PROGRAMA DE CIENCIAS DE LOS DATOS

CURSO: BIG DATA

Tarea #2:

PrestoDB

Profesor: MSc. Felipe Meza

Alumno:

Lester Salazar Viales

Randal Salazar Viales

San José, 09 de diciembre de 2019.

PrestoDB

Lester Salazar Viales

Randal Salazar Viales

Historia

Presto fue diseñado y desarrollado en Facebook para que sus analistas de datos ejecuten consultas interactivas en su gran almacén de datos en Apache Hadoop. Fue diseñado específicamente para llenar el vacío / necesidad de poder ejecutar consultas rápidas contra almacenes de datos que almacenan petabytes de datos.

Antes de constuir Presto, los analistas de datos de Facebook confiaban en Apache Hive (que creó y lanzó en 2008) para ejecutar análisis SQL en su almacén de datos de múltiples petabytes en Hadoop.

Hive tuvo un impacto significativo en el ecosistema de Hadoop para simplificar trabajos complejos de Java MapReduce en consultas similares a SQL, al tiempo que podía ejecutar trabajos a gran escala.

Hive era inadecuado para la escala de Facebook y se inventó Presto para llenar el vacío para ejecutar consultas rápidas.

El desarrollo original comenzó en 2012, con un sistema de consulta interactivo que podría operar rápidamente a escala de petabytes. Se lanzó en toda la compañía en la primavera de 2013.

En noviembre de 2013, Facebook abrió Presto bajo la Licencia de Software Apache, y lo puso a disposición de cualquiera para descargarlo en Github.

En 2014, Netflix reveló que usaron Presto en 10 petabytes de datos almacenados en el Amazon Simple Storage Service (S3).

En enero de 2019, se anunció la Presto Software Foundation. La fundación es una organización sin fines de lucro dedicada al avance del motor de consulta SQL distribuido de código abierto de Presto. El desarrollo de Presto continúa independientemente con:

PrestoDB : mantenido por Facebook (*https://prestodb.github.io.)*

PrestoSQL : mantenido por Presto Software Foundation con alguna polinización cruzada de código (*https://prestosql.io.)*.

En septiembre de 2019, Facebook donó PrestoDB a la Fundación Linux que establece la Fundación Presto.

Hoy, Presto se ha convertido en una opción popular para hacer consultas interactivas en Hadoop, y tiene muchas contribuciones de Facebook y otras organizaciones.

¿ Qué es Presto ?

Presto es un motor de consulta basado en SQL que utiliza una arquitectura MPP para escalar. Debido a que es solo un motor de consulta, separa el cómputo y el almacenamiento dependiendo de los conectores para integrarse con otras fuentes de datos para realizar consultas. En esta capacidad, sobresale frente a otras tecnologías en el espacio proporcionando la capacidad de consultar contra:

Bases de datos tradicionales
- MySQL
- PostgreSQL
- Microsoft SQL Server
- Amazon Redshift
- Teradata

Bases de datos no relacionales
- Mongodb
- Redis
- Cassandra
- Apache HBASE

Formatos de archivo en columnas como ORC, Parquet y Avro, almacenados en:
- Amazon Simple Storage Service (Amazon S3)
- Google Cloud Store
- Tienda de blogs de Azure
- Hadoop Distributed File System (HDFS)
- Sistemas de archivos agrupados

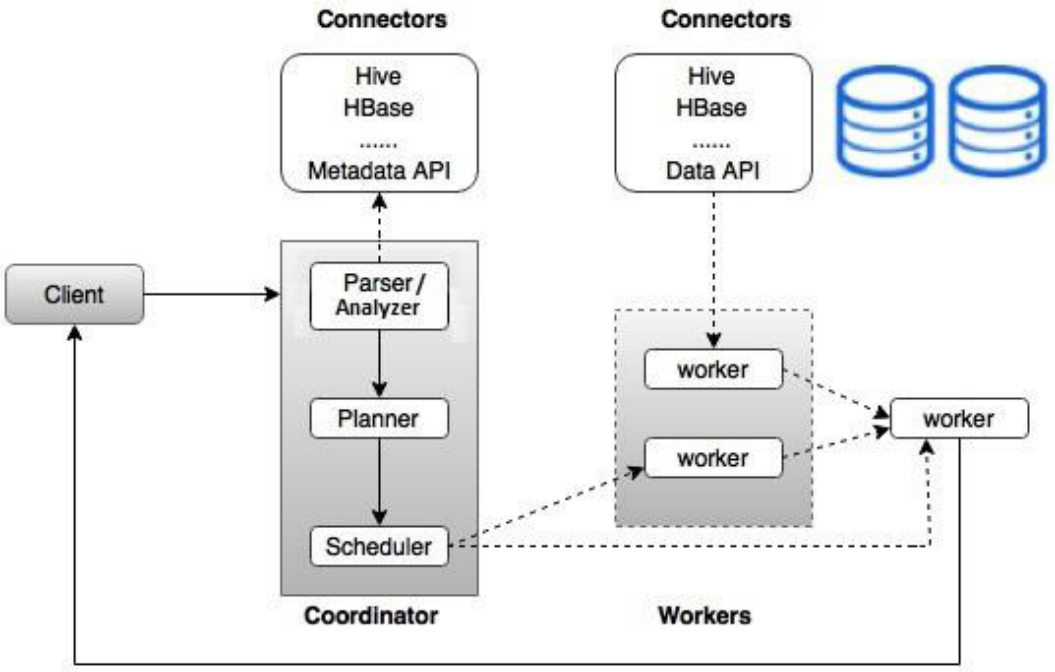
Presto se basa en Java y también puede integrarse con otras fuentes de datos de terceros o componentes de infraestructura.

Presto y Hadoop

Presto es un motor de consulta SQL distribuido de código abierto diseñado para consultas rápidas e interactivas sobre datos en HDFS y otros. A diferencia de Hadoop / HDFS, no tiene su propio sistema de almacenamiento. Por lo tanto, Presto es complementario de Hadoop, con organizaciones que adoptan ambos para resolver un desafío comercial más amplio. Presto se puede instalar con cualquier implementación de Hadoop y está empaquetado en la distribución de Amazon EMR Hadoop.

Apache Presto - Architectura

La arquitectura de Presto es casi similar a la arquitectura clásica de DBMS MPP (procesamiento masivo en paralelo). El siguiente diagrama ilustra la arquitectura de Presto.



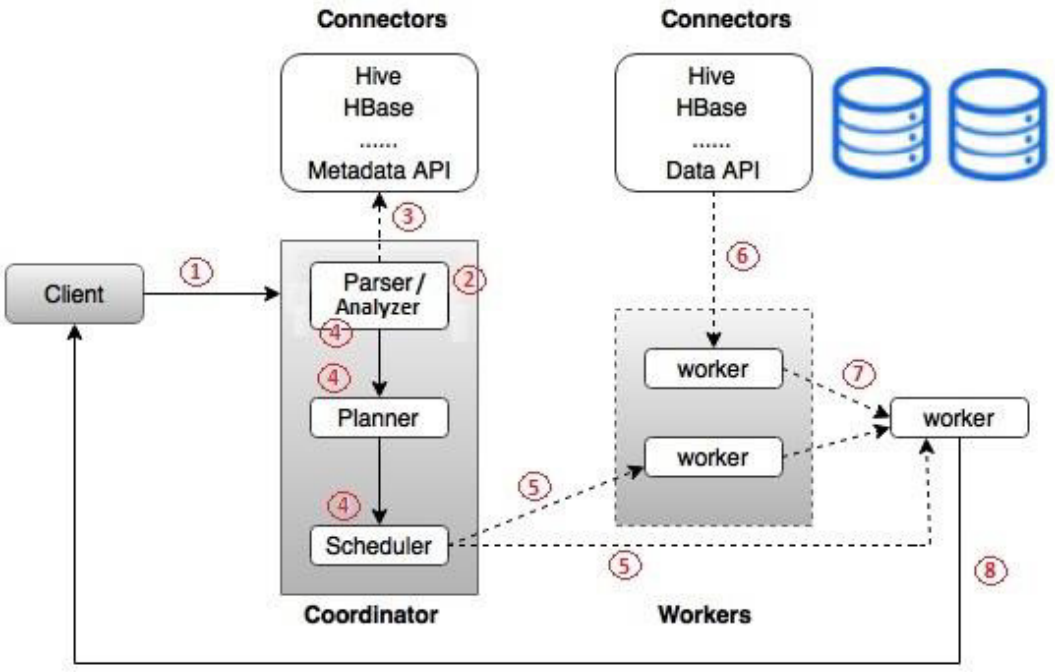
El diagrama anterior consta de diferentes componentes. La siguiente tabla describe cada uno de los componentes en detalle.

No.	Componente & Descripción
1	**Client** El cliente (Presto CLI) envía consultas SQL a un coordinador para obtener el resultado.
2	**Coordinator** El coordinador es un maestro. El coordinador analiza gramaticalmente inicialmente las consultas SQL y luego analiza y planifica la ejecución de la consulta. El programador realiza la ejecución de la canalización, asigna el trabajo al nodo más cercano y supervisa el progreso.

3	**Connector** Los complementos de almacenamiento se llaman conectores. Hive, HBase, MySQL, Cassandra y muchos más actúan como un conector; de lo contrario, también puede implementar uno personalizado. El conector proporciona metadatos y datos para consultas. El coordinador usa el conector para obtener metadatos para construir un plan de consulta.
4	**Worker** El coordinador asigna tareas a los nodos de trabajo. Los trabajadores obtienen datos reales del conector. Finalmente, el nodo trabajador entrega el resultado al cliente.

Presto - Flujo de trabajo

Presto es un sistema distribuido que se ejecuta en un cluster de nodos y utiliza una arquitectura similar a un sistema clásico de gestión de bases de datos de procesamiento masivo en paralelo (MPP). Tiene un nodo coordinador que trabaja en sincronización con múltiples nodos trabajadores.



- El cliente (Presto CLI – **Client** ó usuarios) envían su consulta SQL al coordinador.

- El coordinador inicialmente analiza gramaticalmente las consultas SQL (instrucciones) enviadas. Está diseñado para admitir la semántica ANSI SQL estándar, incluidas consultas complejas, agregaciones, combinaciones, combinaciones externas izquierda / derecha, subconsultas, funciones de ventana, recuentos distintos y percentiles aproximados.

- Con la consulta comprendida, el coordinador utiliza el **conector** (storage pluggin) que está siendo empleado para obtener metadatos (los datos que describen otros datos, el conjunto de datos que describen el contenido informativo de un recurso, de archivos o de información de los mismos). De esta forma, el coordinador conoce dónde localizar los datos que se quieren accesar. El coordinador usa el conector para obtener metadatos para construir un plan de consulta.
- Con esto en mano, el coordinador utiliza un motor de consulta y ejecución personalizado para **analizar**, **planificar** y **programar** un plan de consulta distribuido en los nodos trabajadores.

- El planificador asigna:

    - trabajo a los nodos más cercanos a los datos (**nodos trabajadores**) y supervisa el progreso.
    - designa el **nodo trabajador** (encargado de brindarle los datos al cliente).

- Los nodos trabajadores obtienen los datos solicitados mediante el conector empleado para extraer los datos (Data API).
- Los nodos trabajadores se encargan de trasladar la información extraída al nodo trabajador.

- Finalmente, el nodo trabajador procesa los datos que le fueron facilitados por los nodos trabajadores y entrega el resultado al cliente. El cliente extrae datos del proceso de salida.

Después de compilar la consulta, Presto procesa la solicitud en varias etapas en los nodos de trabajo. Todo el procesamiento está en memoria y se canaliza a través de la red entre etapas, para evitar cualquier sobrecarga de E / S innecesaria. Agregar más nodos de trabajo permite más paralelismo y un

procesamiento más rápido.

Para hacer que Presto sea extensible a cualquier fuente de datos, fue diseñado con abstracción de almacenamiento para facilitar la construcción de conectores conectables. Debido a esto, Presto tiene muchos conectores, que incluyen fuentes no relacionales como Hadoop Distributed File System (HDFS), Amazon S3, Cassandra, MongoDB y HBase, y fuentes relacionales como MySQL, PostgreSQL, Amazon Redshift, Microsoft SQL Server y Teradata.

Los datos se consultan donde se almacenan, sin la necesidad de moverlos a un sistema de análisis separado.

¿Presto está en memoria?

La memoria utilizada por Presto suele estar en el contexto de las JVM (en inglés **Java Virtual Machine**, **JVM**) en sí, dependiendo del tamaño de las consultas y la complejidad de las tareas, puede asignar más o menos memoria a las JVM. Sin embargo, Presto no utiliza esta memoria para almacenar en caché ningún dato.

Modelo de ejecución

Presto admite un motor de consulta y ejecución personalizado con operadores diseñados para admitir la semántica SQL. Además de la programación mejorada, todo el procesamiento está en la memoria y se canaliza a través de la red entre diferentes etapas. Esto evita la sobrecarga innecesaria de latencia de E / S.

¿ Quiénes usan Presto ?


Uber


Dropbox


Nasdaq


airbnb

La implementación de Presto en Facebook es utilizada por más de mil empleados, que ejecutan más de 30,000 consultas, procesando un petabyte de datos diariamente.

En promedio, Netflix ejecuta alrededor de 3,500 consultas por día en sus clústeres Presto.

Referencias

https://en.wikipedia.org/wiki/Presto_(SQL_query_engine)

https://aws.amazon.com/es/big-data/what-is-presto/

https://www.tutorialspoint.com/apache_presto/apache_presto_architecture.htm

https://www.alluxio.io/learn/presto/

https://blog.openbridge.com/what-is-facebook-presto-presto-database-or-prestodb-a-powerful-sql-query-engine-77d4c4a66d4

https://optimalbi.com/blog/2018/05/15/setting-up-prestodb-on-linux/

Link MEDIUM (artículo PrestoDB en Medium):

https://medium.com/@r.salazarvi/prestodb-22c84eca202