Big data real-time analysis

Weihai Shen
wshen24@asu.edu

What’s for?

Analyze real-time DLU (daily launch user) or DNU (daily new user) data of users for ByteDance

1. **The reference value for product marketing.** Regulate the marketing launch based on real-time performance of each launch channel without having to wait until the next day

2. **Real-time hot events.** Utilize the impact of hot events ahead even immediate news

3. **Expose potential issues as soon as possible**. Cut the loss if any exception of DLU or DNU is triggered especially when a new version is launched

Size of datasets

7 ~ 8 billion within a day and still in increase

Dimensions:

1. Channel
2. Mobile brand
3. Operation system
4. Location
5. ...

Metrics:

1. DLU - Daily Launch User
2. DNU - Daily New User

Size of datasets

Write qps

- 1 million/s at its peak

Read qps

- Relative small, several hundreds/s

Alternatives

Druid

Druid is a column-oriented, open-source, distributed data store written in Java. Druid is designed to quickly ingest massive quantities of event data, and provide low-latency queries on top of the data. **But, druid only provides approximate values for the distinct count due to speed up** [1]. In Bytedance, many analysis measurements are calculated based on distinct users, which are critical to the company, accuracy is the highest priority. Approximate aggregation is unacceptable.

Alternative - cont.

MySQL - data is too huge

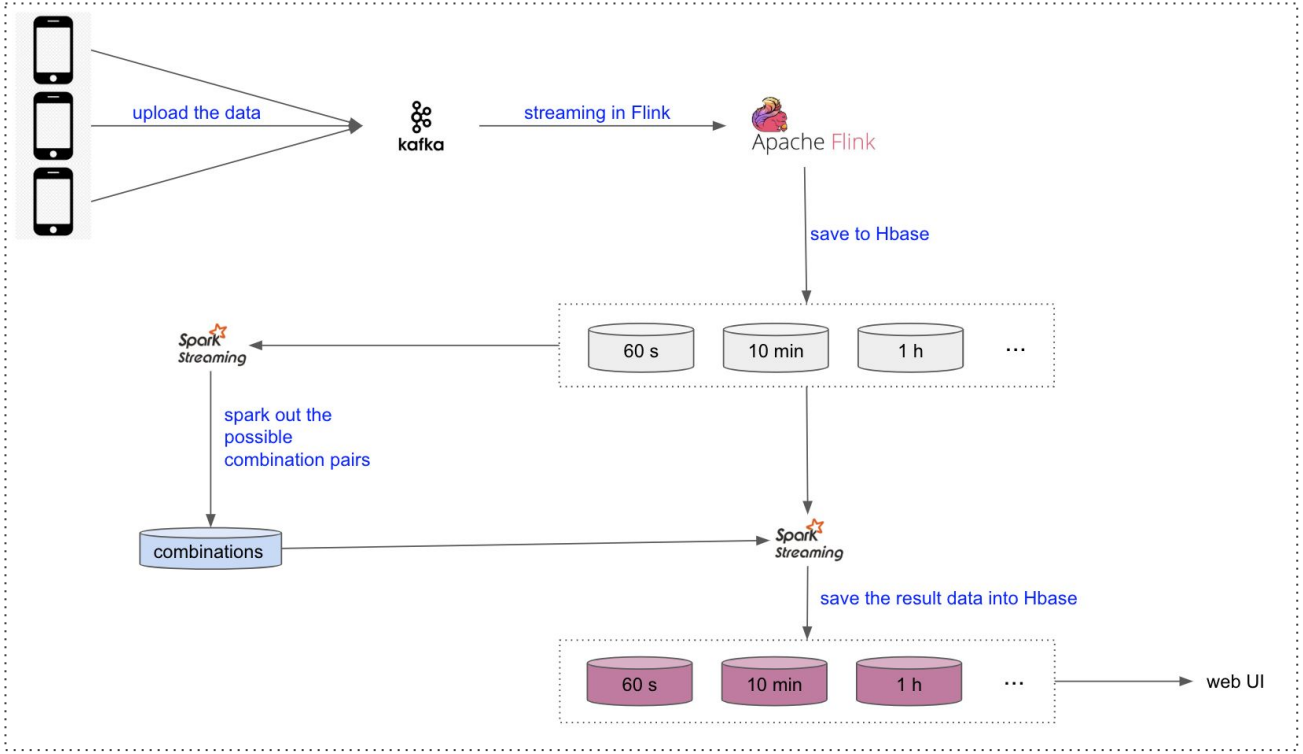
Hive (or hdfs) - Write & Read is too slow

Teradata - Read speed is super slow

Granularity

- 1 minute
- 10 minutes
- 1 hour
- 1 day

Overall design



Overall design - cont.

1. Upload data to the kafka cluster by clients
2. Flink (or storm in the previous system) consumes kafka and then dumps info into corresponding granularity detailed hbase

   {“key”: “23#20170101#13#298789”, “value”: “os:ios, os_version:0.98, brand:xiaomi, …”}

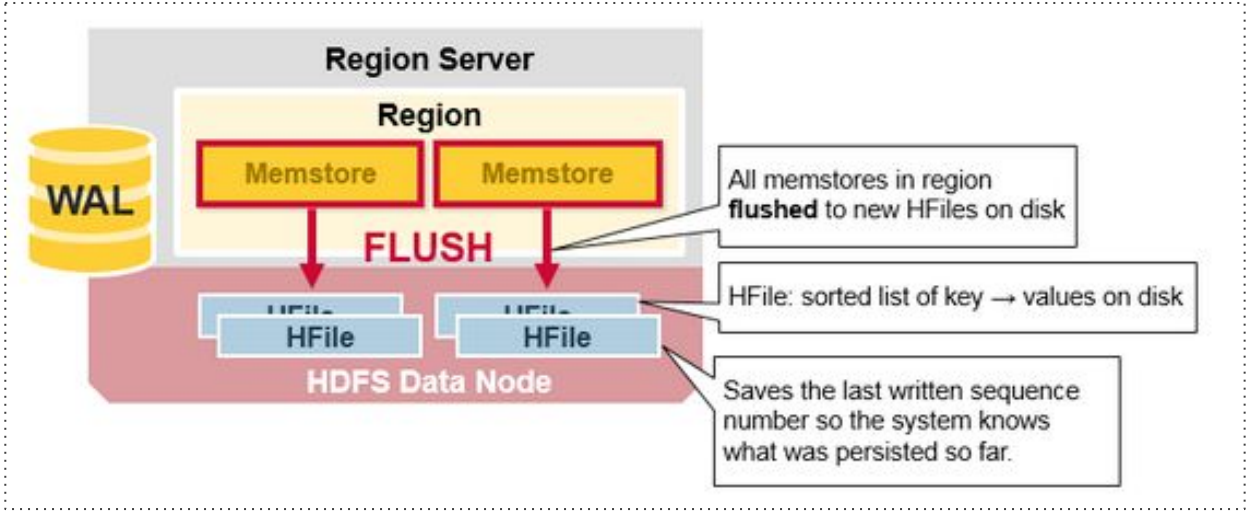
3. Aggregate all possible data from detailed hbase to result hbase

   {“key”: “23#20170101#13#os:android#brand:xiaomi#-”, “value”: “dlu:19023”}

4. API retrieves aggregated data from result hbase
5. Web UI displays the curve

Why Hbase?

- In fact, Bytedance builds a new big-data key-value software based on rocksdb, https://github.com/facebook/rocksdb, similar to hbase
- WAL: Write-Ahead-Log. Great Performance on Writing with huge data



Row_key of detailed hbase

Key

- {salt}#{date_format}#{app_id}#{device_id}

Value

- detailed info

What's salt?

- Distributed the data evenly to different region server
- The info of same device will be distributed to the same region server for spark to aggregate the data in the following steps
- Partition data into 1000 regions

What's detailed info?

- Pattern: brand:Meizu|os:Android|os_version:0.12

Row_key of result hbase

Key

- {salt}#{date_format}#{app_id}#{dimension_whence_str}#{optional time}

Value

- Integer (DLU or DNU)

What's salt?

- {salt} = hash(date_format + app_id + dimension_whence_str} % 10
- Distributed the result data evenly to different region server

What's dimension_whence_str?

- Pattern: dimension_key_a:value|dimension_key_b:value|dimension_key_c:value

Row_key design - cont.

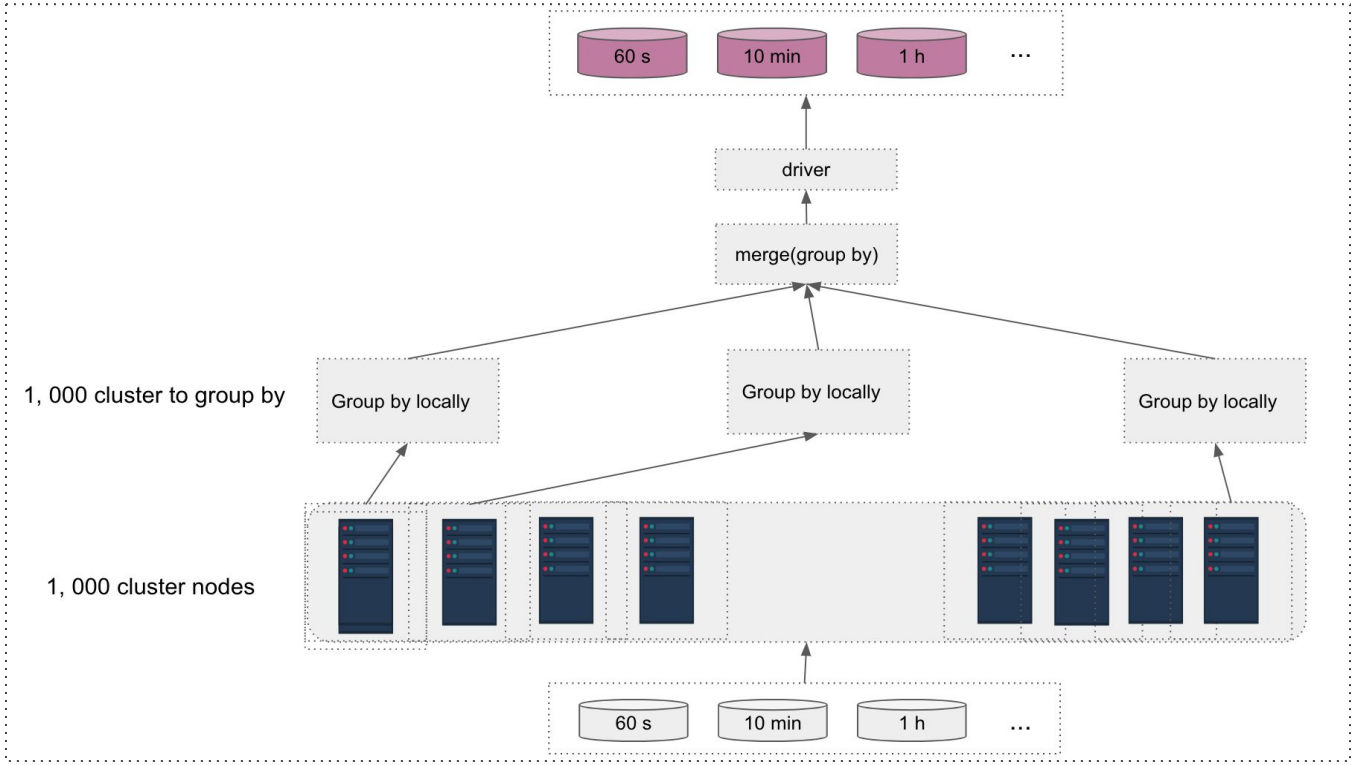
What's the date_format?

- **Day**: %Y%m%d, **hour**: %Y%m%d_%H, **10 minutes**: %Y%m%d_%H%M

What's the optional time?

- **Day**: “”, **hour**: %H, **10 minutes**: %H%M

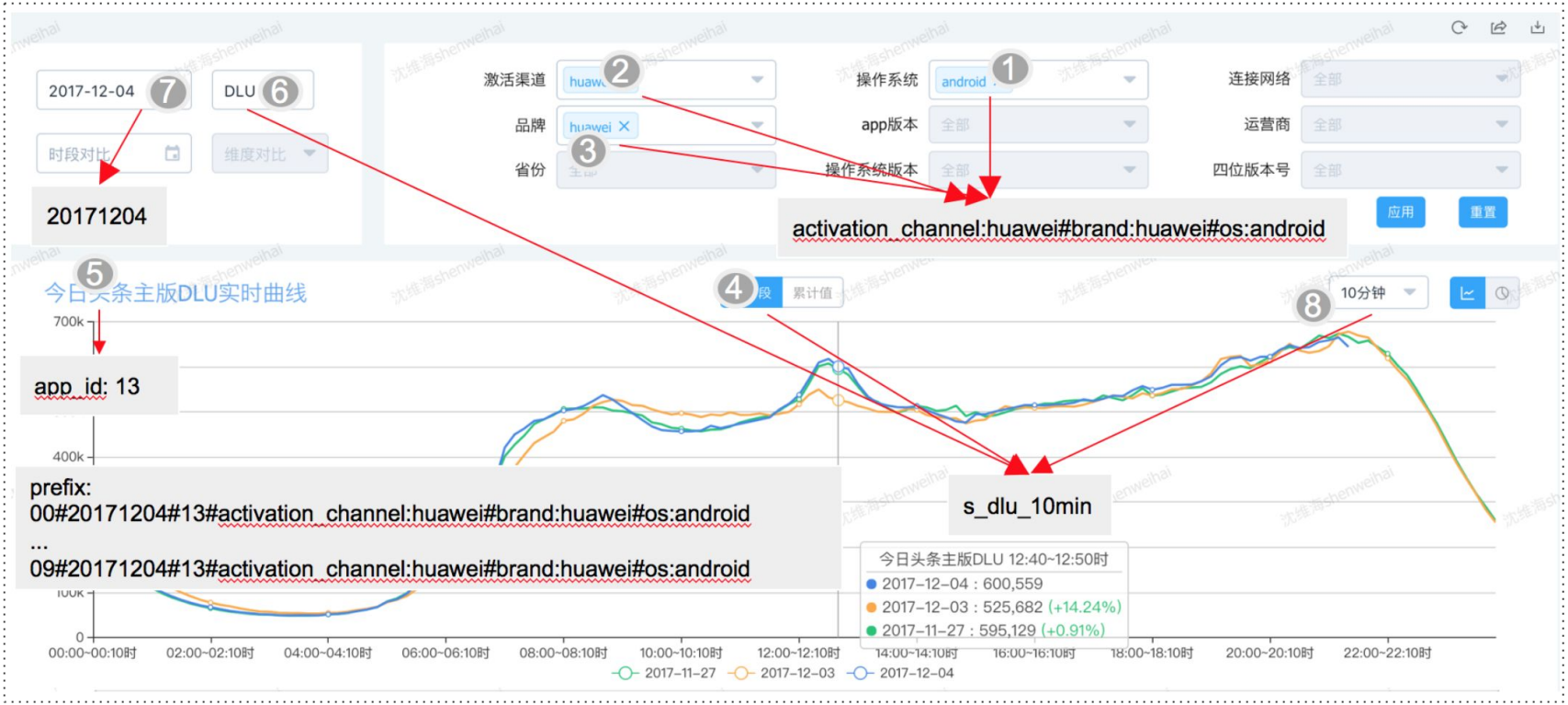
Aggregate by spark



Aggregate by spark - cont.

1. Start 1000 executors and then aggregate data (maybe distinct count) on each executor
2. Merge aggregated data from 1000 executors to aggregate again
3. Spark driver dumps re-aggregated data into result hbase

Sample


2017-12-04
7
DLU
6
时段对比
维度对比
20171204
激活渠道
2
品牌
3
省份
操作系统
1
app版本
操作系统版本
连接网络
运营商
四位版本号
应用
重置
activation_channel:huawei#brand:huawei#os:android
5
今日头条主版DLU实时曲线
4
累计值
8
10分钟
700k
400k
app_id: 13
prefix:
00#20171204#13#activation_channel:huawei#brand:huawei#os:android
...
09#20171204#13#activation_channel:huawei#brand:huawei#os:android
s_dlu_10min
今日头条主版DLU 12:40~12:50时
2017-12-04 : 600,559
2017-12-03 : 525,682 (+14.24%)
2017-11-27 : 595,129 (+0.91%)
0
00:00~00:10时
02:00~02:10时
04:00~04:10时
06:00~06:10时
08:00~08:10时
10:00~10:10时
12:00~12:10时
14:00~14:10时
16:00~16:10时
18:00~18:10时
20:00~20:10时
22:00~22:10时
2017-11-27
2017-12-03
2017-12-04

Sample - cont.

- Using section 4, 6, 8 determines the corresponding Hbase table
- Using section 1, 2, 3, 5, 6, 7 combines the row_key's prefix:

  {salt} #00#20171204#13#activation_channel:huawei#brand:huawei#os:android

Q & A