Simple Strategies For Faster Knowledge Discovery In Big Data

Dr. Ritesh Agrawal  
Lead Data Scientist

Zheng Shao  
Staff Software Engineer
* 6 continents, 70+ countries, 450 cities
* 1st billionth trip took 5 years. 2nd billionth trip took 1.5 years
* 5+ million trips every day
Data-Driven Culture
Infrastructure Challenge

* Help discover

* Quickly: having low query latency

* Efficiently: minimal cost
How to Build An Effective Infrastructure

Data
eg: log presto queries

Analytics
eg: 20% compute resources used by timeout queries

Optimization
eg: build query gate

Forecasting
eg: predict future requirements
Data Analytics Optimization Forecasting

- insert
- select
- with
- others

Percent Queries

Number of Joins

- 0
- 1
- 2
- 3
- 4
- 5+

Pct of Total Queries

Failure Pct in Bucket
90% of compute resources is consumed by 10% most expensive queries.
* 90% of queries using [table A] filter on [column x]

* 70% of queries using [table A] join to [table B] on column X

* 90% of queries using [table A, Table B] end using Column X, Y from table A and Column Z from table B
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
[(col_name data_type [COMMENT col_comment], ... [constraint_specification])]  
[ PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]

  CLUSTERED BY (col_name, col_name, ...)
  [SORTED BY (col_name [ASC|DESC], ...)] INTO num_buckets BUCKETS

  SKEWED BY (col_name, col_name, ...)
  ON ((col_value, col_value, ...), (col_value, col_value, ...), ...)

  [ROW FORMAT row_format]
  [STORED AS file_format]
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
[(col_name data_type [COMMENT col_comment], ... [constraint_specification])] [PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]

CLUSTERED BY (col_name, col_name, ...)
[SORTED BY (col_name [ASC|DESC], ...) INTO num_buckets BUCKETS]

SKEWED BY (col_name, col_name, ...)
ON ((col_value, col_value, ...), (col_value, col_value, ...), ...)

[ROW FORMAT row_format]
[STORED AS file_format]
Optimization > Partitioned By

```
|--user
  |--db
    |--table
      |--date=20170101
      |   |--file_00000
      |   |--file_00000
      |--date=20170102
      |   |--file_00000
      |   |--file_00000
      |   |--file_00000
      |--date=20170103
      |   |--file_00000
      |   |--file_00000
```

```sql
SELECT *
FROM [table]
WHERE
date = 20170102
AND event_type = 'click'
```
SELECT * 
FROM [table] 
WHERE 
    date = 20170102 
    AND event_type = 'click'
Optimization > Partitioned By

```
SELECT *
FROM [table]
WHERE
date = 20170102
AND event_type = 'click'
```
A query that used to take 10 minutes now completes within 2 minutes. 80% reduction in elapsed time
* 95% reduction in resource consumption.
Partitioned By: Key Considerations

* Usage:
  * Identify fields that are often used for **filtering** (FILTER CLAUSE)

* Cardinality:
  * Low Cardinality — otherwise explodes meta-store

* Skewness:
  * data should evenly distributed if all the keys are popular
  * skewness is okie if filter value corresponds to smaller dataset
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.].table_name
[(col_name data_type [COMMENT col_comment], ... [constraint_specification])]

PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)]

CLUSTERED BY (col_name, col_name, ...)

[SORTED BY (col_name [ASC|DESC], ...) INTO num_buckets BUCKETS]

SKEWED BY (col_name, col_name, ...)

ON ((col_value, col_value, ...), (col_value, col_value, ...), ...)
Optimization > Clustered By/Bucketing
Optimization > Clustered By/Bucketing
Clustered By/Bucketing: Key Considerations

* Identify tables that are often joined together.

* Cluster tables on join key.
CREATE [TEMPORARY] [EXTERNAL] TABLE [IF NOT EXISTS] [db_name.]table_name
[(col_name data_type [COMMENT col_comment], ... [constraint_specification])] [PARTITIONED BY (col_name data_type [COMMENT col_comment], ...)] [CLUSTERED BY (col_name, col_name, ...)] [SORTED BY (col_name [ASC|DESC], ...)] INTO num_buckets BUCKETS

SKEWED BY (col_name, col_name, ...)
ON ((col_value, col_value, ...), (col_value, col_value, ...), ...)

[ROW FORMAT row_format]
[STORED AS file_format]
SKEWED BY

* splits data so that heavy values are stored in separate files
* Helps with query optimization.

Storage

* Row vs Col.
* Significant improvement by moving to ORC or parquet format.

Other techniques

* execution engine: MapReduce Vs Tez
* Vectorization
* Pre-joined tables
Key Takeaways

- Fast and efficient infrastructure is key to a business’s success.

- Infrastructure **Optimization** is a constantly ongoing process.

- Infrastructure **Data Science** is key to building an efficient infrastructure.

- Start simple
Thank you

Ritesh Agrawal,
Lead Data Scientist, Uber Inc.