墨天轮 | 中国数据库联盟 All China Database Union

DTC 2024.4.12-13

数据技术嘉年华

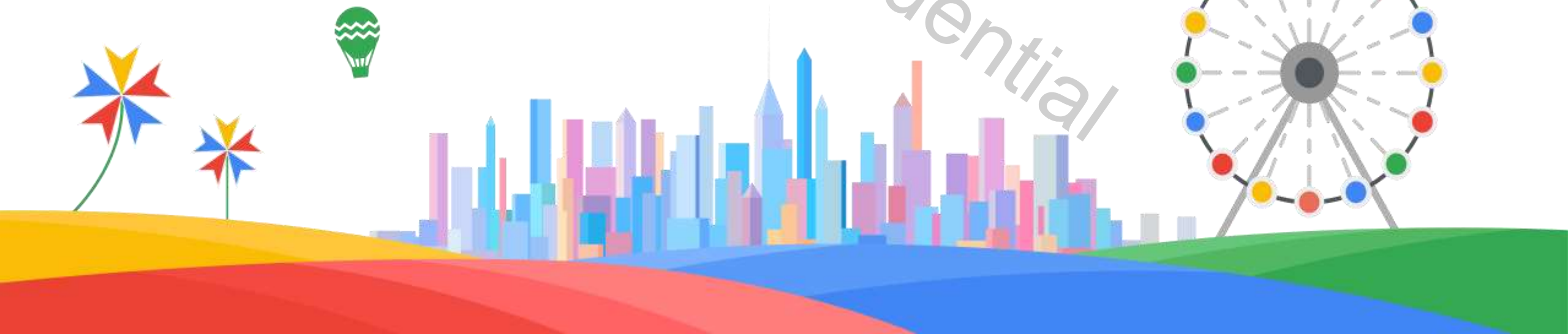
智能·云原生·一体化 —— DB与AI协同创新，模型与架构融合发展

Data Technology Carnival

墨天轮 中国数据库联盟 All China Database Union
DTC 2024

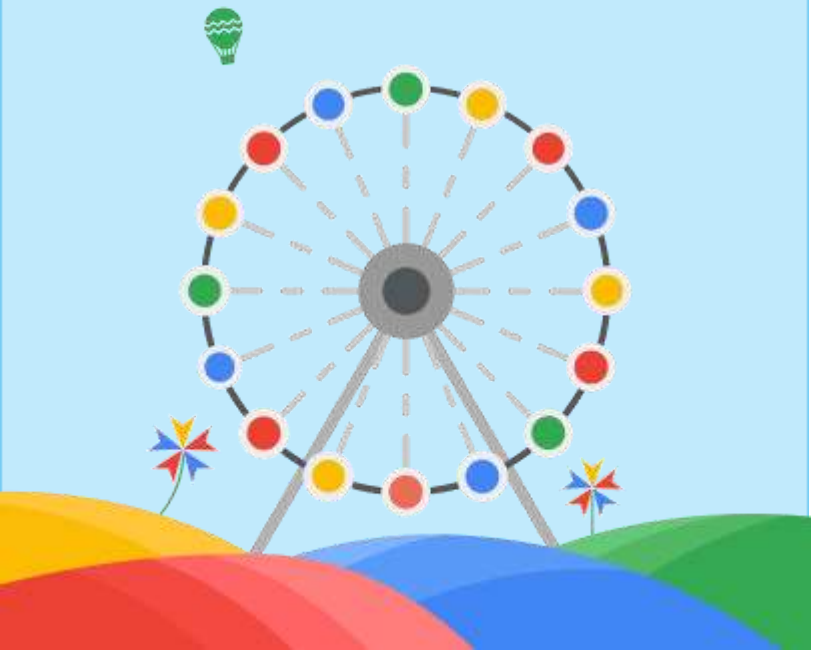
AntDB融合数据库实时流数据处理引擎揭秘

演讲人：亚信安慧-洪建辉



目录

CONTENTS



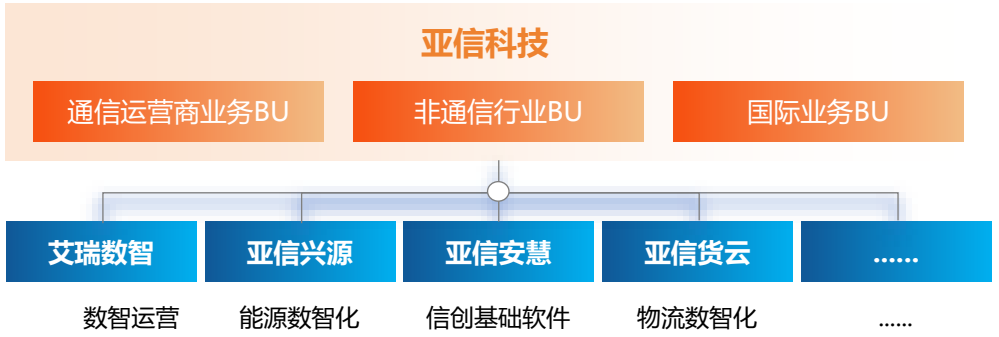
亚信科技-数智化全栈能力提供商

墨天轮 中国数据库联盟 All China Database Union
DTC 2024

人员规模

13000+ 员工
其中**90%+**为技术人员

近年来，亚信科技成功孵化出多个新业务主体，目前已形成“母公司+专业公司”的业务架构



市场地位

10亿+
中国通信行业业务支撑系统软件服务于全国终端用户数量

≈50%
中国通信行业业务支撑系统软件产品及相关服务市场份额

营业规模连年增长
单位：
人民币（亿元）

客户规模

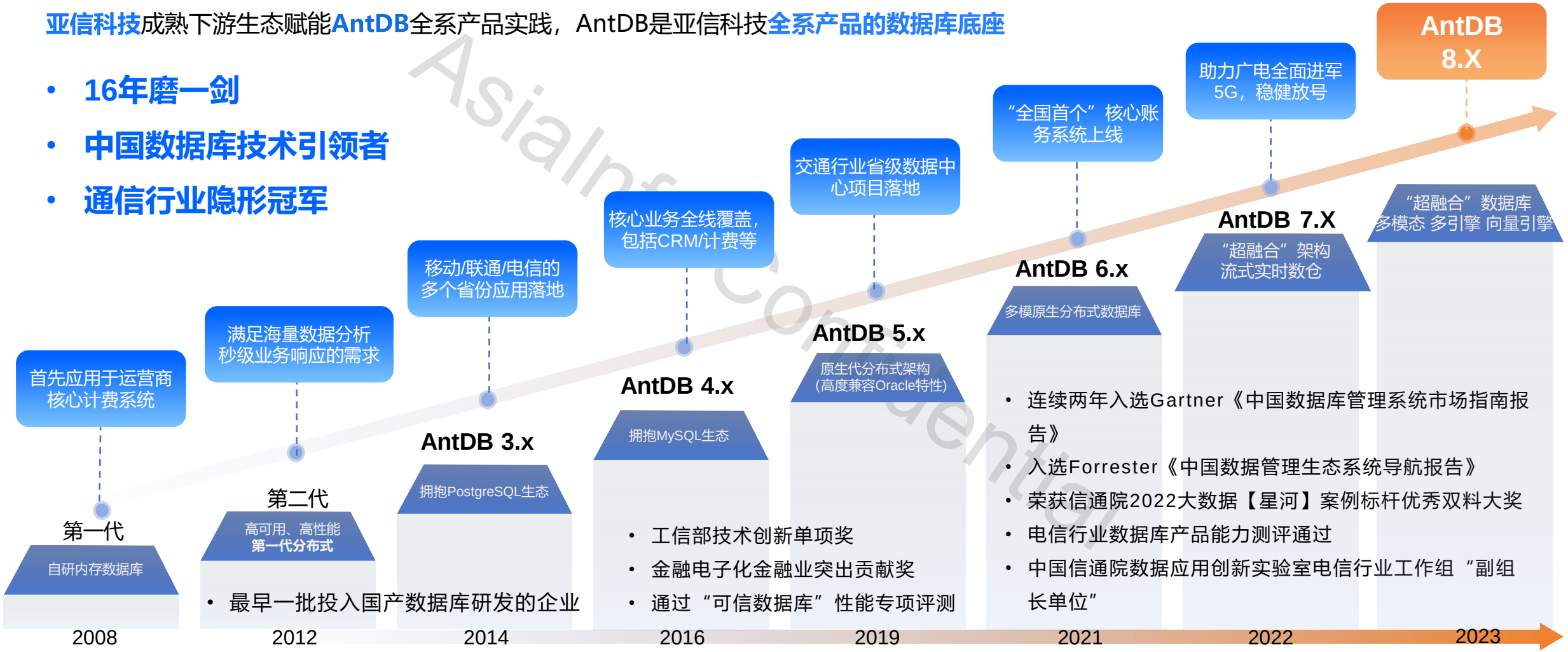
400+家 通信运营商客户[1]
70+家 能源客户
40+家 交通客户
60+家 政务客户
40+家 金融客户
10+家 邮政客户

AntDB融合数据库发展历程

墨天轮 中国数据库联盟 DTC 2024

亚信科技成熟下游生态赋能**AntDB**全系产品实践，AntDB是亚信科技**全系产品的数据库底座**

- **16年磨一剑**
- **中国数据库技术引领者**
- **通信行业隐形冠军**



AntDB 数据库超融合能力

墨天轮 中国数据库联盟 DTC 2024

为用户建立连接，融合6大数据应用需求能力

AntDB-S 流处理能力

AntDB-T 交易型能力

AntDB-M 内存计算能力

AntDB-A 分析能力

AntDB-V 向量引擎能力

AntDB-TS 时序型能力

可作为整体框架提供全部能力,也可以拆分为独立模块

02

AntDB流处理引擎内核揭秘

VS

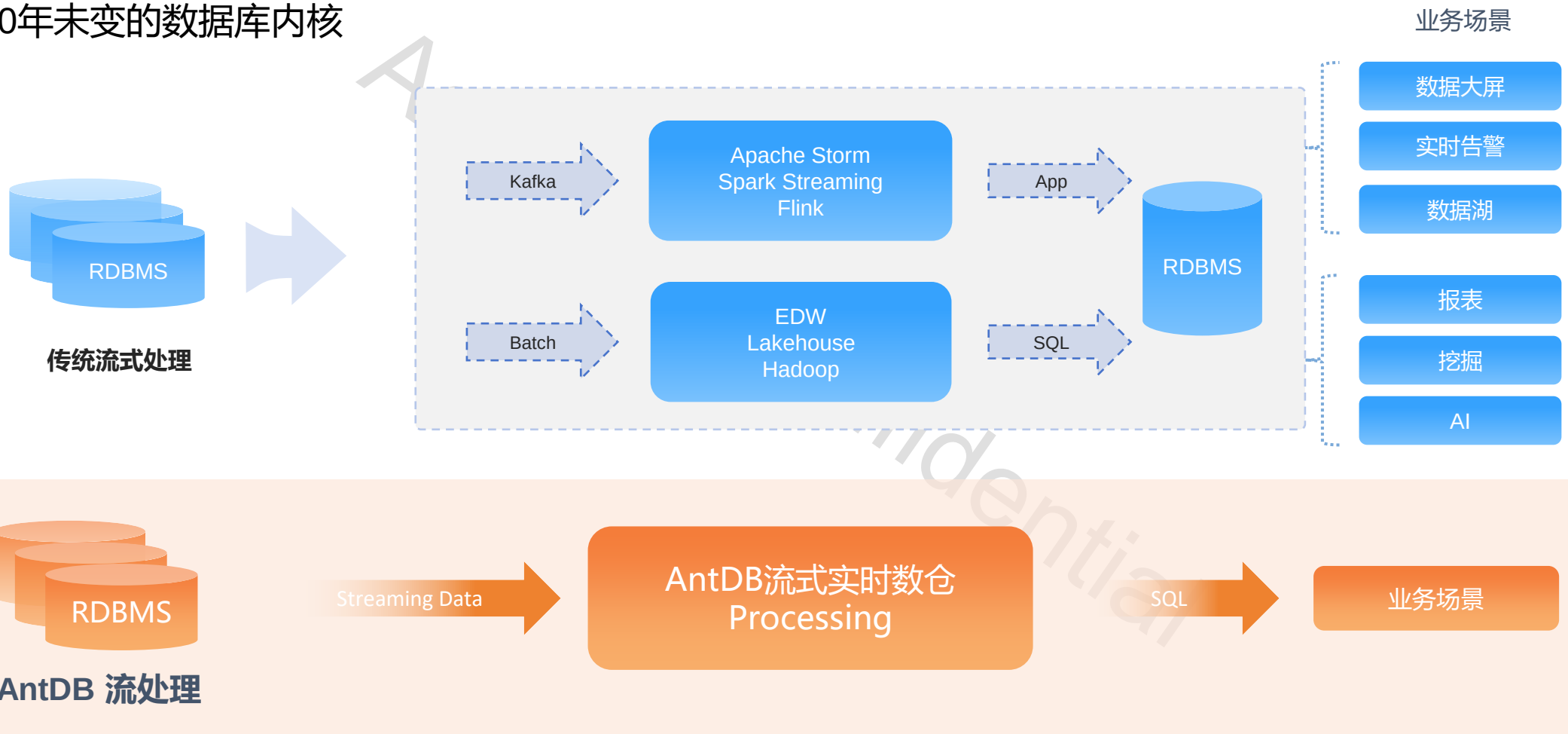
流处理框架

传统数据库框架

AntDB 流式数据引擎-大幅简化架构，大幅提升性价比

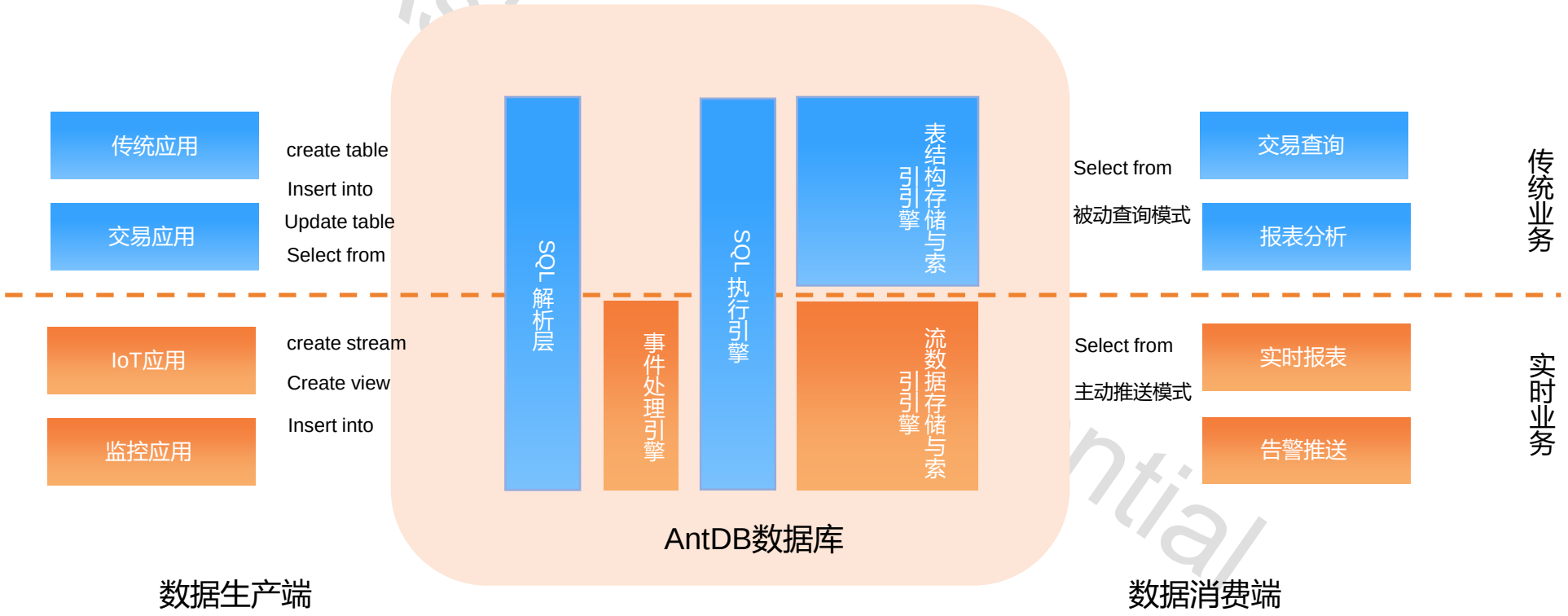
墨天轮 中国数据库联盟 DTC 2024

颠覆50年未变的数据库内核

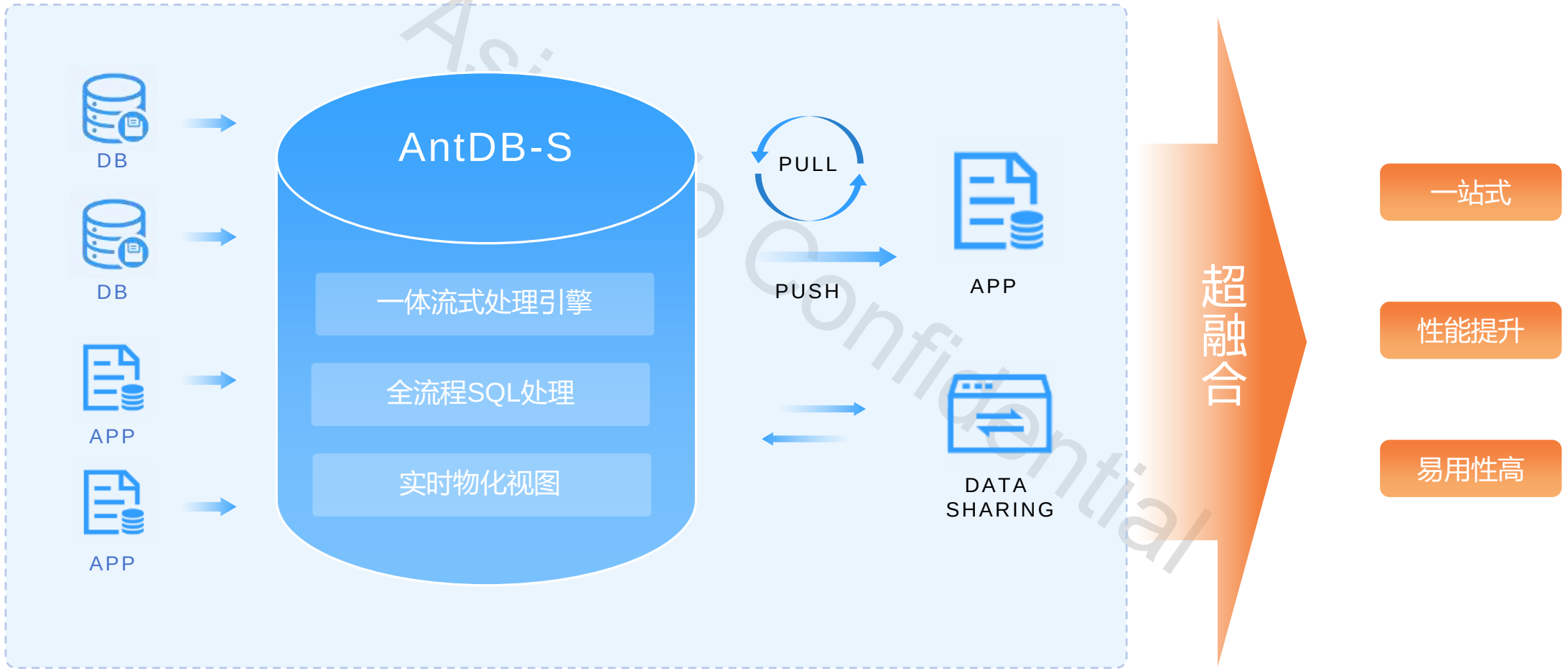


SQL语法与传统数据库全兼容，提供流式处理与推送能力

墨天轮 中国数据库联盟 DTC 2024



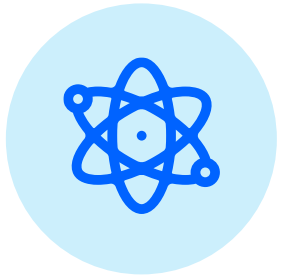
AntDB 流式数据引擎是如何工作的?



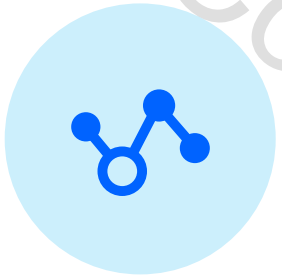
AntDB 流数据库特性优势(一)

针对核心系统性能需求

在AntDB数据库运行，提升>3倍



数据流转
更实时



处理引擎
更简易

处理模型
更对应

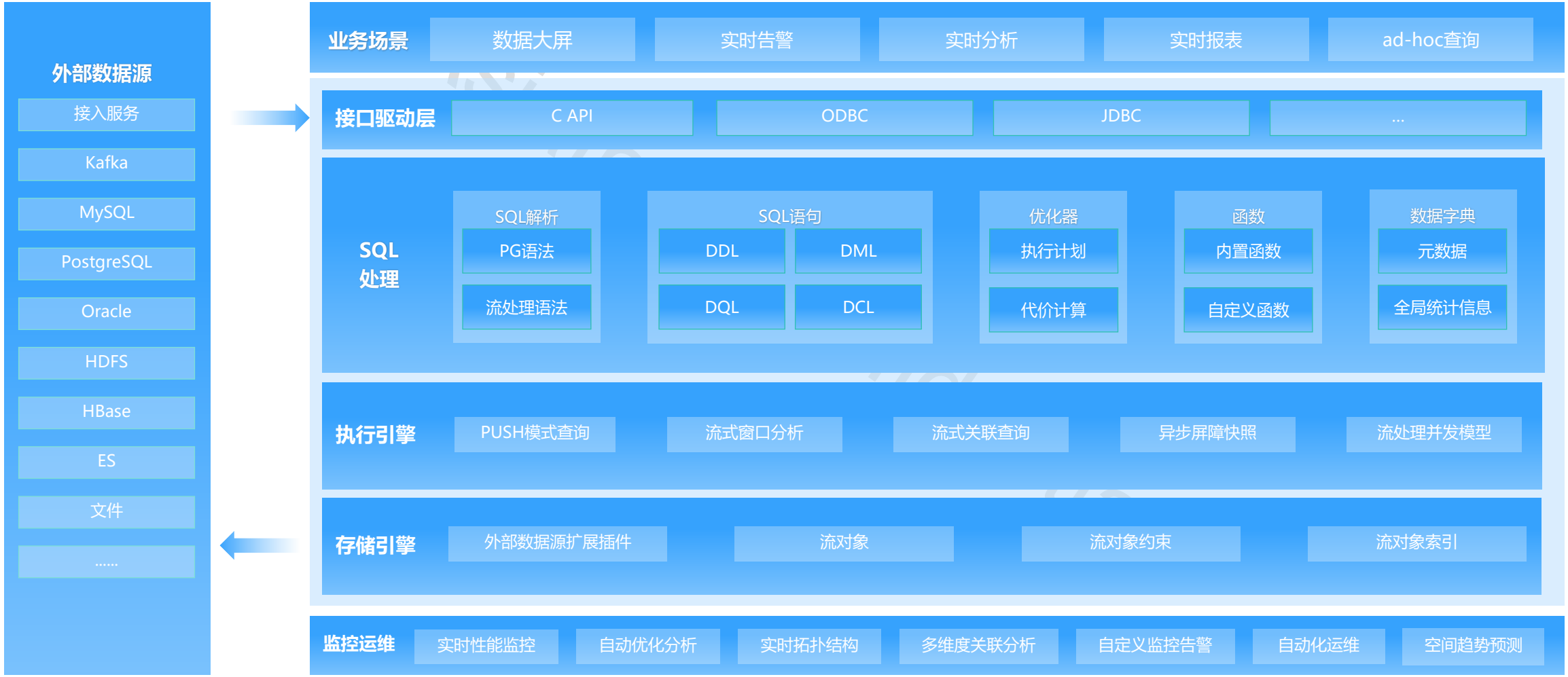
AntDB 流式数据库特性优势

技术堆栈简化：一套技术栈，覆盖数据库应用中的各类场景。一体化的流处理数据库引擎，彻底将流式计算与传统交易、分析型数据存储进行了融合

数据实时流转：原生支持**流流关联**与**流表关联**，打破流引擎与数据库之间的壁垒。

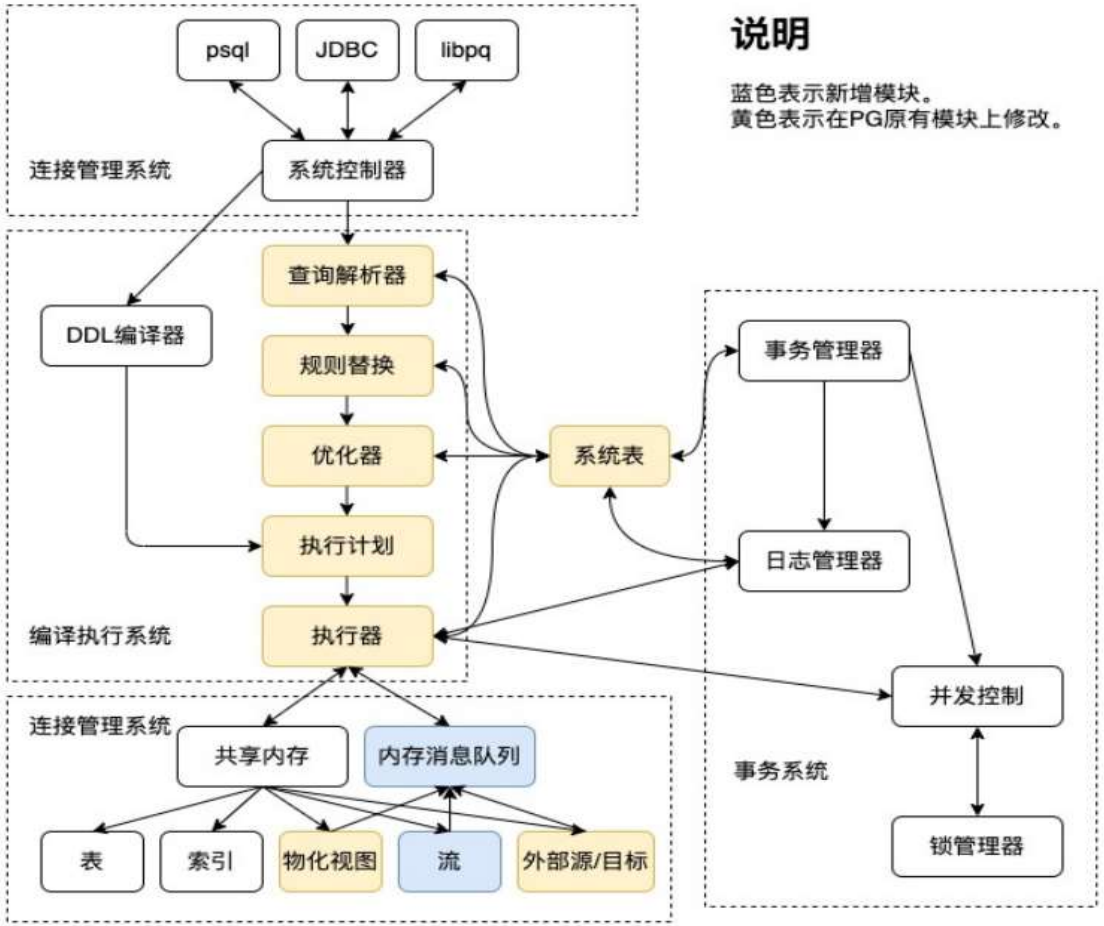
SQL简易：数据在数据库内部，流对象和表对象之间自由流转，用户可以随时通过建立索引、流表关联、触发器、物化视图等方式，对数据进行处理及业务逻辑定制。

AntDB 流数据库核心功能和逻辑模块



AntDB 流数据库功能架构拆解

流式数据库是把流处理引擎的能力合并到数据库内核，与数据库SQL引擎、存储引擎融合在一起，完全以数据库的习惯使用流处理引擎，并且可以和数据库的功能混合使用。



- **接口驱动层面**

  为了方便数据库用户快速切入流数据库，我们针对jdbc、odbc、libpq等接口驱动增加了实时数据推送功能并且完全兼容现有的接口。用户可以像使用传统数据库一样调用jdbc、odbc、libpq接口执行SQL语句使用流数据库。

- **SQL查询引擎层面**

  增加了流处理特有的语法，对优化器以及元数据部分都针对流式处理做了相应修改。

- **执行引擎层面**

  增加了PUSH模式大大的提高了流处理的实时性；流式窗口分析算子支持滚动窗口、滑动窗口、会话窗口；流式关联查询支持双流join以及流表join；异步屏障快照保证了流处理数据的精确唯一性；为流处理特殊设计的并发模型保证了流处理的性能。

- **存储引擎层面**

  增加了流对象存储用于实时存储流数据；为保证流数据的完整性、一致性和正确性在流对象存储上增加了流约束；为了提高双流join的性能增加了流索引，增加外部数据源扩展插件用于读写多种外部数据源，以支持现有流式处理系统的生态，从而无缝替代现有的流式处理系统Flink等。

流式数据处理示例

1、创建流对象

```
benchmark=# CREATE STREAM instructor1 (intime char not null default hlcnextval()ext, salary float);
CREATE STREAM
benchmark=# CREATE STREAM super_instructor as SELECT * FROM instructor WHEREsalary >= 80000 EMIT CHANGES;
```

2、插入数据

```
benchmark=# insert into instructor values('1001','john','dept1',57000);
INSERT 0 1
benchmark=# insert into instructor values('1001','tom','dept1',17000);
INSERT 0 1
benchmark=# insert into instructor values('1003','jack','IT',27000);
INSERT 0 1
```

3、推送流式数据处理结果

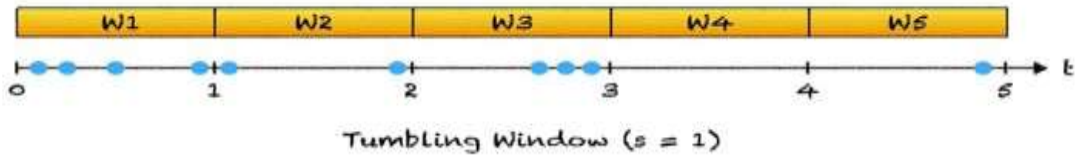
```
benchmark=# select count(id),sum(salary),dept_name from instructor group by dept_name emit changes
 count |  sum  | dept_name
-------+-------+-----------
     2 | 34000 | IT
     1 | 87000 | IT1
     2 | 74000 | dept1
```

流数据窗口介绍

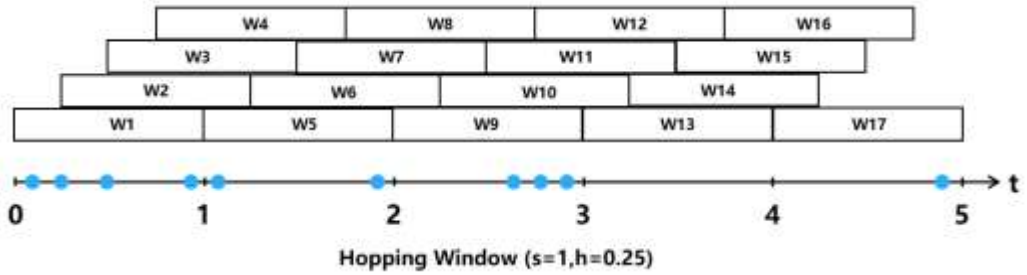
窗口操作是流式系统进行数据流处理的核心，通过窗口操作，可以将一个无限的数据流拆分成很多个有限大小的“桶”，然后在这些桶上执行计算。

流式数据库提供了四种类型的窗口定义：滚动窗口、滑动窗口、会话窗口和全局窗口

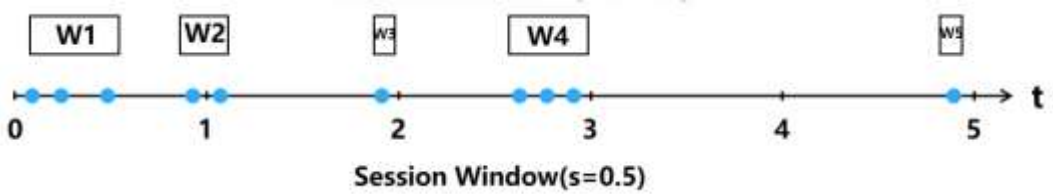
•滚动窗口(Tumbling Window)



•滑动窗口（Hopping Window）



•会话窗口（Session Window）



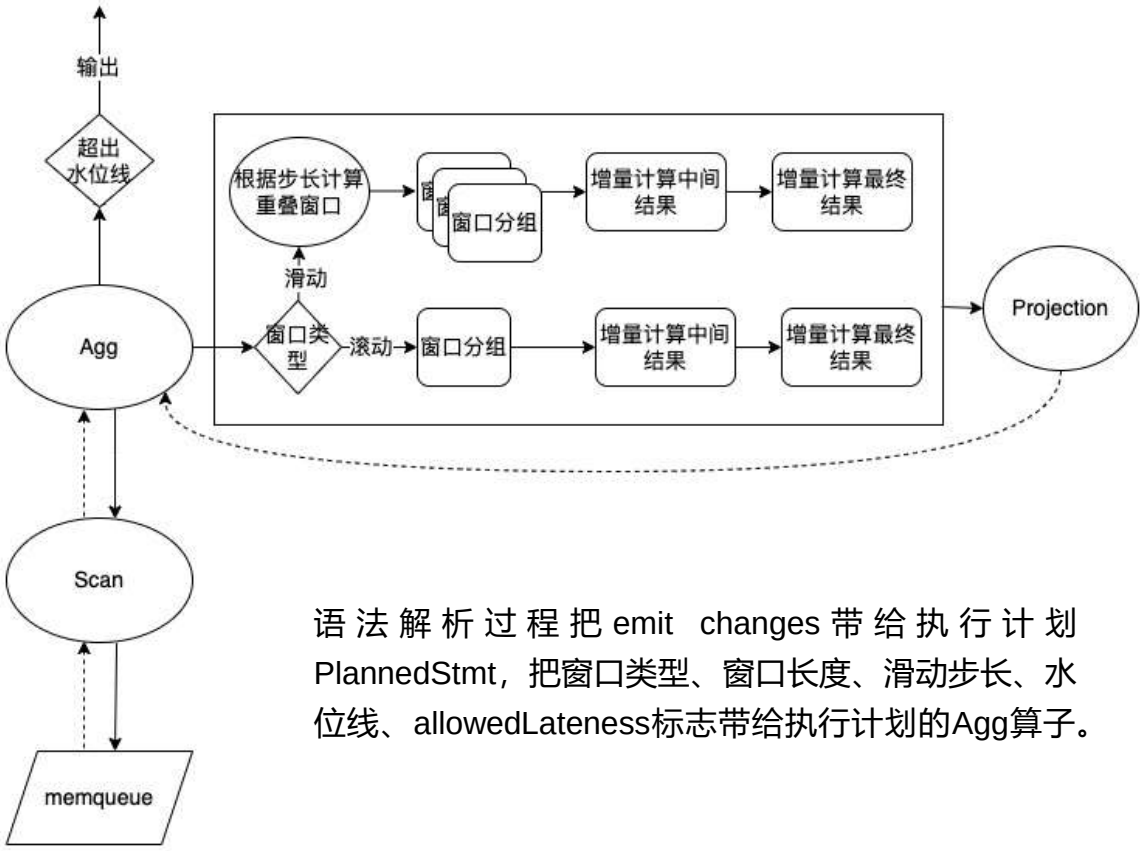
- 全局窗口

全局窗口（Global Window）只有一个窗口且窗口无限大，也就是无窗口定义，因为没有窗口结束时间所以不能等窗口结束后输出统计结果，一有数据立即计算输出结果。

流式窗口处理介绍

墨天轮 中国数据库联盟 DTC 2024

流式窗口统计架构图：



语法解析过程把emit changes带给执行计划PlannedStmt，把窗口类型、窗口长度、滑动步长、水位线、allowedLateness标志带给执行计划的Agg算子。

流式窗口统计语法：

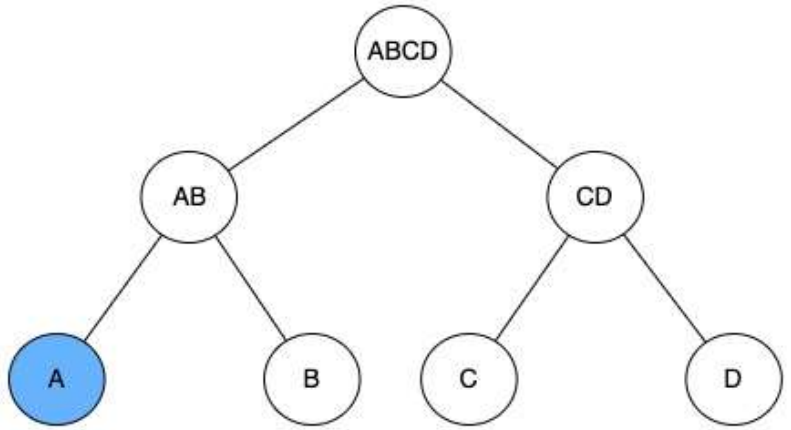
```
SELECT expression [ [ AS ] output_name ] [, ...]
   [, WindowBegin() [[ AS ] output_name]
  [, WindowEnd() [[AS] output_name]] ]
FROM stream_name
[ WHERE condition ]
GROUP BY grouping_element
[[ { TUMBLE ( event_time_field, <window_size> ) |
HOP ( event_time_field, <window_size>, <slide_size> ) }
[ { WATERMARK | DELAY } <watermark_size> ][ALLOWEDLATENESS
<allowedlateness_size>] EMIT CHANGES] | [EMIT CHANGES]
<window_size> <slide> <watermark_size> are interval 'quantity unit'
```

流表JOIN连接

流表join指的是一个流和一张或多张表join，流表join的结果也是流。

```
SELECT [ * | expression [ [ AS ] output_name ] [, ...] ]
FROM {stream_name | table_name} [ [ AS ] alias ]
JOIN {stream_name | table_name} [ [ AS ] alias ]
[ ON join_condition | USING ( join_column [, ...] ) ]
[ WHERE condition ] [ EMIT CHANGES ]
```

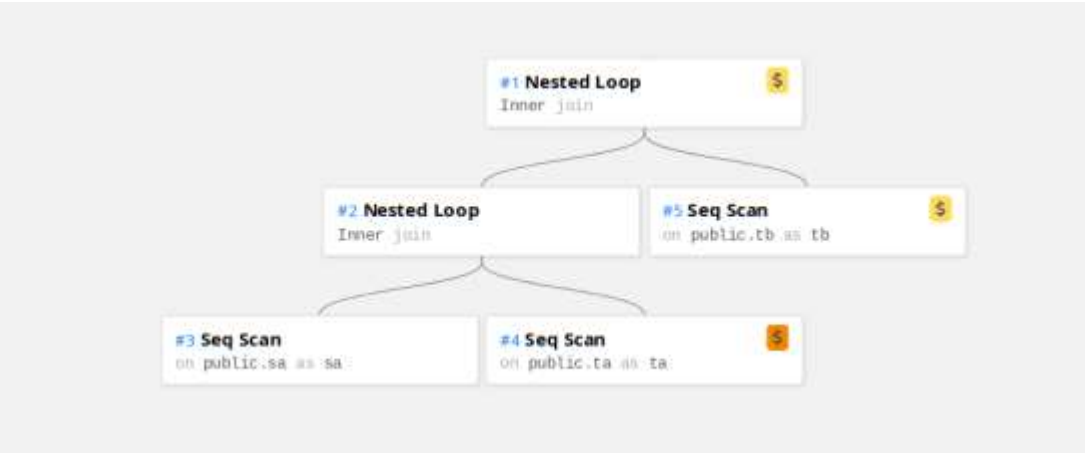
由于流是push模式获取增量数据，表是pull模式获取历史数据，执行器执行过程是先获取执行计划左子树再获取右子树，要实现流表join的效果是从流中获取一条tuple就在表中查找符合条件的tuple返回连接结果给上游节点实时推送给客户端，所以需要优化器生成计划树是保证流在执行计划树的最左叶子结点并且流表join的节点类型是NestLoop。

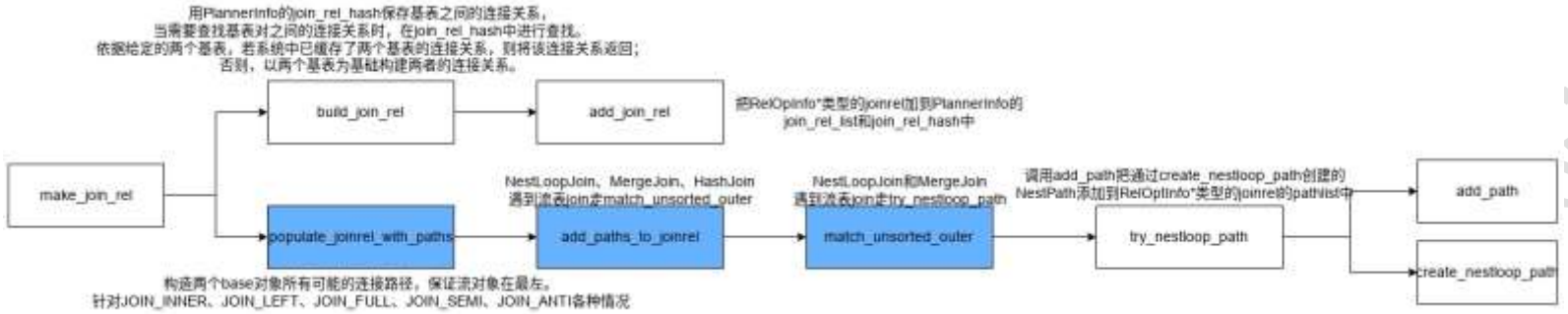


A是流，B、C、D是表。

流表join执行计划

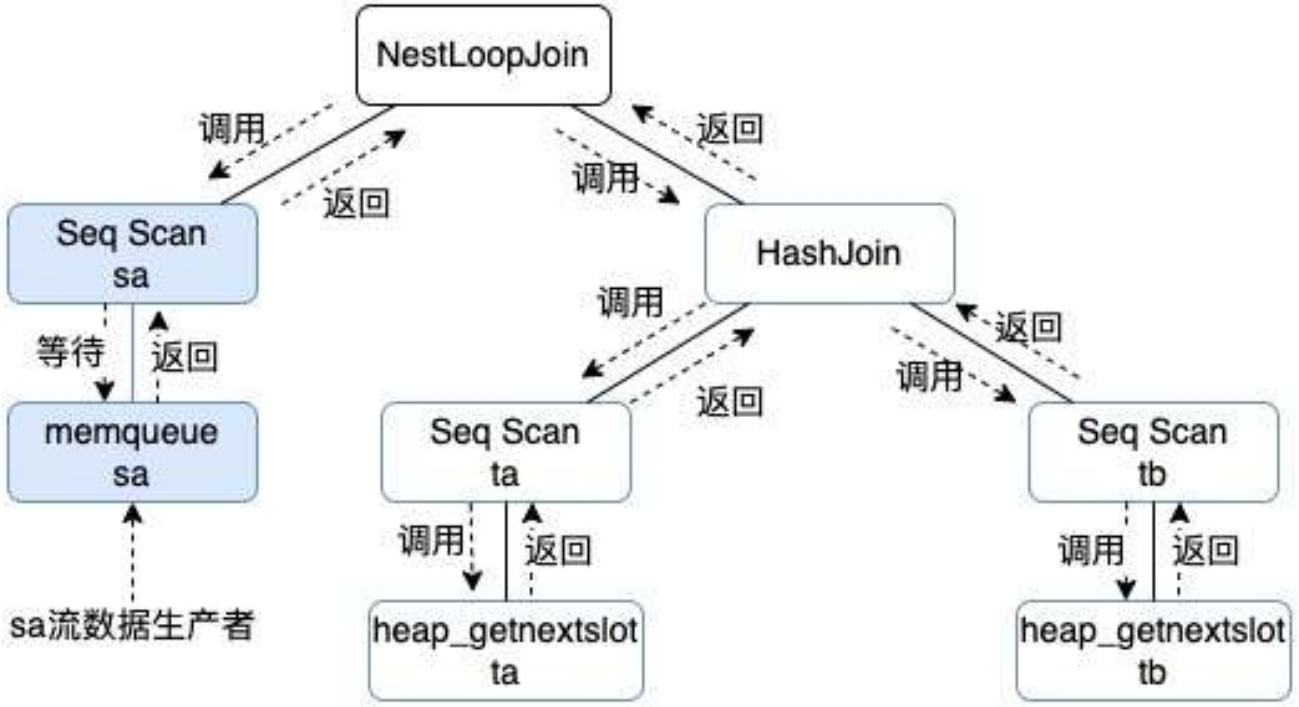

```
mydb=# explain select * from sa join ta on sa.id = ta.id join tb on sa.id = tb.id
emit changes;
                        QUERY PLAN
-----------------------------------------------------------------
 Nested Loop  (cost=0.00..128.63 rows=6 width=52)
   Join Filter: (sa.id = tb.id)
   ->  Nested Loop  (cost=0.00..90.06 rows=1 width=16)
         Join Filter: (sa.id = ta.id)
         ->  Seq Scan on sa  (cost=0.00..1.01 rows=1 width=8)
         ->  Seq Scan on ta  (cost=0.00..89.02 rows=2 width=8)
   ->  Seq Scan on tb  (cost=0.00..22.70 rows=1270 width=36)
(7 rows)
```





改造优化器，在构造两个base对象所有可能的连接路径时，遇到流和表，保证流对象在左边，并且创建NestLoop Join路径。物理优化阶段构造两个base对象之间连接关系的代码结构。

流表JOIN执行器交互过程



执行器优化过程

数据库的执行器针对JOIN算子是

- 先获取左子节点的计算结果
- 然后获取右子节点的计算结果进行关联，不断迭代

如果遇到流，则获取对应的消息队列记录和右子树获取的表记录进行关联，如果消息队列为空则等待。

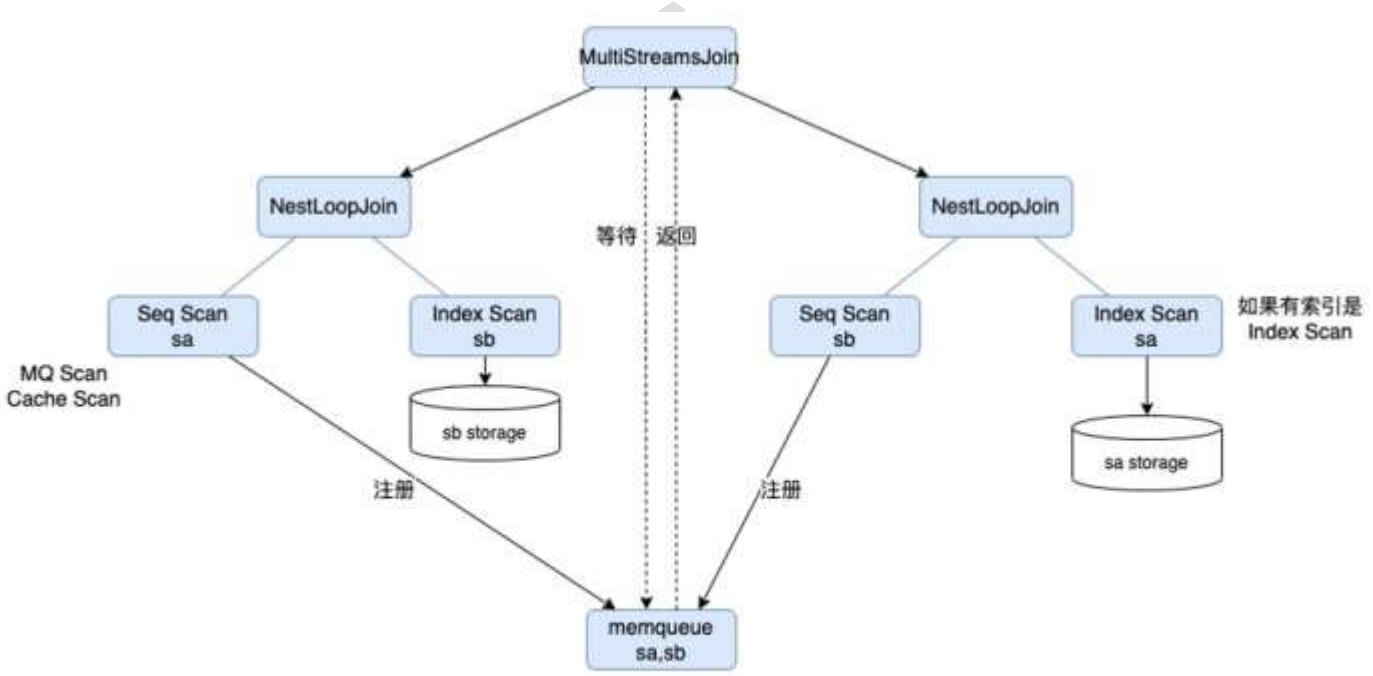
双流join执行计划

双/多流join有三种类型：常规join，区间join和窗口join。

- 多流常规join是最常见的一种join类型，join左边流的新增记录会和右边流的历史以及新增记录进行关联，右边流的新增记录也会和左边流的历史以及新增记录进行关联。
- 多流区间join表示左边流的新增记录和前后指定时间范围内的右边流的记录进行关联。
- 多流窗口join表示在指定时间窗口范围内，左流新增记录和右流历史及新增记录关联，右流新增记录和左流历史及新增记录关联。

```
mydb=# explain select sa.id said, sb.id sbid, sa.name, sb.age from sa,sb where sa.id = sb.id emit changes;
                          QUERY PLAN
---------------------------------------------------------------------
MultiStream  (cost=0.00..77302.20 rows=14351 width=44)
  ->  Nested Loop  (cost=0.00..77302.20 rows=14351 width=44)
        Join Filter: (sa.id = sb.id)
        ->  Seq Scan on sa  (cost=0.00..22.70 rows=1270 width=36)
        ->  Seq Scan on sb  (cost=0.00..32.60 rows=2260 width=8)
  ->  Nested Loop  (cost=0.00..77302.20 rows=14351 width=44)
        Join Filter: (sa.id = sb.id)
        ->  Seq Scan on sb  (cost=0.00..32.60 rows=2260 width=8)
        ->  Seq Scan on sa  (cost=0.00..22.70 rows=1270 width=36)
(9 rows)
```

双流JOIN执行过程



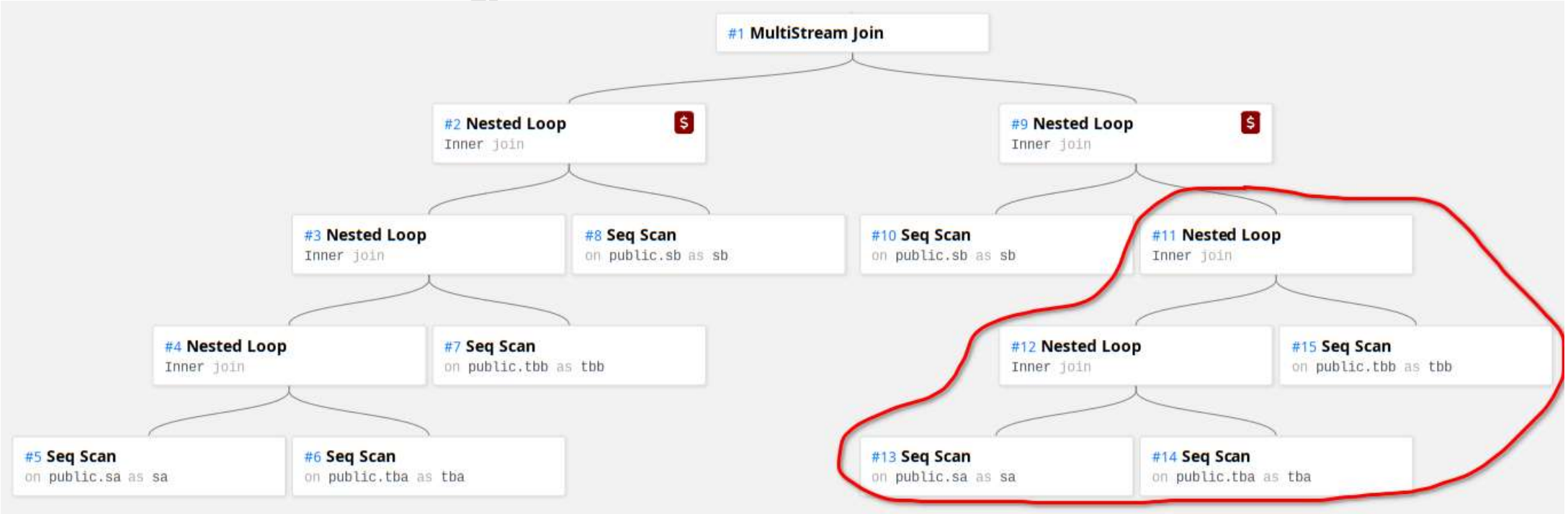
MultiStreamsJoin是新增算子节点，左右子树都是NestLoopJoin。
左子树NestLoopJoin算子以sa做为流sb做为表，右子树NestLoopJoin算子以sb做为流sa做为表。
MultiStreamsJoin获取memqueue的消息，如果是sa的Tuple，执行左子树NestLoopJoin算子；如果是sb的Tuple，执行右子树NestLoopJoin算子。
NestLoopJoin算子操作和流表join一样。

执行器优化过程

双流join的顶层算子是MultiStreamJoin，其左右子树都是NestLoopJoin算子，左子树表示以sa为流sb为表做流表join，右子树表示以sb为流sa为表做流表join，左右子树NestLoopJoin的结果就是MultiStreamJoin的结果。

多流多表JOIN执行计划

多流多表join其实就是流表join和多流join的结合。下面以双流sa、sb和双表tba、tbb做常规join为例说明执行计划。



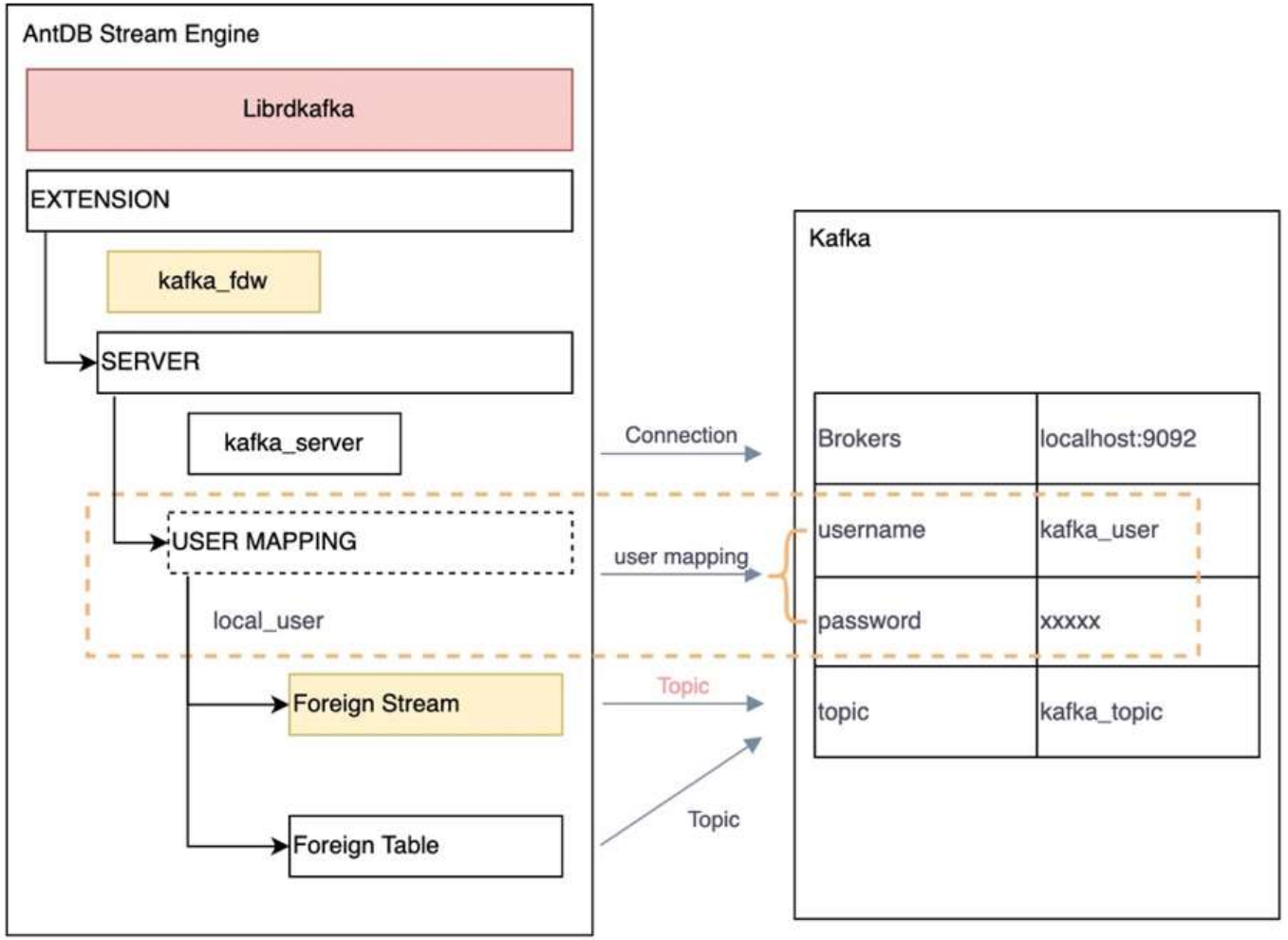
双流双表join的顶层算子依然是MultiStreamJoin，其左右子树都是NestLoopJoin算子，由于涉及多个对象join所以是多个join算子嵌套，左子树表示以sa为流sb、tba、tbb为表做流表join，右子树表示以sb为流sa、tba、tbb为表做流表join，如果某个子树下面涉及的对象都是表则优化器根据代价估算生成最优的join类型，比如图中红线圈出的部分，根据代价估算有可能生成HashJoin或者MergeJoin，一个流和多个表join的结果也是流，它们合并的结果就是MultiStreamJoin的结果。

访问外部流数据

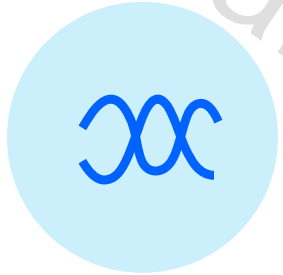
AntDB新增FOREIGN STREAM对象，用于处理外部流数据服务

创建外部表步骤如下：

1. CREATE EXTENSION:创建插件，修改kafka_fdw，ForeignScan支持消费kafka topic并插入数据队列
2. CREATE SERVER:创建服务
3. CREATE USER MAPPING:创建用户映射
4. CREATE FOREIGN STREAM/FOREIGN STREAM:创建外部表/外部流



AntDB流式数据引擎产品能力迭代方向



弹性伸缩能力

支持更多类型的外部表

更长的watermark

03 案例：电信领域核心系统中的应用

CRM核心系统全域数据库替换选型依据

墨天轮 中国数据库联盟 DTC 2024

- **选型准备：因地制宜，从“三个维度、八大要素”选取和构建最合适的技术体系**

自主维度
- 企业资质
- 知识产权
- 自主程度

技术维度
- 研发侵入度
- 运维侵入度

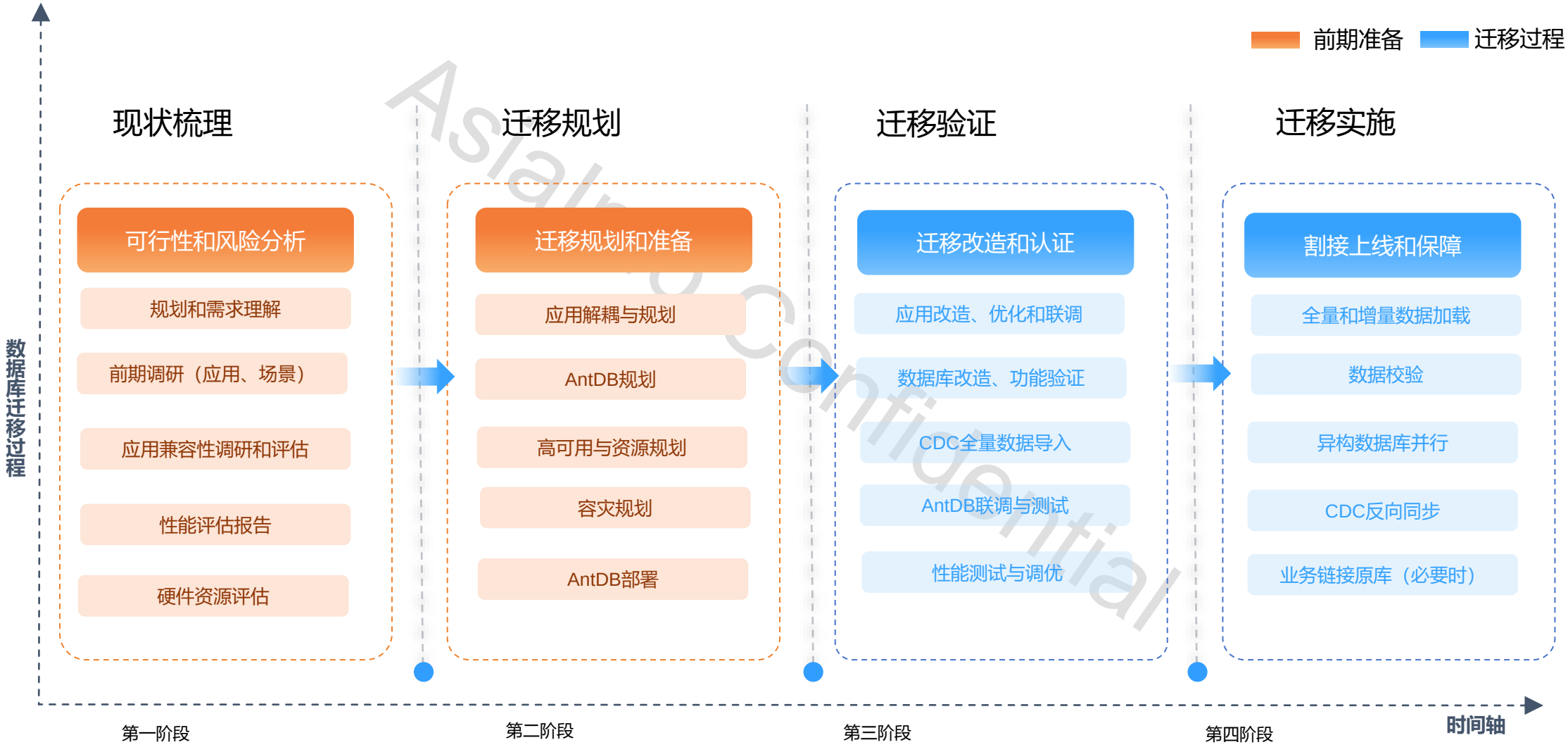
生态维度 ★
- 技术生态
- 产品生态
- 合作生态

评估数据库自主可控
- 技术路线要能突破高并发、高连接的重磅场景

评估研发运维侵入度
- 功能均衡，具有较强的API兼容性、更高性价比的支持和更强大的可扩展性

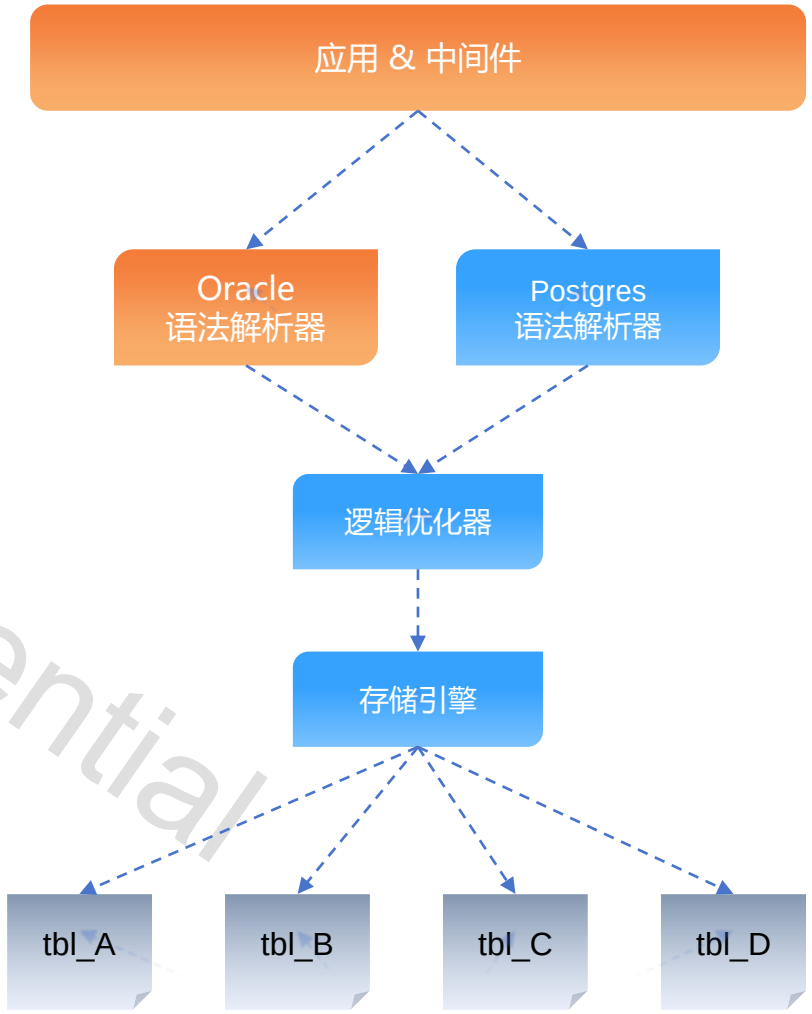
评估生态成熟发展
- 社区繁荣，开放度高，生态增长快

CRM核心系统迁移到AntDB数据库时间计划流程



AntDB 的 Oracle 兼容能力助力应用迁移

AntDB 在原生 Postgres 语法解析基础上，构建了独立的 Oracle 语法解析器。两套语法解析引擎共存，上层 “应用 & 中间件” 则可按需选择合适的语法模式。



架构特点：

- ✓ 独立的 Oracle 语法解析引擎，兼容适配更方便快捷
- ✓ 对原生 Postgres 解析引擎无侵入，完全不影响原生功能
- ✓ 双解析引擎共存，能适配更多的复杂场景
- ✓ 可在多种范围提供 Oracle 兼容能力

多范围兼容：

- ✓ **语句级**：支持通过 Hint 在语句级启用Oracle兼容
- ✓ **会话级**：支持通过会话参数，在当前会话级启用 Oracle 兼容
- ✓ **全局级**：支持通过数据库全局参数，在全库启用 Oracle 兼容

CRM核心系统全域数据库替换技术创新

墨天轮 中国数据库联盟 DTC 2024

SQL多模态解析引擎，极大提升Oracle兼容性

AntDB 6.X在原生Postgresql语法解析基础上构建了独立的Oracle语法解析器，实现对Oracle**语法、函数、特性**的多方面兼容能力，**减少了超过2万处代码修改**，大幅降低应用适配改造的工作量。

	兼容点	兼容说明	对业务影响
语法兼容	**外关联 (+)**	**兼容 Oracle 特有的 (+) 外关联语法**	**支持带 (+) 外关联的 SQL 直接运行，且结果准确**
	connect by	兼容 Connect By 层次查询，且**支持循环检测**	支持业务逻辑中的复杂树型查询能力
	ROWNUM	兼容 Oracle ROWNUM，且可与其他条件组合使用	**减少超过1万处使用 rownum 进行分页代码的修改**
	sysdate	兼容类 Oracle 的系统时间函数，不带时区和毫秒	**减少超过1万处 sysdate 相关 SQL 代码的修改**
函数兼容	to_date	支持 Oracle 标准与非标准输入下的 to_date 函数	**减少超过2千处非标准 to_date 用法业务代码修改**
	ora_hash	支持指定 Hash 桶数量的 Hash 函数	支持业务逻辑中的均衡数据分批操作
特性兼容	大小写	兼容输出字段全大写，避免通过列名无法获取数据	通过兼容性改造，使得 AntDB 的输出字段名称与 Oracle 输出字段名称一致，避免部分潜在的通过名称获取数据的业务逻辑的修改
	函数别名	支持函数计算字段使用全称为别名，保持完整的 Oracle 查询输出格式	
	隐式转换	支持各种类型下的数据隐式转换规则，**与 Oracle 转换规则一致，且转换规则可配置**	**避免大量类型隐式转换相关的报错，此类报错难以事先排查**
	dual 表	支持 Oracle 中的虚拟 dual 表， 且支持在子查询和嵌套查询中使用 dual 表	部分查询使用使用了 Oracle 虚拟表
	系统视图	支持部分 Oracle 特有的系统视图	支持系统视图，且通过视图定义的修改，支持了后台进程以及取数进程的正常运行

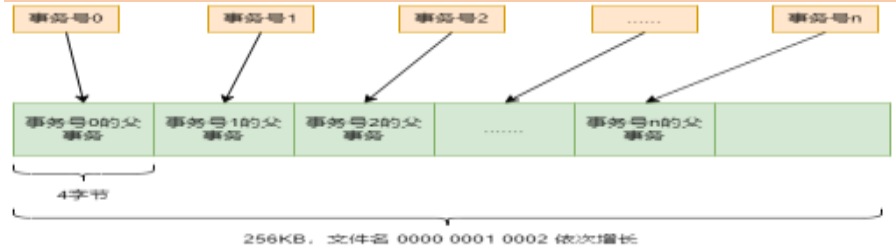
CRM系统全域数据库替换技术创新

AntDB子事务性能优化

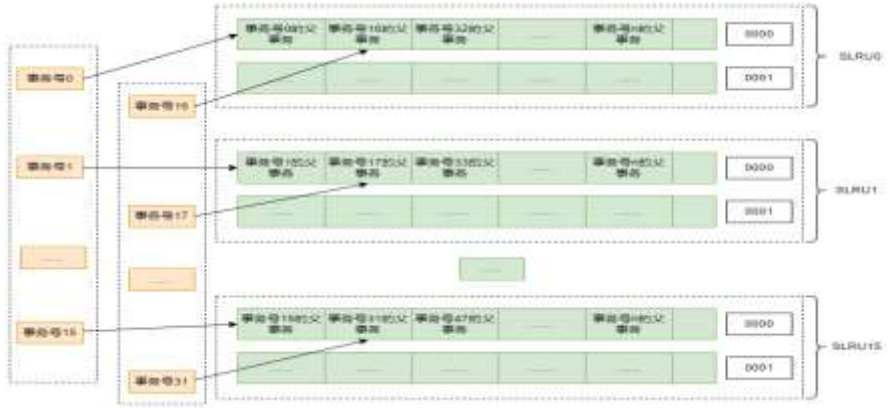
问题现象

启用子事务功能之后，当出现 长事务长时间不提交 或者 并发短事务不及时提交 时，数据库出现 子事务锁资源争用的现象，大量进程在读取子事务状态时，出现锁排队的情况，最终导致数据库性能下降，业务响应时间超时的现象。

改造前



改造后



解决方案

将单个子事务锁拆分成多个，根据现场子事务数量以及测试结果，**子事务锁由原来的1把拆分为256把**，结合hash算法利用其离散性，将子事务信息打散保存到不同的子事务缓存中，每个子事务缓存独立加锁，降低锁的粒度，从而提升子事务性能。

应用效果

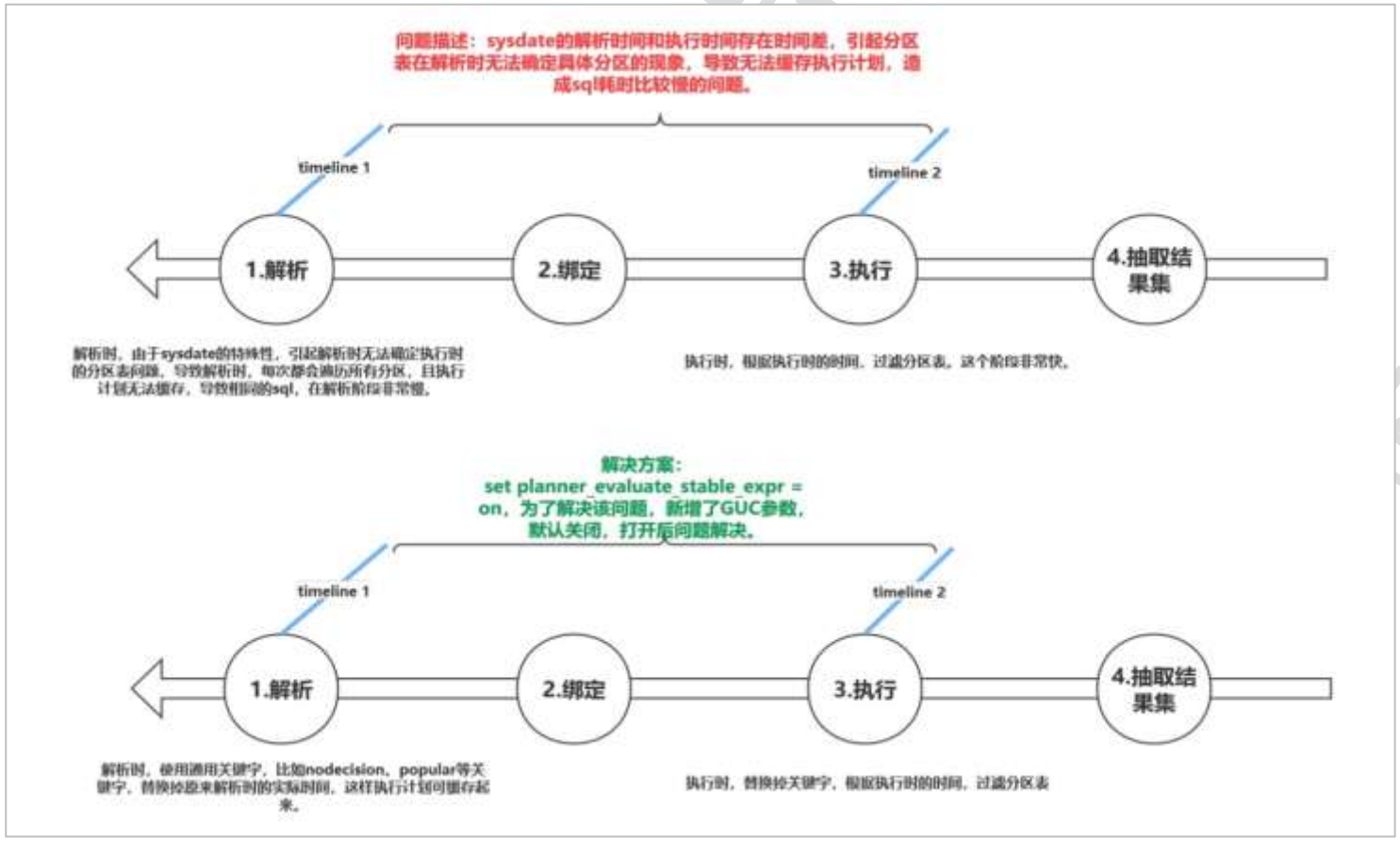
应用了上述优化之后，数据库以前由于子事务性能问题无法跑通的批量并发定时任务，全部作业成功。客户中心数据库的业务量至少上升 **40%以上**，SQL的平均执行效率上升**10%以上。**

CRM系统全域数据库替换技术创新

AntDB分片裁剪性能优化

sysdate作为数据库获取系统时间的函数，被业务系统大量使用，且常常作为分区表的分区字段使用。

AntDB添加一种新的函数属性，在解析阶段使用常量替换sysdate，提前将多余的分区过滤，大幅降低解析阶段的耗时。



sysdate性能优化后的效果

目前分区表毫秒级响应，完全满足业务的日常使用，部分SQL的性能甚至优于oracle。

谢谢观看

THANKS FOR WATCHING

DTC 2024