**Comparison of Query runtimes-**

**on AWS Athena, Hive, Spark SQL and Presto on EMR**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Athena** | | **Hive on EMR** | | **Spark SQL on EMR** | | **Presto on EMR** | |
| **Spicepay Client Query** | **Run 1** | **Run 2** | **Run 1** | **Run 2** | **Run 1** | **Run 2** | **Run 1** | **Run 2** |
| SELECT count(log\_date\_time) FROM "spicemoney"."raw\_aeps\_trans\_res" where year(log\_date\_time)=2017 and month(log\_date\_time)=12 | 2.09s | 0.81s | 24.1s | 22s | 28.01s | 22.2s | 25.71s | 23.93s |
| SELECT count(log\_date\_time) FROM "spicemoney"."aeps\_trans\_res" where year='2017' and month='12' | 3.90s | 1.57s | 20.1s | 15.12s | 20.19s | 6s | 26.13s | 11.18s |
| SELECT count(log\_date\_time) FROM "spicemoney"."aeps\_trans\_res\_year\_month" where year='2017' and month='12' | 2.29s | 0.94s | 7.35s | 4.23s | 3s | 1.5s | 6.35s | 3.10s |
| SELECT count(log\_date\_time) FROM "spicemoney"."aeps\_trans\_res\_year" where year='2017' and month(log\_date\_time)=12 | 3.32s | 1.27s | 7.48s | 5.35s | 2.18s | 1.31s | 6.48s | 4.15s |
| SELECT count(log\_date\_time) FROM "spicemoney"."part1" | 1.5s | 0.57s | 2.30s | 1.33s | 0.7s | 0.3s | 1.30s | 1.11s |
| SELECT count(log\_date\_time) FROM "spicemoney"."part2" | 1.87s | 0.61s | 1.70s | 1.20s | 0.5s | 0.2s | 1.53s | 1.12s |

**What is AWS Athena?**

Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to setup or manage, and you can start analyzing data immediately. You don’t even need to load your data into Athena, it works directly with data stored in S3. To get started, just log into the Athena Management Console, define your schema, and start querying. Amazon Athena uses Presto with full standard SQL support and works with a variety of standard data formats, including CSV, JSON, ORC, Apache Parquet and Avro. While Amazon Athena is ideal for quick, ad-hoc querying and integrates with Amazon QuickSight for easy visualization, it can also handle complex analysis, including large joins, window functions, and arrays.

**What is Apache Hive?**

Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data query and analysis. Hive provides an SQL-like interface called HiveQL to query large dataset stored in Hadoop’s HDFS and compatible file systems such as Amazon S3.

**What is Presto?**

Presto is a high-performance, distributed SQL query engine for big data. Its architecture allows users to query a variety of data sources such as Hadoop, AWS S3, MySQL, and other relational and non-relational databases. One can even query data from multiple data sources within a single query.

**What is Apache Spark?**

Apache Spark is a unified analytics engine for large-scale data processing. Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance.  
It can run in Hadoop clusters through YARN or Spark’s standalone mode, and it can process data in HDFS, HBase, Cassandra, Hive, and any Hadoop Input Format. It is designed to perform both batch processing (similar to MapReduce) and new workloads like streaming, interactive queries, and machine learning. As can be seen in the above table, this was the quickest to give results.

**Commonalities**

All three projects- hive, presto and spark are community-driven open-source software released under the Apache License. They are distributed “Big Data” software frameworks. BI tools connect to them using JDBC/ODBC and they provide query capabilities on top of Hadoop and AWS S3. They have been tested and deployed at petabyte-scale companies and can be run on-prem or in the cloud. Athena is the one which is fully managed by AWS.

**Differences**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Hive on EMR** | **Presto on EMR** | **Spark on EMR** | **Athena** |
| **Function** | MPP SQL engine | MPP SQL engine | General purpose execution framework | Serverless interactive query service |
| **Processing Type** | Batch processing using Apache Tez or MapReduce compute frameworks | Executes queries in memory, pipelined across the network between stages, thus avoiding unnecessary I/O | Optimized directed acyclic graph (DAG) execution engine and actively caches data in-memory | Uses Presto with ANSI SQL support and works with a variety of standard data formats, including CSV, JSON, ORC, Avro, and Parquet. |
| **SQL Support** | HiveQL | ANSI SQL | Spark SQL | Standard SQL |
| **Usage** | Optimized for query throughput | Optimized for latency | General purpose, often used for data transformation and Machine Learning workloads | Fully-managed query engine |
| **Use cases** | Large data aggregations | Interactive queries and quick data exploration. | General purpose, often used for data transformation and Machine Learning workloads. | Analyze data stored in S3, run ad-hoc queries using ANSI SQL, without the need to aggregate or load the data into Athena. Amazon Athena can process unstructured, semi-structured, and structured data sets. |