Lecture 33:
• Hive wrap up
• Presto intro

Announcements:
• HW-5 due
• R-5 out
• Exam 2: Friday
• Last Assignment: Out tomorrow ...

Hive Example

Example (†): Generate tables using HiveSQL’s multi-table insert

• Facebook status updates and profile information

• To identify trends in status changes for a day by gender and school

```
FROM (  
    SELECT s.status, p.school, p.gender  
    FROM status_updates s JOIN profiles p ON (s.user_id = p.user_id)  
    WHERE s.ds = '2009-03-20'  
) s1

-- load school summary table
INSERT OVERWRITE TABLE school_summary PARTITION(ds='2009-03-20')  
SELECT s1.school, COUNT(*)  
GROUP BY s1.school

-- load gender summary table
INSERT OVERWRITE TABLE gender_summary PARTITION(ds='2009-03-20')  
SELECT s1.gender, COUNT(*)  
GROUP BY s1.gender

(†) From: Thusoo et al. “Hive – A Warehousing Solution Over a Map-Reduce Framework”, VLDB’09
```
Hive Example

Note on GroupByOperator:

- different Group-By “modes”, e.g., distinct vs non-distinct aggregation
- different “states” of a group by
- basic idea: two pass iterate and merge vs one-pass iterate and terminate

HASH mode: for non-distinct, performs hash-based partial aggregation

MERGEPARTIAL mode: final step for non-distinct aggregations

...various other subtleties and details (which we skipped over)

Hive Improvements

Some Later Improvements to Hive (†):

1. Optimized Record Columnar (ORC): many advantages (already discussed)
2. Improved Query Planner: reduce unnecessary ops and data movements
   (a). Remove unnecessary map phases
   (b). Remove unnecessary data loading
   (c). Remove unnecessary data partitioning

(†) From: Huai et al. “Major Technical Advancements in Apache Hive”, SIGMOD’14
Presto

**Presto**: An (open-source) distributed SQL query engine (†):
- Developed at Facebook to (better) support their SQL analytics
- Development started circa 2012, deployed in 2013
- Used by many companies, contribute back (e.g., Netflix, Uber w/ Parquet)
- Commercial offerings, e.g., AWS Athena (interactive SQL)

A “Cluster-Native” System
- ... 100s of concurrent queries per cluster
- Emphasis on optimization, performs code generation

Query Data Where it Lives
- ... i.e., a “federated” DBMS
- All data comes from data “connectors” (plugins)
- Can read from external DBMSs, do joins between data, etc.

† Sethi, et al. “Presto: SQL on Everything”, ICDE’19

---

Presto Use Cases

Around early 2010’s, more movement towards distributed SQL engines ...
- dedicated, cluster-based systems
- that leverage MapReduce ideas, shared disk architecture, etc.

1) **Interactive Analytics**
- quickly analyze “small” datasets (50GB – 3TB)
- users type in SQL and use tools (visualizations, notebooks, etc.)
- latency sensitive (users waiting for results)

2) **Batch ETL (over data warehous)**
- jobs scheduled via automated pipelines / workflow systems (... Airflow)
- more expensive jobs (volume, processing)
- throughput sensitive
Presto Use Cases

(3) A/B Testing (for UI/product changes)
• collect data and analyze results via interactive tool
• tool involves various statistical tests, many dimensions, time-ranges
• queries generated by tool on the fly (often many joins)
• optimized for these specific, limited query “shapes”
• latency sensitive (w/in seconds)

(4) App Analytics (external user-facing reporting tool)
• offered to app devs that use Facebook APIs
• generate queries are highly selective, large data volumes
• strict latency requires (< 5s)
• high availability, many concurrent queries