Hive Query Performance Patterns

The speed with which a query completes depends upon what operations Hive must perform to execute the query. The sections below present query patterns in order from the fastest to the slowest.

Understanding these patterns requires an understanding of map and reduce phases, so if you have not yet read “Understanding Map Tasks and Reduce Tasks,” you should do so before continuing. How to know what phases a particular query requires may seem mysterious at this time. With practice, you'll get better at distinguishing map tasks from reduce tasks. The next reading, “Understanding Execution Plans,” will give you some tools to help you figure this out, too.

1. Only Metadata

The fastest type of query requires Hive to retrieve only metadata from the metastore, not data from the file system. An example of this is a DESCRIBE command.

```
DESCRIBE customers;
```

2. Fetch Tasks

The next fastest type is a query that executes as a *fetch task*. These are SELECTqueries that do not require the underlying data processing engine. The Hive server executes these queries by fetching data directly from the file system and processing it internally. This eliminates the overhead of starting separate processes to execute the job, which reduces query latency.

```
SELECT * FROM customers LIMIT 10;
```

A fetch task can do more than simply fetching the data and returning it, but there are limitations. To execute as a fetch task, a SELECT statement must not include the DISTINCT keyword and must not use aggregation, windowing, or joins. Also, the input data must be smaller than one gigabyte. Some of these requirements can be changed using Hive configuration properties.

3. Only Map Phase

Next fastest is the type of query that requires only a map phase and no reduce phase. For example, when a query inserts data into another table, Hive executes the query as a map-only job.

```
INSERT INTO TABLE ny_customers

   SELECT * FROM customers
```

```
    WHERE state = 'NY';
```

4. Map and Reduce Phases

Slower yet is the type of query that requires both map and reduce phases, such as a query that performs aggregation. To execute this example, Hive projects and filters the data in the map phase, then aggregates it using the COUNT function in the reduce phase.

```
SELECT COUNT(cust_id)
  FROM customers
  WHERE zipcode=94305;
```

5. Multiple Map and Reduce Phases

The slowest type of query is one that requires multiple map and reduce phases. This example is similar to the previous one, but it also sorts the results and returns only the first 10 rows. Executing this query requires a sequence of multiple map and reduce phases.

```
SELECT zipcode, COUNT(cust_id) AS num

   FROM customers

   GROUP BY zipcode

   ORDER BY num DESC
```

```
LIMIT 10;
```