



网易数据基础平台建设

网易杭州研究院
蒋鸿翔

部门介绍

杭州研究院 — 数据科学中心



定位

- 服务网易公司集团内部基础平台建设
- 对外尝试输出成熟的商业化方案，赋能外部企业客户



规模

- 浙江省网易大数据重点企业研究院
- 内部服务于电商、游戏、传媒、教育、金融等各个部门，平台数据 > 100PB
- 对外向包括：传媒、金融、快递、农业等不同行业输出数据解决方案，帮助企业信息化转型



产品

- 数据库产品：RDS、DDB、NTSDB
- 大数据产品：网易猛犸、网易有数、网易哈勃

个人画像

《MySQL内核：InnoDB存储引擎 卷1》作者之一，数据科学中心架构师，网易数据库内核和数据仓库平台负责人，长期从事数据库内核技术和大数据平台底层技术开发，主导网易数据库内核整体技术方案和大数据平台先进技术调研和实现，先后主导了内部MySQL分支InnoSQL、HBase、自研时序数据库、自研实时数据仓库等各种不同的平台，具有丰富的数据库内核和大数据平台相关经验。

MySQL

HBase

Impala

Kudu

Druid

NTSDB

数据库

大数据

数据仓库

分布式

时序数据

Kylin

我们不生产数据，我们只是数据平台的提供商

解决业务/用户在数据治理中的各种问题，让业务/用户能更高效地管理自己的数据，进而产生更大的价值

- 现有功能流程整合，节省用户使用成本
- 新功能/平台不断调研，丰富平台功能
- 新平台功能、性能改造，从而满足用户大规模使用需求
- 根据业务实际需求，输出相应解决方案

<http://note.youdao.com/noteshare?id=20de2248c5d6019a2f85093349ce3bd3>

大纲



01 数据库技术



02 大数据技术

PART 01: 数据库技术



数据库技术

InnoDB

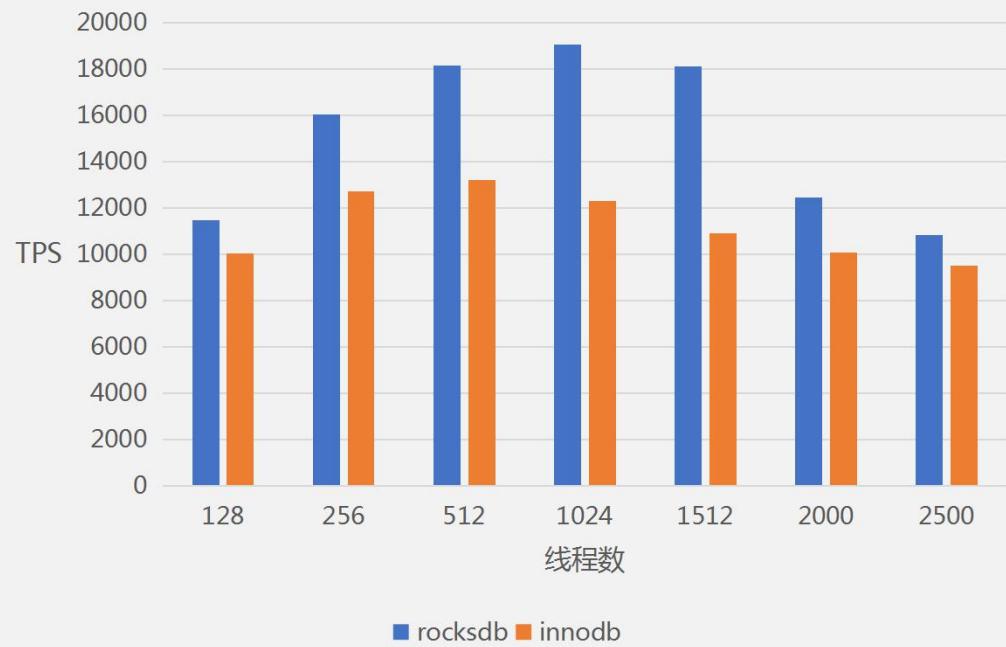
NTSDB

应用特点

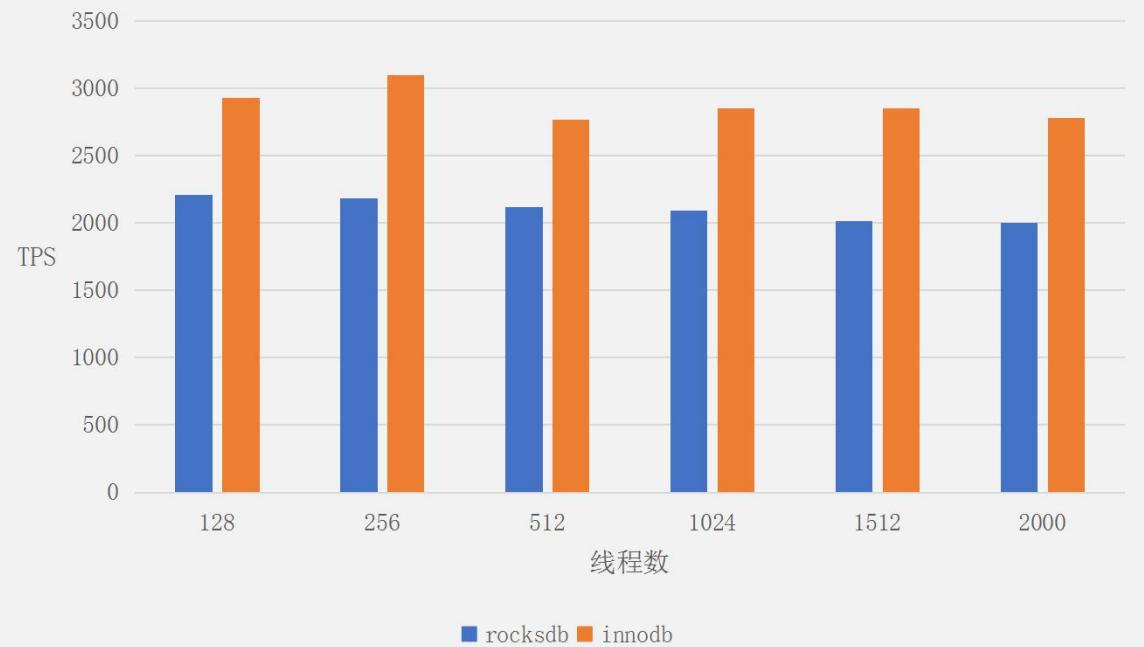
- 大数据量写入，底层LevelDB，基于LSM结构
- 高数据压缩，节省实例存储空间
- 结合DDB可以进行分布式扩展

InnoRocks : 性能对比

innosql throughput



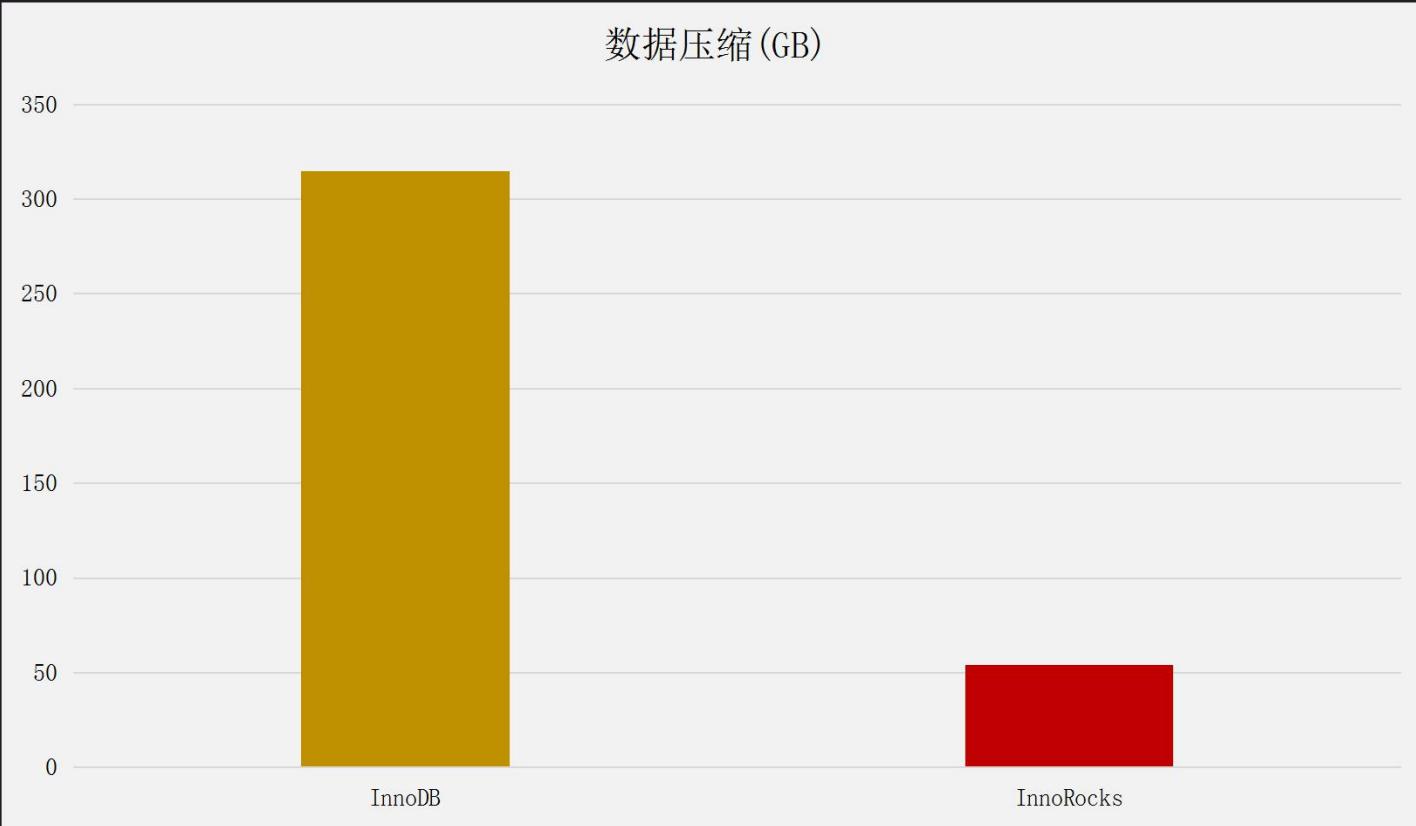
innosql read throughput



写入对比

读取对比

InnoDB : 存储对比



300GB原始数据，分别导入到InnoDB(未压缩)和InnoRocks后的存储容量对比，InnoDB为315GB左右，InnoRocks为50 ~ 60GB，存储容量是InnoDB的1/6 ~ 1/5

InnoRocks : 场景

InnoRocks有较高的数据压缩比，数据写入能维持在一个比较低的延时，不会随着外部压力的增加出现频繁波动，比较适合用于以下场景：

- 大量数据写入场景，比如日志、订单等
- 需要高压缩以便存储更多的数据，InnoDB --> InnoRocks
- 对写入延迟波动比较敏感，HBase --> InnoRocks
- 相对较低的延迟要求(10 ~ 50ms)下替换缓存场景(延迟<5ms)，节省内存成本，Redis --> InnoRocks

NTSDB : 架构



NTSDB : 特点



NTSDB : 场景

内部平台监控系统



NTSDB : 场景

工业互联网方案



技术交流



数据库知乎专栏



HBase & 时序博客

PART 02: 大数据技术



大数据平台

大数据应用开发层

大数据开发套件(可视化IDE)

数据加工

数据集成

数据开发

任务运维

自助分析

数据管理

数据计算

离线计算
Hive

流式计算
Sloth

内存计算
Spark

资源管理

统一资源管理与调度
Yarn

数据存储

分布式文件系统
HDFS和Kudu

分布式数据库
HBase

数据集成

全量/非实时接入
Sqoop

实时/增量接入
NDC和DataStream

数据源

结构化数据
如RDBMS备库

半结构化数据
如JSON

非结构化数据
如音频文件

作业流开发
Azkaban

权限管理
Ranger

多租户管理

元数据管理

数据质量校验
DQC

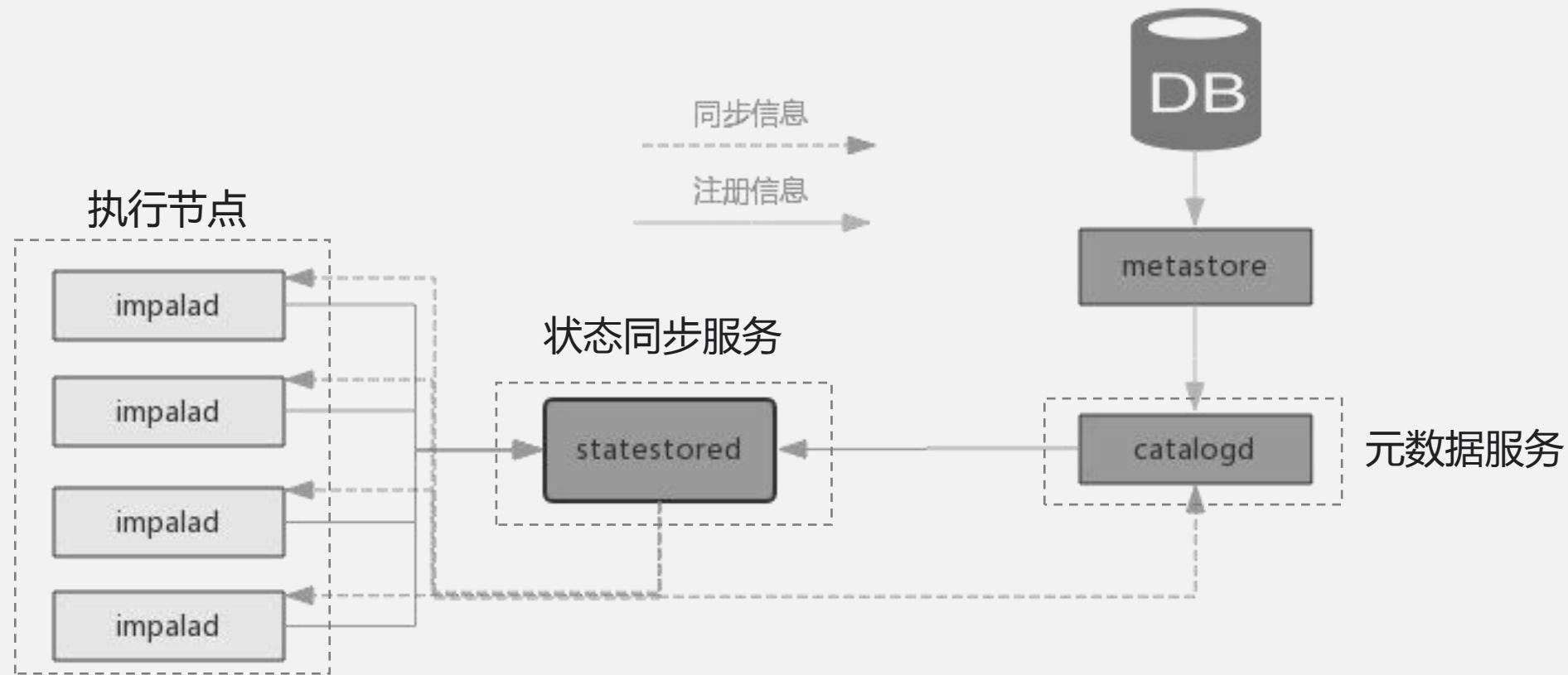
秘钥管理
Kerberos

运维监控
Ambari

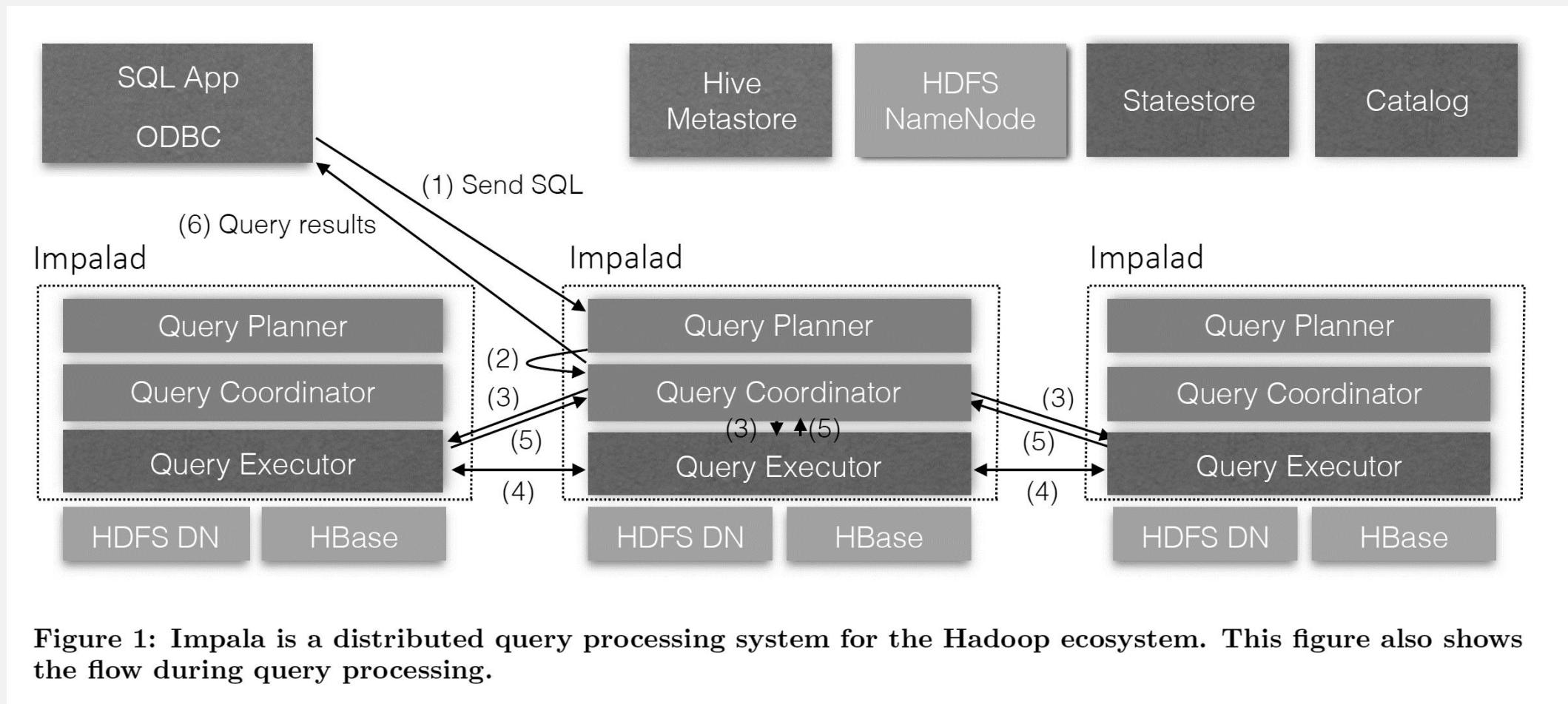
Impala

Impala解决内部大数据量下的ad-hoc查询问题

Impala : 架构



Impala : 执行方式



Impala : 高性能

元数据缓存

MPP并行计算

Codegen
技术，支持
LLVM , JIT

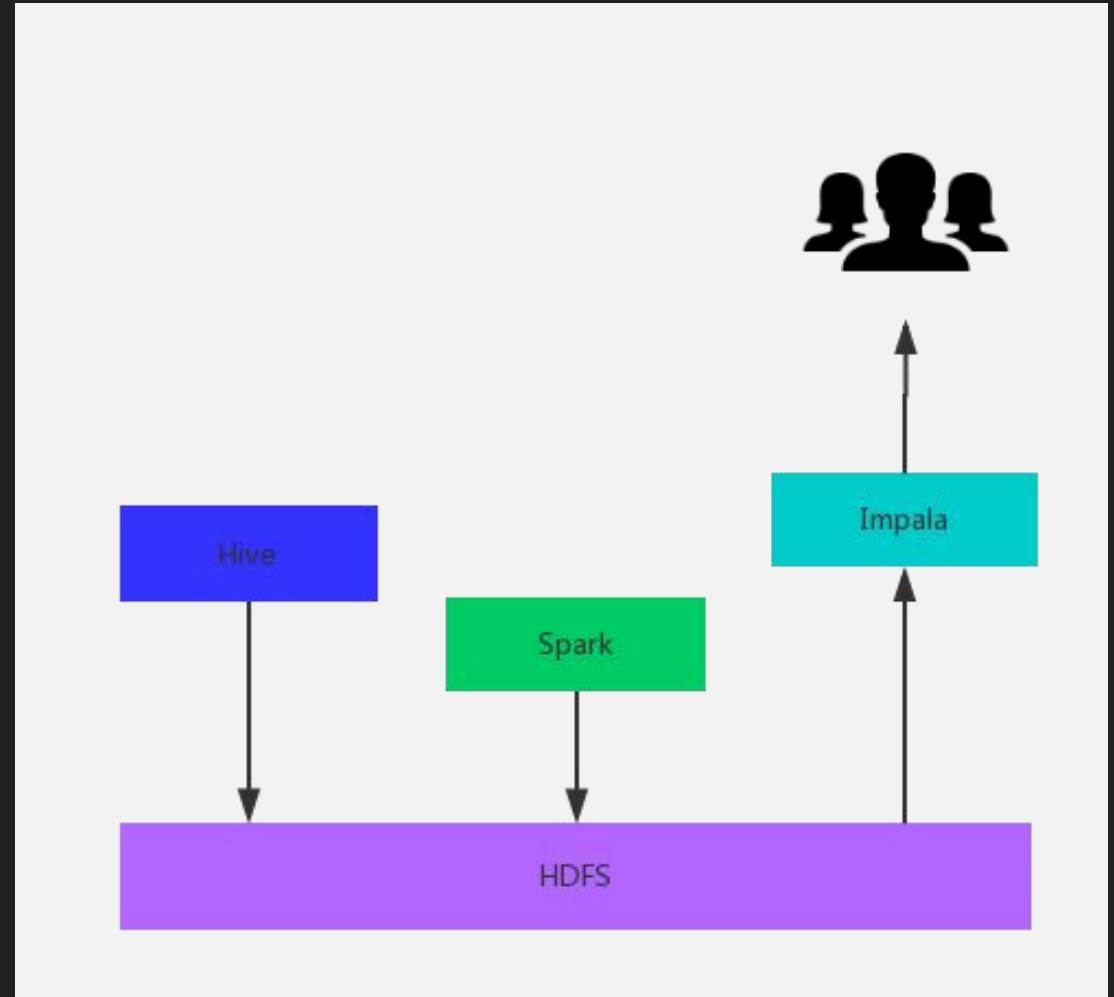
支持HDFS
本地读
(shortcircuit)

算子下推
(>,<,In List,
runtime filter)

Impala : 一般使用方式

适用场景

- 适用于做ad-hoc查询，尽量少做ETL工作
- 适用于不会经常进行元数据变更的场景
- 底层I/O资源充足更能体现性能



Impala : 缺陷

- MPP结构，没有统一Master入口，负载均衡存在一定问题
- 在Hive或者Spark中进行数据变更，与Hive的元数据同步需要手动操作(refresh , invalidate)
- 底层数据权限粒度控制不够
- 元数据缓存机制，在元数据量较大时，会对整个元数据服务造成影响
- 每个coordinator节点都能接收SQL，没有集中统一的SQL管理

Impala : 改进

- 基于Zookeeper的Load Balance机制
- 管理服务器保存最近几天的SQL和执行过程，便于后续SQL审计，超时SQL自动kill
- 底层权限代理功能，解决业务直接拷贝数据到HDFS上的权限问题
- 增加与Hive的元数据同步功能，Hive记录元数据变更，Impala拉取变更自动同步
- 元数据过滤功能：支持库级别正则过滤和表级别自定义过滤
- 集成ElasticSearch作为存储引擎，使用Impala进行SQL查询ES数据

遗留问题：

- 元数据容量问题，过滤只能解决部分问题，需要解决元数据过大产生的各种问题
- 共享存储下I/O资源问题，Impala + Alluxio来缓解I/O，提升查询性能

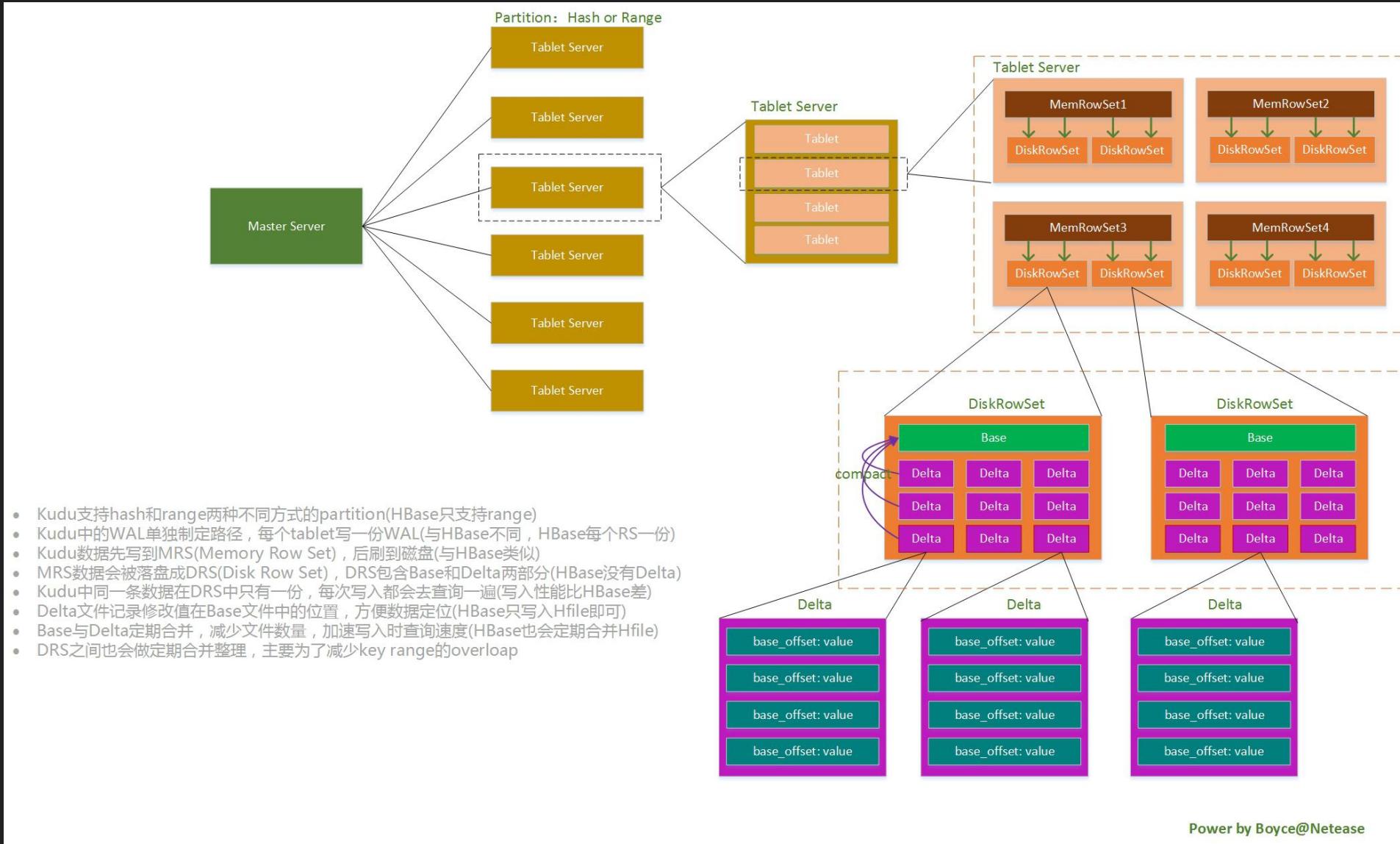
Kudu

Kudu用于解决离线数据实时性问题

Kudu : 架构



Kudu : 架构



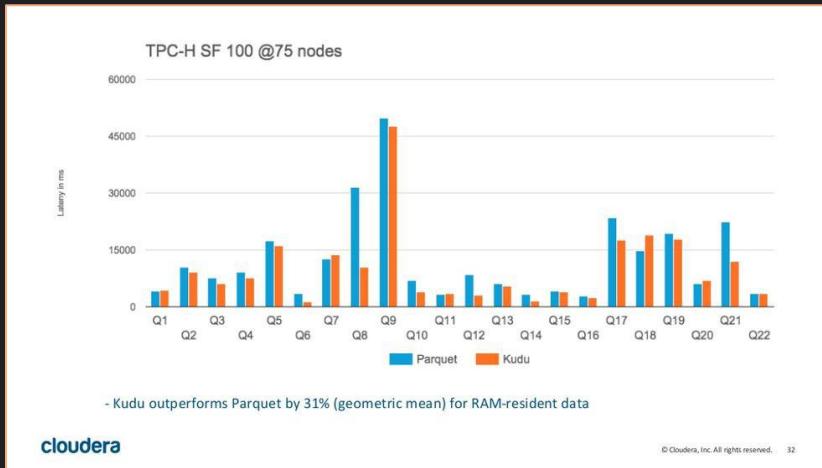
Kudu : vs HBase

	HBase	Kudu
集群架构	Master-Slave结构	Master-Slave结构
选主方式	ZK选主	Raft内部自动选主
数据分布	Range方式分区	Range、HASH分区，支持组合分区
数据写入	HDFS(Pipeline)	Raft多副本
数据格式	ColumnFamily级别列存	RowGroup形式，同一个RG内部列存(类似Parquet)

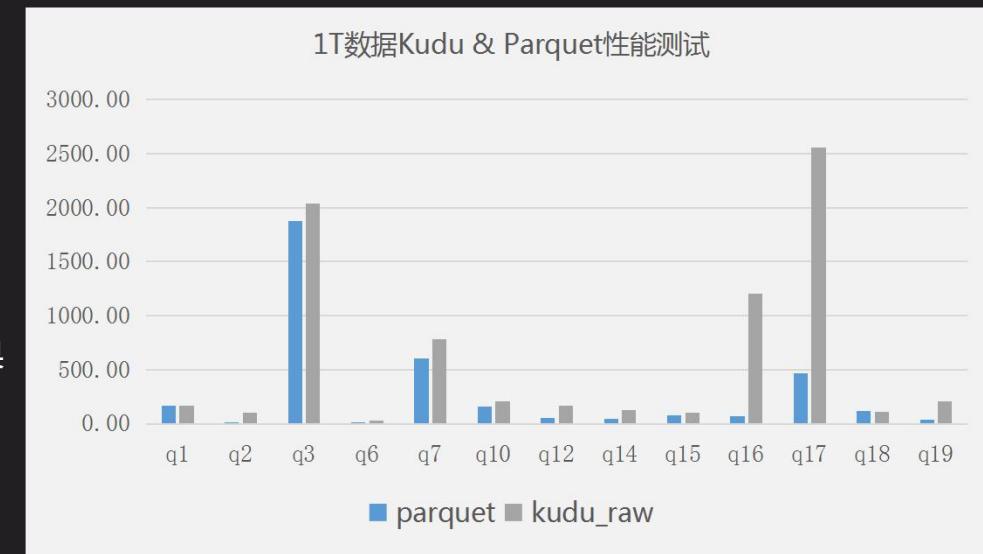
Kudu : 缺陷

Kudu的缺陷

- Kudu由于需要支持update，在内存 & 磁盘上数据的存储采用Base + delta形式，Base记录基本的数据，delta记录修改的数据，所以数据读取时需要同时读取Base + delta两部分数据。
- 通过Impala查询性能与Parquet比有不小差距
- 整个集群做的还不够完善，缺乏像HBase这种Region的Split & Merge功能



官方TPCH测试结果



我们TPCH测试结果

Kudu优化

- 支持Kudu tablet的split
- 支持指定列的TTL功能
- 支持Kudu数据Runtime Filter功能
- 支持Kudu创建Bitmap索引

Kudu : Runtime Filter

User表a(10万记录)

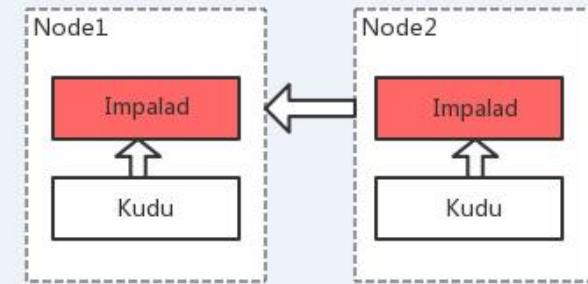
p289443643	1990-11-05	男	中国	广东	深圳
p297993524	1989-10-25	女	中国	江西	南昌
p302543202	1994-10-20	女	中国	广东	广州
p308578250	1990-12-02	女	中国	广东	广州
p347396619	1979-5-08	男	中国	广东	东莞
p358023170	1989-1-28	男	中国	辽宁	大连
p359123611	1993-4-27	女	中国	广东	阳江
p370138980	1996-9-30	男	中国	湖北	武汉
p402117135	1977-7-31	女	中国	北京	北京

Event表b(10亿记录)

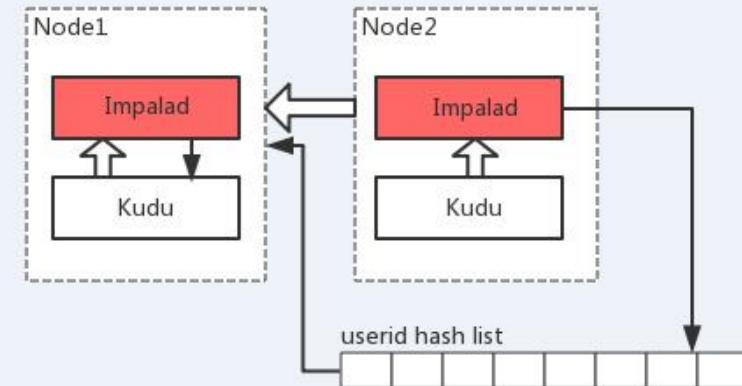
iPhone9,1	未知	750x1334	750	1334	中国移动	true	wifi	中文
iPhone9,2	未知	1242x2208	1242	2208	中国移动	true	wifi	中文
iPhone7,1	未知	1242x2208	1242	2208	中国移动	true	wifi	中文
iPhone8,1	未知	750x1334	750	1334	I WIND	true	wifi	中文
iPhone7,2	未知	750x1334	750	1334	Dialog	false	3G	English
iPad3,1	未知	640x960	640	960	未知	false	unreachable	中文
iPad3,1	未知	640x960	640	960	未知	false	unreachable	中文
iPad3,1	未知	640x960	640	960	未知	false	unreachable	中文

select xxx from user a, event b on a.userid = b.userid where xxx

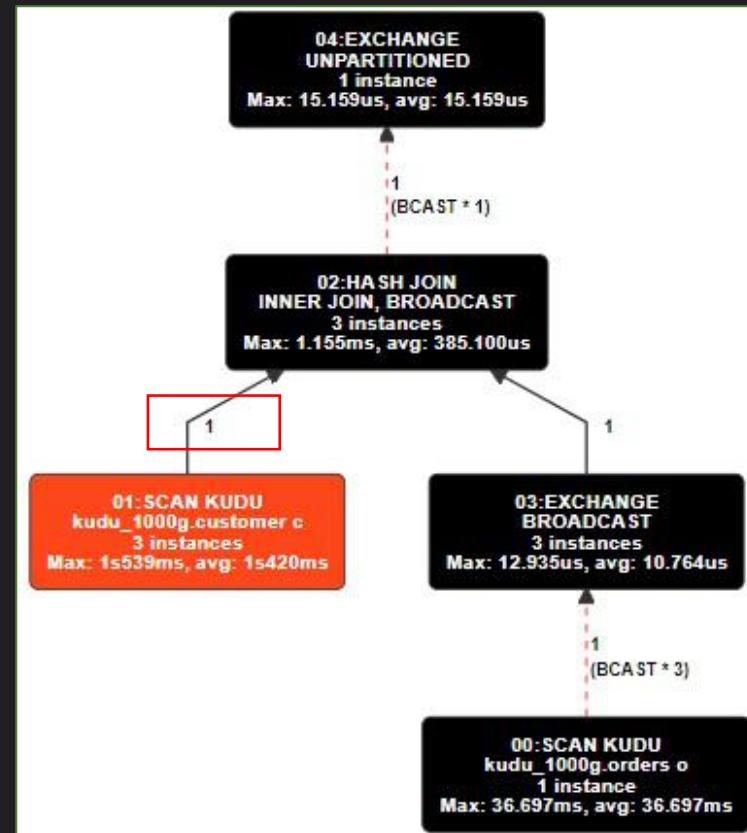
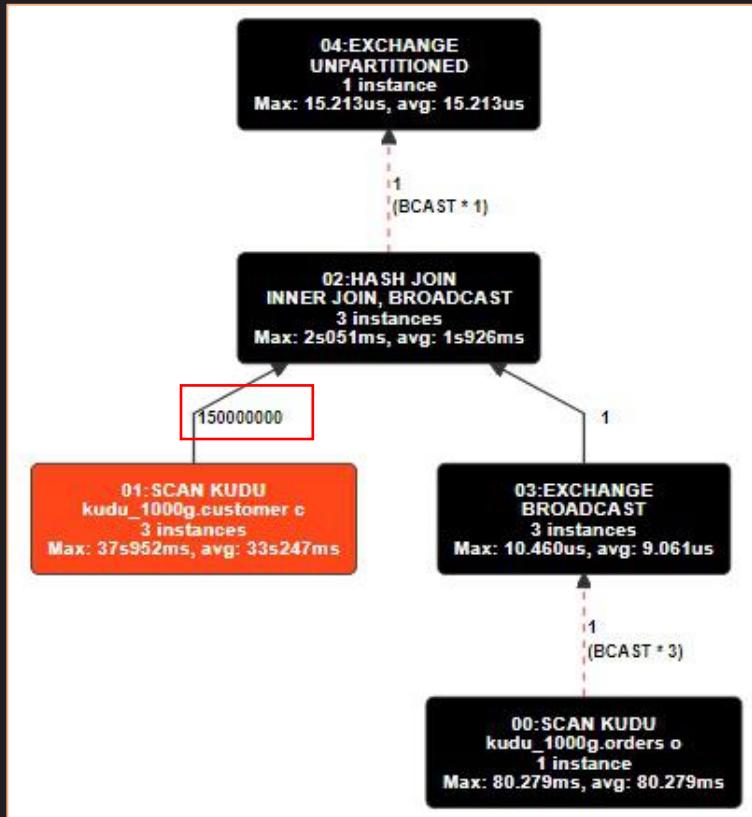
没有runtime filter



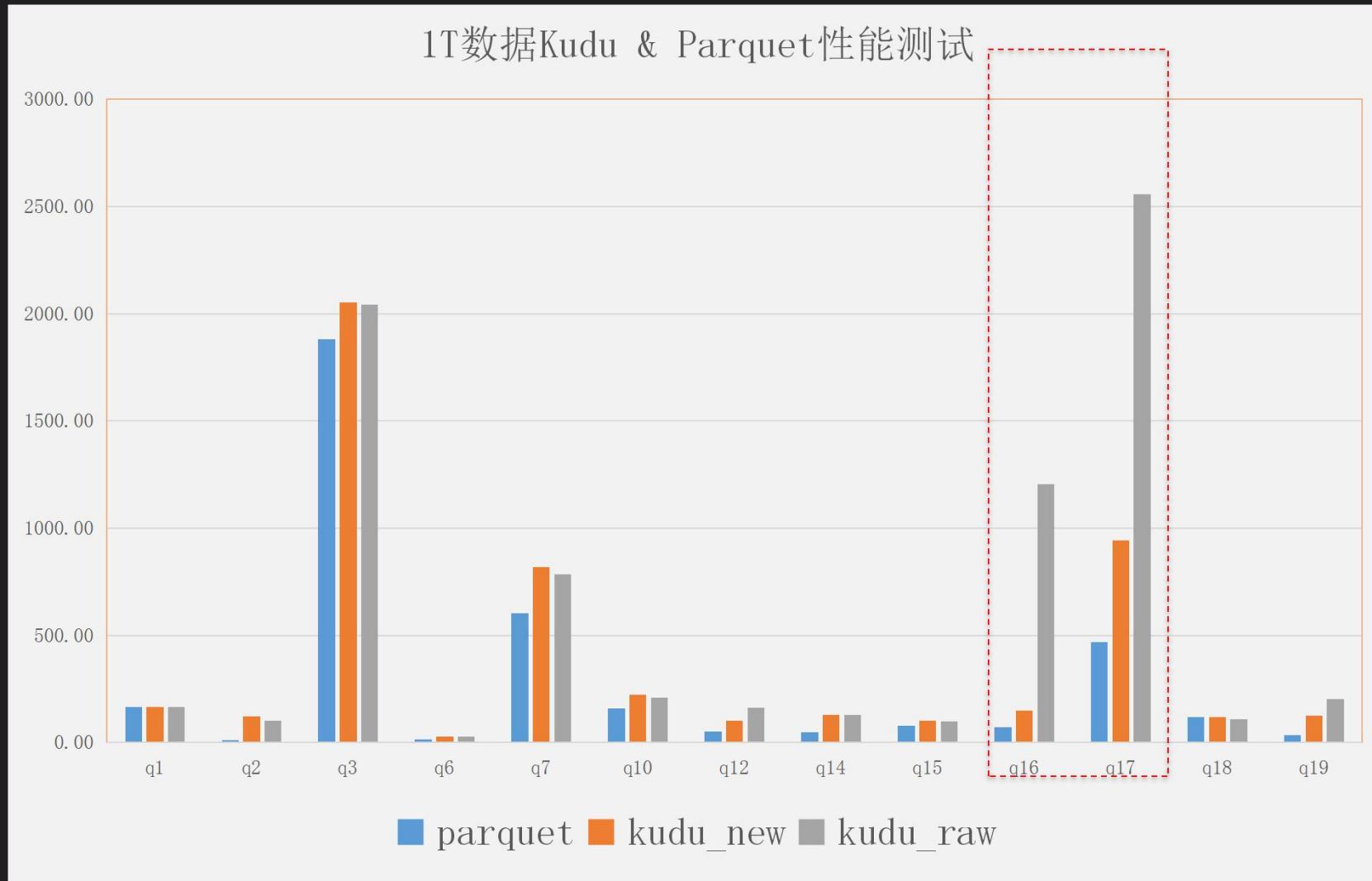
加入runtime filter功能



Kudu : Runtime Filter



Kudu : Runtime Filter



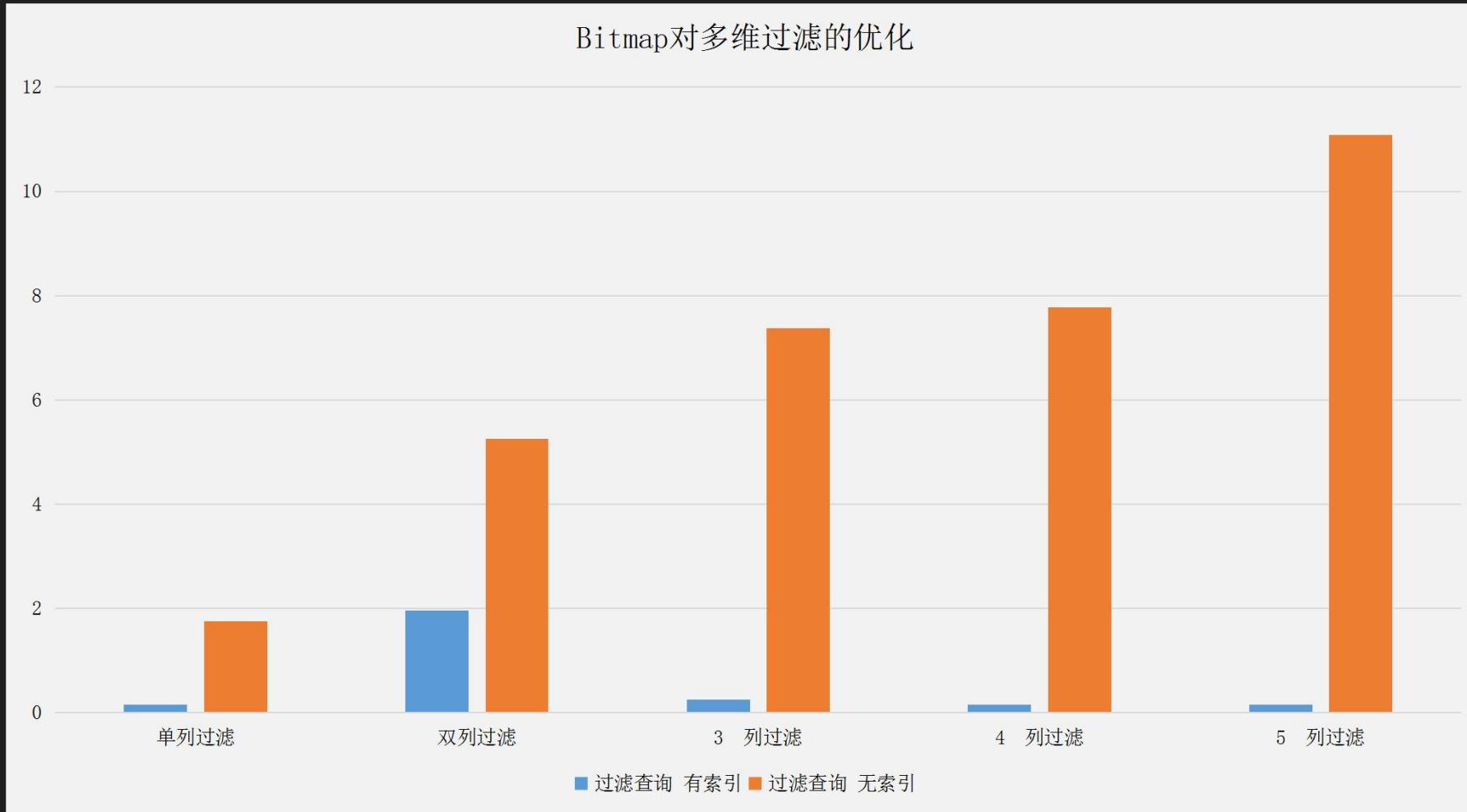
Kudu : Bitmap

适用场景

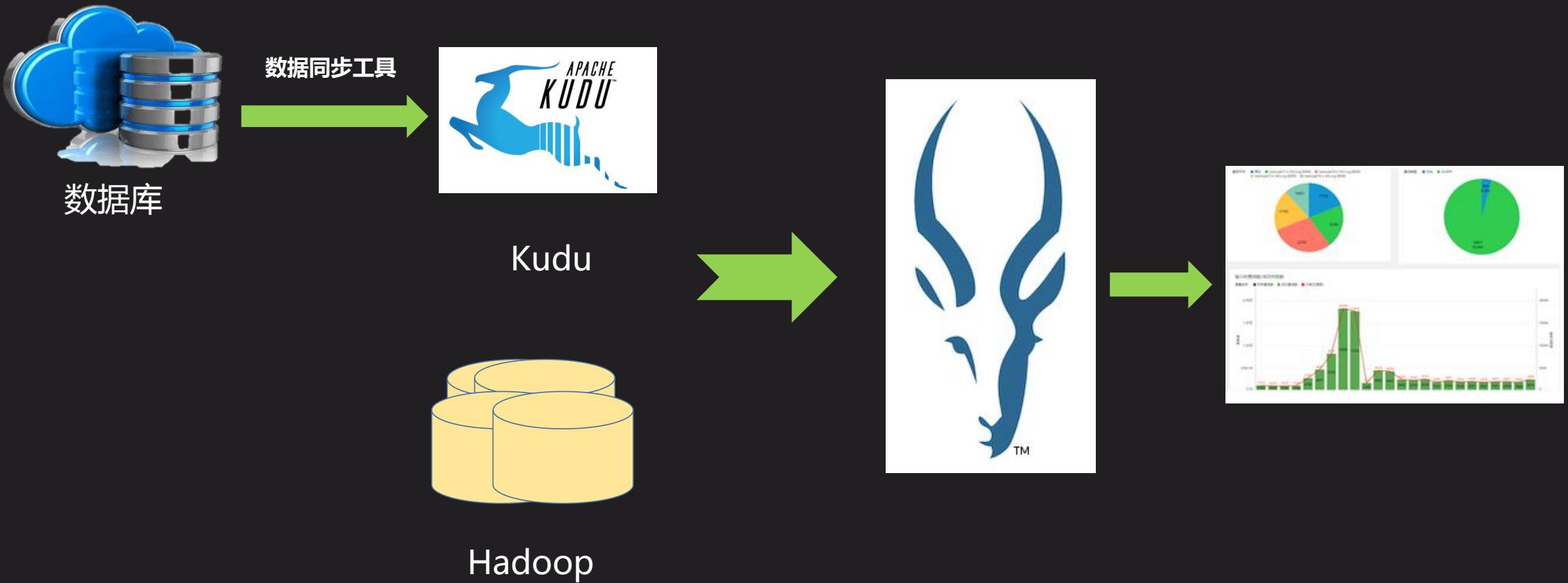
- count distinct值相对数量不多，列的离散性比较差，与unique key相反
- 适用于宽表类型的多维过滤查询、简单过滤聚合查询：select sum(a) from xxx where a=???
- 适用于group by等有聚合的查询：select sum(a) from xxx group by a;
 - group一般实现：sort & hash
 - Bitmap可以简化sort或者hash等大计算量

Kudu : Bitmap测试

TPC-H Factor = 1TB , Part表数据量：2亿，字节数：24GB



Kudu : 场景





网易大数据官网



个人微信号