

Hive QL Project

Mindy Jen
Feb. 5th, 2021



Hive QL



- wrote Hive QL to handle structured data
- run Hive QL to execute MapReduce tasks
- Map/Filter [1]: `SELECT _fields FROM _table_1 a JOIN _table_2 b ON a.foreignKey = b.foreignKey WHERE _conditions` --> transformations
- Map/Filter [2]: `GroupBy` (Shuffling), `OrderBy` (Sorting), `SCLFTNs` (length, size) --> transformations
- Reduce: `AGGFTNs` (MAX, MIN, AVG, COUNT, SUM) --> actions (take, collection, foreach, reduce)





MapReduce
Data Processing
& Resource Management

HDFS
Distributed File Storage



MapReduce
Data Processing (dynamic)

Other Data
Processing
Frameworks

YARN

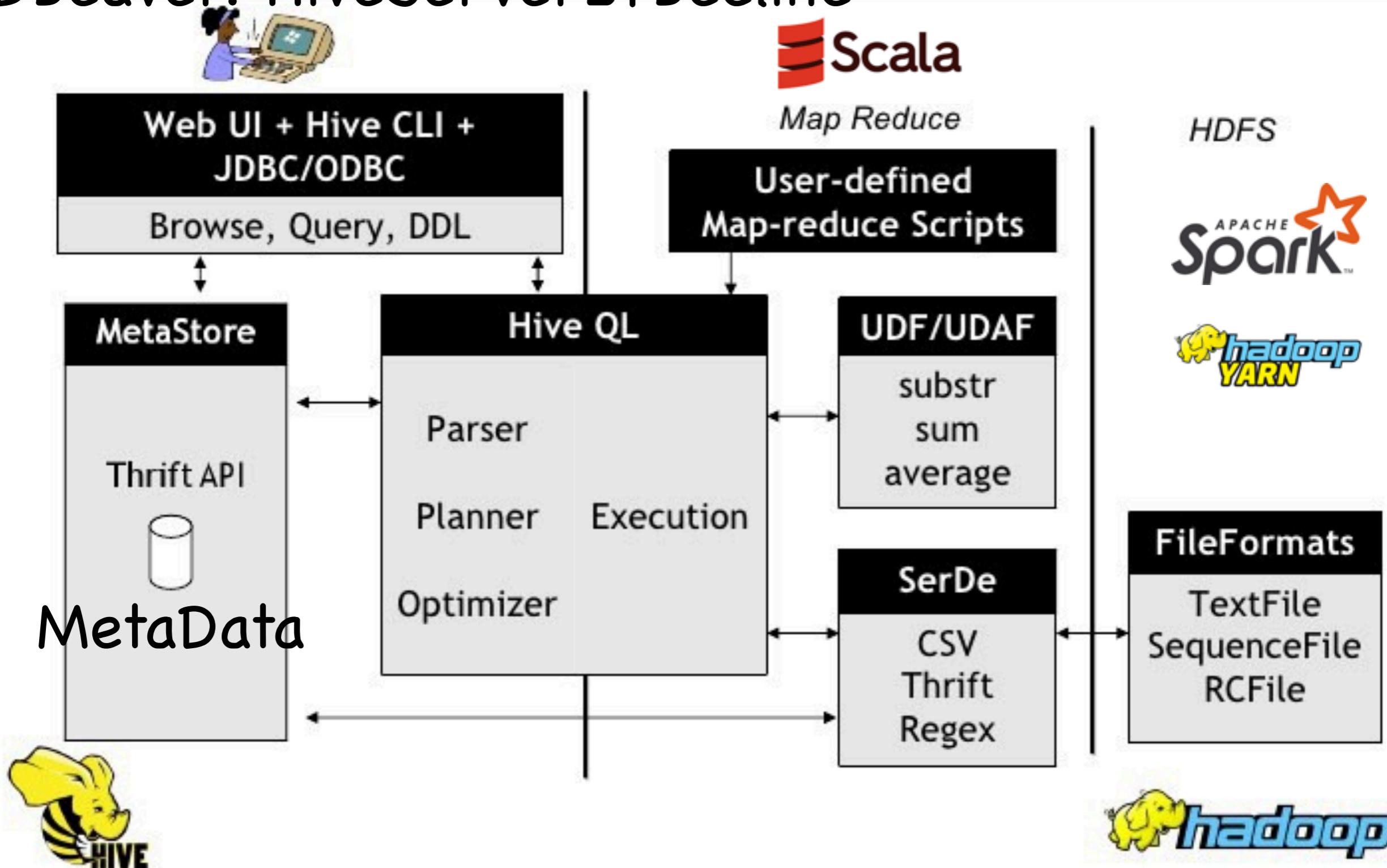
Data Storage (static)

HDFS

Distributed File Storage

Hive Architecture

DBeaver: HiveServer2+Beeline



All Applications

data skim

The image shows a composite screenshot of four different software interfaces. In the top-left, the DBeaver interface is visible with a Database Navigator pane on the left and a SQL Editor pane on the right containing several database queries. The top-center features a 'VIEW CALENDAR' button with a small icon above it. The bottom-left is the MySQL Workbench interface, also showing a Database Navigator and a SQL Editor with similar queries. The top-right is a terminal window titled 'Ubuntu 20.04 LTS' showing the execution of a MapReduce job with logs like 'INFO mapreduce.Job ... completed successfully'. The bottom-right is a Visual Studio Code window displaying Scala code for a 'RandomSampleMapper' class, with tabs for 'RandomSampleMapper.scala', 'SampleDriver.scala', and 'build.sbt'. A large, semi-transparent watermark reading 'data skim' is overlaid across the entire image.

1.Which English wikipedia article got the most traffic on January 20, 2021? [pageview-2021-0120.tsv \(Wednesday\)](#)

⌚ v.0:

SELECT *

Ans: p1q1t1.tsv update_210130

https://github.com/renjmindy/210104-usf-bigdata/tree/main/proj1_0/Q1A

FROM P1Q1T1

WHERE domain_code LIKE 'en%'

ORDER BY count_views DESC

LIMIT 10;

2. What English wikipedia article has the largest fraction of its **readers** follow an **internal link** to another wikipedia article?

clickstream-enwiki-2020-12.tsv & pageview-2020-1229.tsv (Tuesday/after Xmas/before new year)

- ⌚ **model 1:** groupBy - prev/curr/prev+curr

- ⌚ total clicks via internal link-only (external/other)
- ⌚ ratio of clicks via internal link-only (external/other)
- ⌚ ratio of internal link-only (external/other)

Ans: p1q2t2_8.tsv update_210202 https://github.com/renjmindy/210104-usf-bigdata/blob/main/proj1_0/Q2A/p1q2t2_8.tsv

- ⌚ **model 2:** groupBy - prev/curr/prev+curr

- ⌚ ratio of total clicks to total views (internal link required)
- ⌚ presumably page view is uniformly distributed

Ans: p1q7t5_1.tsv update_210203 https://github.com/renjmindy/210104-usf-bigdata/blob/main/proj1_0/Q2A%2B/p1q7t5_1.tsv

3. What series of wikipedia articles, starting with Hotel California, keeps the largest fraction of its **readers** clicking on **internal links**? This is similar to (2), but you should continue the analysis past the first article. There are multiple ways you can count this fraction, be careful to be clear about the method you find most appropriate.

`clickstream-enwiki-2020-12.tsv & pageview-2020-1229.tsv` (Tuesday/after Xmas/before new year)

- **model 1:** requiring `groupBy - prev/curr/prev+curr` be “`Hotel_California`”

- total clicks via internal link-only (external/other)
- ratio of clicks via internal link-only (external/other)
- ratio of internal link-only (external/other)

Ans: `p1q3t1_3.tsv` update_210202 https://github.com/renjmindy/210104-usf-bigdata/blob/main/proj1_0/Q3A/p1q3t1_3.tsv

- **model 2:** requirning `groupBy - prev/curr/prev+curr` be “`Hotel_California`”

- ratio of total clicks to total views (internal link **required**)
- presumably page view is uniformly distributed

Ans: `p1q8t7_1.tsv` update_210203 https://github.com/renjmindy/210104-usf-bigdata/blob/main/proj1_0/Q3A%2B/p1q8t7_1.tsv

3. What series of wikipedia articles, starting with Hotel California, keeps the largest fraction of its readers clicking on internal links? This is similar to (2), but you should continue the analysis past the first article. There are multiple ways you can count this fraction, be careful to be clear about the method you find most appropriate.

⌚ model 3: ...

ongoing... can be done by the end of today (02/05/2021)

4. Find an example of an English wikipedia article that is relatively more popular in the America than elsewhere. There is no location data associated with the wikipedia **page-views** data, but there are timestamps. You'll need to make some assumptions about internet usage over the hours of the day. [pageview-2020-1229.tsv \(week day/after Xmas/before new year\)](#)

⌚ v.0:

- ⌚ 00-05: Asia/Europe
- ⌚ 06-11: America/Asia/Europe
- ⌚ 12-17: America/Europe
- ⌚ 18-23: America/Asia

```
SELECT page_title, SUM(count_views) AS total_view_counts
FROM p1q4t1d20122900
WHERE domain_code LIKE 'en%'
GROUP BY page_title
-- ORDER BY count_views DESC
ORDER BY total_view_counts DESC
LIMIT 600;
```

Ans: p1q4t1_d201229_xx_xx_X.tsv update_210131 https://github.com/renjmindy/210104-usf-bigdata/tree/main/proj1_0/Q4A

5. Analyze how many users will see the average vandalized wikipedia page before the offending edit is reversed. https://dumps.wikimedia.org/other/mediawiki_history/2021-01/ wiki_db = enwiki

- groupBy - page_id

- average over ratios of summed event_user_count

to summed page_revision_count

- Cuts:

- page_is_deleted = TRUE AND
- revision_is_deleted_by_page_deletion = TRUE AND
- revision_is_identity_reverted = TRUE OR revision_is_identity_revert = TRUE

```
SELECT page_id,
       ROUND(Avg(avgSum_users_per_page), 2) AS avg_users_per_page
FROM P1Q5T4
GROUP BY page_id;
```

/wiki_db	/month	/revision_count
/wikidatawiki	/2019-12	/21511762
/commonswiki	/2019-12	/6046688
/enwiki	/2019-12	/4756250
/arwiki	/2019-12	/1599840
/frwiki	/2019-12	/903838
/dewiki	/2019-12	/795638
/eswiki	/2019-12	/710516
/viwiki	/2019-12	/679525
/ruwiki	/2019-12	/652051
/itwiki	/2019-12	/563592

only showing top 10 rows

Ans: p1q5t4_1.tsv update_210202 https://github.com/renjmindy/210104-usf-bigdata/blob/main/proj1_0/Q5A/p1q5t4_1.tsv

6. Self Analysis. https://dumps.wikimedia.org/other/mediawiki_history/2021-01/ wiki_db = enwiki

Entity	Event Type	Meaning	ratio (no cuts)	ratio (cuts)
revision	create	edit	92.76%	0.03%
page	create	create	3.75%	0.00%
page	delete	delete	0.70%	0.00%
page	move	change	0.71%	0.00%
page	restore	undelete	0.004%	0.00%
user	create	register	2.09%	0.00%
user	rename	change	0.005%	0.00%
user	altergroups	change	0.005%	0.00%
user	alterblocks	block/ unblock	0.010%	0.00%

- similar to 5.
- break down to three entities
- split into various event types

Ans: p1q6t1_*.tsv update_210202 https://github.com/renjmindy/210104-usf-bigdata/tree/main/proj1_0/Q6A