Tech Star
Qunar

微信扫描关注

# 海量数据，快速聚合

## Druid和caravel在大住宿的应用

去哪儿?
Qunar.Com

Tech Star
Qunar

微信扫描关注

- 姓名：王澍
- 部门：大住宿事业部 数据中心

去哪儿?
Qunar.Com

# Agenda

一．Druid 架构

二．Druid在大住宿的应用

三．与其他OLAP引擎的比较

四．踩过的坑

# Druid

Druid 是一个用于大数据实时查询和分析的高可用、高性能分布式OLAP系统，旨在对PB级数据提供快速查询和分析的服务。

最新版本为0.9.1.1，在Apache license 2.0协议下开源。

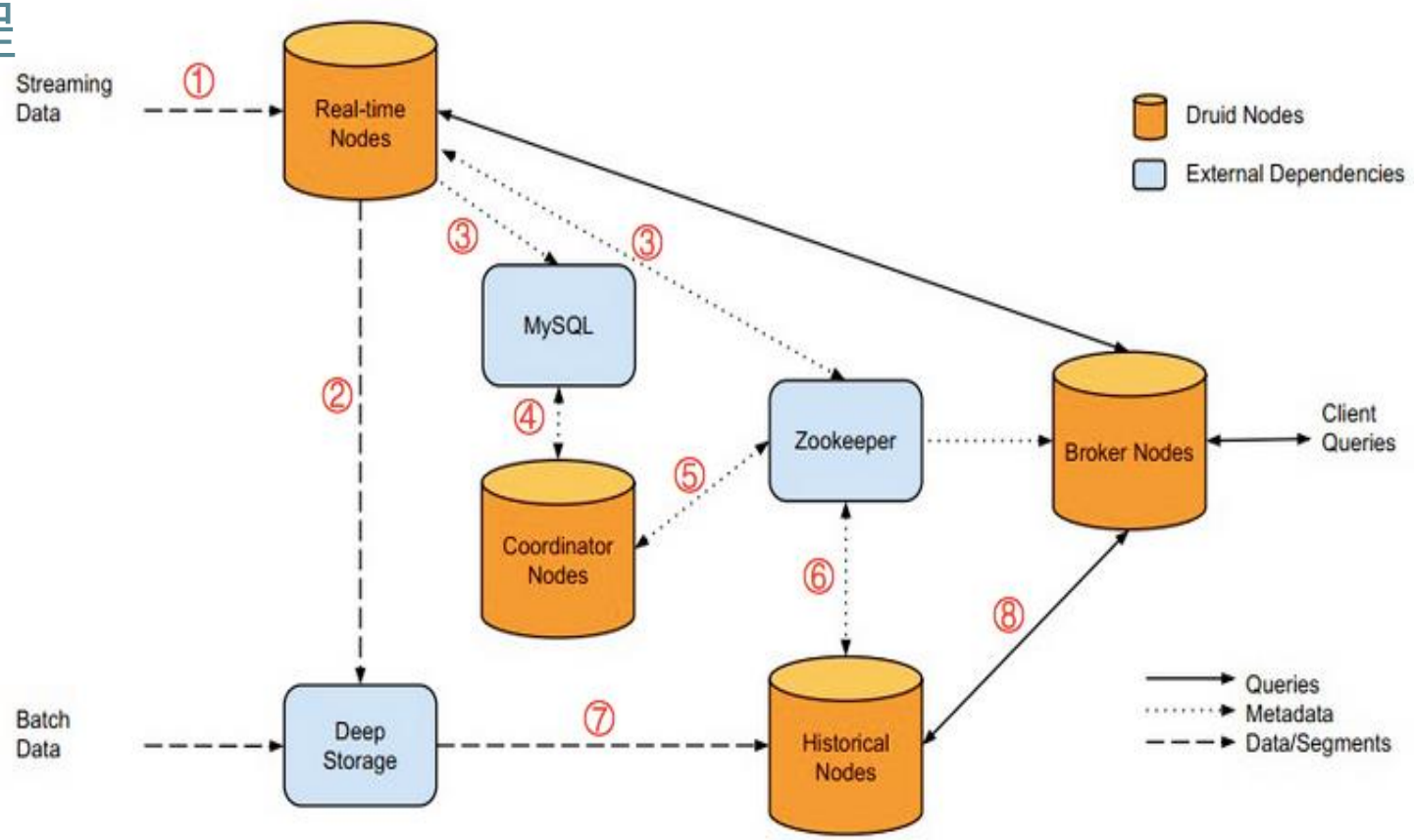
# Druid整体架构

Druid集群中有5中不同角色的进程组成：

- realtime
- coordinator
- historical
- broker
- indexer

同时依赖3个外部服务

- zookeeper
- metadata server
- deep storage



# Druid如何存储数据

- Druid将数据分成三段：时间戳，维度和指标
- 时间用来对数据切片。维度值会建立多种索引，用来快速检索指标值得部分汇总结果
- 不保存原始明细数据，只有部分汇总

| timestamp | dimension | | metric | |
|---|---|---|---|---|
| order_time | hotel_seq | city_name | room_fee | commission |
| 2016-07-1 10:00:00 | beijing_city_27 | beijing_city | 200 | 20 |
| 2016-07-1 10:02:00 | shanghai_city_13 | shanghai_city | 250 | 20 |
| 2016-07-1 10:05:00 | beijing_city_27 | beijing_city | 350 | 40 |
| 2016-07-1 10:07:23 | beijing_city_35 | beijing_city | 250 | 25 |

- 数据中city_name一列会存储下面三个索引信息。核心是倒排bitmap索引

1：枚举值字典

```
{
    "beijing_city":  0,
    "shanghai_city": 1
}
```

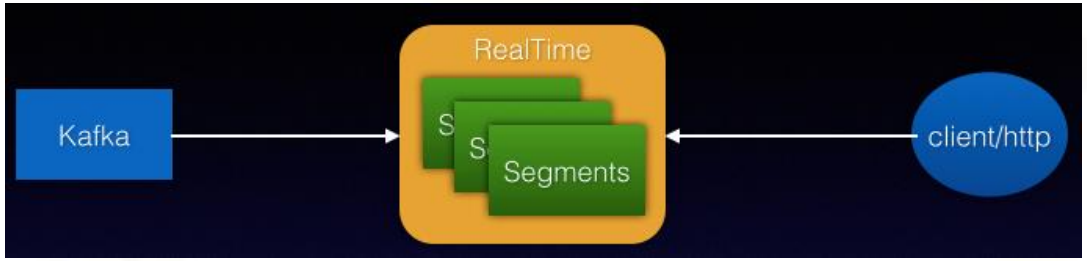
2．列数据

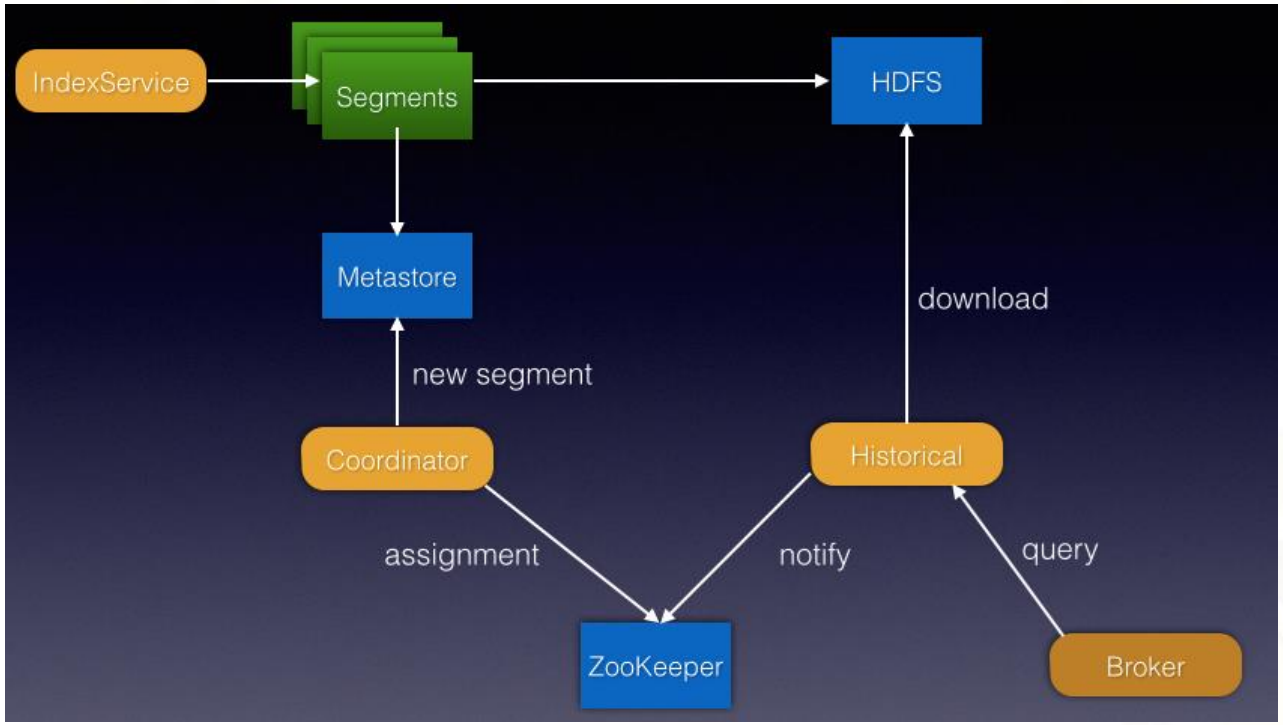
```
[0,   1,   0,   0]
```

3．倒排bitmap索引:

```
"beijing_city":  [1,0,1,1]
"shanghai_city": [0,1,0,0]
```

# Druid架构 – 数据流向



# Druid应用

- **应用场景：**订单数据的多维度实时监控和探索性分析

**难点：**订单会变更，部分维度信息会在订单处理过程中改变。部分指标信息无法实时获取，只能通过离线方式批量导入

**方案：**

分为离线批量导入和实时计算接入两部分。

实时部分接收业务线实时消息，抽取部分重要维度和指标，接入Realtime节点

离线部分定期将离线计算得到的准确结果导入，覆盖掉实时部分的结果

目前druid集群使用4台实体机，离线index采用MR完成，deep storage采用HDFS。

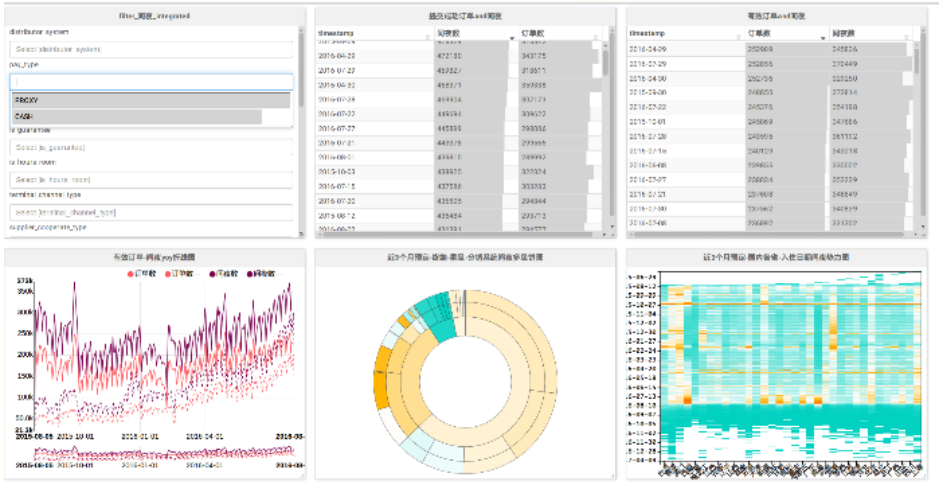
实际使用中，对酒店1.4亿订单，超过50各维度的汇总，索引120G。索引全部订单需要4个小时左右。查询相应基本在10ms左右

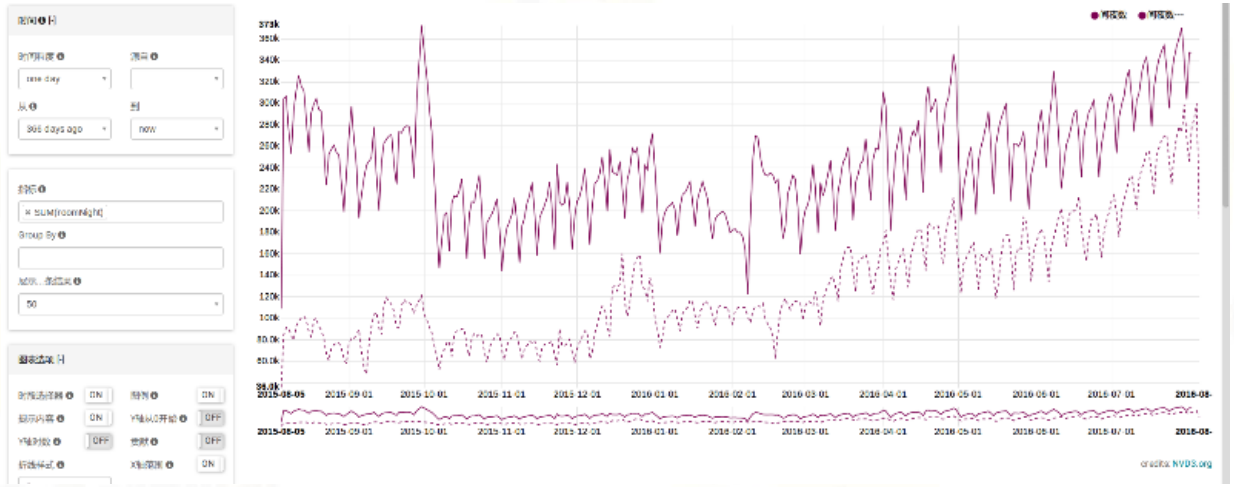
# Caravel

**遇到的问题：**

Druid查询使用的是自己的DSL，类似ES，不好书写。->caravel拖拽选择式配置

明细无法导出。caravel支持Druid和presto，计划改造caravel，汇总走Druid，明细走presto





# Caravel vs. Saiku

| | Caravel | Saiku |
|---|---|---|
| 优点 | Apache License 2.0协议<br>开发非常活跃<br>使用直观，配置灵活<br>二次开发相对简单 | 可以控制行级权限<br>部分汇总在后端完成，不太依赖数据库性能<br>适合详细的分析 |
| 缺点 | 目前bug有点多<br>权限管理过于细碎，不好配置<br>适合画图，不太适合数据表 | 配置复杂<br>仅支持JDBC数据源 |

# 总结

Pros:

- 高可用，水平扩展。靠堆硬件就能解决性能问题
- 高效和压缩和索引算法，能做到PB级数据量下的过滤汇总10ms以下的相应时间
- 实时导入和批量索引结合
- schema灵活。同一个datasource下的不同segment可以有不同的指标与维度

Cons:

- 数据经过初步聚合索引，无法看到明细数据
- 无法更新已导入的数据。更新需要重新索引整个segment
- 维度和指标需要预先定义好，添加新指标需要重新索引，才能对历史数据生效

# 与其他OLAP引擎的对比

- 对比ES：

  ES偏向于文本检索。虽然ES目前支持聚合查询，但是聚合操作占用的资源非常高。

  Druid更适合聚合分析。

- 对比Spark/Hive/Impala/Presto等查询引擎：

  这些查询引擎重点是更灵活的查询方式和完整的SQL支持。

  Druid专注于对热点数据的探索分析。

# Druid vs. Kylin

| Kylin | Druid |
| --- | --- |
| 依赖HBase和Hadoop | 索引和存储可以没有外部依赖 |
| 支持SQL查询 | 查询采用自己的DSL，官方计划支持类SQL查询 |
| 支持星形仓库模型 | 对join支持有限 |
| cube存储空间大 | 压缩效果好，空间占用较小 |
| 不支持实时数据。最新版部分支持 | 支持实时数据 |

# 踩过的坑

- 时区问题。Druid切片依赖时间戳，默认时间戳为UTC，可能会导致切片不正确。需要通过指定JVM默认时区的方法指明时区。
- 官方文档中有对protobuf的支持，实际使用中不支持include，不支持嵌套结构。
- 对非JSON的数据，尽量使用标准CSV格式。否则各种转转义问题会很麻烦。
- 不要把订单号之类的唯一键作为维度，没有意义。
- 适当选择分片大小。使用中发现通过MR索引，实际的索引会在reduce步进行，每个reducer处理一个分片。所以适当增加reduce内存。数据倾斜时可以考虑用多次MR索引。

Tech Star
Qunar

微信扫描关注

# Q&A

去哪儿?
Qunar.Com

Tech Star
Qunar

微信扫描关注

# 感谢聆听

去哪儿?
Qunar.Com