



GOTC 2023

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

OPEN SOURCE, INTO THE FUTURE

「AI is Everywhere」专场

本期议题：AI 数据库 OpenMLDB 应用实践

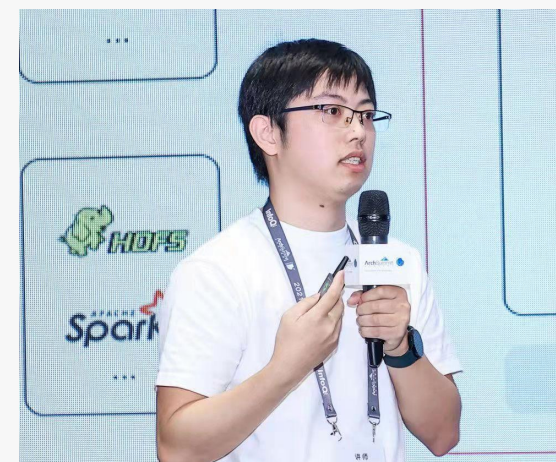
陈迪豪 2023年05月28日

About Me

陈迪豪

chendihao@4paradigm.com

- 第四范式平台架构师
- 开源项目 OpenMLDB 核心研发与 PMC 成员
- 开源项目 HBase / OpenStack / TVM 贡献者
- 目前专注于分布式计算存储系统以及AI应用



目录 Contents

1. 人工智能工程化落地的数据和特征挑战
2. OpenMLDB：线上线下一致的生产级特征平台
3. 社区案例分享

1. 人工智能工程化落地的数据和特征挑战

正确、高效的 AI 数据和特征供给成为数据侧的新挑战

95%

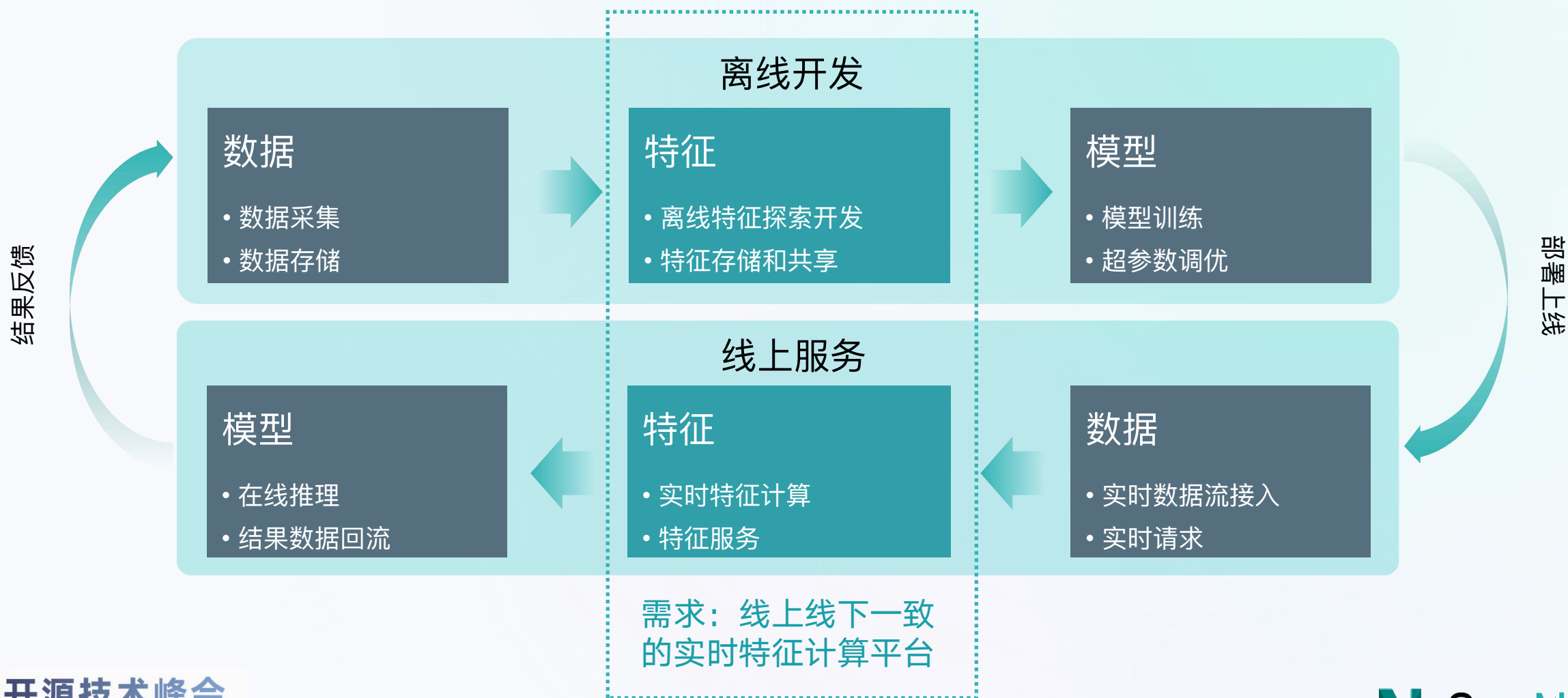
时间精力花费在数据上

Source: How to Operationalize Machine Learning and Data Science Projects, Gartner



≠ AI

机器学习应用从离线开发到上线全流程



实时机器学习决策，需要毫秒级的数据和特征计算能力

两大 AI 应用：感知类、**决策类**

硬实时特征计算真正满足决策需求 – **实时数据、实时计算**

流式计算为 Big Data 和 BI 设计



硬实时场景蕴藏巨大商业价值，鲜有通用商业化产品

银行要求毫秒级业务响应

以某银行反欺诈场景为例

客户需求：

特征计算响应时间 20ms 内，高准召率的事中反欺诈系统

解决方案	响应时间	准召率
传统规则系统	~200ms	较差
客户自研系统	~50ms	中等
第四范式先知	<20ms	优等

产品支撑：高性能、高准确率实时特征数据系统

举例：实时事中反欺诈交易

刷卡记录



卡号	刷卡金额	刷卡时间
012159	1000	2022/01/12 08:00:00

虚拟插入

卡号	刷卡金额	刷卡时间 (已排序)
012112	223	2022/01/12 02:00:00
012159	15	2022/01/12 06:00:00
012159	1000	2022/01/12 07:59:55
012159	2000	2022/01/12 07:59:57
012159	1000	2022/01/12 08:00:00

基于窗口聚合

10s

3h

特征计算

生成的特征

卡号	刷卡金额	过去10秒内：刷卡次数 刷卡最大金额 最小金额 平均金额	过去三小时内：刷卡次数 刷卡最大金额 最小金额 平均金额
012159	1000	3 2000 1000 1333	4 2000 14 1003

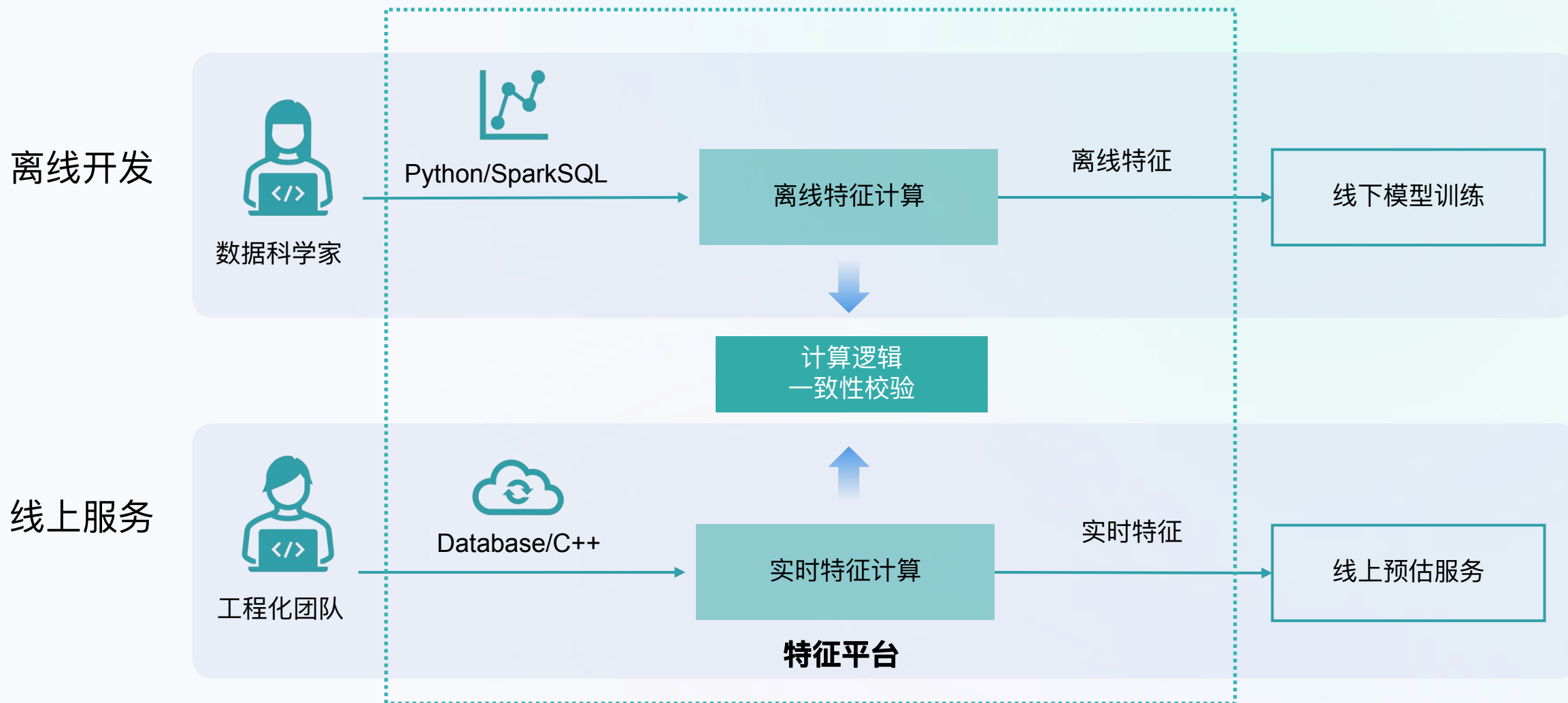
工程化需求

1. 线上线下一致性
2. 低延迟、高并发、高可用

模型推理

欺诈交易?

传统方式：离线开发和线上服务分离，高成本投入



线上线下一致性可能的原因

工具能力的不一致性

离线开发



Python

```
import pandas as pd
t1 = pd.read_csv("data.csv")
account_feat = t1['account'].std()
```

$$\sqrt{\sum_{i=0}^N \frac{(x_i - \mu)^2}{N - 1}}$$

标准差

(Bessel's Correction)

线上应用



MySQL

```
select std(account) as account_feat from t1;
```

$$\sqrt{\sum_{i=0}^N \frac{(x_i - \mu)^2}{N}}$$

标准差

需求沟通的认知差

Varo

Banking Benefits Resources Blog Help

LOG IN

OPEN ACCOUNT

A bank for all of us



Money shouldn't just work for some of us. It should work for all of us.

	Account Balance
线上应用	current "account balance"
离线开发	"account balance" as of yesterday

线上线下一致性校验带来的高昂工程化落地成本

线下开发



两组 不同技能栈的开发人员投入

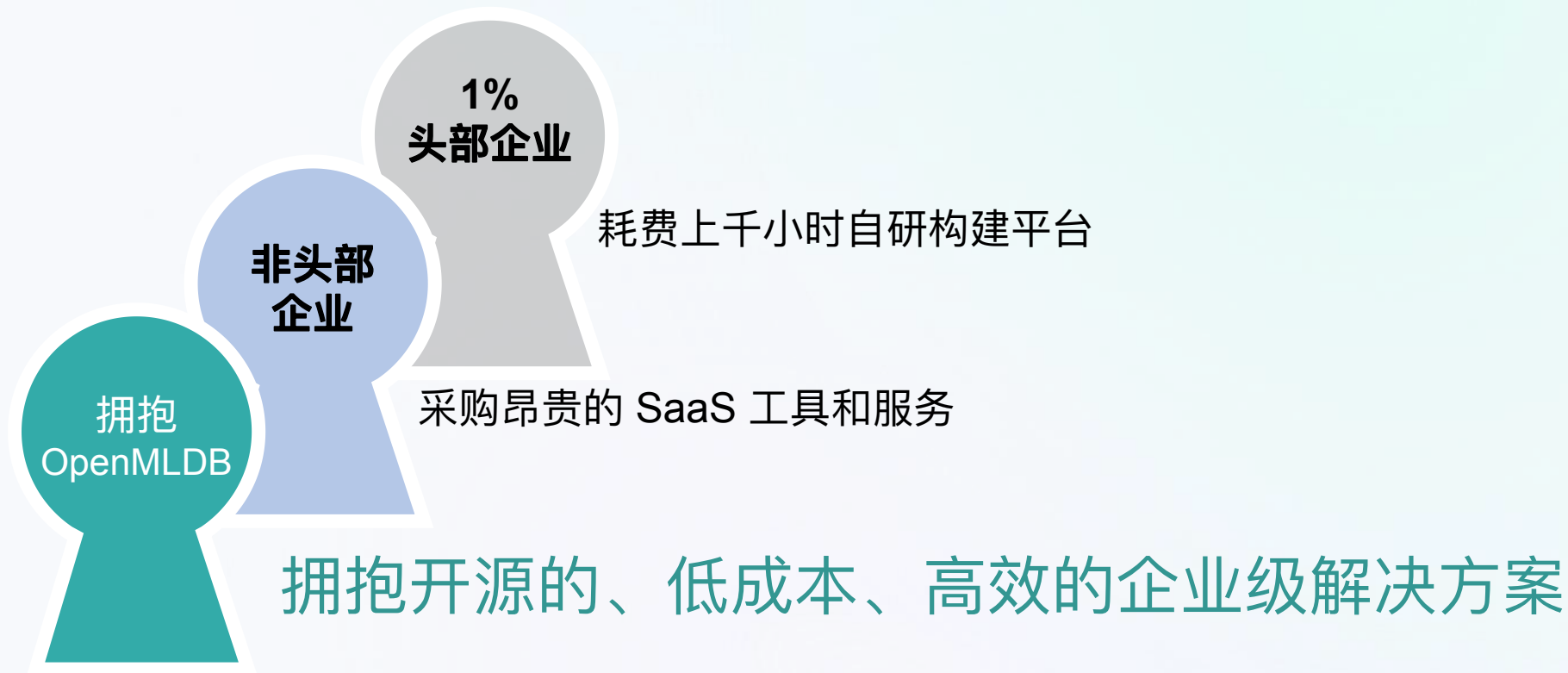
对齐+校验

两套 系统的开发、运营



线上服务

特征平台工程化解决方案



2. OpenMLDB: 线上线下一致的生产级特征平台

OpenMLDB 发展历程：从闭源走向开源



开源前，跟随第四范式 先知 平台，在 100+场景 落地，覆盖超过 300个节点。
开源后，以开放姿态积极拥抱社区开发者、整合开源生态。

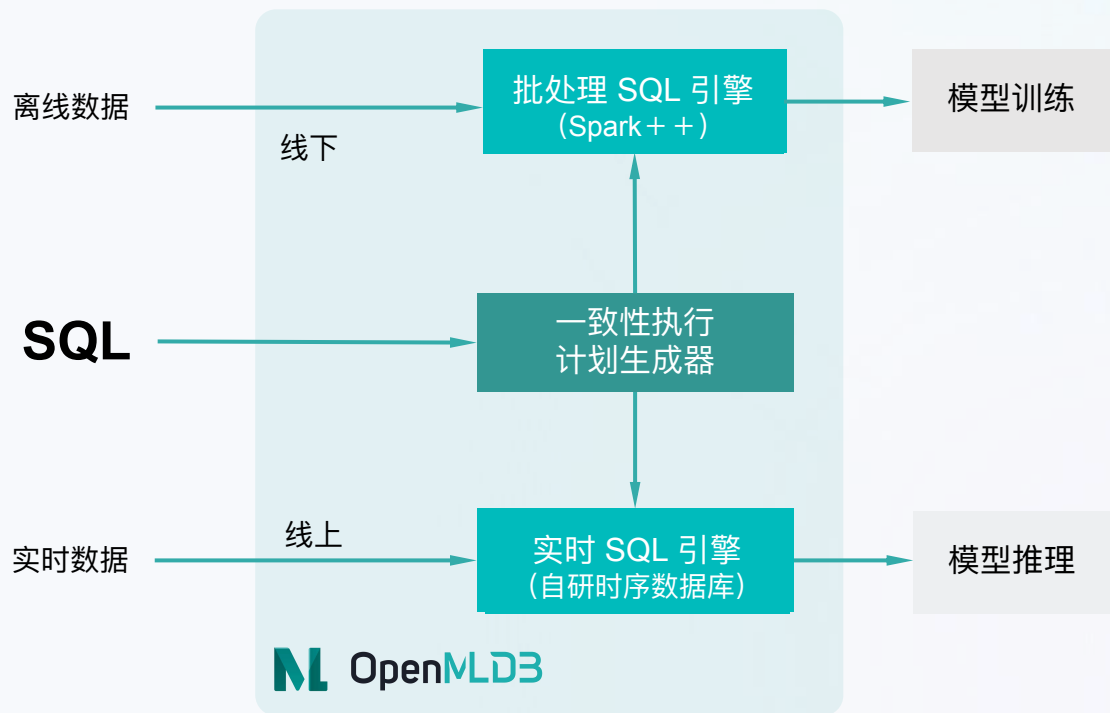


主要使用场景

信用卡现金分期精准营销	贷前风险评分	营销获客	个性化推荐	反洗钱可疑交易智能识别
信用卡账户风险预警	合规额度决策	风险管理	投顾客户挖掘	信用卡申请反欺诈
欺诈养卡防控	理财个性化推荐	零售贷款反欺诈	历史客户激活	客户流失预警
网点流量预测	交易欺诈评分	现金分期个性化推荐	信用卡交易反欺诈	金融产品推荐

OpenMLDB: 开源机器学习数据库, 线上线下一致的特征平台

OpenMLDB 抽象架构



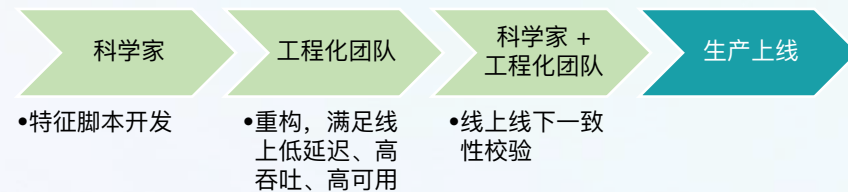
实现开发即上线, 节省数月人天成本

基于 OpenMLDB 流程

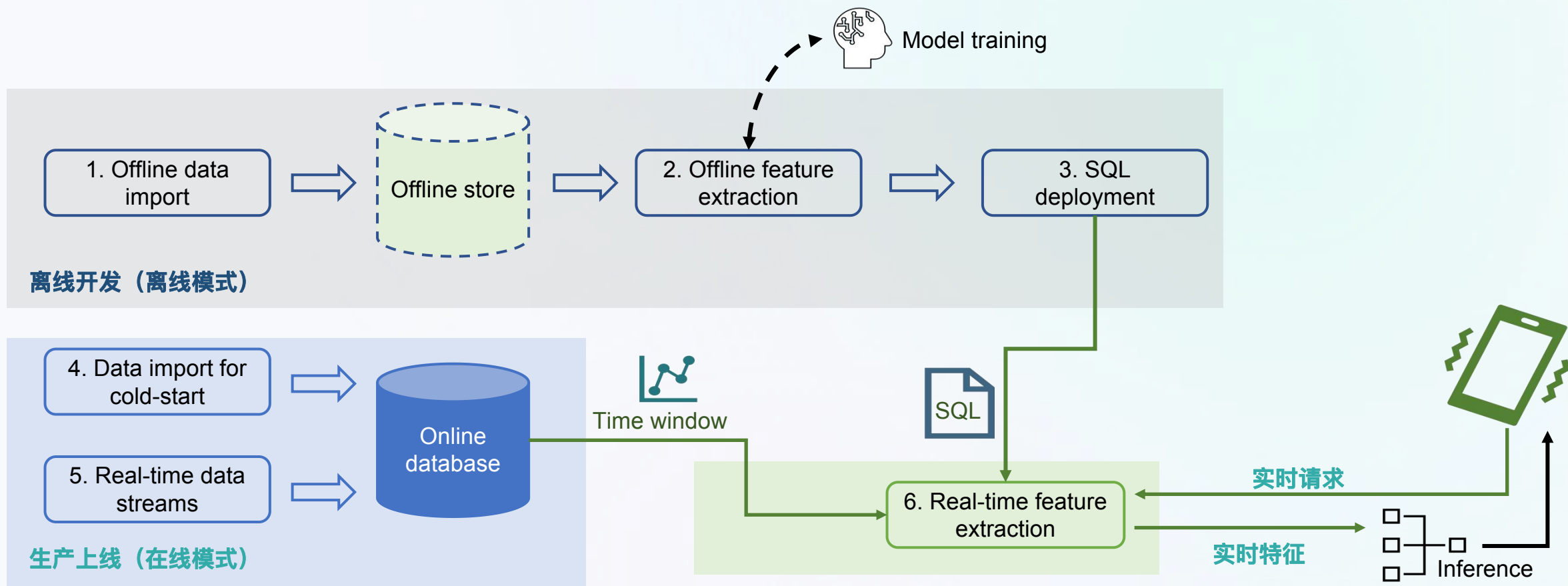


OpenMLDB

原有流程



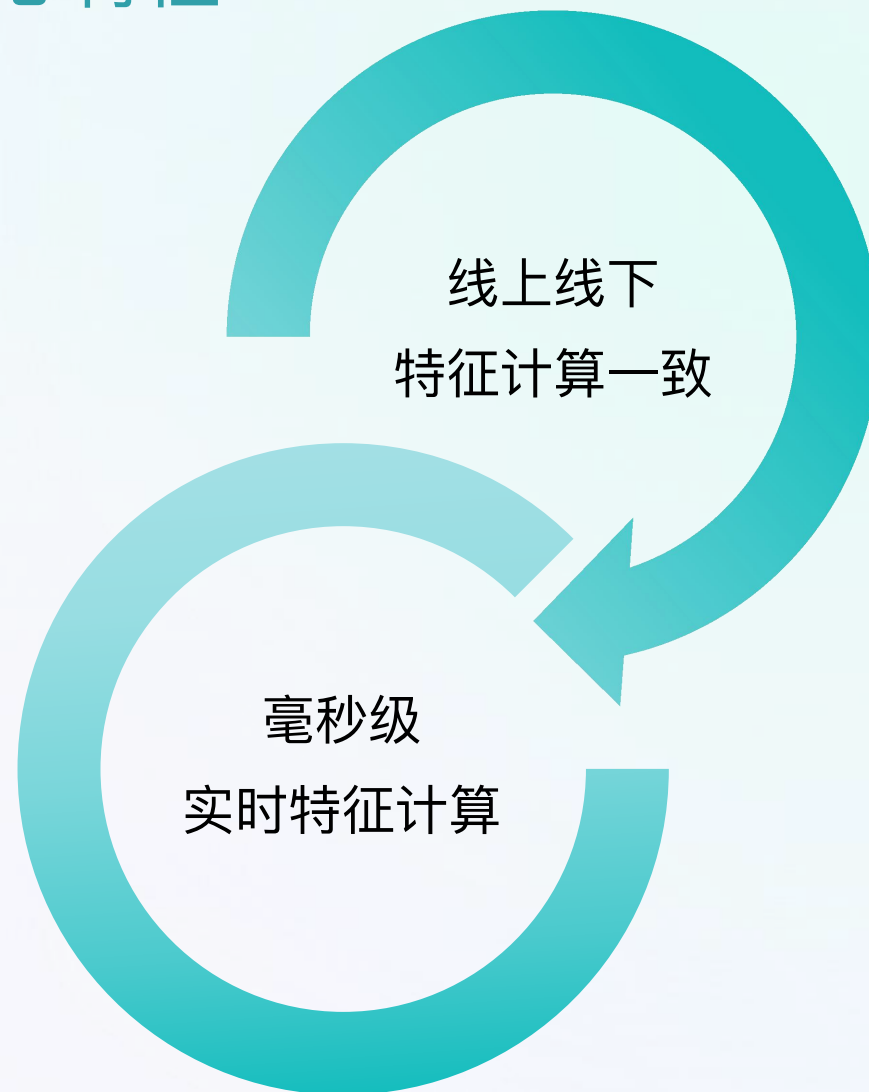
从离线开发到线上服务完整流程



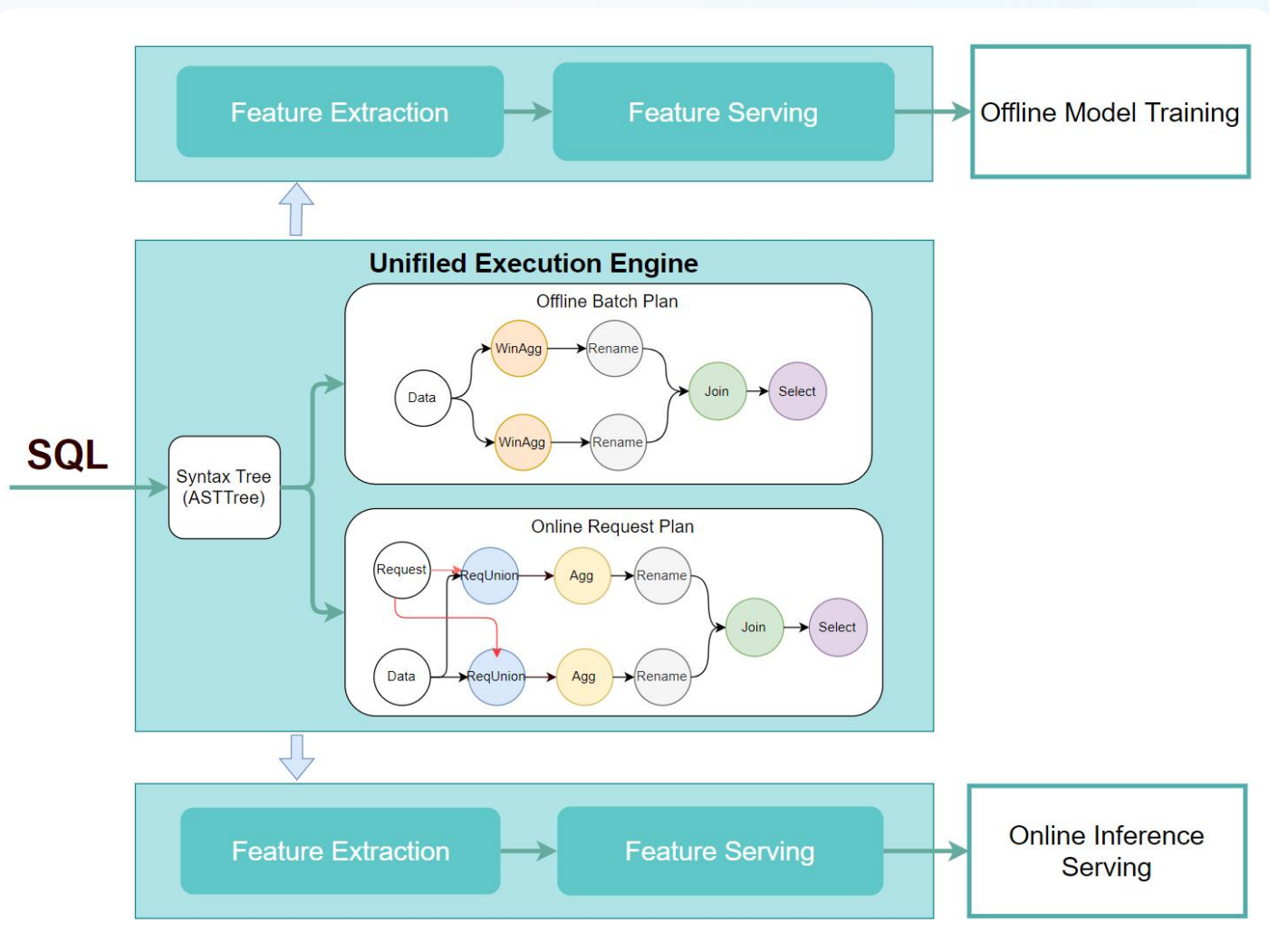
解决一个核心问题，提供一个核心特性

核心问题

核心特性



核心组件一：线上线下一致性执行引擎



- 统一的底层计算函数
- 逻辑计划到物理计划的线上线下一致性执行模式自适应调整

线上线下一致性得到天然保障

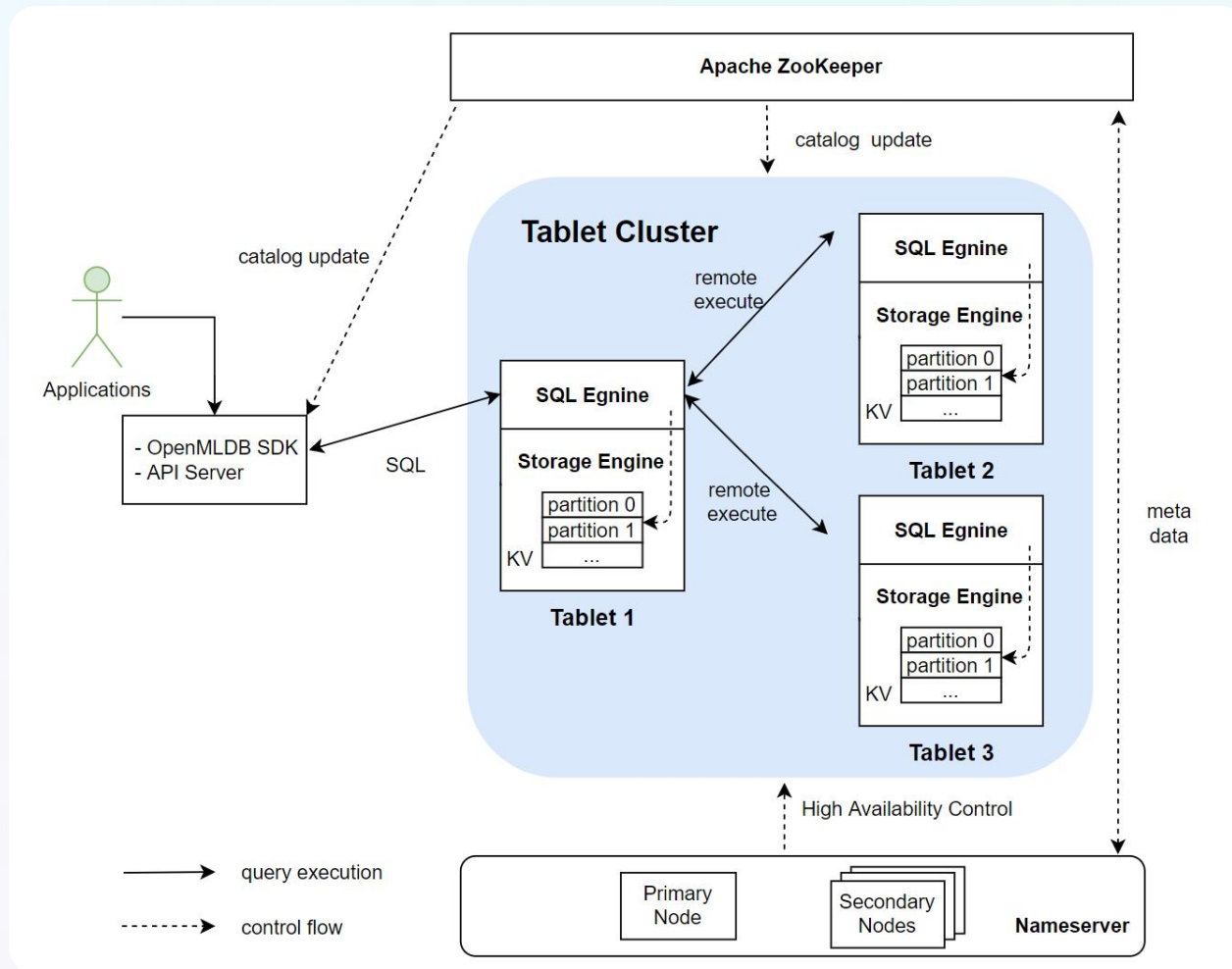
核心组件二：高性能实时 SQL 引擎

分布式实时 SQL 引擎主要模块

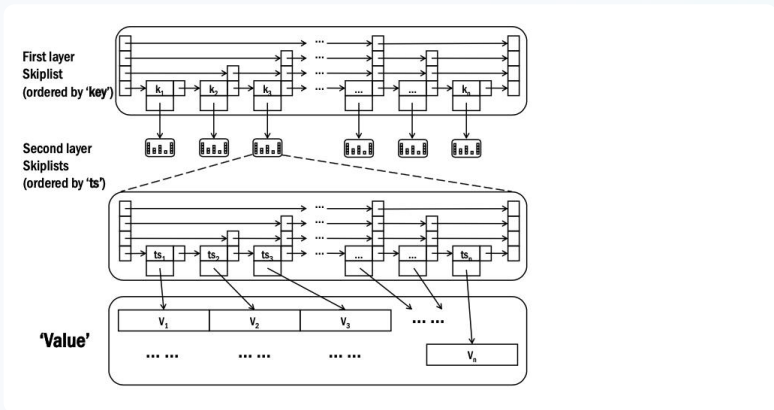
- **ZooKeeper** – 元数据存储和管理
- **Nameserver** – tablet 管理和故障转移
- **Tablets**
 - 分布式 SQL 执行引擎
 - 分布式存储引擎：
内存、磁盘双存储引擎
- **高性能、可扩展、高可用**

详细线上引擎架构描述参见：

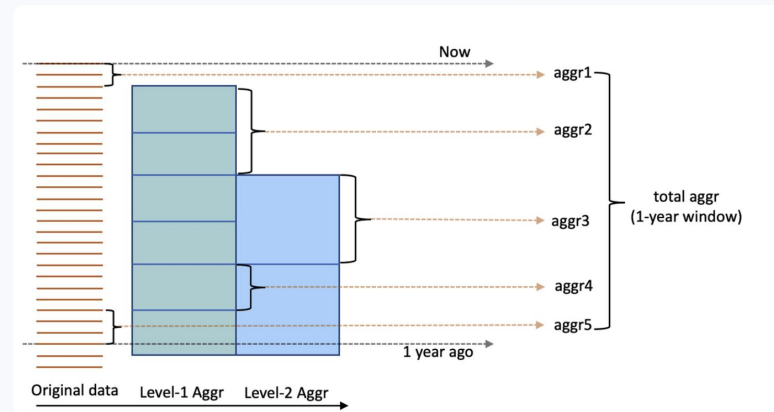
<https://openmldb.feishu.cn/wiki/wikcnavULzxKH5Aka3ox0871R2f>



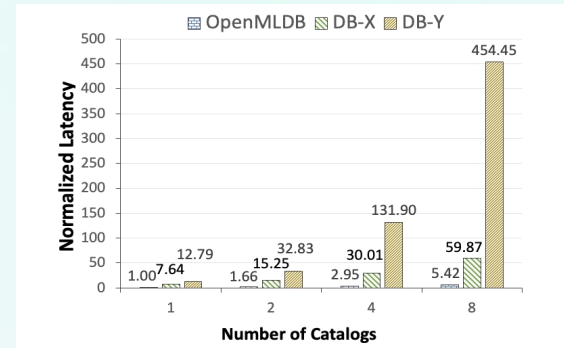
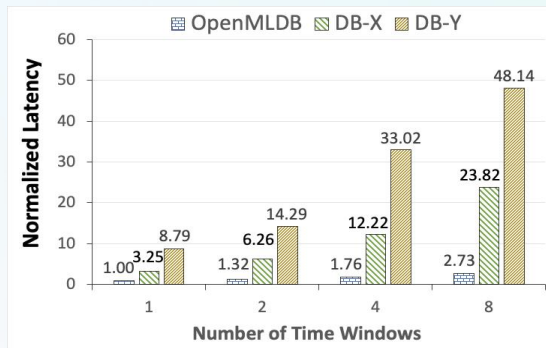
核心组件二（续）：高性能实时 SQL 引擎 – 核心优化技术



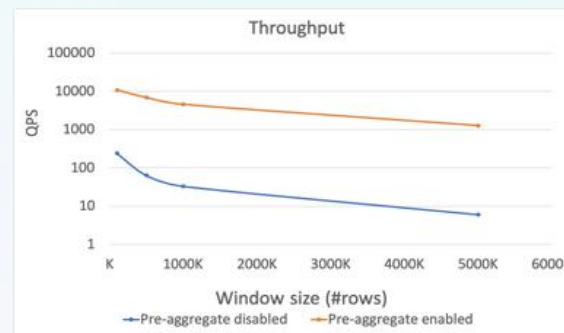
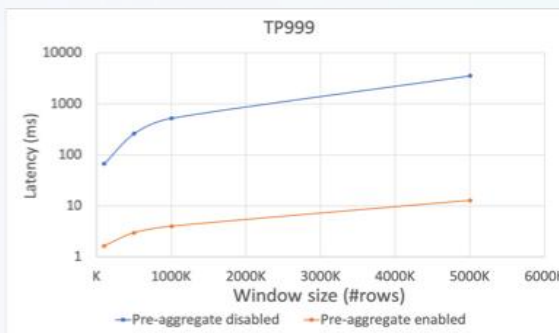
核心优化：双层跳表（内存引擎）



核心优化技术：预聚合



实时计算性能：OpenMLDB 和其他商业化内存数据库作比较



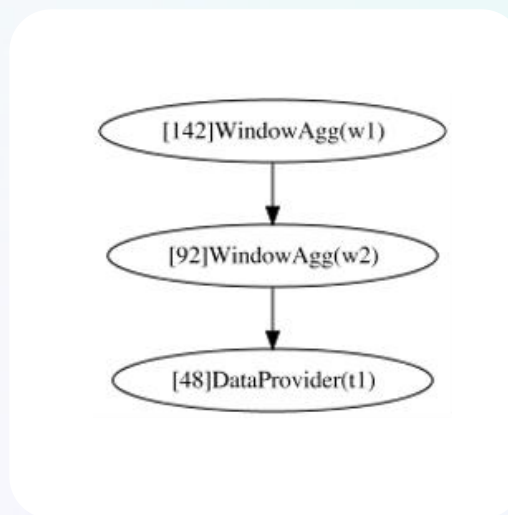
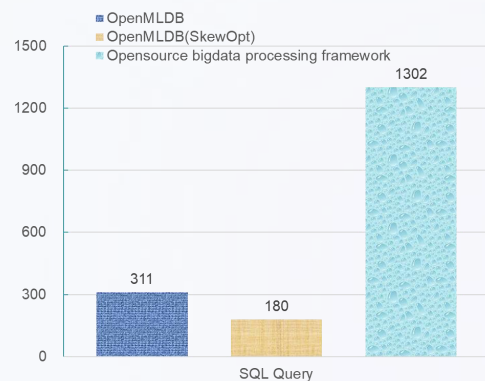
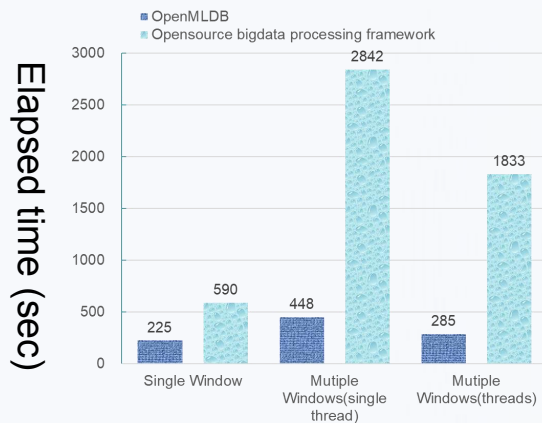
OpenMLDB 使用预聚合技术的性能提升

完整测试报告: <https://openmlDB.feishu.cn/wiki/wikcnZRB9VRkqgD1vDFu1F9AaTh>

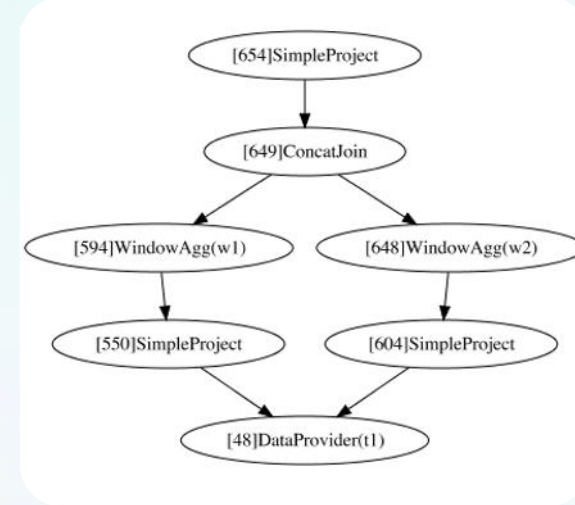
核心组件三：面向特征计算的优化的离线计算引擎

- 多窗口并行计算优化
- 数据倾斜计算优化
- SQL 语法扩展
- 针对特征计算优化的 OpenMLDB Spark 发行版

```
SELECT
  min(age) OVER w1 as w1_min_age,
  min(age) OVER w2 as w2_min_age
FROM t1
WINDOW
  w1 as (PARTITION BY name ORDER by age ROWS BETWEEN 10 PRECEDING AND CURRENT ROW),
  w2 as (PARTITION BY age ORDER by age ROWS BETWEEN 10 PRECEDING AND CURRENT ROW)"
```

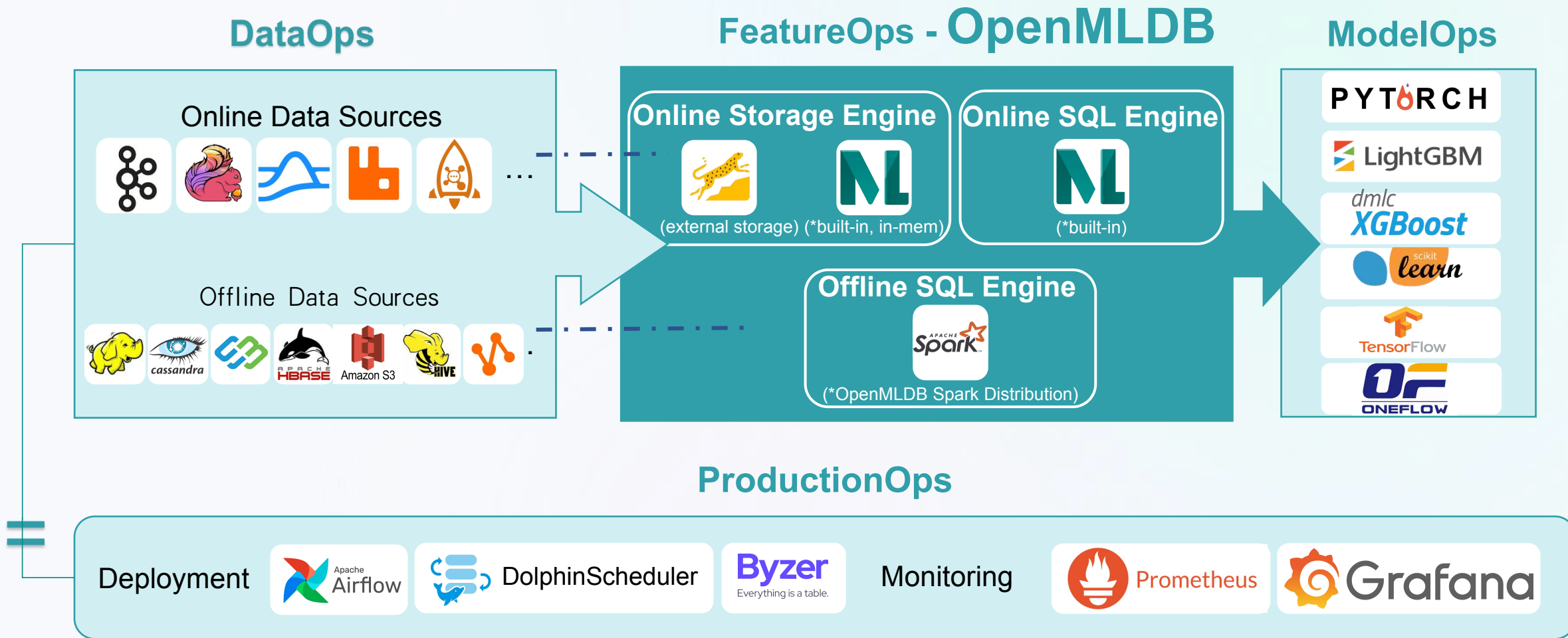


Spark 3.0.0



OpenMLDB

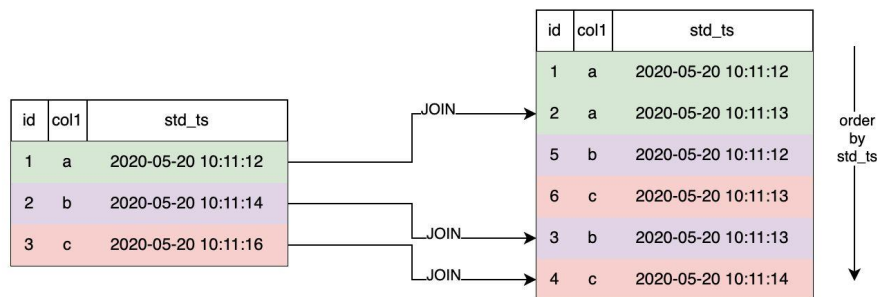
OpenMLDB 开源生态



高级功能：针对特征工程的 SQL 扩展

LAST JOIN

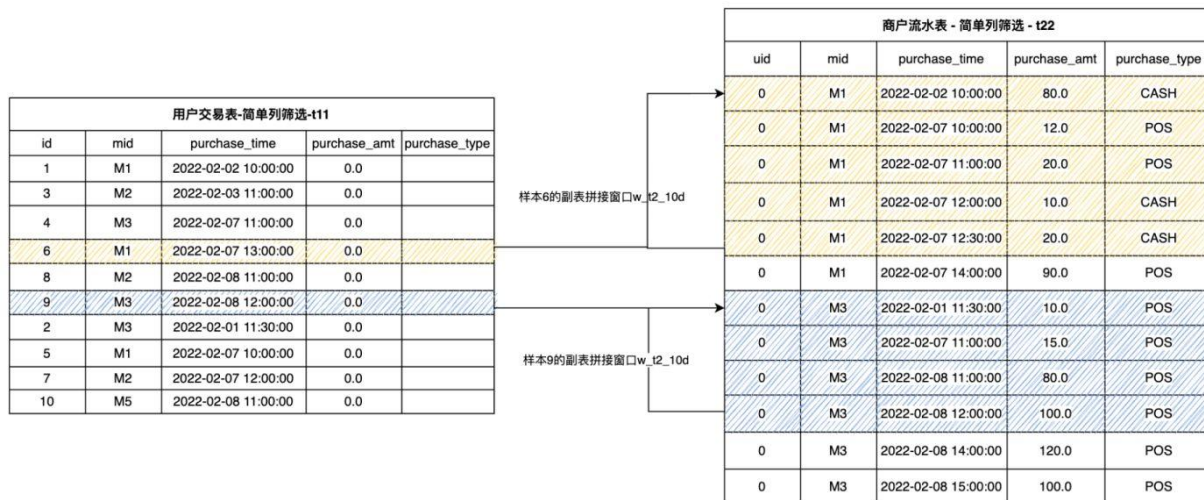
多行匹配时，仅匹配最新记录



id	col1	std_ts	id	col1	std_ts
1	a	2020-05-20 10:11:12	2	a	2020-05-20 10:11:13
2	b	2020-05-20 10:11:14	3	b	2020-05-20 10:11:13
3	c	2020-05-20 10:11:16	4	c	2020-05-20 10:11:14

WINDOW UNION

跨表的 join 和窗口聚合操作 (point-in-time), 避免特征穿越

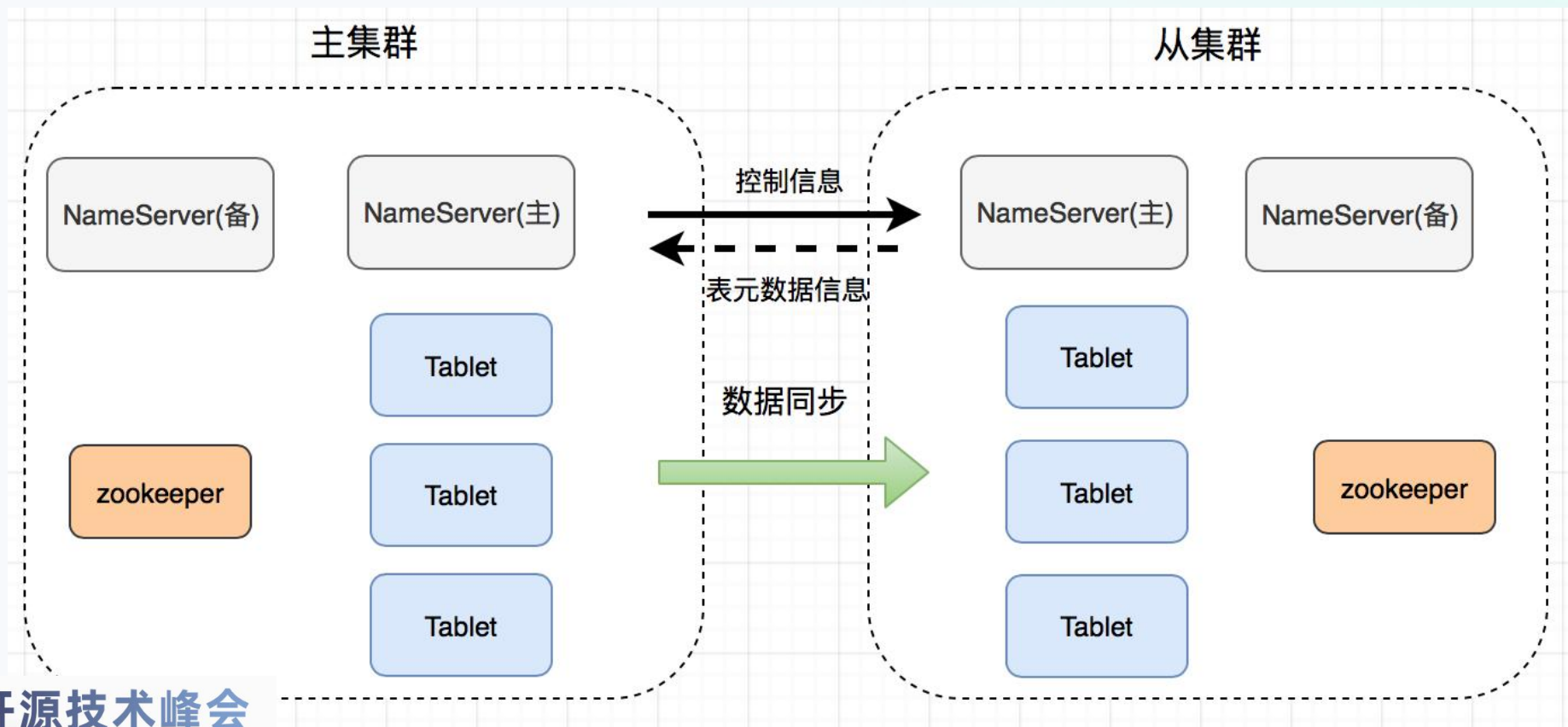


高级功能：跨机房容灾

构建多个 OpenMLDB 集群，部署位主从复制模式，实现跨机房容灾

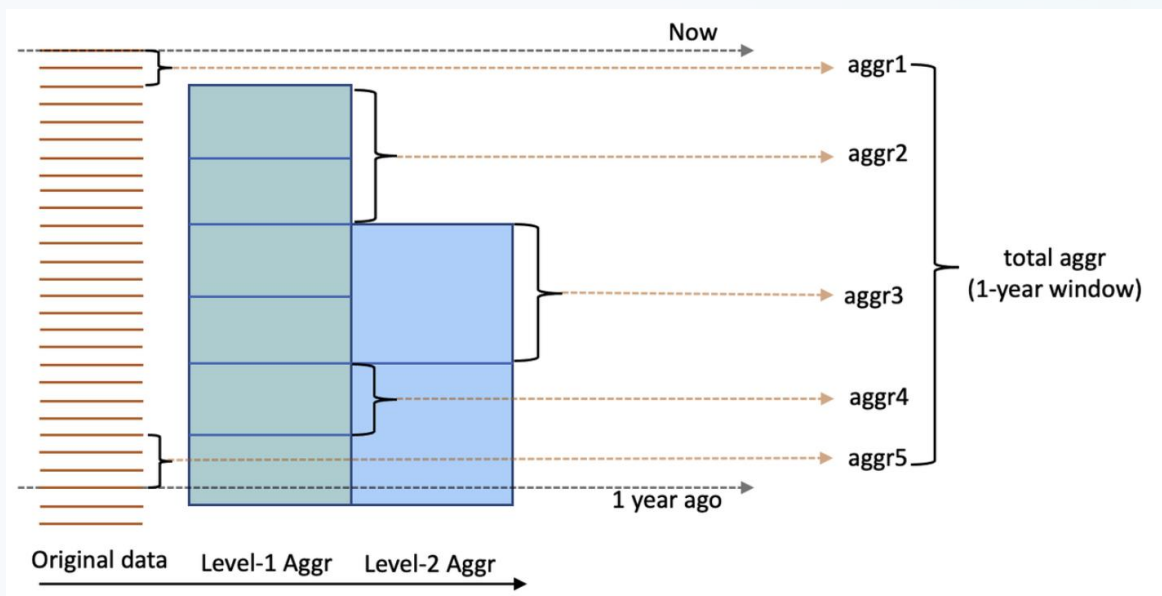
主集群：能支持 读写 的集群，并且可以给从集群同步数据

从集群（一个或多个）：只提供 读 请求的集群，数据和主集群保持一致

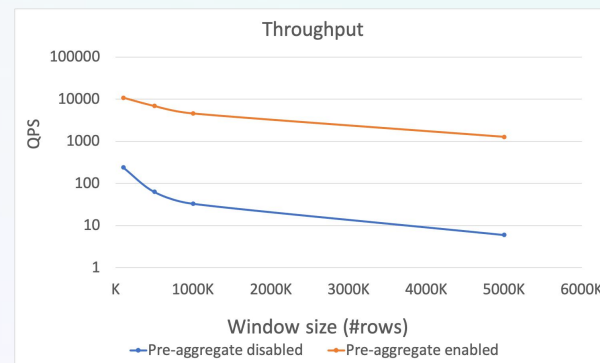
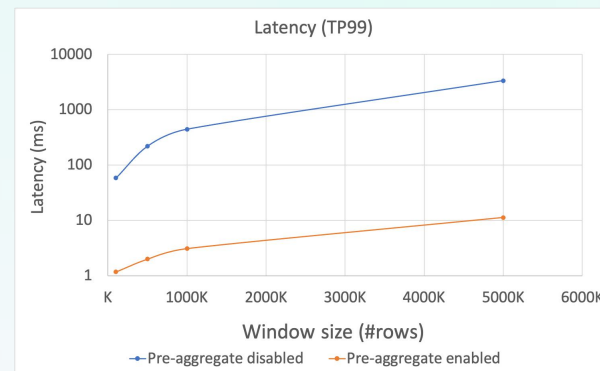


高级功能：线上预聚合

当窗口内数据量巨大（上百万），启用预聚合方式，显著优化此种情况下的延迟



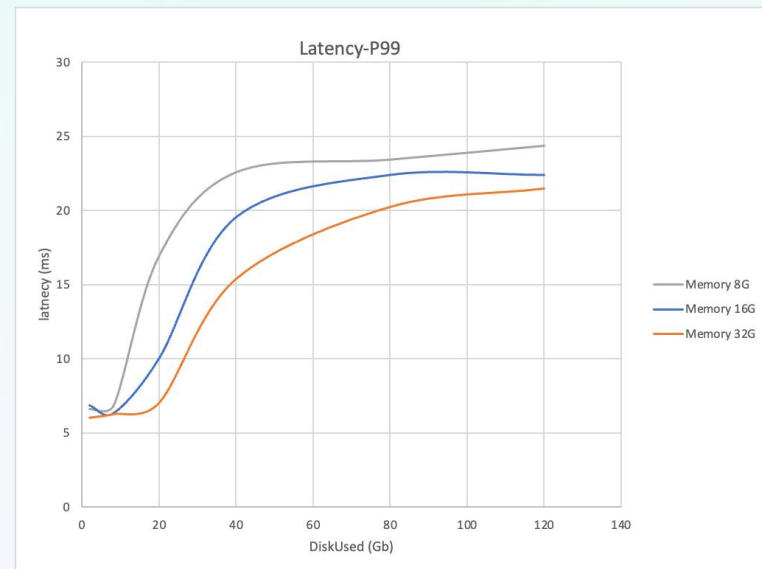
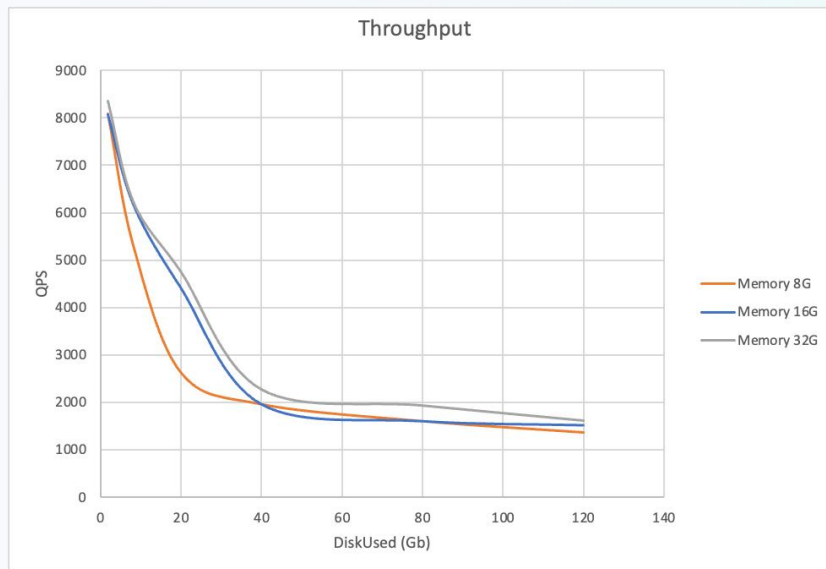
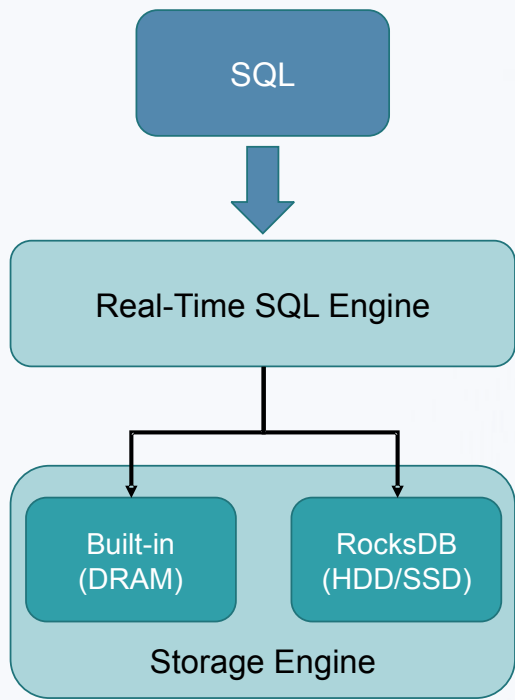
基于两层预聚合表的跨度一年的实时窗口计算



使用预聚合技术，性能提升 1-2 个数量级

高级功能：双存储引擎架构

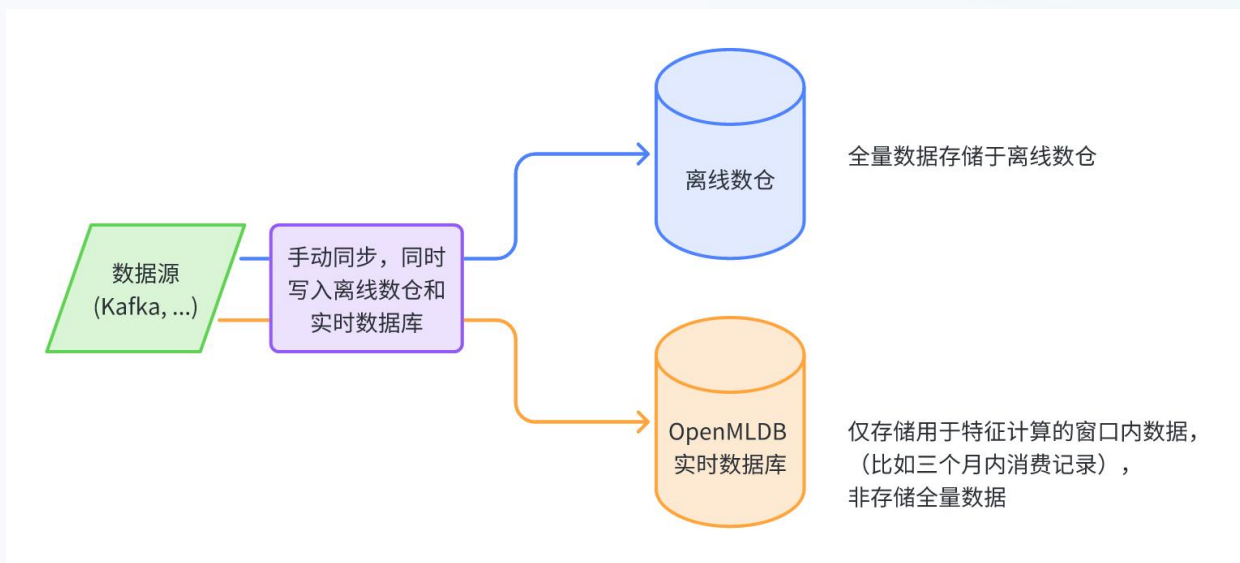
线上引擎的磁盘存储引擎提供了比内存存储引擎 5倍 以上节省成本的方案，但是会带来一定性能衰退



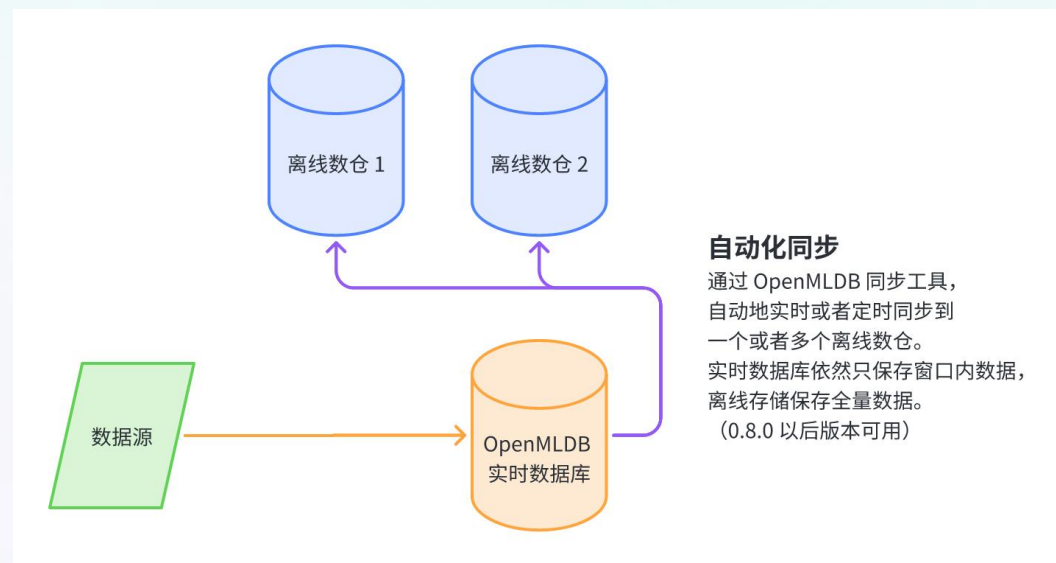
QPS/TP99 均有 4-5x 的性能衰退，取决于应用场景按需选择

高级功能：自动化在线离线数据同步

在线、离线数据源同步，从手动同步机制创建，进化为自动化同步机制



手动建立离线、在线数据同步



自动化在线到离线数据同步
(v0.8.0 以后可用)

高级功能：智能化运维和诊断工具

智能诊断工具：自动化服务异常状态检查，日志智能搜集，

智能运维工具： *openmldb_ops*，一键数据和表格恢复，自动扩缩容分片平衡，

服务状态检查

自动化异常状态诊断
(版本校验，配置文件检查， ...)



一键数据恢复

异常状态下数据快速恢复
(网络异常， *auto_failover=false*, ...)



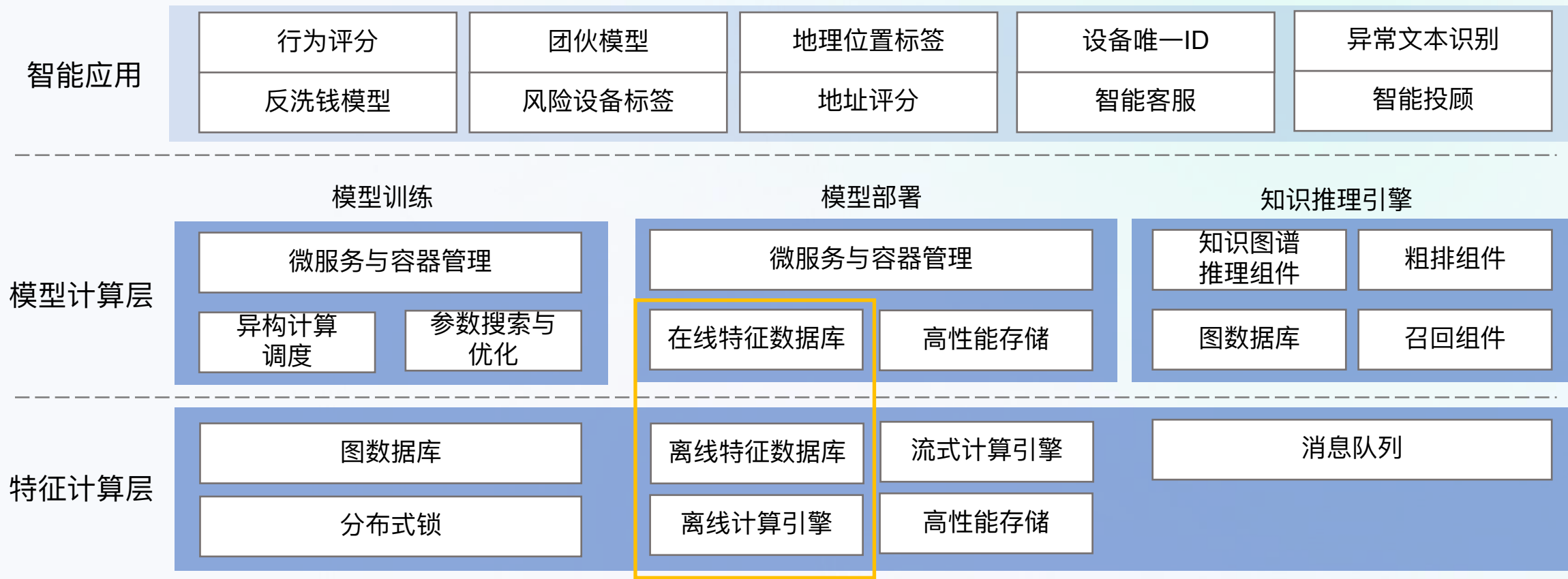
分片自动平衡

扩缩容后分片自动迁移重平衡
(基于启发式算法)



3. 社区案例分享

OpenMLDB 案例 – Akulaku 智能计算架构中的特征平台



场景驱动: OpenMLDB

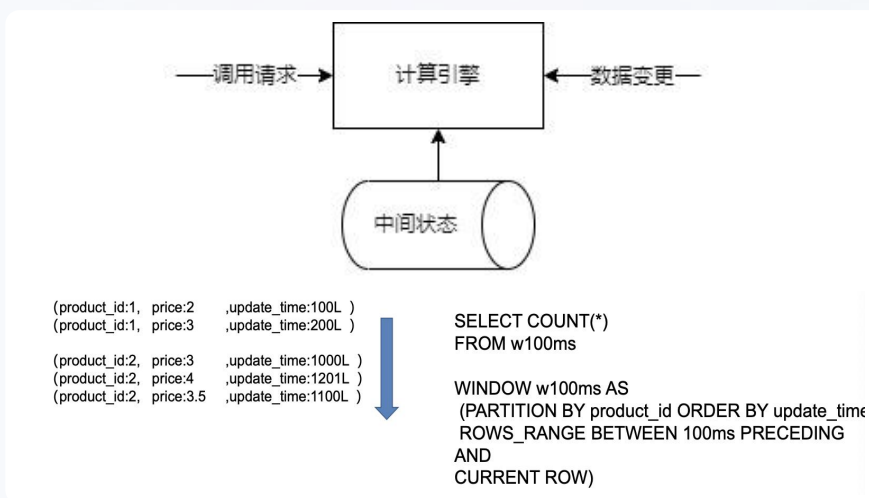
Akulaku 智能风控场景，对 10 亿条订单进行窗口特征计算，达到 4 毫秒延迟性能

特征计算环节难点

- **线上部署**：低延迟，高时效性，尽可能反映数据变更
- **线下分析**：高吞吐量
- **逻辑一致**：线下分析和线上部署的逻辑需要完全一致

OpenMLDB 解决方案

- **场景驱动**：业务调用环节驱动，实时计算结果，现用现算
- **具体方案**：1) 使用SQL作为离线和在线的桥梁；2) 在线基于时序数据库做时间滑窗



基于 OpenMLDB 的业务实现

- **场景**：近1天订单个数实时计算
- **数据量**：10亿条订单数据/天
- **需求**：实时更新，时间窗口实时滑动，存在复杂关联需求
- **测试结果**：4毫秒延迟





欢迎加入 OpenMLDB 社区

OpenMLDB 微信交流
群



OpenMLDB 中文官网: <https://openmldb.ai/>

GitHub: <https://github.com/4paradigm/OpenMLDB>



THANKS