Big Data Tools 1003 (2021F-T1)

Assignment 2

Submitted by: Aadarsha Chapagain

Assignment 2: Provide a 3-page summary together on Spark, Kafka, HBase, and Cassandra

Spark

Spark is unified analytical engine which is used in processing of large data sets. Spark is based on Hadoop Map reduce mechanism and it does the work more efficiently. Spark offers the more types of computation which include interactive queries and batch processing. The main reason behind the increased speed of Spark is it's in memory cluster Computing.

Spark was designed with the additional functionality for streaming, interactive queries, and iterative algorithms. Some of the features of Apache Spark are

Speed

Spark stores the intermediate data in memory which increases it speed by large factor. Reducing the read and write operation on disk helps spark to increase its speed

Multiple language Compatibility

Spark application can be written in different language. It provides built in API for python, java, and Scala.

Advance Analytics

Beside Map reduce, It also apports streaming, SQL queries, Graph algorithm and Machine learning (ML)

There are three ways of deployment of Spark

- Stand Alone
- Hadoop Yarn
- Spark in MapReduce (SIMR)

Kafka

Kafka is open-source software platform developed to manage the event Streaming or real time data feeds. It is written in Scala and java. Event streaming is the process of collecting or capturing the data from various sources like database, IOT, cloud Services in the form of the events. These recorded events are used for analytics, real-time processing, replying to these events or just route

these feeds or event to different location. Kafka implements the data pipeline which ensures the continuous flow of data wherever and whenever needed.

The two major components in Kafka are server and clients

Server

Cluster of one or more servers are formed in Kafka. Some of them works as a storage layer called brokers, other server run with Kafka connect to either connect to other Kafka cluster or continuously import or export data with the relational database or other existing systems. Scalability, fault-tolerance are some of the strong features of Kafka.

Clients

Kafka clients allows you to write the distributed and fault tolerance application in various language such as java, scala, python, C/C++ and it also provide the Rest APIs.

Some of the use cases of Kafka are

- Commit logs
- Real time Messaging
- Event streaming
- Website Activity Tracking

HBase

HBase is non-relational column-oriented database that run on top of HDFS. Since it is not relational database, it does not support structured language such as SQL. HBase is ideal for storing large dataset in the way which provides fault tolerance. So, it is suitable for read/write of large sets of random data and real time processing. Like most of the non-relational data, HBase is horizontally scalable.

HBase is column-oriented and table in it are sorted by row. Table schema must be defined in HBase which contains columns family in the form of key value pair. A column family can contain any number of columns and new columns can be added to the column families any time which gives the flexibility to adapt with the changing application.

Features of HBase

- Linearly scalable
- Atomic read and write which means that if a process is reading or writing then other process are prevented from read/write operations
- Provide Java API for client to use
- Unlike relational database it does not enforce relationship between data

Difference between RDBMS and HBase

- RDBMS deals with SQL whereas HBase deals with NoSQL database.
- There is fixed Schema in RDBS but there is no Schema in HBase.
- RDMS is row oriented, HBase is column oriented
- Compared to HBase, RDBMS has slower retrieval of data
- Usually, RDMS can handle only structured data but HBase can handle all types of data that are structured, semi structured and un-structured data.

Cassandra

Cassandra is column oriented, distributed, No SQL to manage huge amount of data across the commodity server, which is fault tolerance, provides availability.

Cassandra provides dynamo style replication model. As most of the column-oriented database it also implements column family concepts that is grouping of similar or related column under same column family.

All nodes in the cluster have same role in Cassandra they are independent to each other but interconnected. Each node in the cluster can handle independent read and write operation no matter where the data are in cluster. If a node goes down, then the request is handled by another node which provide no single point of failure mechanism.

Features of Cassandra

- It helps to make data distribution easy. Let's suppose there are six node, we can use partition algorithm to calculate the token range for the six node and distribute the data accordingly
- It can manage all type of data structured, semi structured and unstructured data and data structures can be changed according to the requirement
- It is linearly scalable and as we increase the node in the cluster the response time of the cluster will improve.
- Cassandra supports ACID property of transactions that are (Atomicity, Consistency, Isolation, and Durability)