

# 美团AdHoc统一查询引擎实践

美团 吴彪

# InfoQ 企业会员

## 企业数字化传播一站式服务

InfoQ 企业会员是为满足企业在中国开发者群体中的品牌曝光需求而推出的一款矩阵化资源包。可为企业提供包括“企业号服务”、“企业动态宣发”、“品牌展示通道”在内的多项专属权益与服务，助力企业高效触达开发者群体，提升数字化时代影响力。



### 企业号服务

深度触达 300 万中高端开发者



### 企业动态宣发

新媒体矩阵覆盖百万粉丝



### 品牌展示通道

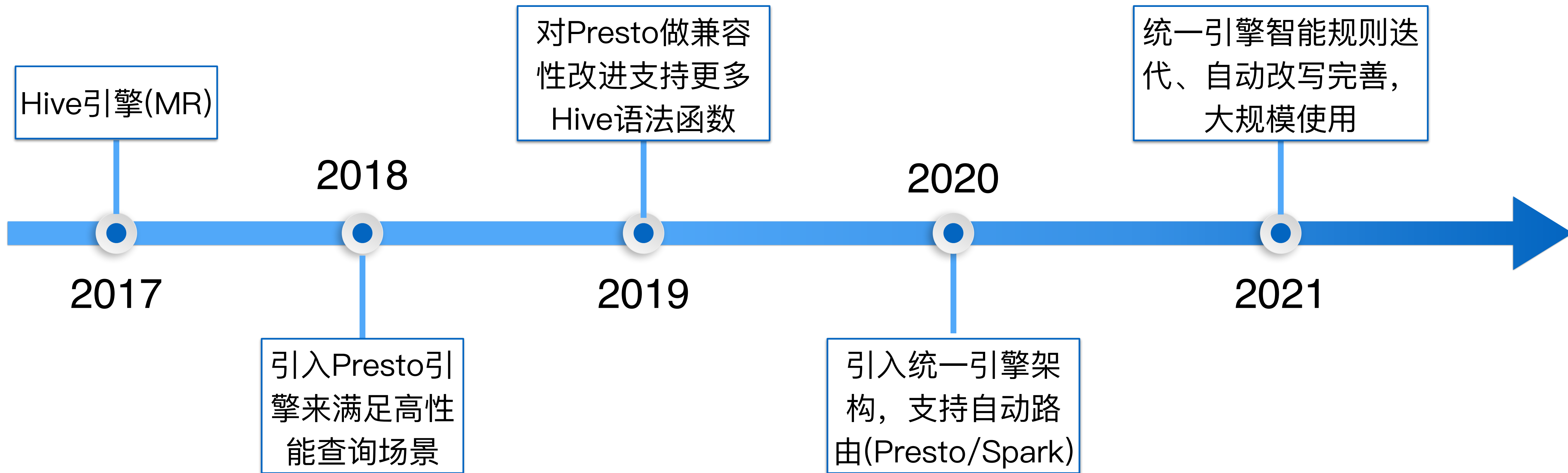
线上平台 10 万+ 流量曝光

# 目录

- 美团AdHoc查询引擎演进
- 多引擎架构的痛点
- 统一引擎实践
- 展望未来



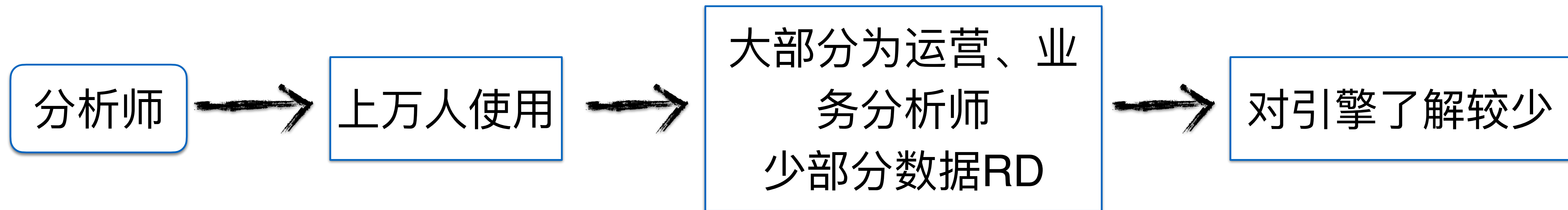
# 美团AdHoc引擎演进



# 应用场景

计算引擎	特点	应用场景	查询使用占比
Hive	稳定性高 性能差	ETL生产 查询	54%
Presto	稳定性差 高性能	查询	46%

# 分析师群体



# 多引擎存在的问题

分析师难以判  
断用什么引擎

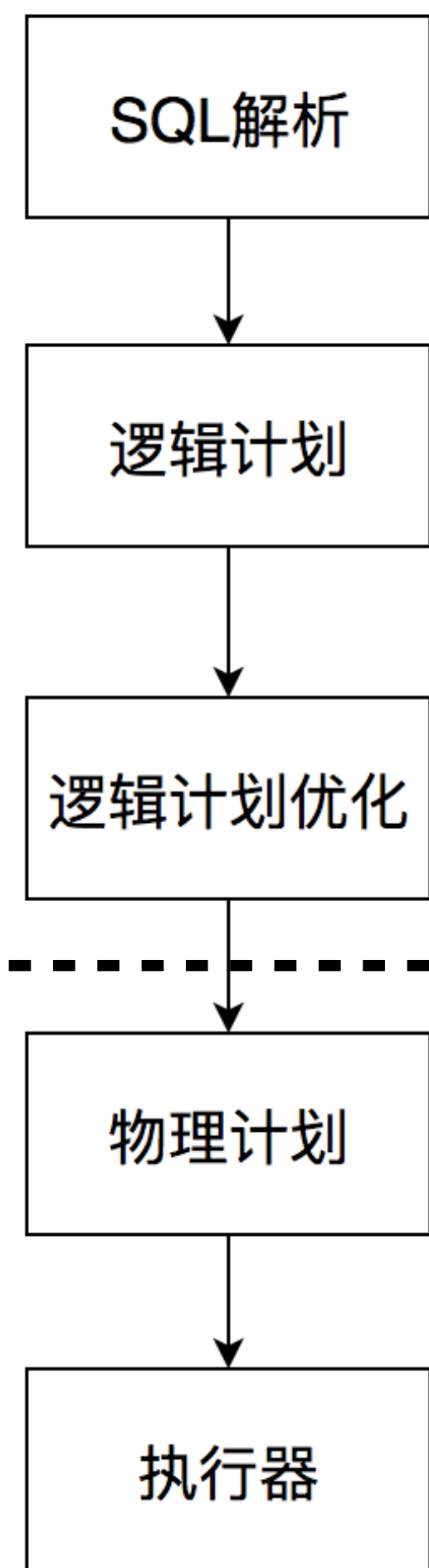
排查问题思路  
不同

多引擎架构

引擎间语法、  
函数不兼容

引擎迭代升级  
成本高

# 统一引擎架构设计



解析层

执行层

## 优点

- 用户接口统一
- 各个引擎实现思路类似

- 降低引擎数，减少维护成本
- 排查问题、日志统一

## 缺点

- 与引擎解析存在重叠
- 还需要维护多个引擎

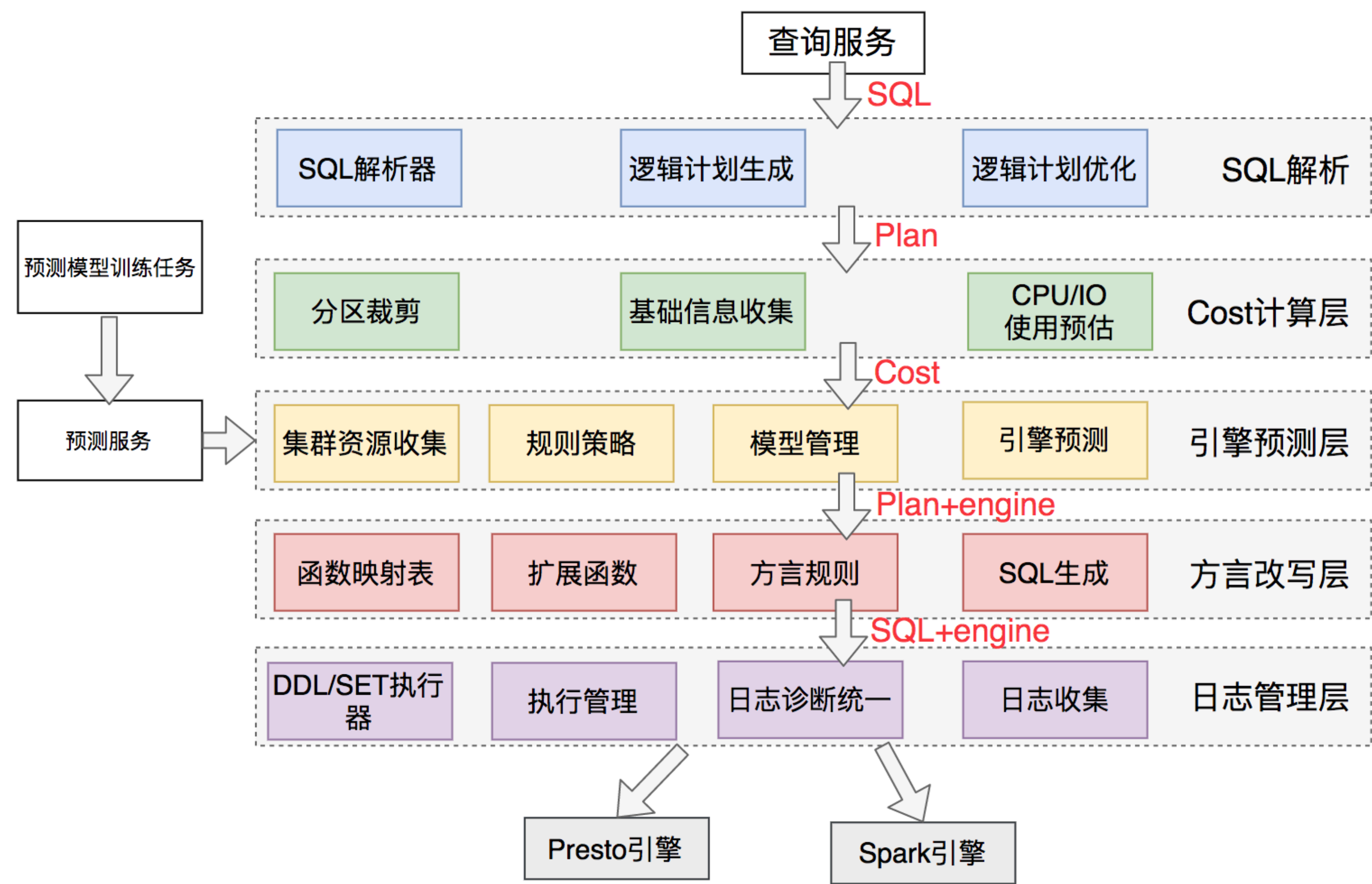
- 存量查询还需要解决解析兼容性
- 架构设计差异大

## 权衡

- 对用户收益
  - 统一解析层可达到统一引擎目标
- 模块依赖
  - 执行层前必定有解析层，面临同样问题
- 实现风险
  - MPP/DAG架构整合风险高



# 统一引擎整体架构



兼容存量SQL语法/函数

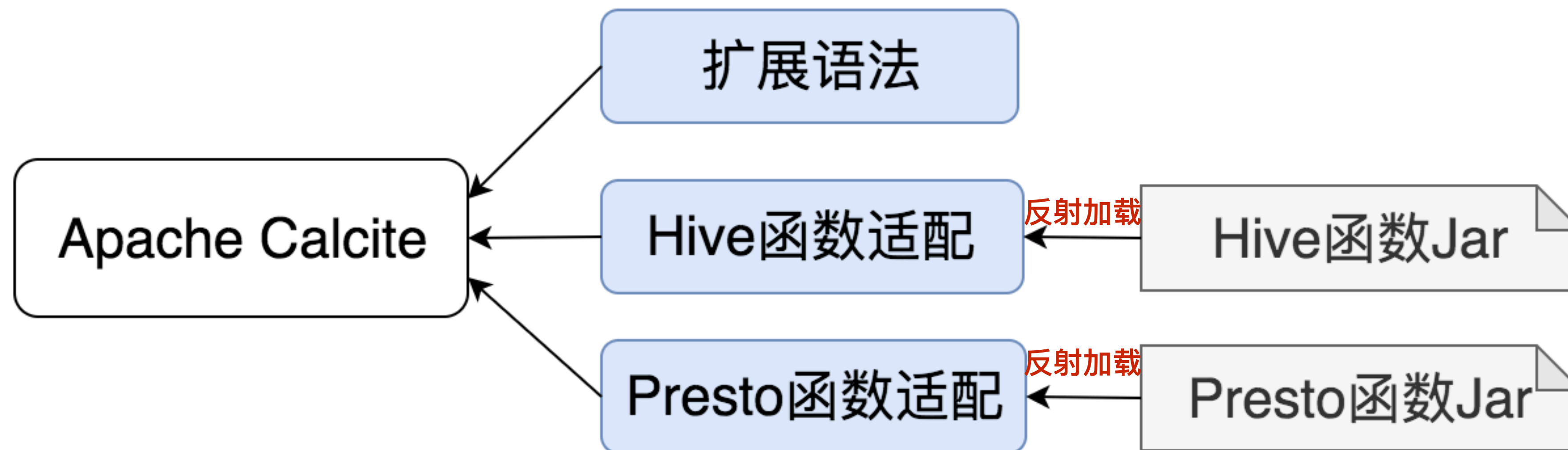
智能计算适合引擎

解决引擎间语法兼容性

解决问题分析不统一

# 解析层实现

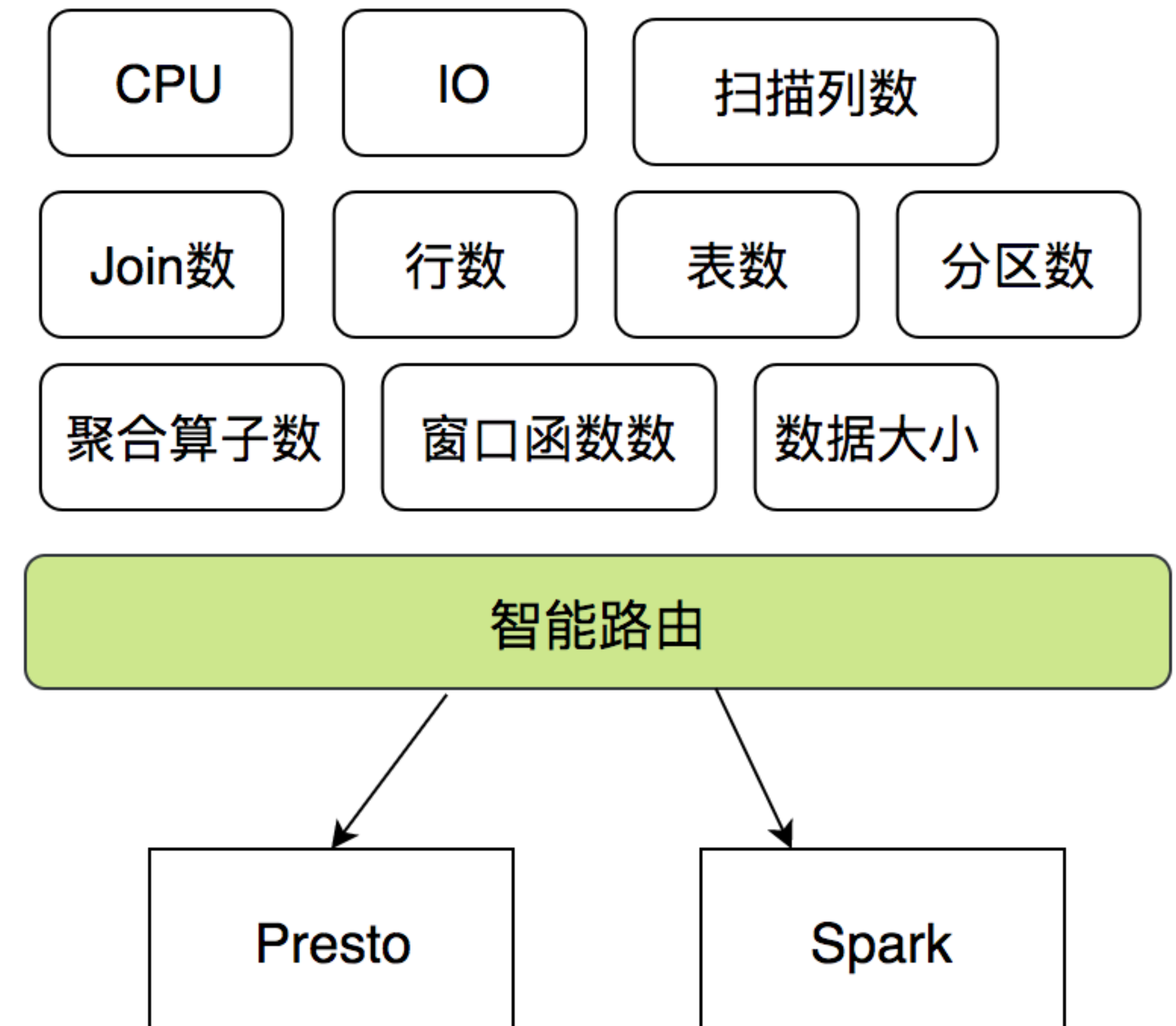
- 核心问题：兼容存量Presto/Spark语法的SQL



# 引擎预测

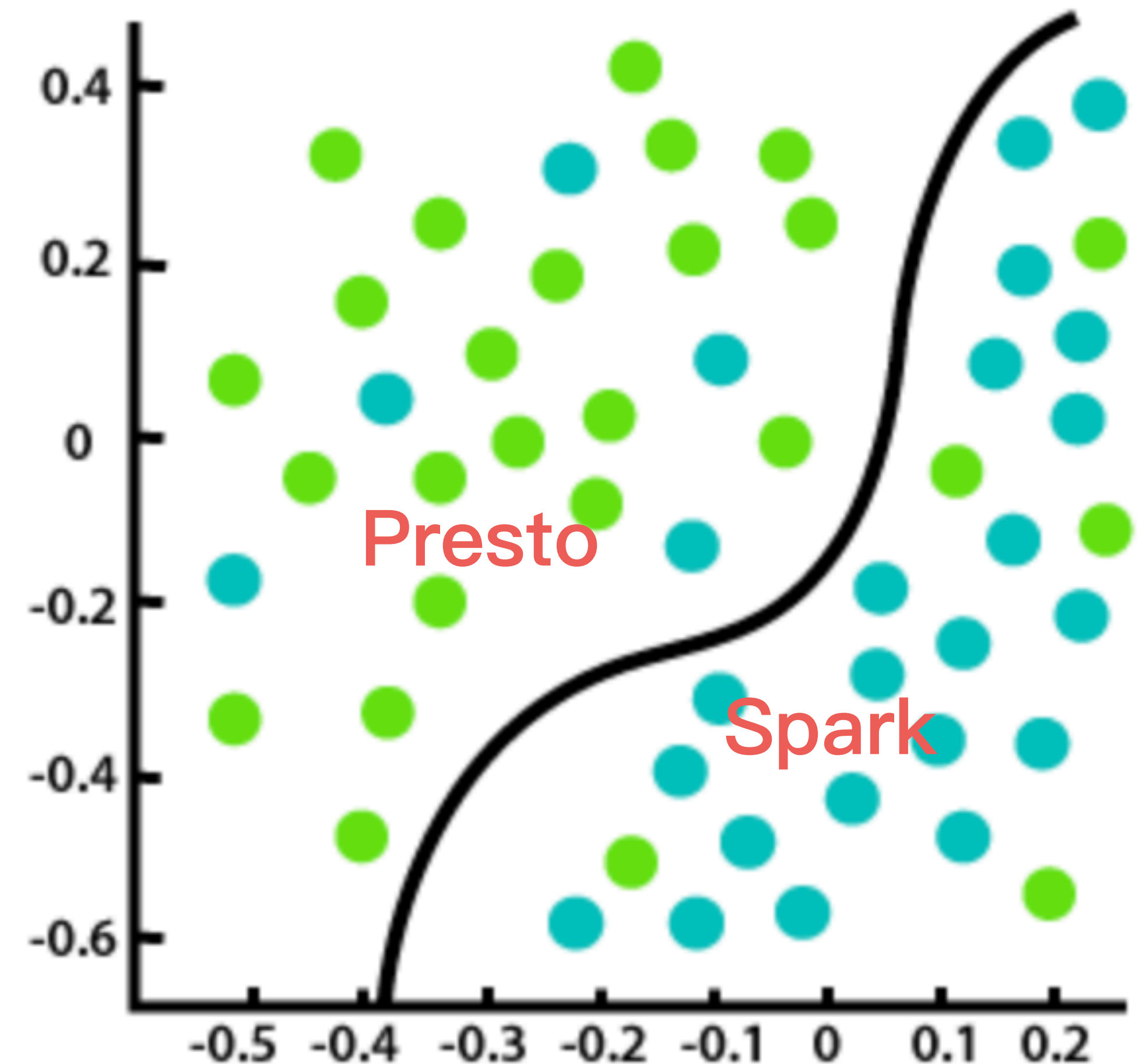
- 选择引擎方法
  - 引擎参数限制(分区数、数据量)
  - 一些经验(单表过滤、小数据量计算)
  - 集群负载

以上条件外的查询怎么办?



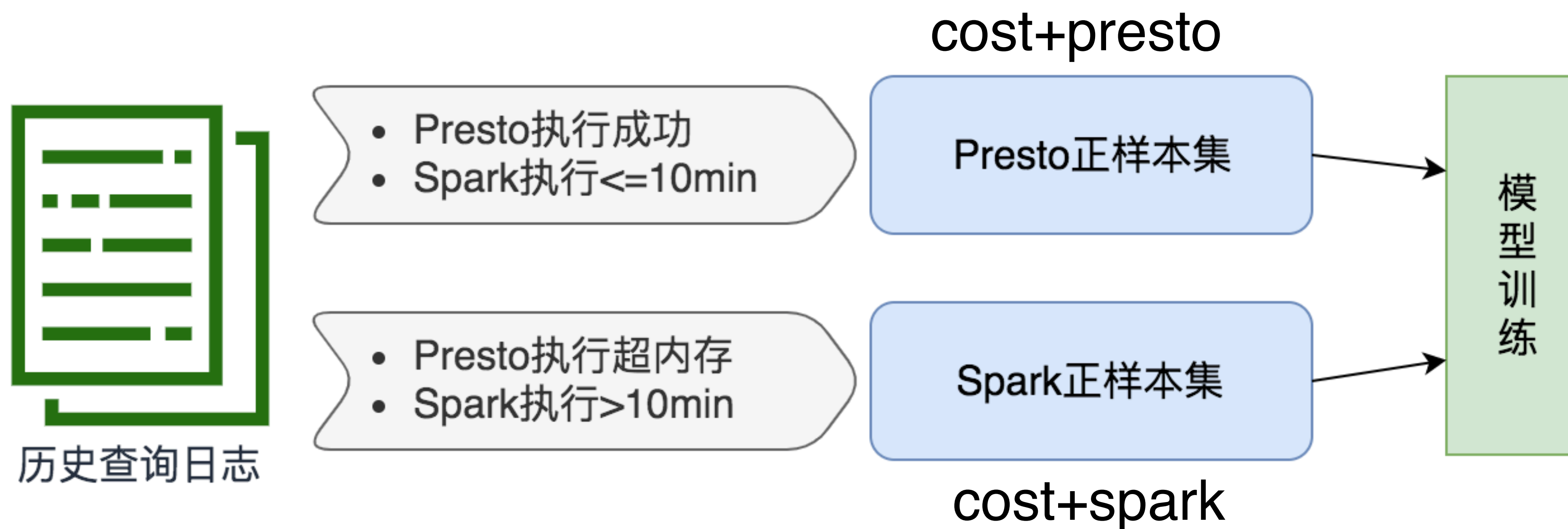
# 引擎预测

- 不同复杂度SQL适合那个引擎：  
分类问题
- 机器学习领域常见分类算法：  
决策树、逻辑回归等
- 训练集如何构造？  
历史数据无法知道那个引擎最优





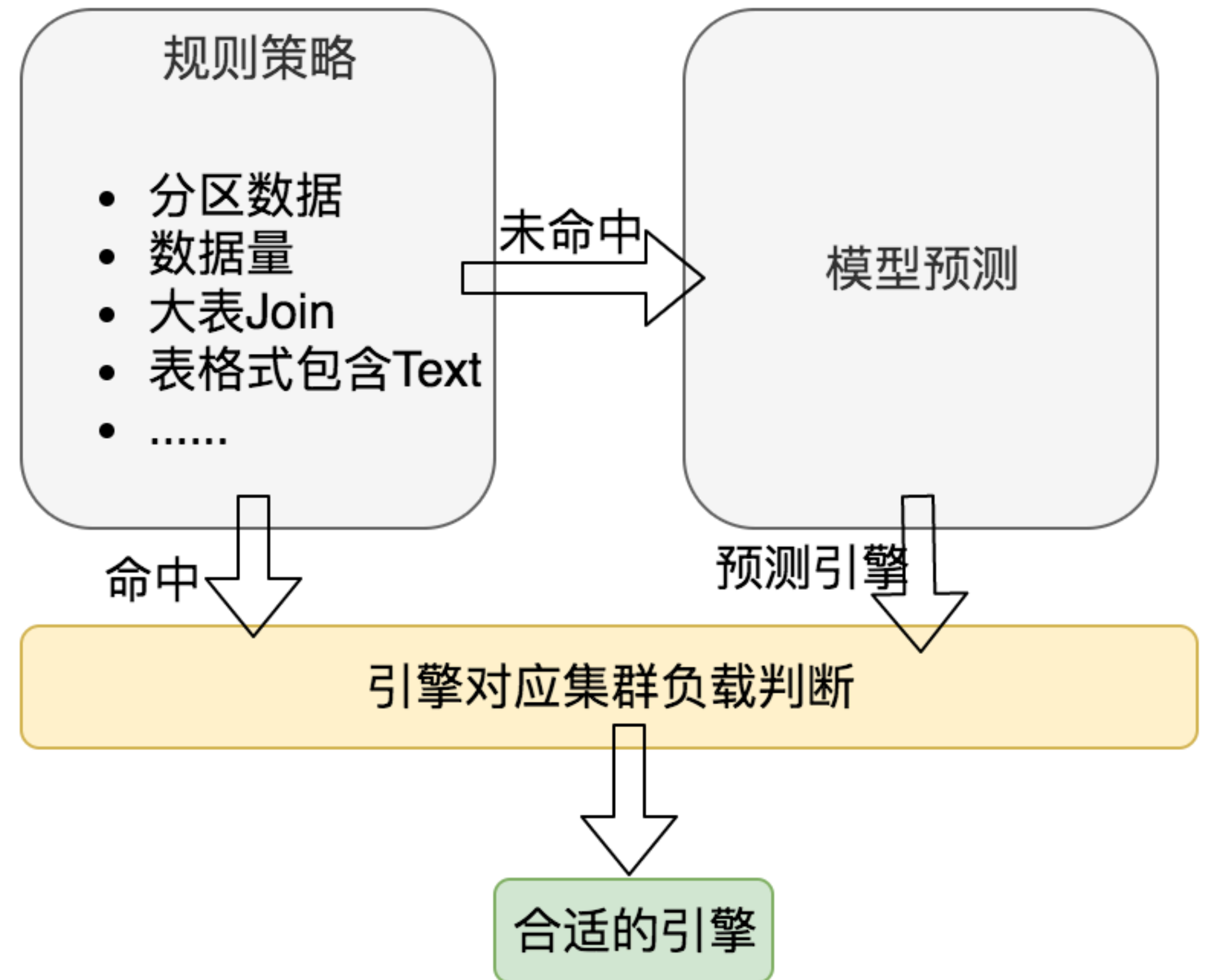
# 训练集构建





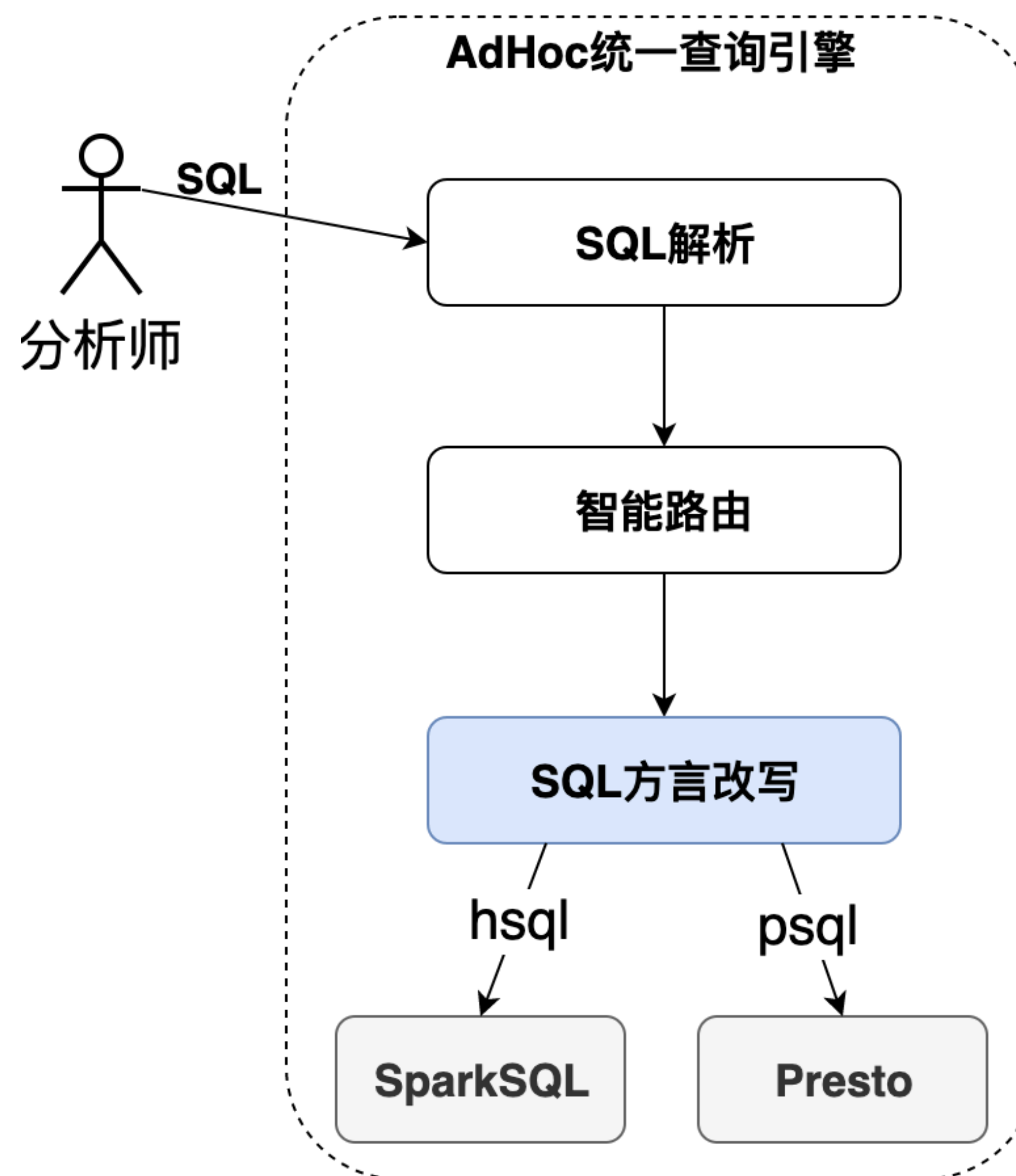
# 引擎预测

- 结合规则策略和模型预测来覆盖所有查询
- 集群负载策略避免资源太高影响性能
- Presto执行失败可自动failover到spark引擎



# 方言改写

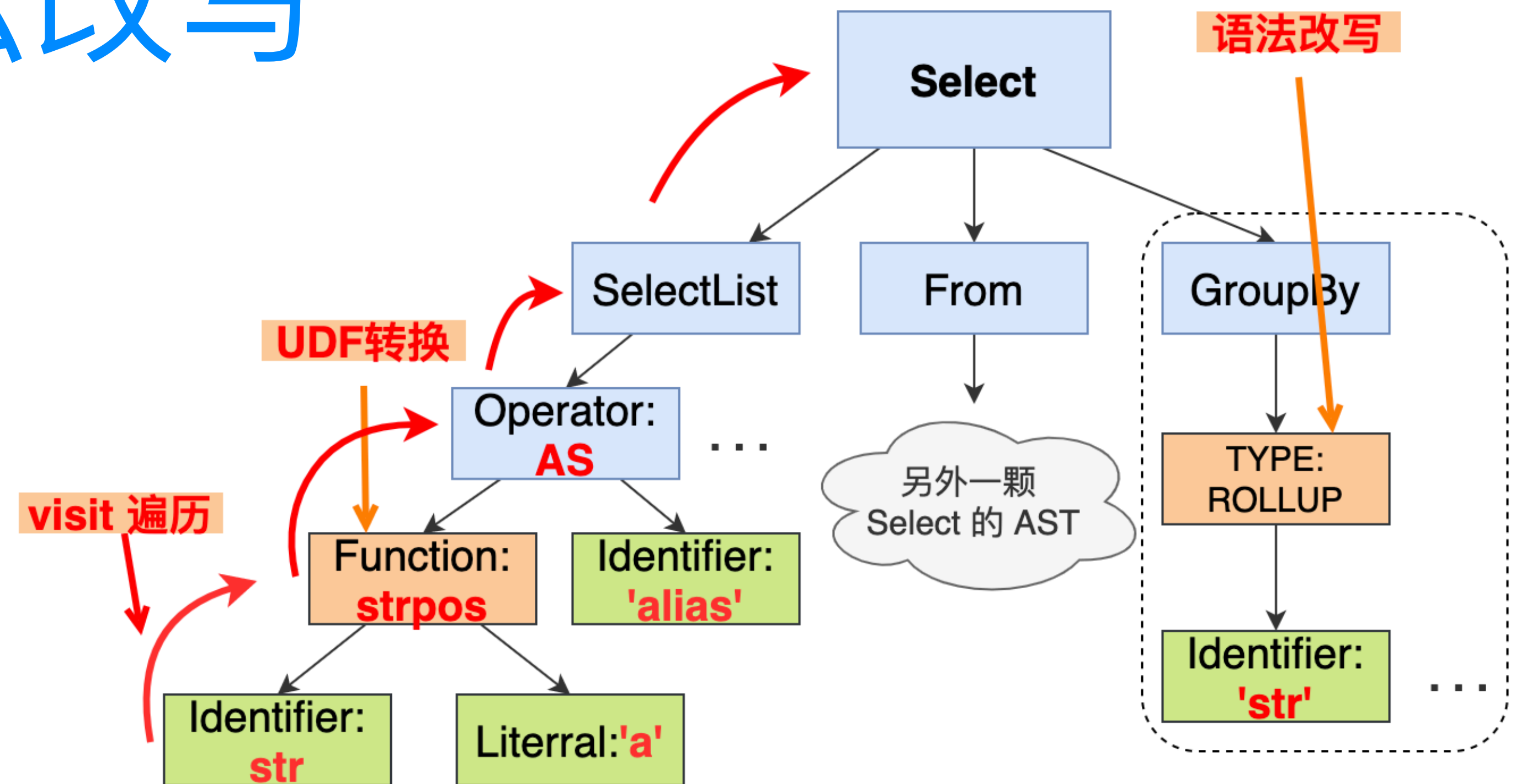
- 问题
  - 语法存在差异
  - 引擎间函数不统一、接口不兼容
  - Spark有大量存量用户开发的UDF



# 方言改写—语法改写

- 翻译

SQL → AST → 方言SQL



```
select id,  
       name,  
       sum(price)  
from test  
group by cube(id,name)
```

方言改写

```
select id,  
       name,  
       sum(price)  
from test  
group by id,name with cube
```

SparkSQL

# 方言改写—函数

- 对于不同名、不同参数位置的函数，可通过映射表解决

```
other.to.sparksql.contains=array_contains  
other.to.sparksql.date_format=presto_date_format  
other.to.sparksql.format_datetime=presto_date_format  
other.to.sparksql.approx_percentile=percentile_approx  
other.to.sparksql.array_sort=sort_array  
other.to.sparksql.strpos(0,1)=locate(1,0)  
other.to.sparksql.array_agg=collect_list
```

函数映射表



```
select strpos('acdefg','b')
```

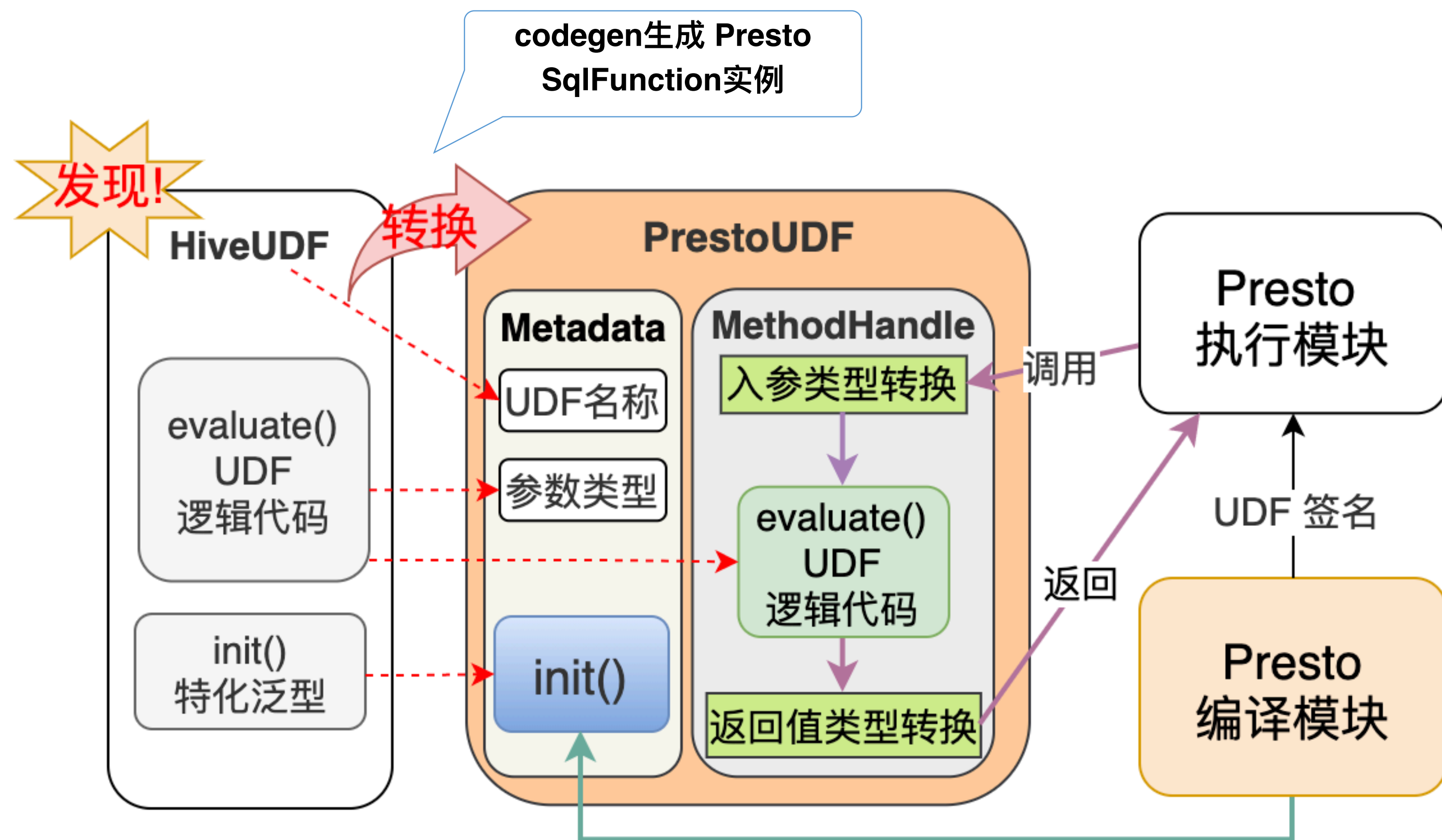
↓ 函数映射

```
select locate('b','acdefg')
```



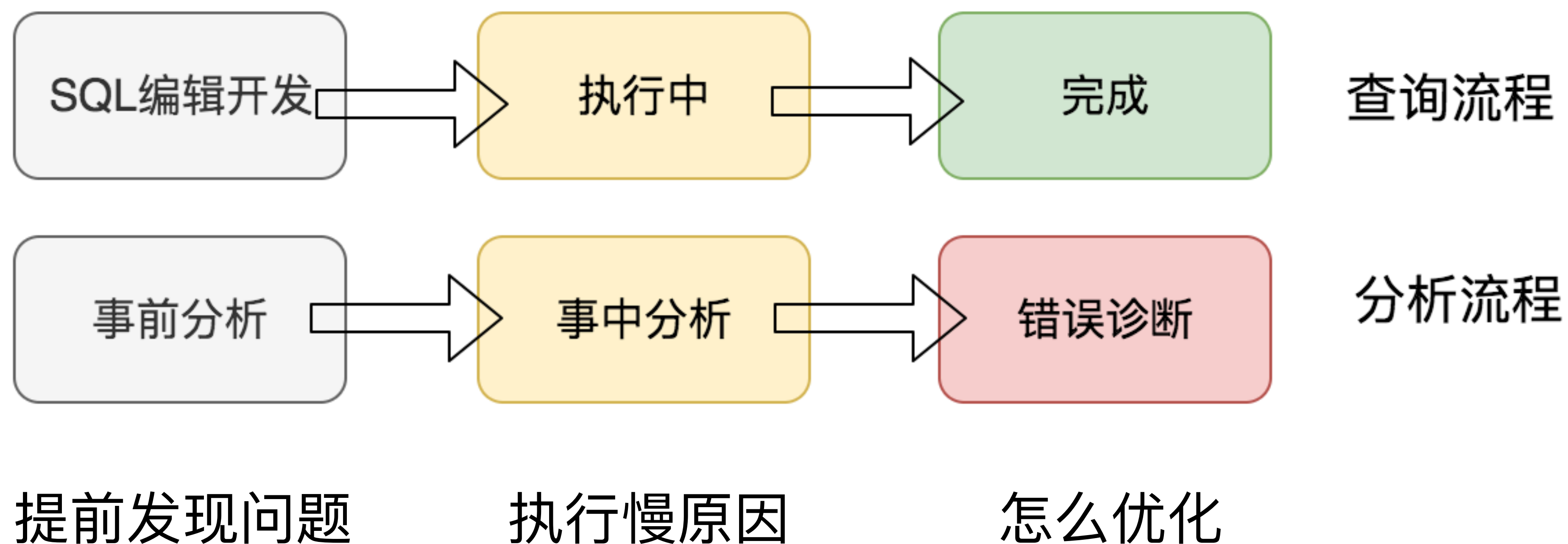
# 方言改写—函数

- Presto HiveUDF适配器支持大量存量SparkUDF



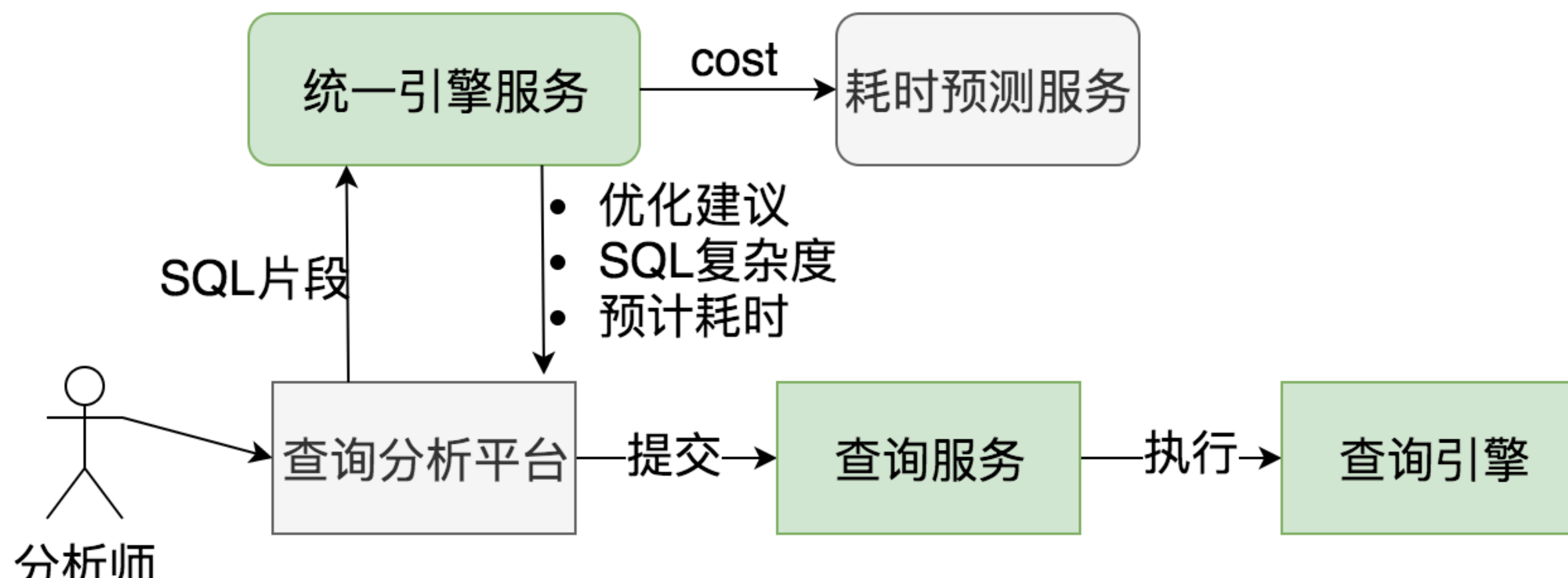


# 异常诊断



# 异常诊断—事前分析

- 语法检查
- 优化建议
- 预估耗时



## ! 【慢查询警告】：

查询扫描总分区数大于1000, 请限制表分区或对数据分开进行查询 相关数据表: hiv

查询join大表, 扫描总数据超过1TB, 建议减少数据量 相关数据表: hi

建议添加分区过滤条件 相关数据表: hi

presto引擎修改建议: sql中包含count(distinct field), 该语句执行较慢。如果业务逻辑允许非精确, 建议使用approx\_distinct替换, 加快查询速度

查询表中存在text格式表, 且查询数据量较大, 可能导致查询缓慢。建议将text格式表换成ORC格式或通过缩小分区范围来减少扫描数据量

相同表多次查询, 建议合并成一个子查询, 提升查询速度

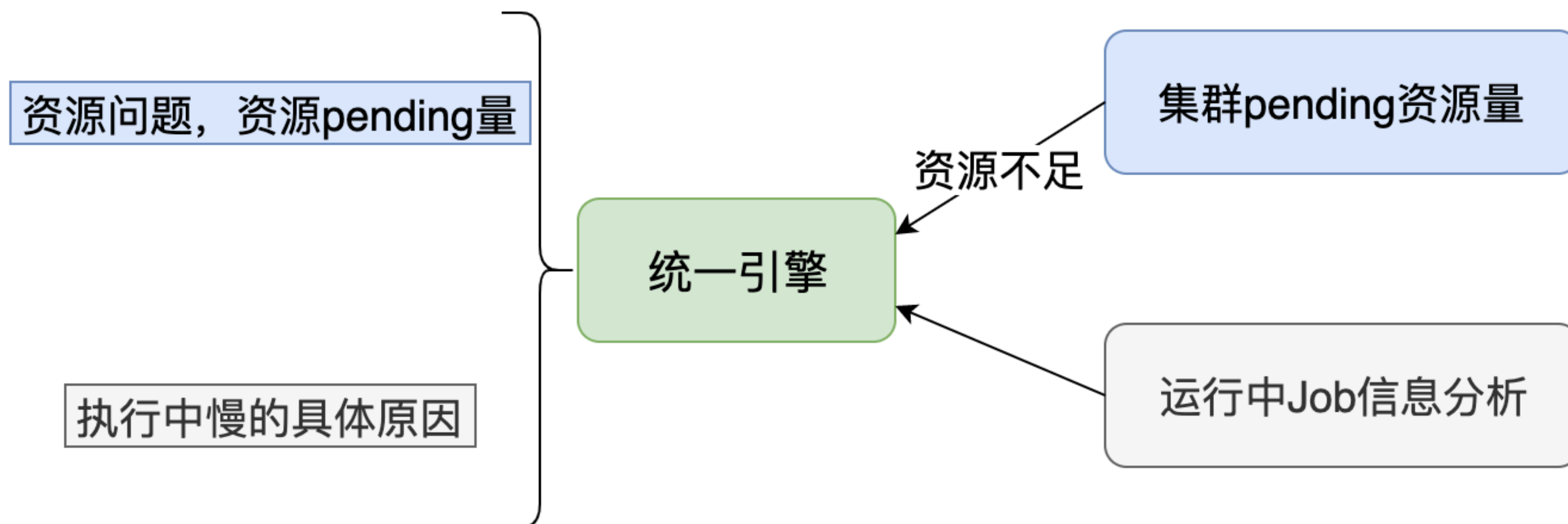
该条提示对您有帮助吗?

有帮助

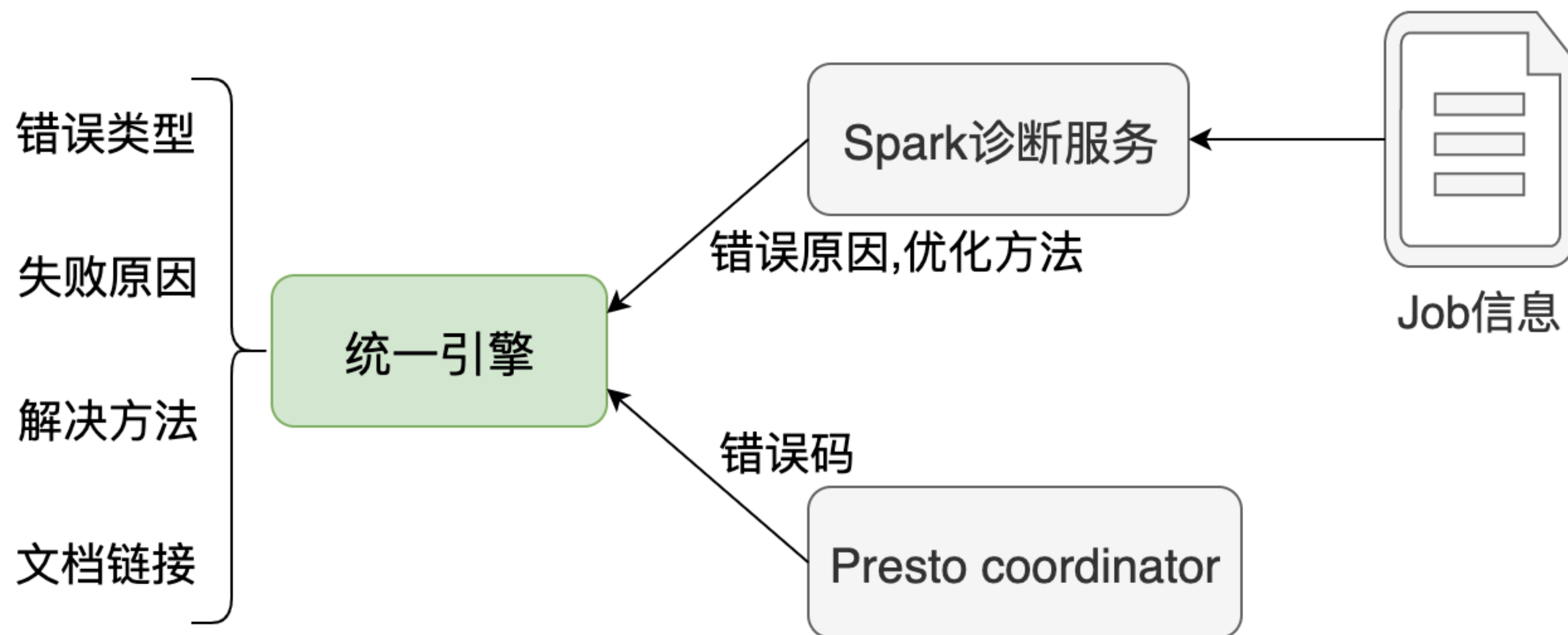
无帮助

# 异常诊断—事中分析

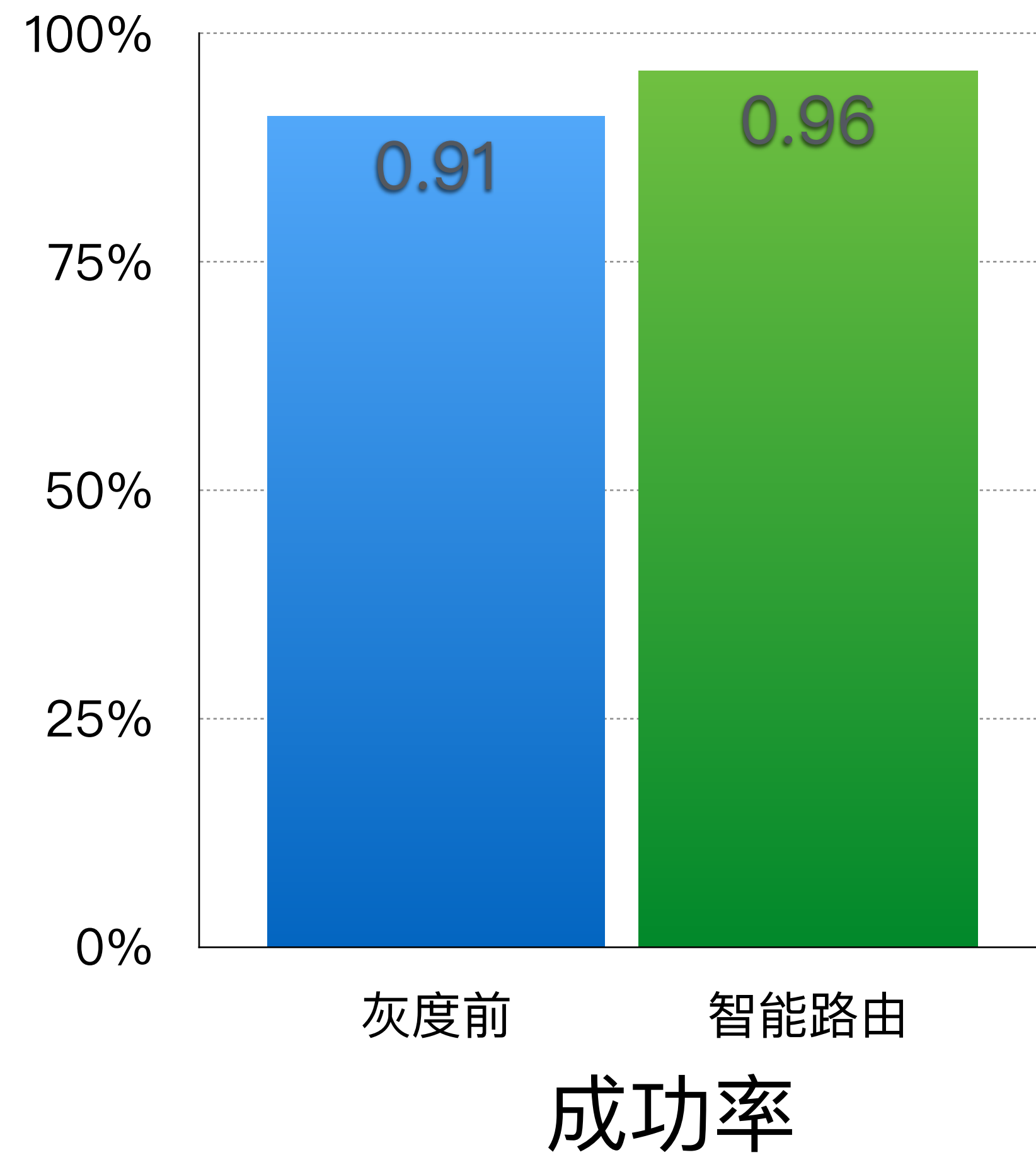
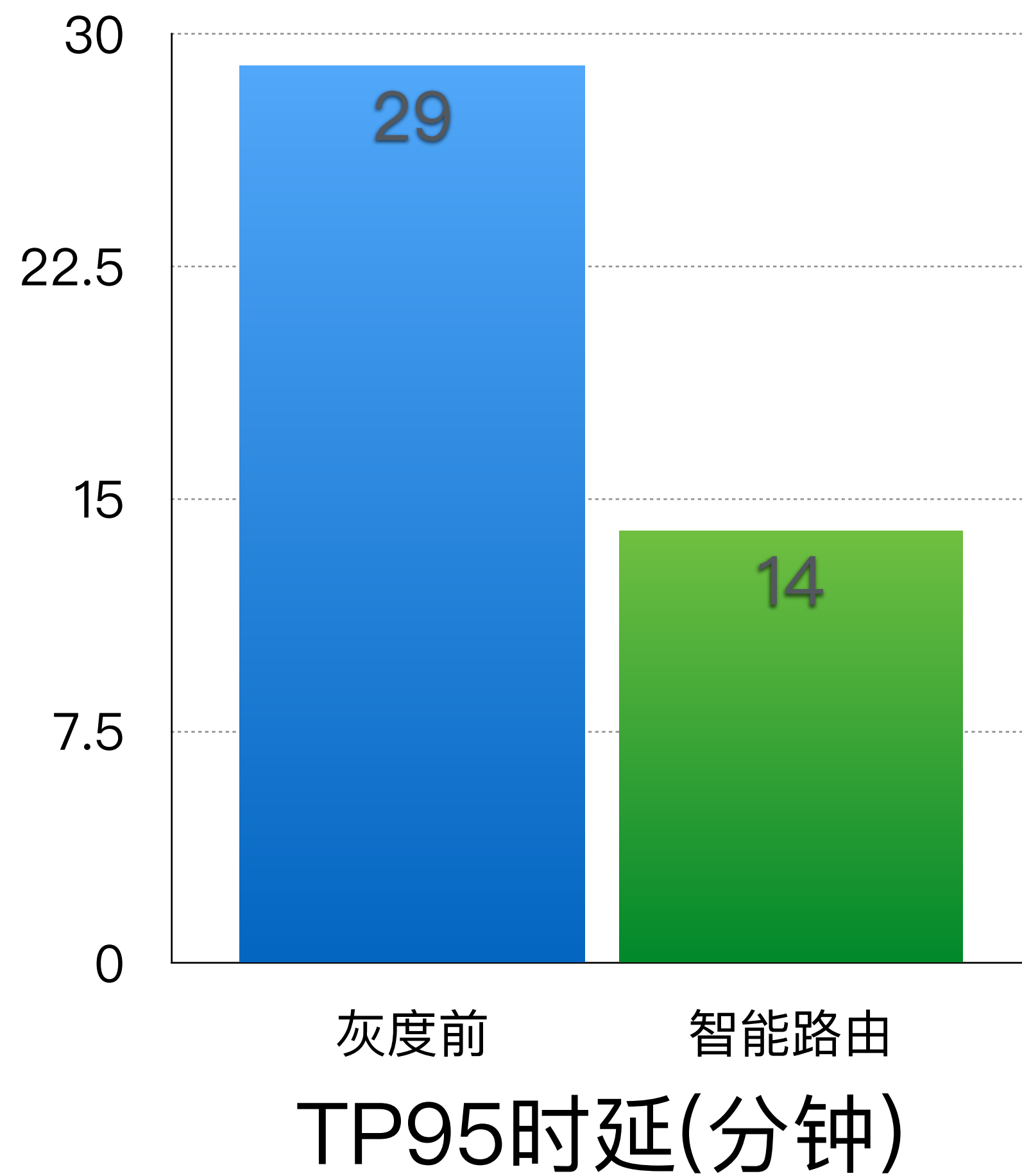
- 为什么查询卡住了?
- 为什么进度一直99%?
- 什么时候能执行完?



# 异常诊断—错误归因



# 效果





# 现状问题

用户

自定义函数

failover时间长

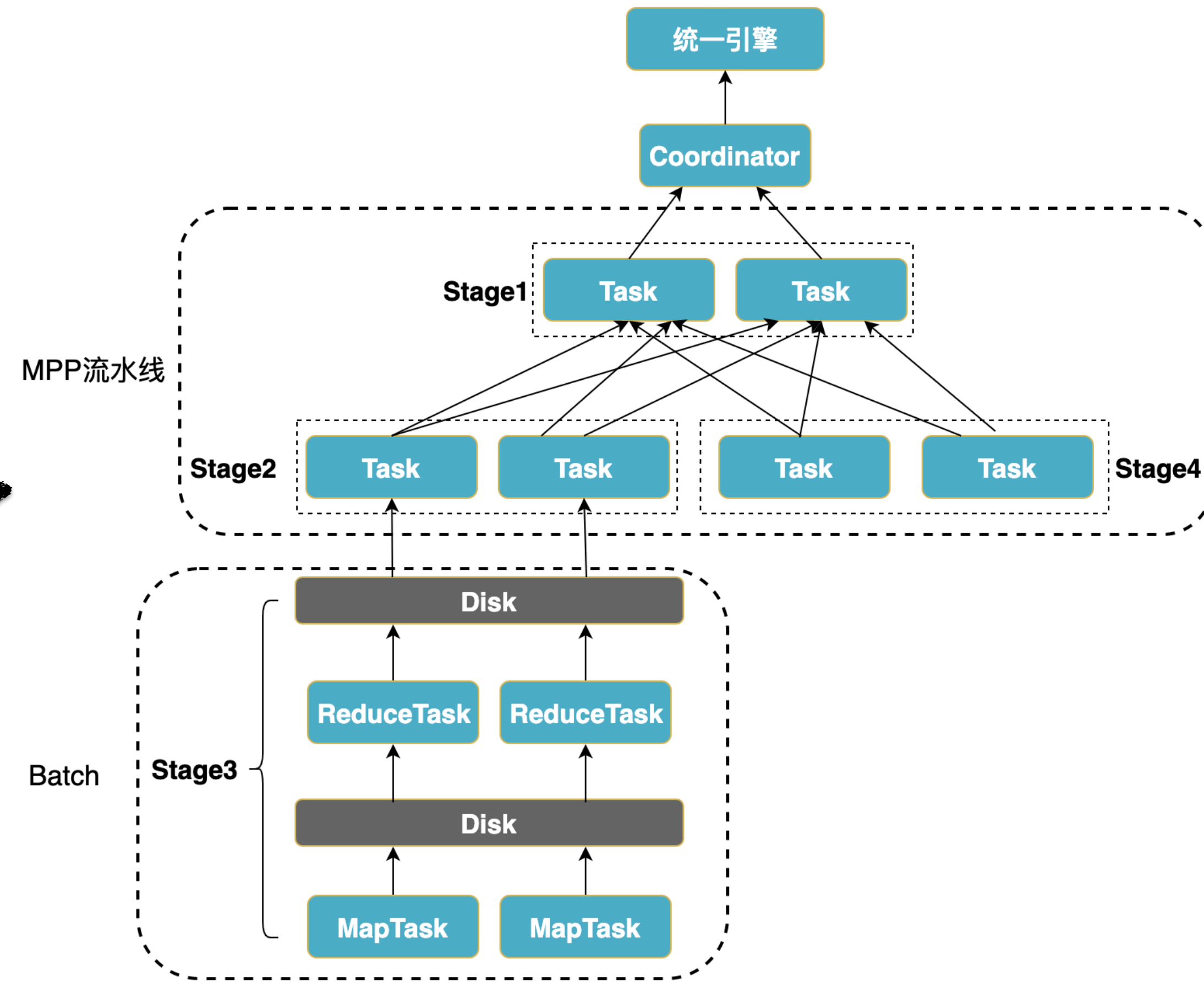
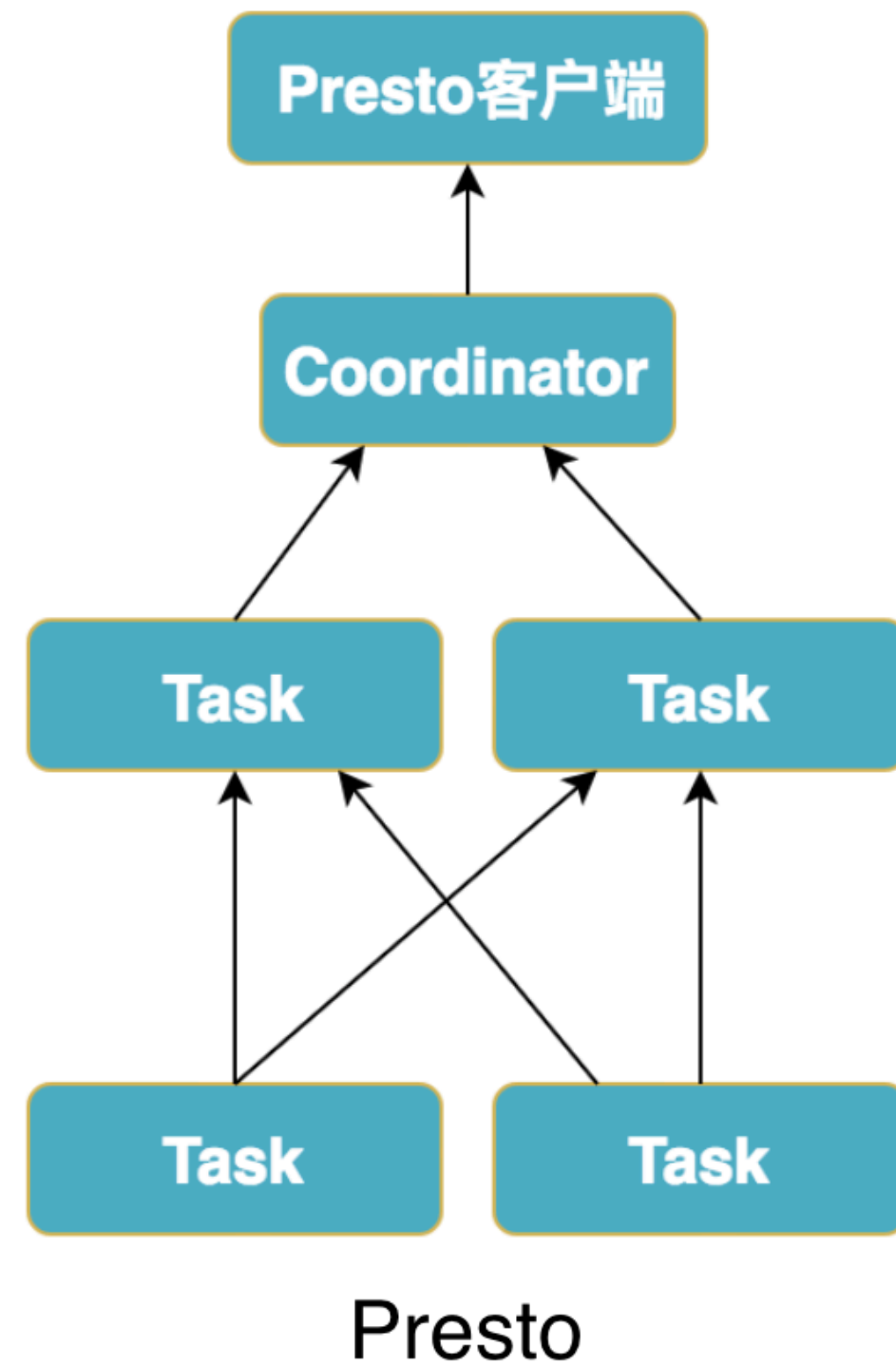
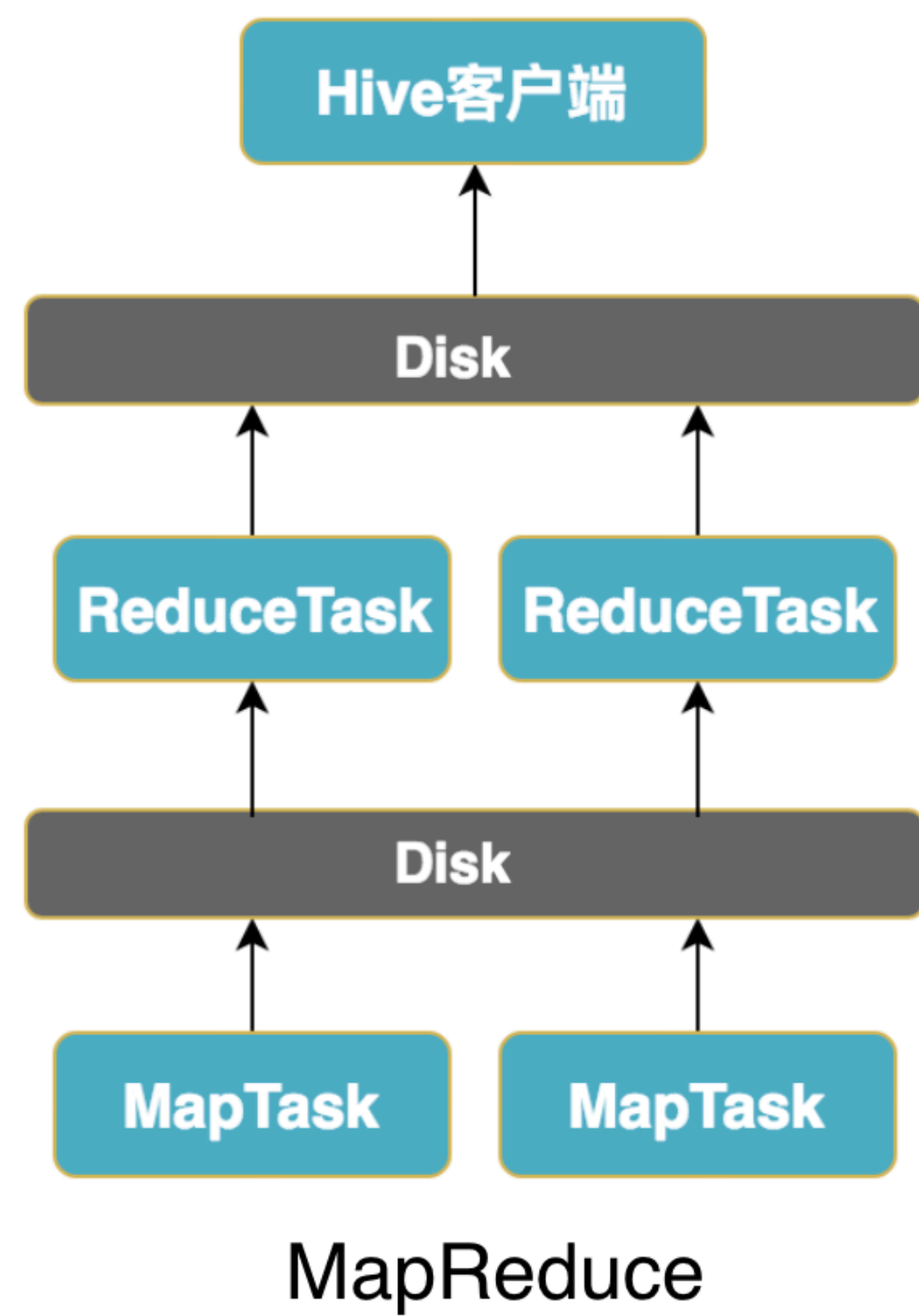
技术

重复解析

cost粒度粗

语法完全对齐

# 未来展望



# 未来展望

- 完善查询事中分析
- 探索两种计算架构的整合
  - 基于列统计信息计算cost
  - 逻辑执行计划转换路由
  - 基于现有引擎实现另一种计算架构



# 精彩继续！ 更多一线大厂前沿技术案例

📍 北京站

## GITC

全球大前端技术大会

时间：10月30-31日

地点：北京·国际会议中心

扫码查看大会  
详情>>



📍 北京站

## QCon

全球软件开发大会

时间：10月30-11月1日

地点：北京·国际会议中心

扫码查看大会  
详情>>



📍 上海站

## QCon

全球软件开发大会

时间：11月25-26日

地点：上海·宏安瑞士大酒店

扫码查看大会  
详情>>





想一想，我该如何把这些  
技术应用在工作实践中？

---

THANKS