

# 中通快递基于 SelectDB 的应用实践

童孝天 高级研发工程师

# 目录

01 背景介绍

02 应用实践

03 对比测试

04 未来展望

01

# 背景介绍

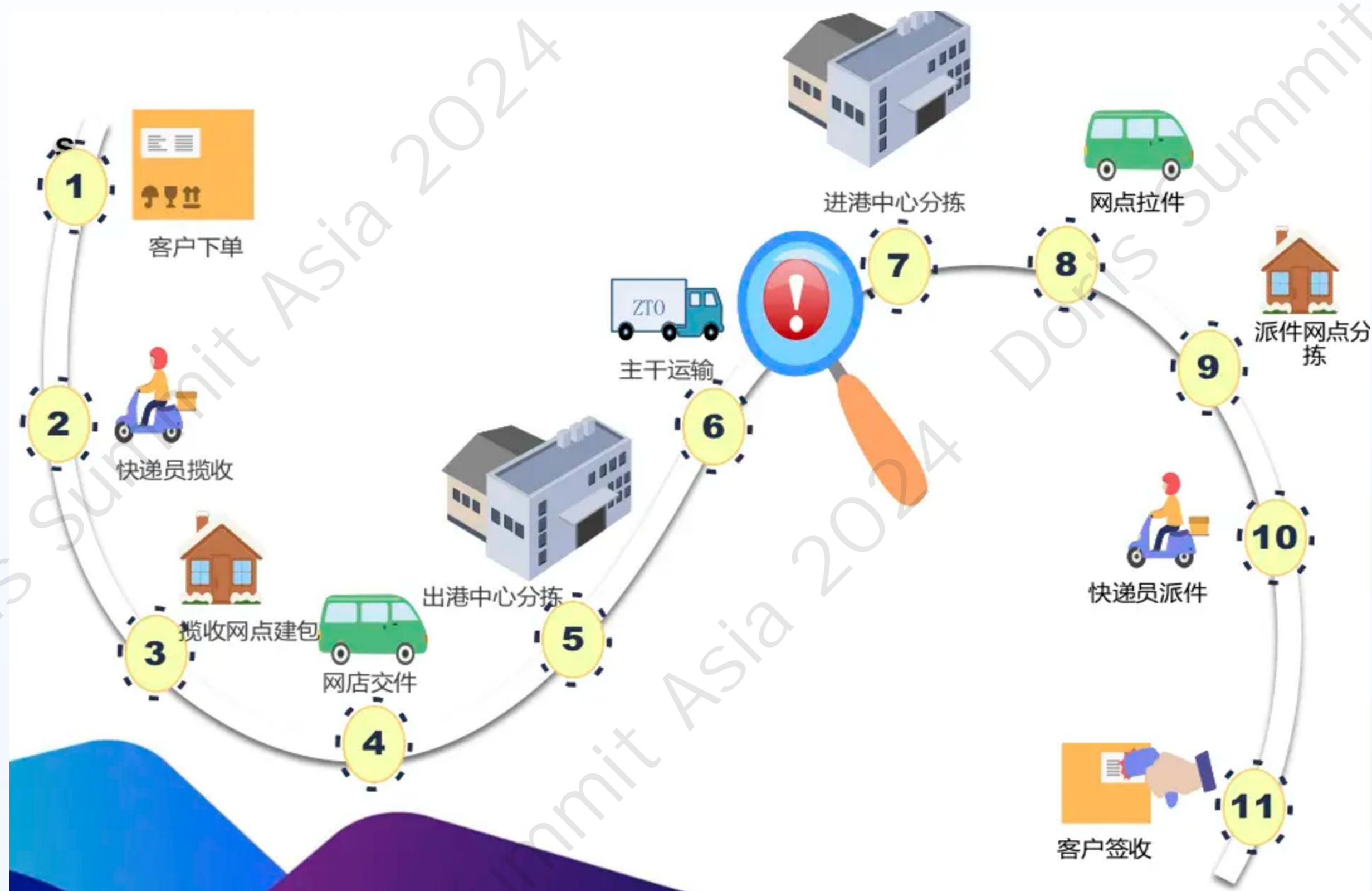


# 公司简介

## 简介

中通快递股份有限公司成立于2002年5月8日，总部位于上海，是一家集快递、物流及其他业务于一体的大型集团公司，中通快递的包裹量在2024年第三季度达到了87.2亿件，同比增长15.9%，市场份额为20.0%左右。

中通科技是中通快递旗下的互联网物流科技平台，拥有一支千余人规模的研发团队，秉承着“互联网 + 物流”的理念，与公司的战略、业务紧密的衔接，为中通生态圈的业务打造全场景全链路的数字化平台服务。



03

# 应用实践

# 选型背景

随着业务的不断发展，之前双十一的业务量到现在已成为每日的常态。为了满足各大业务场景对实时分析时效性的要求，同时保证数据快速写入和极速查询，需要一个合适的 OLAP 引擎补充原有的离线数仓架构体系，痛点具体如下：

## 数据时效不足

离线数据仓库使用离线抽取的方案，数据时效性为 T+1，而报表、数据大盘要求数据实时更新，当前架构无法满足

## 查询效率低

BI报表/离线分析需满足秒级别查询响应，离线数据仓库执行引擎主要是 Trino 及 SparkSQL，需读取和写入 HDFS 中的数据，执行时长一般为分钟级别，影响查询效率

## 维护成本高

整个技术栈涉及组件繁多，包括 Trino/HDFS/Yarn/HBase，之前线上也有一些实时场景使用 ClickHouse，但是维护较为复杂，且有些场景需求也无法满足



# 应用场景

借助 SelectDB 的高效的数据更新，低延时的实时写入以及优异的查询性能，我们在以下几个场景中进行应用实践：

## 复杂的实时需求

构建实时宽表，横跨快递的整个生命周期，需要汇总多条 kafka 数据流，进行数据整合，在通过 Flink 计算后，将数据写入 SelectDB 集群，并在此基础上进行准实时的数据分析

## 离线数据查询

对于 BI 报表/离线分析需满足秒级别查询响应，我们将离线数据计算后的数据定期导入实时数仓 SelectDB 表中，可以实现快查询，满足离线数据分析和决策的需求

## 简单的实时需求

对于一些较为简单的实时需求，利用 SelectDB 的部分列更新，以及其高效的查询能力，给 C 端或者实时大屏提供数据服务接口的能力

# 实时宽表

借助其他 OLAP 数据库，构建大宽表，基于宽表做分钟级的准实时分析：

6 亿+

日处理数据量

45 亿+

数据总量

200 列+

字段总量

**存在的问题：**总任务数 50+，时效在 5-10 分钟，随着任务数量的增加，同一时间点的任务数会增加，导致集群负载变大，影响任务的执行时间，时效难以保证



# 使用 SelectDB 带来的收益

## 更稳定

支持分区表，扫描的分区数据固定，相对稳定，原 OLAP 对全局索引不是很友好，每次都会扫描全表数据，对集群影响更大

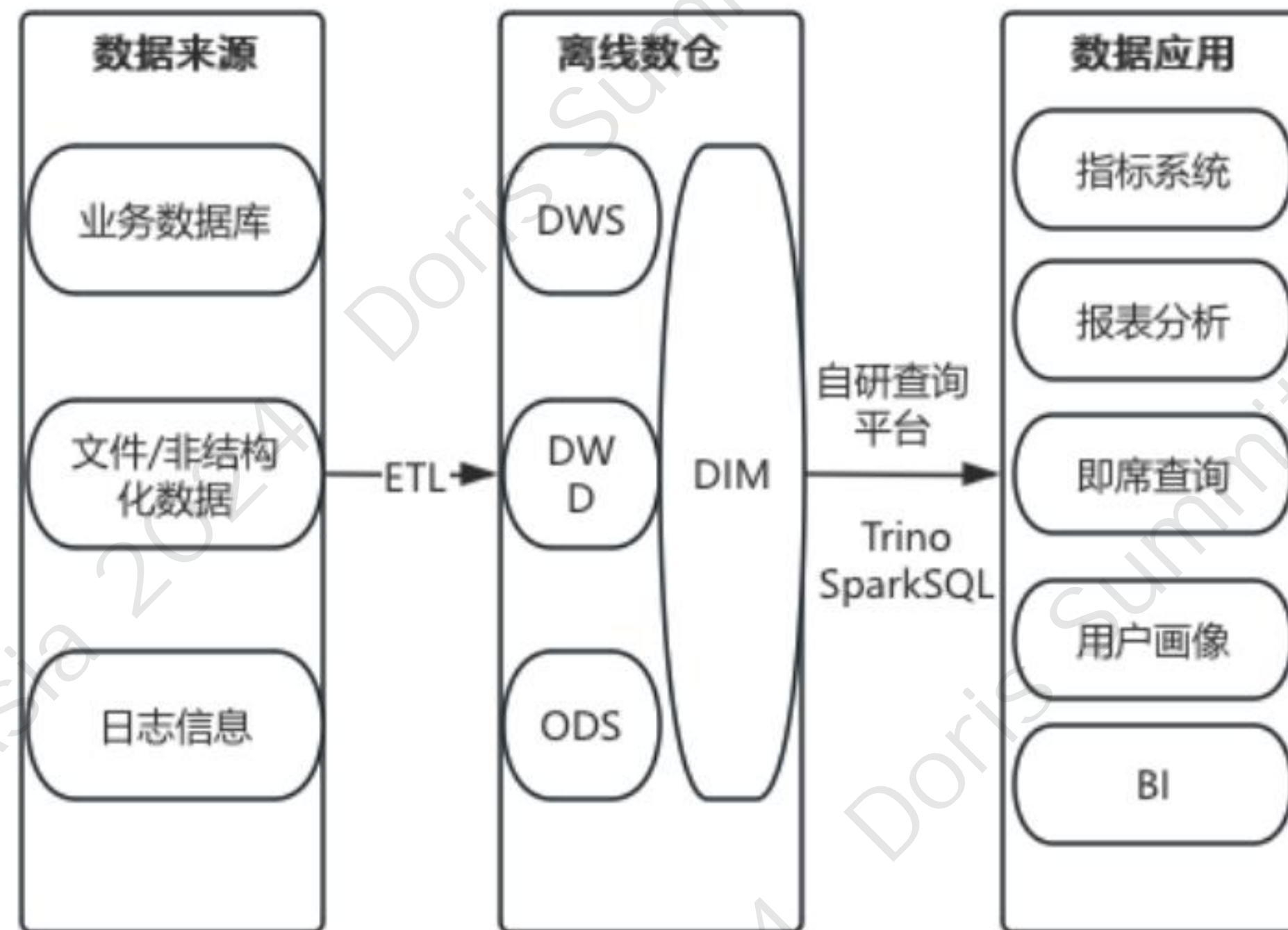
## 更高效

相比原 OLAP 数据库，部分 SQL 性能提升 5-10 倍，从原来 10 分钟左右到目前 90% 以上的分析能到 1 分钟内，部分可以秒级响应

## 更节约

资源仅使用原有机器的 1/3，便可以 cover 住原来的所有业务

# BI 报表/离线分析加速



## 查询不够稳定

在超大 Hadoop 集群下，namenode 的轻微抖动会严重影响一些短而小的 adhoc 查询和报表分析，导致查询不够稳定。

## 不支持高并发

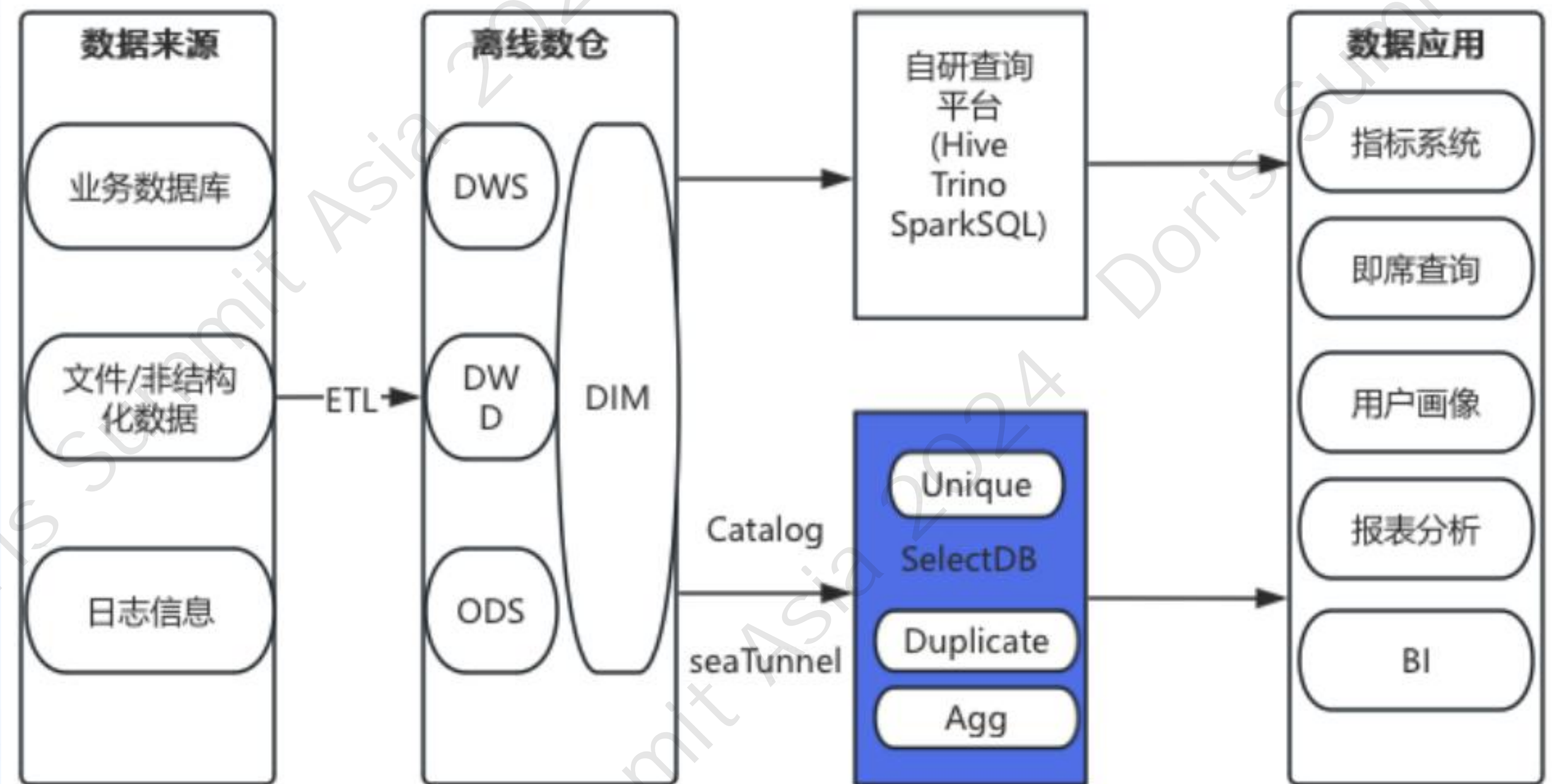
使用的 Trino 和 SparkSQL 在处理大规模数据集方面表现出色，但面对高并发查询时，处理效率与我们的预期存在较大差距。



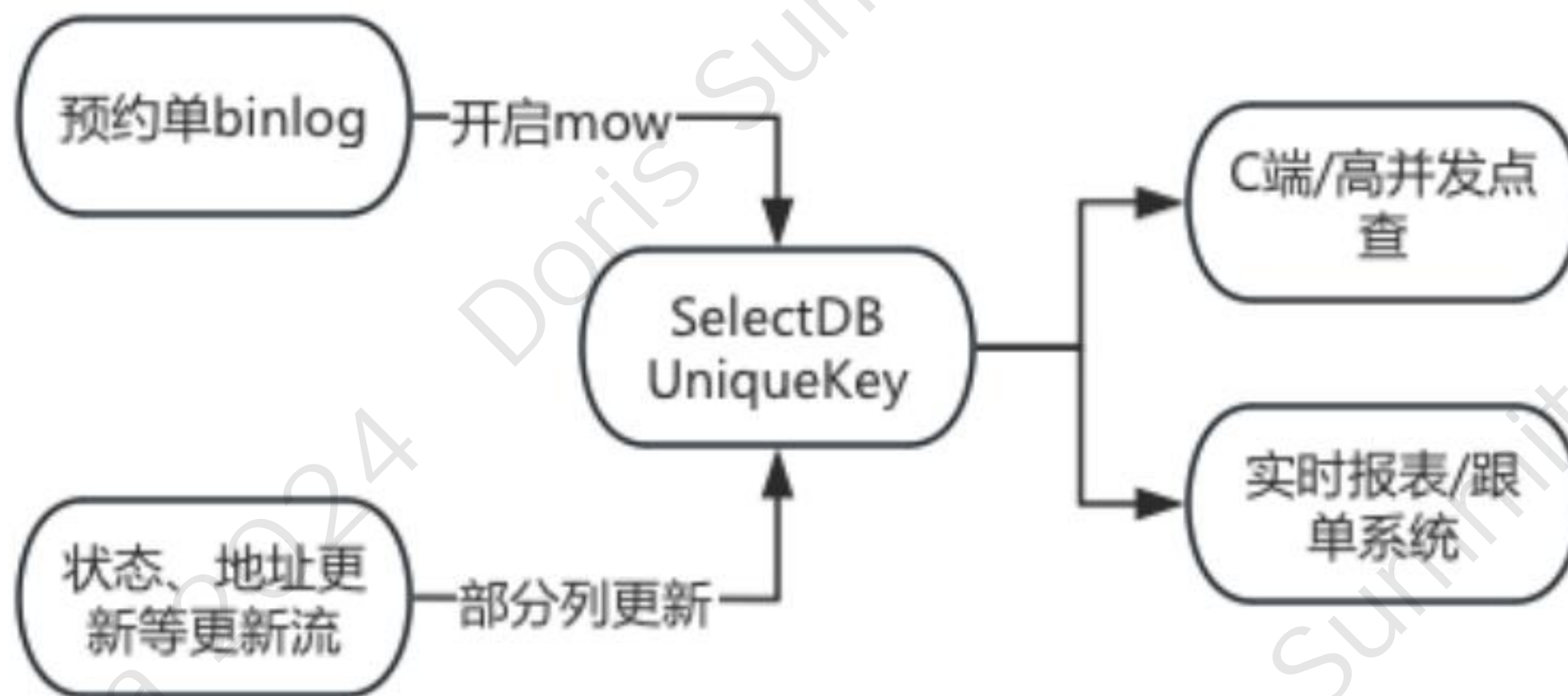
# 引入 SelectDB

## 极致的加速体验

- 在实时数仓建设阶段，我们把离线数据 DIM 维度层、应用层的数据通过 SeaTunnel 写入了 SelectDB 中，实现了结果表的查询加速，从而实现每秒上 2K+ 数量级的 QPS 并发查询，数据报表更新及时度大大提高
- SelectDB 提供了灵活丰富的 SQL 函数公式，并拥有高吞吐量的计算能力，数据分析师、产品经理等业务人员通过 Metabase（数据探索与可视化工具）+ SelectDB 即可基本满足 BI 的数据探索需求，大部分查询响应速度都在秒级完成



# 高并发与分析场景实践



对表结构的设计需要结合业务，因地制宜，合理规划 Key 和分区分筒列，一般将 where 条件或者 join 的字段定义成分桶较为合适

## 极致的时效性

作为数据服务提供数据大屏，对实时性要求高，借助倒排、BloomFilter 来支持多维分析，通过合理的分区分桶，在查询时过滤非必要的的数据，使数据扫描快速定位，加速查询响应时间

## 需要进行更新

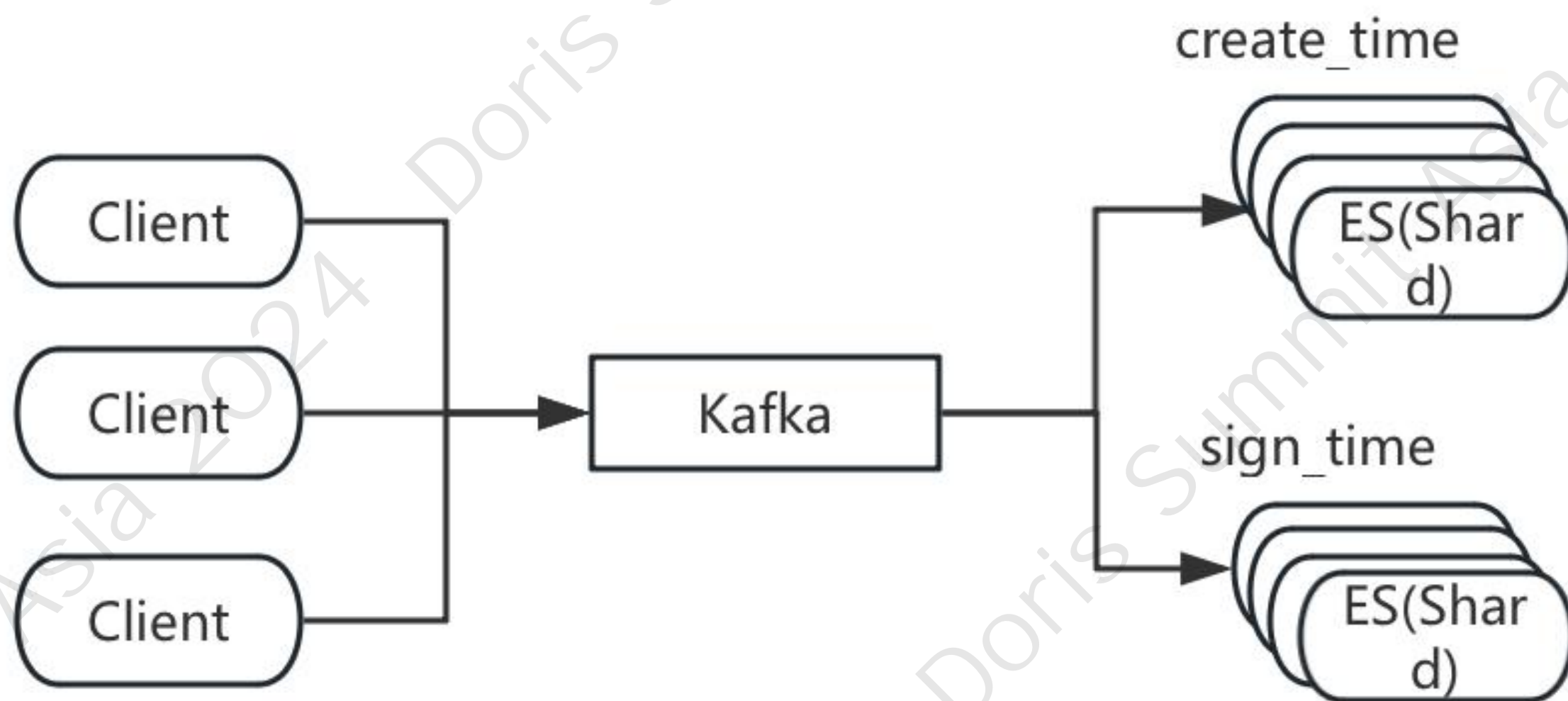
支持主键表（Unique Key）进行高效的数据更新，并对 Upsert、条件更新/条件删除、部分列更新、分区覆盖等各类更新提供了完备的支持，满足高效灵活的数据更新需求



03

# 对比测试

# ES 使用场景



## 写入吞吐大/高并发点查询

快递信息流转数据量巨大，对写入要求高，并且对订单 ID 等有高并发点查的需求。

## 多维查询

创建时间/签收时间 分片

为了加速查询性能，减少扫描数据，分别按照揽收时间/签收时间分片，冗余多份数据，加速查询

# ES 相关业务痛点

## 开发成本高

- 根据逻辑字段按照固定算法确定数据所属分片，指定分片查询；
- 需要额外了解 ES 查询语法，门槛高。

## 存储成本高

- 全字段索引：索引在表创建时是固定的，基于多维度查询创建了全字段索引，存储成本相对较高。

## 写入性能受限

- 数据写入时倒排索引需要进行分词，排序等一些 CPU 密集型操作，大量的倒排索引会导致写入性能下降。



# 基于 SelectDB 带来的收益

## 开发灵活

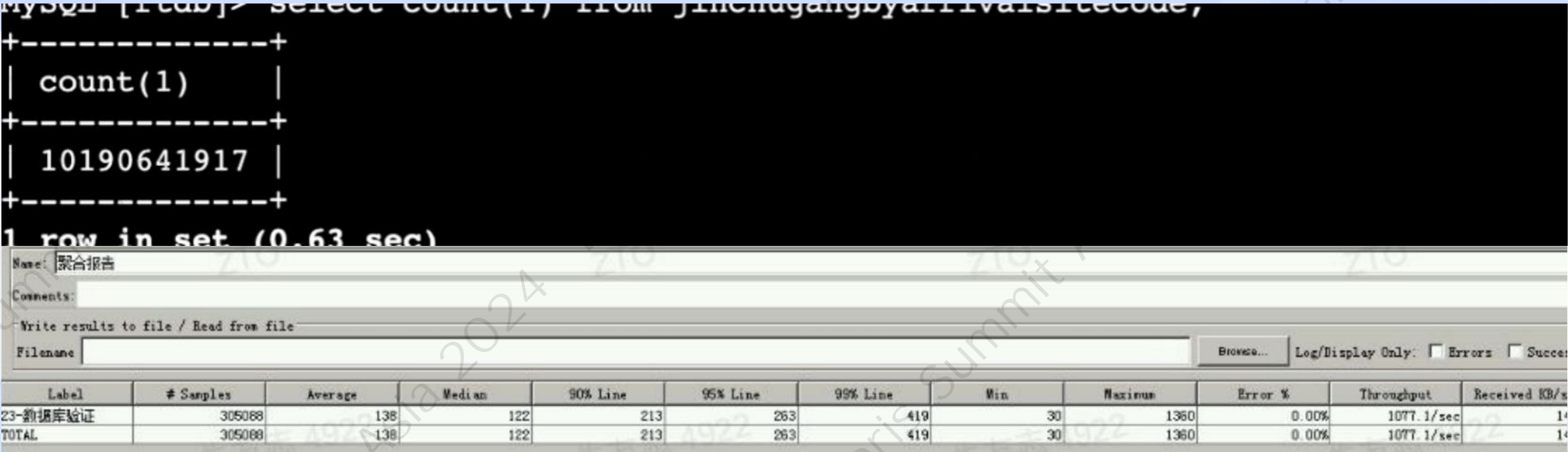
SelectDB 兼容 MySQL，使开发操作简单、使用门槛低，并且易于部署、迁移与运维。

## 存储成本低

按需设置倒排索引，降低数据存储，另外 SelectDB 列存采用的 ZSTD 压缩算法，可实现 8:1 的压缩比，进一步降低存储成本。

## 多场景下的查询加速

高频使用 SelectDB 分区分桶裁剪功能，在查询时过滤非必要的数​​据，因而在百亿数据集下，依然能有很快的响应时间和更高的并发。





# SelectDB Vs Presto

```
MySQL [finebi]> select substr(report_time, 1, 10) as date,
hive_test.dw.dw_log_metric_a as page,
tag2 as tag,
PV,
UV
from
and (
tag2 != 'test0001')
order by date, page, tag;
```

日期	所访问页面	PV	UV
2024-11-06	首页	34	20
2024-11-06	寄国际页	16	13
2024-11-06	寄港澳台页	4	3
2024-11-07	寄国际页	11	7
2024-11-07	寄国际页	10	4

6 rows in set (34.46 sec)

```
MySQL [finebi]> select substr(report_time, 1, 10) as date,
hive_test.dw.dw_log_metric_a as page,
tag2 as tag,
PV,
UV
from
and (
tag2 != 'test0001')
order by date, page, tag;
```

日期	所访问页面	PV	UV
2024-11-06	首页	34	20
2024-11-06	寄国际页	16	13
2024-11-06	寄港澳台页	4	3
2024-11-07	寄国际页	11	7
2024-11-07	寄国际页	10	4

6 rows in set (32.98 sec)

SelectDB

```
presto:default> select substr(report_time, 1, 10) as date,
hive_test.dw.dw_log_metric_a as page,
tag2 as tag,
PV,
UV
from
and (
tag2 != 'test0001')
order by date, page, tag;
```

日期	所访问页面	PV	UV
2024-11-06	寄国际页	16	13
2024-11-06	寄国际页	4	3
2024-11-07	寄国际页	11	7
2024-11-07	寄国际页	10	4

(6 rows)

```
presto:default> select substr(report_time, 1, 10) as date,
hive_test.dw.dw_log_metric_a as page,
tag2 as tag,
PV,
UV
from
and (
tag2 != 'test0001')
order by date, page, tag;
```

日期	所访问页面	PV	UV
2024-11-06	寄国际页	16	13
2024-11-06	寄国际页	4	3
2024-11-07	寄国际页	11	7
2024-11-07	寄国际页	10	4

(6 rows)

## 数据量

单分区: 100GB

数据格式: ORC

SelectDB 较 Presto 有1-2倍的性能提升

## 参数优化

dfs.client.socket-timeout=500

解决长尾问题, hdfs集群较大, 负载较高时, 调低socket-timeout, 快速失败, 从新的block块进行读取重试, 提升稳定性

04

# 未来展望

# 未来规划

## 更直观的 Profile

目前 SQL 调优，严重依赖 Profile，优化难度大，期望 Profile 能精简，更直观，方便用户调优

## 加强在数据湖上的应用

借助 Multi-Catalog 功能，支持多种异构数据源上的联邦查询，提升 SelectDB 在 Hudi/Paimon 等数据湖上的分析能力

## 引入多租户和资源隔离

降低用户之间的影响，针对不同的用户优先级，分配相应的资源

## 打通 Hive 外表权限验证

使用 Hive-Catalog 后的权限希望能透传 JDBC 账号，用于和现有大数据权限体系打通，进行权限管控





# Thanks for Watching!