IPINYOU 品友互动
人群定向 智慧传播

ClickHouse

高性能分布式分析数据库

欧阳辰

品友互动

2017.8.5

内容提要

- 我是谁？
- 什么ClickHouse
- 技术特点和使用场景
- 到底有多快？
- 为什么这么快？
- 品友的小实践

我是谁？

欧阳辰　>17年的软件研发老兵

7年

3年

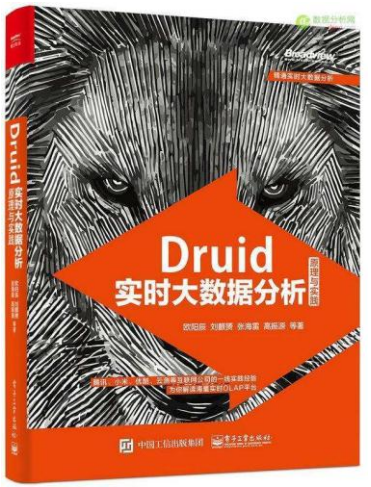
10年

2.5年

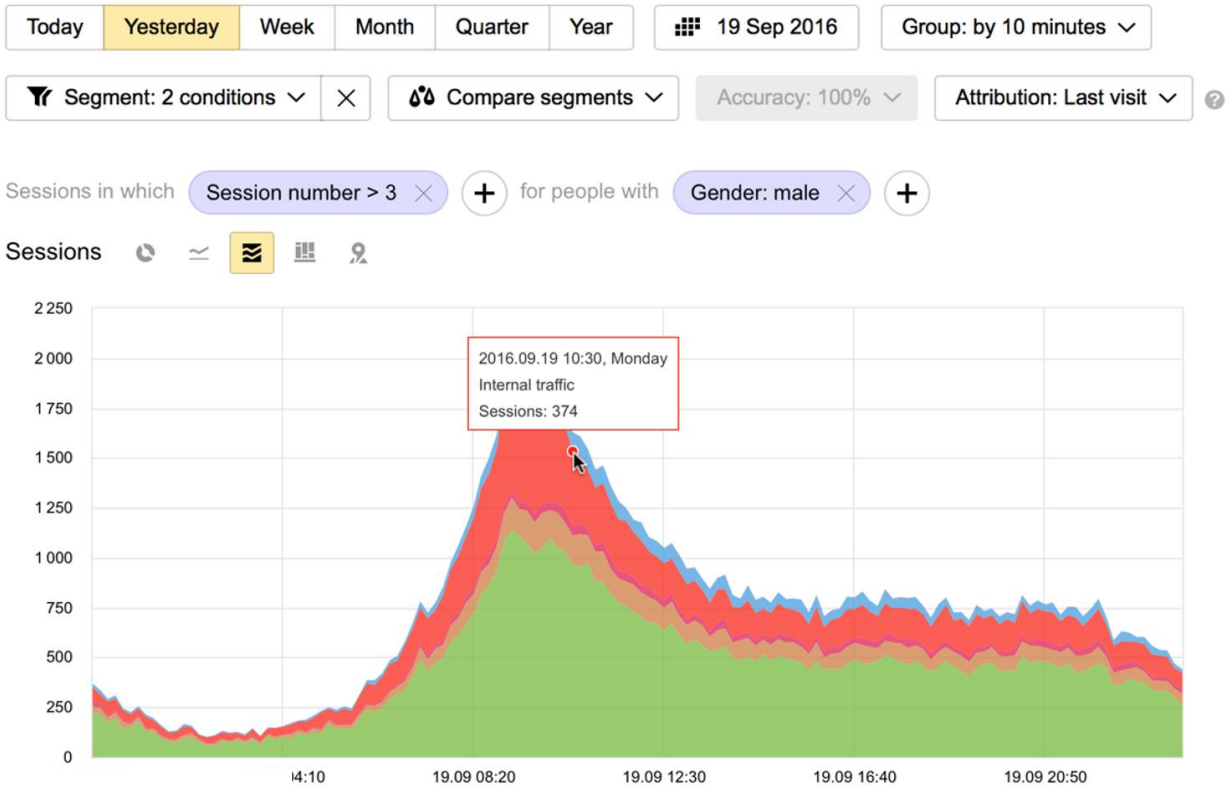
新征程

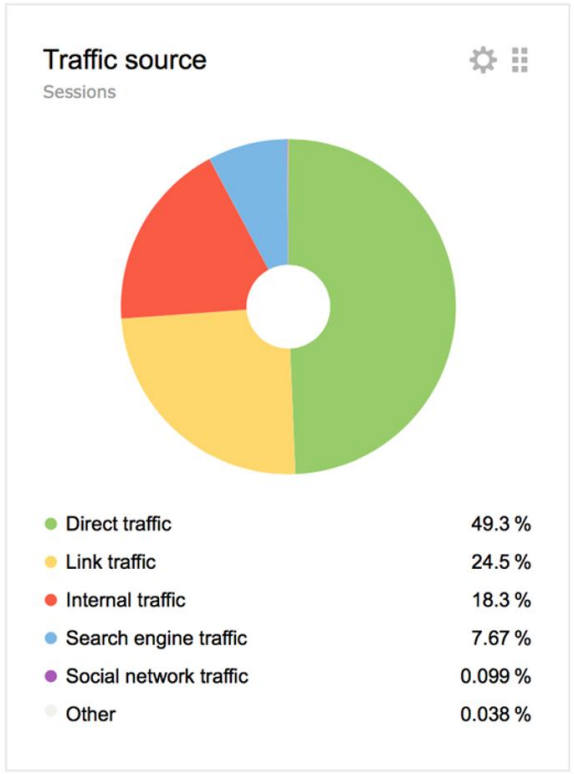
广告平台，大数据

www.ouyangchen.com



Yandex.Metrica *(类百度统计 or Google Analytics)*

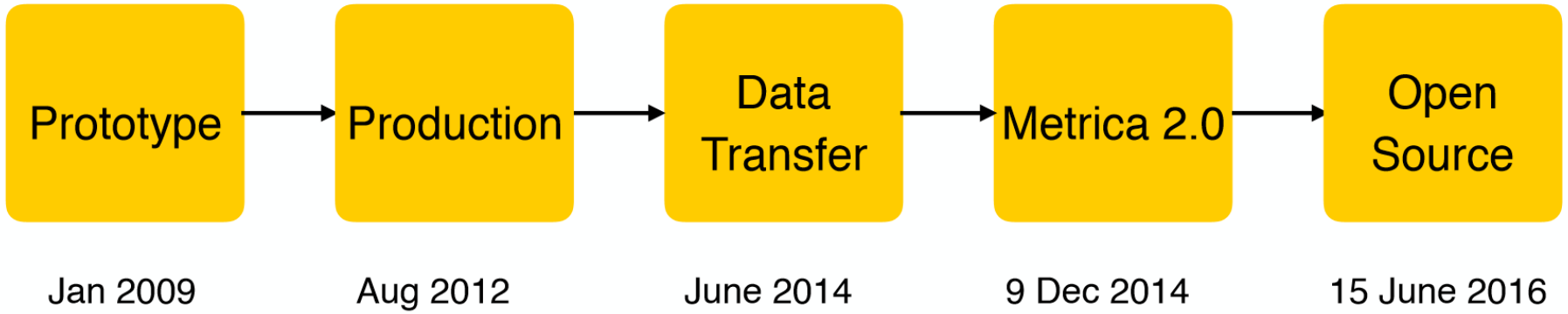




Yandex
Technology,
Technology,
Technology,
Local Focus,
Culture.

数据量：200+亿事件/天，100K+ 分析查询/天，数百万网站

ClickHouse timeline



2017.7

- 超过20+项目in Yandex
- Open-source from June 2016
- Production outside Yandex

实时大数据分析的特点

传统数据库

- 很多表
- 主键连接
- 复杂SQL查询

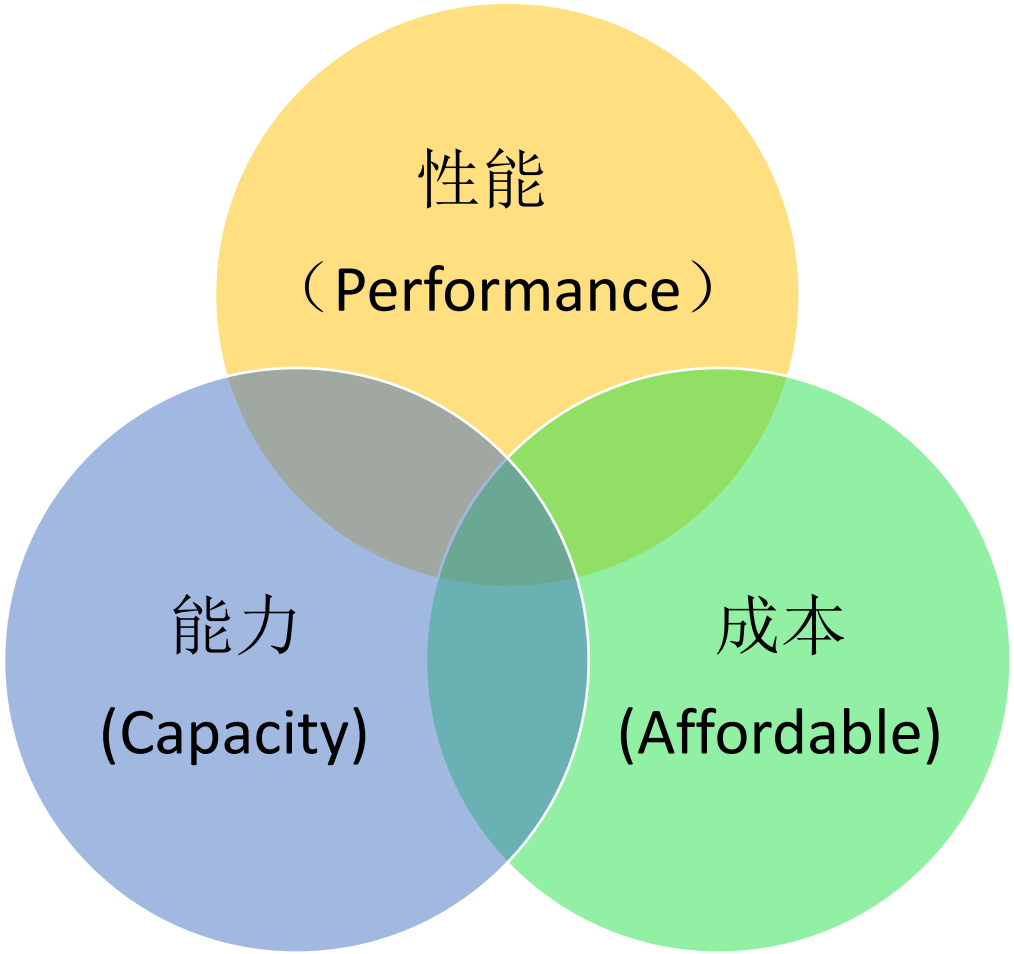
例如，销售数据，管理行为等

实时大数据库分析

- 更少的表
- 更多的列
- 事件 -> 属性

例如，用户行为，服务器日志，IOT，用户画像

大数据分析的CAP



例如：

- Druid: A,P
- Vertica: C,P
- Presto: C
- ClickHouse: P, C, A?

ClickHouse的技术特性和不完美

ClickHouse 关键功能和应用场景

关键功能	应用场景
深度列存储	广告网络和RTB
向量化查询执行(Vectorized Query Execution)	电信
数据压缩	电子商务
并行和分布式查询	信息安全
实时数据注入	监测和遥感
跨数据中心的备份	商业智能
磁盘上的数据访问局部性（Locality of reference)	网络游戏
类SQL 支持	物联网
局部和分布式的Join	
可插入式的纬度表(dimension table)	
预估查询处理	
支持IPV6数据格式	
网站和应用分析	

https://clickhouse.yandex

ClickHouse的不完美：

1.不支持Transaction, OLTP

2.聚合结果必须小于一台机器的内存大小

3.缺少完整的Update/Delete操作

4.不适合典型的Key-Value存储

5.不支持Blob/Document类型数据

6.仅仅支持Ubuntu OS，其他用Docker

Event-oriented RDBMS

谁在用ClickHouse?

适合的场景
• 日志分析，数据量大，PB级 • 复杂的随机查询，例如SQL • 数据不需要进行更改 • 非交易(Transaction)数据

谁在用ClickHouse
• Yandex.Metrica ：Web Event Analystics • CERN(European Organization for Nuclear Research) • LifeStreet • Cloudflare (HTTP Logs Analysis) • 品友互动 : User Behavior Analysis • 评估阶段：腾讯，阿里, OneAPM,等等

IPINYOU 品友互动
人群定向 智慧传播

Yandex.Metrica的设计理念和部署

设计目标

- 快速为王
- 实时数据
- 支持PB级别数据
- 多个数据中心容错
- 查询语言灵活

部署情况

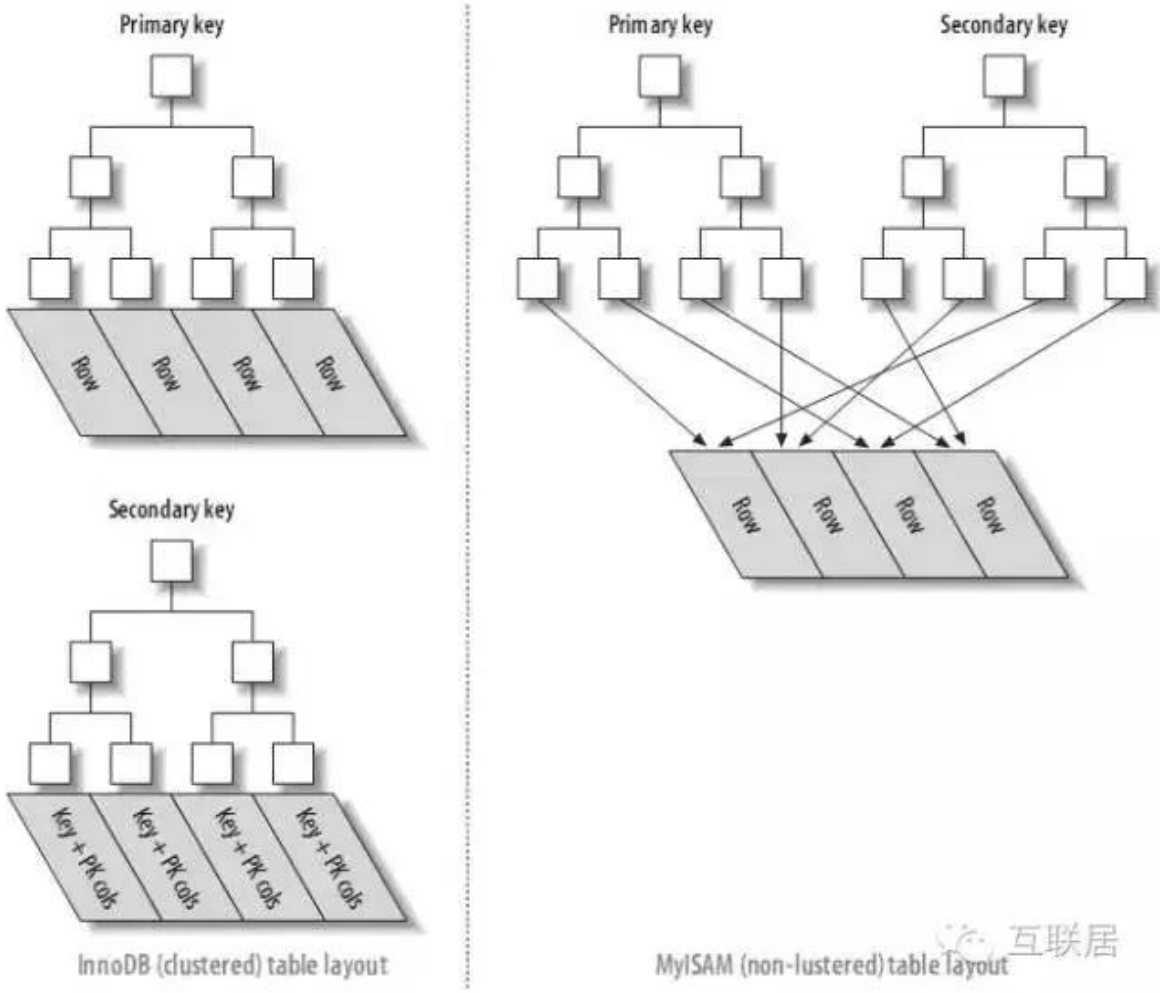
- 3 PB数据
- 412 节点
- 6 数据中心
- <几个小时宕机in 4 years

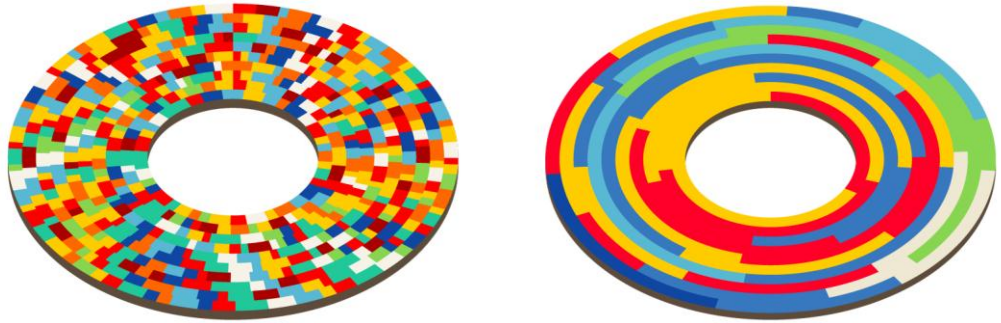
聚焦在快速查询！

ClickHouse/Metrica发展简史

- 第一阶段MYISAM (LSM-Tree) (2008-2011)
- 阶段二: Metrage (从2010-现在/End)
- 阶段三: OLAPServer (2009-2013)
- 第四阶段:ClickHouse（2011-现在）

阶段一： MyISAM V.S InnoDB





数据在磁盘存放的分布

阶段二：Metrage

技术特点

- 数据通过小批量Batch存储
- 支持高强度的写操作
- 读数据量非常小
- 数据被压缩成块(LZ4,QuickLZ)
- 采用稀疏索引

部署情况

- 39*2台服务器
- 3万亿行
- 1千亿事件/天

阶段三：OLAPServer（2009-2013）

技术特点

- 支持维度和指标(Metrics)
- 列值的基数不能太大
- 列式存储，高压缩
- 使用VectorWise方法(SIMD)
- 只支持1-8字节数据类型
- 不支持URL类型

部署情况

- 7280亿行数据
- 大部分查询小于50毫秒

阶段4：ClickHouse(2009-2016(开源)-now)

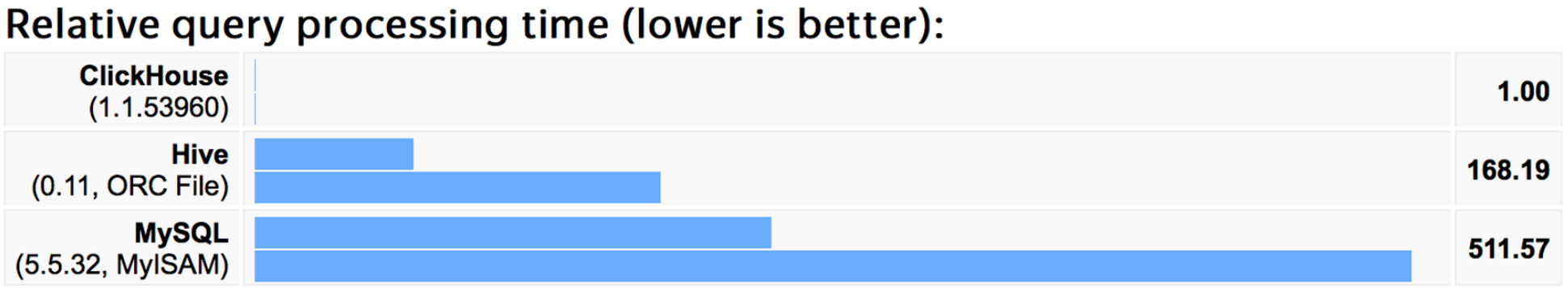
设计背景

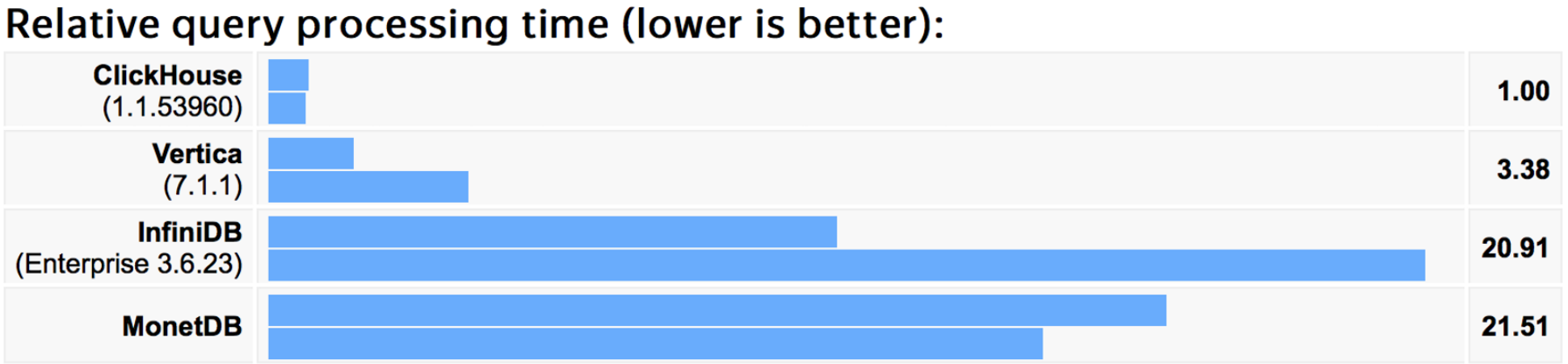
- 基数大的列，聚合意义不大，例如URL等
- 过多维度组合会导致组合爆炸
- 用户常常只关心聚合后的数据中的非常一一小部分数据，因此大量聚合预计算是得不偿失的。
- 聚合后的数据，数据修改会非常困难，很难保证存储的逻辑完整性

部署情况

- 2015年，12月
- 超过11万亿行，数据表有200列
- 机器从60台增长到400台
- 性能远超Vertica

ClickHouse到底有多块？ 官方测试数据





More info: https://clickhouse.yandex/benchmark.html

ClickHouse 到底有多快？例子1

订单表

```
CREATE TABLE lineorderfull (
        LO_ORDERKEY             UInt32,
        LO_LINENUMBER           UInt8,
        LO_CUSTKEY              UInt32,
        LO_PARTKEY              UInt32,
        LO_SUPPKEY              UInt32,
        LO_ORDERDATE            Date,
        LO_ORDERPRIORITY        String,
        LO_SHIPPRIORITY         UInt8,
        LO_QUANTITY             UInt8,
        LO_EXTENDEDPRICE        UInt32,
        LO_ORDTOTALPRICE        UInt32,
        LO_DISCOUNT             UInt8,
        LO_REVENUE              UInt32,
        LO_SUPPLYCOST           UInt32,
        LO_TAX                  UInt8,
        LO_COMMITDATE           Date,
        LO_SHIPMODE             String
)Engine=MergeTree(LO_ORDERDATE,(LO_ORDERKEY,LO_LINENUMBER),8192);
```

客户表

```
CREATE TABLE customerfull (
        C_CUSTKEY       UInt32,
        C_NAME          String,
        C_ADDRESS       String,
        C_CITY          String,
        C_NATION        String,
        C_REGION        String,
        C_PHONE         String,
        C_MKTSEGMENT    String,
        C_FAKEDATE      Date
)Engine=MergeTree(C_FAKEDATE,(C_CUSTKEY),8192);
```

节点硬件：

CPU：24 cores E5-2643 v2 @ 3.50GHz

存储： PCIe Flash storage

数据大小：

- 订单表：150亿行
- 客户表：7500万行

导入数据库后，订单表存储大小为464GB，（压缩率3X)

ClickHouse 到底有多快？例子1-Continued

```
SELECT
    toYear(LO_ORDERDATE) AS yod,
    sum(LO_REVENUE)
FROM lineorderfull
GROUP BY yod
```

	1个节点	3个节点
时间(s)	9.7	3.2

```
SELECT sum(LO_EXTENDEDPRICE * LO_DISCOUNT) AS revenue
FROM lineorderfull
WHERE (toYear(LO_ORDERDATE) = 1993)
 AND ((LO_DISCOUNT >= 1)
 AND  (LO_DISCOUNT <= 3)) AND (LO_QUANTITY < 25)
```

	1个节点	3个节点
时间	3.1	1.29

```
SELECT sum(LO_REVENUE)
FROM lineorderfull
WHERE LO_CUSTKEY IN
(
    SELECT C_CUSTKEY AS LO_CUSTKEY
    FROM customerfull
    WHERE C_REGION = 'ASIA'
)
```

	1个节点	3个节点
时间	28.9	14.1

```
SELECT
    C_REGION,
    sum(LO_EXTENDEDPRICE * LO_DISCOUNT)
FROM lineorderfull
ANY INNER JOIN
(
    SELECT
        C_REGION,
        C_CUSTKEY AS LO_CUSTKEY
    FROM customerfull
) USING (LO_CUSTKEY)
WHERE (toYear(LO_ORDERDATE) = 1993) AND ((LO_DISCOUNT >= 1) AND (LO_DISCOUNT <= 3)) AND (LO_QUANTITY < 25)
GROUP BY C_REGION
```

	1个节点	3个节点
时间	31.4	25.1

- 订单表：150亿行,客户表：7500万行

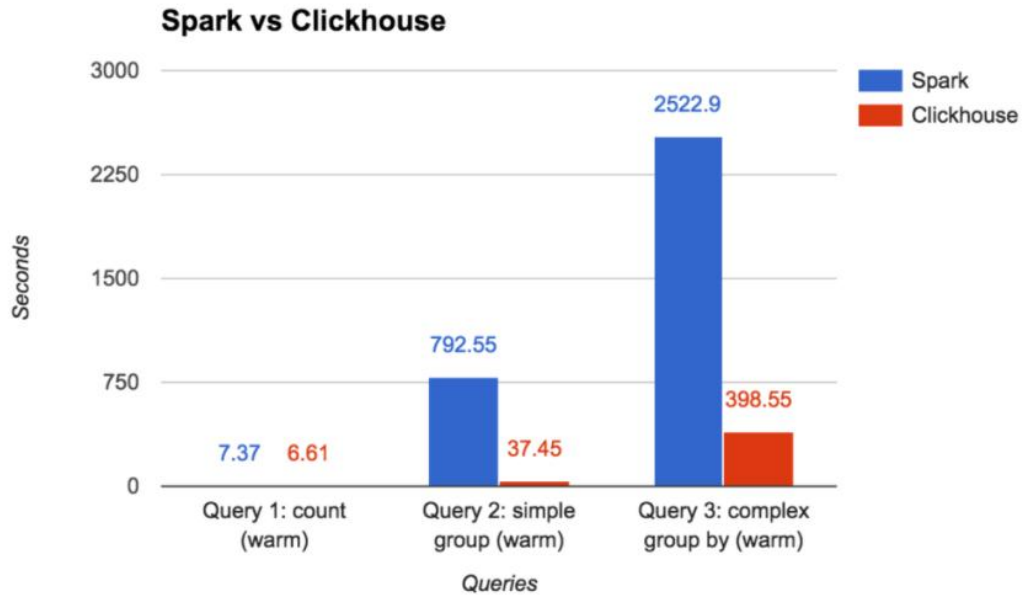
ClickHouse 到底有多快？ClickHouse V.S Spark SQL

[Wikipedia Page Counts]

200亿行

Benchmark summary

Size / compression	***Spark v. 2.0.2***	***ClickHouse***
Data storage format	Parquet, compressed: snappy	Internal storage, compressed
Size (uncompressed: 1.2TB)	395G	212G



Test	***Spark v. 2.0.2***	***ClickHouse***	***Diff***
Query 1: count (warm)	7.37 sec (no disk IO)	6.61 sec	~same
Query 2: simple group (warm)	792.55 sec (no disk IO)	37.45 sec	21x better
Query 3: complex group by	2522.9 sec	398.55 sec	6.3x better

ClickHouse 到底有多快？品友互动

部署：

- 内存128G/CPU24核/CentOS7.1
- 软件：ClickHouse 1.1
- 配置 3 Shards，1 replica

数据表：

- 数据量 30亿+, 多维，Ad-hoc查询
- Distributed ENGINE
- 采用SAMPLE

A	B	C	D	E	F	G	H
查询类型	是否采样	并发数	是否带日期范围	耗时	处理行数	处理速度（rows/s）	处理速度（GB/s）
count查询	否	1	否	0.4s	3.13 billion	7.36 billion	14.73
聚合查询	否	1	是	5.4s	417.96 million	77.14 million	8.74
聚合查询	否	10	是	70s	417.96 million	5.94 million	0.67
聚合查询	否	1	是(所有日期)	42s	3.13 billion	73.50 million	8.33
聚合查询	否	1	否	41s	3.13 billion	74.94 million	8.35
count查询	是	1	否	0.38s	375.17 million	971.52 million	9.72
聚合查询	是	1	是	0.83s	50.10 million	59.93 million	7.26
聚合查询	是	10	是	7.2s	50.10 million	6.93 million	0.83
聚合查询	是	1	是(所有日期)	5.9s	375.17 million	63.52 million	7.7
聚合查询	是	1	否	6.2s	375.17 million	60.34 million	7.2

Sample

ClickHouse 到底有多快？品友互动 2

部署：

- 内存128G/CPU24核/CentOS7.1
- 软件：ClickHouse 1.1
- 配置 3 Shards，1 replicas

数据表（小表）：

- 数据量 5000万+, 多维，Ad-hoc查询
- Distributed ENGINE

A	B	C	D	E	F	G
查询类型	并发数	是否带日期范围	耗时	处理行数	处理速度（rows/s）	处理速度（GB/s）
count查询	1	否	0.02s	54.00 million	1.86 billion	3.73
聚合查询	1	是	0.04s	557.36 thousand	11.43 million	0.71
聚合查询	100	是	0.03s	557.36 thousand	17.35 million	1.07
聚合查询	1	是(所有日期)	0.42s	54.00 million	127.77 million	6.01
聚合查询	1	否	0.38s	54.00 million	141.52 million	6.37

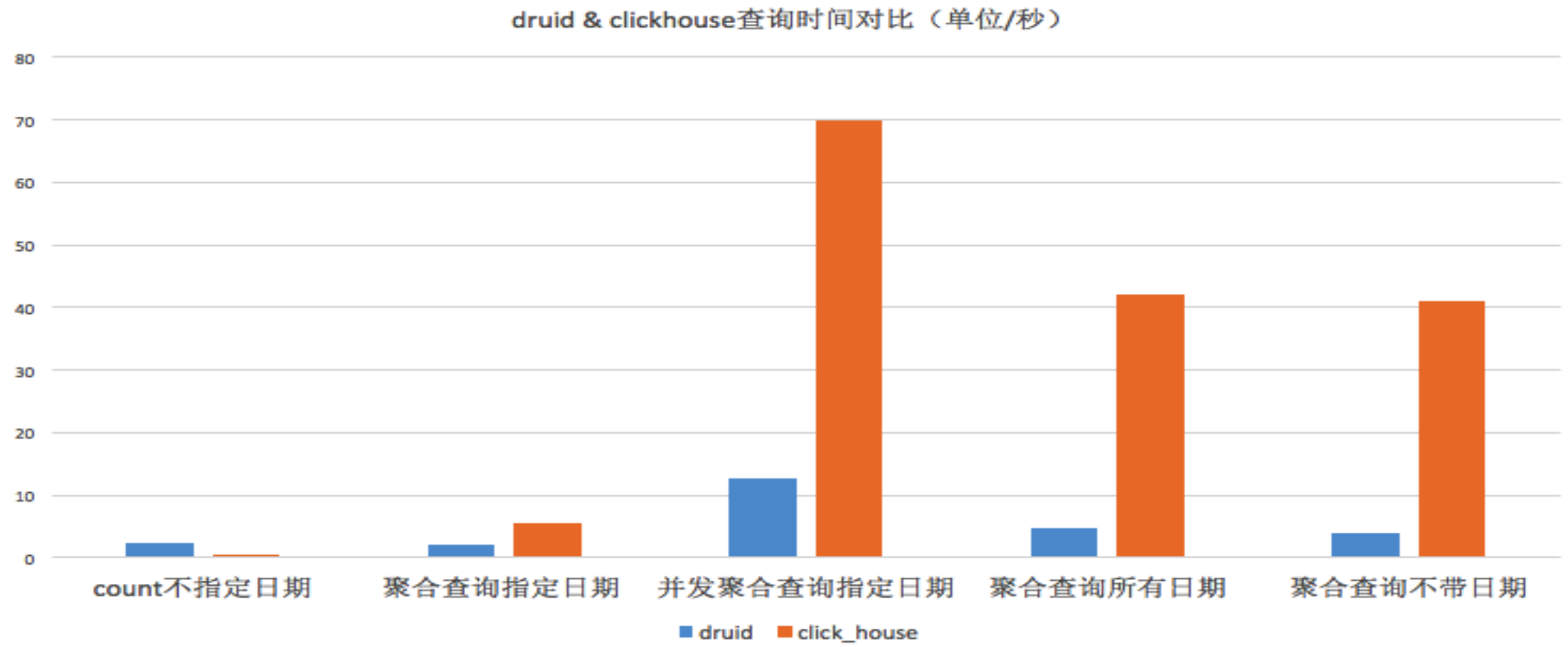
ClickHouse 到底有多快 V.S Druid？品友互动 3.

部署：

- •Druid 0.10.0, Hadoop 2.7,Hive 2.3
- •离线任务加载数据、将hive与druid整合后通过jdbc方式查询
- •2 Historical+2Broker

数据表（小表）：

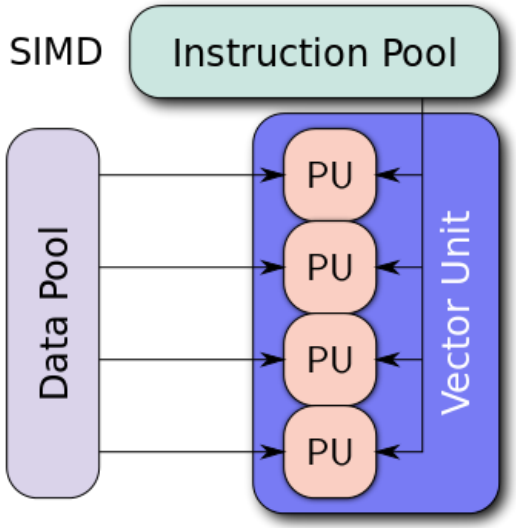
- 数据量比较大，单日5亿数据，多维
- 按小时聚合
- 数据量 30亿+



ClickHouse为什么这么快？

性能为王的原则，每一个改动都需要经过性能测试。

- Vectorized Query Execution技术
  - 利用CPU的SIMD （Single Instruction Multiple Data）
  - 来自VectorWise公司（Actian now!)
  - 参考“*Vectorization vs. Compilation in Query Execution*”
- Runtime Code Generation技术
  - Java JIT/Reflection; C++ LLVM；
- C++ 14特性
  - TCMalloc类似技术



ClickHouse的一些技术能力

接口支持

- 命令行客户端
- HTTP
- JDBC接口
- Python
- PHP
- NodeJS
- Go
- Perl wrappers

性能之外特点

- 数据压缩能力，对于PB级别数据，非常有意义
- 实时数据摄入能力
- 支持模糊计算（approximated calculations）
- 异步多Master备份

非常活跃社区

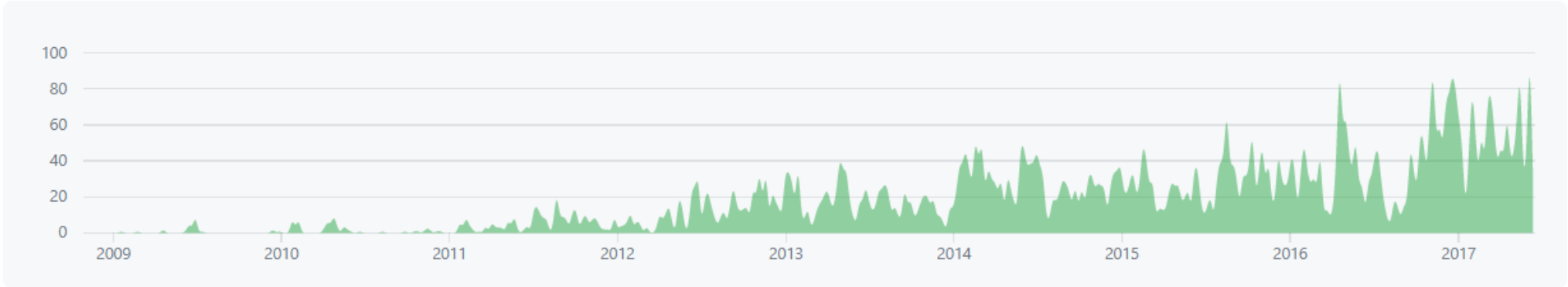
Watch 211 | Star 2,623 | Fork 340

Contributors | Commits | Code frequency | Punch card | Network | Members | Dependents

Nov 30, 2008 – Jul 21, 2017

Contributions: Commits

Contributions to master, excluding merge commits



Click House 开发规划

Q3 2017

- `SYSTEM` queries
- Limit on parallel replica downloads
- Finalize `NULL` support
- `SELECT db.table.column`

Q4 2017

- Arbitrary partitioning key for MergeTree engine family
- Better compliance of `JOIN` syntax with SQL standard
- Resource pools for queries (CPU, disk I/O, network bandwidth)

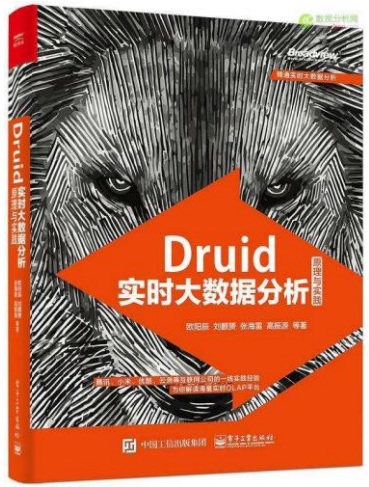
Q1 2018

- Basic support for `UPDATE` and `DELETE`

参考资料

- https://clickhouse.yandex/
- https://www.percona.com/blog/2017/07/06/clickhouse-one-year/
- https://aferdyp.github.io/clickhouse/2017/02/23/building-clickHouse.html

谢谢！



www.ouyangchen.com