同程艺龙

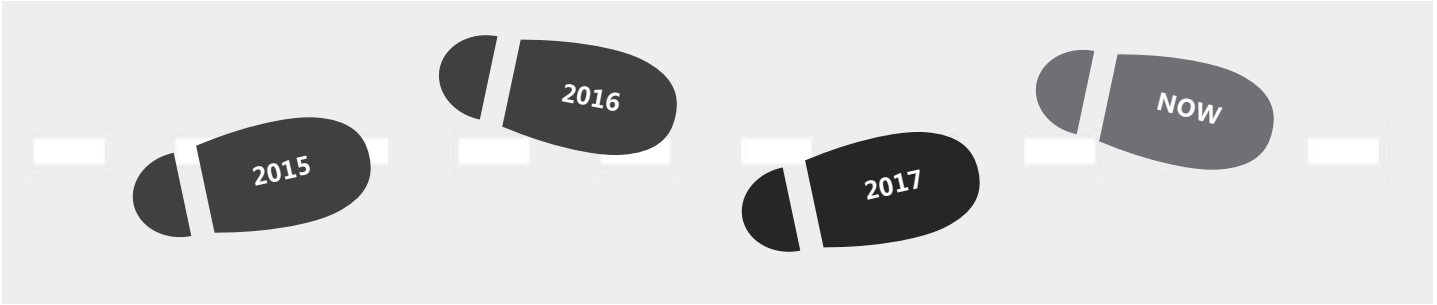
ClickHouse

在海量数据下的实践应用

同程艺龙-谢磊 2018-11月

关于我 (大数据计算研发组)

https://github.com/lamber-ken



Hadoop生态圈

流计算 Flink

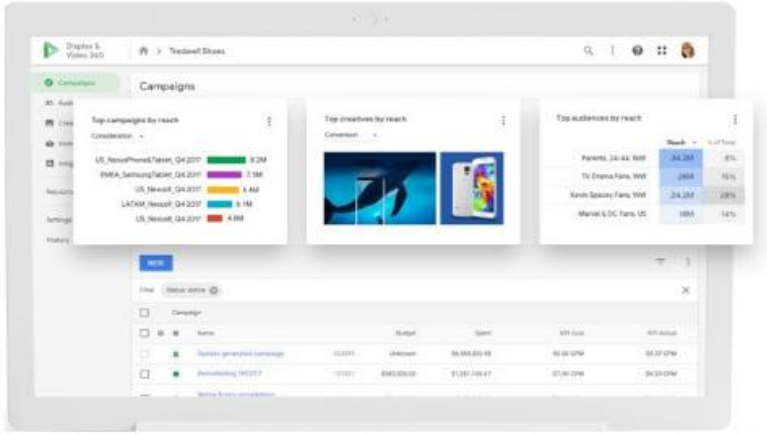
OLAP领域

目录

了解探索

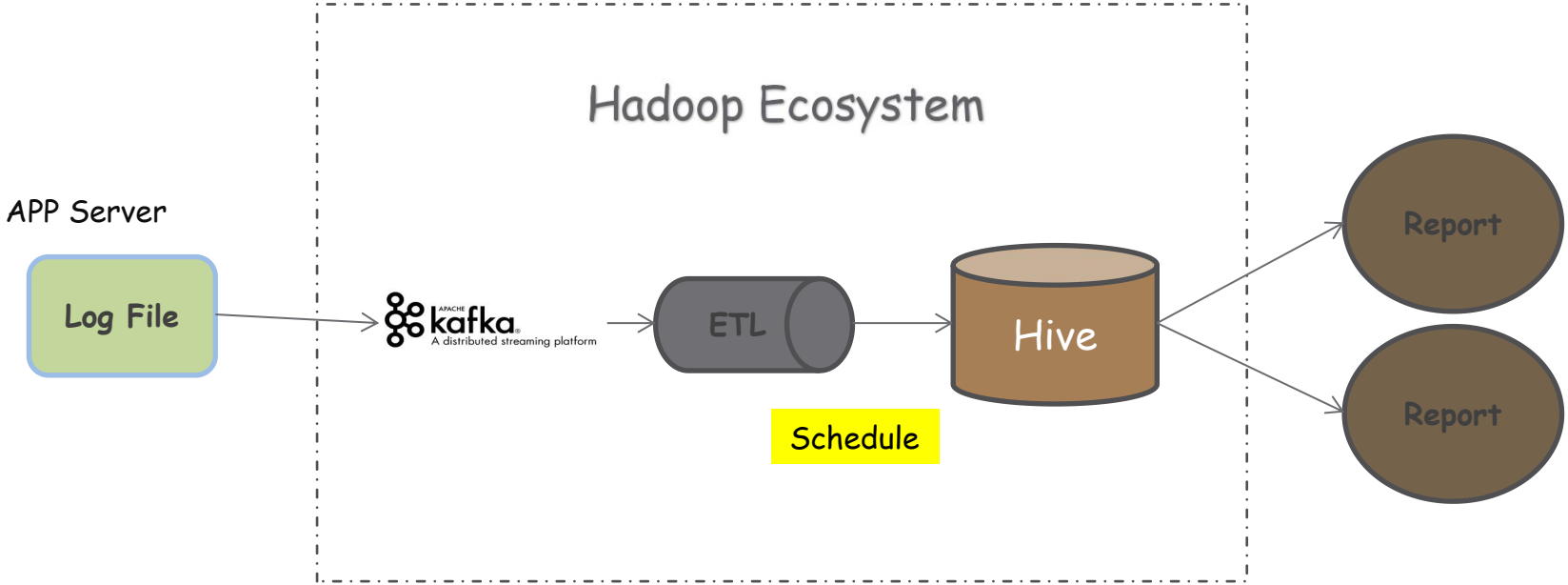
从需求谈起

- 海量数据
- 实时导入
- 实时查询
- 多维 聚合分析



用户轨迹行为分析

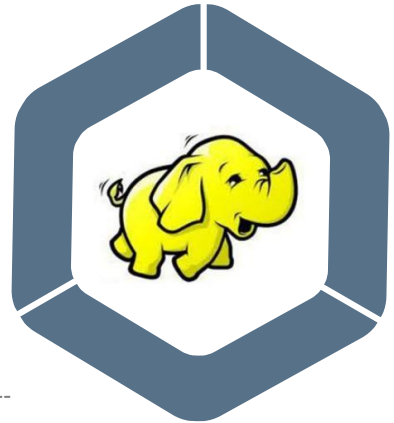
经典架构


Hadoop Ecosystem
APP Server
Log File
APACHE kafka
A distributed streaming platform
ETL
Hive
Schedule
Report
Report

架构分析

数据的实效性

中间过程经过Kafka、ETL、调度处理，报表的实效性不理想



即席分析性能

Hive存储是hdfs文件系统，查询效率不高，不适合即席查询

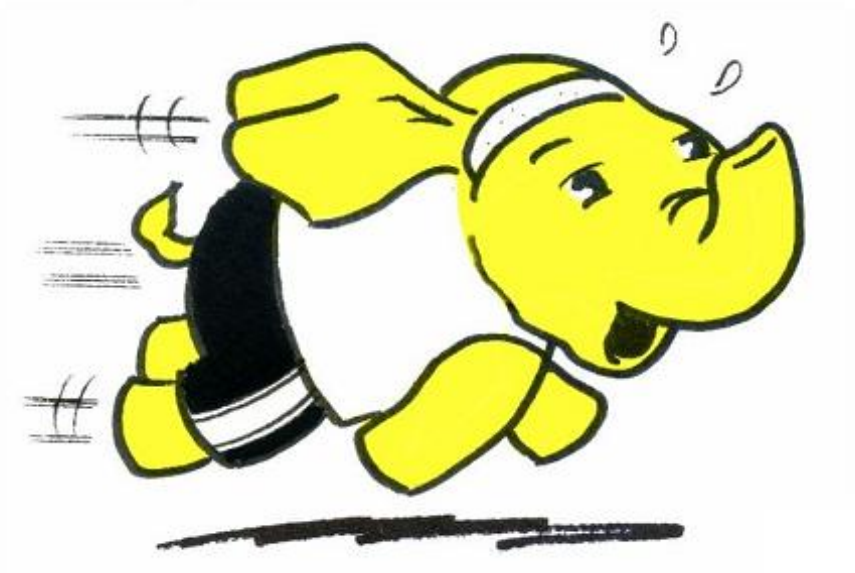
涉及Hadoop组件多

涉及Flume、Kafka、HDFS等等，数据冗余过多，同时需要深厚的知识储备

数据链路长

数据链路处理流程长，繁琐容错也不好

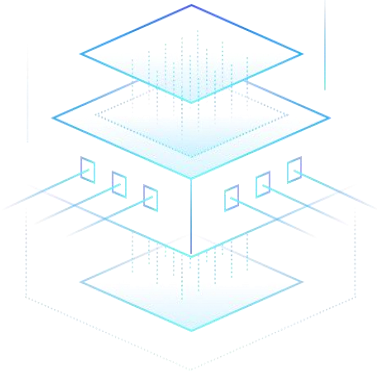
美好愿景



竞品分析

Apache Druid

Apache Kylin

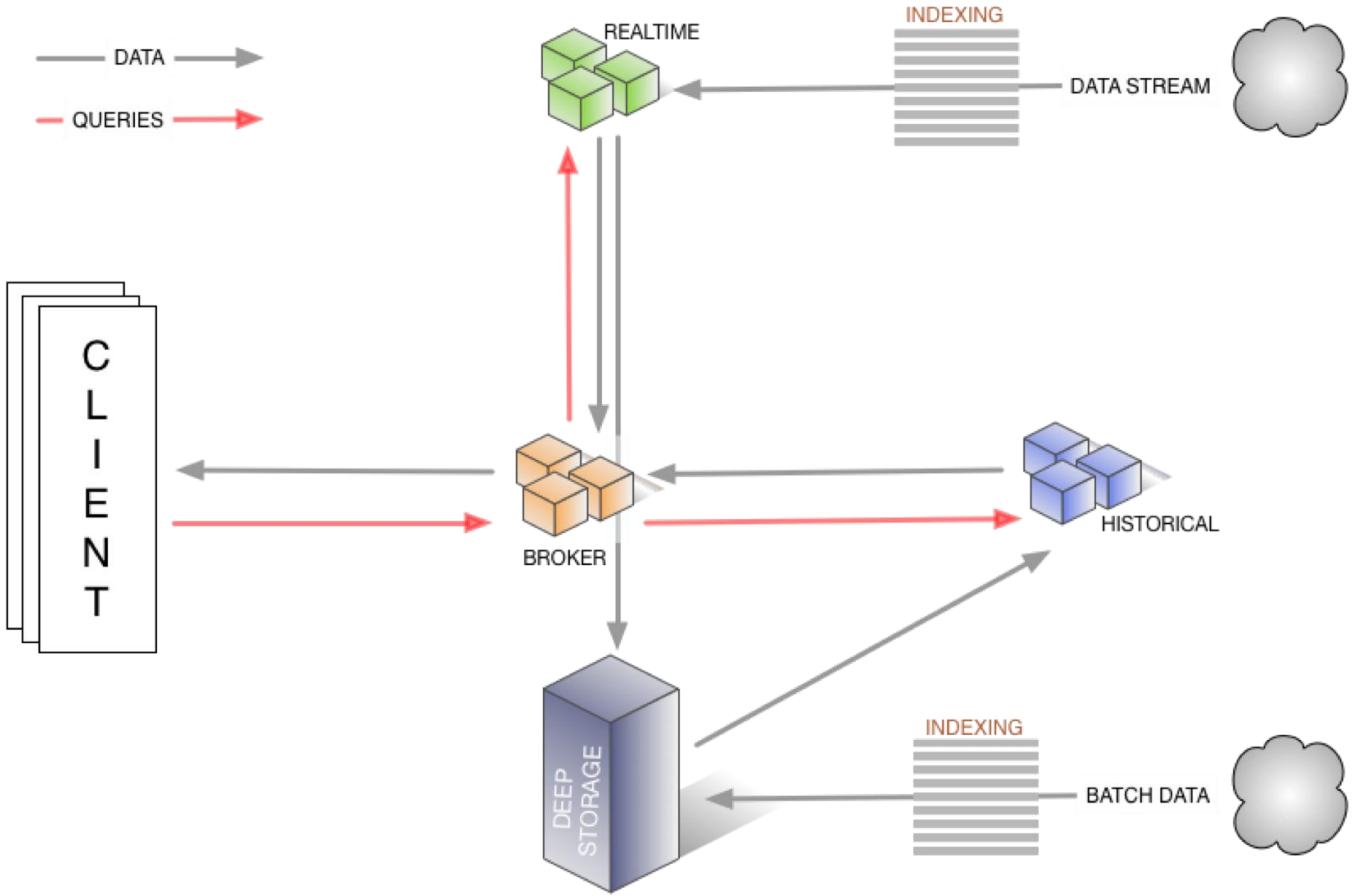


Apache Doris

同程艺龙

Druid

回到原点 !!!



其他方案

Apache Kylin

Apache Doris

...

同程艺龙

Meetup

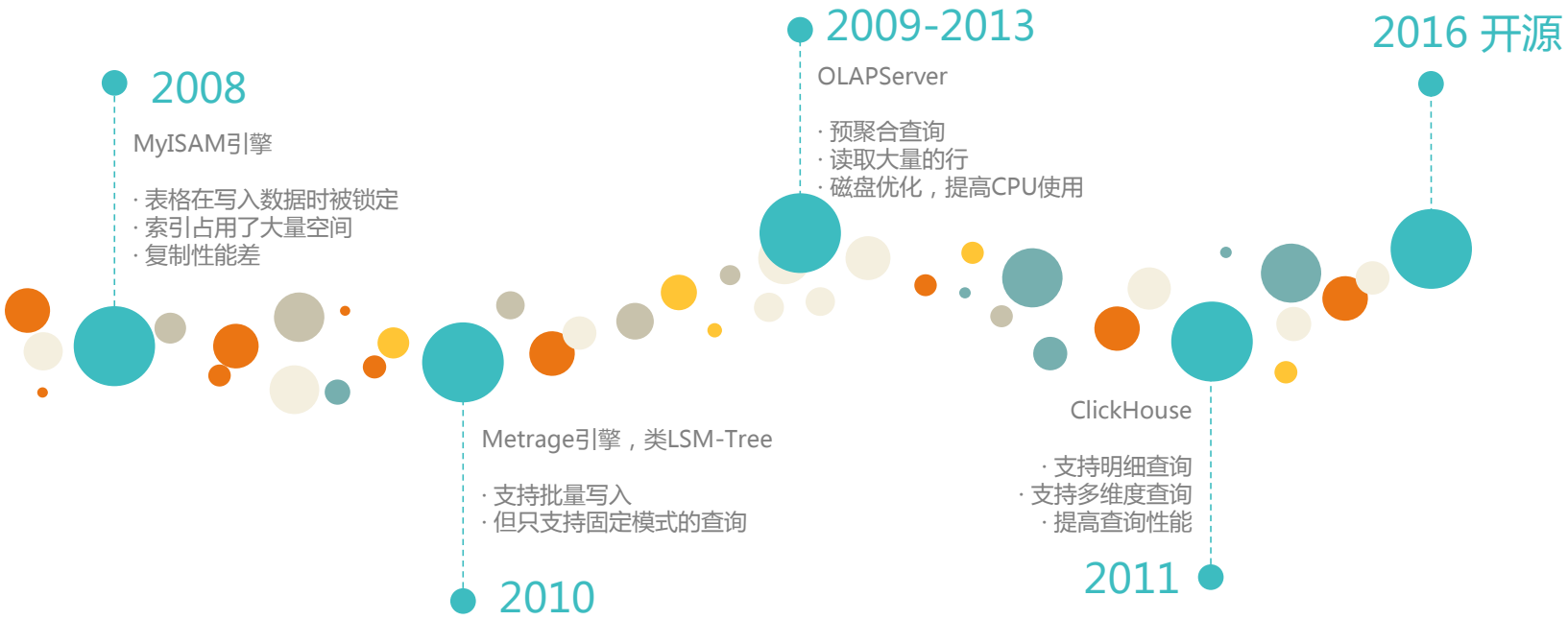
ClickHouse

https://clickhouse.yandex

同程艺龙

Yandex Яндекс

ClickHouse



同程艺龙

ClickHouse (https://github.com/yandex/ClickHouse)

列式存储

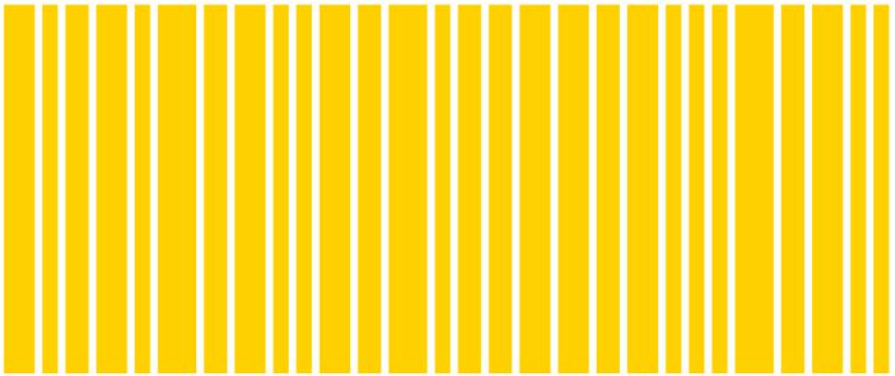
数据极限压缩

原生支持 SQL

极致性能

客户端丰富

明细查询



ClickHouse

不支持 Transaction 即OLTP

不适合K-V存储

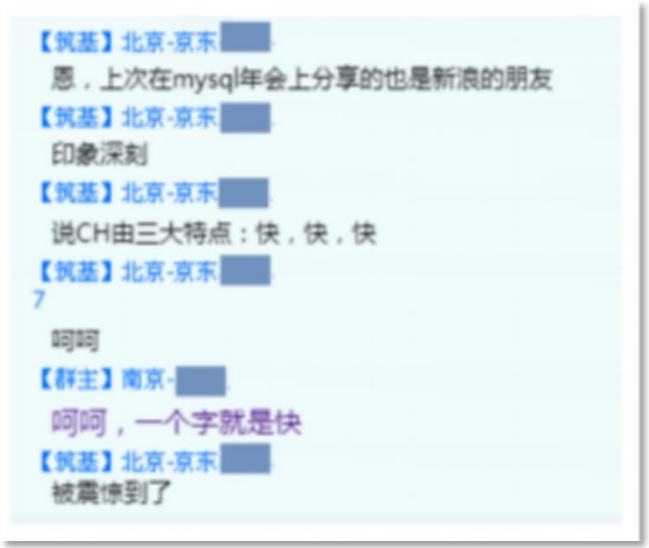
不支持Blob等文档类型数据

Update/Delete 支持不完善

同程艺龙

社区评价


【筑基】杭州-
同步使用mysql、hive、ck
【筑基】杭州-
ck及时查询 速度快

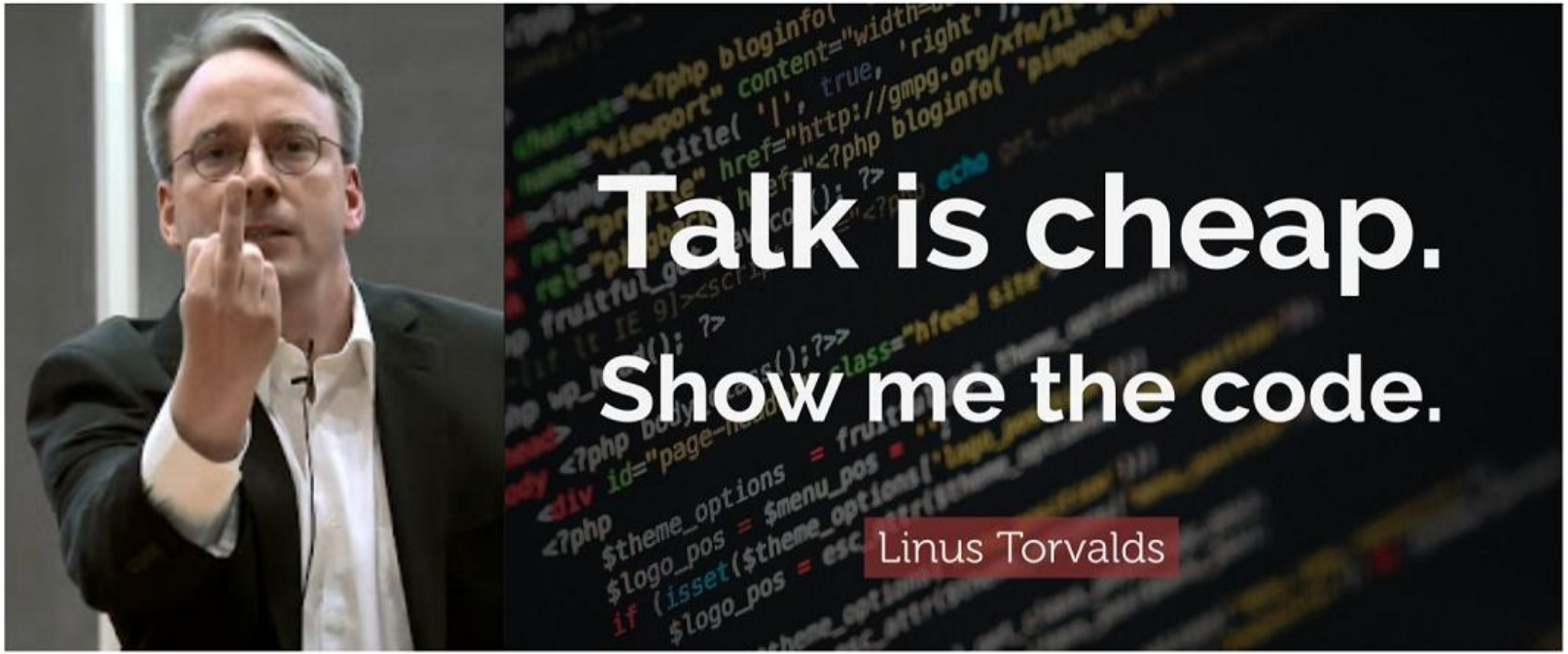

【筑基】北京-京东
恩，上次在mysql年会上分享的也是新浪的朋友
【筑基】北京-京东
印象深刻
【筑基】北京-京东
说CH由三大特点：快，快，快
【筑基】北京-京东
7
呵呵
【群主】南京-
呵呵，一个字就是快
【筑基】北京-京东
被震惊到了


【渡劫】北京-
我觉得clickhouse性能很厉害了


【分神】CMD
当时亲手测试过一些指标。当场吓傻！呵呵！


【元婴】大连-大羊-
我光 跑一下美航数据 1，8亿那个 就已经服了
【元婴】北京-
这么牛逼

邏一邏



同程艺龙

Try Demo

- I5-4590CPU 8GMEM 20GSSD
- Docker容器单节点部署
- 1亿航班数据下载以及导入

109 字段

31G原始数据

```
## Docker安装
docker run -d --name test-ck-server \
    yandex/clickhouse-server

docker run -itd --name test-ck-client \
    --link test-ck-server:clickhouse-server \
    yandex/clickhouse-client --host clickhouse-server

## 下载ZIP数据
for s in `seq 1987 2017`
do
for m in `seq 1 12`
do
wget http://transtats.bts.gov/PREZIP\
/On_Time_On_Time_Performance_${s}_${m}.zip
done
done

## 数据导入
for i in *.zip;
do
echo $i; unzip -cq $i '*.csv' | sed 's/\.00//g' | \
clickhouse-client --host=clickhouse-server \
--query="INSERT INTO ontime FORMAT CSVWithNames";
done
```

https://clickhouse.yandex/docs/en/single/#ontime

数据压缩

```
SELECT
    table,
    sum(rows) AS rows,
    formatReadableSize(sum(data_compressed_bytes) AS c) AS comp,
    formatReadableSize(sum(data_uncompressed_bytes) AS r) AS raw,
    c / r AS comp_ratio
FROM system.parts
WHERE active
GROUP BY table
```

```
┌─table──┬──────rows─┬─comp─────┬─raw──────┬─com
p_ratio─┐
│ ontime │ 100618278 │ 7.71 GiB │ 31.23 GiB │  0.2469174
└────────┴───────────┴──────────┴──────────┴──────────
```

https://clickhouse.yandex/docs/en/single/#ontime

聚合查询

```
SELECT avg(c1)
FROM
(
    SELECT
        Year,
        Month,
        count(*) AS c1
    FROM ontime
    GROUP BY Year, Month
)
```

```
1 rows in set. Elapsed: 0.663 sec. Processed 100.62 million rows.
```

https://clickhouse.yandex/docs/en/single/#ontime

简单易用

性能强悍

进一步深入

MySQL建表

```
CREATE TABLE test (
          id INT (10) NOT NULL auto_increment,
          NAME VARCHAR (10),
          age INT (1),
          PRIMARY KEY (id)
)
```

```
ENGINE = MyISAM [Innodb]
```

（默认）

MyISAM　　表级锁、不是事物安全的、适合大量SELECT 等

Innodb　　行级锁、支持事物 等

建表语法

```
CREATE TABLE datacenter.events ON CLUSTER datacenter
(
    partition Date,
    hour UInt8,
    min UInt8,
    plattype String,
    tracktype String,
    platid Int32
)
ENGINE = MergeTree
PARTITION BY partition
ORDER BY (hour, min, plattype, tracktype)
SETTINGS index_granularity = 8192;
```

查询语法

```
SELECT [DISTINCT] expr_list
    [FROM [db.]table | (subquery) | table_function] [FINAL]
    [SAMPLE sample_coeff]
    [ARRAY JOIN ...]
    [GLOBAL] ANY|ALL INNER|LEFT JOIN (subquery)|table USING columns_list
    [PREWHERE expr]
    [WHERE expr]
    [GROUP BY expr_list] [WITH TOTALS]
    [HAVING expr]
    [ORDER BY expr_list]
    [LIMIT [n, ]m]
    [UNION ALL ...]
    [INTO OUTFILE filename]
    [FORMAT format]
    [LIMIT n BY columns]
```

丰富函数

- Functions for working with strings
- Functions for working with dates and times
- Mathematical functions
- Higher-order functions

```
lambda --->   x -> 2 * x, str -> str !=
Referer.
```

丰富表引擎

MergeTree

- ReplicatedMergeTree

Replace / Update

- CollapsingMergeTree
- ReplacingMergeTree

Pre-aggregate

- AggregatingMergeTree

Special

- Log、Buffer、MaterializedView

Distributed

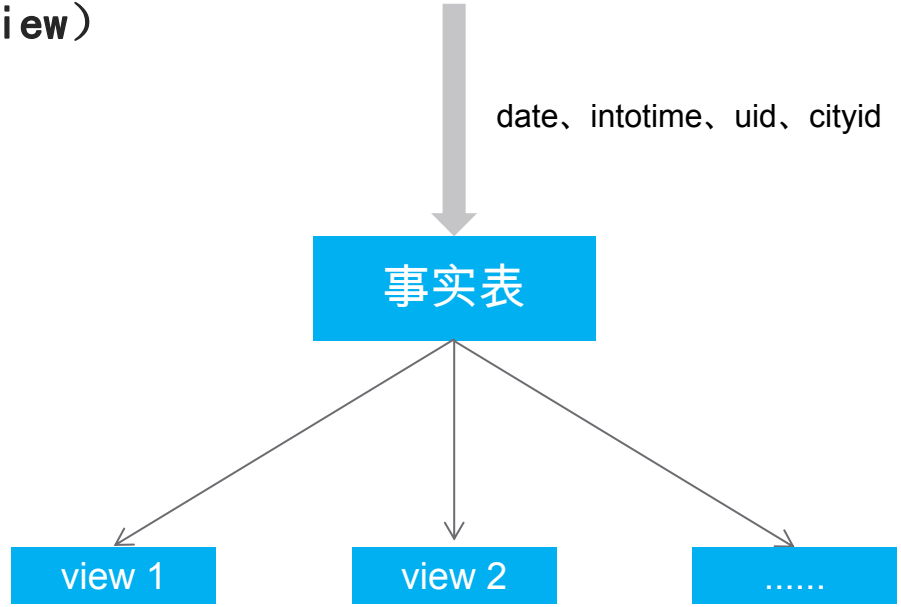
Distributed(logs, default, hits[, sharding_key])

- 本身不存储数据，需要结合其他引擎

物化视图(MaterializedView)

- 类CUBE方式，避免查询海量明细数据
- 任意时刻可 DDL 操作
- 实时聚合，无需调度
- 事实表引擎可为NULL，提升空间利用

!! 注意额外写入压力



同程艺龙

物化视图(MaterializedView)

物化视图

```
CREATE MATERIALIZED VIEW
datacenter.report
ENGINE = AggregatingMergeTree
PARTITION BY partition
ORDER BY (hour, min)
AS SELECT
    partition,
    toHour(intotime) AS hour,
    toMinute(intotime) AS min,
    countState() AS pv,
    uniqState(uid) AS uv
FROM datacenter.event
GROUP BY
    partition,
    hour,
    min
;
```

原始表

```
CREATE TABLE datacenter.event
(
    partition Date,
    intotime DateTime,
    uid String,
    cityid Int64
)
ENGINE = MergeTree
PARTITION BY (partition)
ORDER BY (cityid)
;
```

查询示例

```
SELECT
    partition,
    countMerge(pv) AS pv,
    uniqMerge(uv) AS uv
FROM datacenter.report
GROUP BY partition
```

```
┌─partition─┬─pv─┬─uv─┐
│ 2018-10-22 │  4 │  2 │
└────────────┴────┴────┘
```

其他特性

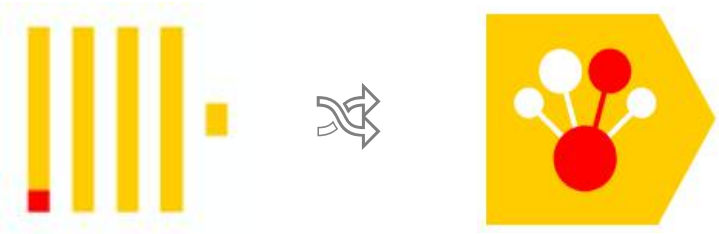
mysql(..)

```
SELECT uniq(uid) FROM mysql('10.10.10.10:3306', '<database>', '<table>', 'root', '123456')
```

odbc(..) --> SQLSERVER、POSTGRES

```
INSERT INTO history
SELECT *
FROM odbc('Driver={ODBC Driver 13 for SQL Server};Server=10.10.10.10;
Address=10.10.10.10,2433;Database=<database>;Uid=<user>;Pwd=<password>', '<table>')
```

其他特性(SQL + ML)



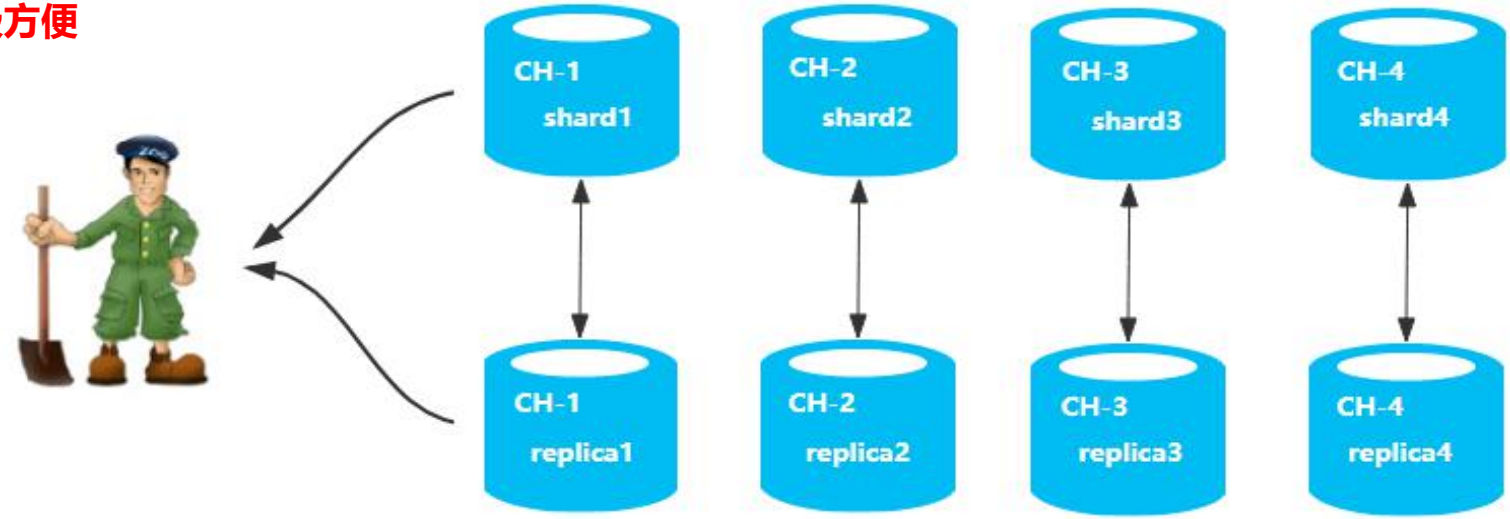
https://catboost.ai

http://madlib.apache.org

部署实践

部署方案 SaltStack + Docker + ZooKeeper

- 配置参数**变量化**
- **升级方便**



表引擎：ReplicatedMergeTree + Distributed

同程艺龙

重点监控

prometheus + clickhouse_exporter

同程艺龙

重点监控

prometheus + clickhouse_exporter

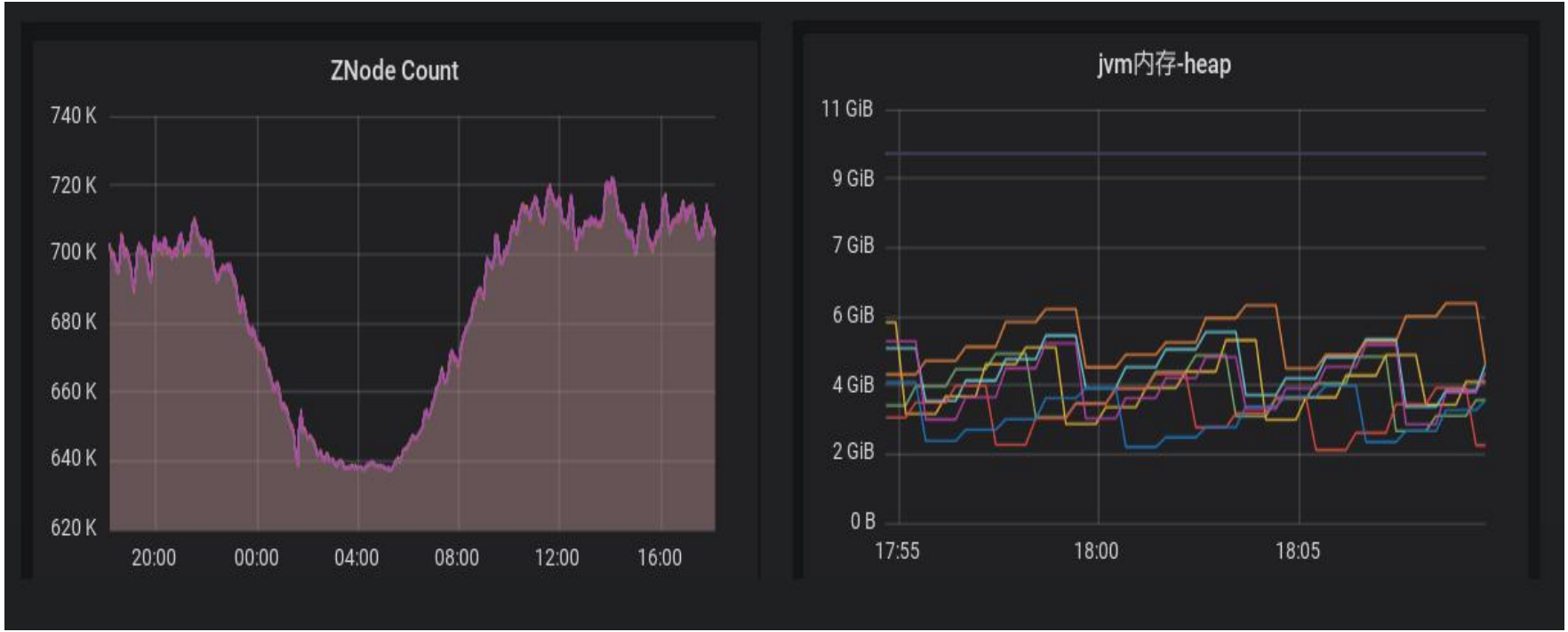
Top slow queries by type: All; user: All; query type: select

minstart	duration	count	query
2018-11-07 18:02:59	283.50 ms	4	SELECT [redacted] /HERE (partition = '2018-11-07 (ptime <= '2018-11-07 18:02:00') AND (unionid = '')
2018-11-07 18:01:59	283.50 ms	6	SELECT [redacted] WHERE (partition = '2018-11-07 (ptime <= '2018-11-07 18:01:00') AND (unionid = '')
2018-11-07 18:00:59	274.75 ms	4	SELECT [redacted] WHERE (partition = '2018-11-07 (ptime <= '2018-11-07 18:00:00') AND (unionid = '')
2018-11-07 17:59:59	206.33 ms	6	SELECT [redacted] WHERE (partition = '2018-11-07 (ptime <= '2018-11-07 17:59:00') AND (unionid = '')
2018-11-07 17:58:59	221.67 ms	6	SELECT [redacted] WHERE (partition = '2018-11-07 (ptime <= '2018-11-07 17:58:00') AND (unionid = '')
2018-11-07 17:58:00	223.50 ms	6	SELECT [redacted] WHERE (partition = '2018-11-07 (ptime <= '2018-11-07 17:57:00') AND (unionid = '')
2018-11-07 17:57:00	201.50 ms	4	SELECT [redacted] WHERE (partition = '2018-11-07 (ptime <= '2018-11-07 17:56:00') AND (unionid = '')

同程艺龙

重点监控

prometheus + clickhouse_exporter



生产案例

同程艺龙

type All | top elements 30 | initial user All | query type select

数据总量

630,144,589,336

各DB的总量

database	parts	rows	comp	raw	comp_ratio
	69404	423110352105	38.22 TiB	134.53 TiB	0.28
	1688	149904728988	7.30 TiB	106.56 TiB	0.07
	446	6559928707	201.28 GiB	1.82 TiB	0.11
	334	13066741473	255.63 GiB	1.01 TiB	0.25
	203	23104531295	626.68 GiB	2.29 TiB	0.27
	163	12633613519	580.89 GiB	2.42 TiB	0.23
	108	4211197326	313.68 GiB	1.82 TiB	0.17
	15	14839210	1.89 GiB	4.30 GiB	0.44

架构改进

- 数据实效性高
- 分析性能强
- 数据链路短

留存改进

```
SELECT
    sumIf(a, a >= 1) AS d0,
    sumIf(b, a >= 1) AS d1
FROM
(
    SELECT
        groupBitOrIf(1, (partition = '2018-08-14')) AS a,
        groupBitOrIf(1, (partition = '2018-08-15')) AS b
    FROM events
    WHERE partition IN ('2018-08-14', '2018-08-15')
    GROUP BY loginkey
)
```

行转列计算，提高CPU利用率

uid1	2018-08-06		2018-08-08
uid2		2018-08-07	
uid3	2018-08-06		

留存改进

```
SELECT
    sum(r[1]) AS r1,
    sum(r[2]) AS r2,
    sum(r[3]) AS r3
FROM
(
    SELECT
        uid,
        retention(date = '2018-08-10', date = '2018-08-11', date = '2018-08-12') AS r
    FROM events
    WHERE date IN ('2018-08-10', '2018-08-11', '2018-08-12')
    GROUP BY uid
)
```

同程艺龙

经验分享

1. 数据如何高速写入ClickHouse集群？

```
<dependency>
    <groupId>ru.yandex.clickhouse</groupId>
    <artifactId>clickhouse-jdbc</artifactId>
    <version>0.1.42</version>
</dependency>
```

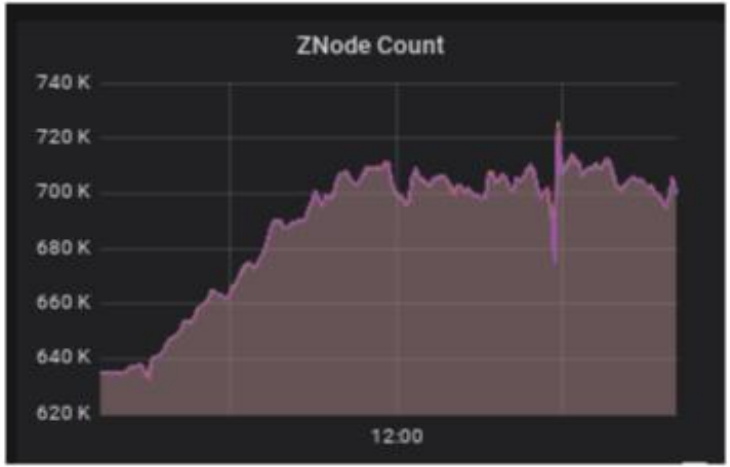
插入写入时，采用 BalancedClickhouseDataSource 轮询 写入**local表**

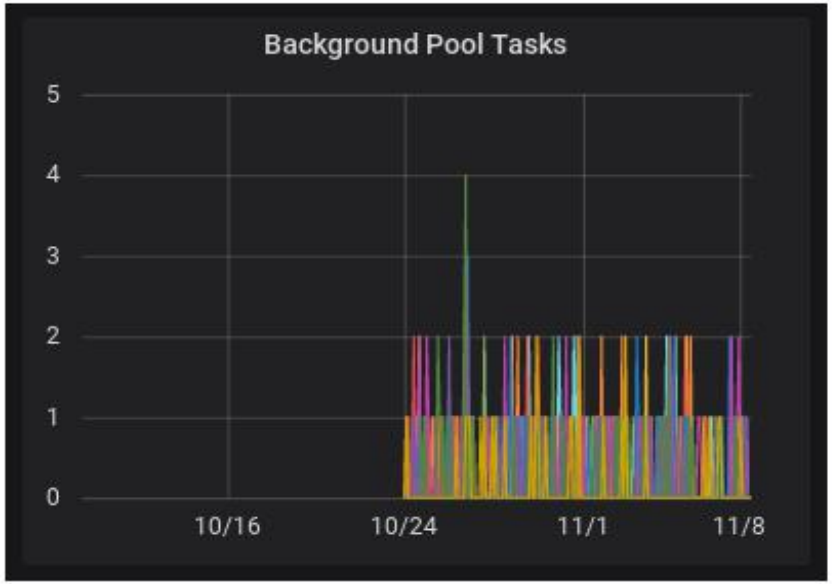
2. 可以connection 连接池吗？

```
private void inseartData() throws Exception {

    Connection connection = dataSource.getConnection();
    String sql = "INSERT INTO test (date,name) values (?,?)";
    PreparedStatement statement = connection.prepareStatement(sql);

    for (int i = 0; i < 1000; i++) {
        statement.setDate(1, new Date());
        statement.setString(2, data);
        statement.addBatch();
    }
    statement.executeBatch();
    statement.close();
    connection.close();
}
```

每次获取新connection，拒绝使用连接池
否则会失去负载均衡的意义

3. DB::Exception: Too many parts. Merges are processing significantly slower than inserts





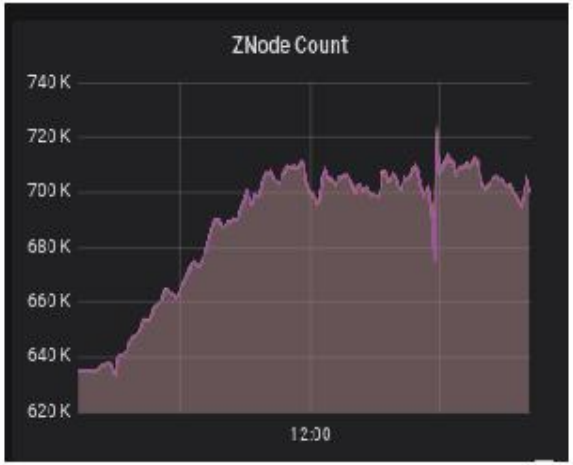
原因

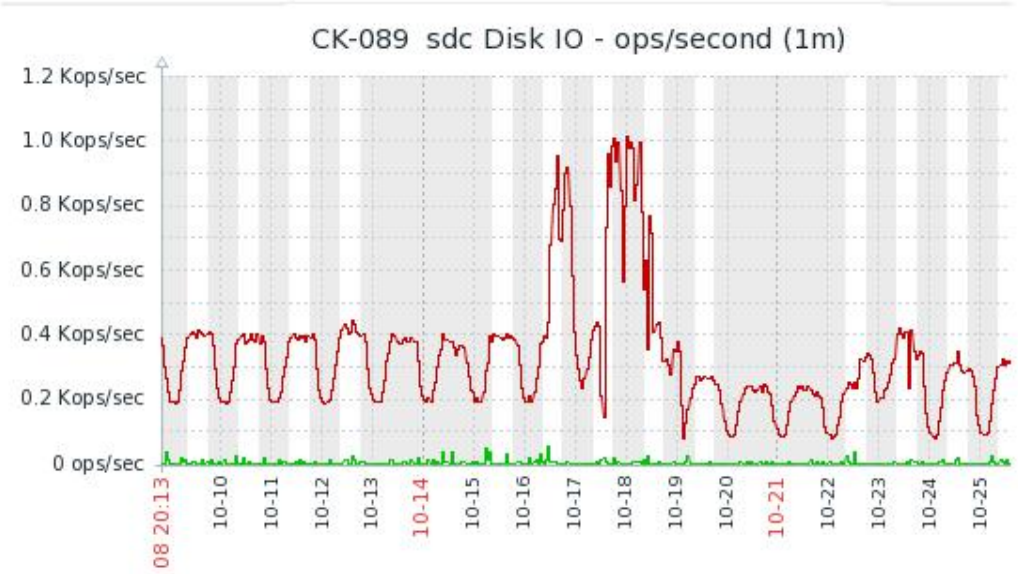
高并发，低批次插入（size < 100）

解决

不能单条插入，batch size 5000 起步

4. DB::Exception: Cannot allocate block number in ZooKeeper





原因

每个批次里面，partition很多

解决

解决脏数据后再进行插入

6. 强制系统保留内存给操作系统

```
# centos6
sysctl -a  | egrep  'min_free_kbytes|extra_free_kbytes'
vm.min_free_kbytes = 90112
vm.extra_free_kbytes = 0


# centos7

sysctl -a  | egrep  'min_free_kbytes|extra_free_kbytes'
vm.min_free_kbytes = 90112
```

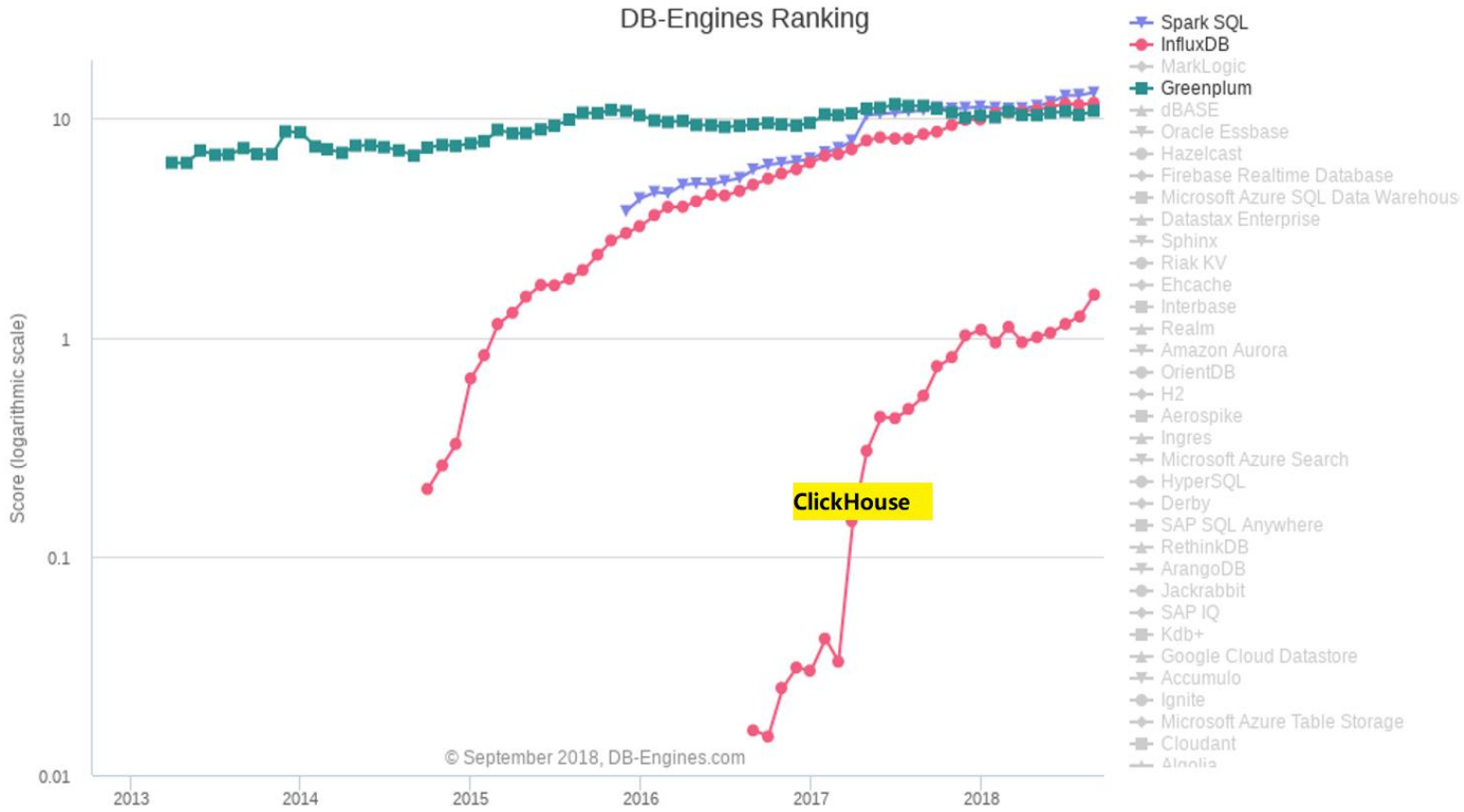
7. 其他坑？

https://clickhouse.yandex

http://www.clickhouse.com.cn

https://github.com/yandex/ClickHouse/issues

社区生态


DB-Engines Ranking
Score (logarithmic scale)
10
1
0.1
0.01
2013
2014
2015
2016
2017
2018
ClickHouse
© September 2018, DB-Engines.com
Spark SQL
InfluxDB
MarkLogic
Greenplum
dBASE
Oracle Essbase
Hazelcast
Firebase Realtime Database
Microsoft Azure SQL Data Warehous
Datastax Enterprise
Sphinx
Riak KV
Ehcache
Interbase
Realm
Amazon Aurora
OrientDB
H2
Aerospike
Ingres
Microsoft Azure Search
HyperSQL
Derby
SAP SQL Anywhere
RethinkDB
ArangoDB
Jackrabbit
SAP IQ
Kdb+
Google Cloud Datastore
Accumulo
Ignite
Microsoft Azure Table Storage
Cloudant
Algolia

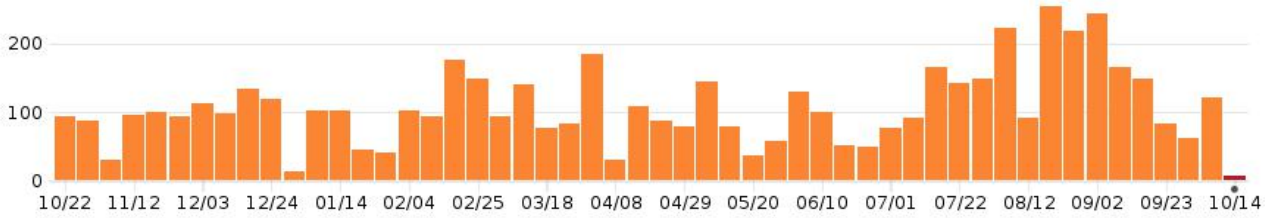
同程艺龙

yandex / **ClickHouse**

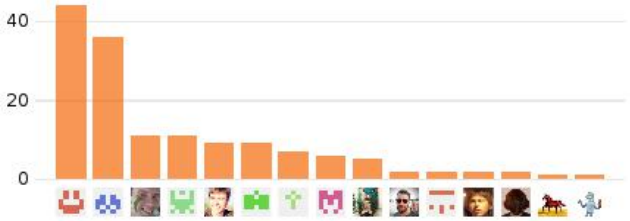
Watch 391 | Star 5,227 | Fork 807

Code | Issues 529 | Pull requests 29 | Projects 0 | Wiki | Insights

- Pulse
- Contributors
- Community
- **Commits**
- Code frequency
- Dependency graph
- Network
- Forks



Excluding merges, **19 authors** have pushed **109 commits** to master and **152 commits** to all branches. On master, **236 files** have changed and there have been **4,287 additions** and **1,447 deletions**.



国内社区

同程艺龙

京东

新浪

科大讯飞

虎牙直播

贝壳

青云

饿了么

http://www.clickhouse.com.cn

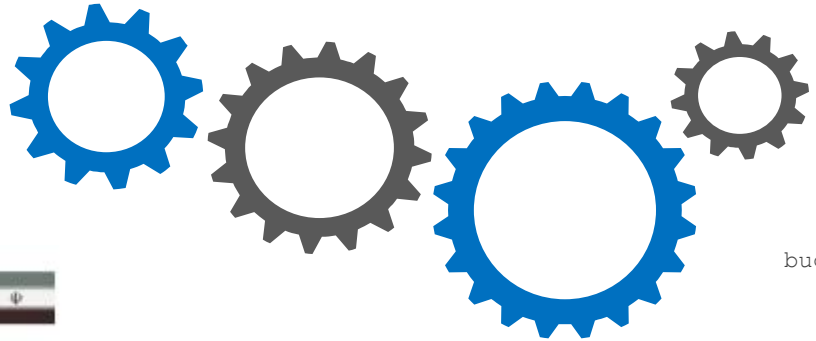
社区工作

GITHUB

https://github.com/yandex/ClickHouse

retention留存函

社区合作推进，方便留存分析



官方中文翻

https://clickhouse.yandex/docs/zh/single

countEqual

bug修复，返回值超过 4294967295 则返回0

ZKADMIN

https://github.com/lamber-ken/zookeeper-admin

希望大家有收获

同程艺龙

感谢聆听