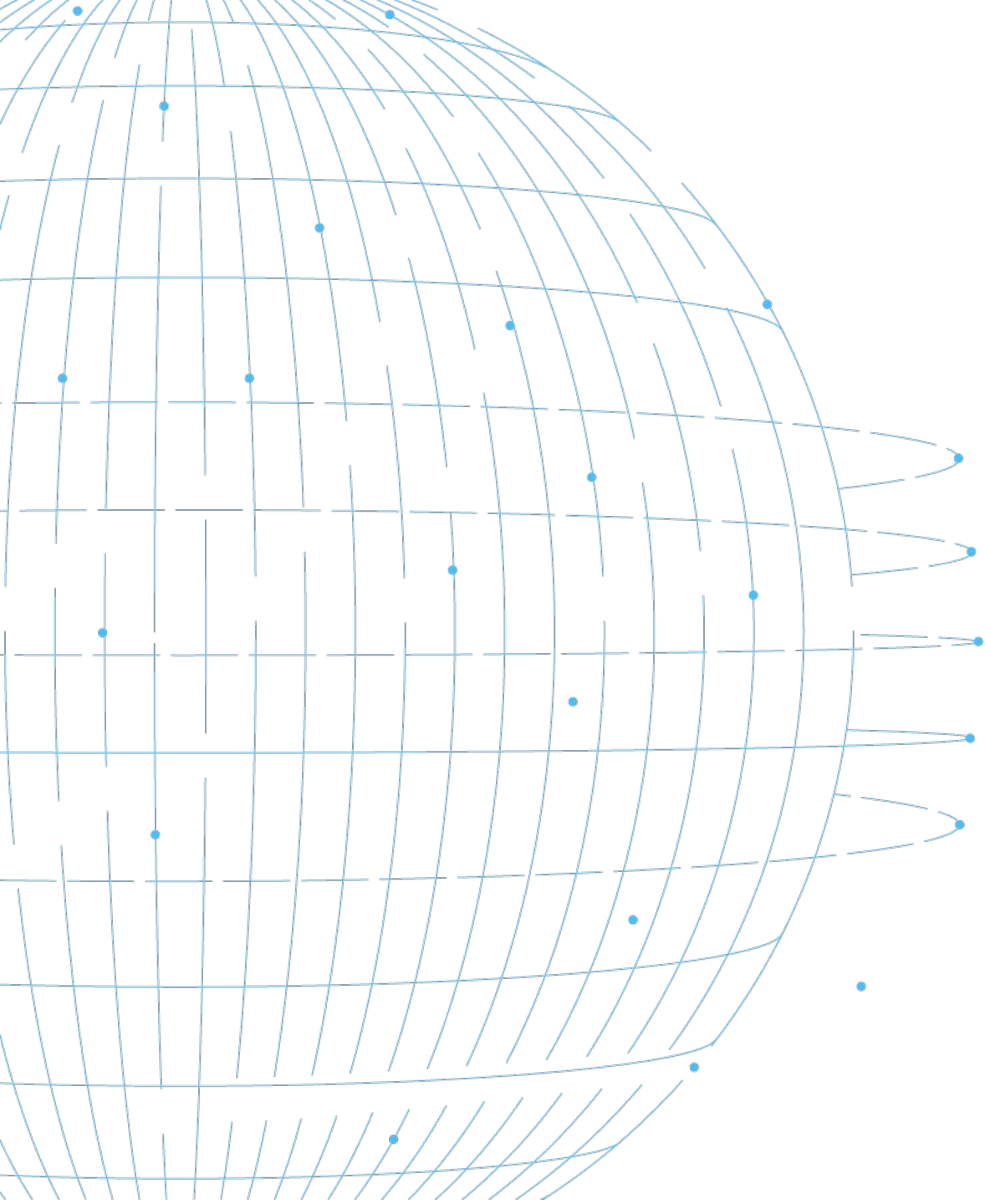


# ClickHouse在BIGO的 实践及优化

徐帅 2021.06



- 目录

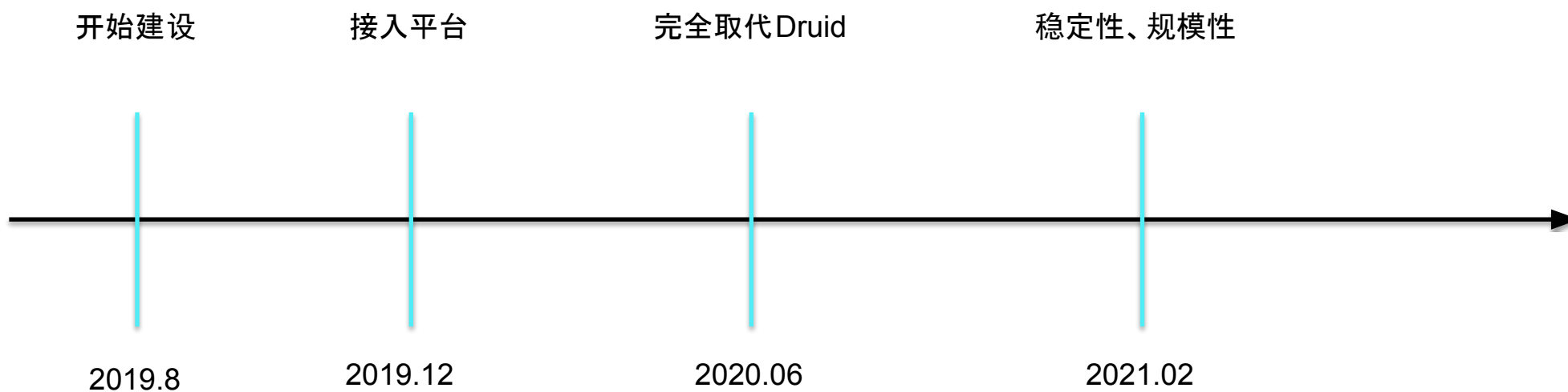
- ClickHouse在BIGO的发展及现状
- 数据接入平台
- 核心改造
- 业务场景
- 未来展望



# 01

## ClickHouse在BIGO的发展及现状

# ClickHouse在BIGO的发展及现状



# ClickHouse在BIGO的发展及现状

规模：

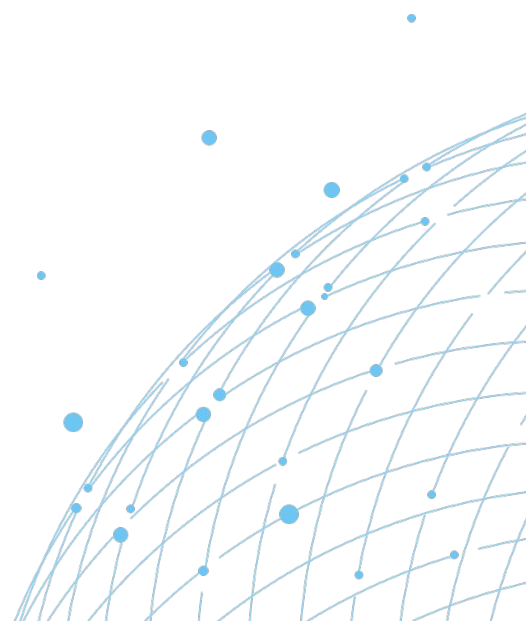
- 200+服务器
- 4K+张表
- 总量万亿条记录
- 日增百亿条

覆盖

- 数据分析、广告、直播、客户端、ToB等公司所有业务线

业务场景

- 多维分析、ETL、监控、ABTest





# 02

## 数据接入平台

# ClickHouse接入平台

背景：

- ClickHouse对绝大部分业务用户门槛有点高
- 统一管控
- 降低ClickHouse Server的连接数

# ClickHouse接入平台

## 统一的接入平台

- 用户只需申请上报, 填上必要的参数即可
- 自动创建local及分布式表
- 自动审核
- 数据自动接入

新增元数据

×

\* 名称

任务名称, 唯一, 不能以 \_all 结尾

描述

请输入 描述

\* 类型

请选择

▼

\* schema信息

查看

\* 实例个数 ?

1

\* table engine

ReplicatedMergeTree(默认引擎)

▼

行过滤器 ?

过滤字段

过滤条件

过滤值

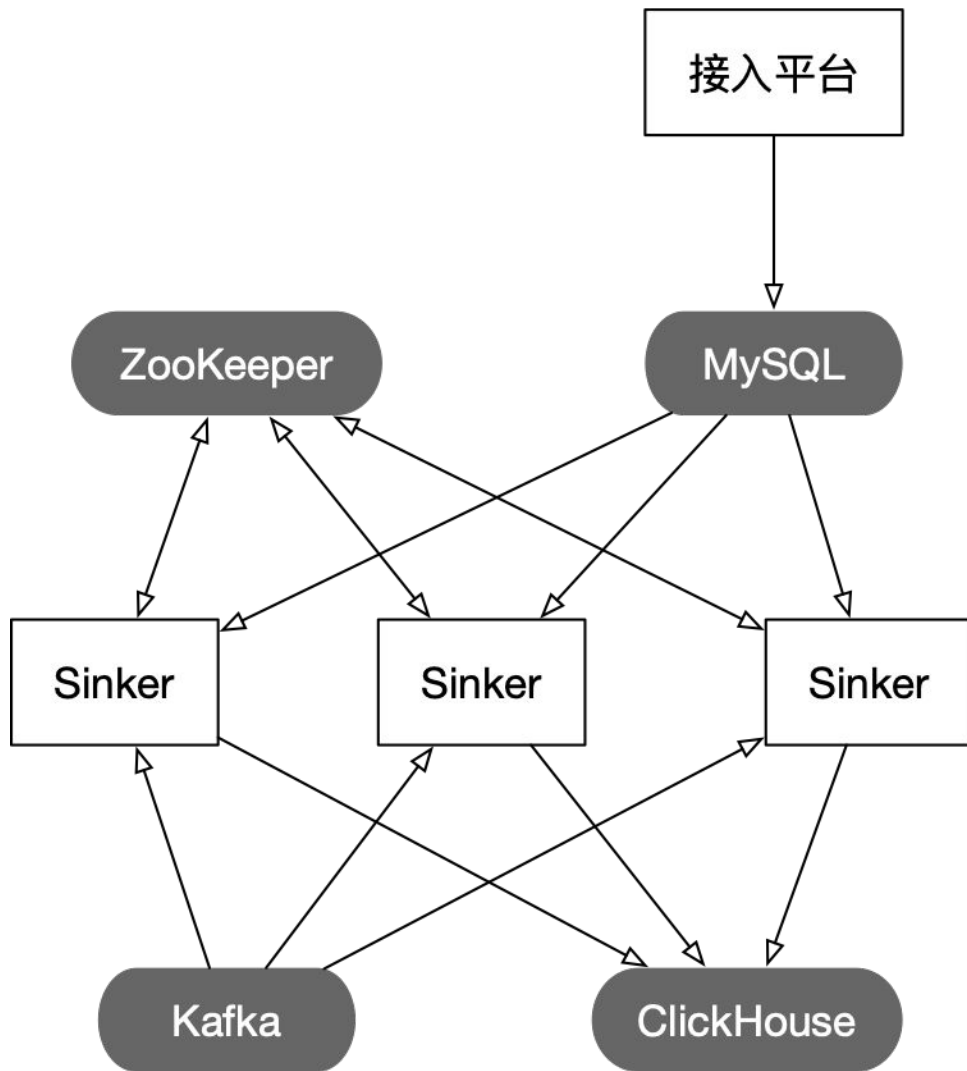
+



# ClickHouse接入平台

接入平台架构

- 高可用
- 负载均衡



# ClickHouse接入平台

支持公司内所有的数据格式：

- Json
- ProtoBuf
- Baina
- 正则日志
- Mysql的Binlog

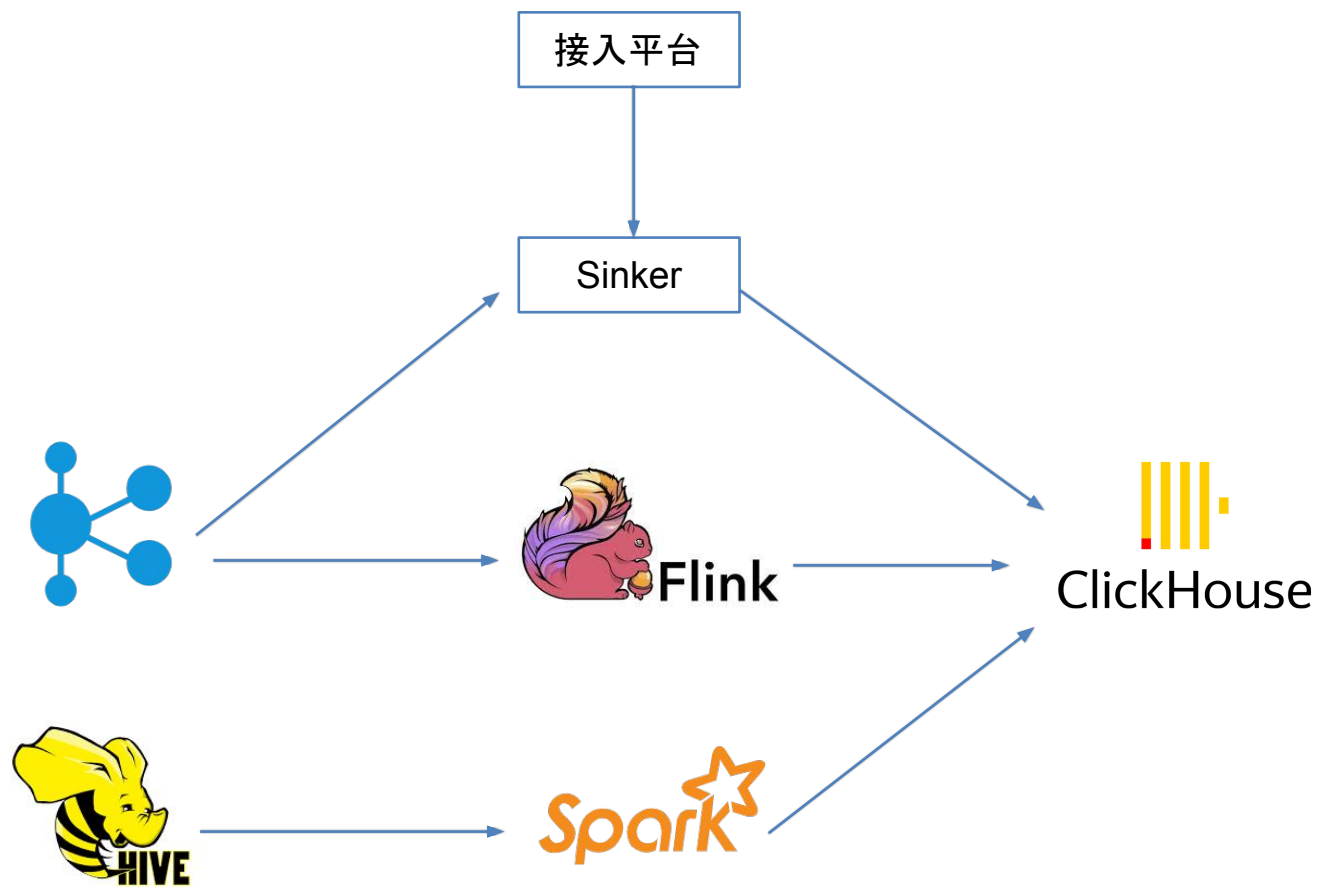
# ClickHouse接入平台

## 离线导入

- 用户需要将hive里的结果导入到ClickHouse中
- 提供基础Spark任务, 用户只需指定表及分区
- 控制向ClickHouse写入速度
- 覆盖升级、Failover场景
- 端到端Exactly once

# ClickHouse接入平台

全貌





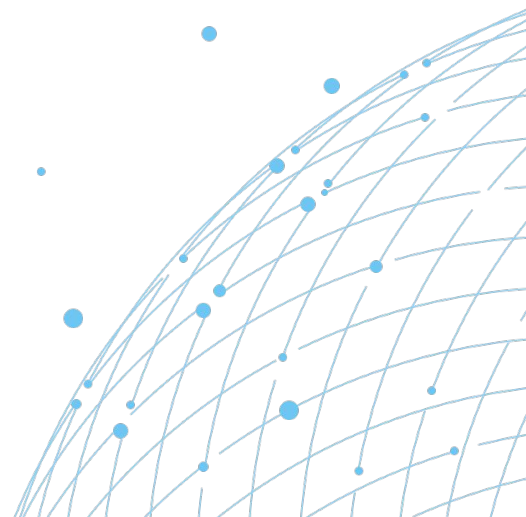
# 03

核心改造

# 核心改造

## 启动加速

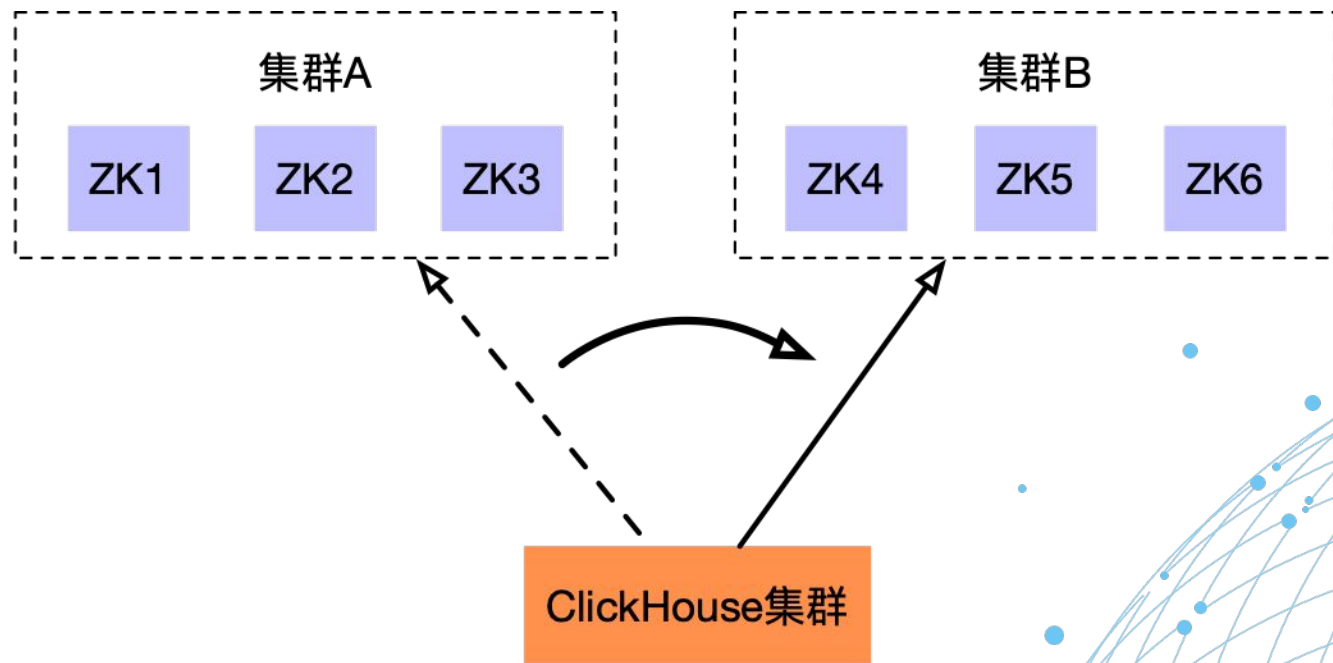
- 启动时要Load 70W+个文件
- 耗时可达30分钟
- 将columns count ttl checksum等元信息存入rocksdb
- 启动时从rocksdb加载
- 启动时间缩短为3分钟



# 核心改造

## ZooKeeper动态加载

- ZK节点升级及搬迁需要
- 实现了多次ZK集群的无感迁移
- 代码贡献到社区



# 核心改造

支持实时Exactly Once写入

- Sinker只能保证At Least Once
- 部分关键业务没有主键, 无法去重





# 核心改造

基于Flink提供Exactly Once写入

- 提供write commit接口
- 利用flink的checkpoint机制, 实现2PC
- Write时先写临时part
- Commit时更新为正式part



# 核心改造

基于Flink提供Exactly Once写入

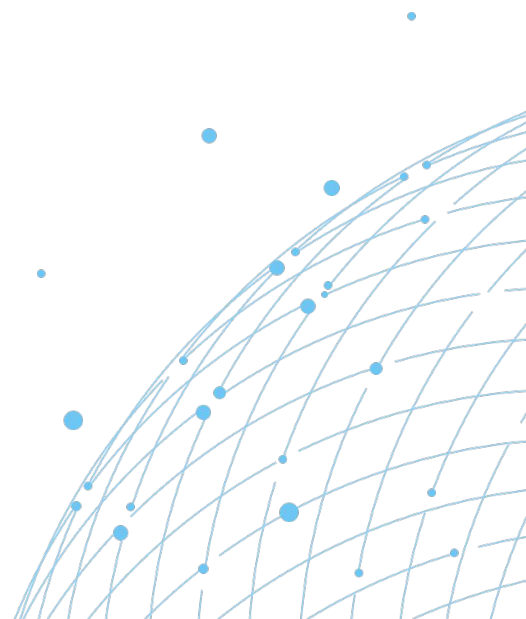
- 提供write commit接口
- 利用flink的checkpoint机制, 实现2PC
- Write时先写临时part
- Commit时更新为正式part



# 核心改造

## 自研管理平台

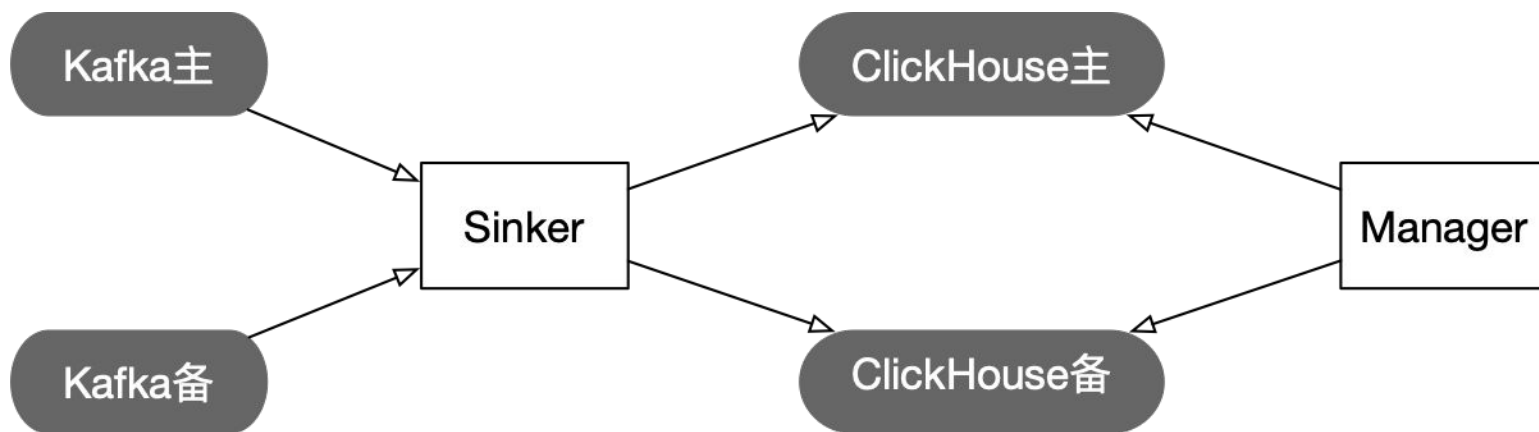
- ClickHouse手动挡操作
- 机器、磁盘上下线频繁
- 完全可视化
  - 集群扩缩容
  - 机器、磁盘上下线
  - 用户接入
  - 表的管理、变更
  - 集群使用情况分析, 用户治理
  - 关键监控、告警



# 核心改造

关键业务高可用

- 部分在线服务需要极高SLA保证
- 支持同一个表写入两个集群中
- 查询时自动切换



The background is a solid blue color. On the left side, there is a faint, abstract pattern consisting of a grid of thin, light blue lines. Overlaid on this grid are numerous small, light blue dots. Some of these dots are connected by thin, light blue lines, creating a sense of movement or data flow. The overall effect is a modern, technological aesthetic.

# 04

## 业务场景

# 业务场景

## ETL

- 面向逻辑相对简单的ETL场景
- 降低用户多个系统切换的成本

任务参数

<> 格式化

✓ 保存

```
1 WITH
2   protoData_uri AS uri,
3   protoData_countDistA AS countA,
4   protoData_countDistB AS countB,
5   protoData_countDistC AS countC,
6   protoData_countDistD AS countD,
7   (countA + countB + countC + countD) as rescount
8 SELECT
9   day,
10  toStartOfHour(reportTime) AS reportTime,
11  uri,
12  toString(uid) as uid64,
13  sum(rescount) AS res_count
14 FROM #{source}
15 GROUP BY
```



# 业务场景

## 实时监控

- 与监控平台打通

\* 英文名称

\* 集群

\* 指标推往

描述

```
1 select
2     ,
3     ,
4     sum(1) / 300 as value
5 from
6
7 where
8     day = today()
9     and toStartOfMinute(now()) - 300 <= toStartOfMinute(rtime)
10 group by
11     ,
12     ,
```

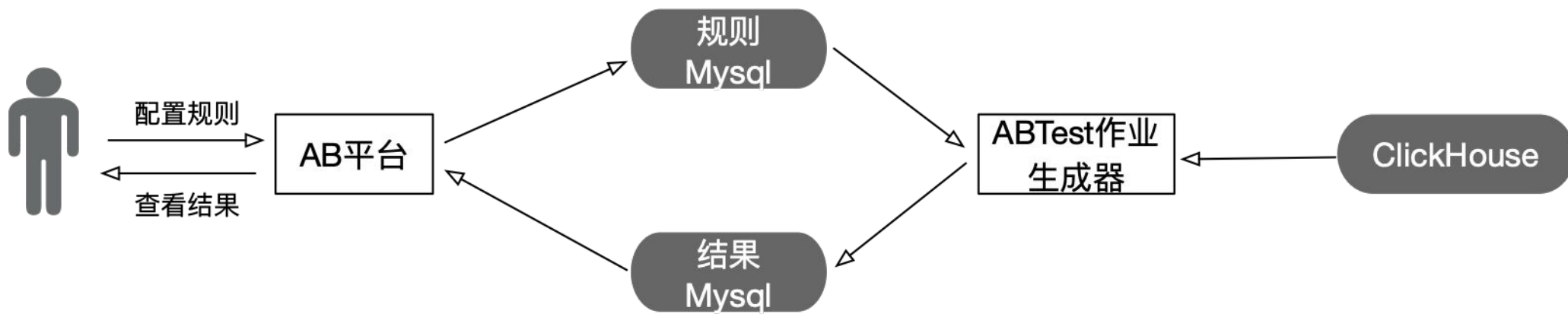
<> 格式化

▶ 运行

修改

# 业务场景

## ABTest





# 业务场景

## 自助分析

事件

A

+筛选条件 复制 X

APP启动(启动成功)

总人数

+ 新增事件

用户属性

1 用户组1

+筛选条件

满足条件

请选择筛选条件

请选择计算方式

请输入筛选值

+ 添加对比用户

统计维度 按 请选择维度

分组查看

事件分析图表

时间 过去

14

天



5.00

4.00

3.00

2.00

1.00



ClickHouse



# 业务场景

## MySQL导入

- 实时消费bin log

字段	类型	是否主键	备注
a	int	是	
b	int	是	
c	int	是	
d	string	否	
e	string	否	
__version	bigint	否	__version作为ReplicatedReplacingMergeTree 的ver参数，保证对于某个唯一主键组合(a, b, c), 在ck中最终只有一条数据存在，且该条数据对应着最新的版本
__deleted	tinyint	否	__deleted表示本条数据是否被删除
update_time	DateTime	否	binlog数据插入ck的当前时间

# 业务场景

## MySQL导入

- 相同主键保证到同一个server里
- 自动生成查询使用的物化视图

```
CREATE VIEW target_view
as select
    a,
    b,
    c,
    argMax(e, __version) as e,
    argMax(f, __version) as f
    argMax(__deleted, __version) as __deleted
FROM table
GROUP BY a, b, c
order by a, c, c
having __delete = 0
```



# 05

未来展望

# 未来展望

- 高压下ClickHouse的稳定性
- 运维平台的完善
- 单集群扩展能力

•THANKS

