Упрощение работы с MapReduce

План

- HIVE
- Impala
- PIG
- HUE

cloudera IMPALA

Эпиграф №1

Решение суперачивки в MySQL

- SELECT country, SUM(payout) FROM log GROUP BY country;

Решение суперачивки с использованием python + hadoop

- Mapper.py
  Import sys
  For line in sys.stdin:
  ….
- Reducer.py
  import sys
  prevkey = "_"
  res = 0.0
  for line in sys.stdin:
  …
- hadoop jar /opt/cloudera/parcels/CDH-5.3.0-1.cdh5.3.0.p0.30/jars/hadoop-streaming-2.5.0-mr1-cdh5.3.0.jar" …

Эпиграф №2

- “Вы даже не представляете себе на что способен аналитик, которому в руки дали SQL”

HIVE

- Движок, позволяющий транслировать SQL-like запросы (HiveQL) в серии Map-Reduce job-ов
- Язык запросов похож на SQL92
- Есть JDBC – коннектор, можно использовать с существующим кодом

DISCLAMER

- HIVE Живой
- Информация может устареть
- Базовые идеи останутся

Хранение данных

- Информация хранится в обычных файлах(на HDFS или S3)
  - Text file
  - Sequence file
  - RDFiles
  - Parquet
- Мета – информация хранится в RDBMS
  - По умолчанию apache derby, но может быть MySQL, Postgres или Oracle

Хранение данных

- Кроме файлов вообще в любом движке, который подходит для MapReduce
- Для нестандартных хранилищ нужно писать UDF

Data Units (HIVE)

- База данных (Не Hbase)
- Таблицы
- Partitions (Не Column Families)
  - Группировка по полю или нескольким.
- Buckets
  - Много разных значений

Пример создания таблицы

```
CREATE TABLE page_view(viewTime INT, userid BIGINT,
            page_url STRING, referrer_url STRING,
            friends ARRAY<BIGINT>, properties MAP<STRING, STRING>
            ip STRING COMMENT 'IP Address of the User')
COMMENT 'This is the page view table'
PARTITIONED BY(dt STRING, country STRING)
CLUSTERED BY(userid) SORTED BY(viewTime) INTO 32 BUCKETS
ROW FORMAT DELIMITED
      FIELDS TERMINATED BY '1'
      COLLECTION ITEMS TERMINATED BY '2'
      MAP KEYS TERMINATED BY '3'
STORED AS SEQUENCEFILE;
```

- LOAD DATA INPATH '/user/data/pv_2008-06-08_us.txt' INTO TABLE page_view PARTITION(date='2008-06-08', country='US')

Пример External table

```
CREATE EXTERNAL TABLE page_view_stg(viewTime INT, userid BIGINT,
                page_url STRING, referrer_url STRING,
                ip STRING COMMENT 'IP Address of the User',
                country STRING COMMENT 'country of origination')
COMMENT 'This is the staging page view table'
ROW FORMAT DELIMITED FIELDS TERMINATED BY '44' LINES TERMINATED BY '12'
STORED AS TEXTFILE
LOCATION '/user/data/staging/page_view';

hadoop dfs -put /tmp/pv_2008-06-08.txt /user/data/staging/page_view

FROM page_view_stg pvs
INSERT OVERWRITE TABLE page_view PARTITION(dt='2008-06-08', country='US')
SELECT pvs.viewTime, pvs.userid, pvs.page_url, pvs.referrer_url, null, null, pvs.ip
WHERE pvs.country = 'US';
```

Create table as select

- Позволяет сохранить результат сложного mapreduce в таблице

ACID

- INSERT/DELETE/UPDATE появились сравнительно недавно.
- Хранятся delta-файлы и используется механизм “compaction” как в Hbase
- Транзакционная модель появилась летом 2014

- Лучше использовать Oracle ☺️

Индексы

- Поддержка ограничена (хотя индексы и появились в последних версиях HIVE)
- В первую очередь стоит пользоваться партициями и бакетами
- HIVE лучше подходит для полного сканирования данных

Join

- Классический Join очень дорогая операция в hive – требует сортировки обоих таблиц в mapreduce
- Лучше не Join’ить большие таблицы

Hive не дает эффективный и полный SQL

Hive позволяет те же MapReduce писать легче/привычнее

MapJoin

- Если одна из таблиц которые надо join’инть – можно использовать MapJoin
- Пример:

```
select /*+ MAPJOIN(time_dim) */ count(*) from store_sales join time_dim on (ss_sold_time_sk = t_time_sk)
```

- Mapjoin на порядки более дешевая операция по сравнению с классическим Join’ом

User Defined Functions

- Можно описывать на внутреннем языке
  - https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF
    - Выбрать 2-е число по возрастанию из 5-ти
- Можно описывать на Java и подключать
  - https://cwiki.apache.org/confluence/display/Hive/HivePlugins
    - Определение страны по IP-адресу
    - Порождающие функции – например брать данные из HBASE
- Есть уже готовые плагины

Checklist

- Данных много (“BigData”)
- Данные в основном только добавляются (не нужно модифицировать)
- Не нужен случайный доступ
- Задачи анализа хорошо описываются SQL-ем

- Идеальный паттерн – хранение и анализ логов

Hive / не Hive

- 1
- 2
- 3


cloudera®
IMPALA

Предпосылки

- Основная проблема HIVE – его “не интерактивность”.
- Используя MapReduce, интерактивности добиться сложно
- Хочется совместить способность переваривать большие данные и интерактивность

Идеи IMPALA

1. Используем хранилище HIVE
2. Отказываемся от хадуповского MapReduce и все операции делаем в памяти
3. …
4. Profit!

Сравнение с HIVE

- Плохо работает, когда данные не влазят в память кластера (суммарную)
- Позволяет добиться интерактивности (в десятки раз быстрее hive) – можно использовать для BI

Как влезть в память

- Партиционирование
- Использование “колоночных” форматов хранения, позволяющих считывать в память только часть таблицы
  - Parquet
  - ORC

PIG

Предпосылки PIG

- Основная идея – упрощение работы с map-reduce (см эпиграф №1 ☺)
- Не все, что можно реализовать в MapReduce, ложится на SQL поэтому HIVE не всегда подходит
- Некоторые люди больше любят императивные, а не декларативные языки
- Нужен специальный язык работы с данными

PIG

- Императивный язык программирования для манипуляции с большими данными
- Одна строчка кода может разложиться на огромный MapReduce job

Примеры использования PIG

```
raw = LOAD 'excite.log' USING PigStorage('\t') AS (user,
time, query);

clean1 = FILTER raw BY
org.apache.pig.tutorial.NonURLDetector(query);
clean2 = FOREACH clean1 GENERATE user, time,
org.apache.pig.tutorial.ToLower(query) as query;
```

STORE clean2 INTO '/tmp/tutorial-join-results' USING PigStorage();

Подробно смотри тут:
https://pig.apache.org/docs/r0.7.0/tutorial.html

HIVE vs PIG

- Чаще дело вкуса
- Hive
  - Аналитика
  - SQL привычка и код
- PIG
  - Для программистов(императивный)
  - Много не стандартных функций

HUE

- Читается как “хью”
- Расшифровывается как “hadoop user experience”
- Представляет из себя удобный пользовательский интерфейс для программ из стека hadoop
- Последние версии включают мощные средства аналитики и визуализации данных
- http://gethue.com/

Возможности HUE

- Просмотр HDFS
- Просмотр хранилища данных HIVE
- HIVE-консоль
- Pig-консоль
- Impala-консоль
- Hbase shell
- Визуализация данных
- Просмотр запущенных Mapreduce программ
- Управление пользователями
- И многое другое

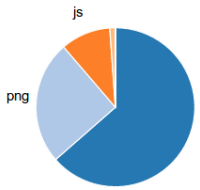
Search | Apache Logs

Marker Map

user_agent_family

- Chrome (354)
- **Sogou Explorer** +
- Other (28)
- IE (17)
- Firefox (1)
- Arora (0)
- Chrome Mobile (0)
- FacebookBot (0)
- **Googlebot** +
- Mobile Safari (0)
- Show more...

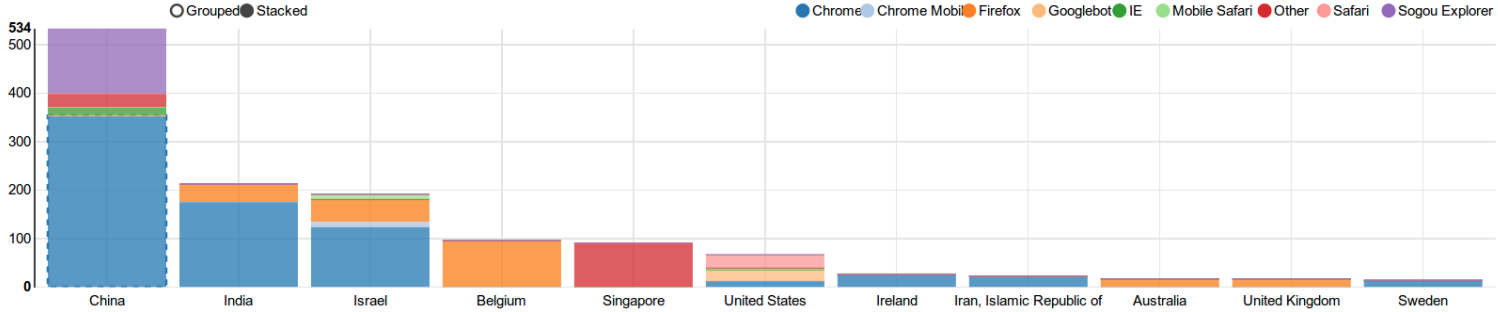
extension



Filter Bar

user_agent_family	app	protocol	country_name:user_agent_family
selected	selected oozie search hbase sqoop	selected HTTP/1.1	selected China:Chrome
excluded Googlebot Sogou Explorer	excluded	excluded HTTP/1.0	excluded

country_name



Grid Results

Filter fields

All (28) / Current (5)

Field Name
version
app
bytes
city
client_ip
code
country_code
country_code3
country_name

bytes field analysis

Terms | Stats | country_code

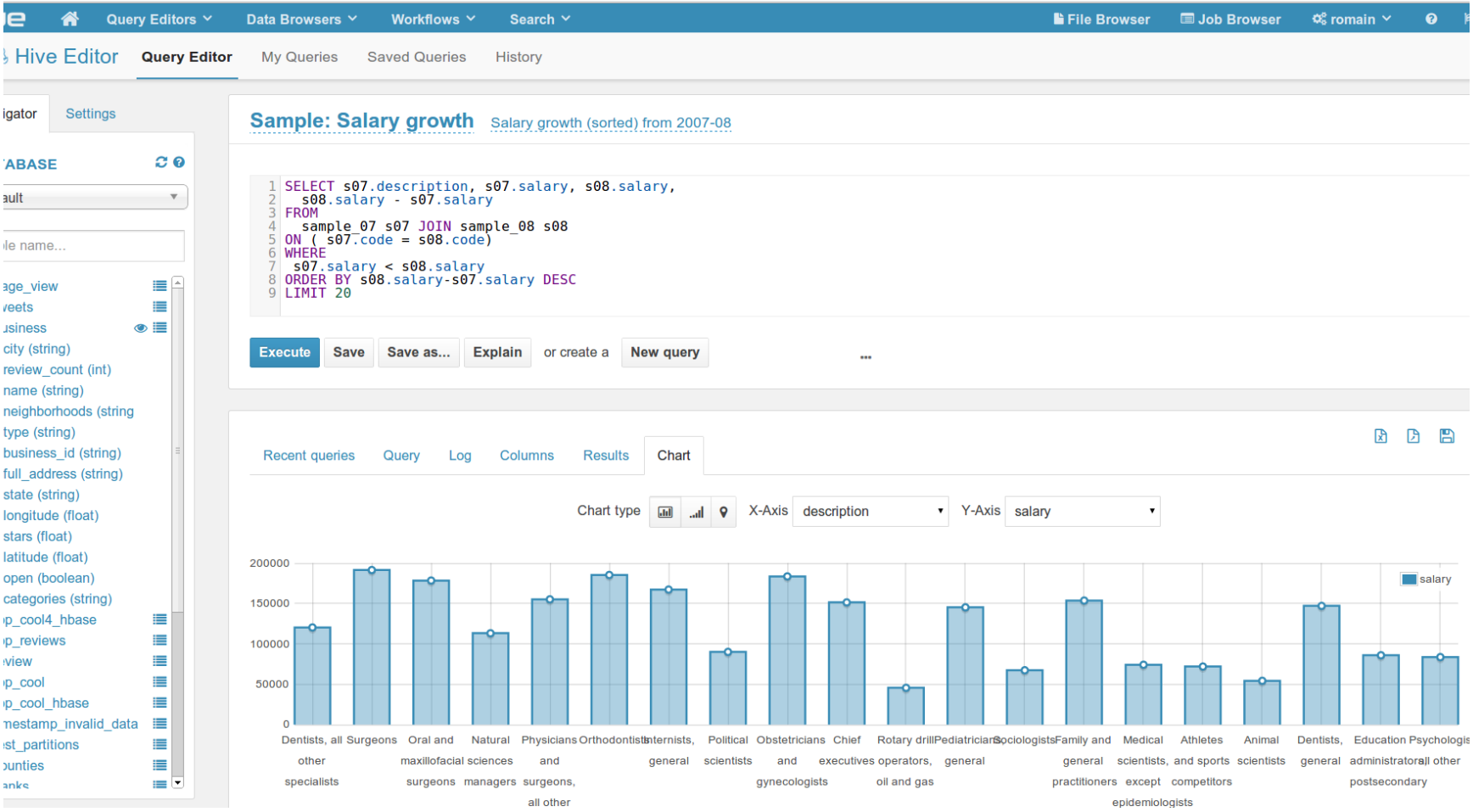
count	354
missing	0
max	45949
sum	822194
min	187
sumOfSquares	17006790050
stddev	6539.740603991179
facets	

CN		
	count	354
	missing	0
	max	45949
	sum	822194
	min	187

f 354 results →

request

229000%22%20TO%20%229999%22%5D%7Cuser_location%3A%22new%20york%22&(
228000%22%20TO%20%228999%22%5D%7Cuser_location%3A%22london%22&query&
228000%22%20TO%20%228999%22%5D%7Cuser_location%3A%22london%22&query&
228000%22%20TO%20%228999%22%5D|user_followers_count:[%22100%22%20TO%20(
229000%22%20TO%20%229999%22%5D%7Cuser_followers_count%3A%5B%22400%2
229000%22%20TO%20%229999%22%5D%7Cuser_followers_count%3A%5B%22100%2
229000%22%20TO%20%229999%22%5D%7Cuser_followers_count%3A%5B%22500%2
229000%22%20TO%20%229999%22%5D%7Cuser_followers_count%3A%5B%22700%2
229000%22%20TO%20%229999%22%5D%7Cuser_location%3A%22espa%C3%B1a%2
229000%22%20TO%20%229999%22%5D%7Cuser_location%3A%22london%22&query&


Query Editors
Data Browsers
Workflows
Search
File Browser
Job Browser
romain
Hive Editor
Query Editor
My Queries
Saved Queries
History
Settings
Sample: Salary growth
Salary growth (sorted) from 2007-08
SELECT s07.description, s07.salary, s08.salary,
s08.salary - s07.salary
FROM
sample_07 s07 JOIN sample_08 s08
ON (s07.code = s08.code)
WHERE
s07.salary < s08.salary
ORDER BY s08.salary-s07.salary DESC
LIMIT 20
Execute
Save
Save as...
Explain
or create a
New query
Recent queries
Query
Log
Columns
Results
Chart
Chart type
X-Axis
description
Y-Axis
salary
200000
150000
100000
50000
0
salary
Dentists, all other specialists
Surgeons
Oral and maxillofacial surgeons
Natural sciences managers
Physicians and surgeons, all other
Orthodontists
Internists, general
Political scientists
Obstetricians and gynecologists
Chief executives
Rotary drill operators, oil and gas
Pediatricians, general
Sociologists
Family and general practitioners
Medical scientists, except epidemiologists
Athletes and sports competitors
Animal scientists
Dentists, general
Education administrators, postsecondary
Psychologis
all other


Query Editors
Data Browsers
Workflows
Search
File Browser
Job Browser
romain
Search
Apache Logs
impala
hive
Thailand (328)
Brazil (289)
Show more...
extension
css
png
js
Grouped
Stacked
Enable selection
China (1743)
United States (5333)
India (1007)
73
60
50
40
30
20
10
0
12:15:49 2014-05-04
01:05:49 2014-05-04
01:55:49 2014-05-04
02:45:49 2014-05-04
03:35:49 2014-05-04
04:25:49 2014-05-04
05:15:49 2014-05-04
06:05:49 2014-05-04
06:55:49 2014-05-04
07:45:49 2014-05-04
country_code3
region_code
user_agent_family
4,595
4,000
3,000
2,000
1,000
0
Chrome
Firefox
Googlebot
IE
bytes
0 - 900000 (8083)
900000 - 1800000 (0)
1800000 - 2700000 (0)
2700000 - 3600000 (0)
3600000 - 4500000 (0)
4500000 - 5400000 (0)
Grid Results
Filter fields
All (28) / Current (4)
Field Name
version
app
Showing 1 to 40 of 8083 results
code
app
request
bytes
302 impala GET /impala/list_designs?q-page=1&q-type=impala HTTP/1.0 446
200 impala GET /impala/list_designs?q-page=1&q-type=impala HTTP/1.0 10839
302 impala GET /impala/query_history?q-type=impala&q-user=rw1hd7z&q-auto_query=off HTTP/1.1 549

Спасибо за внимание