|  |  |  |
| --- | --- | --- |
| **UMBC – CSEE Department** |  | Dr. Waleed Youssef |
| **Data Science Program** |  | youssef1@umbc.edu |
| **Fall 2022** |  |  |
| **DATA 603 – Big Data Platforms** | | |
|  |  |  |
| **Homework #8 – Apache Hive** | | |
|  |  |  |

**Questions:**

1. **[10 points]** Give brief overview of Apache Hive? Why is it used? What are its main components?

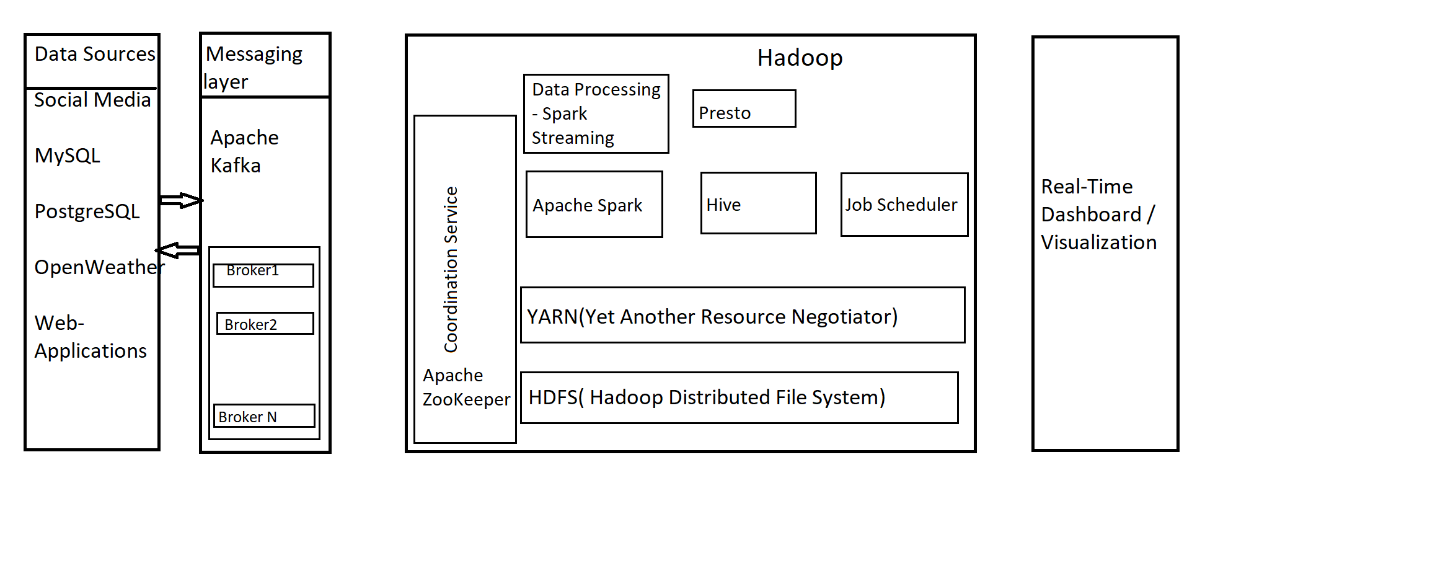
It is a data warehouse component mainly used to compliment the Hadoop File System with its interface. It was originally developed by Facebook but it is not maintained by Apache Software Foundation as Apache Hive. Hive provides SQL intellect in BigData ecosystem so that users can write SQL like queries called HQL or Hive Query Language to extract data from Hadoop. SQL type queries are converted into MapReduce queries by hive.

It can be used for Online Analytic processing (OLAP) since it is scalable, fast, and flexible. It is a great platform for SQL users to write SQL like queries to interact with large data sets that reside on HDFS filesystem.

The main components of Hive are:

* **User Interface:** It is an interface between user and hive. It enables user to submit queries and other operations to the system. Hive Web UI, Hive command line, and Hive HD are different User Interface supported.
* **Meta Store:** Hive has a database in the master node. It chooses respective database servers to store schema or Metadata of tables, databases, columns in a table, their data types and HDFS mapping. This is a Relational database and has structured data.
* **HiveQL Process Engine:** HiveQL is like SQL for querying on schema info on Meta Store. It is one of the replacements of traditional approach for MapReduce program. Instead of writing MapReduce program in Java, we can write a query for MapReduce job and process it.
* **Execution Engine:** The conjunction part of HIVEQL process Engine and MapReduce is Hive Execution Engine. Execution engine process the query and generate results, similar to MapReduce results.
* **Hive Storage ( HDFS OR HBASE):** HDFS or HBASE are the data storage techniques to store data into file system. Others storage such as AWS S3 or Local OS storage can also be used.

1. **[10 points]** Draw an architectural diagram of Hive with Hadoop and Spark? Show all components.



1. **[10 points]** What is the Hive SerDe interface for IO? What is it used for? Describe its benefits?

SerDe is short for Serializer/Deserializer. The Deserializer interface takes a string or binary representation of a record, and translates it into a Java object that Hive can manipulate. The interface handles both serialization and deserialization. It interprets the results of serialization as individual fields for processing. A SerDe allows Hive to read in data from a table, and write it back out to HDFS in any custom format. Anyone can write their own SerDe for their own data formats.

Benefits:

Flexibility: Hive SerDe allows user to use any data format that you want, as long as we can write a SerDe for it.

Performance: Hive SerDe can be very efficient, especially when used with the right data format. Parquet format gives best efficiency when dealing with large datasets.

Portability: Hive SerDe is portable across different Hadoop distributions. We can use the same SerDe with different Hadoop distributions, without having to re-write code.

**Ref**: *Confluence Mobile*. (n.d.). Confluence Mobile - Apache Software Foundation. Retrieved, from <https://cwiki.apache.org/confluence/display/hive/serde>

1. **[10 points]** What is the difference between Hive managed tables and external tables? Give examples?

**Hive Managed tables**:

Hive is responsible for the life cycle of these tables. Managed tables are created by default in hive. That means any table hive do not explicitly specify as an external table, will be created as an internal or managed table.

When we drop managed tables from hive, not only metadata is deleted from Hive but data is also deleted from HDFS.

E.g.: create table test (id int, name string);

Hive keeps managed tables in sub-directory under database directory. We can specify different location but may cause confusion in the future. So if we want to store data in a different location it is better to use External tables.

**External Tables in Hive:**

When we create tables using EXTERNAL keyword, it tells hive that table data is located at a location other than default location in database. Hence, we need to specify the location in the query.

Hive only drops metadata for external table keeping original data at its location.

E.g.: create external table dept\_test (id int, name string) location "path";

**Reference:** Mogal, M. (n.d.). *External Vs Managed (Internal) Tables in Hive*. Analyticshut. Retrieved, from <https://analyticshut.com/external-vs-internal-tables-hive/>

1. **[20 points] Research Question –** Amazon Web Services (AWS) offer a service called Elastic MapReduce (EMR). Perform the following:
   * **[4 points]** Describe what is EMR?

Amazon EMR is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyze vast amounts of data. Using these frameworks and related open-source projects, you can process data for analytics purposes and business intelligence workloads. Amazon EMR also lets you transform and move large amounts of data into and out of other AWS data stores and databases, such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB.

**Reference:** <https://docs.aws.amazon.com/pdfs/emr/latest/ManagementGuide/emr-mgmt.pdf>

* + **[4 points]** What are the benefits of EMR?

**Cost savings:**

Amazon EMR pricing depends on the instance type and number of

Amazon EC2 instances that you deploy and the Region in which we

launch our cluster. On-demand pricing offers low rates.

**AWS integration:**

Amazon EMR integrates with other AWS services to provide capabilities

and functionality related to networking, storage, security, and so on, for

our cluster. E.g.: S3 to store data, IAM(Identity and Access

Management) to configure permissions.

**Scalability and flexibility:**

Amazon EMR provides flexibility to scale our cluster up or down as your

computing needs change. We can resize our cluster to add instances for

peak workloads and remove instances to control costs when peak

workloads subside.

**Reliability:**

Amazon EMR monitors nodes in our cluster and automatically

terminates and replaces an instance in case of failure.

**Security:**

Amazon EMR leverages other AWS services, such as IAM and Amazon

VPC, and features such as Amazon EC2 key pairs, to help you secure

our clusters and data.**Encryption:**

Amazon EMR supports optional Amazon S3 server-side and client-side

encryption with EMRFS to help protect the data that we store in Amazon

S3. With server-side encryption, Amazon S3 encrypts your data after we

upload it.

* + **[4 points]** Compare EMRs to traditional models?

|  |  |  |
| --- | --- | --- |
| **Feature** | **Amazon EMR** | **Traditional Big Data Model** |
| **Type of model** | Cloud-based | On-premises |
| **Management** | Managed by Amazon Web Services (AWS) | Managed by the organization |
| **Cost** | Pay-as-you-go | Organization must pay for buying, improving, or maintaining physical assets required to host Big Data models |
| **Scalability** | Scalable up or down to meet demand. Pay for only the resources used. | Scalable up or down to meet demand but requires high cost. |
| **Security** | Managed by AWS | Managed by the organization |
| **Compliance** | Compliant with industry standards | Compliant with industry standards |
| **Support** | 24/7 support from AWS | Support from the organization's IT department |
| **Updates** | Automatically updated by AWS | Updates managed by the organization |
| **Ease of use** | Easy to use, no prior experience required | Can be difficult to use, prior experience required |

* + **[4 points]** Compare EMRs to Cloudera?

|  |  |  |
| --- | --- | --- |
|  | **AWS EMR** | **Cloudera** |
| Auto Scaling | EMR segregates slave nodes into two subtypes – Core Nodes and Task nodes. This results in high scalability and low cost by using the spot instance for task node. | Cloudera does not categorize slave nodes into core and task nodes. So if a node is removed/lost then there is increase the risk of losing HDFS data. |
| Access to Amazon S3 | You can access data on S3 from EMR directly or through Hive Tables. EMR is highly tuned for working with data on S3. | Cloudera uses Apache libraries (s3a) to access data on S3 .But EMR uses AWS proprietary code to have faster access to S3. |
| Availablity | EMR Service monitors the slave nodes and replaces any unhealthy node with a new node. | Unlike EMR, Cloudera does not categorize slave nodes into core and task nodes. This increases the risk of losing HDFS data in case a node is removed/lost. |
| Ease of Use | AWS manages EMR Hadoop service as well as underlying AWS infrastructure. So you can quickly start a new Hadoop cluster quickly and start processing the data. | Cloudera is comparatively more difficult to learn and configure.But once you have it setup, it’s far more flexible than EMR, and there’s no extra infrastructure cost. Cloudera Manager has an easy to use web GUI. This helps manage and monitor Hadoop services, cluster, and physical host hardware. |
| Hadoop Management Console | AWS does not provide any management console like Apache’s Ambari or Cloudera Manager, for EMR. This makes it difficult to manage and track various Hadoop services on a running cluster. | Cloudera also provides Cloudera Director to enable self-service for using CDH in the cloud. It provides an administration experience for central IT to reduce costs and deliver agility. There is interface for end-users provisioning and scaling clusters. |
| On-Premise and Cloud Options | AWS does not provide the on-premise option and rely on the other Amazon services. | Cloudera offers both on-premise and on-cloud options. This helps reuse the on-premise expertise – experience, human resources, and learnings. |

* + **[4 points]** When would on-premises solutions be better than EMRs
    - On-premise solution can offer greater customization for users as we can have full control over the configuration of big data cluster, which allows us to customize infrastructure to meet the requirements.
    - With on-premise solutions, latency can be reduced as data processing tasks can be executed locally, without the need to transfer data to a clod-based cluster, resulting in faster processing speeds.
    - The cost of on-premise solutions can more predictable and potentially lower than EMR, especially for users with large amounts of data or high-performance requirements, as we can optimize the infrastructure to gain maximum efficiency.
    - On-premise solutions provide us with complete control over the infrastructure, allowing for greater flexibility and customization to meet specific needs. This gives an advantage if there are complex or unique requirements.

**Reference**: <https://www.geeksforgeeks.org/amazon-emr/>