

# Clickhouse数据迁移踩坑和最佳实践分享

雁门@DTS

# ClickHouse数据迁移的需求

- 实时性
- 性能
- 数据的清洗和转换能力
- 易用性
- 功能丰富
- 可观测性

# | ClickHouse数据迁移的几种主流方式

- 实时：
  - MaterializeMySQL引擎
  - Canal->Kafka->ClickHouse
  - Flink
  - DTS
  - ...
- 存量：
  - clickhouse-client
  - 外表方式
  - Spark
  - Dataworks
  - ...

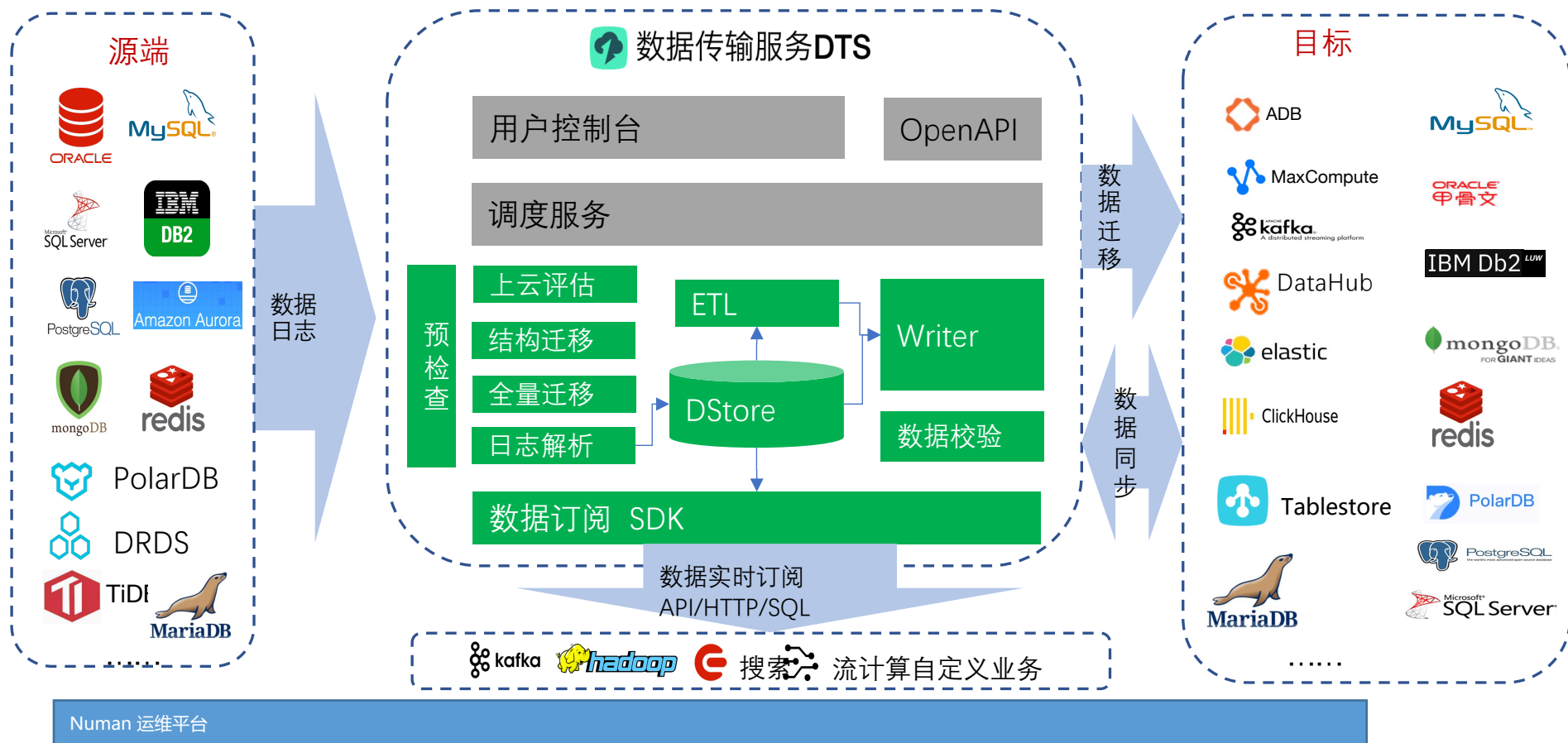
# | MaterializedMySQL的限制

- 阿里云ClickHouse有社区兼容版和企业版，ClickHouse企业版不支持MaterializeMySQL引擎，社区兼容版集群仅20.8及以上版本支持MaterializeMySQL引擎。
- 同步过程无法可视化，同步任务无状态显示，无法直接看到同步性能与延迟。
- MaterializeMySQL引擎不支持无主键表迁移。
- 数据源RDS MySQL集群和目标ClickHouse集群必须属于同一个VPC网络。
- 不支持ETL，无法做库表列映射，也不支持附加列。
- 不支持数据过滤，无法选择同步哪些dml或者ddl，无法通过filter来选择只同步一个表中的部分数据。
- 同步任务无法暂停，只能删除重配，删除任务后目标库同步的库表也会被删除。无法进行断点续传。
- 不支持某些DDL同步，例如create table，修改表的comment, 修改列的comment

# DTS MySQL->ClickHouse同步的优势

- 目标端支持阿里云ClickHouse社区兼容版和企业版。
- 可在阿里云DTS控制台上看到同步过程，比如现在同步任务正处于结构迁移，还是全量迁移，还是进入了增量迁移。可看到同步的RPS，BPS。可看到同步任务是否失败或者有延迟。
- 支持无主键表迁移。
- 由于阿里云DTS是第三方迁移工具，数据源RDS MySQL集群和目标ClickHouse集群可以属于不同的VPC网络。
- 支持ETL，库表列映射，附加列等功能。
- 支持数据过滤，可以只同步某个表指定条件的数据。也可以选择只同步某些DML或者DDL。
- 可以随时暂停、重启同步任务。支持断点续传。删除任务后不会删除目标端库表。
- 支持常见DDL同步，例如create table, alter table, truncate table, drop table等等。
- 高性能

# DTS系统架构及典型场景



- 阿里经济体数据库全球化和异地多活底座
- 9年打磨，支撑双11大促
- DTS支持多达30多种数据源和目标，服务近5万企业客户，已完成近100万个数据库上云

# | DTS同步MySQL数据到ClickHouse最佳实践---账号权限

数据库	库表结构同步	全量同步	增量同步
MySQL	SELECT权限	SELECT权限	待同步对象的SELECT权限 REPLICATION CLIENT、REPLICATION SLAVE、 SHOW VIEW 建库建表的权限，以允许DTS创建库dts，用于记录同步期间的心跳数据
ClickHouse	读写权限		

# | DTS同步MySQL数据到ClickHouse最佳实践---任务创建和配置

## DTS任务创建方式:

- 控制台:
  - DTS控制台
  - ClickHouse控制台
- OpenAPI

## 分片键选择:

- 没有指定分片键, 使用rand()作为分片键。
- 指定了分片键
  - 分片键有多列, 使用sipHash64(A, B, C)作为分片键。
  - 分片键为单列, 且为int, 则就使用该列作为分片键。
  - 分片键为单列, 且不为int, 则使用sipHash64(A)作为分片键。

## 分区键选择:

- 没有指定分区键, 那么结构迁移建表时不会添加partition by xxx, 也即不进行分区
- 分区键, 则使用用户指定的列作为分区键
  - 分区键源端类型为BIGINT, 则建表时添加partition by intDiv(XXX, 18014398509481984)
  - 分区键源端类型为TinyInt, SmallInt, MediumInt, 则建表时添加partition by intDiv(XXX, 4194304)
  - 时间类型(date, datetime和timestamp), 则建表时添加partition by toYYYYMM(update\_time)
  - 其他类型, 则建表时不添加partition by 语句, 也即不进行分区。



# DTS同步MySQL数据到ClickHouse最佳实践---结构迁移

## 建表方式：

- ClickHouse社区版，需要建一个本地表和一个分布式表。分布式表表名与目标端表名一致，本地表表名为分布式表表名+\_local
- ClickHouse企业版只需建一个目标端表名相同的表

## 附加列：

### 23.8版本之前：

列名	数据类型	默认值	说明
_sign	Int8	1	记录DML操作的类型： •Insert: 值为1 •Delete: 值为-1 •Update: 值为1
_version	UInt64	1	数据写入ClickHouse的时间戳

### 23.8版本之后：

列名	数据类型	默认值	说明
is_deleted	Int8	1	记录是否被删除： •Insert: 值为0 •Delete: 值为1 •Update: 值为0
_version	UInt64	1	数据写入ClickHouse的时间戳

## 引擎选择：

CK类型	集群类型	源端是否有主键	目标端CK引擎
社区版	单副本	有主键	ReplacingMergeTree
		无主键	MergeTree
	多副本	有主键	ReplicatedReplacingMergeTree
		无主键	ReplicatedMergeTree
企业版		有主键	ReplacingMergeTree（CK会自动加上Sharding+XXX）
		无主键	MergeTree（CK会自动加上Sharding+XXX）

# DTS同步数据到ClickHouse---全量性能

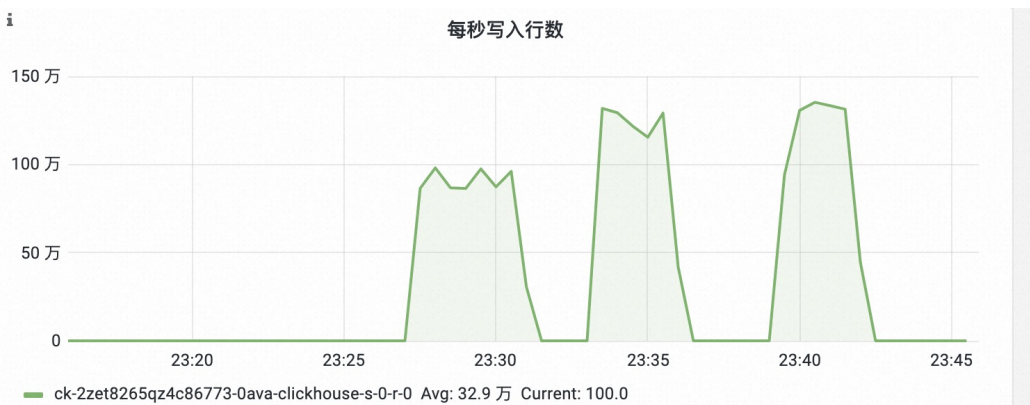
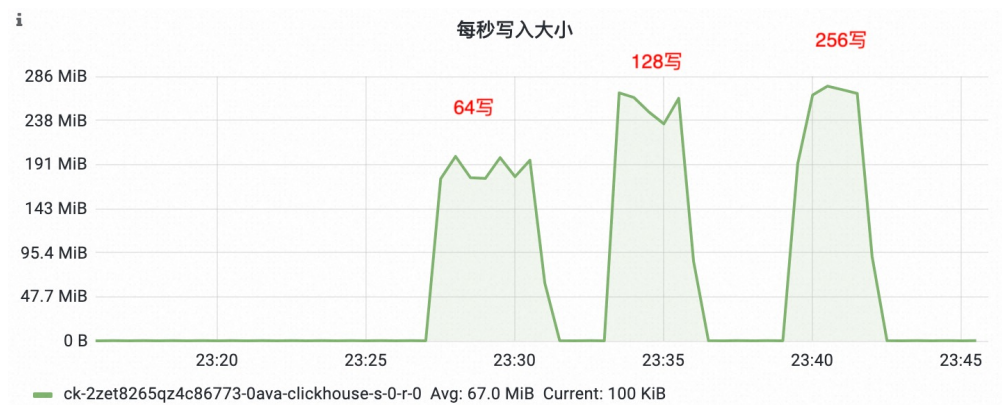
全量开箱性能：

100MBPS、50W RPS



全量优化性能：

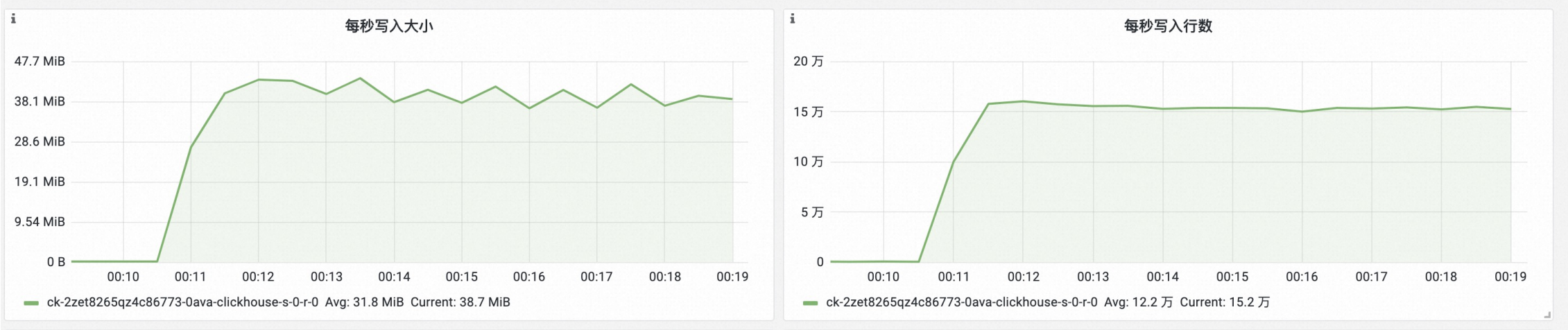
275MBPS、137W RPS



# DTS同步数据到ClickHouse---增量性能

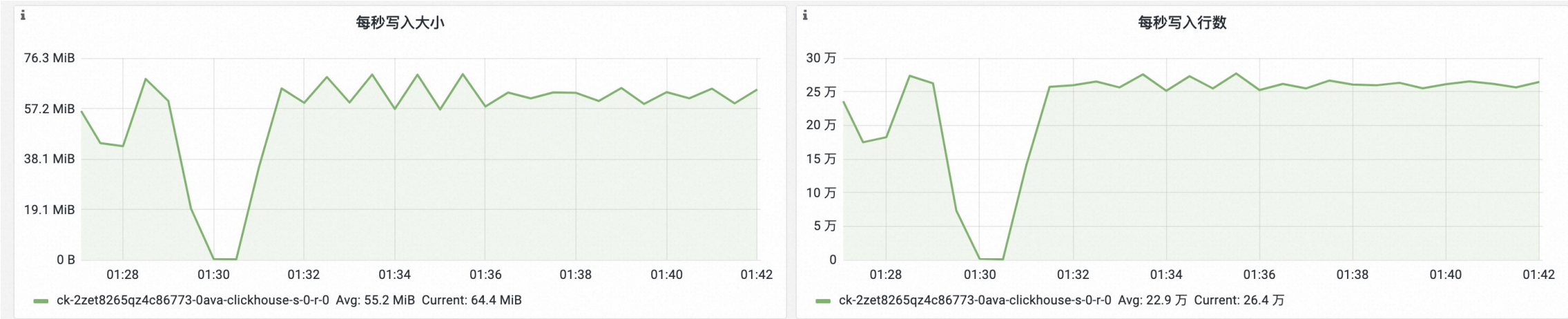
增量开箱性能:

38MBPS、15W RPS



增量优化性能:

50MBPS, 20WRPS



# | DTS同步数据到ClickHouse---常见问题1

## 数据为什么有重复：

ClickHouse本身没有主键约束，DTS在如下两个场景可能会引入重复的数据：

- 在全量迁移时，DTS会将一个表中的数据分成若干切片并发地拉取和写入。当DTS在迁移一个切片的数据时，如果用户暂停任务，然后重启任务。DTS会将当前切片的数据重新迁移到目标端。这样就会造成这个切片的一部分数据有重复。
- 在增量迁移时，由于Mutations查询是一种比较重的操作，DTS采用了和MaterializeMySQL引擎一样的方案，即基于ReplacingMergeTree引擎的版本控制的更新/删除方式。23.8之前版本更新数据时，会insert一条相同的数据,并将\_sign置为1。删除数据时，会insert 一条相同的数据，并将\_sign置为-1。ReplacingMergeTree在合并分区时会删除重复的数据，保留同一组重复数据中版本号\_version取值最大的行。

## •数据重复解决办法：

基于以上原理，我们可以使用如下方法来去除重复数据：

- 执行optimize table xxx来强制ClickHouse进行分区合并，以去除重复数据。
- 查询数据时加上final，如果希望过滤已经删除的数据，23.8之前版本还需要加上\_sign>0。例如: select \* from table\_name final where \_sign>0; 23.8之后版本不需要

# | DTS同步数据到ClickHouse---常见问题2

**datetime类型数据为何和源端不一致？：**

ClickHouse数据类型	最小值	最大值
DateTime64	1925-01-01 08:00:00	2283-11-12 07:59:59
DateTime	1970-01-01 08:00:00	2106-02-07 14:28:15
Date32	1925-01-01	2283-11-11
Date	1970-01-01	2149-06-06