京东ClickHouse的实践之路

吴建超

数据与智能部

2021/12

目录

CONTENTS

01
发展历程

介绍京东OLAP的发展历程、组件选型的考虑、OLAP的现状。

京东OLAP发展历程概览

2018前 MySQL

早期数据量小，需求少，数据由外部系统计算好结果，然后同步到MySQL中。

2018 Kylin

在初期需求较少，主要是维度比较固定的报表场景
Kylin比较好的支持满足了当时的需求，规模50台。

2019 Doris

Doris较好的解决了Kylin的痛点

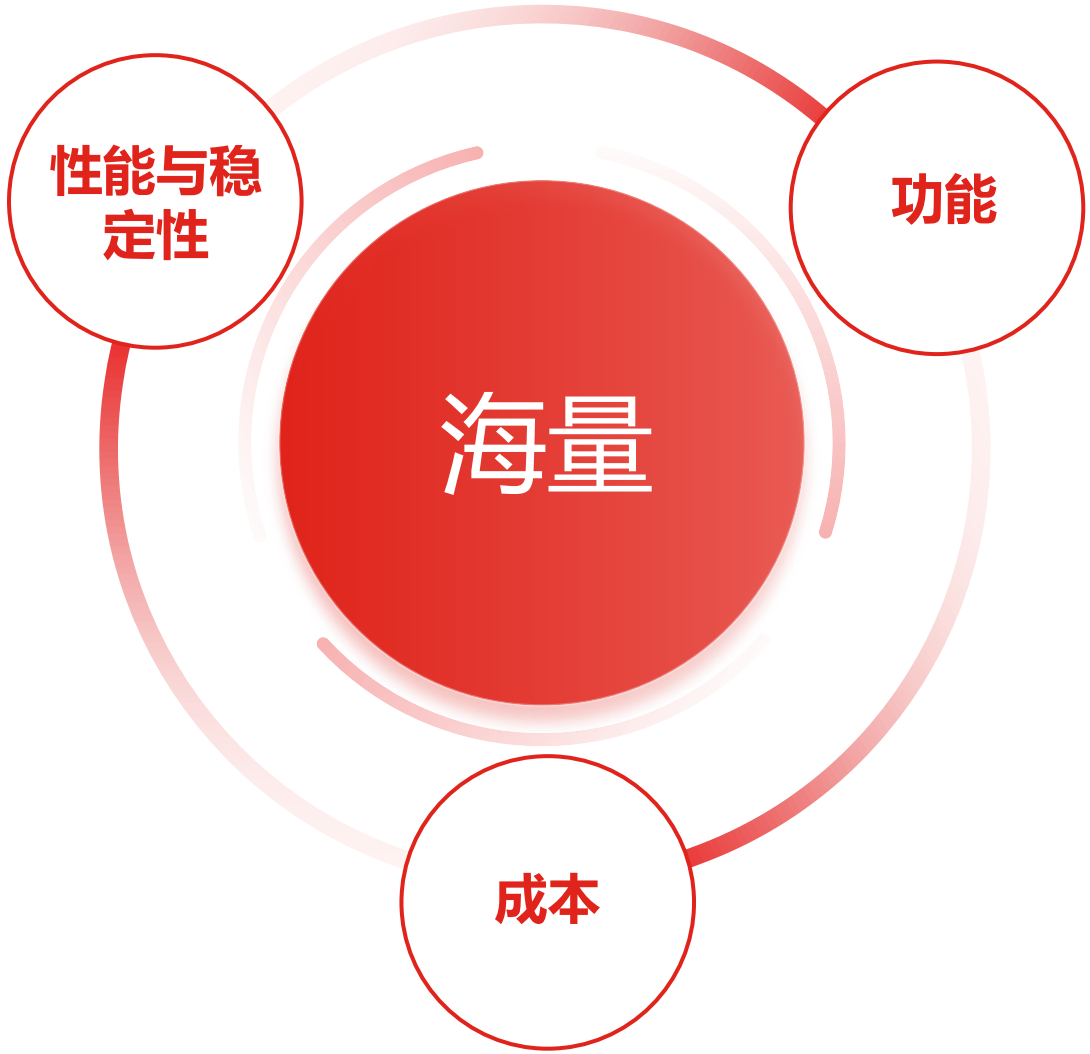
- 不支持实时的场景
- 不支持对明细数据的查询
- 查询不够灵活
- 无法增删维度信息
- Cube机制带来的存储爆炸问题

集群规模到现在也发展到400台

2020 ClickHouse

20年下半年和21年是京东OLAP爆发增长的时期，集群规模从几十台服务器，增长到3000+。随着业务的增长Doris在性能和稳定性上碰到一些问题。最终选择了ClickHouse，实践证明在稳定性和性能方面ClickHouse表现更佳。

多维分析组件选型考察方面



- **海量数据**
  - 海量数据处理能力是选型的基础
  - 每天千亿数据的导入能力和上亿数据的快速查询能力
- **功能多样性**
  - 即支持离线数据导入方式又支持实时导入
  - 即支持明细计算又能支持预计算
  - 即支持固定的分析场景又能支持Ad-hoc场景
- **使用成本**
  - 用户接入成本，比如：SQL标准、学习成本
  - 日常运维成本，比如：周边工具、日常管理
- **性能与稳定性**
  - 复杂场景仍能保持稳定
  - 优秀的查询性能

ClickHouse为主，Doris为辅

ClickHoue更优的方面

1. 性能更佳，导入性能和单表查询性能更好
2. 稳定性更好，在复杂场景下系统更稳定，更少发生故障
3. 功能丰富，非常多的表引擎，更多类型和函数支持，更好的聚合函数以及庞大的优化参数选项
4. 集群管理工具更多，更好多租户和配额管理，灵活的集群管理，方便的集群间迁移工具

Doris更优的方面

1. 使用更简单，如建表更简单，SQL标准支持更好
2. 运维更简单，如灵活的扩缩容能力，故障节点自动恢复
3. 分布式更强，支持事务和幂等性导数，中心化的元数据管理

为什么选择双引擎?

1. 京东的OLAP场景复杂、数据规模大，ClickHouse灵活的使用方式、良好的稳定性和性能提供了在复杂场景下的基本保障
2. 同时历史用户熟悉Doris不愿意迁移到复杂的ClickHouse上去，所以保留Doris，但更推荐用户使用ClickHouse
3. 保留Doris可以保持对Doris社区新动态的跟进，这样我们可以在Doris和ClickHouse双方互相借鉴一些功能。

京东OLAP现状

产品现状

服务器：**3000+**

日导入：**2.6万亿**

日查询：**2.8亿**

数据量：**10PB级**

2年时间几乎从零开始到现在各个方面逐渐走向正轨，期间经历了4次大促的考验，大促零事故。
目前经历了密集的业务接入时期，正向运维更细致和研发更深入的方向发展。

02
CH实践

介绍ClickHouse在京东的平台化、容器化、集群高可用、集群扩缩容等方面情况。

平台化建设

- 集群部署
- 节点下线
- 替换节点
- 配额管理
- 故障自愈
- 监控告警
- 业务申请

- 实现了大部分操作一键化和少量故障处理自动化
- 运维人效从人均几十台机器，增加了10倍

+ 申请项目

项目ID	项目名称	业务部门	服务类型	集群类型	集群名称	状态	业务级别	双流方案	创建时间
470	[redacted]	京东集团-京...	clickhouse	独占集群	LF2_CK_Pub_55	运行中	0级	否	2021-12-01 11:32:39
469	[redacted]	京东集团-京...	clickhouse	共享集群	LF0_CK_Pub_54	运行中	2级	否	2021-12-01 10:45:45
468	[redacted]	京东集团-京...	clickhouse	业务测试	lf6ckcnts05	运行中	3级	否	2021-11-30 11:32:15
467	[redacted]	京东集团-京...	clickhouse	业务测试	lf6ckcnts05	运行中	2级	否	2021-11-29 17:14:27
466	[redacted]	京东集团-京...	clickhouse	业务测试	lf6ckcnts05	运行中	2级	否	2021-11-29 10:11:10
465	[redacted]	京东集团-京...	clickhouse	业务测试	lf6ckcnts05	运行中	3级	否	2021-11-26 12:57:55

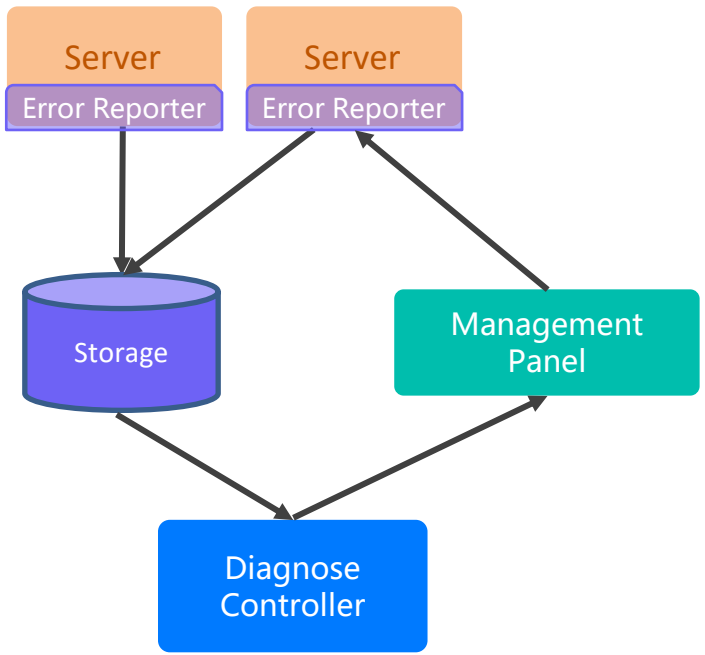
模板修改

项目Id	项目名称	业务部门	业务级别	用户名	最大连接数	最大查询数	超时时间(S)
247	[redacted]	京[redacted]...	0级	ge_user_rw	1	1	1
258	[redacted]	京东[redacted]	1级	mktdata_order	20	100	20
258	[redacted]	京[redacted]...	1级	mktdata_order_r	20	6	20
259	[redacted]	京东集[redacted]...	1级	mktdata_user	20	100	20

平台化建设 – 故障自愈

故障自愈

思路是实现一个自愈框架，可以支持用户自定义异常探测、故障处理Handler以及降级功能。管理工具的自愈模块主要包含：异常发现、异常诊断、系统自愈、报告输出和人工干预几个步骤。



磁盘自愈流程

1. 服务器上的异常检测工具发现掉盘异常，并上报到异常存储中
2. 自愈Controller通过后台任务发现了掉盘异常
3. 自愈Controller再次确认是否掉盘
4. Controller通知管理工具执行掉盘处理Handler（重新mount）
5. 如果执行成功发出报告，并结束流程
6. 如果执行失败则触发系统降级流程（节点下线）
7. 管理工具执行节点下线
8. 如果不能下线则发出告警，通知人工干预

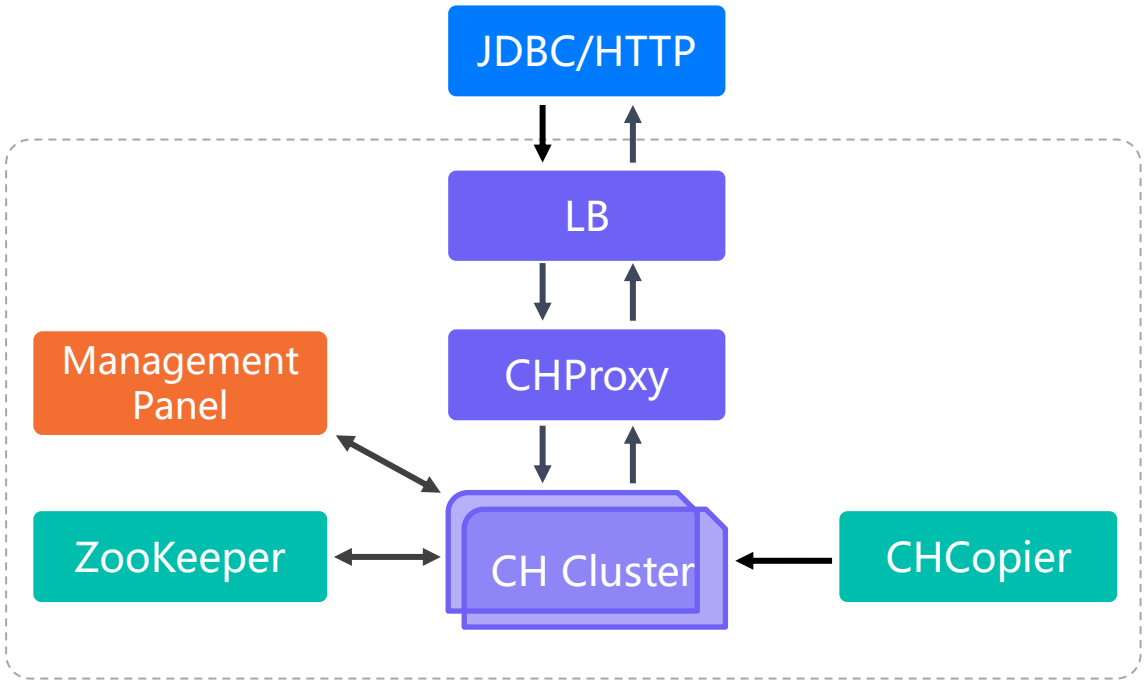
平台化建设 – 配额管理

- 每年两次大促是京东特色场景也是每年最的重要的事件，大促时重点业务需要重点保障，非重点业务需要降级。

- **解决方案：**基于配额设置日常模板和大促模板

- **配额管理依赖ClickHouse提供的丰富的资源隔离的手段：**
  - Concurrent queries：并行请求个数
  - Queries：10s 内请求次数
  - Execution_time ： 查询超时时间

集群高可用 - 单集群架构



各个组件说明：

- LB：分发请求到后台CHProxy节点
- CHProxy：将请求二次分发，达到更均衡的目的
- ZooKeeper：协调者、数据插入BinLog、元数据存储
- CH Cluster：ClickHouse集群，多分片多副本
- CHCopier：负责集群间的数据迁移
- 管理工具：集群部署、扩容、节点下线和替换等

集群高可用 - 双活集群方案

- **要求**

  0级业务必须做双活集群

  集群故障要能1分钟内恢复

- **写入**

  离线：同时向两个集群写数据，写完后校验数据是否一致

  实时：源头上主备机房两条流，每条流写入对应的CH集群

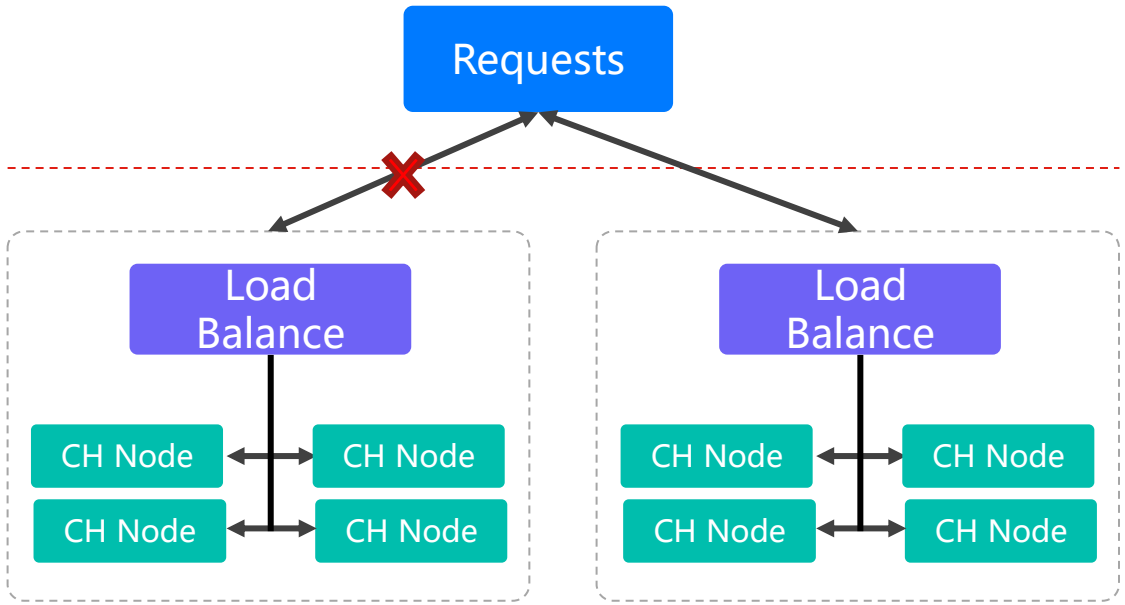
- **查询**

  既可以冷备的方式使用集群，也可以以双活的方式使用（大促场景）

- **存在的问题**

  故障处理完全交由业务处理，是否也可以放在OLAP侧

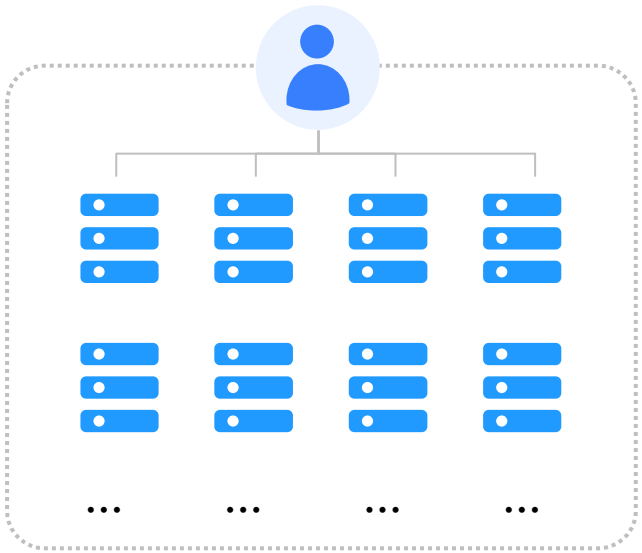
  双倍的资源是否存在资源浪费，备集群是否可以资源降级



容器化建设

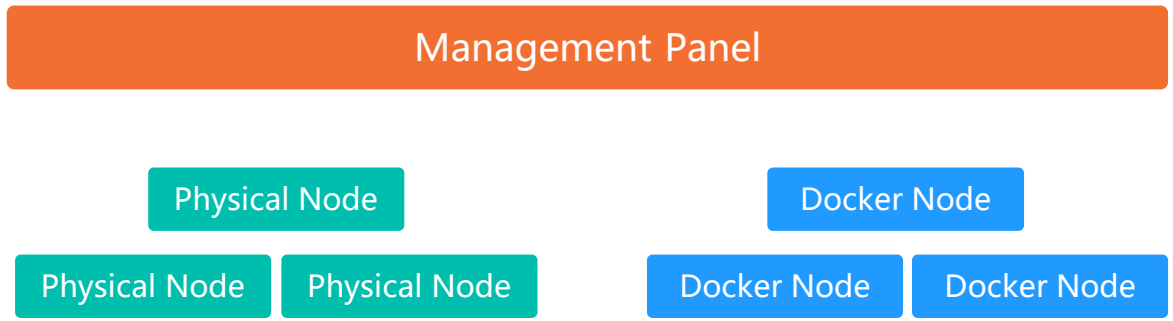
期望收益

- 提高资源使用率
- 进一步提高运维人效
- 为后续做云原生ClickHouse做准备



思路与方法

- 容器化的集群需要兼容管理工具



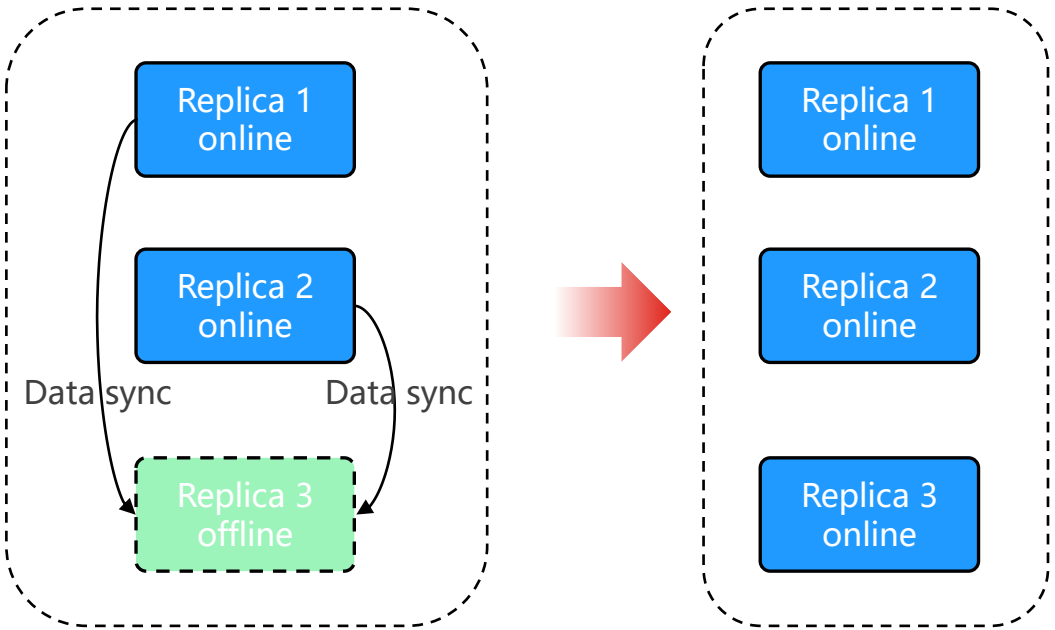
- 基于K8s订做了Controller实现有状态的服务的调度和管理

1. 存储：数据本地盘存储，节点漂移后由上层服务同步数据
2. 模板：支持上层服务自定义模板，包括：分片副本、数据存储目录、日志目录、端口、集群拓扑等信息
3. 亲和性：同一个CH的不同实例不能调度到同一个物理机

扩缩容实践

扩缩副本

- 适用于仅提高/减小集群查询能力的场景
- 缺点是增加或减小的资源必须是单副本资源的倍数并且不能扩存储的容量



扩容副本流程

1. 部署扩容节点，注意此时节点不能上线
2. 获取分布式表对应的shard信息
3. 同步shard里面已存在replica的元数据信息到扩容节点
4. 更新所有节点的metrika信息
5. 检测扩容节点元数据信息是否和已存在的replica一致

注意事项

1. 同步元数据信息完成前，不能同步metrika信息，以免DDDL失败
2. 节点分批扩容（1-5个），以免影响线上请求
3. 系统表也需要同步，比如：quotas、users等

扩缩容实践

扩缩分片

- 既调整查询吞吐量又可以调整集群存储容量
- 痛点是：如何进行数据均衡?

扩容数据均衡

方案1：数据不做均衡

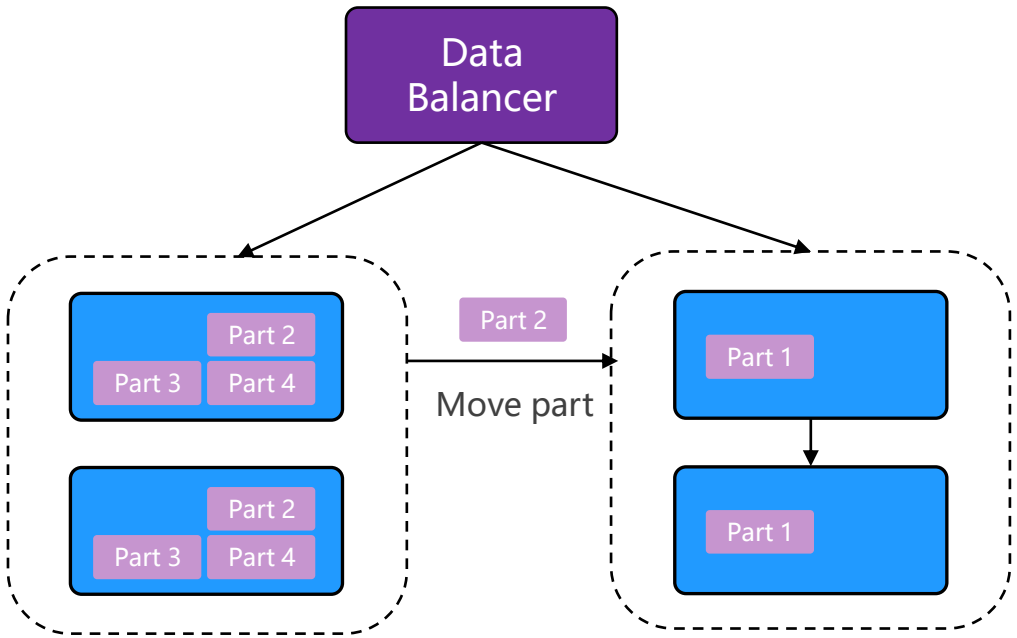
靠数据不断更新达到数据均衡的目的，只适用于数据存储周期短的实时场景，比如：数据存储3天

方案2：创建新表，然后通过Insert into with select的方式

缺点是需要足够的存储，并且对系统负载影响大，需要额外处理增量数据。

Move Part方式

1. 通过Balancer可以定制算法控制数据迁移的速度
2. 相对select insert的方式，通过移动数据文件的方式效率大大提高
3. 不能处理数据通过哈希散列的表



社区回馈

- **对待社区的思路**

1. 紧跟社区，半年升级一次版本
2. 除非大的feature，否则使用社区版本
3. 鼓励社区回馈

- **目前京东对社区贡献有几十个PR，部分列举如下：**

1. #28981 新的查询配额：Query_selects、Query_inserts
2. #19603 Add keeper 4lw commands
3. #27481 sparkbar aggregate function
4. #28325 对string类型支持位运算操作
5. #30325 扩展Zookeeper负载均衡策略

03
典型案例

介绍ClickHouse在京东的典型案例，
主要包含数据更新、大写入量等场景。

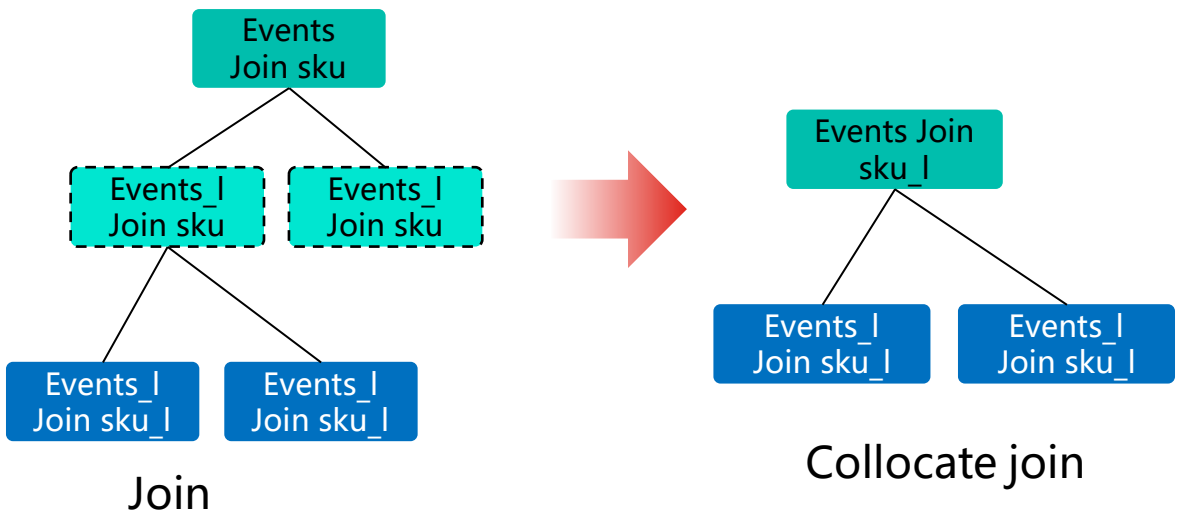
典型场景 – 黄金眼刷岗

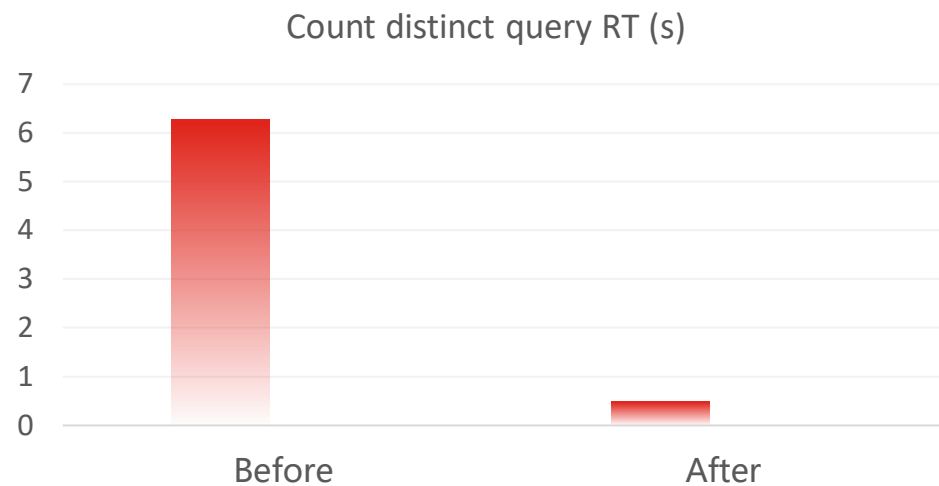
场景描述：

商品信息中包含像部门、类目、采购员等信息，这些信息随着组织架构的调整经常需要变动。商品是京东重要的分析场景，对响应时间要求较高。

解决方案：本地Join

商品信息表和事件流表都按照SKU ID将数据分散到不同的分片，查询通过分布式表（事件流表）Join 本地表（商品表）的方式进行



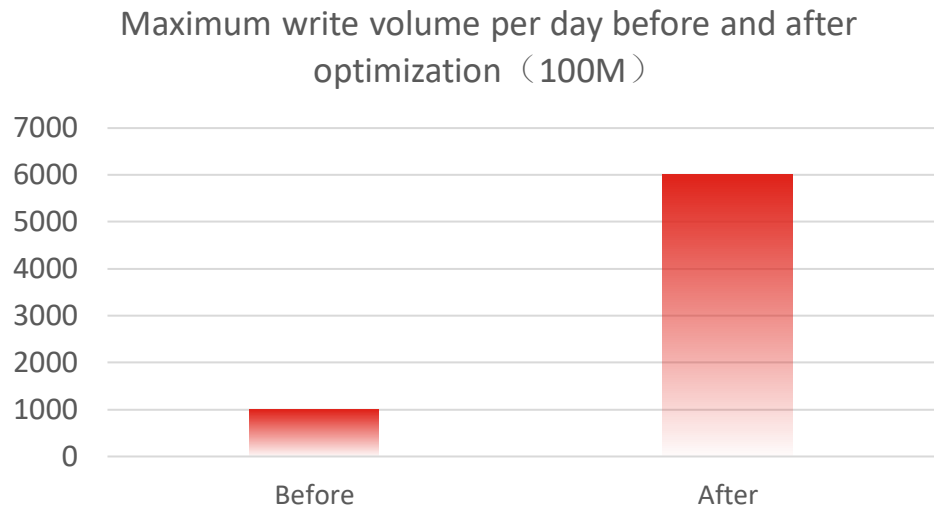


典型场景 – 京东云日志

场景描述：

京东云日志存储，数据实时写入，日常数据大概4000亿条/天，大促高达6000亿条/天

1. CHProxy作流量二次均衡
2. 插入本地表
3. 插入批次大小/时间间隔：10w/30s
4. max_connections：1024
5. background_pool_size：64
6. 多磁盘非Raid的存储策略



04
未来规划

介绍ClickHouse在京东的未来规划。

未来规划

思路：

- 提高人效，解放人力，投入研发
- 由浅入深逐步打造技术影响力

短期：

- 进一步完善集群在线扩缩容功能
- 完善容器化集群建设，并与管理工具进行更深度的整合
- 管理工具提供更多功能，比如：在线修改配置、大查询查杀等

长期：

- 云原生ClickHouse，实现弹性扩缩容
- ClickHouse深入优化，包含Join优化、SQL优化等

Thanks!

wujianchao5@jd.com

技术与数据中心-数据与智能部

数据与智能部

京东