

直播将于 **19:30** 准时开始，请耐心等待 ~



需求&问题收集



关注 SelectDB



异步物化视图 功能解析与场景应用

张文歆 - SELECTDB 资深研发, Apache Doris PMC



目录

背景介绍

功能和原理

Demo演示

未来规划

Q & A

什么是物化视图

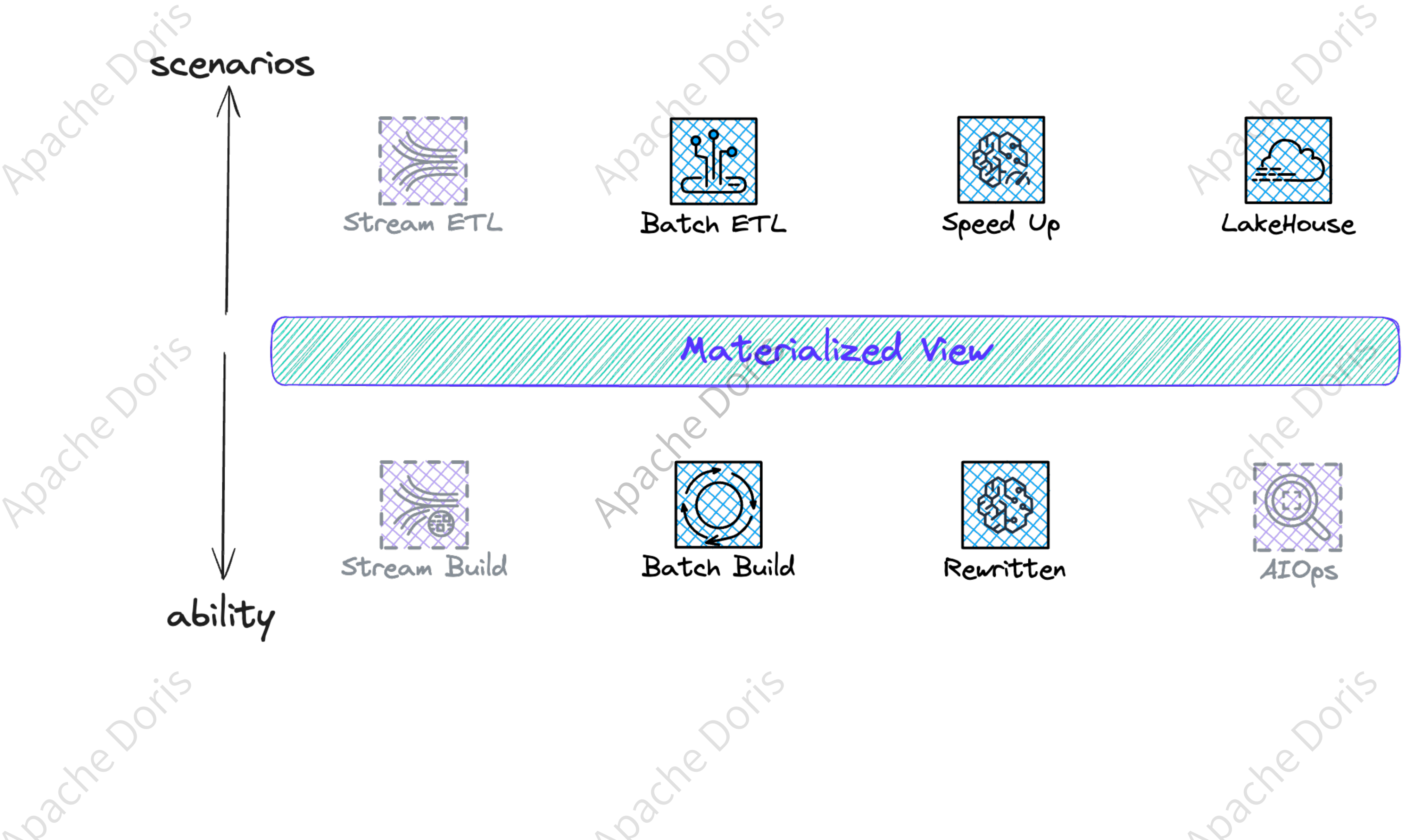
视图

与关系数据库中的普通基本表不同，视图不是物理模式的一部分：作为结果集，它是从数据库中的数据动态计算或整理而成的虚拟表。当请求访问该视图时，对相关底层表中的数据所做的更改会反映在后续调用该视图时显示的数据中。没有使用视图，对于超过第二范式的数据库规范化将变得更加困难。视图可以使得创建无损连接分解更加容易。

物化视图

在计算领域，物化视图是一个包含查询结果的数据库对象。例如，它可能是位于远程位置的数据的本地副本，或者可能是表或连接结果的行和/或列的子集，或者可能是使用聚合函数的摘要。

为什么做异步物化视图



同步和异步物化视图的差异

	同步物化视图	异步物化视图
数据更新方式	同步	异步 / 全量 或 分区增量
一致性	强一致	最终一致
支持表类型	内表	所有
支持内表模型	Duplicate / Aggregate （聚合和前缀索引） Unique （前缀索引）	所有
支持算子	聚合	所有（直查） / 聚合、连接、排序、...（改写）
删除数据限制	只能在所有单表物化视图都有的列上写条件	无限制
数据分布限制	必须和基表相同	可以灵活定义

目录

背景介绍

功能和原理

Demo演示

未来规划

Q & A

构建语法

```
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
```

buildMode	refreshMethod
: BUILD (IMMEDIATE DEFERRED)	: COMPLETE AUTO
;	;

```
refreshTrigger
: ON MANUAL
| ON SCHEDULE refreshSchedule
;

refreshSchedule
: EVERY INTEGER_VALUE mvRefreshUnit (STARTS STRING_LITERAL)?
;

mvRefreshUnit
: MINUTE | HOUR | DAY | WEEK
;
```

关键点

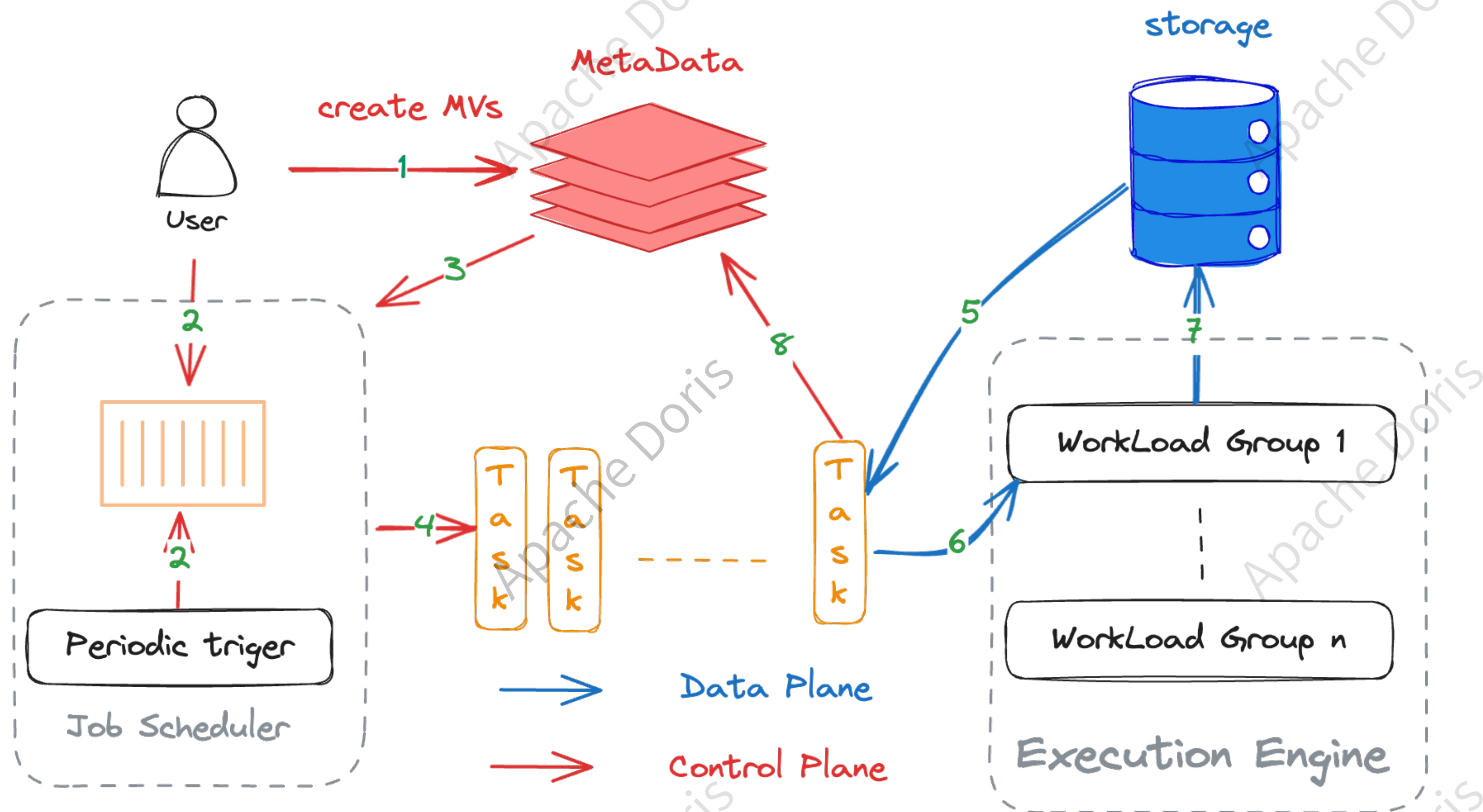
构建模式：立即，延迟

刷新方式：全量，自动

刷新触发：手动，定时

不可指定：列类型，表模型

构建实现逻辑

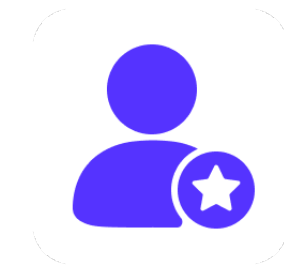


分区增量构建



分区增量构建是什么

- 自动判断基表分区和物化表分区的对应关系
- 只更新受基表数据变更影响的物化表数据



分区增量构建的约束

- 基表只有一个分区字段
- 基表的一个分区数据变更只影响物化表的一个分区



分区增量构建的优势

- 构件时时效性更高，占用资源更少
- 可以利用分区TTL的能力减少存储占用



如何确认使用了分区增量构建

- 如果不能分区增量构建则创建时会报错

基于外表构建的约束

支持哪些外表类型

- Hive
- Iceberg
- Hudi
- Paimon
- JDBC
- ...

哪些外表支持分区增量更新

- Hive
- MySQL
- Iceberg
- Hudi
- Paimon
- ...

注意事项

- 外表默认数据都为新鲜的
- 刷新策略为 complete 时才刷新
- Hive 支持检查数据是否同步
- 其他外表类型暂时不支持

异步物化视图特有的属性

`grace_period`：查询改写时允许物化视图数据的最大延迟时间（单位：秒）。如果分区A和基表的数据不一致，物化视图的分区A上次刷新时间为1，系统当前时间为2，那么该分区不会被透明改写。但是如果`grace_period`大于等于1，该分区就会被用于透明改写

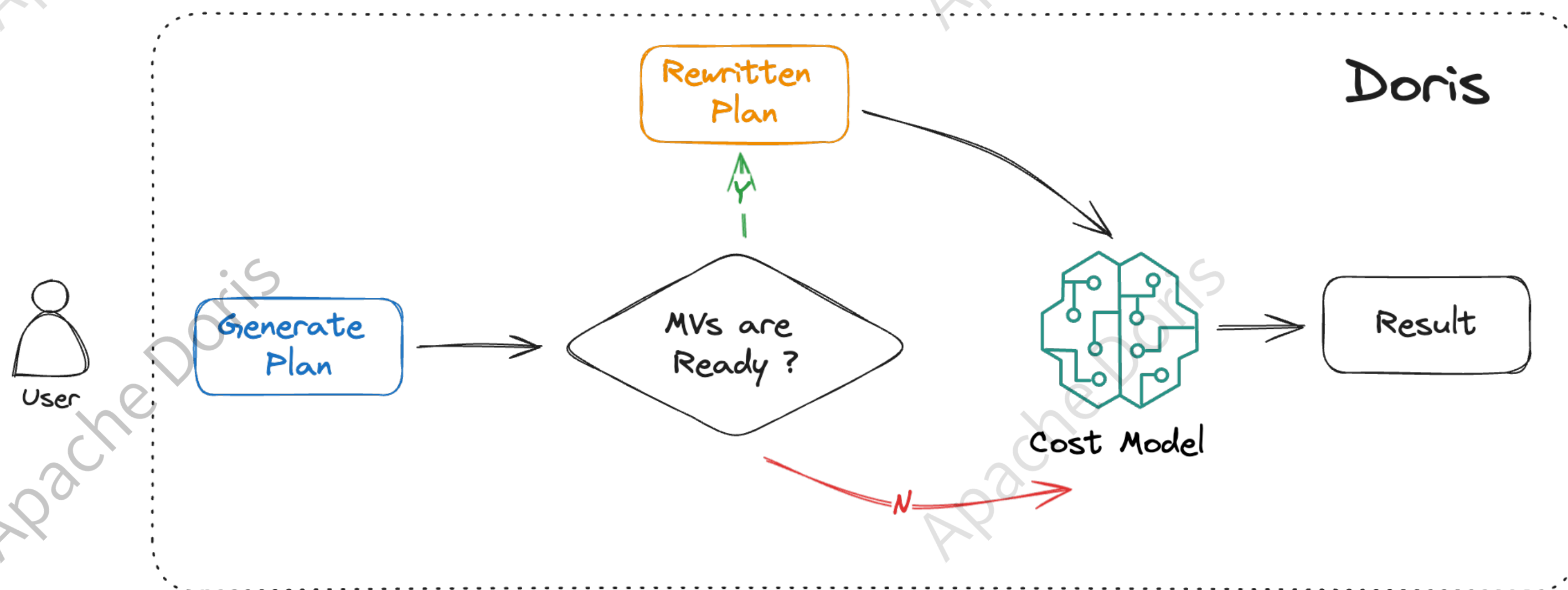
`excluded_trigger_tables`：数据刷新时忽略的表名，逗号分割。例如 `table1,table2`

`refresh_partition_num`：单次insert语句刷新的分区数量，默认为1。物化视图刷新时会先计算要刷新的分区列表，然后根据该配置拆分成多个insert语句顺序执行。遇到失败的insert语句，整个任务将停止执行。物化视图保证单个insert语句的事务性，失败的insert语句不会影响到已经刷新成功的分区。

`workload_group`：物化视图执行刷新任务时使用的workload_group名称。用来限制物化视图刷新数据使用的资源，避免影响到其它业务的运行。workload_group创建及使用 [WORKLOAD-GROUP](#)

透明改写流程

- 1. 用户输入查询，生成计划 Plan 1
- 2. 对 Plan 1 使用物化视图改写，生成若干改写后的计划：Plan 2, Plan 3, ...
- 3. 将以上所有候选放入代价模型中
- 4. 选出代价最小的计划作为最终执行的计划



透明改写能力范围

聚合改写

```
1 CREATE MATERIALIZED VIEW agg_rollup
2 BUILD IMMEDIATE REFRESH AUTO ON SCHEDULE EVERY 1 hour
3 DISTRIBUTED BY RANDOM BUCKETS 3
4 PROPERTIES ('replication_num' = '1')
5 AS
6 SELECT
7     o_shippriority, o_comment,
8     SUM(o_totalprice),
9     MAX(o_totalprice),
10    MIN(o_totalprice),
11    COUNT(*)
12 FROM orders
13 GROUP BY
14 o_shippriority,
15 o_comment;
```

物化

```
1 SELECT
2     o_shippriority,
3     SUM(o_totalprice),
4     MAX(o_totalprice),
5     MIN(o_totalprice),
6     COUNT(*)
7 FROM orders
8 GROUP BY
9 o_shippriority;
```

查询

查询中函数	物化视图中函数	函数上卷后
max	max	max
min	min	min
sum	sum	sum
count	count	sum
count(distinct)	bitmap_union	bitmap_union_count
bitmap_union	bitmap_union	bitmap_union
bitmap_union_count	bitmap_union	bitmap_union_count

透明改写能力范围

JOIN改写

物化

```
1 CREATE MATERIALIZED VIEW join_rewrite
2 BUILD IMMEDIATE REFRESH AUTO ON SCHEDULE EVERY 1 hour
3 DISTRIBUTED BY RANDOM BUCKETS 3
4 PROPERTIES ('replication_num' = '1')
5 AS
6 SELECT t1.l_linenumber,
7        o_custkey,
8        o_orderdate
9 FROM (SELECT * FROM lineitem WHERE l_linenumber > 1) t1
10 LEFT OUTER JOIN orders
11 ON l_orderkey = o_orderkey;
```

查询

```
1 SELECT l_linenumber,
2        o_custkey
3 FROM lineitem
4 LEFT OUTER JOIN orders
5 ON l_orderkey = o_orderkey
6 WHERE l_linenumber > 1 AND o_orderdate = '2023-10-18';
```

支持的 JOIN 类型

- INNER JOIN
- LEFT OUTER JOIN
- RIGHT OUTER JOIN
- FULL OUTER JOIN
- LEFT SEMI JOIN
- RIGHT SEMI JOIN
- LEFT ANTI JOIN
- RIGHT ANTI JOIN

透明改写能力范围

JOIN衍生

```
1 CREATE MATERIALIZED VIEW join_derive
2 BUILD IMMEDIATE REFRESH AUTO ON SCHEDULE EVERY 1 hour
3 DISTRIBUTED BY RANDOM BUCKETS 3
4 PROPERTIES ('replication_num' = '1')
5 AS
6 SELECT
7     l_shipdate, l_suppkey, o_orderdate,
8     SUM(o_totalprice) AS sum_total,
9     MAX(o_totalprice) AS max_total,
10    MIN(o_totalprice) AS min_total,
11    COUNT(*) AS count_all
12 FROM lineitem
13 LEFT OUTER JOIN orders ON lineitem.l_orderkey = orders.o_orderkey AND l_shipdate = o_orderdate
14 GROUP BY
15 l_shipdate,
16 l_suppkey,
17 o_orderdate;
```

物化

```
1 SELECT
2     l_shipdate, l_suppkey, o_orderdate,
3     SUM(o_totalprice) AS sum_total,
4     MAX(o_totalprice) AS max_total,
5     MIN(o_totalprice) AS min_total,
6     COUNT(*) AS count_all
7 FROM lineitem
8 INNER JOIN orders ON lineitem.l_orderkey = orders.o_orderkey AND l_shipdate = o_orderdate
9 WHERE o_orderdate = '2023-10-18' AND l_suppkey = 3
10 GROUP BY
11 l_shipdate,
12 l_suppkey,
13 o_orderdate;
```

查询

衍生条件

查询	物化视图	查询中衍生条件
Inner join	Left outer join	改写后右侧补偿拒绝null的谓词
Inner join	right outer join	改写后左侧补偿拒绝null的谓词
Inner join	full outer join	改写后补偿拒绝null的谓词
left outer join	inner join	右输入有拒绝null的条件
Left outer join	full outer join	join左输入需要有拒绝null的条件
right outer join	inner join	左输入有拒绝null的条件
right outer join	full outer join	join右输入需要有拒绝null的条件

透明改写能力范围

JOIN衍生

1: Left Match

2: Right Match

inner join: 1 + 2

3: Left Not
Match

4: Right NULL

Left Outer join: 1 + 2 + 3 + 4

Right Outer join: 1 + 2 + 5 + 6

5: Left NULL

6: Right Not
Match

Full Outer join: 1 + 2 + 3 + 4 + 5 + 6

透明改写能力范围

谓词补偿

- 包含两种常见场景：
- 1. 查询中包含多个合取谓词：WHERE c1 IS NULL AND c2 > 10。物化中包含查询中的一部分：WHERE c1 IS NULL
- 2. 查询中的谓词在逻辑上是物化中的子集。查询：WHERE c1 > 10。物化：WHERE c1 > 5

MV

c1 > 10

5 < c1 <= 10

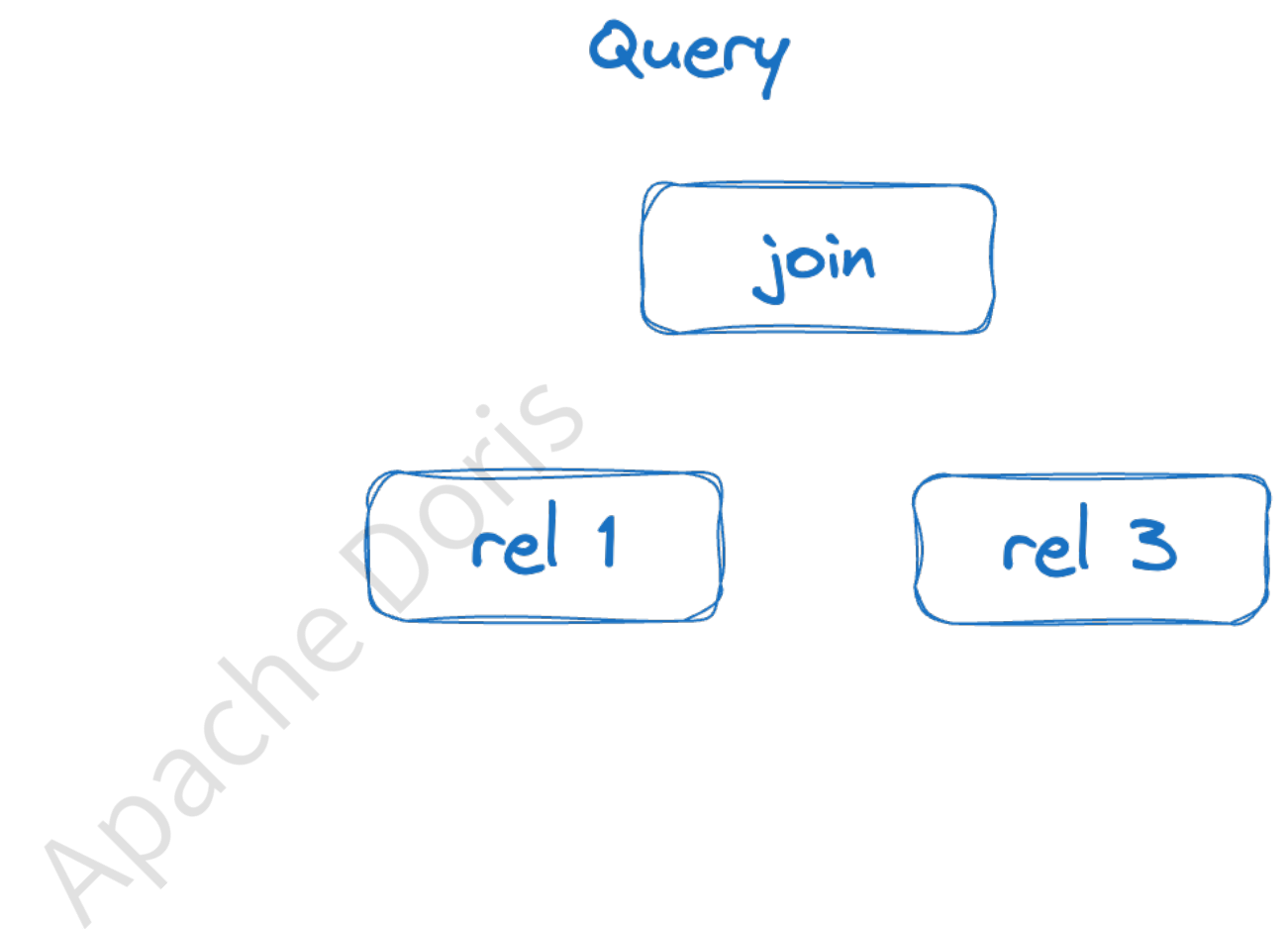
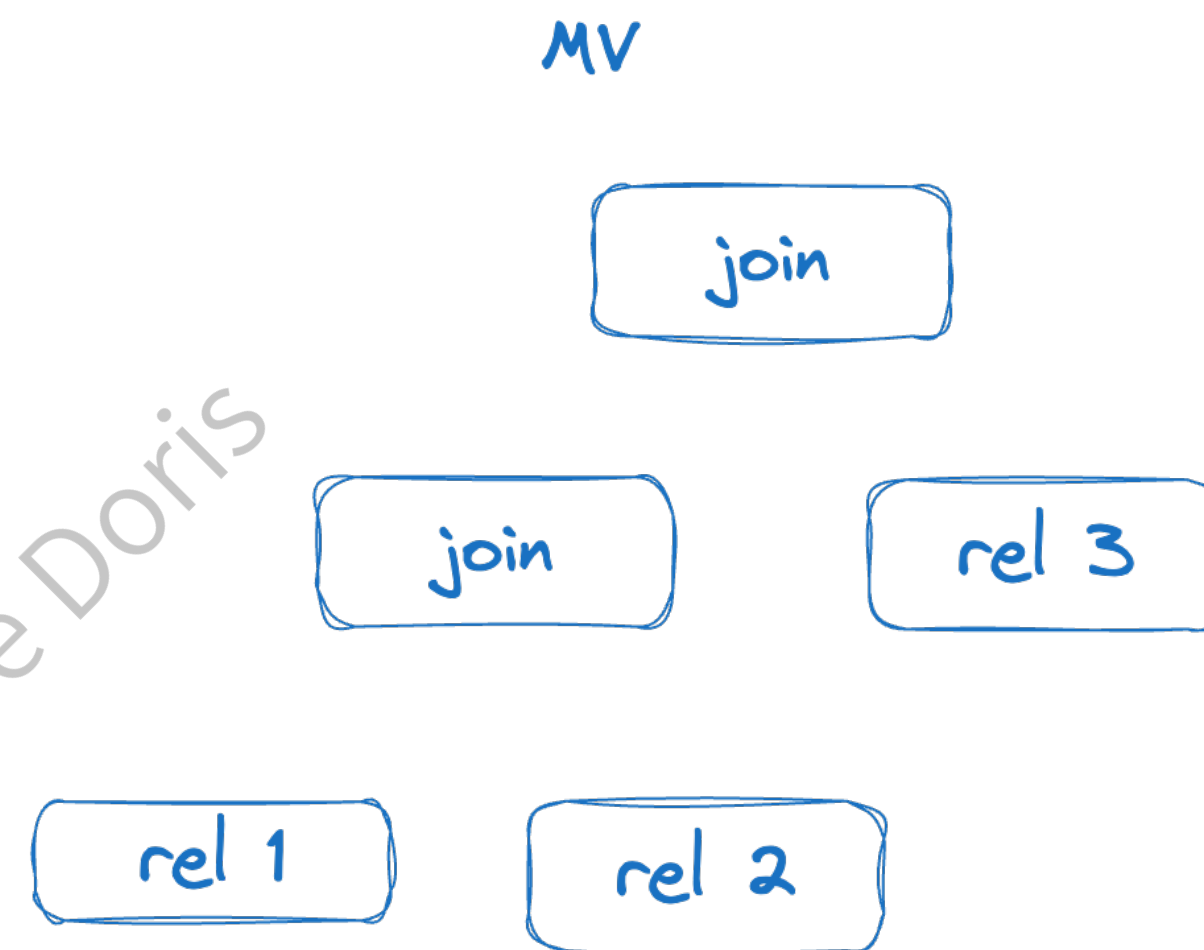
Query

c1 > 10

透明改写能力范围

Query partial (Coming soon)

- 查询中的 JOIN 是物化中的一部分
- 当满足：
 1. rel2 对于 rel1 没有过滤作用
 2. rel2 对于 rel1 没有膨胀作用
- 通常来讲，rel2 和 rel1 的连接键满足主外键关系

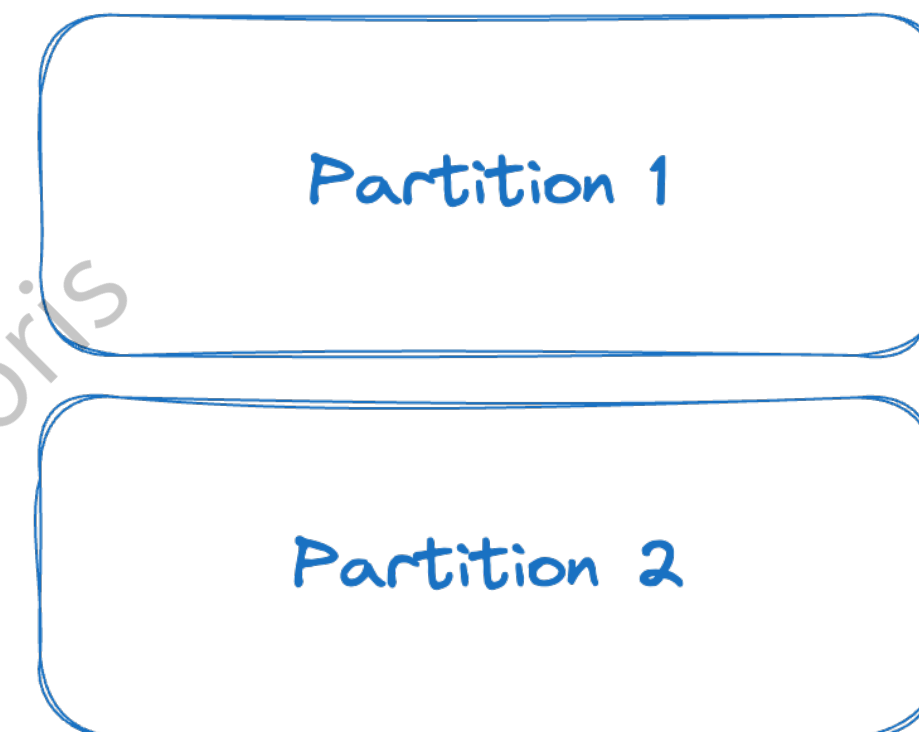


透明改写能力范围

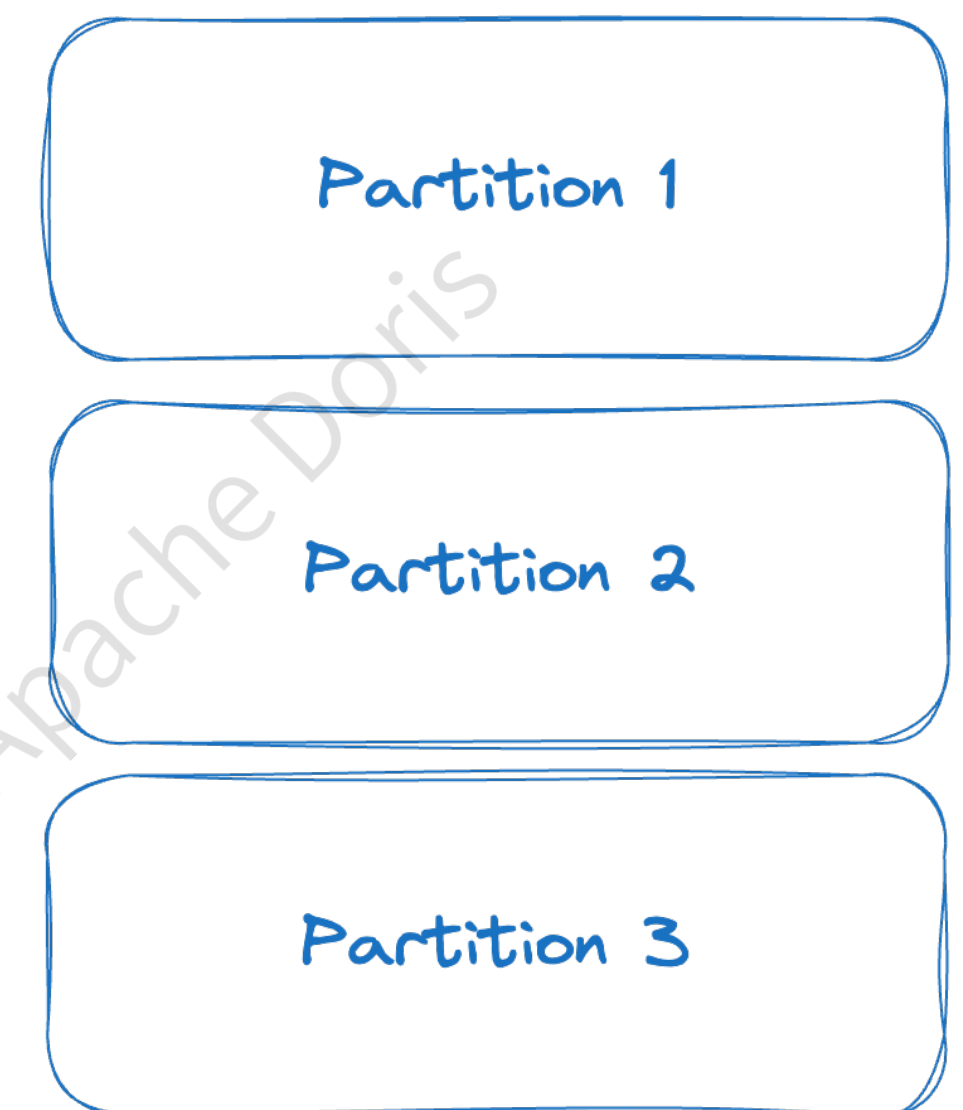
Union 改写 (Coming Soon)

- 物化视图使用分区增量构建
- 物化视图中的部分分区与基表同步，部分不同步
- 查询同时命中了同步和不同步的分区
- 同步分区直接改写为查询物化视图 (D1)
- 未同步的分区保留原始查询 (D2)
- 最终结果为 D1 UNION ALL D2

MV



Query



透明改写最佳实践

- 只在查询加速比明显且使用频繁的场景使用。
- 异步物化视图的构建比较昂贵。要有严格的准入手册，防止异步物化视图的随意创建。
- 持续监控已经创建的物化视图的使用情况，防止无效的构建持续运行。
- 为了能更容易命中透明改写，尽量在物化视图中物化所有的列。
- 物化视图不是灵丹妙药，不能解决所有性能问题。

目录

背景介绍

功能和原理

Demo 演示

未来规划

Q & A

目录

背景介绍

功能和原理

Demo演示

未来规划

Q & A

未来规划

构建

- 分区上卷
- 物化视图嵌套视图和物化视图
- 支持内表全部属性和特性
- 更多种类的外表分区构建
- DAG调度
- 流式构建
- ...

改写

- 支持更多的聚合函数上卷
- 支持更多算子类型
- 支持物化视图嵌套视图和物化视图
- 单表直查改写
- Query Partial
- Union改写
- 通用聚合函数上卷（agg_state）
- ...

运维

- 细粒度构建指标追踪
- 透明改写追踪
- 物化视图使用情况追踪
- 使用Model智能推荐物化视图
- 使用查询历史智能推荐物化视图
- 无用物化视图告警和自动删除
- ...

目录

背景介绍

功能和原理

Demo演示

未来规划

Q & A

欢迎加入 Apache Doris

加入社区用户微信群

扫码添加 Doris 小助手，备注“加群”

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~



Thanks !

