3. Provide short answers (in one to two paragraphs) to the following questions

a. Compare and contrast an Apache Spark data set with a data frame.

DataFrame and Dataset are all distributed collection datasets on the spark platform which are used for facilitating the processing of very large datasets. Both of them supports caching of frequently accessed data, which is helpful when data needs to be reused. Hence reducing the amount of time to complete the similar operation. They also have many common functions like sum, avg, map, filter and groupBy.

Datasets can be regard as a special case of a DataFrame, the main difference is that each record in Datasets stores a strongly-typed JVM object instead of a untyped JVM object. DataFrame is a collection of Datasets[Row], where a 'Row' is a generic untyped JVM object, we cannot know which data and what types of data in the 'Row'. Unlike DataFrame, the type of each row is variable in Datasets, and it is easy to get the information of each row after we define the case class.

b. Compare and contrast reservoir sampling with bloom filter.

Bloom Filter is a space efficient random data structure that uses bits to represent a dataset concisely and to determine whether an element belongs to it. It is a fast-probabilistic algorithm to determine whether an element is in the dataset. However, there is a small probability of error. That is to say, if Bloom Filter returns an answer that an element is not in the dataset, then the element is definitely not there. If it judges elements in a dataset, then the element may or may not be there. As for Reservoir Sampling, it is a series of randomized algorithm. Its purpose is to select k samples from the set S which is containing n values, where n is a large or unknown number. It is especially applicable to situations where all n items cannot be stored in main memory.

Compare these two algorithms, we can find that Bloom Filter is not suitable for situations requiring zero error, but Bloom Filter can save a lot of storage space than other common algorithms, such as hash. Because the space occupied by Bloom-Filter is very small, all Bloom Filter can be run in main memory. For example, if we want to query the data stored on

the hard disk. In this case, for most data on the disk, we only need to access the Bloom-Filter in main memory to determine whether it exists or not. If the result of Bloom Filter says that the data is not exist, then we do not need to query on the hard disk. Even when False Position appears, it will only lead to one redundant disk query. Thus, the searching efficiency is increased greatly. As for Reservoir Sampling, it is often used to randomly extract a line from a file without knowing the total number of lines. For a large dataset, the sampled values can be guaranteed to be random. Its algorithm is not very complicated, and like Bloom Filter, it can save a lot of memory. We will use it when main memory is not large enough, and the amount of data we need to process is huge.

c. Discuss the main differences between Apache HBase with Apache Spark.

Compare Spark with HBase. Spark, as a computing engine, is a framework for carrying out big data operations. HBase, as a database, is where big data is stored and read.

Aparche Spark is a fast and execution engine designed for large-scale data processing. It is a common used parallel computing framework, which is Similar to the framework of Hadoop MapReduce. Spark implements distributed computing based on MapReduce algorithm, which has all the advantages of Hadoop MapReduce. However, unlike MapReduce, the intermediate output can be stored directly in the main memory, which reducing the number of read and write to HDFS. Thus, Spark is much more efficient when data are large. In the other hand, Apache HBase is a distributed, open source database. HBase is not like other relational databases. It is an unstructured data storage database. Another difference is that HBase is based on columns rather than rows. From the technical features of HBase, it is especially suitable for writing simple data such as message application of Facebook and querying massive, simple structured data.

d. List the main benefits of integrating Apache Spark with Hadoop HDFS.

Spark and Hadoop, many of the tasks they handle are the same, but they do not overlap each other in some respects. Spark is in some manner designed to enhance rather than replace the Hadoop. There are some benefits of using Apache Spark with Hadoop HDFS. First, Spark enable processing a large amount of data in main memory quickly. For some applications,

Spark promises the performance is 100 times higher than that of Hadoop MapReduce. Spark provides many libraries such as streaming and graph processing functions which can be used in different situations. Spark SQL is also very useful for accessing structured data.

e. Explain how Hadoop implements computational parallelism in terms of the parallel dwarf/s it employs and Flynn's taxonomy.

Dwarfs define basic application domains and / or parallel programming models. There are 13-Dwarfs now. Hadoop employs MapReduce which is one of the 13-Dwarfs as its core component. All the parallel application development on Hadoop is based on the MapReduce programming framework. The process of MapReduce is mainly divided into 2 stages: Map stage and reduce stage. The map stage is implemented first and then the reduce stage. Before the map function is executed, the input needs to be "split" (that is, the massive data is divided into roughly equal "blocks"), and each map function handles a "split". So that multiple map functions can work simultaneously. What map functions to do is to pre-processing data. The output of each record in split is in the form of <key, value> pair. Before entering the reduce phase, the data which has the same key in each map is grouped together and sent to a same reducer. This involves the case where the output of multiple maps matches with multiple reducers. This process called "shuffling". Then, the Reduce function reduces multiple values with the same key. Finally, a series of values for the key become one value after the Reduce function.

Flynn's taxonomy is a highly efficient way of classifying computers. It can be divided into four computer types: SISD, SIMD, MISD, and MIMD. MIMD means multiple instructions and multiple data, each processing unit has separate instructions and data. This is the most common structure in parallel processing systems, and Hadoop is classified into this category.

f. Outline the main design features of the RDD abstraction for in-memory cluster computing?

There are five main features of the RDD in spark.

1. RDD is a list composed of multiple partitions. For an RDD, each partition is processed by a computational task. Users can specify the total number of partitions of an RDD when they create it, and if not, the default value will be used.

2. Every partition has a function to perform operations on it. This is also the foundation of Spark parallel computing. The computation of RDD in Spark is based on each partition.

3. List of dependences on other RDDs. It is well known that RDD is based on main memory computing, although memory-based computing can bring faster speed, the corresponding effect is not fault-tolerant. As long as one process fails, all data in main memory is lost. Thus, this feature is also a guarantee for RDD's fault tolerance for in-memory cluster computing. In spark parallel computing, each transformation of RDD generates a new RDD, so there is a pipeline-like dependency between RDDs. Spark can recalculate the missing partition data through this dependency instead of recalculating all partitions of the RDD when partition data is lost.

4. We can re-partition each partition. But the prerequisite for this feature is that RDD must be in the form of <key, value> pair. This feature is optional.

5. A list of priority positions of each split is calculated on. For a HDFS file, this list holds the location of the block where each Partition is located. This feature is also optional.