server for Apache MXNet is an open source component built on top of Apache MXNet for serving deep learning models. Apache MXNet is a fast and scalable training and inference framework with an easy-to-use, concise API for machine learning. With Model Server for Apache MXNet, engineers are now able to serve MXNet models easily, quickly, and at scale.

CI/CD
GitLab is a suite of tools for managing different aspects of the software development lifecycle. The core product is a web-based Git repository manager with features such as issue tracking, analytics, and a Wiki. GitLab allows you to trigger builds, run tests, and deploy code with each commit or push. You can build jobs in a virtual machine, Docker container, or on another server.
Bamboo is a continuous integration server that automates the management of software application releases, thus creating a continuous delivery pipeline. Bamboo covers building and functional testing, assigning versions, tagging releases, deploying and activating new versions on production.

3 Question 5

What is the difference between distributed and Federated machine learning? Highlight the architectural differences between the two machine learning environments. Which one of them is more suitable for privacy preserving model training process and why?
Distributed machine learning (DML) mainly deals with the increasingly large amount of training data. These data are usually aggregated and stored in the cloud server where training happens, but when the training process of a large volume of data exceeds the computing power of a single machine, the efficiency will be weakened [7]. DML splits data and/or model into small chunks and placed on different devices, and parallel processing speeds up the process of model training. The classic DML framework includes a central server, some clients, and a data manager, central server and data manager work together to partition model and data into many parts and then distribute learning tasks to the clients. The data partitions may be shared by different clients[8].
Federated learning is introduced by Google in 2016 and their main idea is to build machine learning models on the dataset that are generated by multiple services that

can preventing data leakage [9].A typical Federated learning system contains a central server and some clients. A central server publishes a machine learning task and picks clients to run in each epoch of the training process. The central server sends the model and data to the clients and waits for the response of the training results. Clients train the model with data locally and return the relevant parameters or gradients to the server. The server aggregates all changes and updates the model for the next training period [10].

Federated learning is more suitable for privacy preserving model training process. Federated learning uses devices that can gather data which means there is no data transferred around the system and less risk of data leaks since the less communication. There is no reason for any client-to-client transfers, all update of model parameters will be transferred from client to server, the further reduces the risk of data leaks. In Federated learning, the origin data collected and processed by clients' devices which will not be managed in third-party storage server, which reduced the data leaks since there is no other entity connect to the system and reduce the extra financial cost for data storage and maintenance [10].

References

[1] Django Armstrong et al. “Contextualization: dynamic configuration of virtual machines”. In: *Journal of Cloud Computing: Advances, Systems and Applications* 4.17 (2015).

[2] Django Armstrong et al. “Towards a Contextualization Solution for Cloud Platform Services”. In: *2011 IEEE Third International Conference on Cloud Computing Technology and Science*. 2011, pp. 328–331. DOI: `10.1109/CloudCom.2011.51`.

[3] Red Hat. *USE CASE: Orchestration*. URL: `https://www.ansible.com/use-cases/orchestration`. (accessed: 03.05.2021).

[4] Sahand Hariri and Matias Carrasco Kind. “Batch and Online Anomaly Detection for Scientific Applications in a Kubernetes Environment”. In: New York, NY, USA: Association for Computing Machinery, 2018. DOI: `10.1145/3217880.3217883`.

[5] Mauro Ribeiro, Katarina Grolinger, and Miriam A.M. Capretz. “MLaaS: Machine Learning as a Service”. In: *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. 2015, pp. 896–902. DOI: `10.1109/ICMLA.2015.152`.

[6] Mathias Meyer. “Continuous Integration and Its Tools”. In: *IEEE Software* 31.3 (2014), pp. 14–16. DOI: `10.1109/MS.2014.58`.

[7] Bin Qian et al. “Orchestrating the development lifecycle of machine learning-based iot applications: A taxonomy and survey”. In: *ACM Computing Surveys (CSUR)* 53.4 (2020), pp. 1–47.

[8] Tim Kraska et al. “MLbase: A Distributed Machine-learning System.” In: *Cidr*. Vol. 1. 2013, pp. 2–1.

[9] Qiang Yang et al. “Federated machine learning: Concept and applications”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pp. 1–19.

[10] Sheng Shen et al. “From distributed machine learning to federated learning: In the view of data privacy and security”. In: *Concurrency and Computation: Practice and Experience* (2020).