

直播将于 **19:30** 准时开始，请耐心等待 ~



需求&问题收集



关注 SelectDB



Apache Doris 2.1: 数据湖分析能力全面增强

陈明雨 - Apache Doris PMC Member

邹新一 - Apache Doris Committer



目录

01 Apache Doris 湖仓一体

02 Apache Doris vs. Trino

03 Arrow Flight 高性能数据传输

04 Lakehouse Roadmap

Apache Doris 湖仓一体

湖仓加速



- 高性能执行引擎 + CBO 优化器 + 缓存 + 物化视图
- 相比 Hive、Spark、Trino 3 - 100 倍提升。

统一数据分析网关



- 多源联邦查询 + 统一元数据管理 + 多 SQL 方言支持
- 平滑业务迁移，让跨源分析简单高效。

高效快捷的数据湖构建



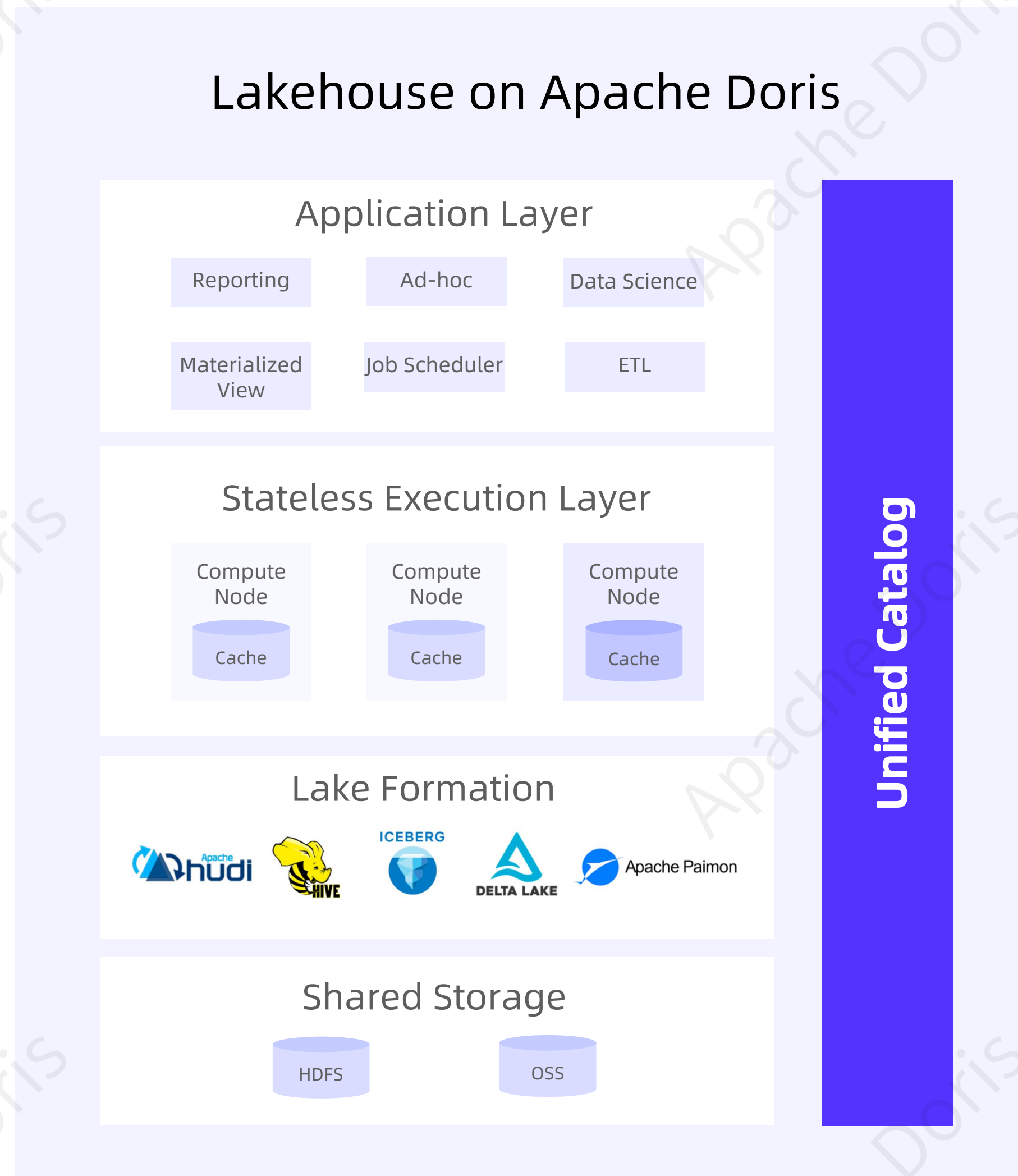
- Hive、Iceberg 等数据湖仓回写 + 内置任务调度
- 统一高效的湖仓数据管理与数据建模。

面向数据科学场景的高性能数据底座



- 数据湖开放格式 + Arrow Flight 高性能数据传输通道
- 为数据科学场景提供高性能数据访问和管理能力。

Apache Doris Lakehouse 架构



Application Layer

- 报表、多维分析、数据科学
- 数据加工、建模、任务管理调度

Stateless Execution Layer

- 无状态的计算集群
- Virtual Warehouse、负载隔离
- 数据缓存服务

Lake Formation

- 支持各类数仓数据格式
- 提供丰富的数据读写能力

Shared Storage

- 无限扩展的数据存储底座

Apache Doris 数据湖能力一览



- Parquet + ORC + Text
- Hive Metastore 1.x / 2.x / 3.x Compatibility
- Write Back (Insert + Insert Overwrite)



- Format V1 + V2
- Parquet + ORC
- Position Delete + Equality Delete
- Time Travel + Schema Evolution
- Multi Type Catalog
- Write Back (WIP)



- COW + MOR
- Incremental Read
- Time Traval



- Native File Read Optimization (Paimon 0.6+)
- Native Deletion Vector Support (Paimon 0.8+)

基于物化视图的数据湖查询加速

湖仓数据物化

Hive、Iceberg、Hudi、Paimon、...
解决远端存储访问性能问题
解决存储资源开销问题



增量构建

分区级别的增量构建
时效性高、资源暂用低



透明改写

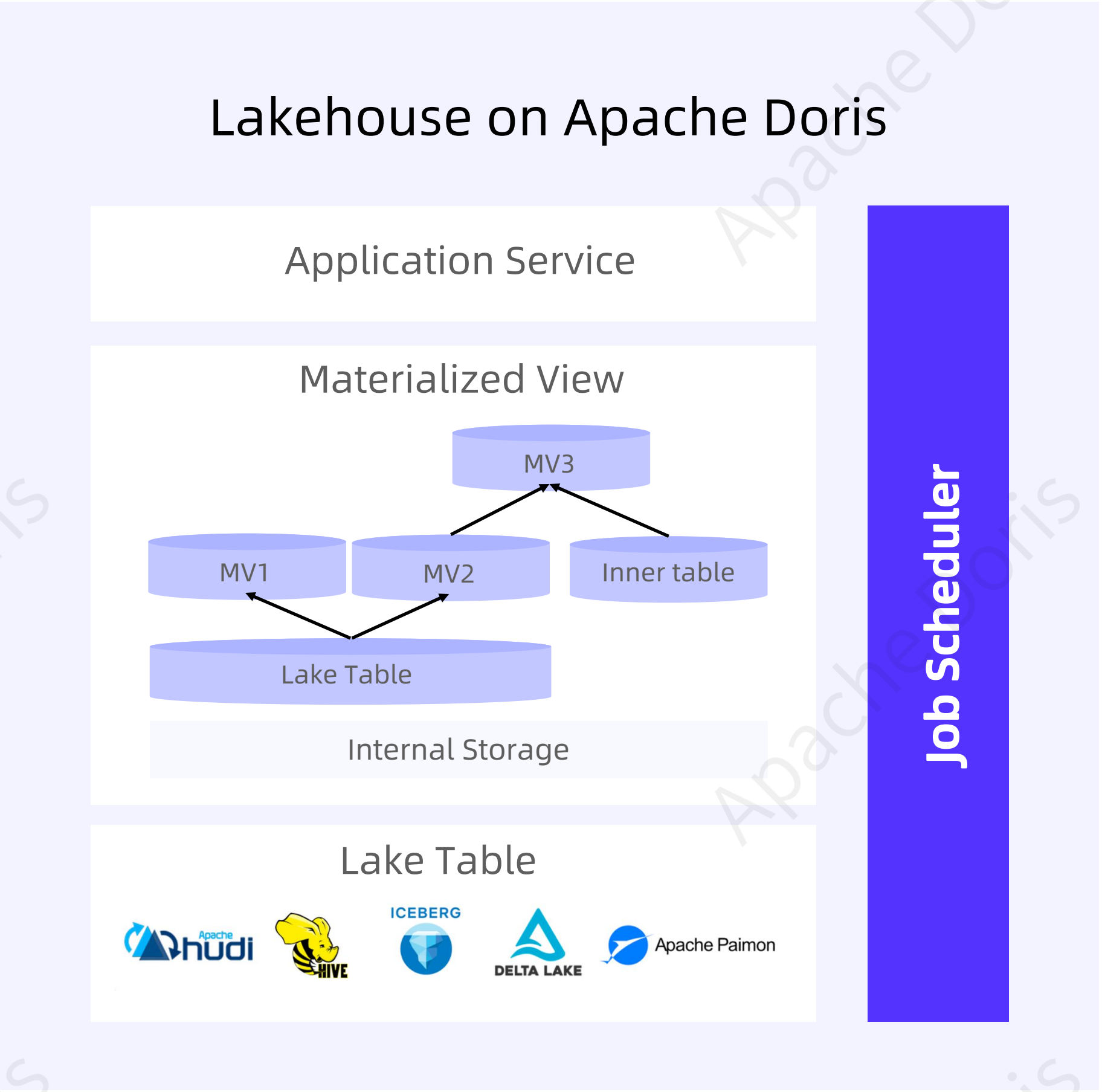
提供数据湖透明查询加速能力



```
CREATE MATERIALIZED VIEW (IF NOT EXISTS)? mvName=multipartIdentifier
  (LEFT_PAREN cols=simpleColumnDefs RIGHT_PAREN)? buildMode?
  (REFRESH refreshMethod? refreshTrigger?)?
  (KEY keys=identifierList)?
  (COMMENT STRING_LITERAL)?
  (PARTITION BY LEFT_PAREN partitionKey = identifier RIGHT_PAREN)?
  (DISTRIBUTED BY (HASH hashKeys=identifierList | RANDOM) (BUCKETS (INTEGER_VALUE | AUTO))?)?
  propertyClause?
  AS query
```



基于物化视图的数据湖查询加速



Advantage

- 避免冗余存储
- 透明加速
- 数仓分层
- 轻量ETL

目录

01 Apache Doris 湖仓一体

02 Apache Doris vs. Trino

03 Arrow Flight 高性能数据传输

04 Lakehouse Roadmap

Apache Doris vs. Trino

What's in Common

- 标准 ANSI SQL 支持
- MPP & Pipeline 执行模型
- 多数据源支持
- 基于代价的查询优化器

Advantages in Apache Doris

- 向量化执行引擎
- 更丰富的统计信息采集策略
- 更丰富的 Join 算法支持: Colocation/Bucket/Shuffle/Broadcast Join; Inner/Outer/Semi/Anti/Cross Join
- 更丰富的缓存类型: Meta Cache, File Cache, Query Cache, Partition Cache
- 更高级的物化视图能力: 高性能数据格式、增量更新、查询透明改写
- SQL 方言支持: Trino/Presto, Clickhouse, PG, etc.
- 兼容 Trino connector

Apache Doris vs. Trino

Doris (2.1.3) 集群环境

- 1 FE + 3 BE
- FE: 阿里云 ecs.g7.xlarge (4C 16G)
- BE: 阿里云 ecs.g7.4xlarge (16C 128G)

Trino (422) 集群环境

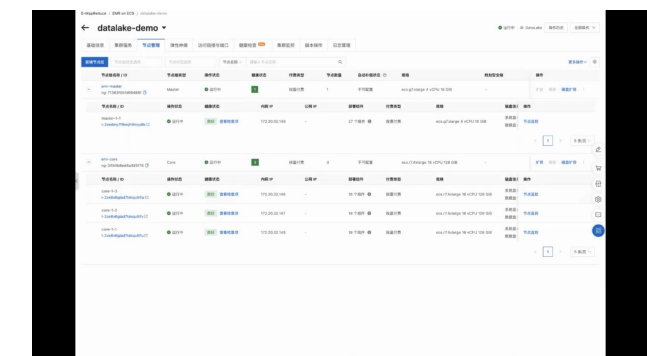
- 1 Coord + 3 Worker
- Coord: 阿里云 ecs.g7.xlarge (4C 16G)
- Worker: 阿里云 ecs.g7.4xlarge (16C 128G)

测试方法

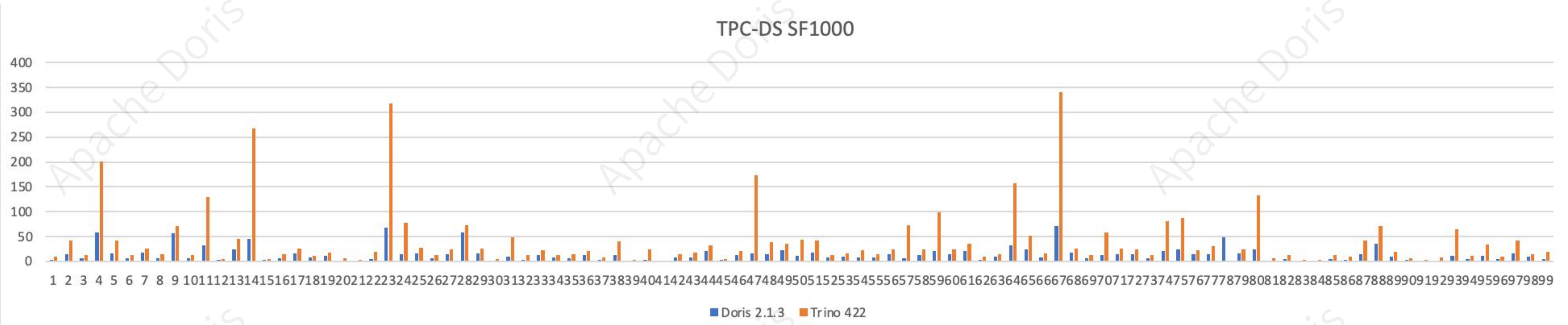
- TPCDS 99 条 SQL
- 每条 SQL 连续跑 3 次，取最后一次
- 无缓存

测试数据集

- 阿里云 DLF + OSS
- Hive TPCDS 1TB
- ORC + ZLIB



Apache Doris vs. Trino

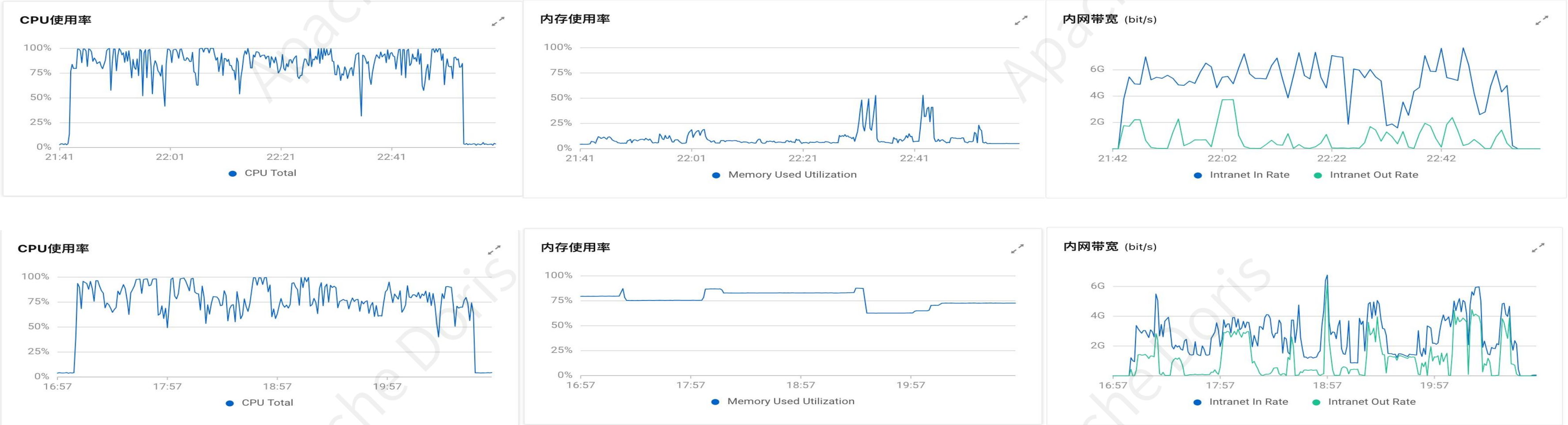


总耗时

资源开销

Doris 1404s

Trino 4043s



Apache Doris Trino Connector 兼容

只需4步，快速对接新数据源

1. 将编译好的 Trino-Connector 放到 FE、BE节点指定位置。
2. 在 Doris 执行 CREATE CATALOG 创建对应的 Catalog
3. 开始查询
4. <https://doris.apache.org/zh-CN/community/how-to-contribute/trino-connector-developer-guide/>

适配示例

- Kafka

```
create catalog kafka properties (  
  "type"="trino-connector",  
  
  "connector.name"="kafka",  
  "kafka.nodes"="localhost:9092",  
  "kafka.table-description-supplier"="CONFLUENT",  
  "kafka.confluent-schema-registry-url"="http://localhost:8081",  
  "kafka.hide-internal-columns" = "false"  
);
```

- BigQuery

```
create catalog bigquery_catalog properties (  
  "type"="trino-connector",  
  
  "connector.name"="bigquery",  
  "bigquery.project-id"="steam-circlet-388406",  
  "bigquery.credentials-  
file"="/path/to/application_default_credentials.json"  
);
```

Apache Doris Trino Connector 兼容

通过 Trino Connector 快速构建测试集

- 演示如何通过 Doris + Aliyun DLF + OSS 快速生成一个 TPC-DS 测试数据集



Apache Doris Trino Connector 兼容

适配更多数据源

- DeltaLake
- Redis
- MongoDB
- ...
- <https://trino.io/docs/current/connector.html>

Apache Doris SQL 方言兼容

基于 SQLGlue 二次开发

- Trino/Presto
- Clickhouse
- PostgreSQL
- ...
- <https://doris.apache.org/zh-CN/docs/dev/lakehouse/sql-dialect/>
- https://play.selectdb.com/sql-converter?case_sensitive=0



目录

01 Apache Doris 湖仓一体

02 Apache Doris vs. Trino

03 Arrow Flight 高性能数据传输

04 Lakehouse Roadmap

Arrow Flight 高性能数据传输

- 在数据科学场景中，经常需要从 Doris 加载大量数据到 Python/Java，过去使用 Pymysql/Pandas 或 JDBC 加载大批量数据的速度很慢。
- Doris 基于 Arrow Flight SQL 协议实现了高速数据链路，支持多种语言使用 SQL 从 Doris 高速读取大批量数据。

```
Python:
conn = flight_sql.connect(uri="grpc://127.0.0.1:9090", db_kwargs={ "root",
""},})
cursor = conn.cursor()
cursor.execute("select * from arrow_flight_sql_test order by k0;")
print(cursor.fetchallarrow().to_pandas())
```

k0	k1	K2	k3	k4	k5
0	0	0.10000	ID	0.000100000	9999999999 2023-10-21
.....					
4	4	122345.54321	ID	122345.543210000	5 2023-10-22

用法和使用 Mysql 协议的各种连接方法类似；

性能提升 10 - 100 倍

此外兼容标准的 JDBC API，支持 BI 工具使用 JDBC 访问；

未来

- 期望支持 ADBC 客户端读写Doris，实现高速数据交互；
- 借助 Arrow Flight 多节点多核特性实现多 BE 并行传输；
- 借助 Arrow Flight SQL 实现 Doris 和 Doris 间的联邦查询；

Arrow Flight 高性能数据传输

Lakehouse on Apache Doris

Application Layer

Python Program

Ad-hoc

Data Science



JDBC

NO



ADBC

YES

Unified Catalog

Lake Formation

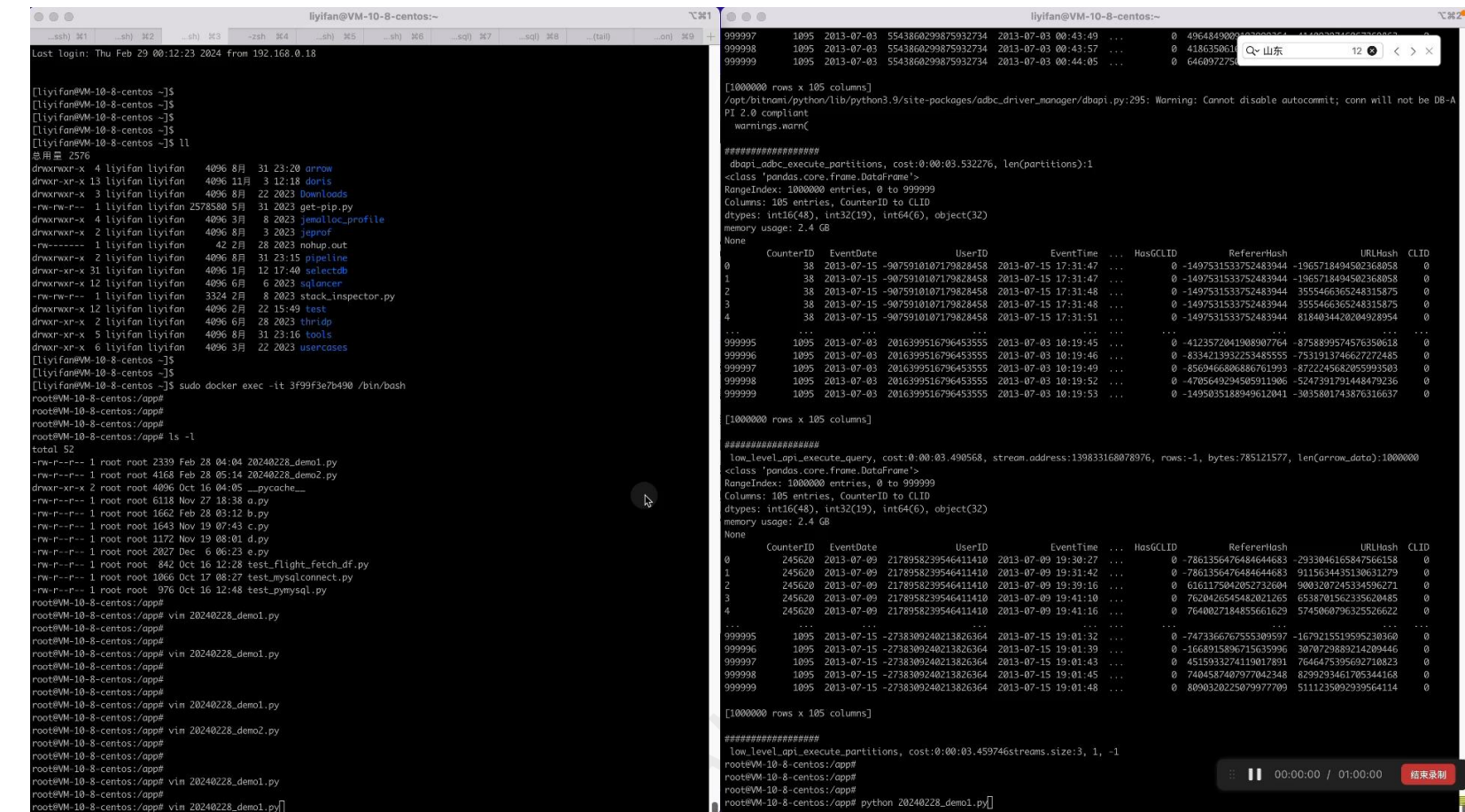


Apache Paimon

Shared Storage

HDF

OS



目录

01 Apache Doris 湖仓一体

02 Apache Doris vs. Trino

03 Arrow Flight 高性能数据传输

04 Lakehouse Roadmap

Roadmap

更多功能支持

- Iceberg: 支持 ORC、Equality Delete
- Paimon: 支持 Deletion Vector
- Hudi: 支持 CDC
- Lakesoul、DeltaLake、BigQuery、MongDB...

更完善的数据集成和数据加工能力

- 基于 Hive、Iceberg 数据湖构建能力: 数据的读取、加工、写回、...
- 各个湖格式的增量物化视图刷新能力。
- 更强大的数据处理能力: 查询落盘、基于DAG的作业调度。

欢迎加入 Apache Doris

加入数据湖分析专项群

扫码添加 Doris 小助手微信，备注“数据湖” / 默认邀请普通用户群

提问赢好礼

提问被选中回复的小伙伴，请添加小助手微信领取 Doris 精美周边，数量有限，先到先得~

Doris 问答论坛，遇到问题先看论坛

地址：ask.selectdb.com



Doris 小助手



扫一扫上面的二维码图案，加我为朋友。

Thanks !

