转转 | DataFun.

ClickHouse在自助分析场景中的探索及实践

王鹏哲 转转 大数据平台&实时计算架构师

让人们享受闲置交易的快乐，让世界因流转更可持续

PART ONE

转转自助分析场景下OLAP选型

OLAP选型背景

行为数据

查询扫描数据量大，精确去重/近似去重/分组计算量大。

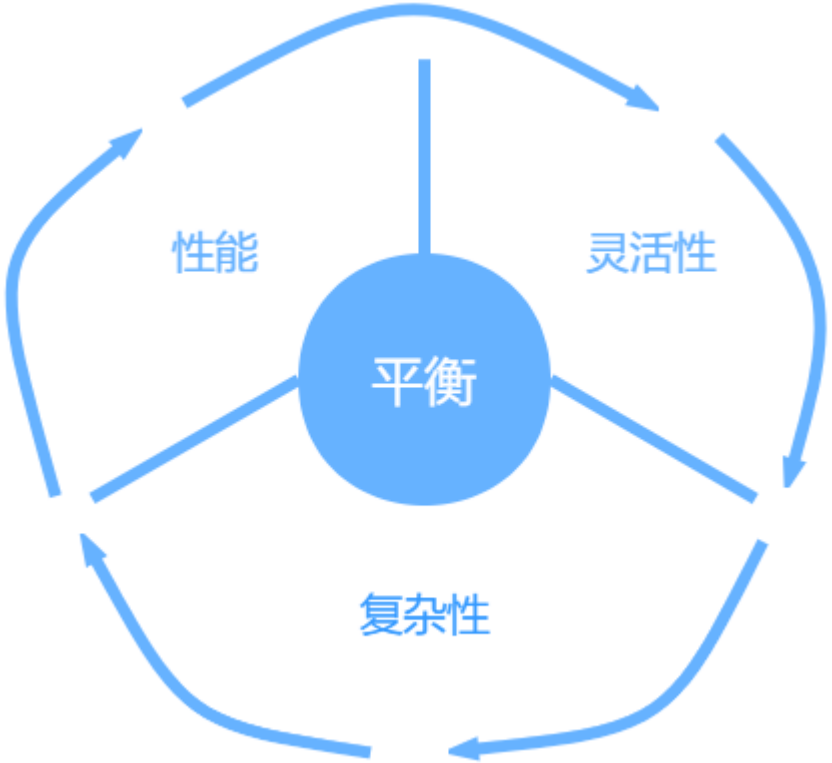
即席查询

传统离线数仓经过数仓分层和汇总层通用指标预计算，但是无法满足用户个性化报表需求。

稳定快速

保证高可用，支持任意指标、任意维度并秒级给出反馈。

OLAP选型考量



性能

数据量级：亿级/百亿级/千亿级
数据时效性：毫秒级/秒级/分钟级

灵活性

查询场景：聚合结果/明细数据
数据链路：离线/实时
查询支撑：高并发、即席查询

复杂性

引擎架构：门槛低、运维简单、扩展性强

OLAP引擎选型

OLAP引擎	技术	优点	缺点	自身存储
Kylin	MOLAP，完全预聚合立方体	1.亚秒级的查询速度，同时支持高并发 2.友好的web界面以管理，监控和使用立方体	1.维度较多时，交叉度太深会导致预计算结果爆炸式膨胀 2. 灵活性较弱，不支持ad-hoc查询	否
Druid	位图索引查找、字符串编码，预聚合技术	1.实时数据摄入 2.高可用、高性能、高并发	1.OLAP场景支持有限，JOIN不成熟 2.无法支持精确去重	是
Impala	MPP系统，SQL On Hadoop	1.计算基于内存，支持使用磁盘进行连接和聚合 2.支持窗口函数、UDF	1.对于内存依赖较大；完全依赖于Hive 2.每当新的文件被添加到HDFS，该表需要被刷新	否
Presto	MPP系统	1. 跨数据源的联邦查询 2. 支持多表 join，支持复杂查询	1. 多张大表关联操作容易OOM 2.并发能力不足	否

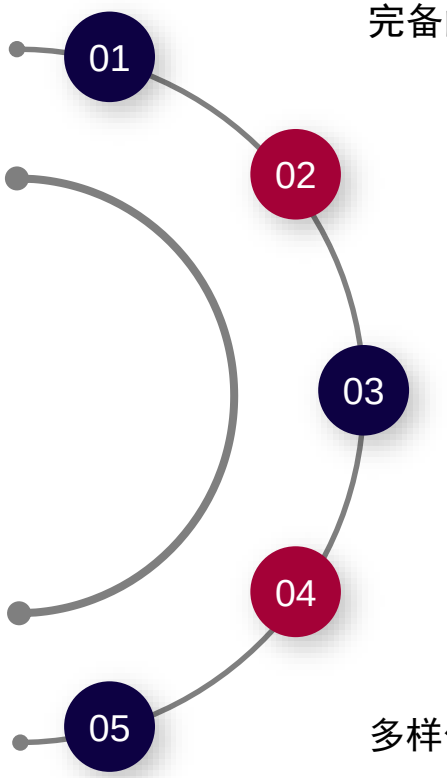
OLAP引擎选型

OLAP引擎	技术	优点	缺点	自身存储
ClickHouse	明细动态聚合查询，物化视图	1.单机性能彪悍 2.列存储、向量化引擎 3.可保留明细数据	1. 没有完整的事务支持 2.分布式表join能力较弱	是
Doris	MPP分布式架构	1.运维简单，支持在线扩缩容 2. 支持事务和幂等性导数，物化视图自动聚合，查询自动路由	1. 版本迭代更新较快，成熟度不足 2. 大规模数据的复杂ETL容易内存不足	是

ClickHouse是什么

ClickHouse是一个面向实时联机分析处理(OLAP)的基于列存储的开源分析引擎。Yandex（俄罗斯最大的搜索引擎）于2016年6月15日开源；开发语言为C++；是一款PB级的交互式分析引擎。

ClickHouse



完备的DBMS功能；较为完善的SQL支持。

列式存储和数据压缩；支持索引。

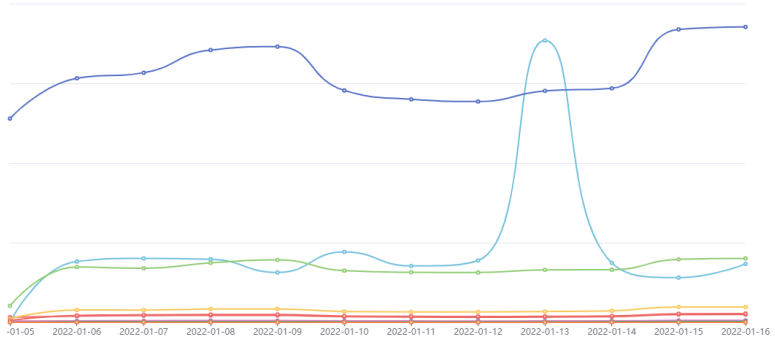
向量化引擎与SIMD提高了CPU利用率，多核多节点并行。亚秒级查询响应。

支持数据复制和数据完整性。

多样化的表引擎。

ClickHouse应用场景

交互式报表



AB TEST

01

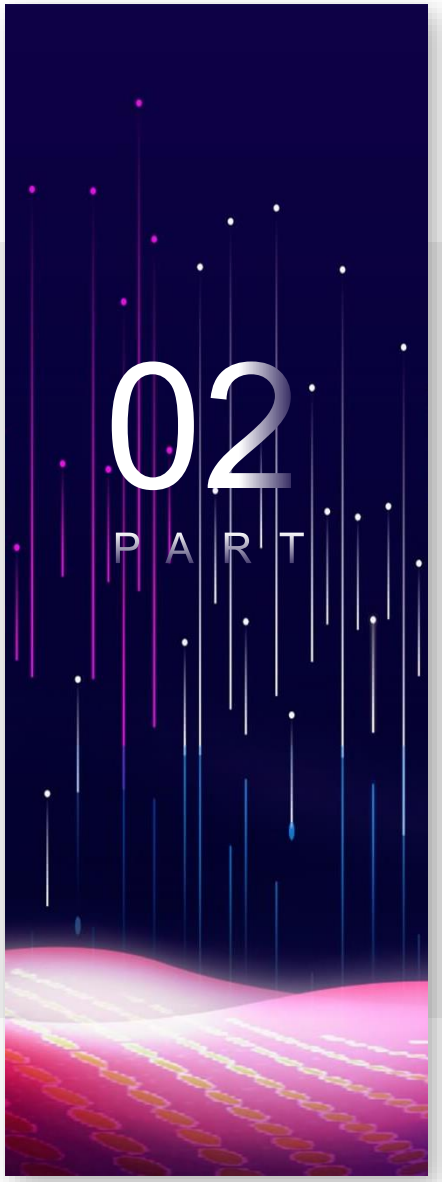
02

03

04

用户画像系统

监控系统



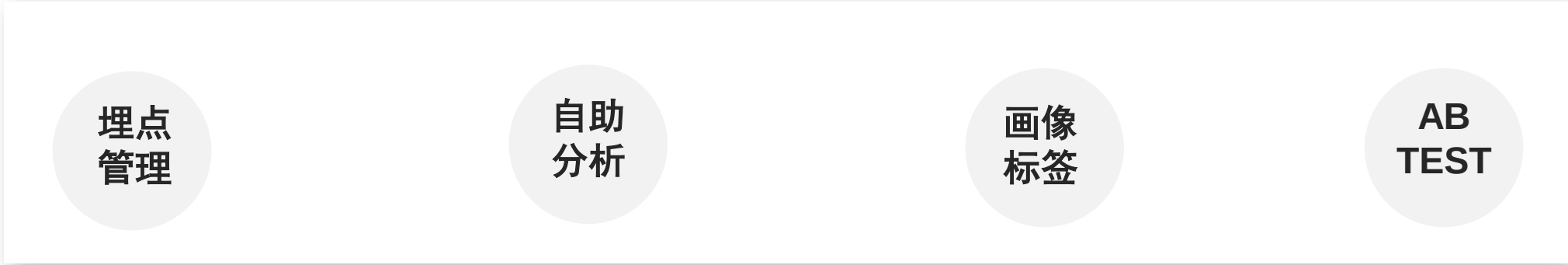
PART TWO

高斯平台自助分析场景

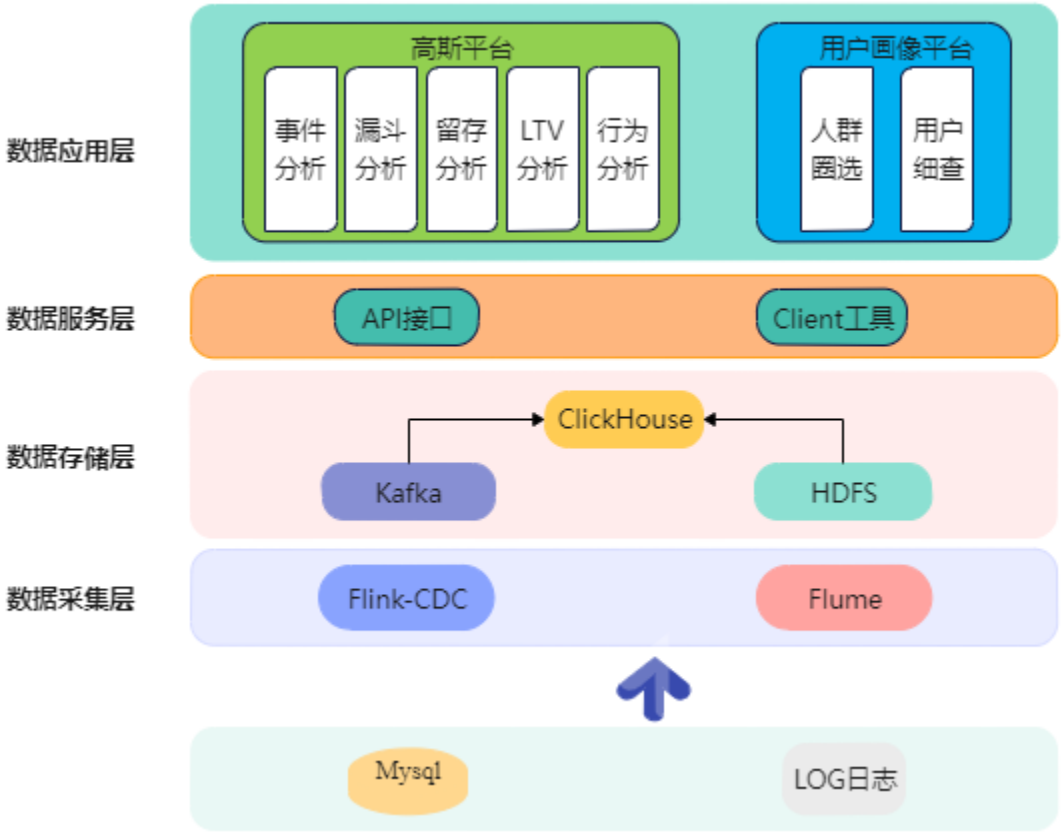
高斯平台

系统介绍

- 埋点数据管理：埋点元数据纳管，埋点质量统一监控。
- 自助分析：基于业务特点和多部门复合需求，提供多维度、多指标的交叉分析能力，全面支撑日常数据分析需求。



高斯平台-系统架构



数据清洗

- 离线数据：Hive ETL
- 实时数据：Flink + 维表关联

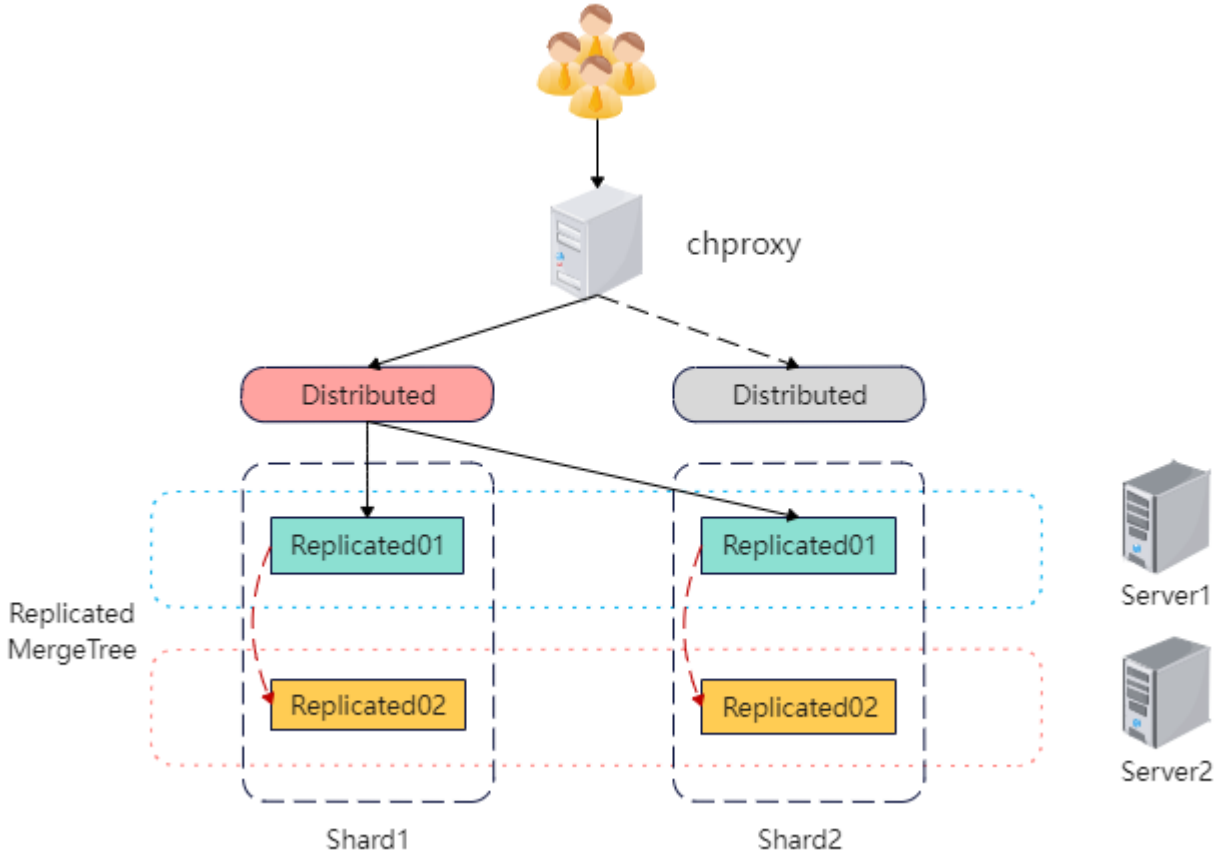
数据接入

- 离线数据：SeaTunnel + 调度平台
- 实时数据：Flink ClickHouseSink

数据服务

- 对外：统一封装服务，外部调用
- 对内：提供客户端工具

高斯平台-高可用架构



优势

由ReplicatedMergeTree表引擎管理数据副本（依赖Zookeeper）。

劣势

集群配置比较复杂，维护成本比较大。

高斯平台-集群现状

支撑能力

业务支撑：在线报表查询 + 数据分析业务

分析能力：报表型查询毫秒级响应；分析型查询大部分秒级响应

集群规模

服务器数量：20

副本设置：双副本

数据规模

存量数据：40T+

日增数据：20亿+

TTL设置：186天

平台统计：核心看板500+　日活跃用户 200+

业务场景-行为分析

业务背景

近期上线一个活动专题，产品/运营想查看该活动页面各坑位的点击效果。

技术实现

MATERIALIZED VIEW（POPULATE）+ SummingMergeTree + MergeTree

业务场景-AB TEST分析

业务背景

转转内部AB实验应用非常广泛，特别用来验证推荐算法和功能优化的效果。传统T+1的离线AB实验指标时效性无法满足业务需求，需要实时实验指标观察功能或者算法上线后的效果。

技术方案

实时写入-问题与挑战

问题：

Too many parts (311). Merges are processing significantly slower than inserts

原因：

频繁小批量插入，merge速度跟不上插入的速度

解决：

1、Client端调整：降低并发度，自定义Sink灵活性增加，可配置时间、批次大小、写入本地表

2、Server端参数调整：parts_to_throw_insert

业务场景-亿级数据JOIN

业务背景

需要用户曝光/点击情况，来JOIN用户画像表，用以匹配圈选人群和用户特征行为的关系。

技术方案

联接字段分桶JOIN

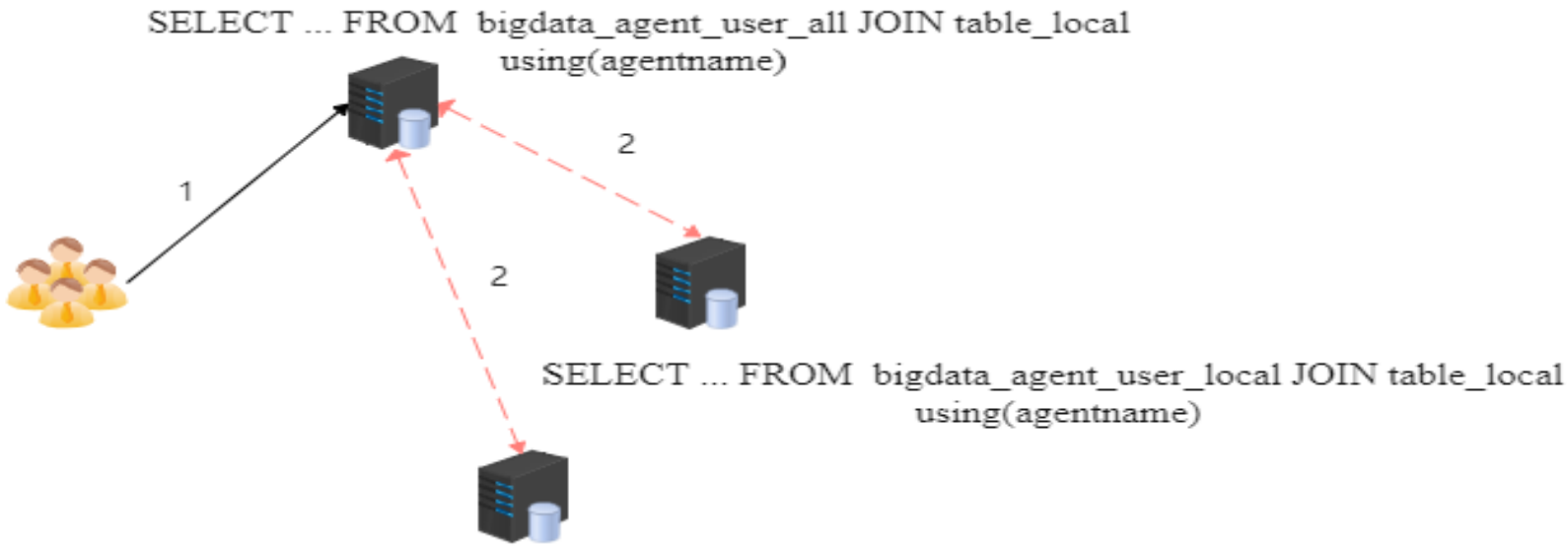
- 本地表写入：数据导入按照关联字段（用户ID）进行分桶写入shard分片
- 分布式表写入：

Distributed(cluster_name, database_name, table_name[, sharding_key])

亿级数据JOIN-底层原理

技术原理

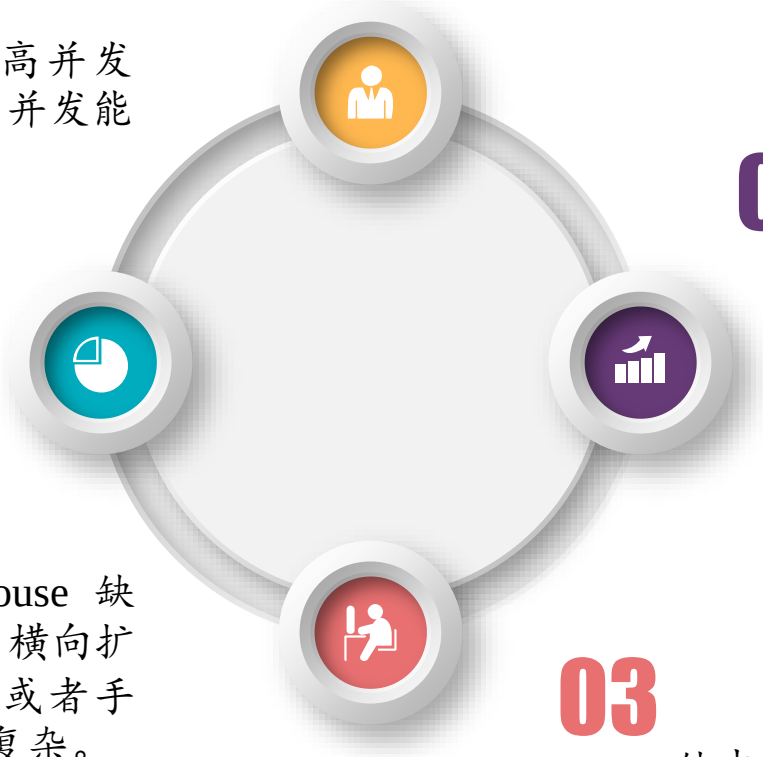
由于数据已经预分区了，相同的JOIN KEY对应的数据一定在一起，不会跨节点存在，所以无需对右表做分布式查询，也能获得正确结果。



高斯平台–当前痛点

01

部分业务场景需要高并发查询，ClickHouse高并发能力弱，性能下降。



02

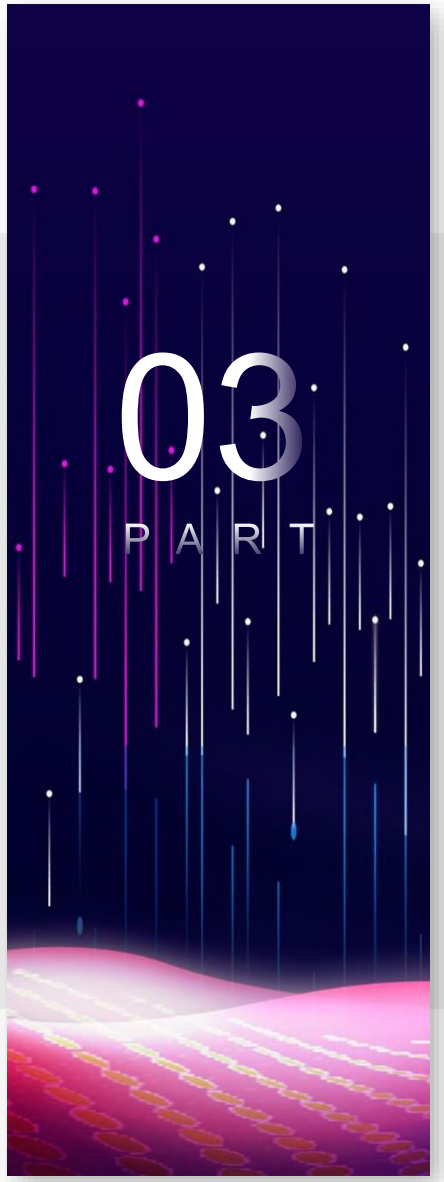
不支持事务性的 DDL 与 DML 操作，而且多副本模式的元数据管理强依赖于 Zookeeper。

03

缺少完整的UPDATE、DELETE操作

04

集群需要扩容时，ClickHouse 缺少自动的 rebalance 机制，横向扩容时需要借助第三方工具或者手动 rebalance,扩容缩容运维复杂。



PART THREE

ClickHouse优化实践

问题实践 - 内存相关

问题报错

Memory limit (total) exceeded: would use 169.14 GiB (attempt to allocate chunk of 4312306 bytes), maximum: 169.14 GiB: While executing AggregatingTransform.

分析及解决

配置参数(max_memory_usage)限制SQL的查询内存使用的上限，当内存使用量大于该值的时候，查询被强制KILL。

- 配置max_bytes_before_external_group_by参数，当使用内存到达该阈值，进行磁盘group by；
- 配置max_bytes_before_external_sort参数，当使用内存到达该阈值，进行磁盘order by；
- count distinct内存不够，推荐使用一些预估函数，这样不仅可以减少内存的使用同时还会提升查询速度；

问题实践 - Zookeeper相关

问题报错

Code: 999. DB::Exception: Received from :9000. DB::Exception: Cannot allocate block number in ZooKeeper: Coordination::Exception: Connection loss.

Code: 225, DB::Exception: Received from :9000,DB::Exception: ZooKeeper session has been expired.

分析及解决

与Zookeeper的连接丢失导致不能分配块号等问题；Zookeeper会话超时。

- 调整MaxSessionTimeout参数，加大Zookeeper会话最大超时时间
- 在Zookeeper中将dataLogDir、dataDir目录分离，可使用SSD存储
- ClickHouse建表的时候添加use_minimalistic_part_header_in_zookeeper参数，对元数据进行压缩存储

性能调优-基础参数

配置参数	参数含义	建议值
max_concurrent_queries	最大并发处理的请求数	默认值100，建议150-300
max_memory_usage	单个查询的最大RAM使用量	建议总内存的80%
max_memory_usage_for_all_queries	单服务器所有查询使用的最大内存	建议总内存的80-90%
max_memory_usage_for_user	用户查询的最大RAM量	
max_bytes_before_external_group_by	group by使用内存超出阈值后会刷新到磁盘进行	max_memory_usage/2
max_bytes_before_external_sort	order by使用内存超出阈值后会溢出磁盘进行排序	
background_pool_size	后台线程池的大小	默认值16，建议32

性能调优-建表/查询规范

数据类型

- 建表时能用数值型或日期时间型表示的字段就不用String；数值类型group by最快
- 不建议使用Nullable，Nullable列无法被索引

查询调优

- 列裁剪和分区裁剪；尽量避免在大数据集上使用虚拟列；
- 数据字典：将一些经常需要关联分析的业务创建成字典表进行join操作，前提是字典表不宜太大，因为字典表会常驻内存
- 物化视图：对于一些确定的数据模型，可将统计指标通过物化视图的方式进行构建，这样可避免数据查询时重复计算的过程
- 使用Global：在分布表in或者join时使用，避免出现查询指数级放大

PART FOUR

ClickHouse未来在转转的规划与展望

未来规划及展望

服务平台化、故障规范化

业务方易用性、多租户隔离、限流熔断、监控报警，业务治理。

ClickHouse容器化部署

存算分离；集群横向扩展、数据均衡。

服务架构智能化

根据业务场景，自适应选择ClickHouse、DorisDB引擎。

ClickHouse内核级优化

实时写入一致性保证；分布式事务支持；移除Zookeeper服务依赖。


转转
DataFun.
THANK YOU