

Apache Doris × 阿里云联合 Meetup

🕒 10月26日 (周六) 13:30-17:15



星火智云基于阿里云SelectDB 在用户圈选场景的应用实践

张远银

星火智云 数据应用与架构 技术负责人



目录

一、业务背景

二、架构演进

三、场景实践

四、未来展望

Apache Doris x 阿里云

Apache Doris x 阿里云

Apache Doris x 阿里云

Apache Doris x 阿里云

一、业务背景

Apache Doris x 阿里云

Apache Doris x 阿里云

Apache Doris

公司简介

星火科技集团成立于2017年3月，下辖星火智云科技公司、安行天下保险经纪公司(全国性保险经纪牌照)、简单宝科技有限公司星火数智科技公司、老司机科技公司。

成立以来，公司始终坚持“**共建、共享、共赢**”的核心经营理念、“**为企业服务的智能营销与运营全面赋能**”的核心发展战略、“**全域营销获客+私域运营服务**”的核心价值定位，致力于建设**全球领先的智能化融合营销服务平台**。

作为智能营销运营服务商，星火科技本着“**务实高效、价值创造**”的精神和“**诚信敬业、创新专业**”的价值观，致力于对保险业、泛金融行业、大健康领域的企业提供“全域营销获客”与“私域运营服务”相结合的一站式智能营销服务平台，提供涵盖“**获客营销分析洞察链路效果归因决策**”、“**用户全生命周期服务运营**”的一体化业绩增长解决方案，助力企业数字化智能化转型，以科技驱动商业革新让商业服务更简单。



背景介绍

背景介绍

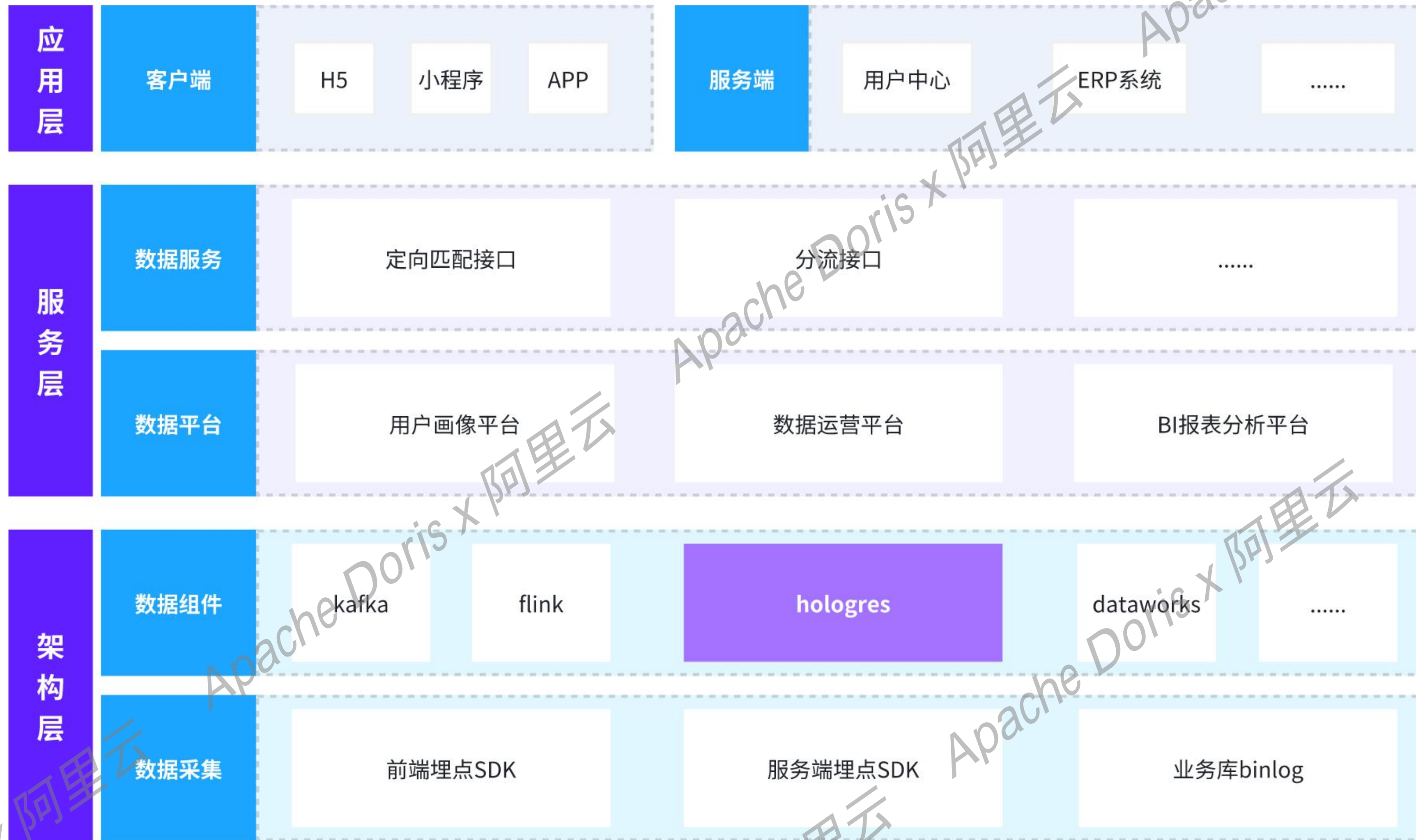
随着业务规模的不断扩展，数据量在膨胀，业务复杂度在提升，且呈现多样性变化。经常需要针对不同业务场景和需求进行定制化开发，特别是**用户圈选**场景，需求灵活多变，往往需要结合**画像**、**行为表**和**业务表**进行**多维度圈选**，来满足业务日常的营销推广，如：消息类营销，包括定时型、触发型消息推送，要保证触达用户的及时性，对圈选和点查性能要求比较高。

解决思路

为了满足业务复杂多样的需求，我们基于阿里云 SelectDB 版构建了整体数仓链路，来达到“**数据驱动经营**”的目的。

二、架构演进

架构v1.0



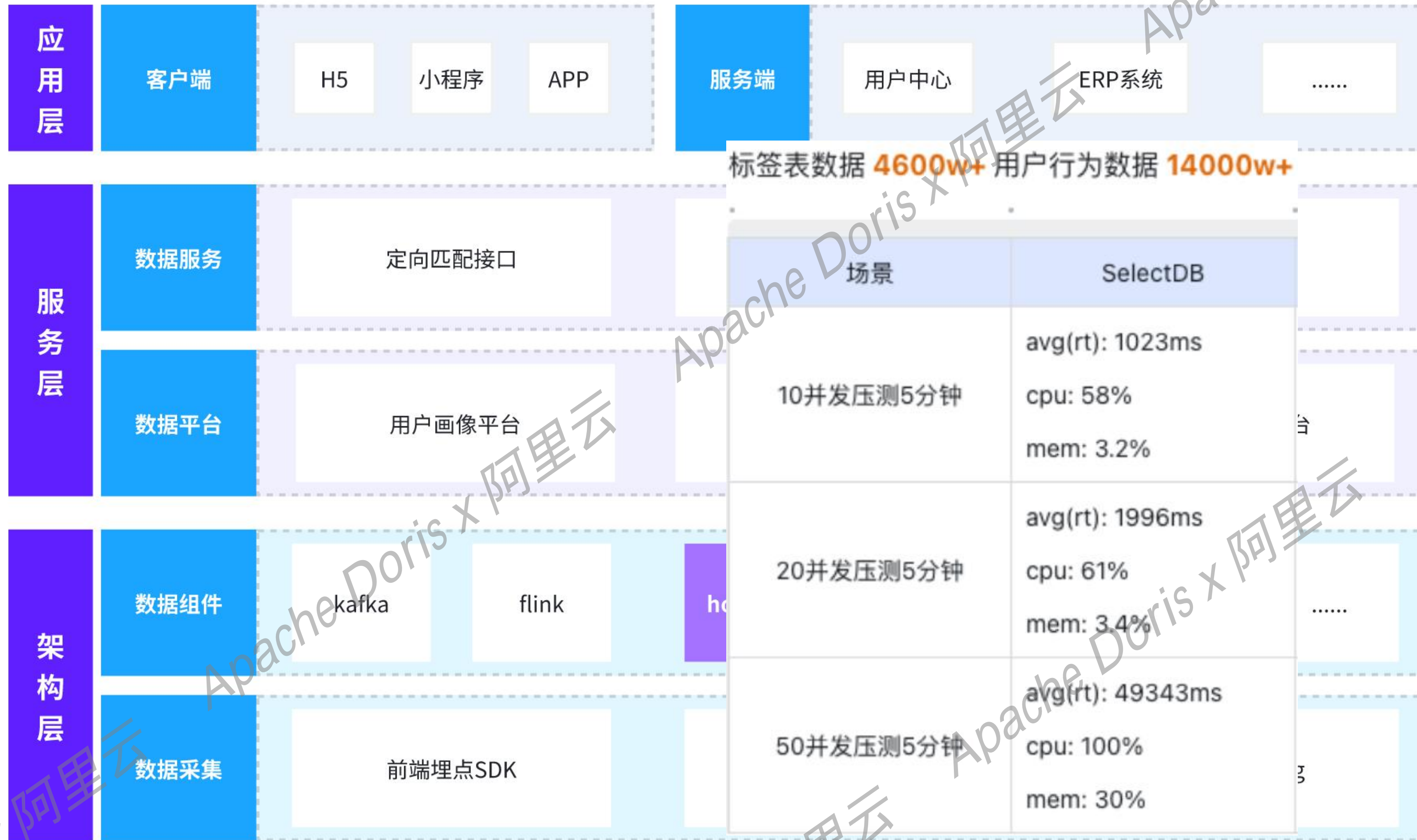
架构设计

- 实时: Flink + Hologres
- 离线: MaxCompute + Hologres

遇到的问题

- 随着业务发展, 发现一些局限, 如: 字段名修改、物化视图等, 对业务实现带来一定的使用成本。
- 在大表 Join 场景表现较弱, 查询耗时较长或查不出数据。
- 不支持跨库查询。

架构v1.1



架构设计

- 实时: Flink + Hologres + [SelectDB](#)
- 离线: MaxCompute + Hologres

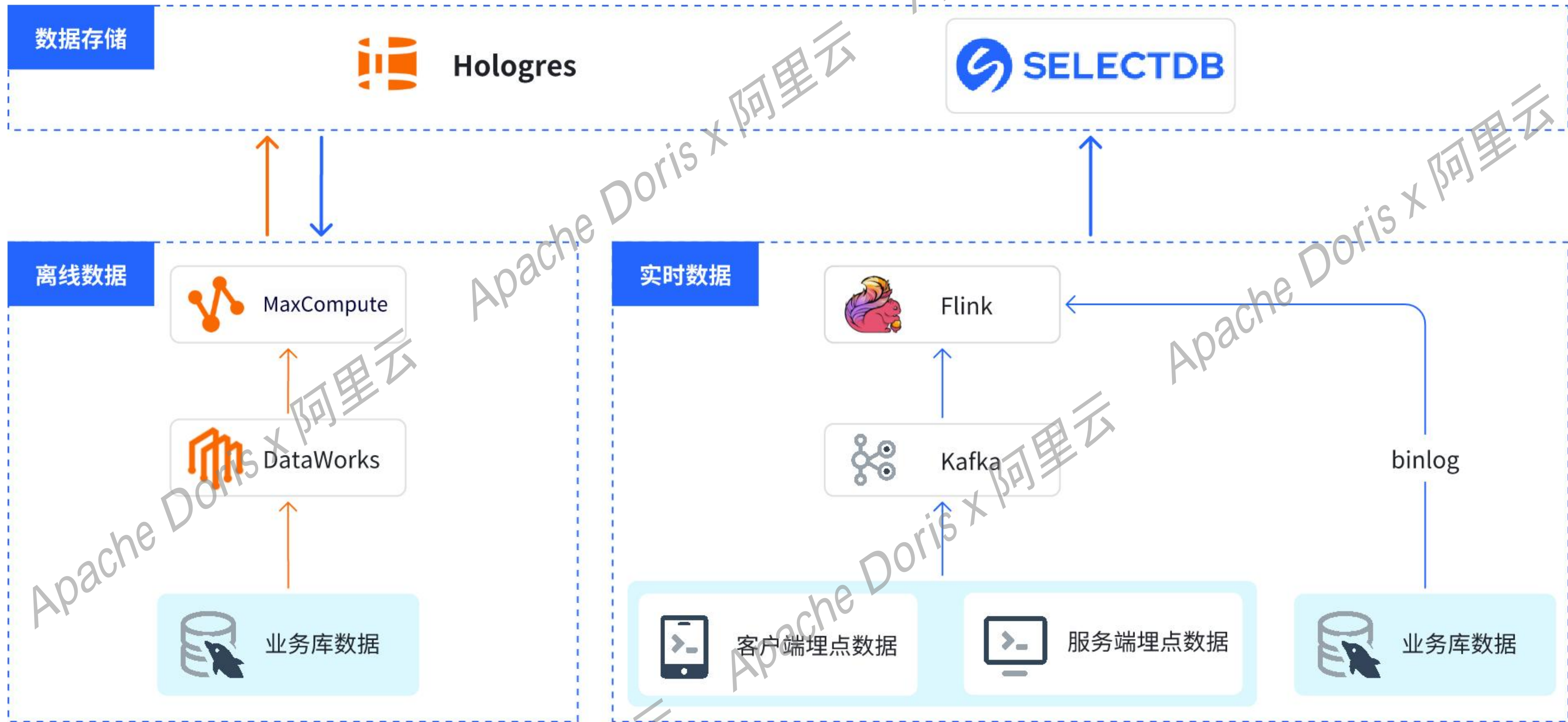
遇到的问题

- 引入新技术栈, 增加学习和使用成本
- 点查和圈选依赖数据源不统一

带来的收益

- 对数据表的修改和维护更加灵活
- 大表或多表 Join 查询效率提升明显
- 能支持跨库查询, 查询性能较好

架构v1.1



Apache Doris x 阿里云

Apache Doris x 阿里云

Apache Doris x 阿里云

Apache Doris x 阿里云

Apache Doris x 阿里云

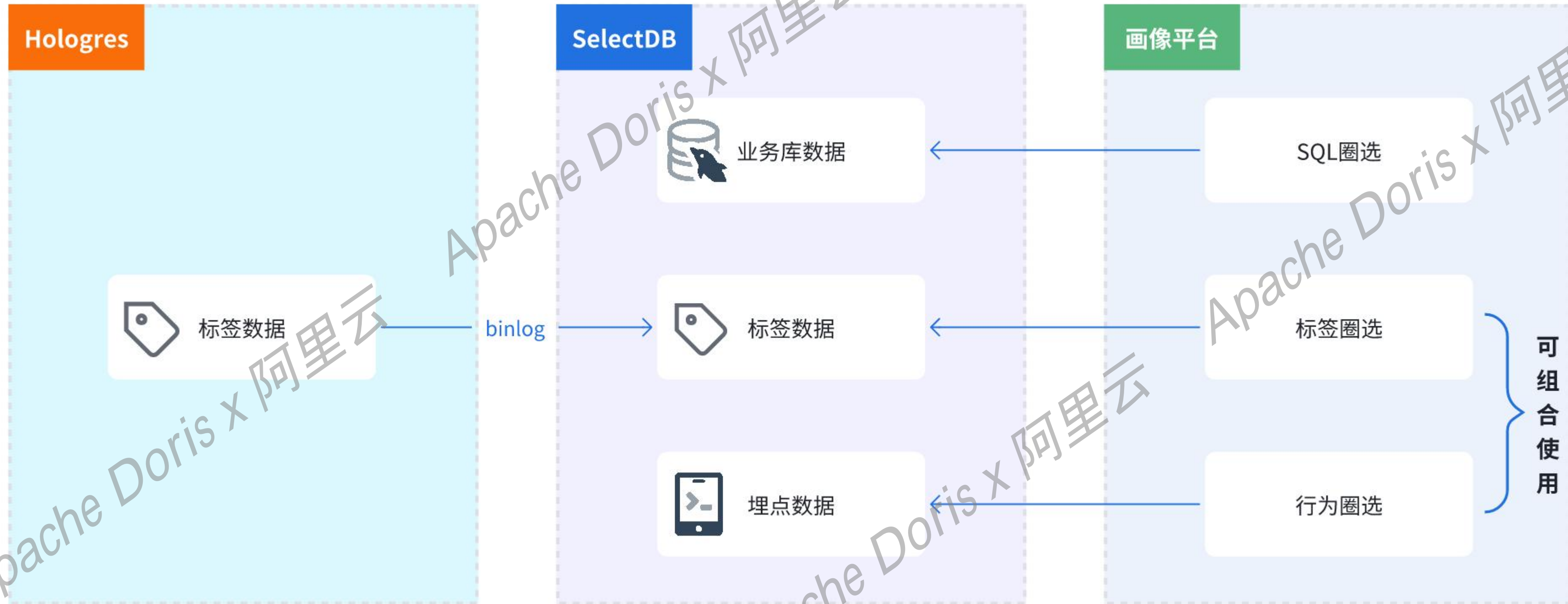
Apache Doris

三、场景实践

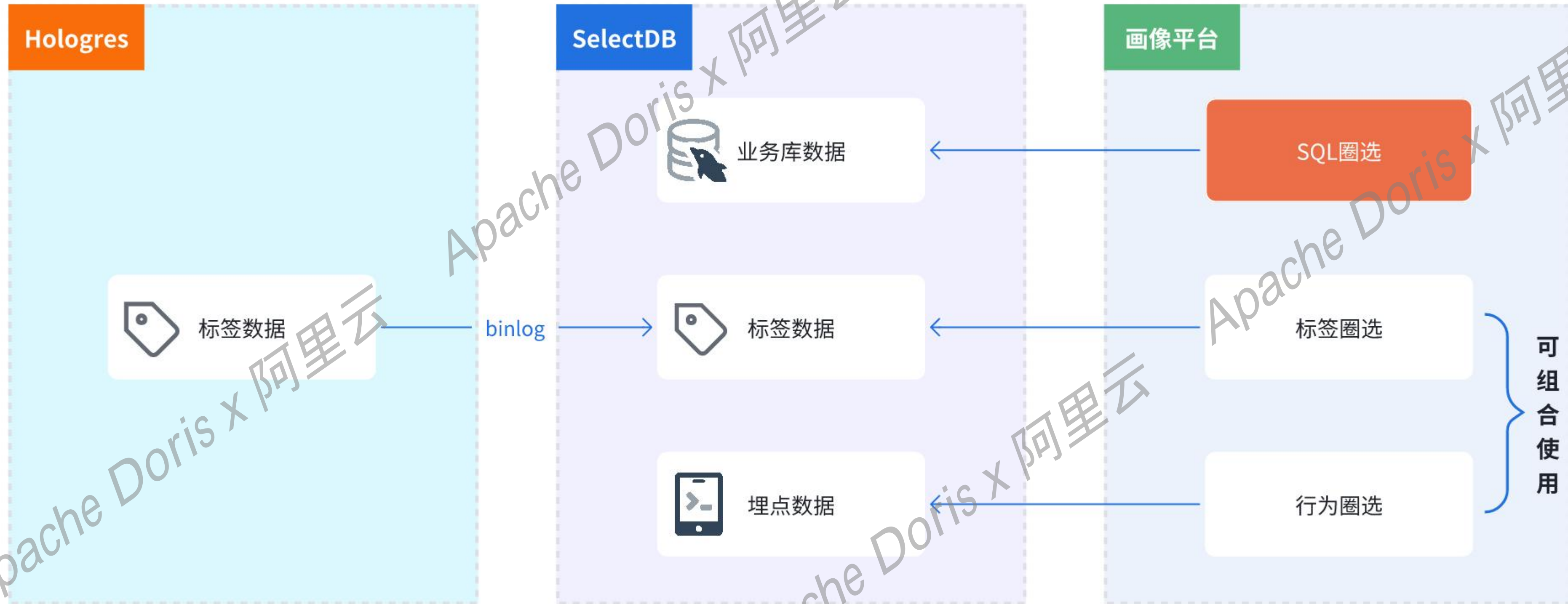
Apache Doris x 阿里云

Apache Doris x 阿里云

场景实践



场景实践



场景实践

应用场景*

精细化运营 数据推送

任务类型*

触发型 定时型

Hologres

SQL*

标

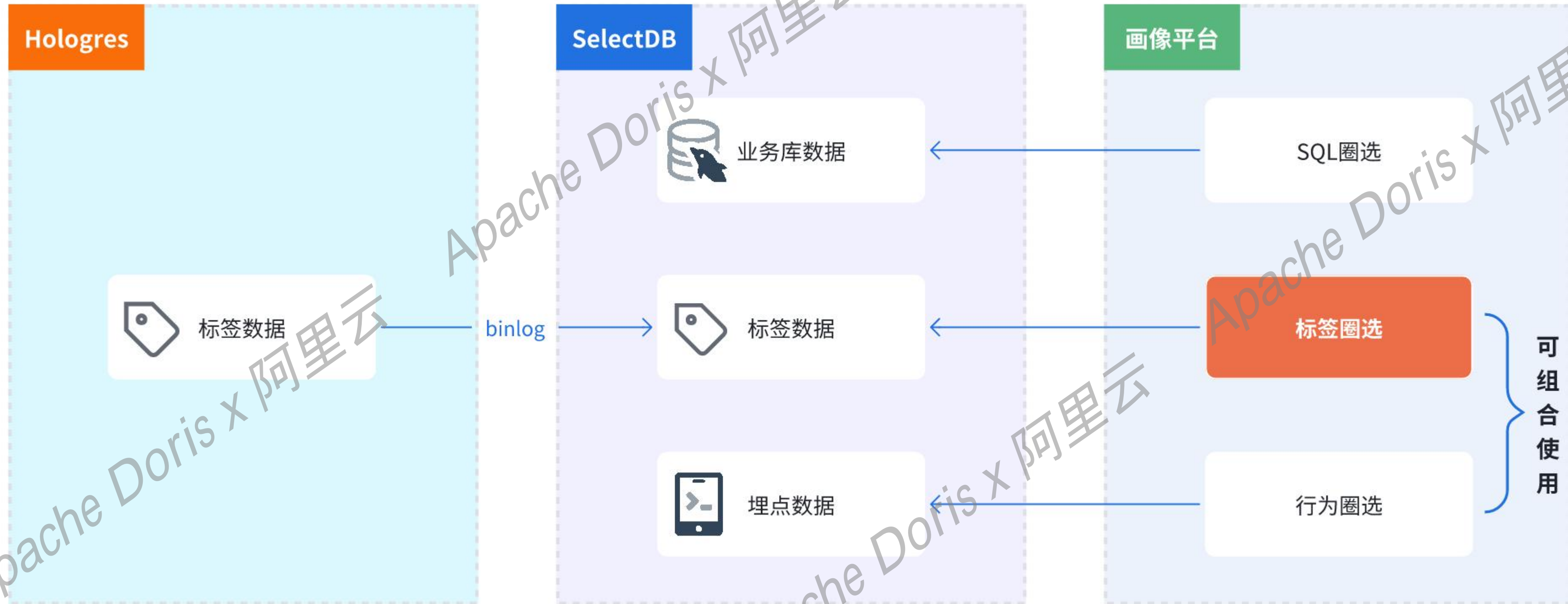
备注



```
select * from table_name where id > 1
```

备注

场景实践



场景实践

Hologres

标

的 单数 等于 0

并且

的 单数 等于 0

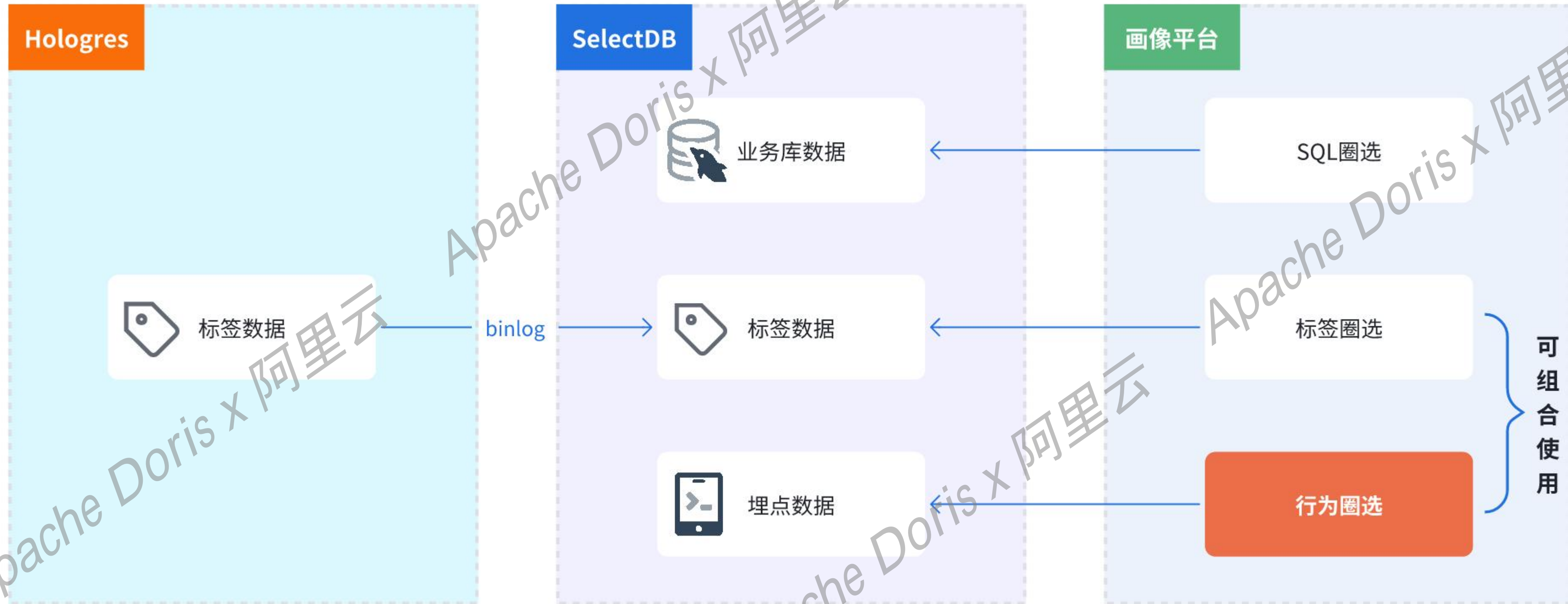
并且

的 单数 等于 0

并且

的 单数 等于 0

场景实践



场景实践

Hologres

设置用户行为 [帮助文档](#)

2024/10/16 17:13:14

且 \Leftrightarrow

2024-10-01 00:00:00 至 2024-10-12 00:00:00 做过 运营计划消息创建

精细化运营计划ID 等于 1522 X

2024-10-01 00:00:00 至 2024-10-12 00:00:00 未做过 运营计划消息触发

精细化运营计划ID 等于 1522 X

且 \Leftrightarrow

消息类型 等于 ai_out_call_sms X

2024/10/16 17:13:14

The screenshot displays the '设置用户行为' (Configure User Behavior) interface in Hologres. It features a sidebar with the 'Hologres' logo and a tag icon. The main area contains a configuration panel with a title '设置用户行为' and a '帮助文档' (Help Document) link. The configuration is structured as follows: a primary filter '且 \Leftrightarrow ' (AND) containing two sub-filters. The first sub-filter is a time range '2024-10-01 00:00:00 至 2024-10-12 00:00:00' with a status of '做过' (Done) and an action of '运营计划消息创建' (Create operation plan message). The second sub-filter is a field filter '精细化运营计划ID' (Refined operation plan ID) with a value of '1522 X' and a status of '等于' (Equal). The second sub-filter is also a time range '2024-10-01 00:00:00 至 2024-10-12 00:00:00' with a status of '未做过' (Not done) and an action of '运营计划消息触发' (Trigger operation plan message). The second sub-filter is also a field filter '精细化运营计划ID' (Refined operation plan ID) with a value of '1522 X' and a status of '等于' (Equal). A second primary filter '且 \Leftrightarrow ' (AND) contains a field filter '消息类型' (Message type) with a value of 'ai_out_call_sms X' and a status of '等于' (Equal). The interface includes a timestamp '2024/10/16 17:13:14' and a watermark 'Apache Doris x 阿里云'.

圈选场景

行为包配置

且 =>

2024-10-01 00:00:00 至 2024-10-12 00:00:00 做过 运营计划消息创建

精细化运营计划ID 等于 1522 X

且 =>

2024-10-01 00:00:00 至 2024-10-12 00:00:00 未做过 运营计划消息触发

精细化运营计划ID 等于 1522 X

且 =>

消息类型 等于 ai_out_call_sms X

SelectDB

```
CREATE TABLE behavior_rule_bitmap (  
  rule_md5 VARCHAR(200) NOT NULL COMMENT "用户行为规则MD5",  
  user_ids BITMAP BITMAP_UNION NULL COMMENT "该规则的用户集合"  
)  
AGGREGATE KEY(rule_md5)  
DISTRIBUTED BY HASH(rule_md5) BUCKETS 8;
```

MySQL

```
CREATE TABLE behavior_rule  
(  
  id int(11) unsigned NOT NULL AUTO_INCREMENT COMMENT 'ID',  
  rule_md5 varchar(180) NOT NULL DEFAULT '' COMMENT '用户行为规则MD5',  
  rule_define text COMMENT '规则表达式定义',  
  biz varchar(128) NOT NULL DEFAULT '' COMMENT '所属业务线',  
  rule_sql text COMMENT '根据表达式解析需要执行的sql',  
  compute_last_time timestamp DEFAULT CURRENT_TIMESTAMP not null comment '上次计算时间',  
  end_time timestamp DEFAULT CURRENT_TIMESTAMP not null comment '结束时间',  
  PRIMARY KEY (id),  
  UNIQUE KEY unique_rule_md5 (rule_md5)  
) ENGINE = InnoDB  
AUTO_INCREMENT = 0  
DEFAULT CHARSET = utf8mb4  
ROW_FORMAT = DYNAMIC COMMENT = '用户行为规则列表';
```

```
CREATE TABLE behavior_package_bitmap (  
  behavior_id VARCHAR(200) NOT NULL COMMENT "行为包id",  
  user_ids BITMAP BITMAP_UNION NULL COMMENT "该行为包的用户集合"  
)  
AGGREGATE KEY(behavior_id)  
DISTRIBUTED BY HASH(behavior_id) BUCKETS 8;
```

点查场景

```
CREATE TABLE user_tag
(
  user_id      INT NOT NULL,
  age          INT,
  gender       CHAR(2),
  city         VARCHAR(50),
  phone_md5    VARCHAR(32),
  register_date DATE,
  order_num    INT,
  first_order_date DATE,
  ...
) UNIQUE KEY(`user_id`)
COMMENT 'OLAP'
DISTRIBUTED BY HASH(`user_id`) BUCKETS 10
PROPERTIES (
  "file_cache_ttl_seconds" = "0",
  "colocate_with" = "tag_join_v2_group",
  "enable_unique_key_merge_on_write" = "true",
  "light_schema_change" = "true",
  "store_row_column" = "true",
  "group_commit_interval_ms" = "10000",
  "group_commit_data_bytes" = "134217728"
);
```

属性	含义	取值
file_cache_ttl_seconds	新导入数据在缓存中期望保持的时间	0
colocate_with	通过直接进行本地数据 Join, 减少数据在节点间的传输耗时	tag_join_v2_group
enable_unique_key_merge_on_write	在创建 Unique 表的语句中被设置为开启时, 对于主键的点查会采用短路径规划来对 SQL 执行进行优化, 仅需执行一次 RPC 即可完成查询	true
store_row_column		true
light_schema_change	主键点查的优化依赖了轻量级 Schema 变更中的 column unique id 来定位列	true
group_commit_interval_ms	大幅提升高并发小写入的性能	10s
group_commit_data_bytes		64MB

点查场景

SelectDB		
资源配置	缓存空间	内核版本
32c 256GB	2000GB	3.0.8
• BE调参		
参数名称	说明	设置结果
disable_storage_page_cache	开启 page cache	默认: true, 设置为: false
disable_storage_row_cache	开启行存	默认关闭: true 设置为: false
storage_page_cache_limit	page cache 使用百分比	设置为 40%
• 压测过程和结果		
调优策略	压测场景	指标情况
未做调优	200并发压测5分钟	avg(rt): 144ms avg(tps): 1369 max(tps): 1829 总请求数: 40.9w
调优后	200并发压测5分钟	avg(rt): 21ms avg(tps): 7460 max(tps): 8240 总请求数: 222.3w

Hologres		
资源配置	存储空间	内核版本
32 Core 128 GB	500GB	r2.1.9
• 压测过程和结果		
压测场景	指标情况	
200并发压测10分钟	avg(rt) 18.12ms max(tps) 4567 总请求数 270.7w	

结论
集群采用读写分离架构，在高并发点查场景下，查询耗时与 Hologres 接近，TPS 相对提升 63.35%，整体表现符合预期，计划未来逐步将点查切换至 SelectDB 上。

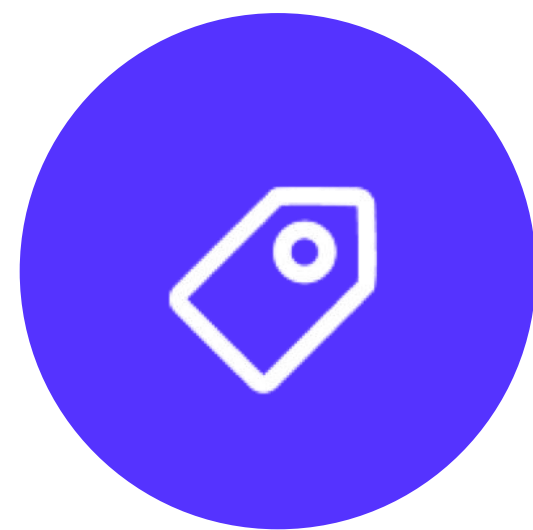
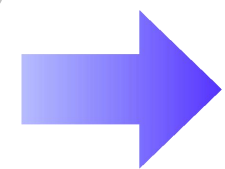
四、未来展望

未来展望



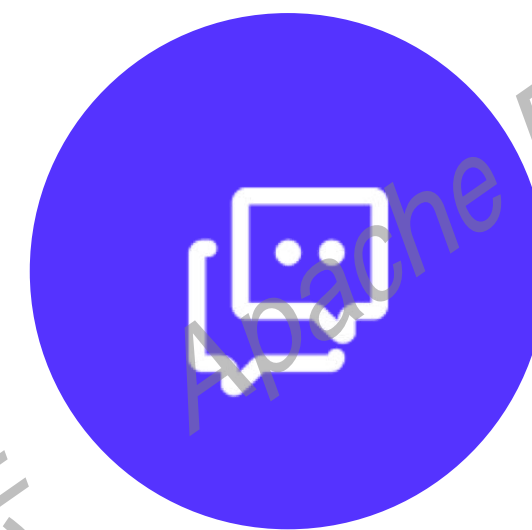
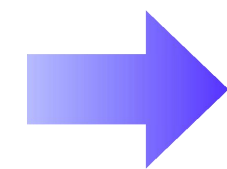
点查迁移

统一技术栈



实时标签

基于行为数据的实时标签改造



实时数仓

调研并完成实时数仓搭建

Thanks !



Apache Doris x 阿里云