PostgreSQL中文社区

2021 PostgreSQL China Conference

主办：PostgreSQL 中文社区

第 11 届 PostgreSQL 中国技术大会

开源论道 × 数据驱动 × 共建数字化未来

TDSQL-A技术构架演进及创新实践

伍鑫 腾讯云数据库专家工程师

CONTENT

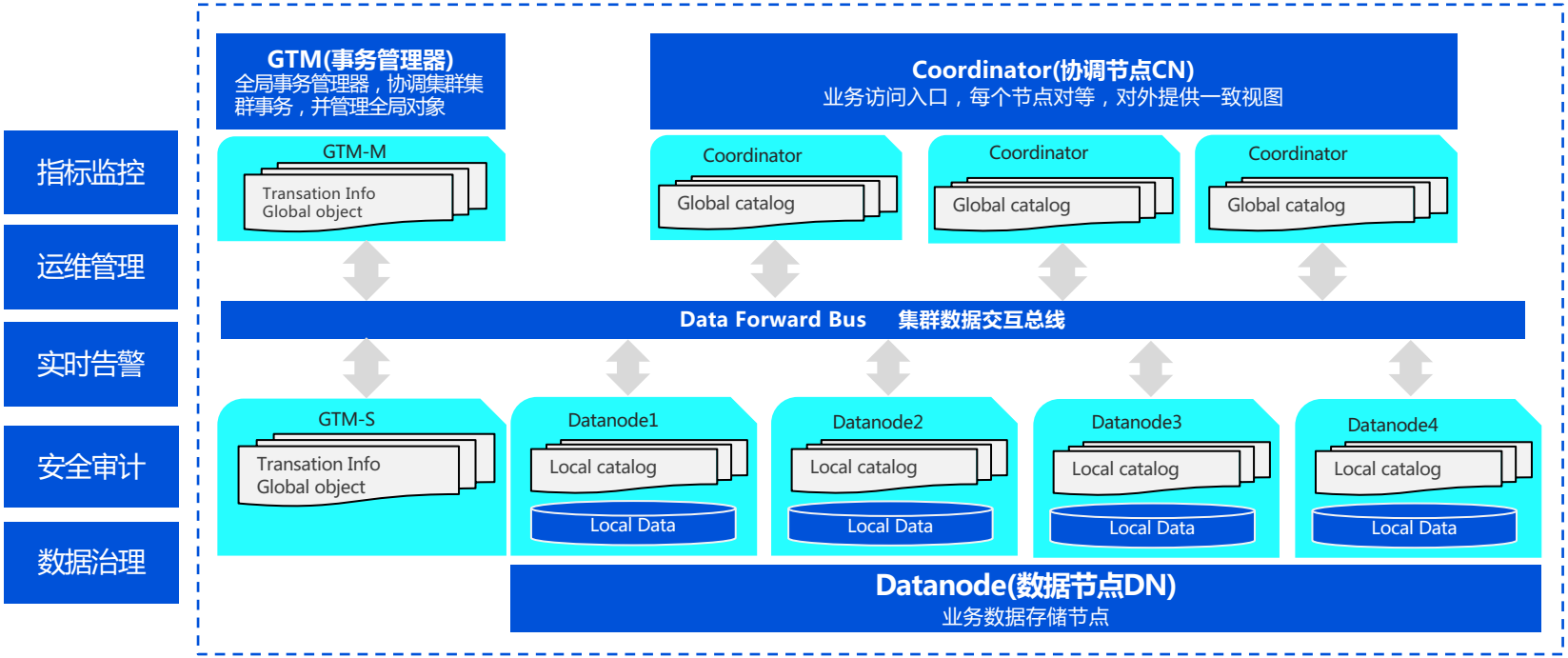
PART 01

TDSQL-A发展历程介绍

TDSQL-A是腾讯基于PostgreSQL自主研发的分布式在线关系型数据仓库

- 无共享 MPP
- 行列混合存储
- 超大规模集群支持
- 超高速计算能力

TDSQL-A整体架构

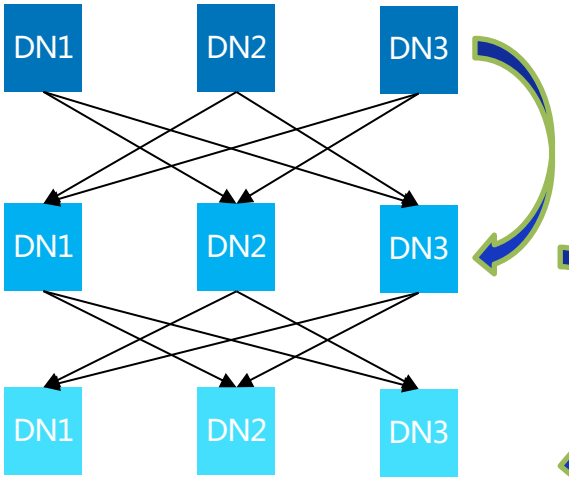

指标监控
运维管理
实时告警
安全审计
数据治理
GTM(事务管理器)
全局事务管理器，协调集群集群事务，并管理全局对象
GTM-M
Transation Info
Global object
Coordinator(协调节点CN)
业务访问入口，每个节点对等，对外提供一致视图
Coordinator
Global catalog
Coordinator
Global catalog
Coordinator
Global catalog
Data Forward Bus 集群数据交互总线
GTM-S
Transation Info
Global object
Datanode1
Local catalog
Local Data
Datanode2
Local catalog
Local Data
Datanode3
Local catalog
Local Data
Datanode4
Local catalog
Local Data
Datanode(数据节点DN)
业务数据存储节点

PART 02

整体构架演进

大规模集群面临的挑战

集群扩展性挑战，分布式JOIN消耗大量网络连接和对应资源



常见做法：

① 第一次重分布： A join B
(N – 1) * P 个连接
(N – 1) * P 个进程

✚

② 第二次重分布：（A Join B）Join C
(N – 1) * P 个连接
(N – 1) * P 个进程

- 问题：单节点连接数太多：

200 个 DN 节点，100 个并发查询，每个查询 5 个重分布

((200 - 1) * 100）*5 = **2万 * 5 = 10万** 连接，每个连接在节点对应一个进程

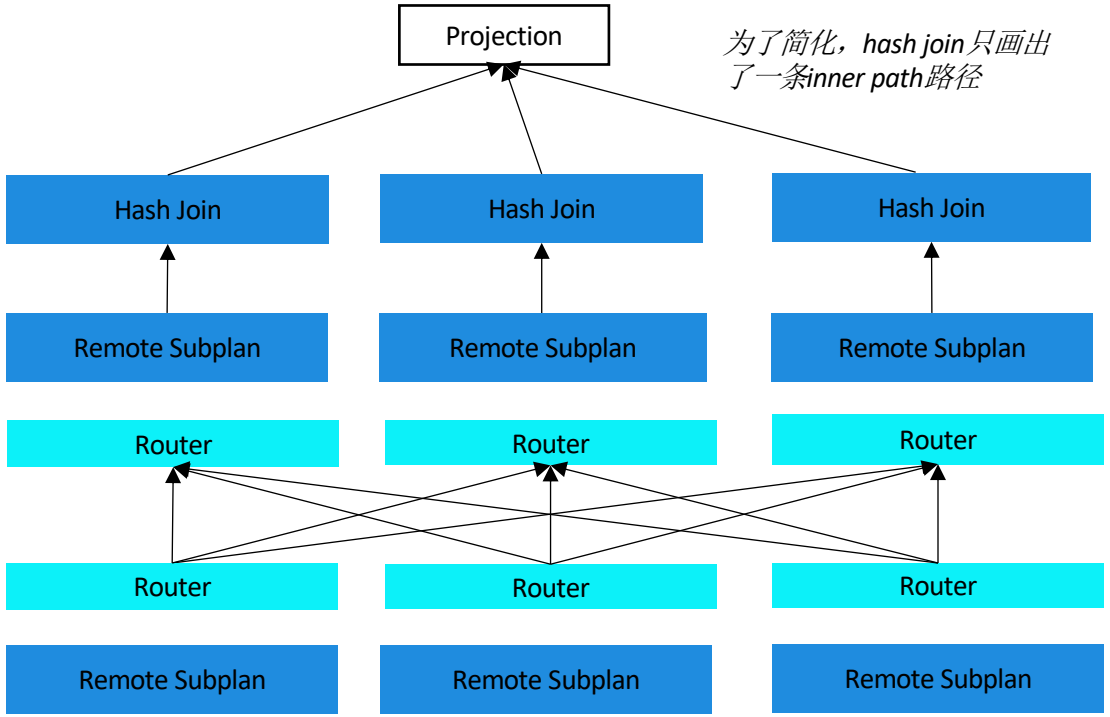
- 限制分布式数据库扩展性的核心问题之一：

服务器连接数过高

异步执行框架

TDSQL-A分布式逻辑框架：

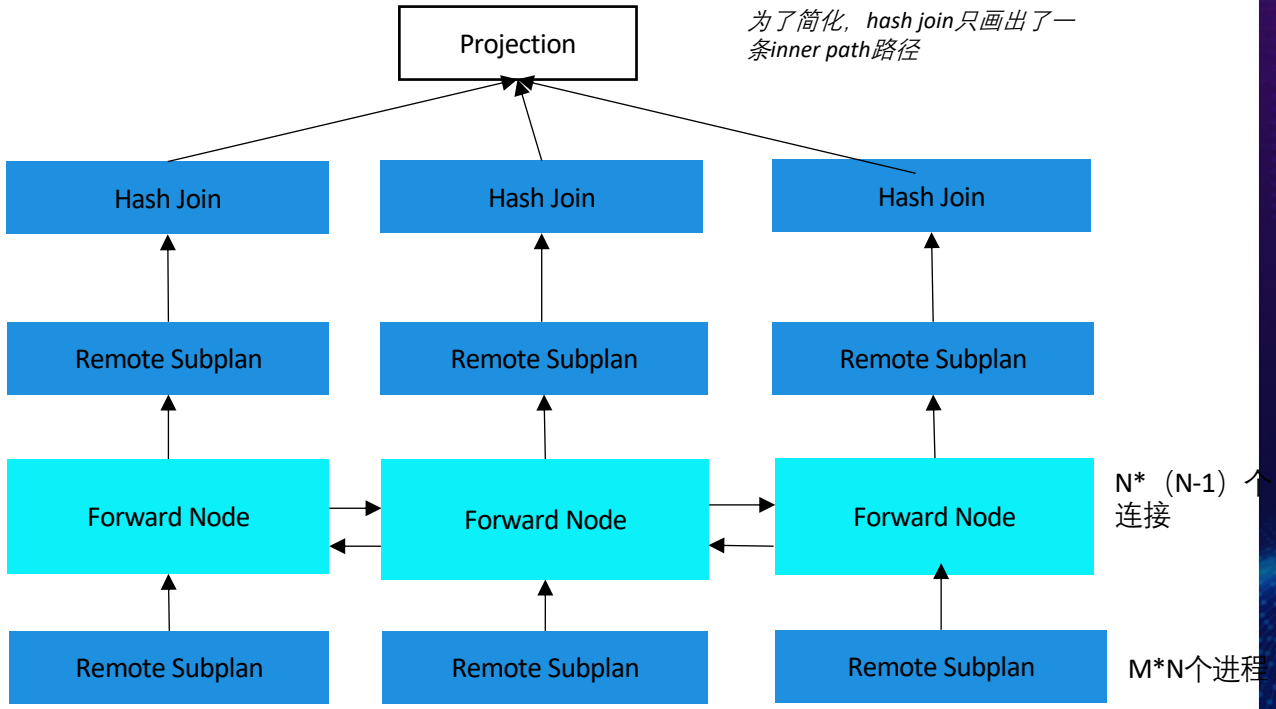
- 在查询优化阶段分析物理查询计划，统一创建DN上的各层执行进程。
- 保证进程间不需要建立冗余进程及连接。
- 不同层级进程间可以异步的启动执行。
- 假设N个节点，M层Join，则会产生M*N个进程数。



数据转发节点支持超大规模集群

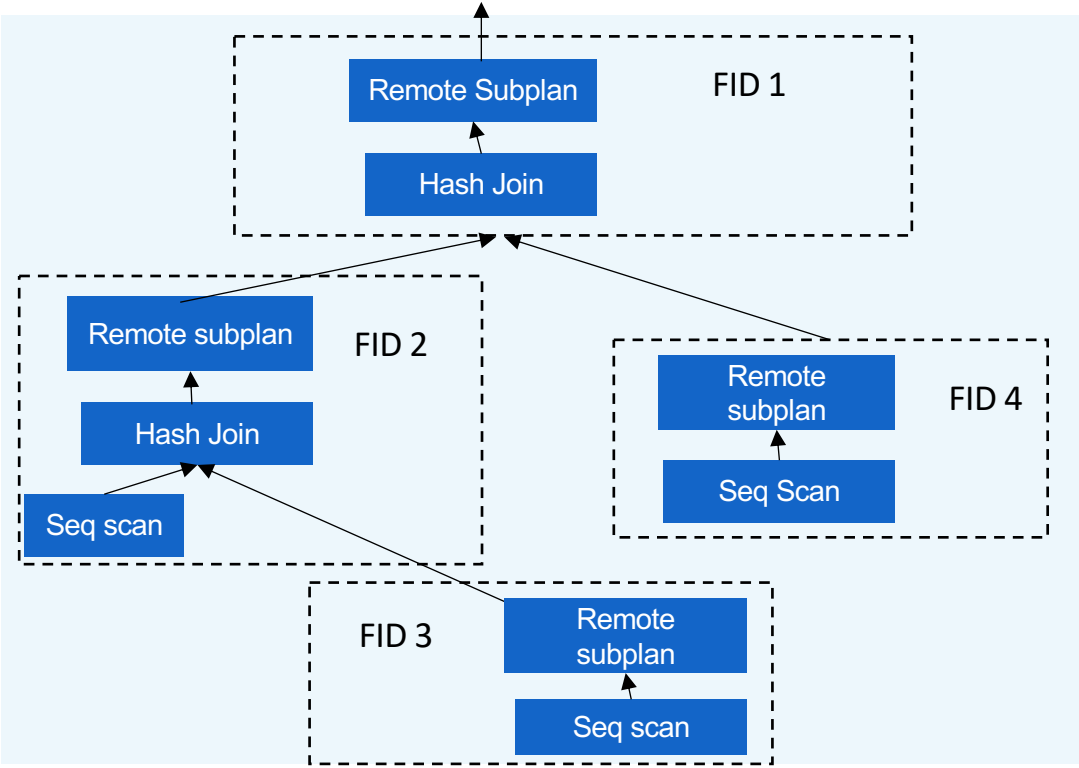
TDSQL-A分布式物理框架：

- 进一步引入Forward Node（FN）来进行节点间数据交互。每台物理机一个FN节点。
- FN与CN/DN通过共享内存进行数据交互，本机数据交互可以不走网络层。
- 假设N个节点，M层Join，且不管查询多复杂，只有N*(N-1)个网络连接数。



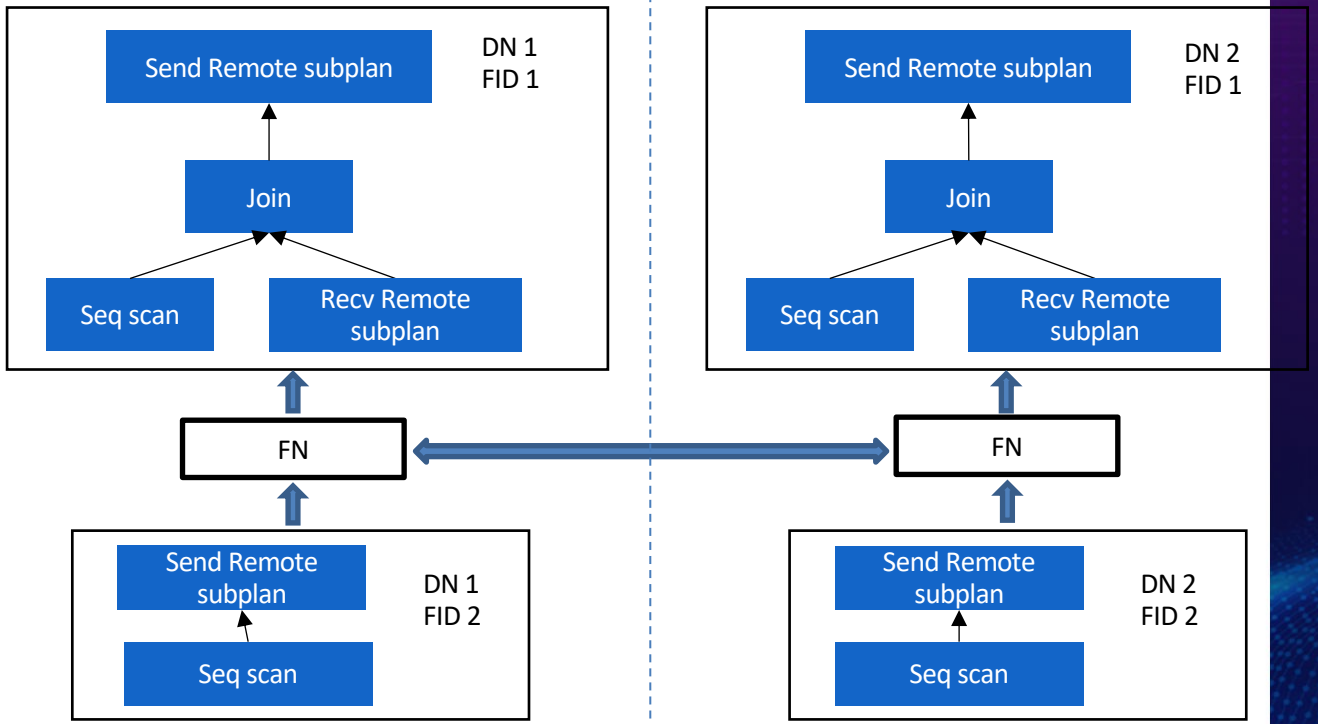
查询计划分片

- 包括数据重分布代价在内，优化器生成代价估算最优的执行计划。
- 递归遍历执行计划，对计划树划分分片（Fragment）。
- 通过FID对计划分片进行管理。



查询分片通过FN节点进行交互

- CN下发每个分片对应的执行计划片段。
- 每个分片在每个执行节点上创建一个进程，执行对应的执行计划。
- 不同层级的进程异步启动执行，通过FN进行数据交互。



PART 03

自研列式存储

支持行列混合存储

- 支持按照行存储和列存储建表
- 列表和行表之间可以进行相互操作
- 行列表之间的混合查询保证事务一致性

姓名	部门	年龄
蜘蛛侠	工程部	18
超人	外联部	100
火箭浣熊	外联部	6
闪电侠	工程部	17

按行存储表：

1、每行数据存储所有列

2、一次磁盘IO可以访问一行中所有列

3、适合OLTP场景

按行存储

蜘蛛侠	部门	年龄
超人	工程部	18
火箭浣熊	外联部	100
闪电侠	外联部	6
蜘蛛侠	工程部	17

按列存储表：

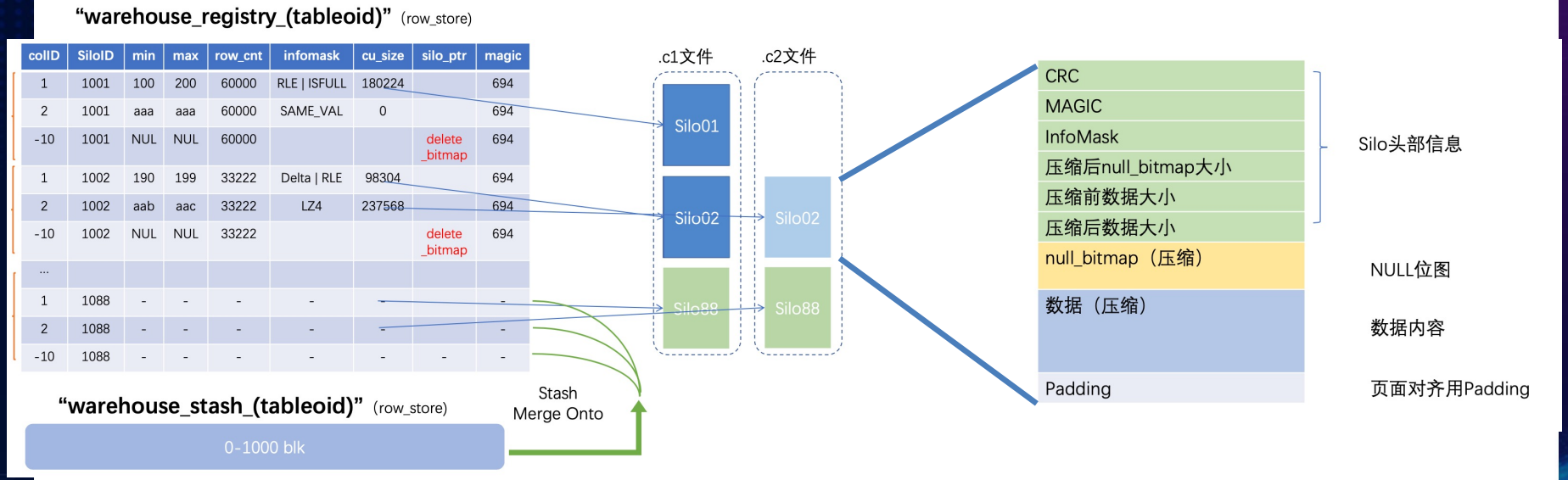
1、每列单独存储，多个列逻辑组成一行

2、一次磁盘IO只包含一列数据

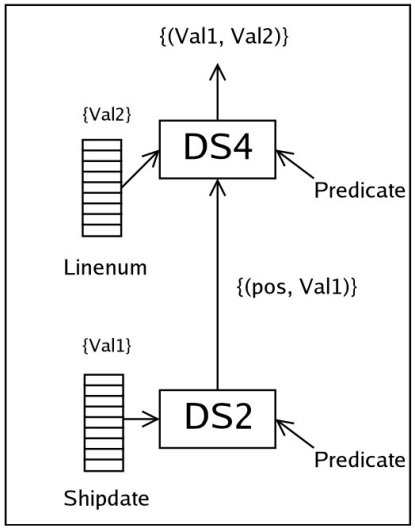
3、方便做数据压缩

按列存储

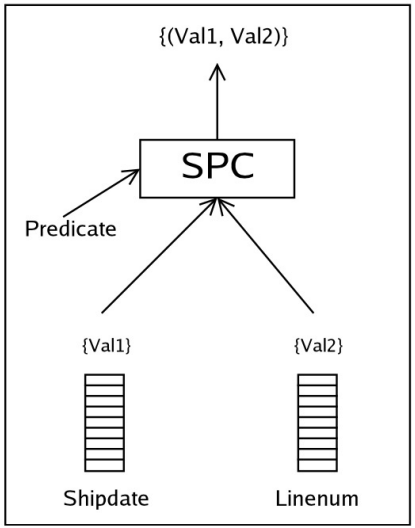
TDSQL-A自研列式存储



列存储延迟扫描优化



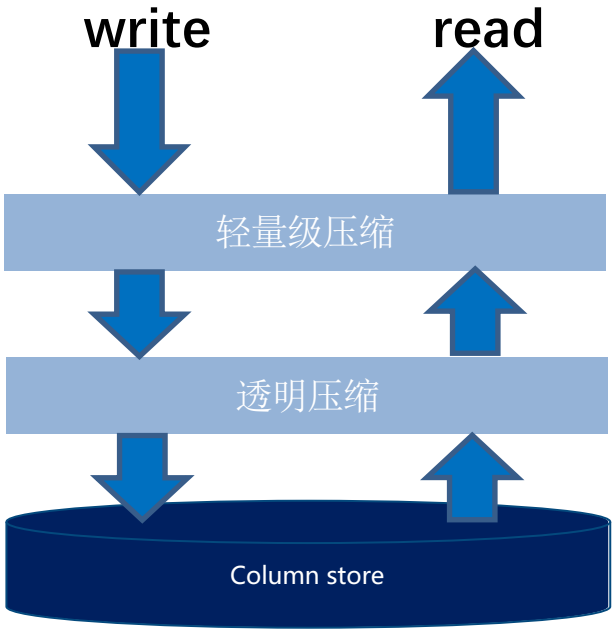
(a)



(b)

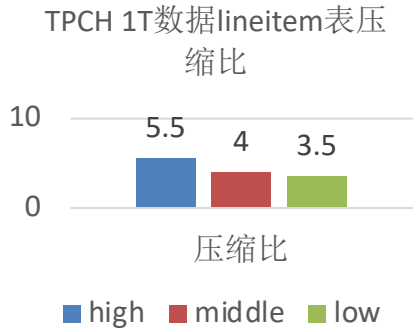
支持Late Read（a）多列扫描时、逐列进行predicate。相比传统方式（b）减少后续列的数据扫描量。

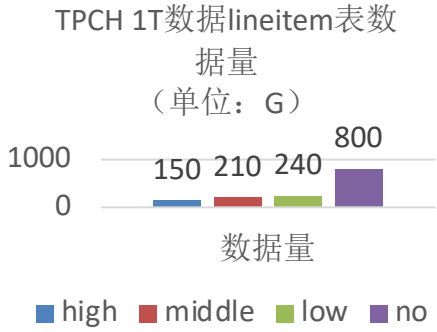
列存数据压缩能力增强



- 透明压缩算法：zstd, Lz4
- 轻量级压缩算法：Delta, RLE, Dictionary

压缩级别	文本类型	整数类型	Numeric类型
low	Lz4	Delta+RLE	1) 能转化为int32/int64：Delta+RLE; 2)不能转化的：Lz4
middle	Dict / Lz4 (Dict优先级高)	Delta+RLE+Lz4	1) 能转化为int32/int64：Delta+RLE+Lz4; 2)不能转化的：Lz4
high	Dict / Zstd (Dict优先级高)	Delta+RLE+Zstd	1) 能转化为int32/int64：Delta+RLE+Zstd; 2)不能转化的：Zstd





PART 04

执行引擎能力提升

多层级并行能力提升

select * from tbl_a, tbl_b where tbl_a.f1 = tbl_b.f2;

TBL_A(f1--分布列, f2)

TBL_B(f1--分布列, f2)

全并行计算可以榨干硬件的潜力

是做复杂查询的必经之路。

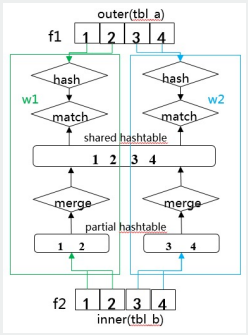
节点级并行

CN

TBL_A.f1 = TBL_B.f2

DN1

TBL_A.f1 = TBL_B.f2



进程级并行

SSE 2/3 OP

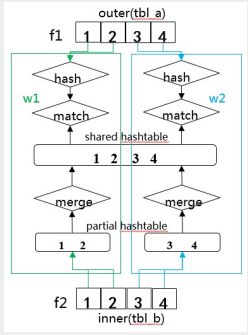
SIMD指令级并行

X 4 X 3 X 2 X 1

Y 4 Y 3 Y 2 Y 1

DN2

TBL_A.f1 = TBL_B.f2



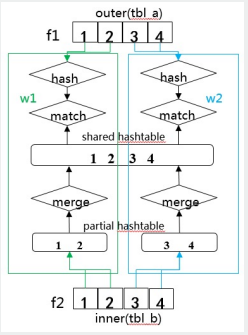
SSE 2/3 OP

X 4 X 3 X 2 X 1

Y 4 Y 3 Y 2 Y 1

DN 3

TBL_A.f1 = TBL_B.f2



SSE 2/3 OP

X 4 X 3 X 2 X 1

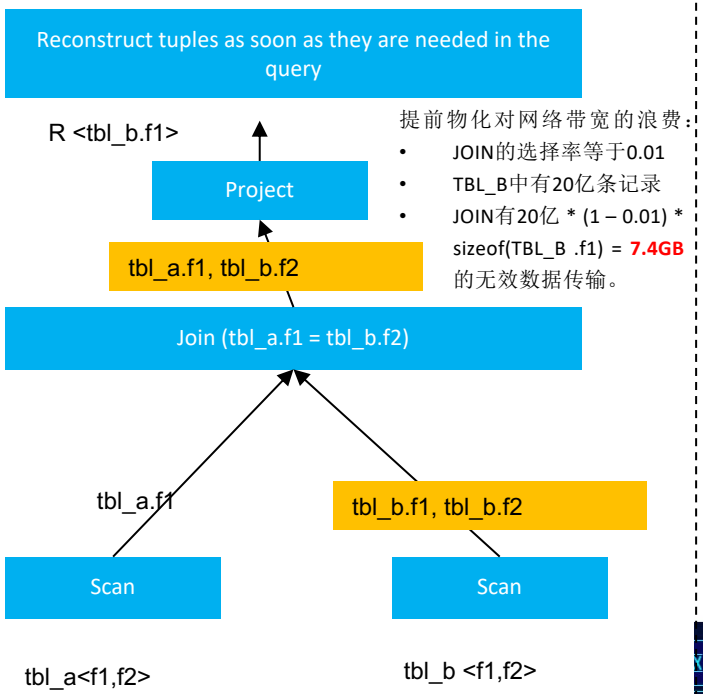
Y 4 Y 3 Y 2 Y 1

未来

基于列存表的分布式延迟物化能力

```
select tbl_b.f1 from tbl_a, tbl_b where tbl_a.f1 = tbl_b.f2;
```

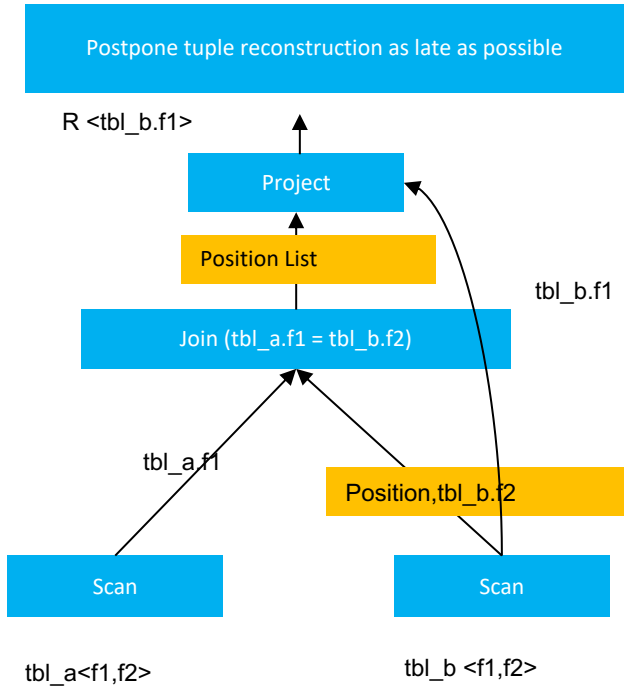
提前物化（常见做法）



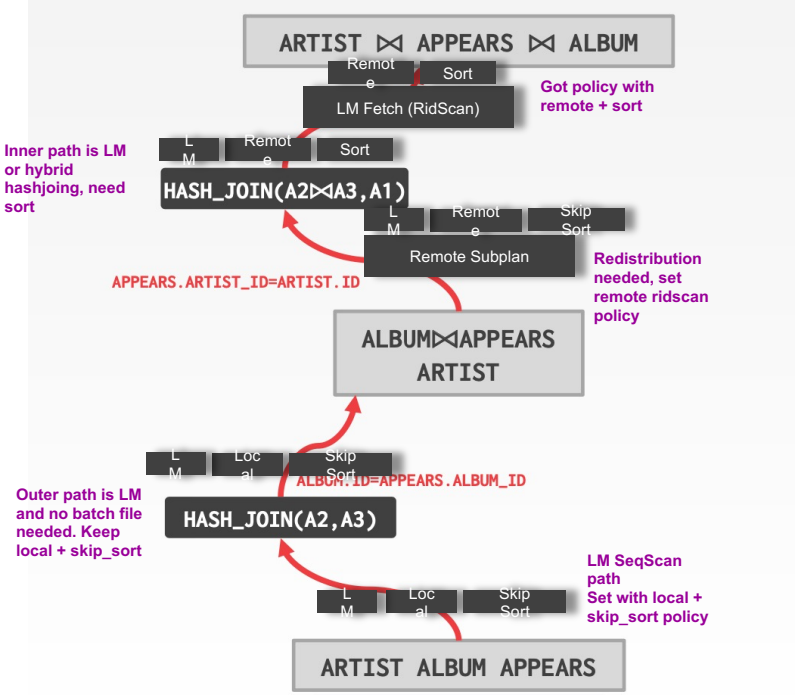
提前物化对网络带宽的浪费：

- JOIN的选择率等于0.01
- TBL_B中有20亿条记录
- JOIN有20亿 * (1 – 0.01) * sizeof(TBL_B .f1) = **7.4GB** 的无效数据传输。

TDSQL-A延迟物化



基于CBO的延迟物化计划生成



- 1. path生成过程中根据当前节点特性调整需要的策略
- 2. 在SeqScan生成LM path时，都是GtidScan local scan以及不需要sort。
- 3. RemoteSubpath的生成会造成策略从local scan改成remote scan
- 4. HashJoin path会根据自身状态改变GtidScan策略。
  - Hybrid HashJoin（batch>1）会导致RidScan需要Sort
  - Inner Path如果是LM的，同样会导致需要Sort

向量化执行引擎

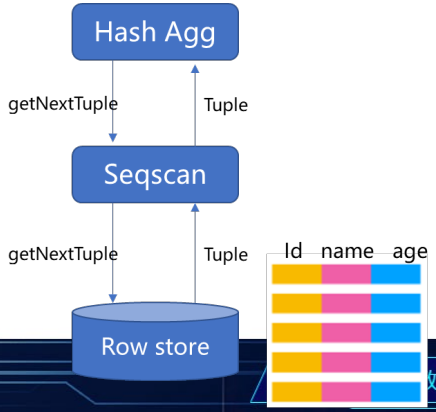
传统的查询执行引擎与向量化查询执行引擎对比

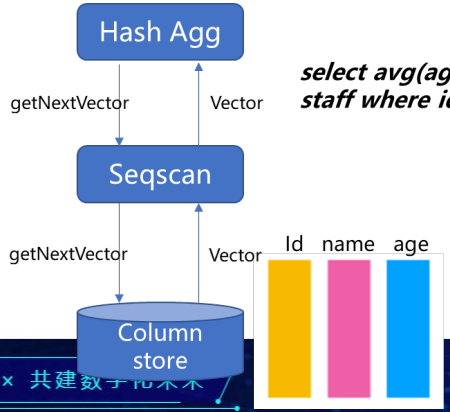
传统查询执行引擎采用火山模型，按照一次处理一个元组的方式，逻辑简单，但效率比较低。

- CPU时间大部分在遍历查询操作树，而不是真正处理数据。
- 数据和指令的缓存命中率低，需要从内存或者磁盘读取。
- 无法利用现有新硬件提供的SIMD能力来加速查询的执行。

向量化查询执行引擎仍然采用火山模型，但是按照一次处理一组元组的方式，需要批量处理，效率高。

- 减少函数调用开销，提高指令、数据的缓存命中率，提升CPU的执行效率。
- 按照列组织形式可以将一组元组表示成一组列向量，每个列向量对应的一整块连续数据可以读入缓存进行处理。

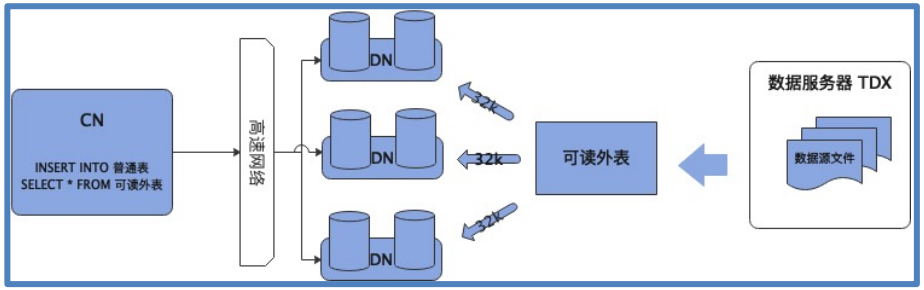




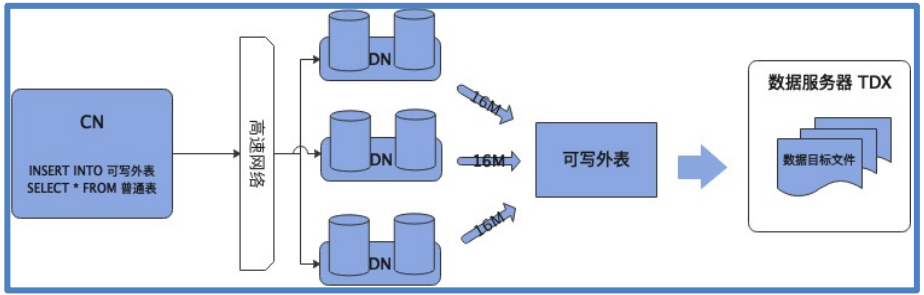
select avg(age) from staff where id > 10;

TDSQL-A高速数据交互工具（TDX）

数据导入



数据导出



- TDX服务器负责外部数据源对接
- TDSQL-A引擎通过外部表定义与TDX服务器资源进行绑定。
- 数据由DN节点并行进行导入与数据重分布，充分利用分布式系统资源。
- 支持并行多任务导入导出、管道、错误表等高级功能，提高用户体验。
- 相比传统Copy入库出库性能有数十倍提升。

腾讯云上线

- 异步执行框架
- FN能力提升
- 自研列存储
- 分布式延迟物化技术

构架优化

- 列存优化升级
- 向量化引擎深度优化
- 算子并行计算优化
- SIMD优化场景覆盖

持续打造生态

- 持续融合PG社区能力
- Oracle兼容能力持续提升
- 支持大数据生态对接
- 机器学习算法支持

THANKS

谢谢观看