Druid在今日头条广告的应用

2017.03.04

刘红亮

概 览

- 选择Druid的背景
- 应用实践
- 经验总结

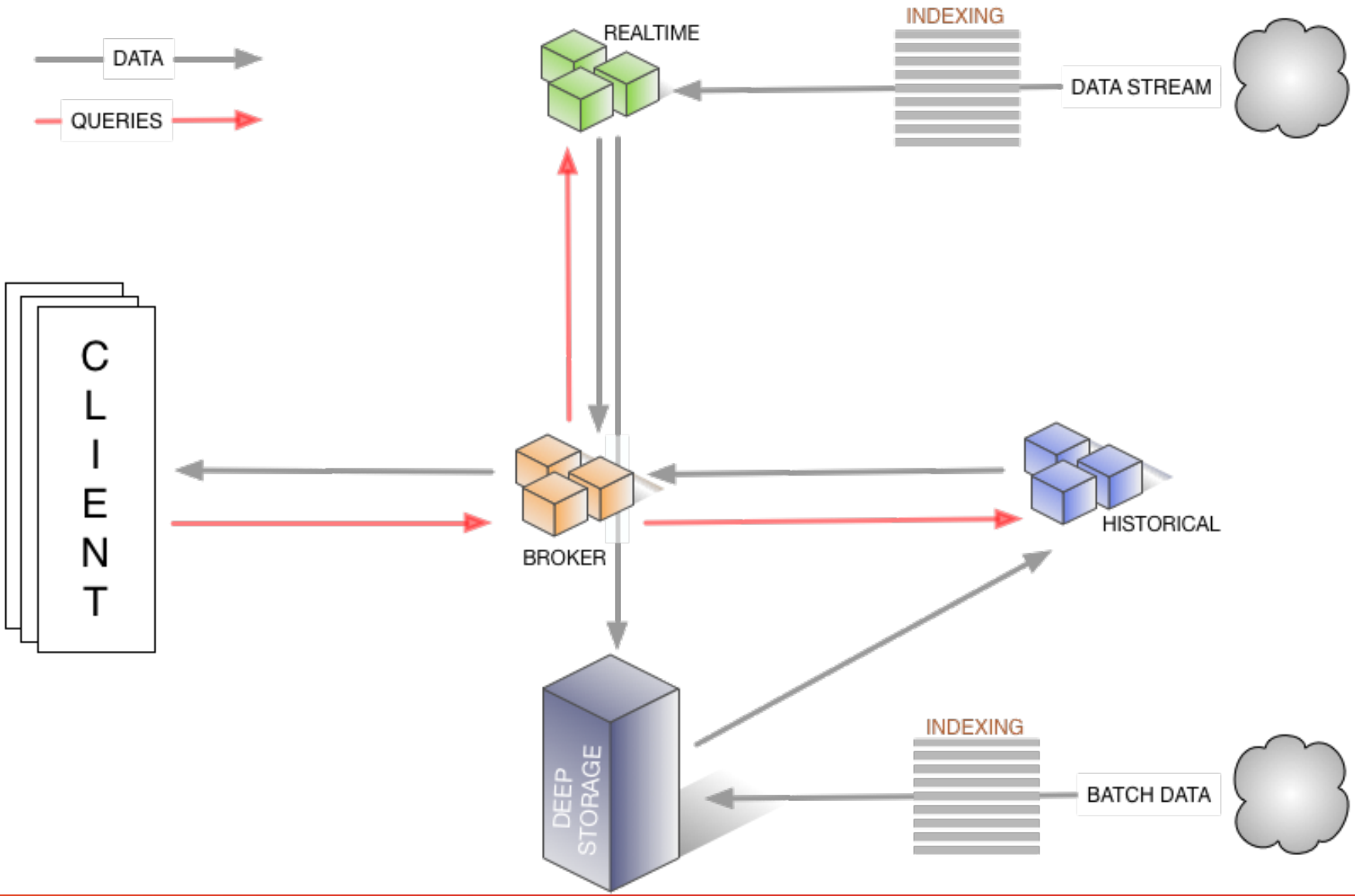
需求背景

- 广告数据特点
  - 广告维度多，维度间相互交叉
  - 广告数据量大，要求实时性强
  - 使用用户多，要求响应时间短
- 需求
  - 一周、一个月甚至更长时间的TB级别广告统计数据快速交互式查询
  - 实时数据的查询
  - 多维度分析图表的查询需求

Druid 简介

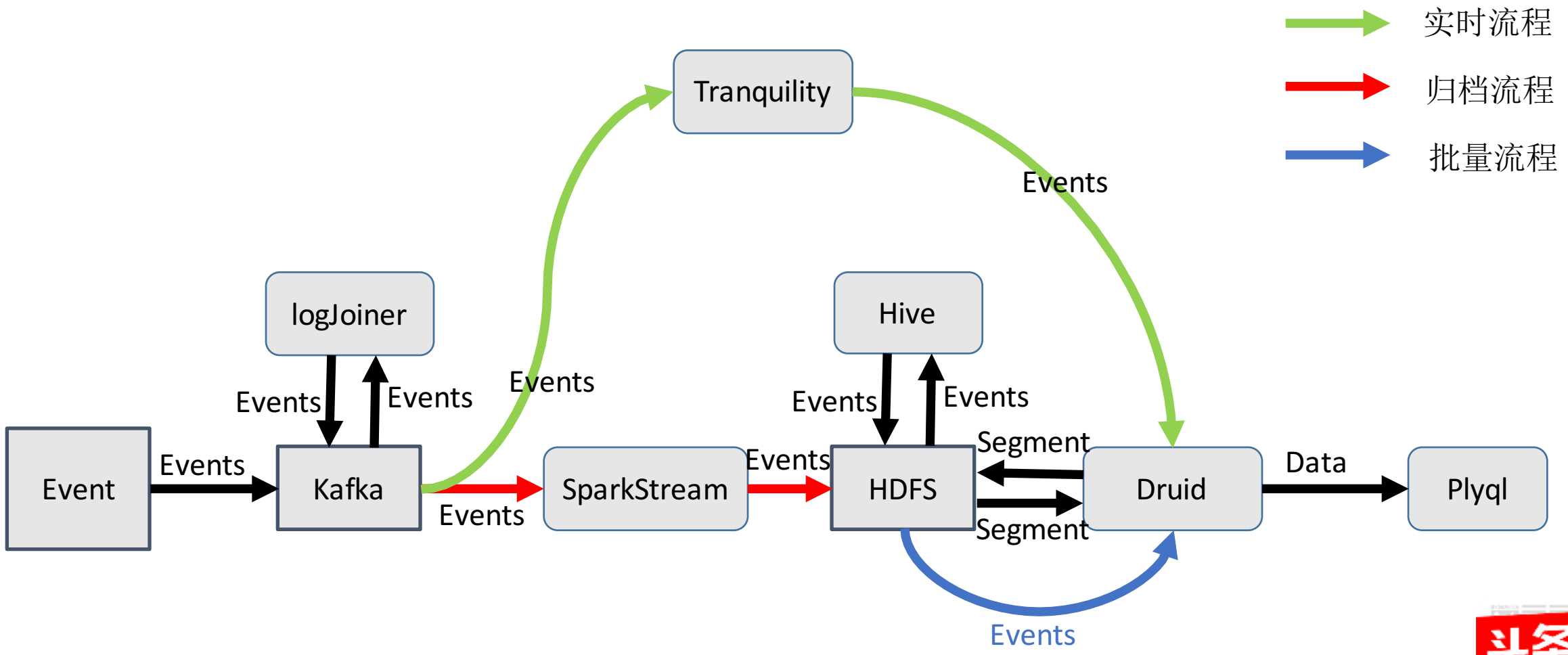
- 高可用性，Segment shard机制
- 高性能，亚秒级查询响应
- 高吞吐，支持实时数据接入，批量数据接入
- 正确性，lambda架构能够在T+1时间校正实时数据
- 查询有segment级别缓存
- 堆外内存复用，避免GC问题
- …

Druid 架构


DATA
QUERIES
REALTIME
INDEXING
DATA STREAM
CLIENT
BROKER
HISTORICAL
DEEP STORAGE
INDEXING
BATCH DATA

- 选择Druid的背景
- 应用实践
- 经验总结

系统架构

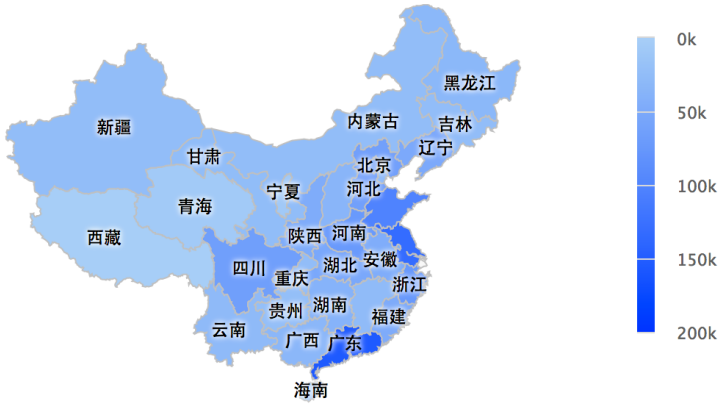

实时流程
归档流程
批量流程
Tranquility
Events
logJoiner
Events
Events
Events
Events
Hive
Events
Events
Event
Events
Kafka
Events
SparkStream
Events
HDFS
Segment
Segment
Druid
Data
Plyql
Events

应用情况

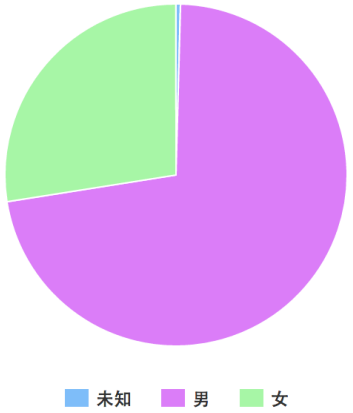
- 按查询最低粒度创建DataSource
  - 目前38个dimension，9个metric
  - 小时粒度的Segment，平均每个Shard 700MB，每天1.3T
- 按BI查询需求抽出中间表
  - 从低粒度表reindex出来，14个dimentsion， 9个metric
  - 天粒度的Segment，每个Shard 40MB，一个副本
- 按实时分析、计算与监控需求创建DataSource
  - 目前13个dimension，9个metric
  - 15分钟粒度的Segment，每个Shard 500MB，一个副本

受众分析

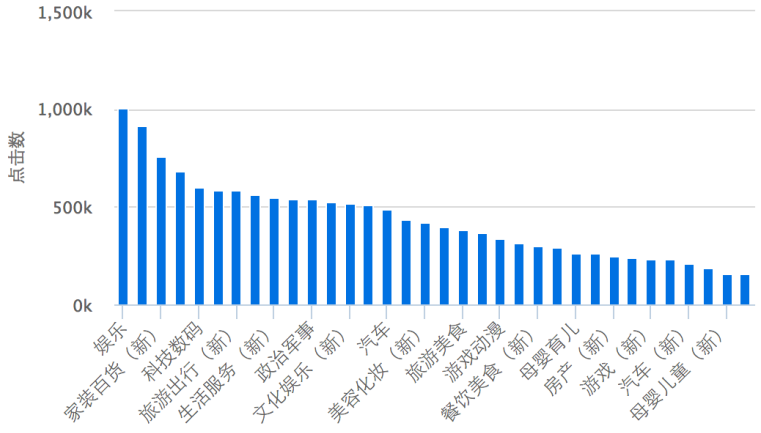
省级地域分布



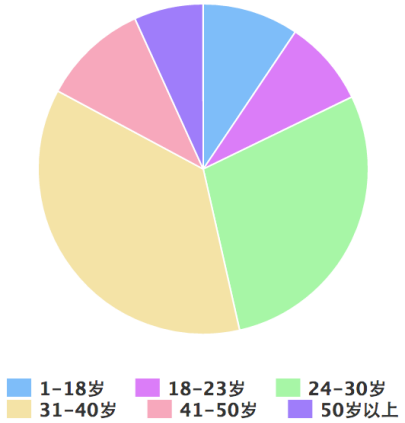
性别分布



兴趣分类分布

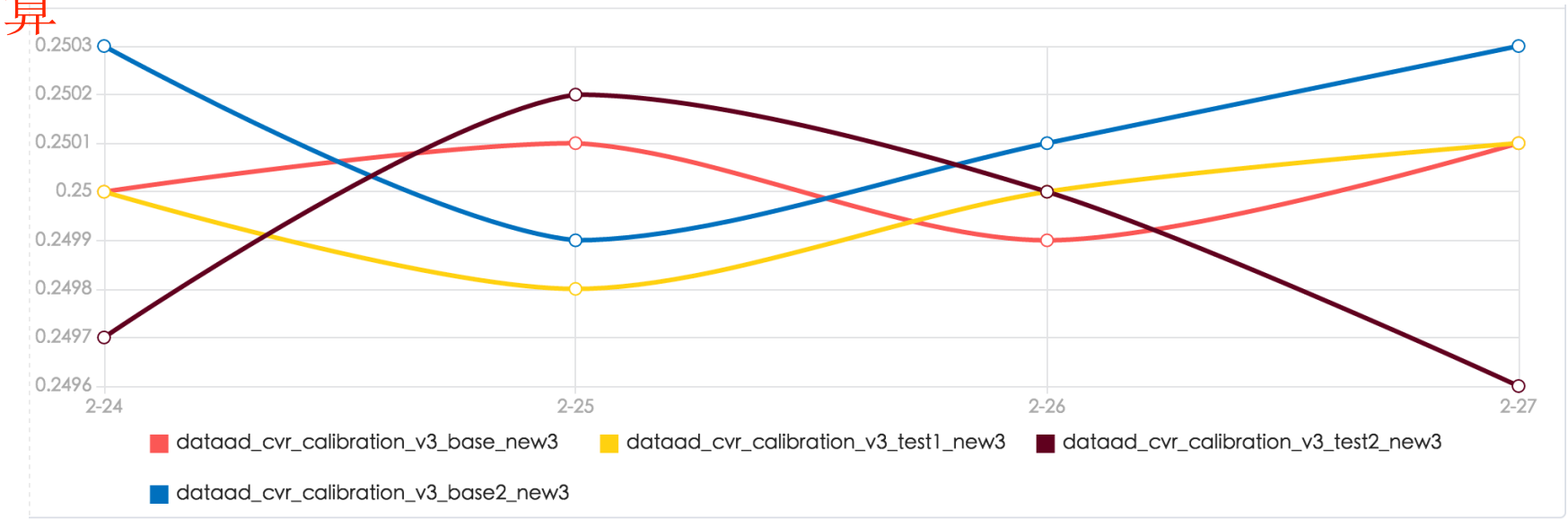


年龄分布



BI dashboard

A/B测试实时计算



试验分组	收入	展示	点击	ctr2	cpm2	ctr3	cpm3
dataad_cvr_calibration_v3_base_new3							
dataad_cvr_calibration_v3_base2_new3							
dataad_cvr_calibration_v3_test1_new3							
dataad_cvr_calibration_v3_test2_new3							

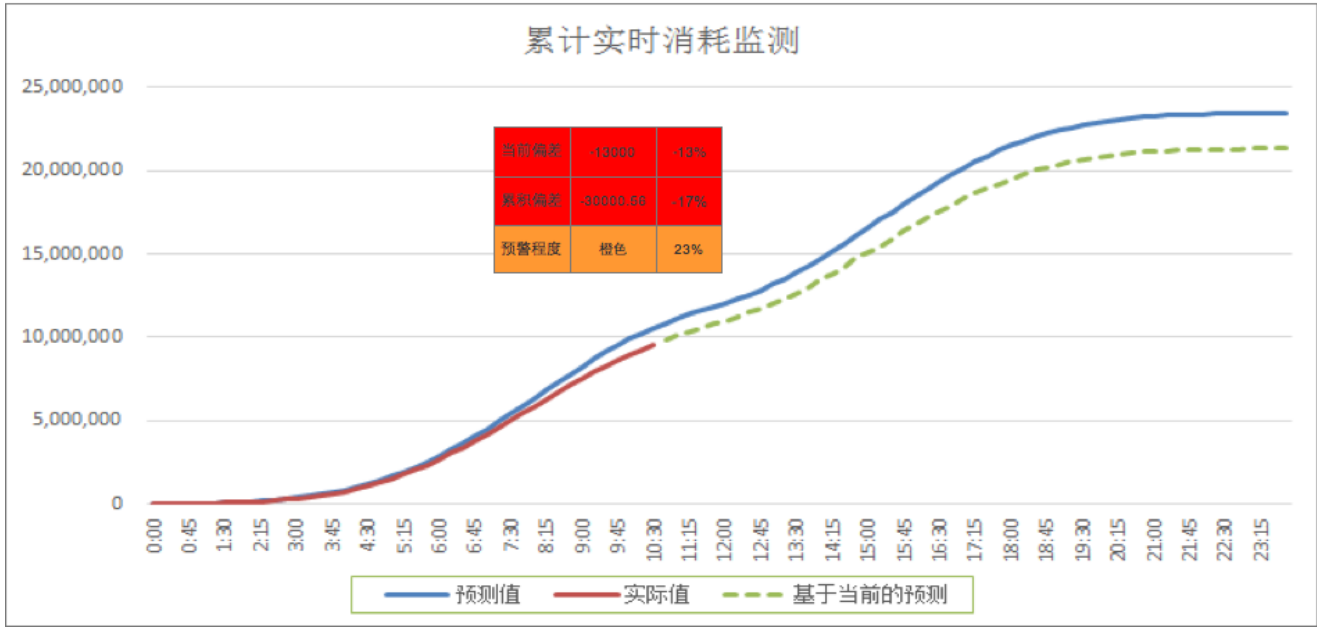
实时监控预警

上头条 | 监控指标： Cost | 时间： 2017-02-07 12：12：06 | 预警程度： 23%

细分维度	视图样式	预测模型
用户特征		
流量来源		
广告属性		

当前偏差最大维度	地域	-23%
差异最大指标	北京	-37%
累积差异最大维度	计费方式	-24%
累积差异最大指标	18-24	+17%

	全天总量	累积完成	完成百分比	前一时间段（15mins）	前一小时
预测	25000000	10000000	43.63%	300000	1000000
实际	22000000	9900000	42.58%	290000	990000

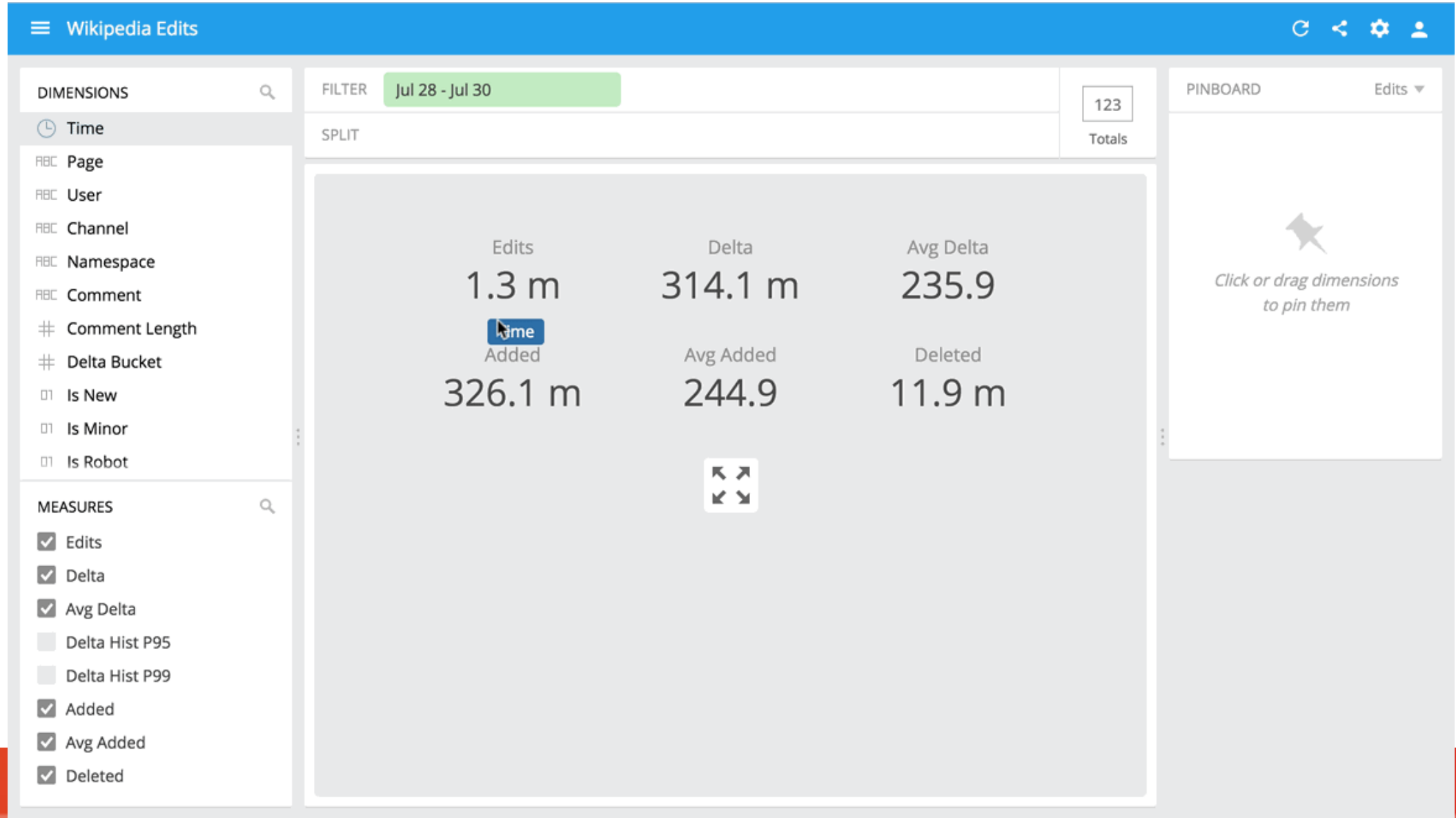


- 选择Druid的背景
- 应用实践
- 经验总结

查询工具

1. Druid：提供HTTP REST形式的查询接口
2. Sql4D：提供SQL的客户端连接以及JDBC
3. Plyql：imply.io自家产品，基于Plywood的SQL client
4. Pydruid：python客户端， Airbnb/Superset使用Pydruid

查询工具


Wikipedia Edits
DIMENSIONS
Time
Page
User
Channel
Namespace
Comment
Comment Length
Delta Bucket
Is New
Is Minor
Is Robot
MEASURES
Edits
Delta
Avg Delta
Delta Hist P95
Delta Hist P99
Added
Avg Added
Deleted
FILTER
Jul 28 - Jul 30
SPLIT
123
Totals
PINBOARD
Edits
Click or drag dimensions to pin them
Edits
1.3 m
Delta
314.1 m
Avg Delta
235.9
Added
326.1 m
Avg Added
244.9
Deleted
11.9 m

部署情况

- 机器：15 nodes, 40 cores, 500GB RAM, 2.4T SSD disk
- Historical nodes
  - 15 * 4 * （ 39 thread, 600GB disk ）
- MiddleManage nodes
  - 15 * 20 worker
- Broker * 3, Coordinator, Overlord * 1
- Tranquility
  - 40 *（ 5 thread, 5 Cores, 8GB ）
  - 20 *（ 10 thread, 10 Cores, 10GB ）

JVM配置

- Broker nodes
  - Heap: 20G-30G
  - MaxDirectMemorySize: bufferSize * (processing.numMergeBuffers[0] + processing.numThreads[39] + 1)
- Historical nodes
  - Heap: 250MB * processing.numThreads
  - MaxDirectMemorySize: bufferSize * (processing.numMergeBuffers[0] + processing.numThreads[39] + 1)
- Overlord, Coordinator nodes
  - Heap: 3G

JVM配置

- MiddleManager nodes
  - Heap: 64M
  - Peon: worker.capacity * index runner JVM
    - Threadpool: druid.processing.buffer.sizeBytes[1,073,741,824] * (druid.processing.numThreads[2] + 1)
    - Ingestion thread & Persisit thread

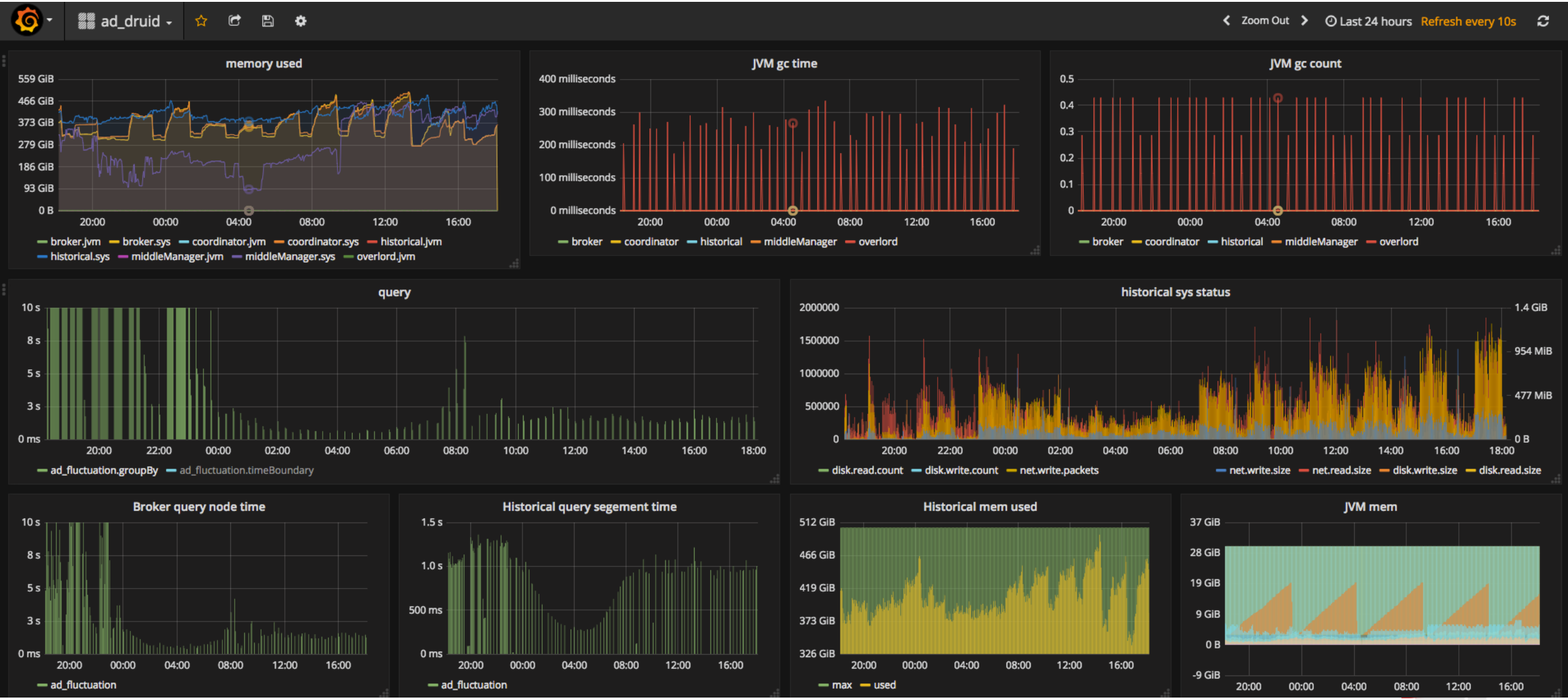
集群性能

	cluster scan rate (rows/s)	core scan rate (rows/s)
count(*)	11,203,801,760	18,673,002
count(*), sum(cost)	6,862,515,059	11,437,525
count(*), group by m1	1,188,869,390	1,981,448
count(*), sum(cost)group by m1	802,015,065	1,336,691

查询数据量： 500亿行，11.3T，21000个shards，没有cache

Metrics监控

- Graphite-emitter
  - HistoricalMetricsMonitor
    - query/time
    - query/segment/time
  - JvmMonitor
    - jvm/gc/count
    - jvm/gc/time
  - SysMonitor
    - sys/disk/write/size
    - sys/net/write/size


ad_druid
Zoom Out
Last 24 hours
Refresh every 10s
memory used
559 GiB
466 GiB
373 GiB
279 GiB
186 GiB
93 GiB
0 B
20:00
00:00
04:00
08:00
12:00
16:00
broker.jvm
broker.sys
coordinator.jvm
coordinator.sys
historical.jvm
historical.sys
middleManager.jvm
middleManager.sys
overlord.jvm
JVM gc time
400 milliseconds
300 milliseconds
200 milliseconds
100 milliseconds
0 milliseconds
broker
coordinator
historical
middleManager
overlord
JVM gc count
0.5
0.4
0.3
0.2
0.1
0
query
10 s
8 s
5 s
3 s
0 ms
20:00
22:00
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
ad_fluctuation.groupBy
ad_fluctuation.timeBoundary
historical sys status
2000000
1500000
1000000
500000
0
1.4 GiB
954 MiB
477 MiB
0 B
disk.read.count
disk.write.count
net.write.packets
net.write.size
net.read.size
disk.write.size
disk.read.size
Broker query node time
ad_fluctuation
Historical query segement time
1.5 s
1.0 s
500 ms
Historical mem used
512 GiB
466 GiB
419 GiB
373 GiB
326 GiB
max
used
JVM mem
37 GiB
28 GiB
19 GiB
9 GiB
0 B
-9 GiB

Q&A

头条

THANK YOU