

# Apache Doris 在虎牙业务 监控场景下的实践

陈仕明

虎牙 数据平台负责人

# 目录

1. OLAP 在虎牙的场景应用
2. 技术选型及设计
3. 关键技术实现
4. 未来规划



旗下产品



●“虎牙直播”诞生

- 8月10日，虎牙公司正式注册成立
- 深耕游戏直播，转战移动互联网

- 虎牙直播月活用户突破1亿
- 5月11日纽交所成功**上市**，股票代码HUYA
- 布局海外市场，发行海外产品**Nimo TV**

- 全年营收突破100亿
- Nimo TV海外月活突破3000万
- **腾讯控股虎牙**

● 布局新赛道

2020

2018

2016

2014

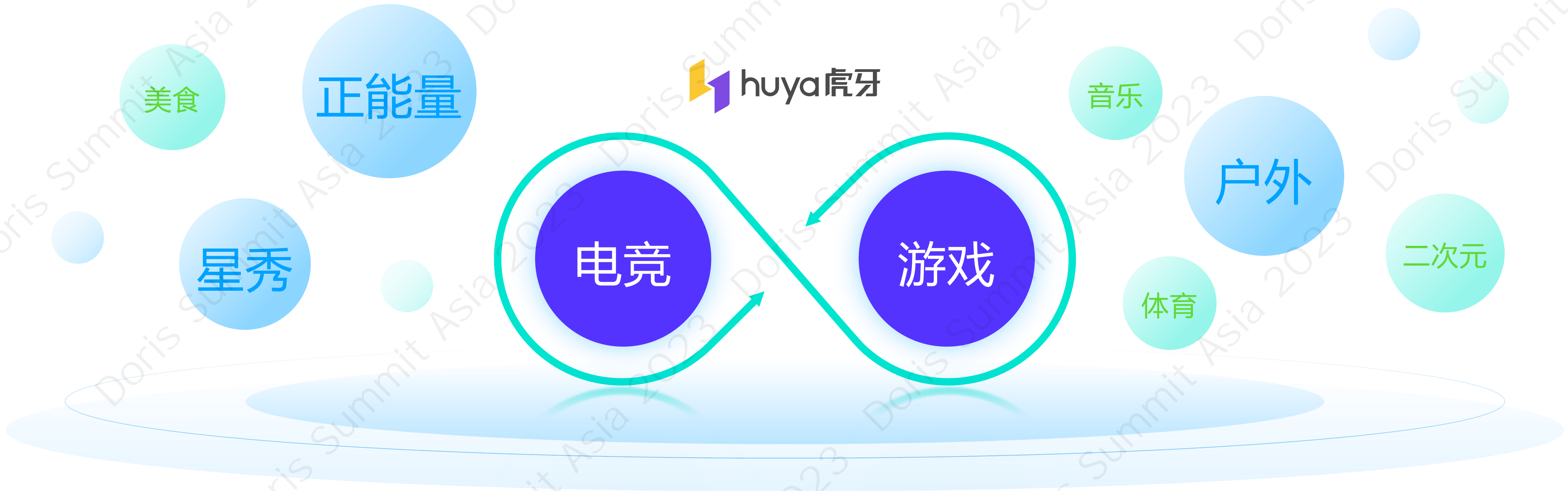
2023





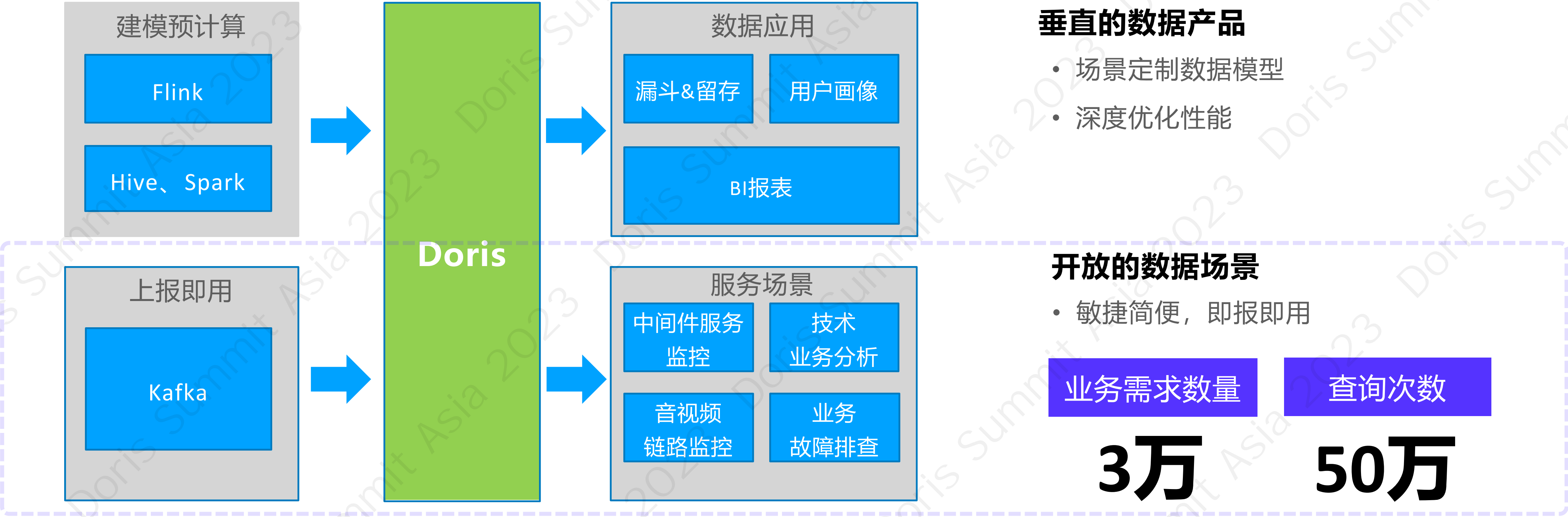
以电竞+游戏直播为核心 构建多元内容生态体系

截至2023年3月共播出官方赛事**182个**，共**7570场**，总直播时长**90840小时**



# 1 OLAP 在虎牙的场景应用

# 虎牙 OLAP 的应用



# 业务监控的典型场景案例

音视频全链路故障监控&排障	面向产品研发的业务分析	面向时间线的应用
<ul style="list-style-type: none"><li>• 推流/拉流链路监控及切换</li><li>• 大主播开播质量监控及排查</li><li>• P2P效果分析</li></ul>	<ul style="list-style-type: none"><li>• 技术开发的产品分析</li><li>• 中台产品的运营分析</li></ul>	<ul style="list-style-type: none"><li>• 微服务调用监控</li><li>• 基础运维监控</li><li>• 中间件监控</li></ul>

## 跟BI相比

- 更敏捷：无需建模，开发门槛更低，开发效率更高
- 更高吞吐：分钟粒度数据，每天万亿规模
- 延时更低：分钟内延时，否则可能误判/漏判

## 跟 Metric 监控相比

- 模型复杂：维度更多，聚合方式更灵活
- 计算复杂：存在关联分析场景，拆分集群较难
- 数据量更大：明细数据排障



使用方式

即报即用，无需建模

数据接入

采样控制

开关：

关

采样比例 ：

请输入

平台：

数据-dims

dims 配置：

dataflag	String	描述	数据标识	默认值	请输入
anchoruid	UInt64	描述	主播uid	默认值	请输入
ratio	Int32	描述	码率 (-1:所有码率汇总, 0:原画, 200	默认值	请输入
pcu	UInt32	描述	对应的pcu	默认值	0
pointits	UInt32	描述	pcu对应的时间戳	默认值	0

+ 添加 Dims 配置

如果是入到大数据的指标，修改字段类型请联系大数据团队支持

默认字段：

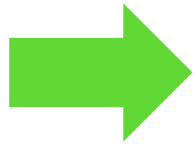
数据-fields

fields 配置：

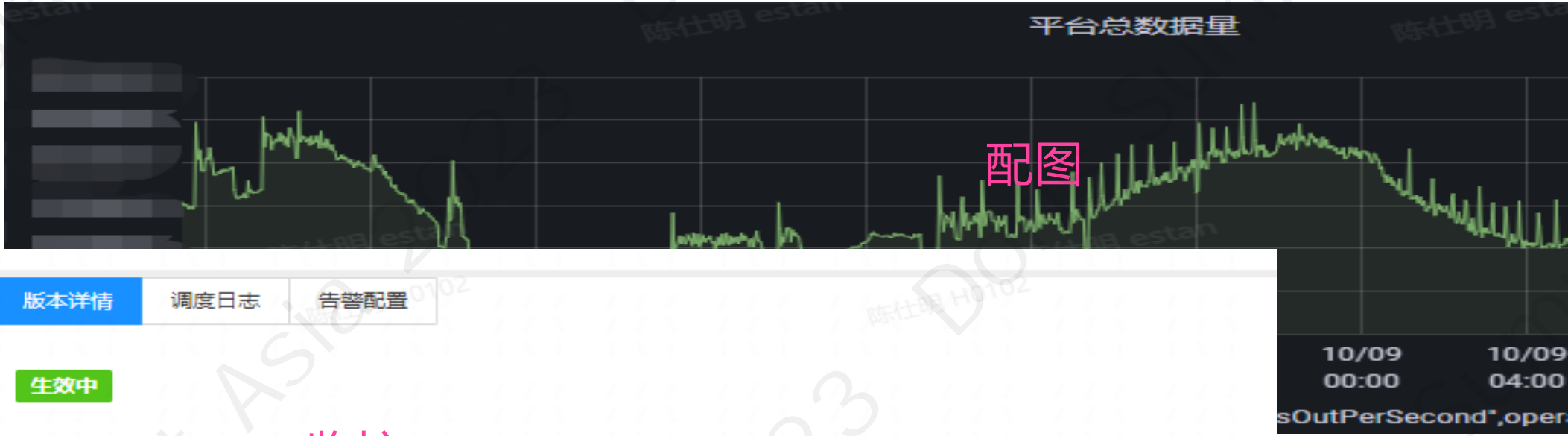
field0	Int8	描述	no use
--------	------	----	--------

+ 添加 fields 配置

如果是入到大数据的指标，修改字段类型请联系大数据团队支持



即刻使用



版本详情

调度日志

告警配置

生效中

名称：赛事4K异常监控

数据源：doris

最后更新者：dw\_chenyixian

描述：赛事4K异常监控

当前版本信息

创建于：2023-09-28 17:46:31

最后更新于：2023-09-28 18:42:26

数据集

异常任务修复

异常任务自愈

自动预览

列维

行维

过滤

指标

异常任务修复

time	cluster	tag
		Could not obtain block
		File does not exist
		file not found
		heap
		MAX_FAILED_UNIQUE_FETCHE
		Could not obtain block

分析



## 2 技术选型及设计

# 系统要求 & 选型

## 强大的分析性能

- 计算复杂性：单表查询，多表关联，细粒度分组
- 性能：万亿写入，分钟内延时

## 平台可透明优化

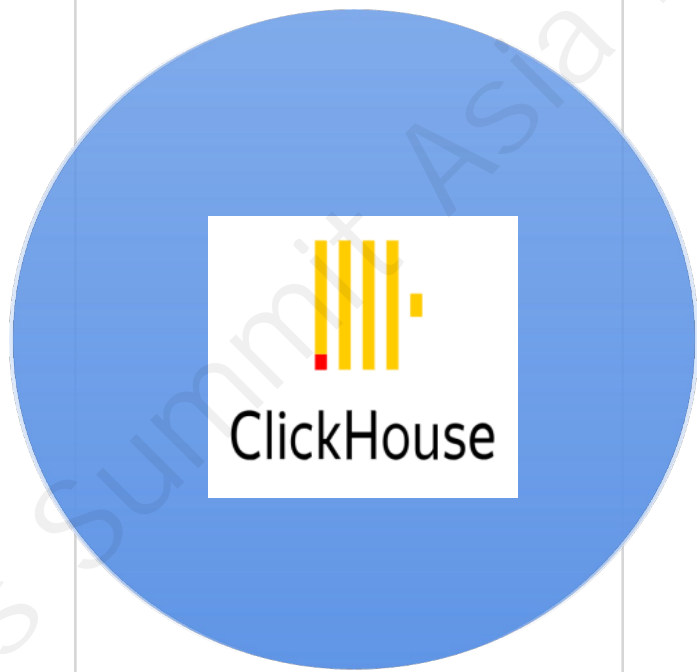
- 索引
- 物化视图的预计算、sql rewrite

## 数据可靠

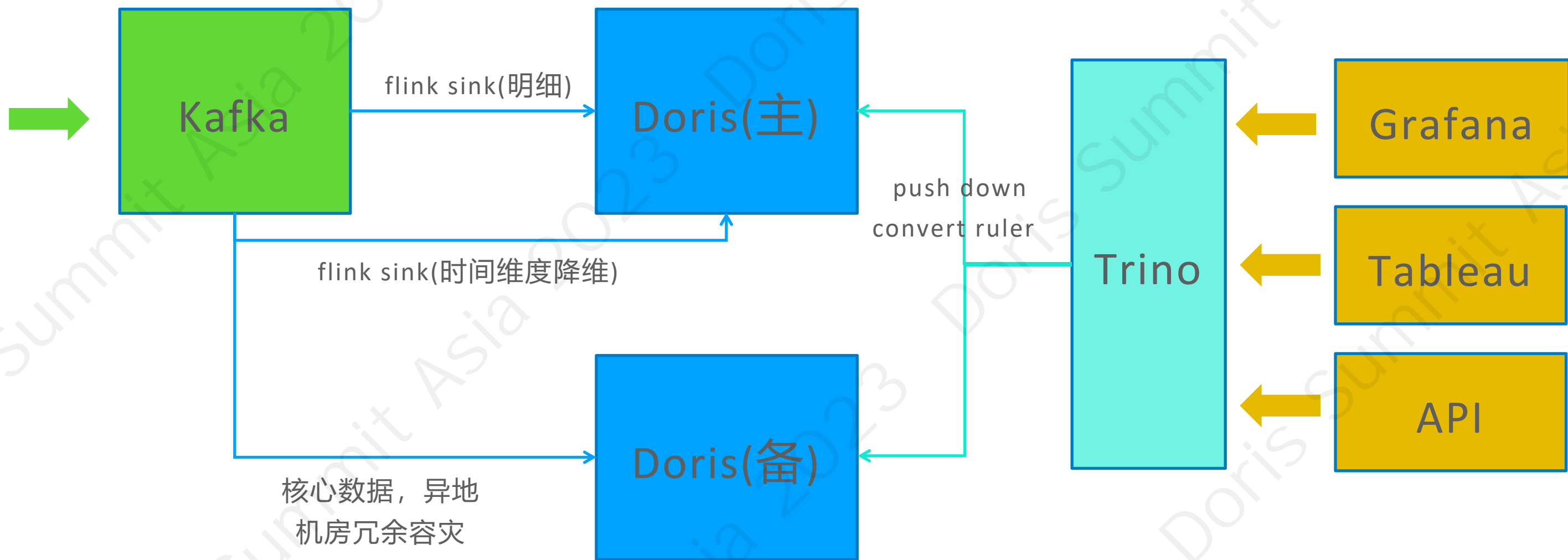
- 数据最终一致性
- 事务可见性

## 运维简单

- 便捷的扩缩容
- 集群跨机房
- 租户和场景的隔离



# 系统方案



## 核心特点

- **部署**: 主备集群, 机房容灾
- **写入**: Flink写入, 提前按需时间降维 (不使用Doris物化视图)
- **查询**: trino接入, 向上屏蔽



落地情况

节点

200+

Table

5K

数据规模

5-7K 亿/天

实时

10秒-2分钟

查询量

50W 次

查询延时

<3秒 98%

# 3 关键技术实现

# 系统建设的核心挑战



01

## 高可靠保障

- 监控场景：分钟内延迟
- S赛事：峰值暴涨2-3倍

02

## 运营&治理

- 避免场景滥用
- 隔离性

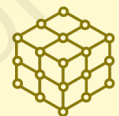
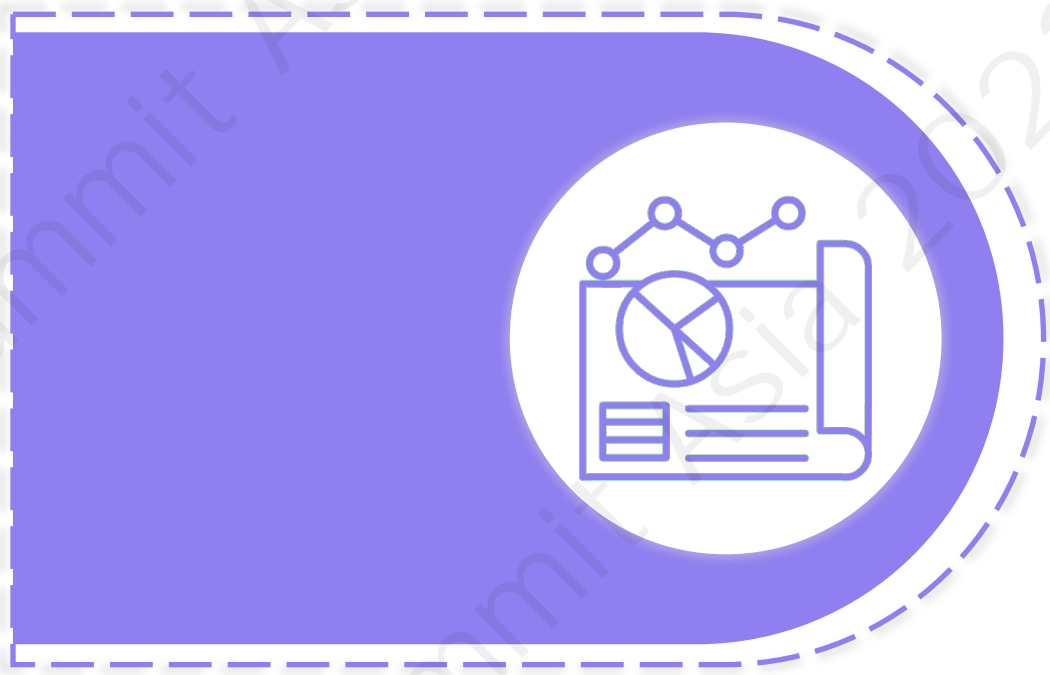
03

## 迁移

- 迁过来：历史包袱太重
- 牵出去：应对未来风险

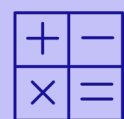


# 集群高可靠



## 容灾

- 多副本：保障存储和计算的可靠
- 机房级热备：核心数据按需冗余，查询路由复用备集群算力，提高备用集群资源利用率



## 赛事临时扩容

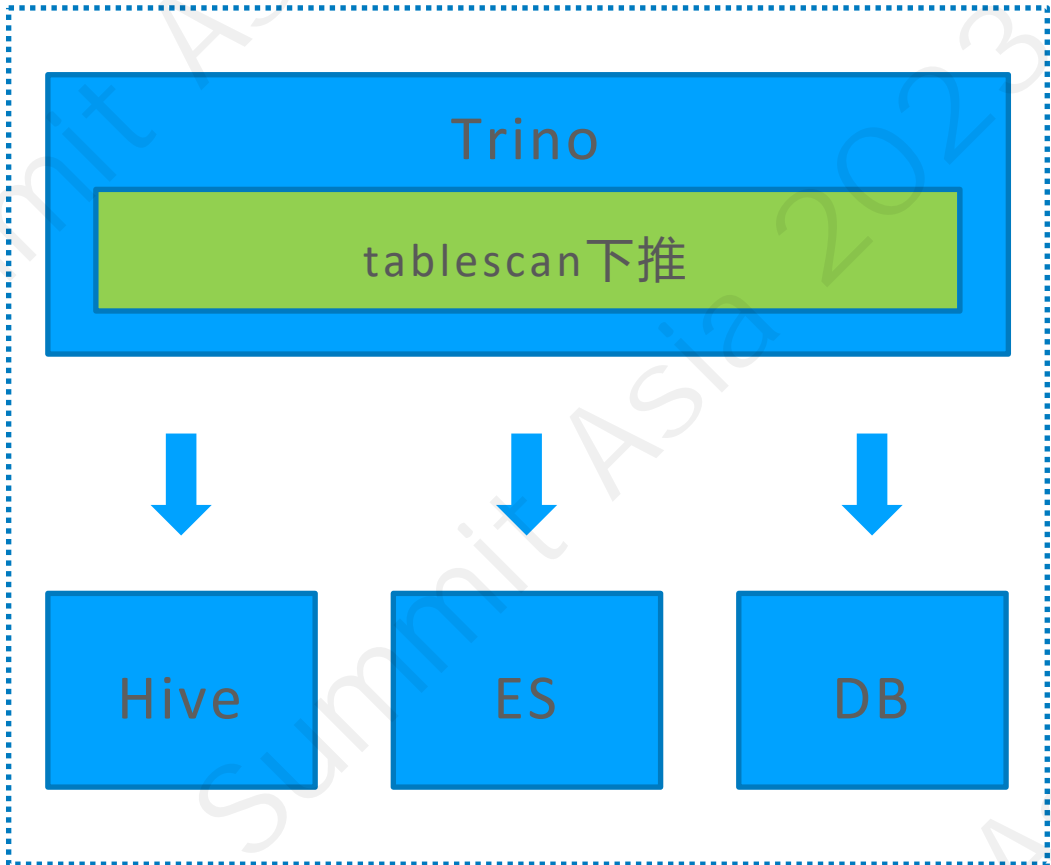
- 提前云上算力扩容，node group标注云上节点
- 预先table list，标记云上存储副本，云上分担计算压力
- 事后15%上限比例进行缩容，IO影响较大

查询

greenplum, kylin, impala, kudu, druid, clickhouse, doris/sr, 各种云。。。

对用户透明, OneSQL & OneOLAP? ? ?

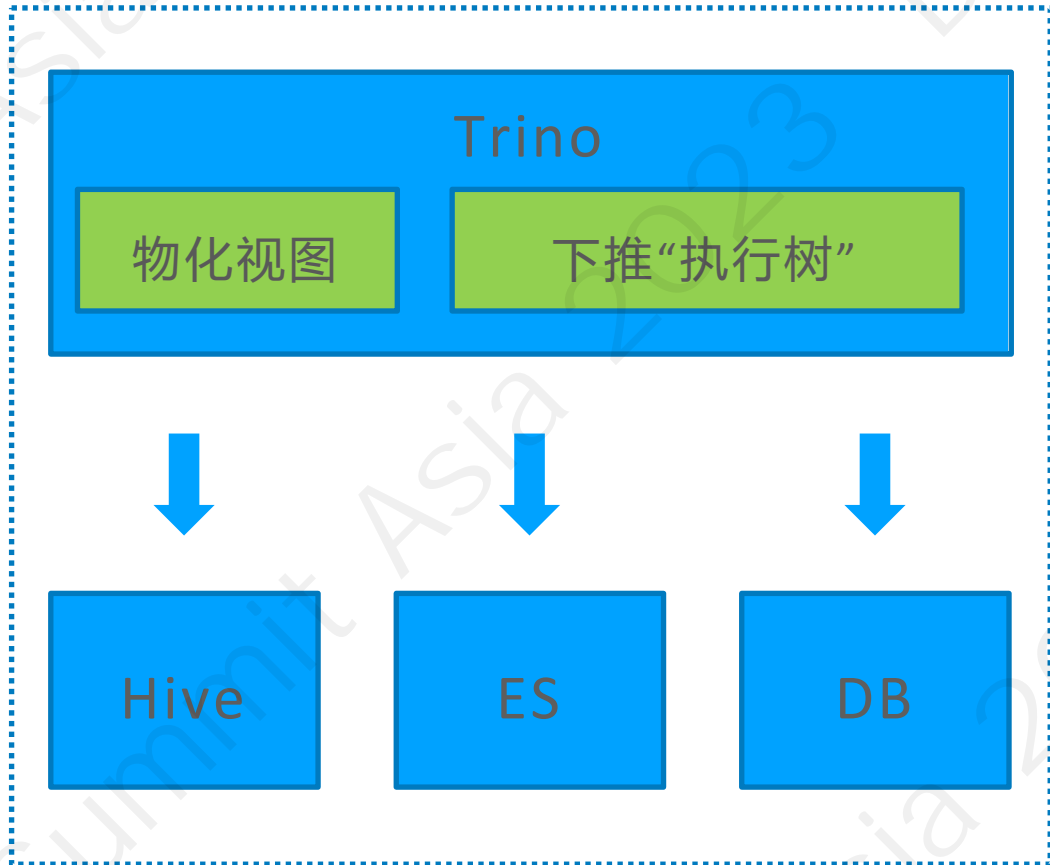
“敏捷”开发的几万SQL将来如何升级到其他方案? ? ? 无压力的引入新技术



集成"数据"

- OneSQL, 但性能较低
- Catalog不同, 存储位置对用户不透明

取长补短  
不重复造轮子

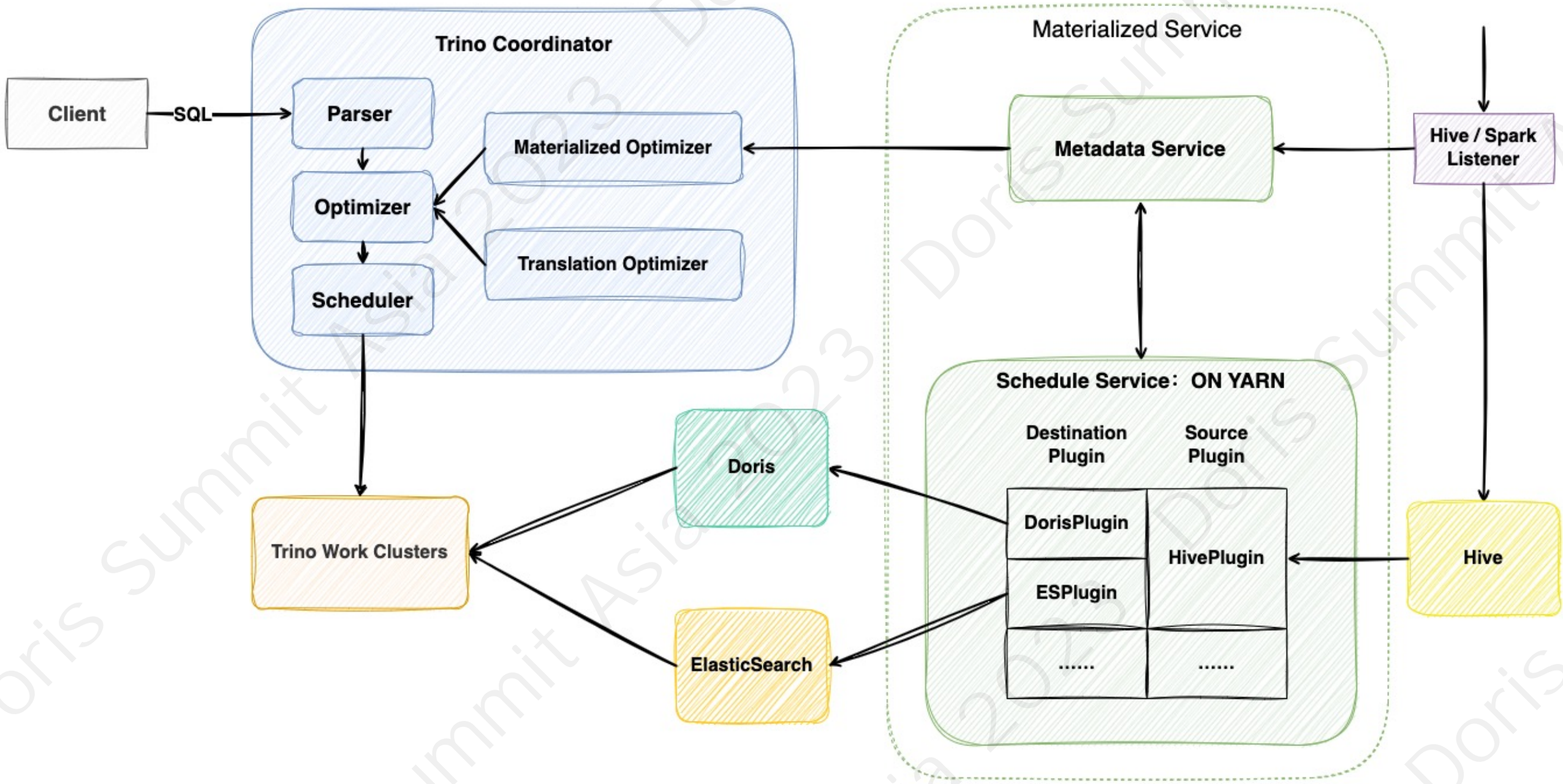


集成"计算"

- 逻辑元数据, 存储透明化
- 引擎各司其职, 性能有保障



查询



**物化：** mv根据场景可选择不同的存储引擎，如Doris、ES、Phoenix，“一张表”支撑各类场景，SQL Auto Rewrite

**Pushdown：** 将SQL分支“尽可能”的pushdown到下层OLAP引擎计算，保障计算性能

**跨源：** 和离线数仓、DB进行关联分析，不仅异构，还能异地

**数据湖：** 虚拟数据湖，集成数据分析更高效，成本更低

DTCC 2023  
第十四届中国数据库技术大会  
DATABASE TECHNOLOGY CONFERENCE CHINA

专场6	8月17日 下午		[ 数据库内核•原创探索 ]	
	时间	主持人	王广友	
	13:30-14:10	大数据异构数据库物化视图设计与落地	章逸群	广州虎牙信息科技有限公司

Trino Doris Connector



# Trino Doris Connector

```
trino> explain select platform,version,row_number() over(partition by platform order by cnt desc) _rn from (  
-> SELECT  
->   sum(1) cnt,  
->   platform,  
->   version  
-> FROM huya_ana_doris.bizer_default.hyrn_rn_load_time  
-> WHERE  
->   _ts>date_add('day',-1,now())  
->   and isext = '1'  
->   and platform = 'adr'  
-> GROUP BY platform,version  
-> )a;
```

单表查询，完全pushdown

untitled

trino> explain select url\_extract\_protocol(platform) protocol,sum(cnt) cnt  
-> from (  
-> SELECT platform,version ,count(1) cnt  
-> FROM huya\_ana\_doris.bizer\_default.hyrn\_rn\_load\_time  
-> WHERE \_ts>date\_add('day',-1,now())  
-> group by platform,version  
-> )  
-> GROUP by url\_extract\_protocol(platform);

Fragment 0 [SINGLE]  
Output layout: [url\_extract\_protocol\$gid, sum]  
Output partitioning: SINGLE []

Fragment 0 [SINGLE]  
Output layout: [\_pfnrtd, \_pfnrtd\_4, \_pfnrtd\_3]  
Output partitioning: SINGLE []  
Stage Execution Strategy: UNGROUPED\_EXECUTION  
Output[day, cnt, cnt]  
Layout: [\_pfnrtd:timestamp(0), \_pfnrtd\_4:bigint, \_pfnrtd\_3:bigint]  
Estimates: {rows: ? (?), cpu: ?, memory: ?, network: ?}  
day := \_pfnrtd  
cnt := \_pfnrtd\_4  
cnt := \_pfnrtd\_3  
RemoteSource[1]  
Layout: [\_pfnrtd:timestamp(0), \_pfnrtd\_4:bigint, \_pfnrtd\_3:bigint]

Fragment 1 [HASH]  
Output layout: [\_pfnrtd, \_pfnrtd\_4, \_pfnrtd\_3]  
Output partitioning: SINGLE []  
Stage Execution Strategy: UNGROUPED\_EXECUTION  
LeftJoin[["\_pfnrtd" = "expr\_2"]][hashvalue\_6]  
Layout: [\_pfnrtd:timestamp(0), \_pfnrtd\_4:bigint, \_pfnrtd\_3:bigint]  
Estimates: {rows: ? (?), cpu: ?, memory: ?, network: ?}  
Distribution: PARTITIONED  
RemoteSource[2]  
Layout: [\_pfnrtd:timestamp(0), \_pfnrtd\_4:bigint, \$hashvalue:bigint]  
LocalExchange[HASH][hashvalue\_6] ("expr\_2")  
Layout: [expr\_2:timestamp(0), \_pfnrtd\_3:bigint, \$hashvalue\_6:bigint]  
Estimates: {rows: ? (?), cpu: ?, memory: 0B, network: ?}  
RemoteSource[3]  
Layout: [expr\_2:timestamp(0), \_pfnrtd\_3:bigint, \$hashvalue\_7:bigint]

Fragment 2 [SOURCE]  
Output layout: [\_pfnrtd, \_pfnrtd\_4, \$hashvalue\_5]  
Output partitioning: HASH [\_pfnrtd][hashvalue\_5]  
Stage Execution Strategy: UNGROUPED\_EXECUTION  
ScanProject[table = huya\_ana\_doris:Query[SELECT '\_pfnrtd\_0', count(\*) AS '\_pfnrtd\_1' FROM (SELECT '\_ts', date\_trunc('\_ts', 'day') AS '\_pfnrtd\_0' FROM 'bizer\_default'.hyrn\_rn\_load\_time WHERE '\_ts' >= ?) GROUP BY '\_pfnrtd\_0']]  
Layout: [\_pfnrtd:timestamp(0), \_pfnrtd\_4:bigint, \$hashvalue\_5:bigint]  
Estimates: {rows: ? (?), cpu: ?, memory: 0B, network: 0B}/rows: ? (?), cpu: ?, memory: 0B, network: 0B}  
\$hashvalue\_5 := combine\_hash(bigint '0', COALESCE("\$operator\$hash\_code"("\_pfnrtd"), 0))  
\_pfnrtd := \_pfnrtd\_0:timestamp(0):DATETIME  
\_pfnrtd\_4 := \_pfnrtd\_1:bigint:bigint

Fragment 3 [SOURCE]  
Output layout: [expr\_2, \_pfnrtd\_3, \$hashvalue\_8]  
Output partitioning: HASH [expr\_2][hashvalue\_8]  
Stage Execution Strategy: UNGROUPED\_EXECUTION  
Project[]  
Layout: [expr\_2:timestamp(0), \_pfnrtd\_3:bigint, \$hashvalue\_8:bigint]  
Estimates: {rows: ? (?), cpu: ?, memory: 0B, network: 0B}  
\$hashvalue\_8 := combine\_hash(bigint '0', COALESCE("\$operator\$hash\_code"("expr\_2"), 0))  
ScanProject[table = huya\_ck:Query[SELECT "day", count(\*) AS "\_pfnrtd\_0" FROM "gamelive"."dis\_dws\_huya\_product\_dau" WHERE "day" = ? GROUP BY "day"] columns=[day:date:Date, \_pfnrtd\_0:bigint:bigint], grouped = false]  
Layout: [expr\_2:timestamp(0), \_pfnrtd\_3:bigint]  
Estimates: {rows: ? (?), cpu: ?, memory: 0B, network: 0B}/rows: ? (?), cpu: ?, memory: 0B, network: 0B  
expr\_2 := CAST("day" AS timestamp(0))  
\_pfnrtd\_3 := \_pfnrtd\_0:bigint:bigint

跨源查询，尽量分别pushdown到底层OLAP计算，最终再trino中汇总

## Pushdown:

- 算子：JOIN、UNION、TOPN、ORDER BY、AGGR、FILTER、WINDOW
- 函数：Trino内置函数、UDF、UDAF，逐步覆盖
- Exp：部分JSON Exp，Regular Exp，Lambda Exp，Date Exp
- Hint：可透传给doris

## 落地效果:

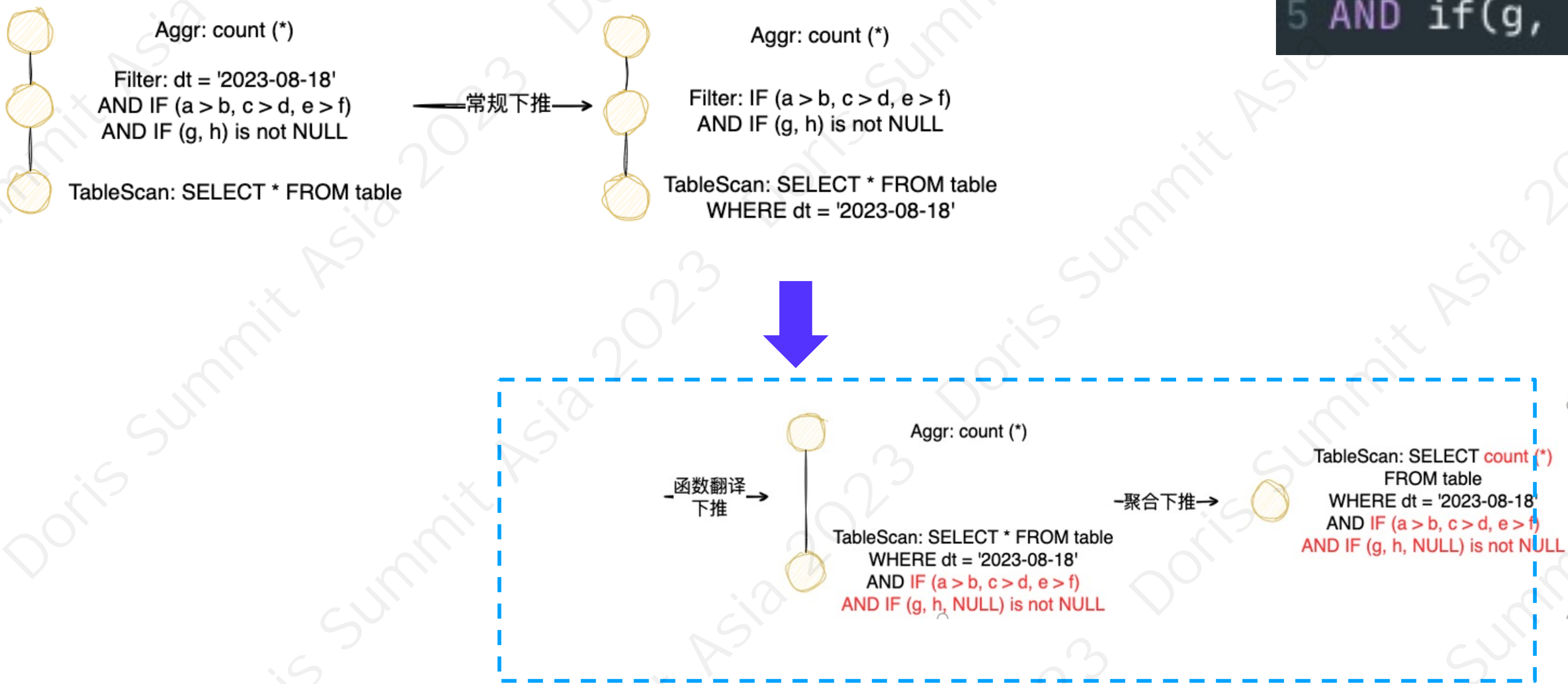
- 实际业务场景：查询次数50万每天，单次查询耗时增加20ms，SQL完全pushdown覆盖率99.5%
- TPC-H基准测试：与直接查询doris性能相同

# Trino Doris Connector

## Trino SQL 下推的瓶颈在哪里？

- 算子？ 支持
- 函数？ 不支持

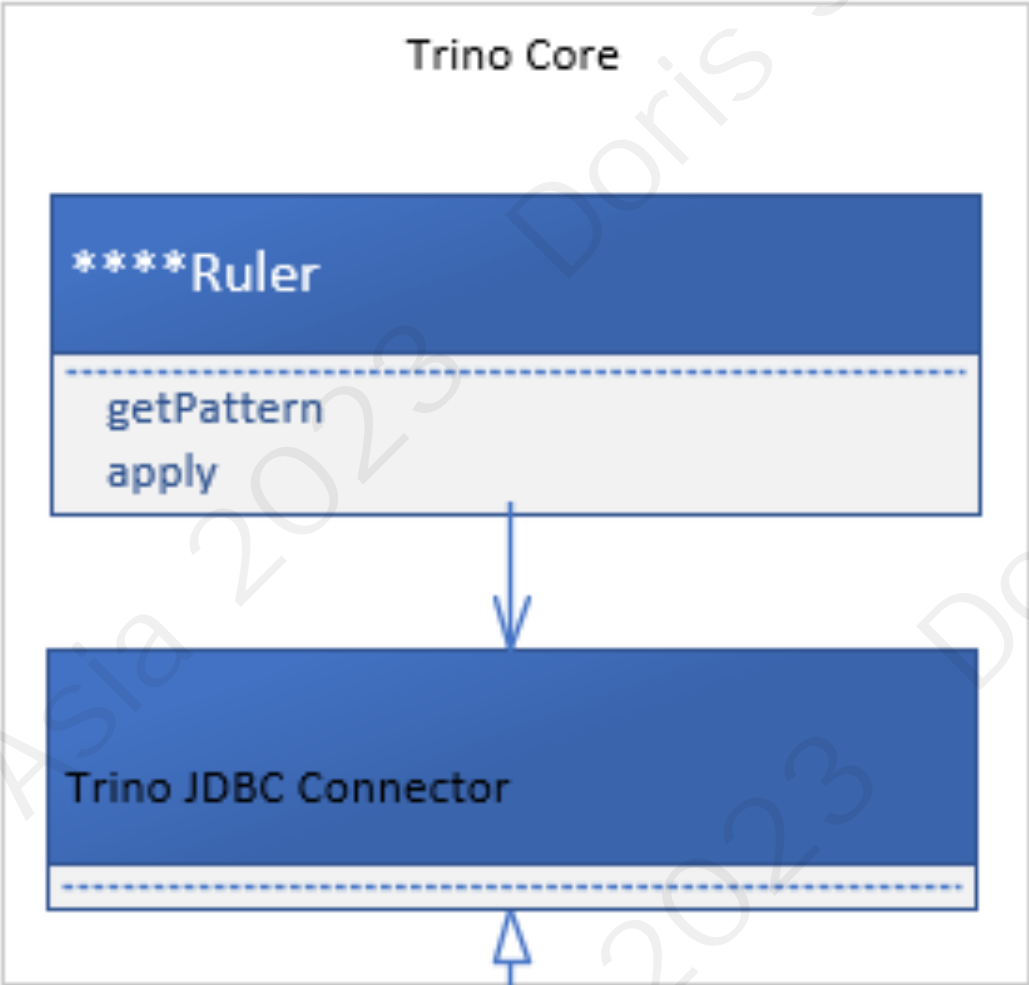
```
1 SELECT count(*)
2 FROM table
3 WHERE dt = '2023-08-18'
4 AND if(a > b, c > d, e > f)
5 AND if(g, h) is not NULL
```





# Trino Doris Connector

## 注册登记 Ruler

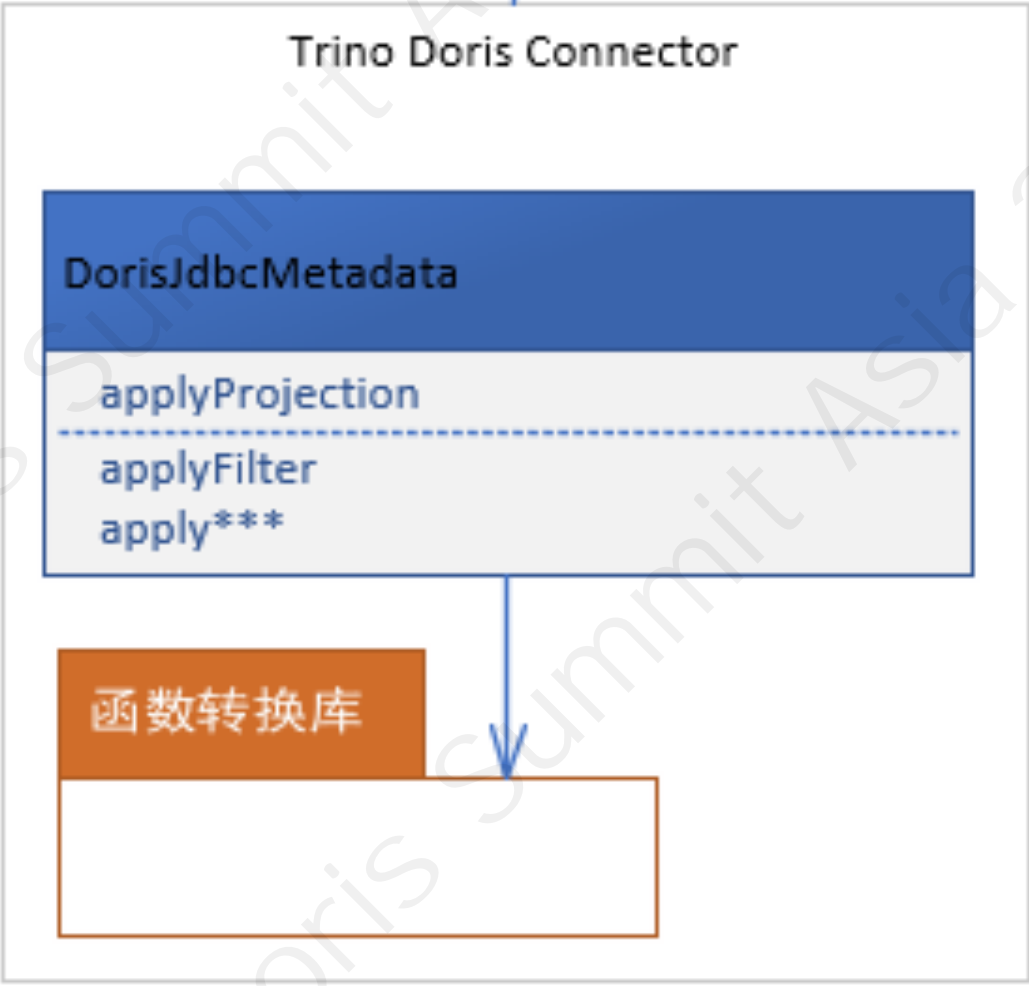


```
Set<Rule<?>> pushIntoTableScanRulesExceptJoins = ImmutableSet.<~>builder()
    .addAll(columnPruningRules)
    .addAll(projectionPushdownRules)
    .add(new PushProjectionIntoTableScan(plannerContext, typeAnalyzer, scalarStatsCalculator))
    .add(new RemoveRedundantIdentityProjections())
    .add(new PushLimitIntoTableScan(metadata))
    .add(new PushPredicateIntoTableScan(plannerContext, typeAnalyzer))
    .add(new PushPredicateAsSqlIntoTableScan(plannerContext, typeAnalyzer))
    .add(new PushSampleIntoTableScan(metadata))
    .add(new PushProjectionFunctionIntoTableScan(plannerContext, typeAnalyzer))
    .add(new PushWindowFunctionIntoTableScan(plannerContext, typeAnalyzer))
    .add(new PushAggregationIntoTableScan(plannerContext, typeAnalyzer))
    .add(new PushDistinctLimitIntoTableScan(plannerContext, typeAnalyzer))
    .add(new PushTopNIntoTableScan(metadata))
    .add(new PushTopNIntoProjectAndTableScan(metadata))
    .add(new PushOrderByIntoTableScan(metadata))
    .add(new RewriteTableFunctionToTableScan(plannerContext))
    .build();
```

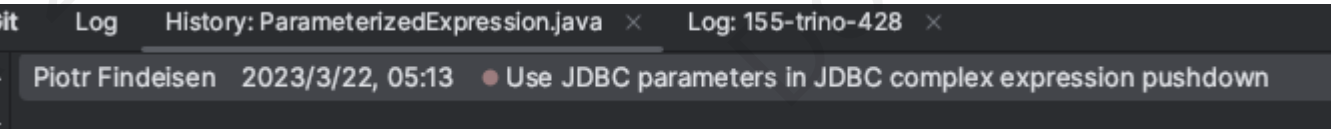
## 社区兼容性:

目前尚不兼容，但社区在往该方向发展

## 注册函数转换



```
public static final List<FunctionMappingGroup> FUNCTION_MAPPINGS =
    ImmutableList.<~>builder()
        .add(group(direct(TrinoFunctionResolves.ABS, DorisFunctionResolves.ABS)))
        .add(group(direct(TrinoFunctionResolves.ARRAY_CONSTRUCTOR, DorisFunctionResolves.ARRAY)))
        .add(group(direct(TrinoFunctionResolves.ARRAY_DISTINCT, DorisFunctionResolves.ARRAY_DISTINCT)))
        .add(group(direct(TrinoFunctionResolves.ARRAY_SORT, DorisFunctionResolves.ARRAY_SORT)))
        .add(group(direct(TrinoFunctionResolves.AVG, DorisFunctionResolves.AVG)))
        .add(group(direct(TrinoFunctionResolves.BITWISE_AND, DorisFunctionResolves.BIT_AND)))
        .add(group(direct(TrinoFunctionResolves.BITWISE_NOT, DorisFunctionResolves.BIT_NOT)))
        .add(group(direct(TrinoFunctionResolves.BITWISE_OR, DorisFunctionResolves.BIT_OR)))
        .add(group(direct(TrinoFunctionResolves.BITWISE_XOR, DorisFunctionResolves.BIT_XOR)))
        .add(group(BITWISE_LEFT_SHIFT_MAPPING))
        .add(group(BITWISE_RIGHT_SHIFT_MAPPING))
        .add(group(direct(TrinoFunctionResolves.COALESCE, DorisFunctionResolves.COALESCE)))
        .add(group(direct(TrinoFunctionResolves.CONCAT, DorisFunctionResolves.CONCAT)))
        .add(group(direct(TrinoFunctionResolves.COUNT, DorisFunctionResolves.COUNT)))
        .add(group(DATE_ADD_MAPPING))
        .add(group(DATE_MAPPING))
        .add(group(DATE_TRUNC_MAPPING))
        .add(group(direct(TrinoFunctionResolves.DATE_FORMAT, DorisFunctionResolves.DATE_FORMAT)))
        .add(group(direct(TrinoFunctionResolves.DAY_OF_WEEK, DorisFunctionResolves.DAY_OF_WEEK)))
        .add(group(direct(TrinoFunctionResolves.EXTRACT_URL_PARAMETER, DorisFunctionResolves.EXTRACT_URL_PARAMETER)))
        .add(group(direct(TrinoFunctionResolves.FLOOR, DorisFunctionResolves.FLOOR)))
        .add(group(FROM_BASE_MAPPING))
        .add(group(FORMAT_MAPPING))
```



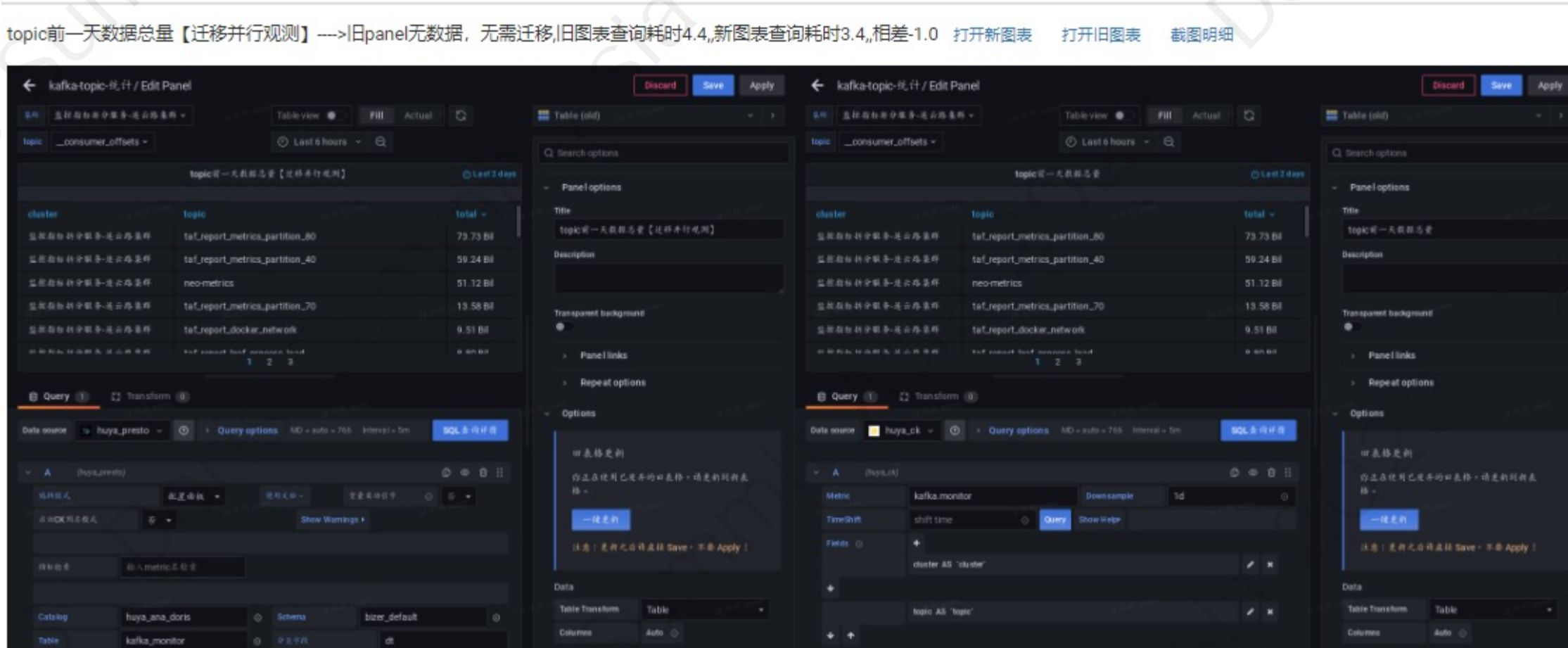
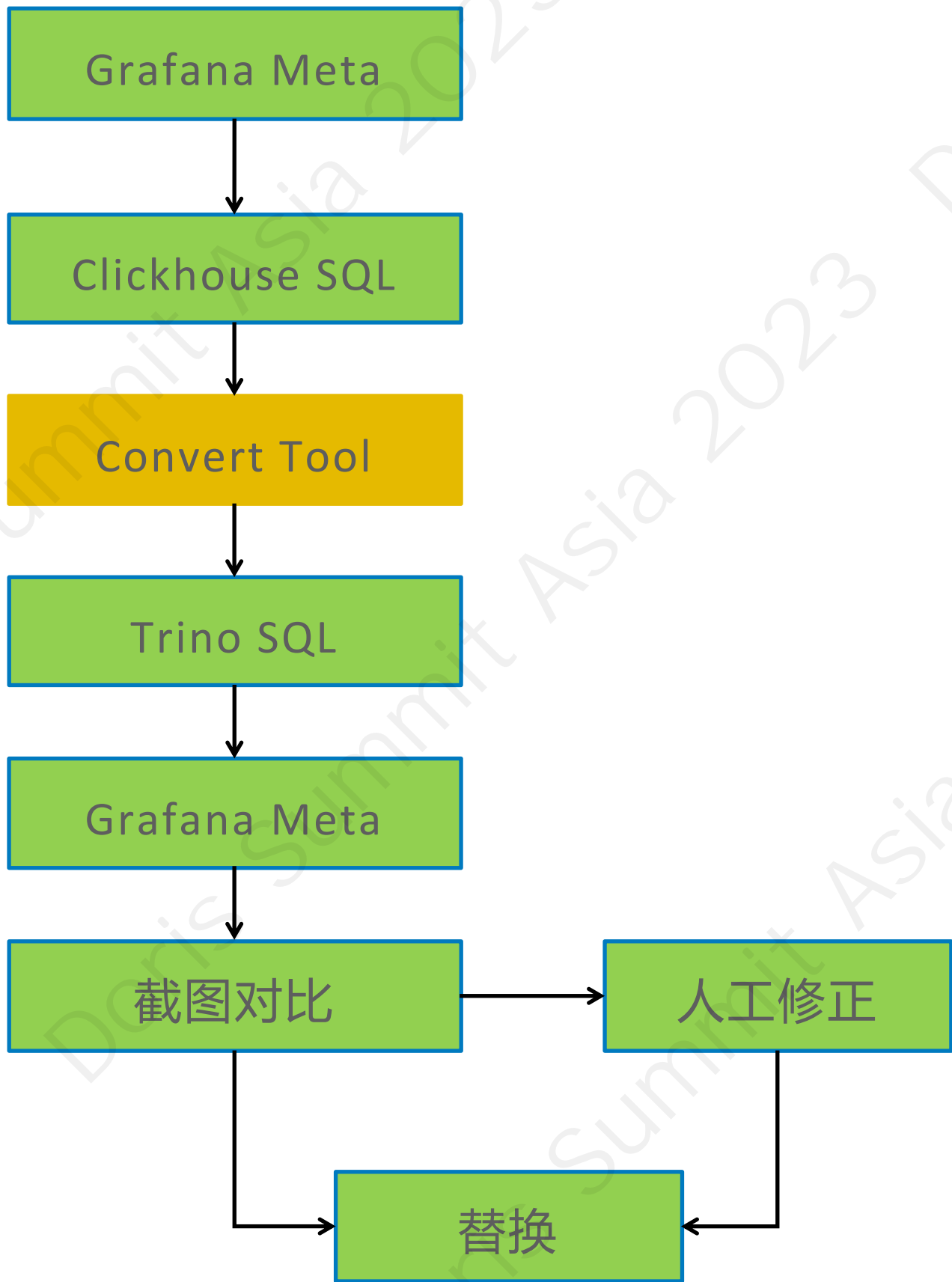


业务迁移

Grafana on Clickhouse  
(3万+图表)



Grafana on Trino



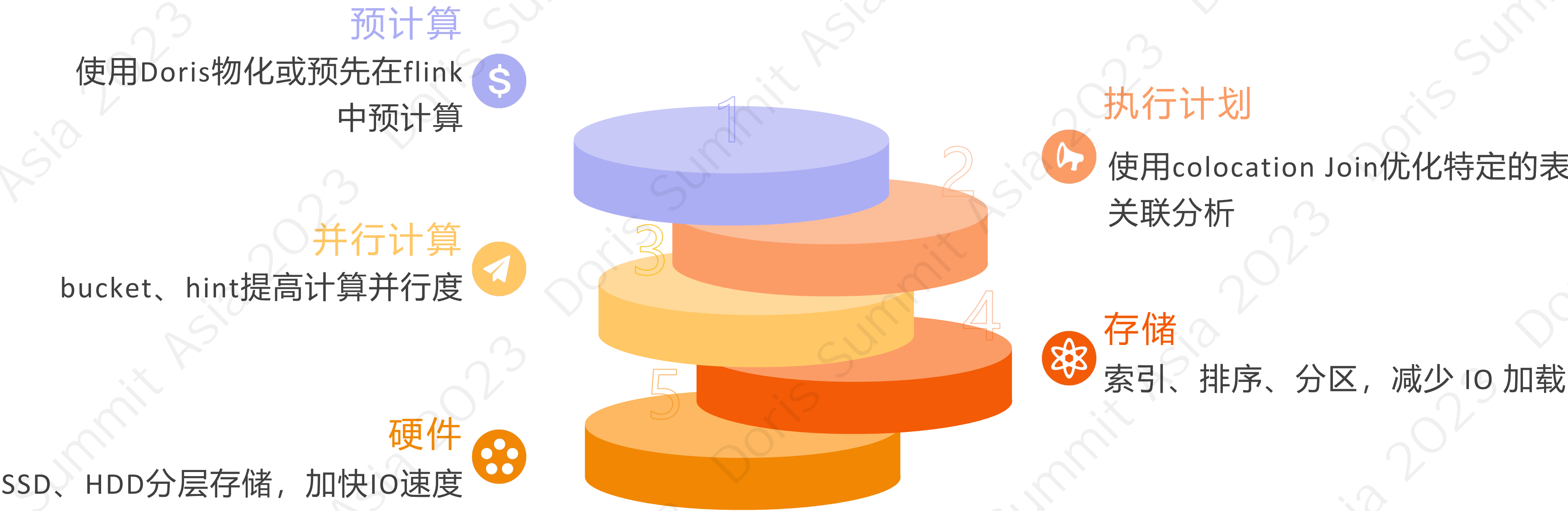
Trino SQL能力扩展:

- clickhouse常用函数: 如uniqif, sumif等
- 别名: where, group中可使用select中别名, 减少SQL代码量
- Any Join

```
select dt,
1 select dt,
2 (case when a=1 then 1 when a=2 then 2 else 0 end) type,
3 type type_2,
4 sum(if(b,1,0)) scount
5 from table
6 where type in (1,2)
7 group by dt type
```

耗时半年!!!

# Doris 查询优化



# 未来规划

## 2.0版本升级

- 计算资源队列，隔离监控查询和分析查询影响
- 全文检索，加速点查场景速度
- 存算分离，和虎牙离在线混部算力打通，提供更充裕的临时算力

## 物化视图

- 异步物化，替代目前flink预计算
- 数据建模方式的改变：后置建模





获取更多社区动态与最佳实践

### Apache Doris 官方平台:

- Apache Doris 官网: [doris.apache.org](https://doris.apache.org)
- Apache Doris GitHub: [github.com/apache/doris/](https://github.com/apache/doris/)

### 获取更多峰会资料:

- Doris Summit 峰会官网: [doris-summit.org.cn](https://doris-summit.org.cn)
- Doris Summit 峰会回放: <https://space.bilibili.com/1196172099/channel/collectiondetail?sid=1824324>