TREASURE DATA™

Treasure Data's Analytic Engines: Powered by Hive and Presto

Treasure Data is a cloud service for collecting, storing and analyzing massive volumes of data. Our customers can utilize the power of distributed query engines without any configuration or maintenance of complex cluster systems.

Unlike many large-scale systems, the Treasure Data Service decouples compute from storage, which makes it possible to scale them independently of each other. With this architecture, we have been able to support a wide range of customer workloads. Treasure Data offers multiple analytic engines in the service. Once you load your data, you decide which engine to use based on your requirements. For query execution, we adopted three open source query engines, Hive, Pig and Presto.

Hive Hive translates SQL queries into multiple stages of MapReduce, and it is powerful enough to handle huge numbers of jobs. MapReduce is fault-tolerant since it stores the intermediate results into disks and enables batch-style data processing. Many of our customers issue thousands of Hive queries to our service on a daily basis. A key advantage of Hive over newer SQL-on-Hadoop engines is robustness: Other engines like Cloudera's Impala and Presto require careful optimizations when two large tables (100M rows and above) are joined. Hive can join tables with billions of rows with ease, and should the jobs fail, it retries automatically. Furthermore, Hive itself is becoming faster as a result of the Hortonworks Stinger initiative.	**Best for:** • Large data aggregations • Large Fact-to-Fact joins • Large distincts (aka de-duplication jobs) • Batch jobs that can be scheduled
Presto In some instances, simply processing SQL queries is not enough – it is necessary to process queries as quickly as possible so that data scientists and analysts can use Treasure Data for quickly gaining insights from their data collections. For these instances, Treasure Data offers the Presto query engine. Presto is an in-memory distributed SQL query engine developed by Facebook that has been open-sourced since November 2013. Presto has been adopted at Treasure Data for its usability and performance.	**Best for:** • Interactive queries (where you want to wait for the answer) • Quickly exploring the data (e.g., what types of records are found in the table) • Joins with a large Fact table and many smaller Dimension tables
Pig With SQL-based engines, you describe the answer you are looking for, and the engine determines in what sequence to execute the various stages of the query. However, sometimes you know more about the data than the query engine and you want to control the exact sequence of query processing. For these situations, a script-based language like Pig Latin is more effective. Pig is an analytic engine that uses Pig Latin to describe how to process data. Pig works with relational data (e.g., data that is found in a table). Like Hive, Pig compiles the job into native MapReduce. Therefore, Pig has performance characteristics similar to Hive.	**Best for:** • When you want to control the exact sequence of query execution • Script-based (i.e., Pig Latin) data processing while still working with table-based data

TREASURE DATA™

Treasure Data's Analytic Engines: Powered by Hive and Presto

How to Best Use Hive and Presto

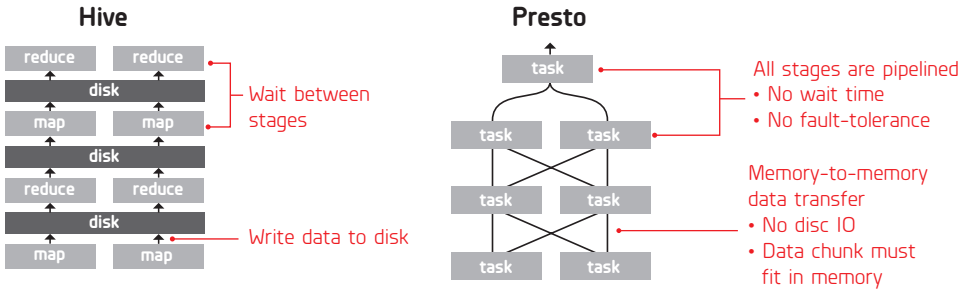
Hive is optimized for query throughput, while Presto is optimized for latency. Presto has a limitation on the maximum amount of memory that each task in a query can store, so if a query requires a large amount of memory, the query simply fails. Such error handling logic (or a lack thereof) is acceptable for interactive queries; however, for daily/weekly reports that must run reliably, it is ill-suited. For such tasks, Hive is a better alternative.

Strengths of Hive and Presto

	Hive	Presto
Optimized for	Throughput	Interactivity
SQL standard fidelity	HiveQL (subset of common data warehousing SQL)	Designed to comply with ANSI SQL
Window functions	Yes	Yes
Large JOINs	Very good for large Fact-to-Fact joins	Optimized for star schema joins (1 large Fact table, many smaller Dimension tables)

In terms of data-processing models, Hive is often described as a pull model, since its MapReduce stage pulls data from the preceding tasks. Presto follows the push model, which is a traditional implementation of DBMS, processing a SQL query using multiple stages running concurrently. An upstream stage receives data from its downstream stages, so the intermediate data can be passed directly without using disks. If the query consists of multiple stages, Presto can be 100 or more times faster than Hive.

Hive vs. Presto



The Treasure Data Difference

Schedule a demo of the service or learn more about Treasure Data:

866.899.5386
info@TreasureData.com

We deliver our unique technology as a prebuilt cloud service, so we get you up and running in days with complete big data processing capabilities. We want you to be successful, spending your time using your data to develop the next big thing—not fighting with your database. We are long-time data practitioners, with a core competency in solving massive data challenges. The Treasure Data Service is transforming how companies use their data. Since the service launched in 2012, thousands use its free Starter version and corporate customers include Pioneer, Pebble, Equifax, GREE and many other companies. Gartner selected Treasure Data for the 2014 "Cool Vendors in Big Data" report.