Other M/R Interfaces: Hive

EN 600.(3|4)20
Instructor: Randal Burns
17 April 2017
Hive

- Hive: data model and system for data warehousing in map/reduce systems.
- HiveQL: SQL programming for Map/Reduce
  - Not SQL 92 complete
  - No transactions, no materialized views, limited subquery support
- The definitive Hive paper
Hive Example: Status Meme

- Table schema:
  
  ```sql
  status_updates(userid int, status string, ds string)
  ```

- Load log files daily:

  ```sql
  LOAD DATA LOCAL INPATH '/logs/status_updates'
  INTO TABLE status_updates PARTITION (ds='2009-03-20')
  ```
Daily Statistics

- Join logs with profiles and figure out the number of tweets from men/women and by school

```sql
FROM (SELECT a.status, b.school, b.gender
      FROM status_updates a JOIN profiles b
      ON (a.userid = b.userid AND
          a.ds='2009-03-20')
    ) subq1
INSERT OVERWRITE TABLE gender_summary
    PARTITION(ds='2009-03-20')
SELECT subq1.gender, COUNT(1) GROUP BY subq1.gender

INSERT OVERWRITE TABLE school_summary
    PARTITION(ds='2009-03-20')
SELECT subq1.school, COUNT(1) GROUP BY subq1.school
```
How do it go?

- Hive puts tables on HDFS as files and runs queries as Hadoop! jobs
Resulting Query Plan (part 1)

- You don’t need to understand. These are the MR jobs generated by the example.
Resulting Query Plan (part 2)

FROM (SELECT a.status, b.school, b.gender
      FROM status_updates a JOIN profiles b
      ON (a.userid = b.userid AND a.ds='2009-03-20')
    ) subq1
INSERT OVERWRITE TABLE gender_summary
    PARTITION(ds='2009-03-20')
SELECT subq1.gender, COUNT(1) GROUP BY subq1.gender
INSERT OVERWRITE TABLE school_summary
    PARTITION(ds='2009-03-20')
SELECT subq1.school, COUNT(1) GROUP BY subq1.school
Take Aways

● Other ways to program M/R
  – More concise, easier to maintain
  – Particularly for data processing tasks that result in multi-stage map/reduce programs

● Ethos: take the best from DBs
  – Declarative languages and optimization
  – Ad-hoc queries

● Ethos: and leave behind the stuff that’s not parallel
  – Indexes, nested sub-queries
The (Un)Reasonable Debate

- Imperative programming
  - How humans think, step by step
  - Program encodes execution instructions

- Declarative programming
  - What! (Not how.)
  - Allows system to optimize execution
  - Non-intuitive (for many)
  - SQL != declarative programming. It is a specific instance that some love and some hate.

- PIG notable for trying to strike a happy balance
  - DB guys don’t see the upside here