Big Data Technologies

Assignment 3

Overview

Part 1 Goals

- Revisit and further explore the concepts of the lectures. Consider prevalent and emerging technologies.

Part 2 Goals

- Experiment with Spark and Koalas. In the tutorials we worked with aggregating data and using the Spark Structured API. The goal now is to consider an ML workflow with a simple implementation.

Submission

- Answers to be submitted on SUNLearn.

Part 1: Technologies

Question 1: Cassandra and CAP [7] Suppose you have a number of models which are serving recommendations that need to be stored for auditing and analysis purposes. This is a high-volume write workload and Cassandra has been chosen as the storage database.

1. As it relates to the CAP theorem, what type of database is Cassandra (AP, CP, CA, CAP)? [1]
2. Suppose you have a three node cluster in a single data centre with consistency level `LOCAL_QUORUM=2` and a replication factor (RF) of 3. How many nodes may fail without affecting operations? [1]
3. Suppose your recommendation system is distributed across multiple regions (or data centres) with `RF=3`, and you want your write operations to be as fast as possible but fault tolerant. Which consistency level would you choose, how many replicas need to respond in that case, and is a read from a client guaranteed to return the latest data? [3]
4. What type of architecture does Cassandra embody (e.g. primary/secondary)? [1]
5. Suppose you are running your Cassandra cluster in a data centre with 40GbE connections, 256GiB RAM, 32 Core Xeon processors, 1TB rotational disks, `RF=1` and a consistency level of `ONE`, but your cluster is suffering from slow write speeds. What would you change? [1]

Question 2: Feature Stores [9] Consider the Uber Machine Learning diagram discussed in the lectures. Suppose you are retailer that is building models on customer data (e.g. demographics, loyalty data, transactional data, etc.). These data are stored in operational databases, obtained via extract, transform and load processes, and ingressed to a data lake in Google Cloud Storage.

Your data science team is currently producing features for models, mostly in isolation. However, it has been suggested that these efforts be consolidated into a central feature store concept.

1. If a general machine learning pipeline comprises the stages (in order); data acquisition, data preparation, model training, model evaluation, model serving, then:
    1. Which stages are simplified by the feature store? [1]
    2. Which stages are consumers of the feature store (in a general case)? [1]
2. We discussed discoverability as a means to share data in big data settings, how are curated data tracked and shared within these settings? [1]

3. Evaluate the *online* and *offline* feature serving/storing mechanism of Hopsworks and Feast:
    1. What underlying technologies are used for storing feature data and which database categories do they fall into (e.g. document-wide)? [2]
    2. Create a comparison table listing three relevant properties/characteristics of the online storage technologies used by Feast and Hopsworks. Provide a third column containing a brief motivation for the relevance or significance of the property. [3]
    3. What streaming infrastructure do they use? [1]

Question 3: Spark [9]

1. A Spark job is broken into stages, and stages into tasks. If a task processes a partition, then what is the ideal number of partitions required to achieve performant parallelism. Explain your answer. [2]
2. Suppose you wish to join a very large dataset (say 1 TiB) with a very small one (less than 100MiB); how would you approach this in order to conserve cluster resources? [2]
3. Suppose you are interested in using managed Spark services in the cloud. Create a comparison table describing Amazon EMR and Google's Dataproc and select three characteristics that you deem the most important. Provide a another column very briefly explaining why your selection is relevant. [3]
4. What operations are subject to lazy evaluation and what is the utility of it? [2]

Part 2: Analysis Tools

Question 4: Koalas and PySpark [15] Your team is primarily using pandas for their data science work but the data engineering team is using Spark to ingress and transform data in the data lake. It has been suggested that you consider migrating to Spark for use cases that require memory to scale beyond a single instance. To this end, you have decided to embrace the Koalas package as a means to bridge the gap between pandas and PySpark.

1. Why would your team not leverage Dask? Provide three key reasons why not and explain their significance. [3]
2. Consider the code below in pandas and Koalas. Suppose you have an RDD spanning three executors (on different nodes). What does Python's builtin `max` do insofar as the data at each of these executors is concerned. What operation/instruction is similar in its action within the PySpark API? [2]

```
import pandas as pd
max(pd.Series([2,3,5]))

import databricks.koalas as ks
max(ks.Series([2,3,5]).to_numpy())
```

3. A core feature of Spark is that RDDs are immutable. However, pandas dataframes are mutable. How does Koalas solve this problem? [2]
4. Consider the home-credit notebook using pandas and implementing a random forest. Your team wishes to use this implementation to test PySpark and the Koalas API.
    1. Port this pandas notebook to PySpark and Koalas. Make every effort to leverage the Koalas API as best you can (it may not always possible). A valid port of the code reproduces the accuracy metric. [5]
    2. Create a pair plot of the external features (`EXT_SOURCE_[1,2,3]`) using Koalas (add this cell at the bottom of the notebook). [2]
    3. How does Spark manage visualisation of large data via the Koalas API. [1]

Question 5: Graph Technologies [18] Your team has identified graph analytics as a means to understand the relationships between buyers and sellers (wallets), and projects and NFTs. Neo4j has been selected as the possible technology to explore these relationships. You have decided to explore the approach by limiting yourself to a single project at first, namely Cryptopunks.

1. Engage in brief exploratory data analysis to understand the data (as it relates to the relationships mentioned above). Document your findings in a notebook. [2]
2. Devise a data model capable of representing the relationships described above. Provide a description and image of the model in a general case (see Neo4j documentation). [5]
3. Load the data into the database. [2]
4. Which wallets hold the largest number of assets/Cryptopunks? Provide your query as well as the top five wallets. [2]
5. Are there any cycles? [2]
6. Which Cryptopunk was traded the most and which wallets held it over time? [2]
7. Apply the Pagerank algorithm to identify significant wallets. Interpret the result. [3]

Hints:

- The first question will require iteration as you attempt to answer the subsequent questions, you may find that your model is inadequate and may wish to update it. Consider finalising that question after you are satisfied that you can address the data retrieval questions.
- Consult the Neo4j documentation whenever possible. You are able to spin up a sandbox instance and it is possible to connect to it from Google Colab. If you are working locally, consider using the Neo4j docker container.
- You can use *Cipher* directly in a Jupyter notebook, or the Python API.
- Visit Etherscan if you want more details on transactions.