DTCC

数，造，未，来

第十二届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2021

2021 年 10 月 18 日 - 20 日 | 北京国际会议中心

Spark
Hadoop
OLAP
HANA
MPP
Oracle

IT168.com ChinaUnix.net ITPUB

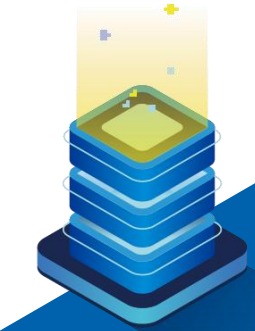
inspur 浪潮 | 云溪数据库

inspur 浪潮 | 云溪数据库

浪潮云溪数据库HTAP解析

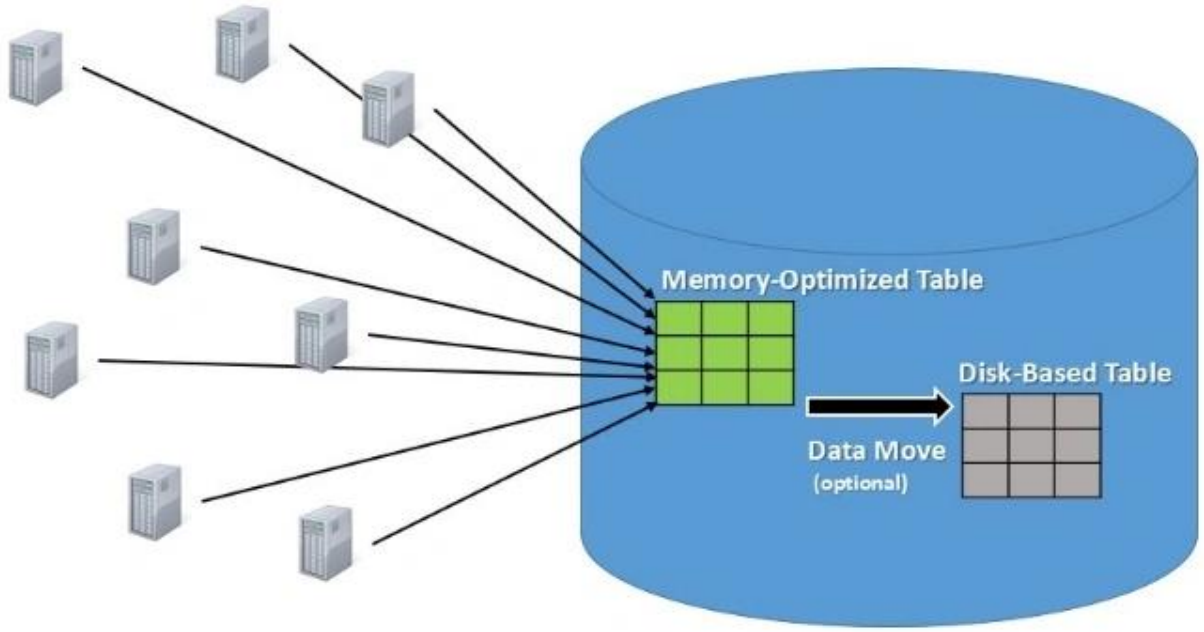
苑晓龙
浪潮云溪数据库资深架构师

目 录



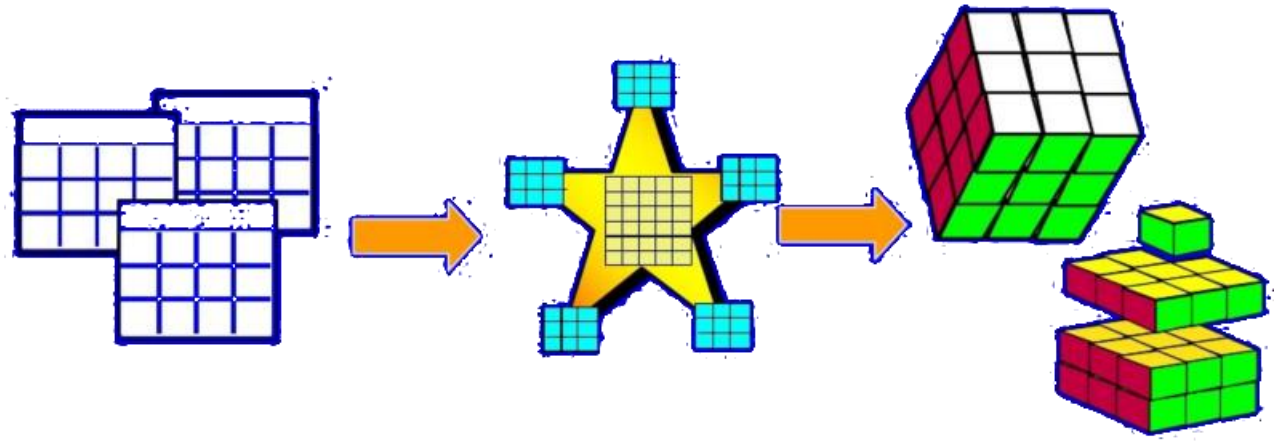
联机事务处理OLTP

- 事件驱动、基于交易的处理过程
- 每次交易访问相对较少的数据
- 大量并发用户添加和修改数据
- 低延时、高容量、高并发
- 并发严格要求交易完整和安全性



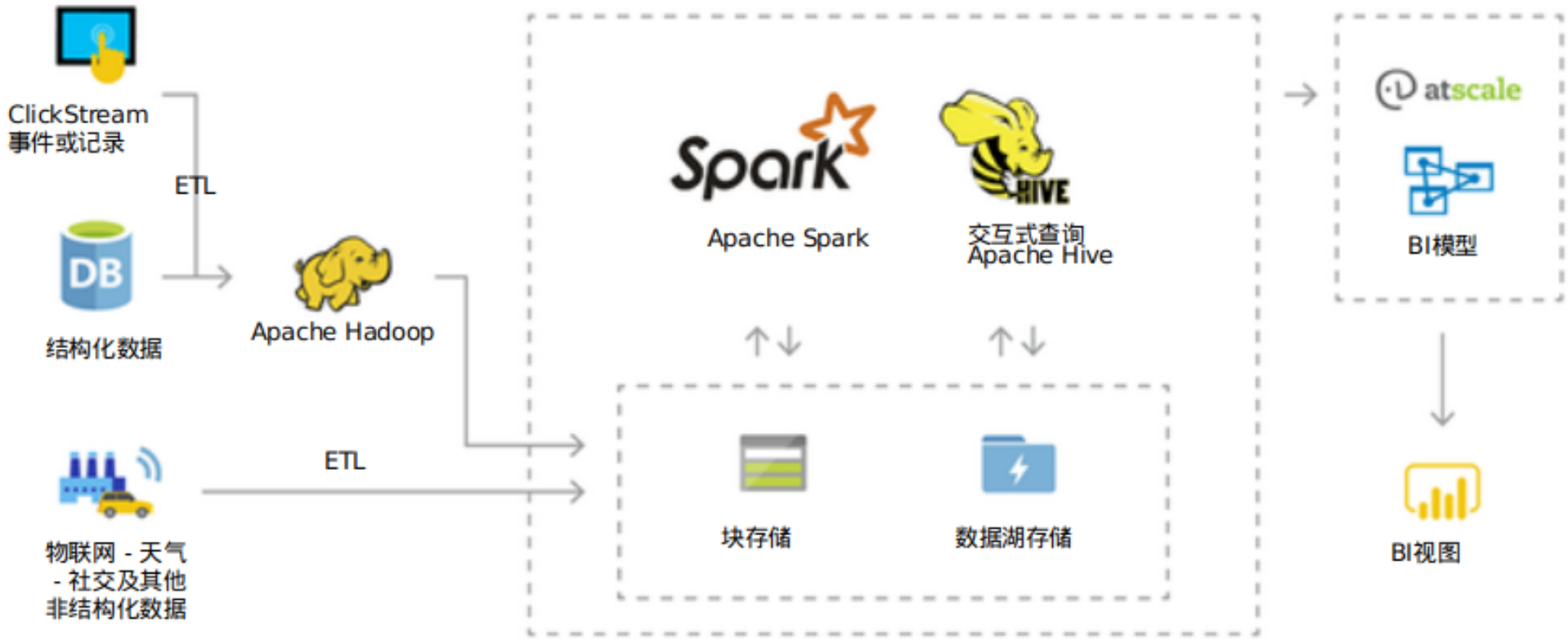
联机分析处理OLAP

- 海量数据，复杂计算分析
- 数据集变更操作较少
- 用户数量相对较小
- 实时性要求相对较低



大数据Hadoop

前提：要分析数据，首先要做ETL



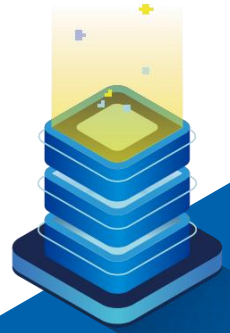
HTAP

混合事务分析处理（Hybrid transaction/analytical processing）

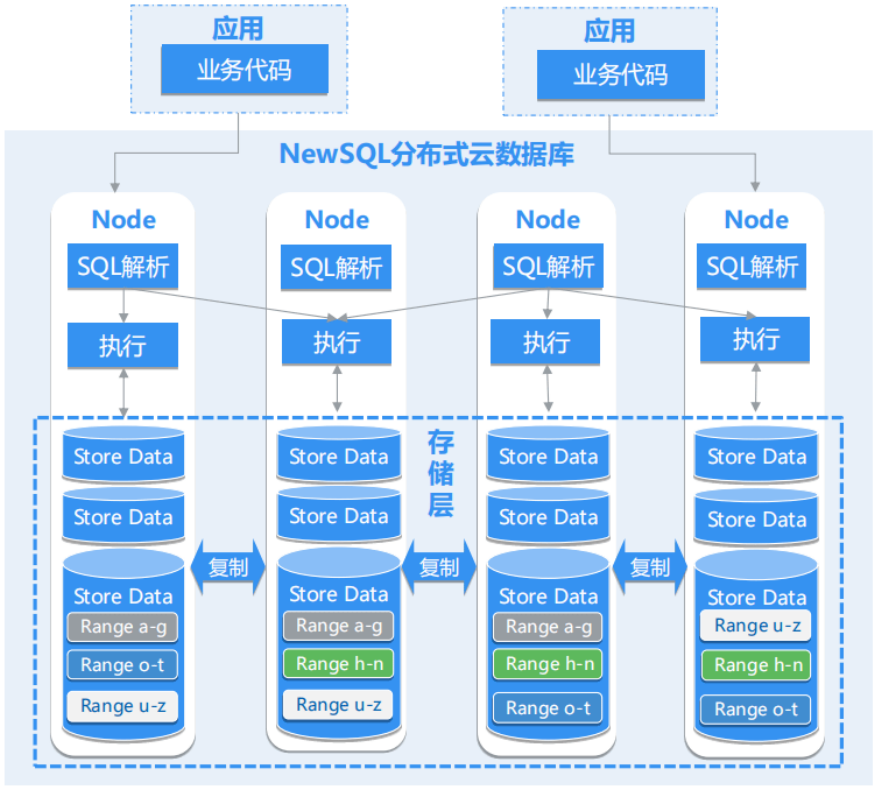
Gartner

- HTAP分析，数据无需从操作数据库(TP)移动到数据仓库
- 事务数据在创建后可随时用于分析
- 分析聚合中向下钻取的原始数据都是最新的数据
- 消除或至少减少了相同数据的多个副本拷贝

目 录



整体架构



- 节点全对等
  - 应用可与任一节点连接并获得响应
- Range范围
  - [StartKey, EndKey)区间，根据Key的范围拆分Range
  - 默认Range大小64MB
- Replica副本
  - 每个Range默认3个副本
  - 采用Multi-Raft协议实现副本的一致性

行/列存数据

ID	NAME	AGE
101	Alice	22
102	Ivan	37
104	Peggy	45
105	Victor	25

- 数据按行存储，一行数据在存储介质中连续存储
- 没有索引的查询会使用大量的IO
- 建立索引和物化视图需要花费大量的时间和资源
- 面对查询的需求，数据库难以满足性能需求

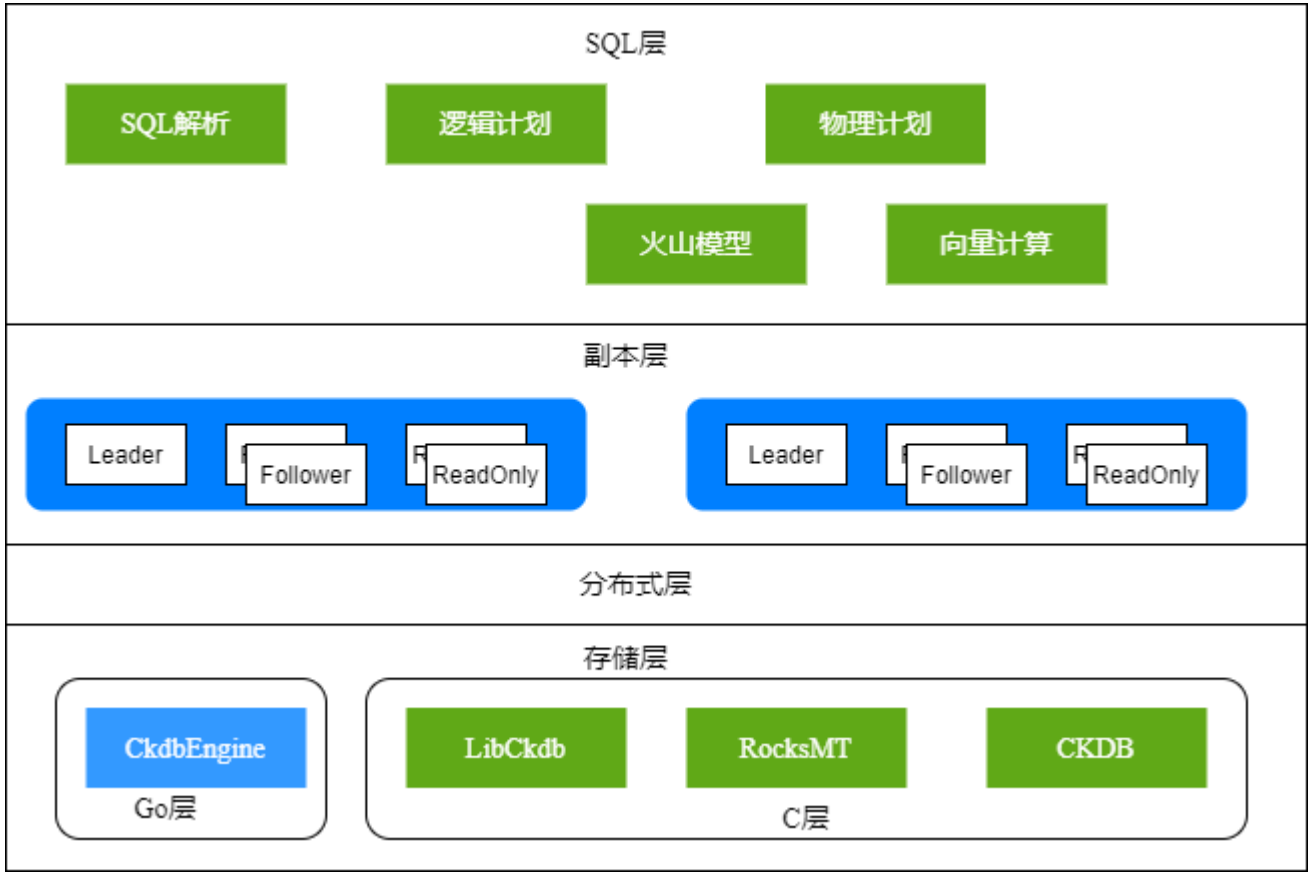
ID
101
102
104
105
108

NAME
Alice
Ivan
Peggy
Victor
Eve

Age
22
37
45
25
19

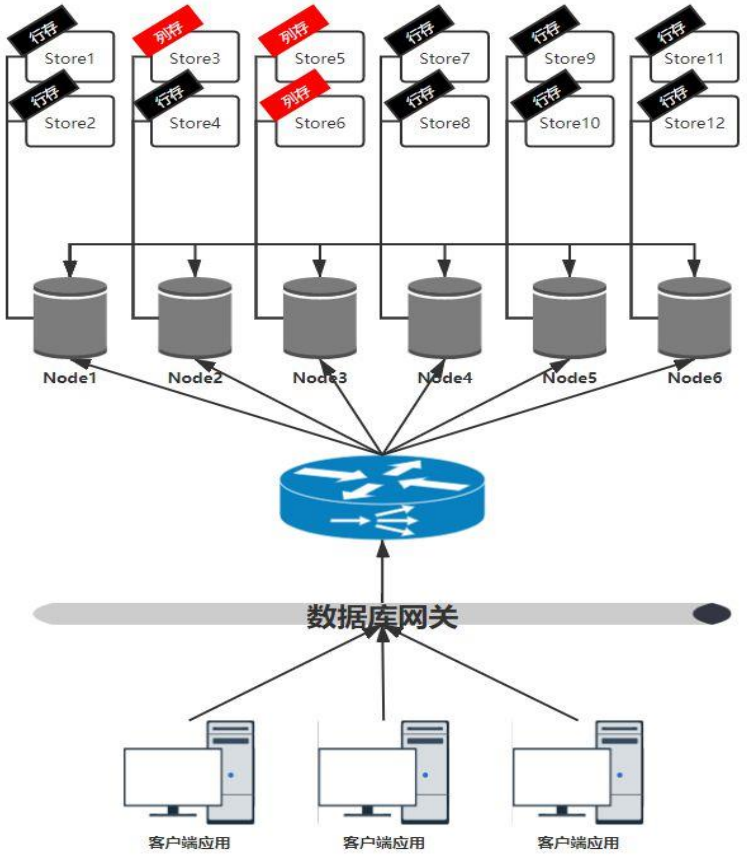
- 数据按列存储，每一列单独存放
- 数据即是索引
- 只访问查询涉及的列，避免大量无效IO
- 每一列可以由一个线程或协程处理，查询并发计算
- 数据类型一致，数据特征相似，可以更高效地压缩

HTAP层次架构



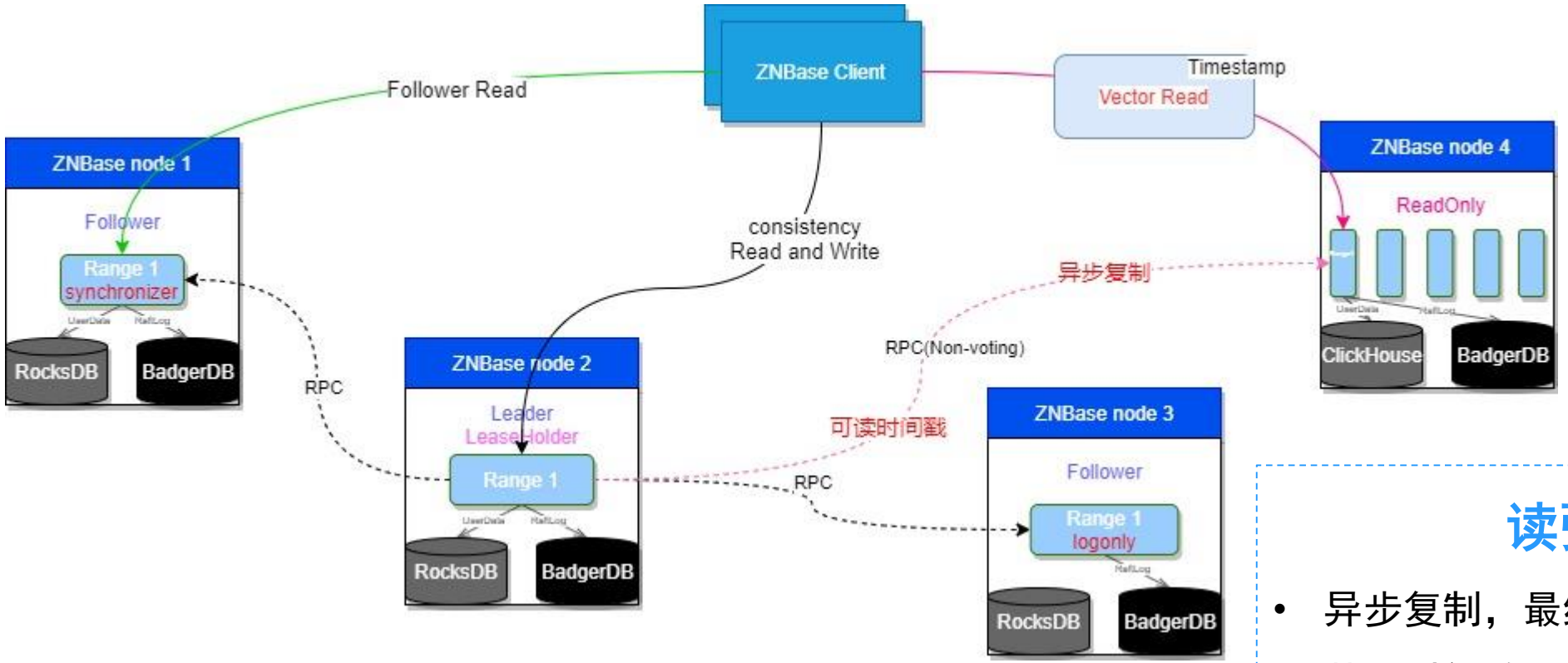
- 向量计算
- 列存副本一致性
- 分布式层计算下推
- 列存引擎实现
- Go层->C层算子下沉
- 计算引擎ZBSpark

HTAP部署架构



- 行存引擎、列存引擎可以混合部署（同一节点）
- 生产环境
  - 每个引擎单独使用一块物理磁盘
  - 将列存引擎与行存引擎部署在不同的节点上(建议)

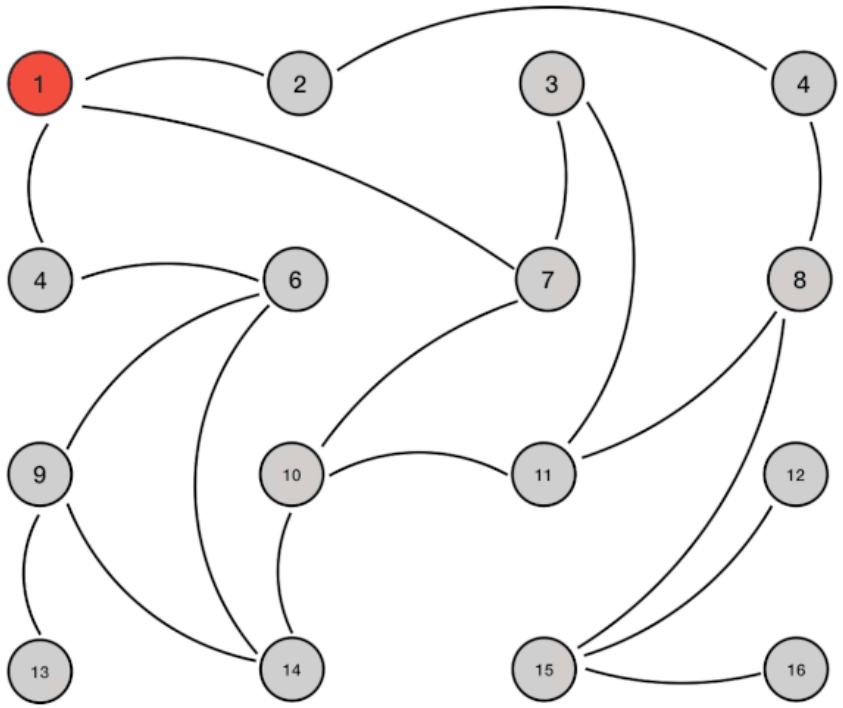
列存副本一致性



读强一致

- 异步复制，最终一致性
- 基于时间戳的读取机制保证事务

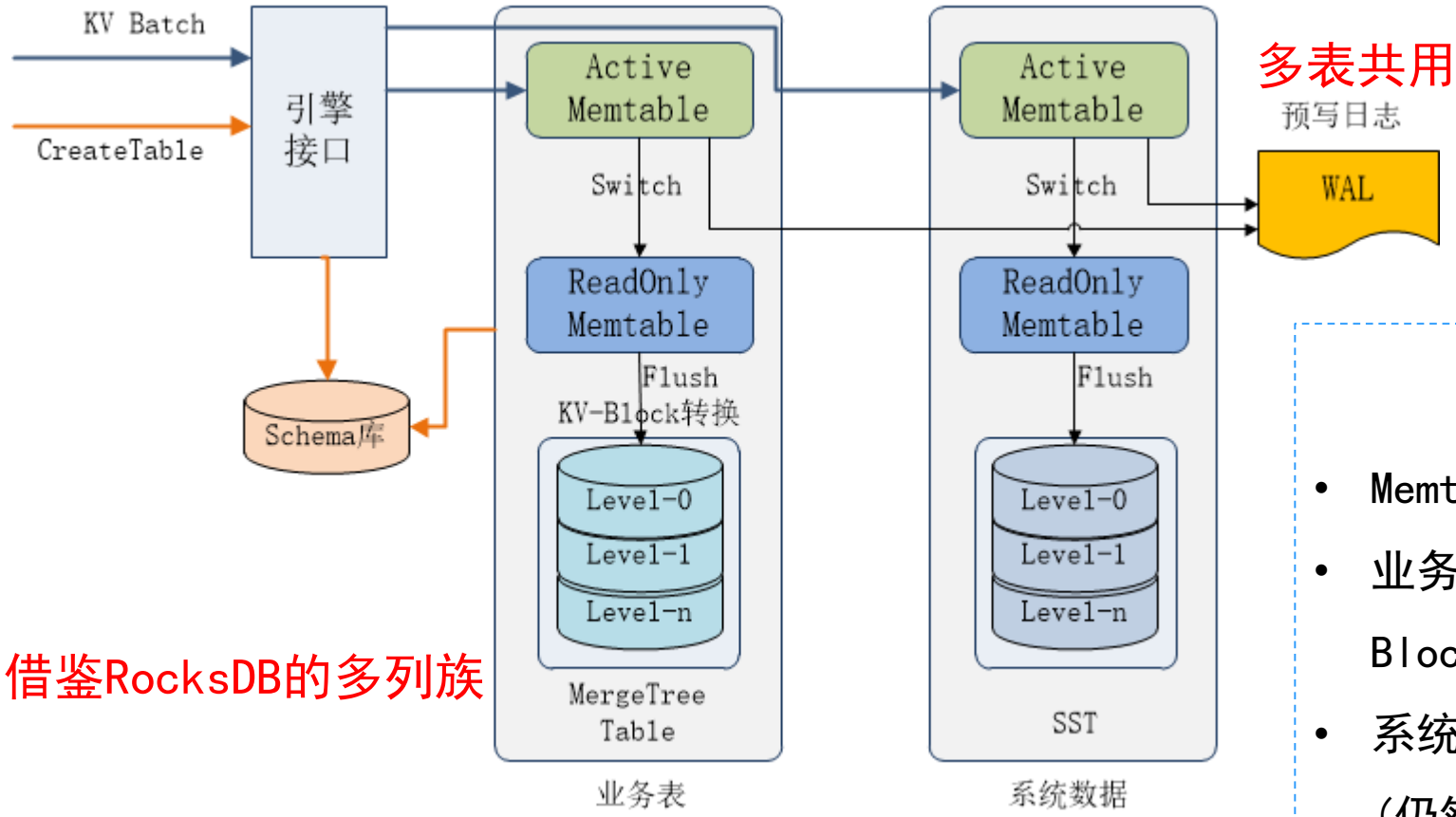
元数据同步



Gossip广播

- 拥有列存副本的数据表schema信息通过Gossip广播
- 列存引擎ckdb接收到该schema信息后将其持久化

列存引擎数据写入

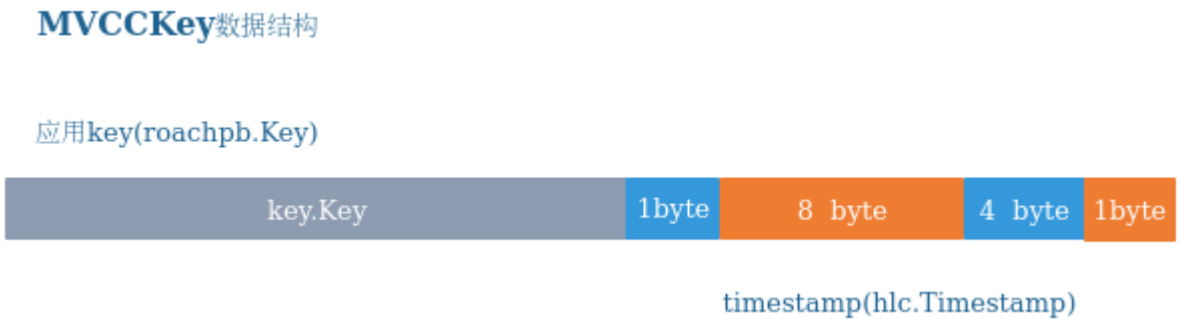


借鉴RocksDB的多列族

行数据-->列存

- Memtable中数据仍然是KV数据
- 业务表数据根据schema持久化到列存Block的MergeTree中
- 系统数据持久化到列存引擎地SST文件中（仍然是KV格式）

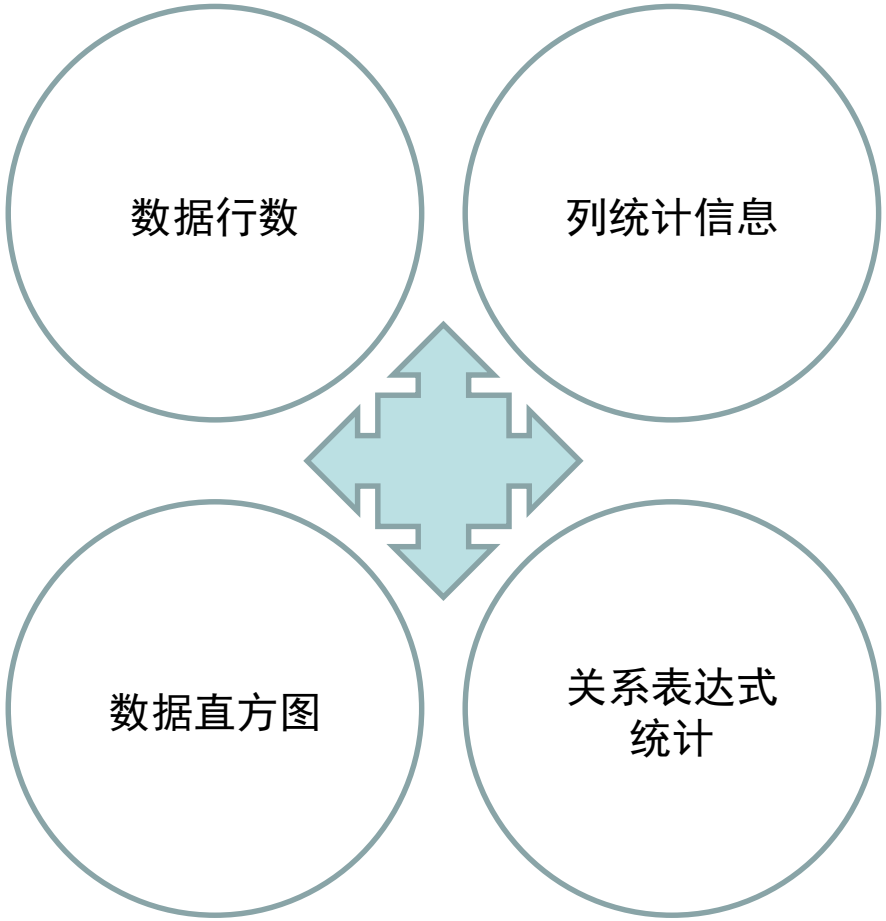
MVCC实现



列存引擎MVCC

- 自定义逻辑时间戳HLC类型
- 自定义MVCC算子，自动过滤出符合要求的数据记录
- 自定义扩展MVCCMergeTree，Merge操作时清理多版本数据、写意图数据

数据统计信息



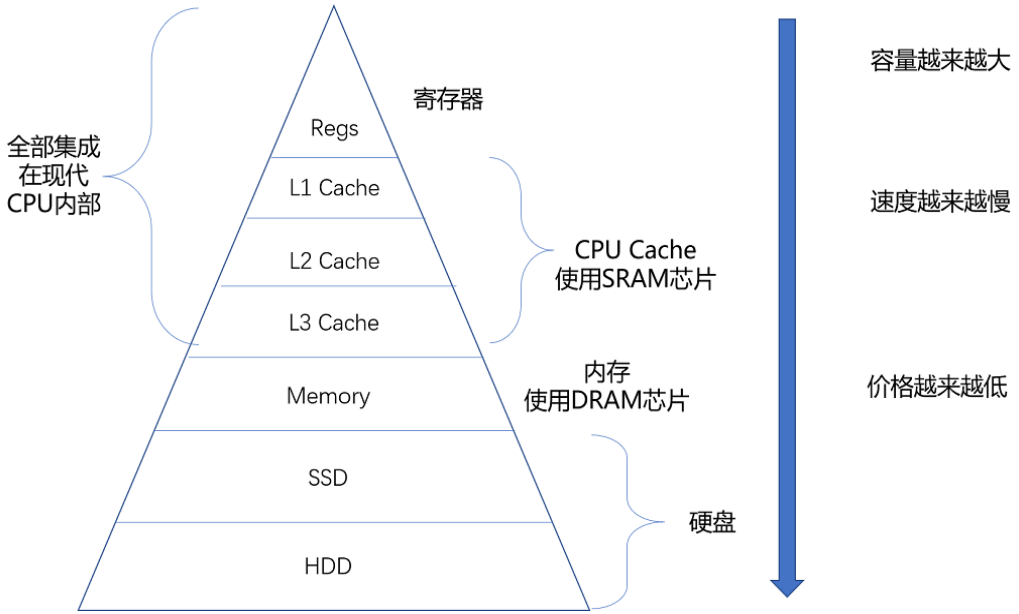
列存引擎

- 列式数据副本的大小
- 每列数据压缩后的字节数

逻辑计划生成

- 基于代价的优化器根据数据统计信息生成逻辑计划
- 逻辑计划智能选择向量计算 or 火山模型计算

存储设备金字塔

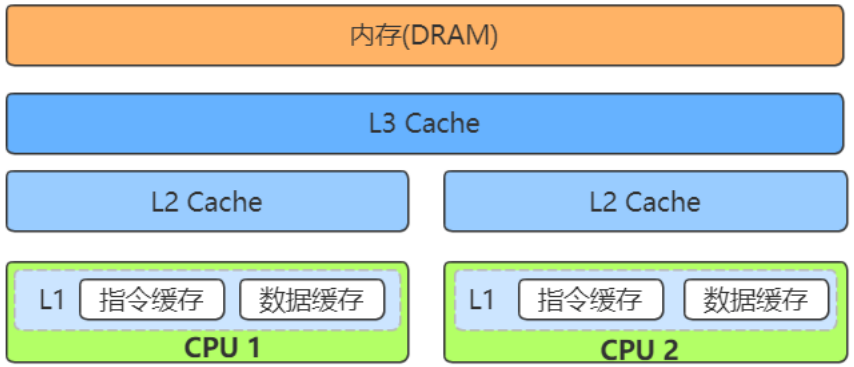


特点

- 越靠近 CPU 速度越快，容量越小，价格越贵。
- 每一种存储器设备只和它相邻的存储设备打交道

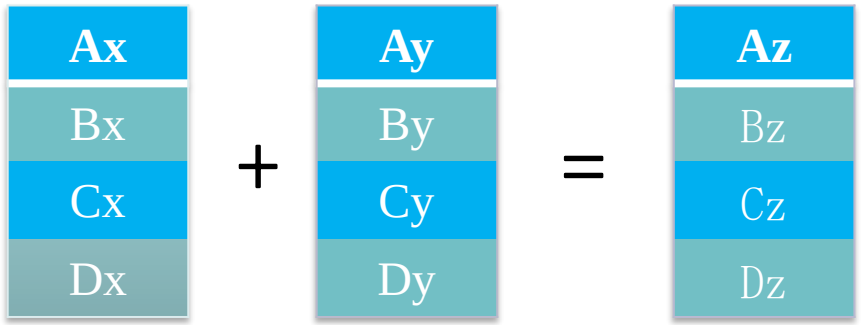
高速缓存(SRAM)

- 每个 CPU 核心都有一块属于自己的 L1 高速缓存，通常分成**指令缓存**和**数据缓存**，分开存放 CPU 使用的指令和数据。
- L1 Cache：往往就嵌在 CPU 核心的内部。
- L2 Cache：同样是每个 CPU 核心都有的，不过它往往不在 CPU 核心的内部。L2 Cache 的访问速度会比 L1 稍微慢一些。
- L3 Cache，则通常是多个 CPU 核心共用的，尺寸会更大一些，访问速度自然也就更慢一些

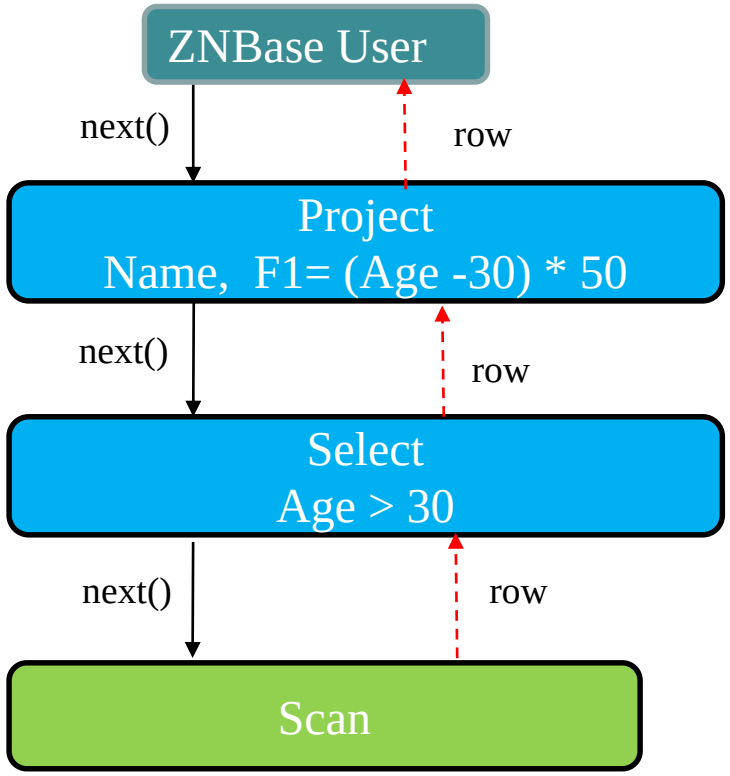


SIMD单指令多数据流

- 全称Single Instruction Multiple Data，能够复制多个操作数，并把它们打包在大型寄存器的一组指令集。
- 以同步方式，在同一时间内执行同一条指令。



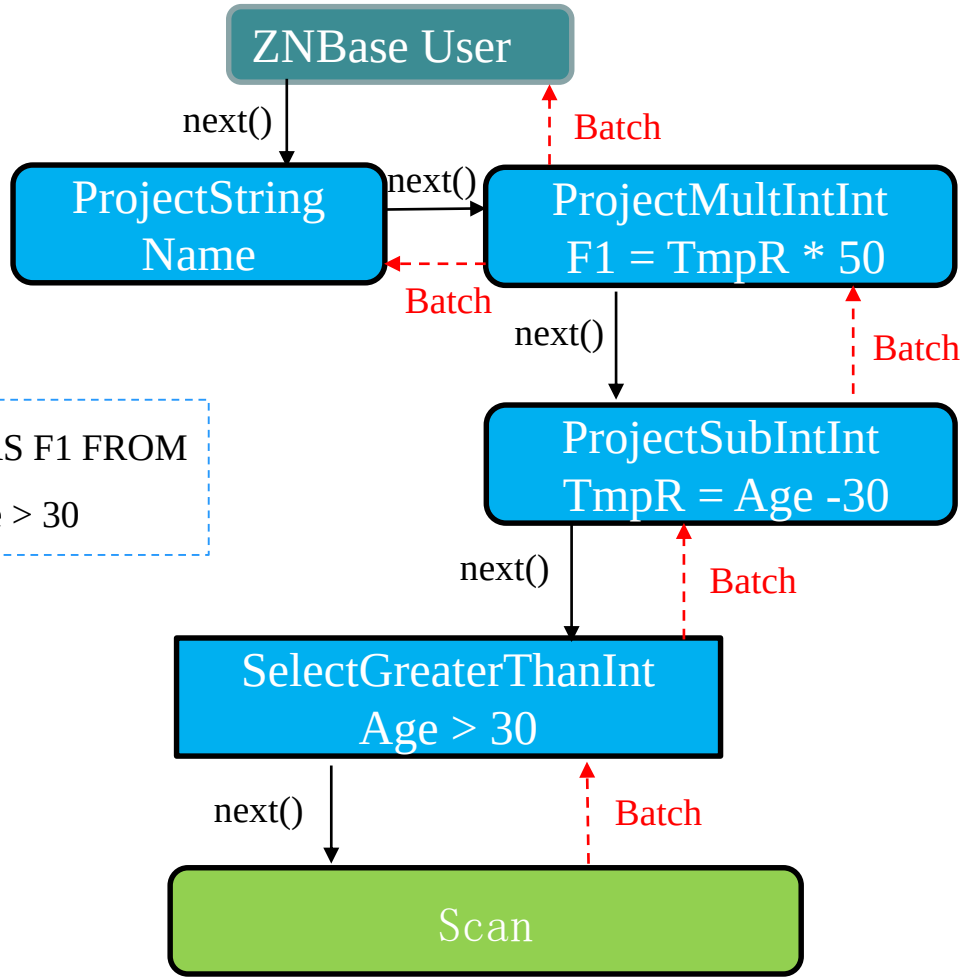
向量计算 vs 火山模型



火山模型

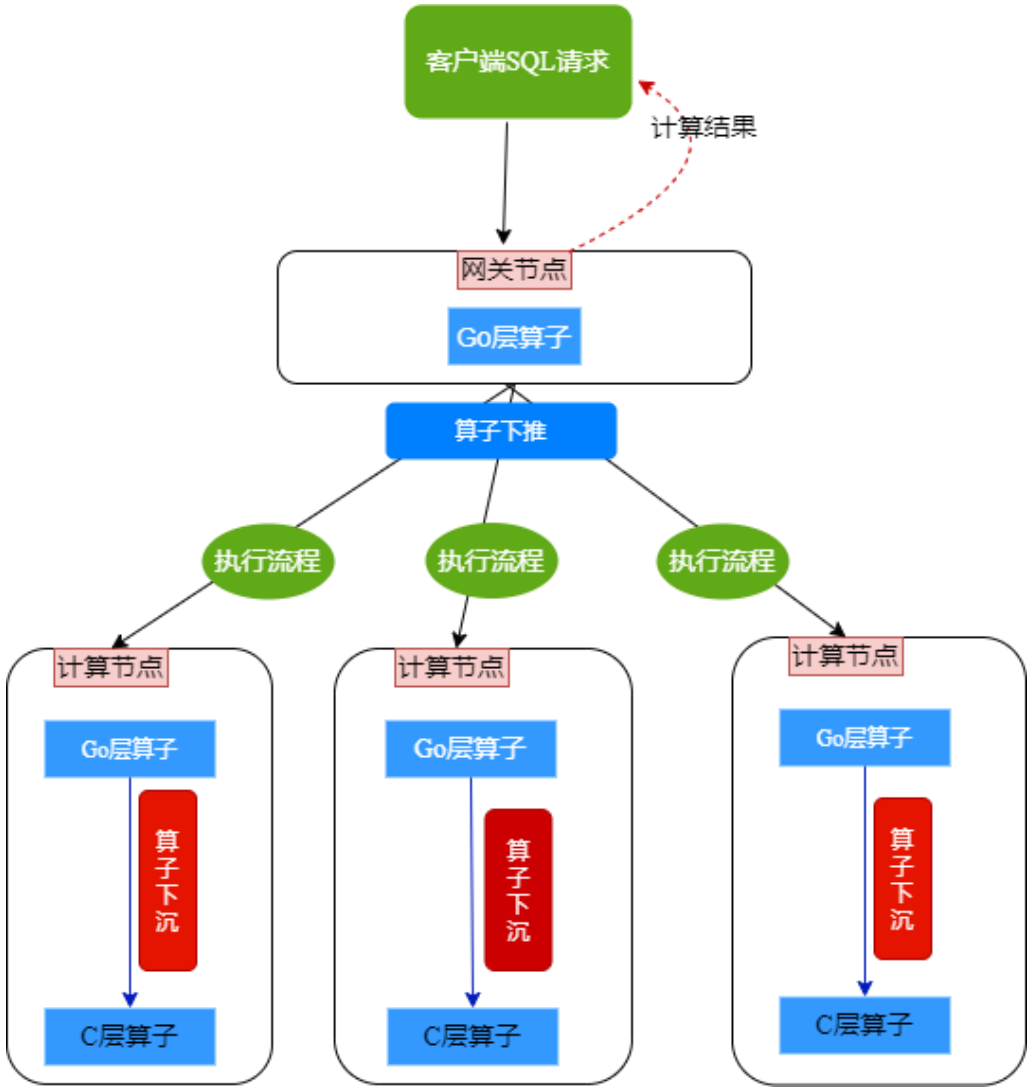
Select Name, (Age-30)*50 AS F1 FROM

TBL_U WHERE Age > 30



向量计算

算子下推&下沉

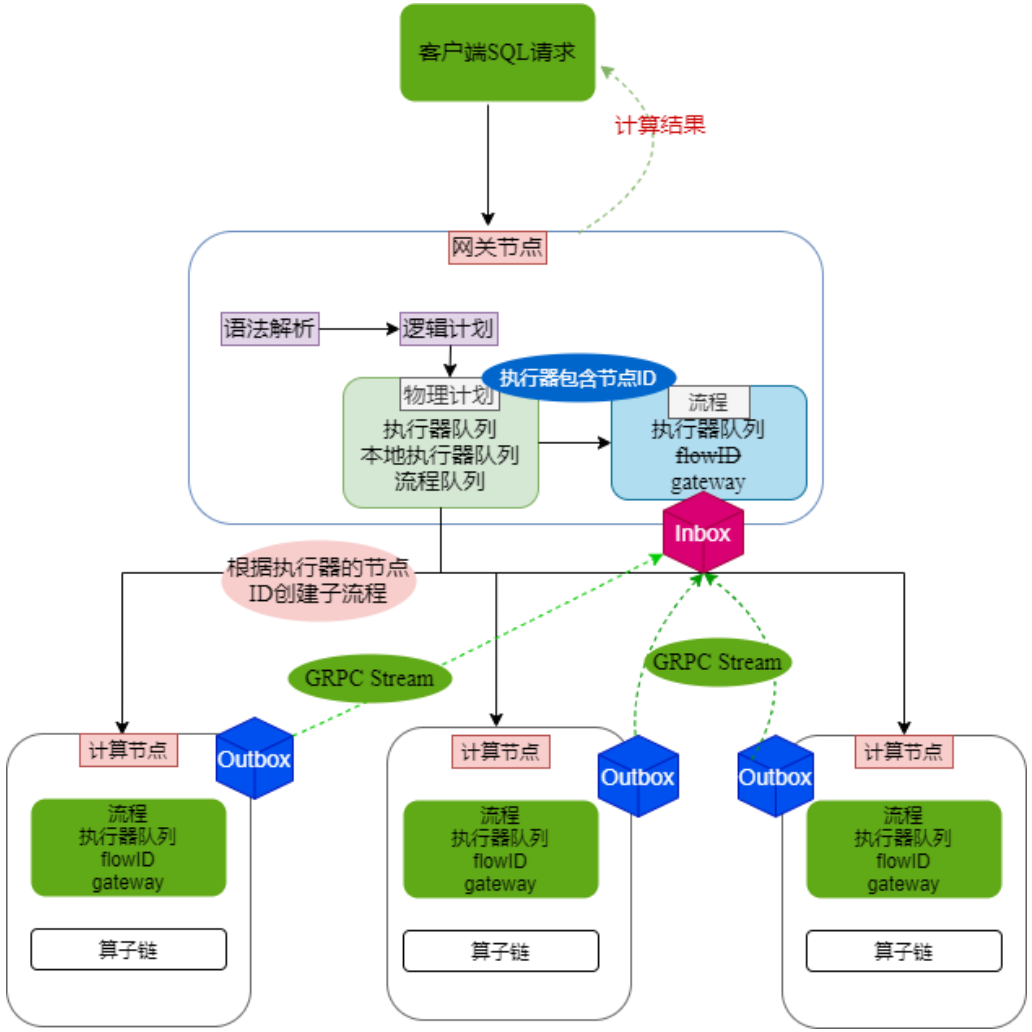


原则：
移动计算的成本远远低于移动数据的成本

算子下推： 降低网络IO
将算子移动到数据所在的节点执行

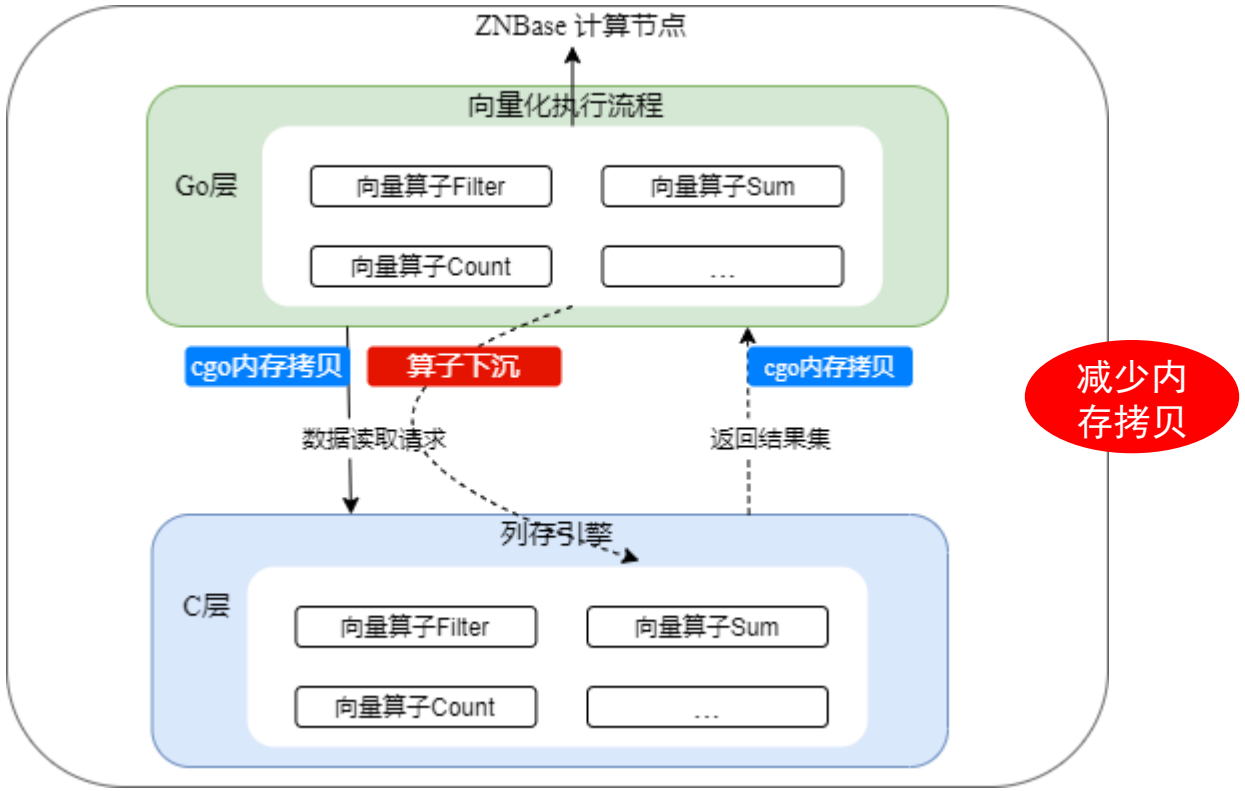
算子下沉： 减少内存拷贝
将计算在节点内部的Go层移动到C层执行

分布式执行流



- 生成分布式执行流程的依据是执行器中的节点ID
- 网关节点的执行流程中仅包含gateway节点ID
- 计算节点的执行流程中包含gateway节点ID及它的流程flowID
- 执行流程涉及远端输入或输出时通过GRPC Stream实现

算子下沉

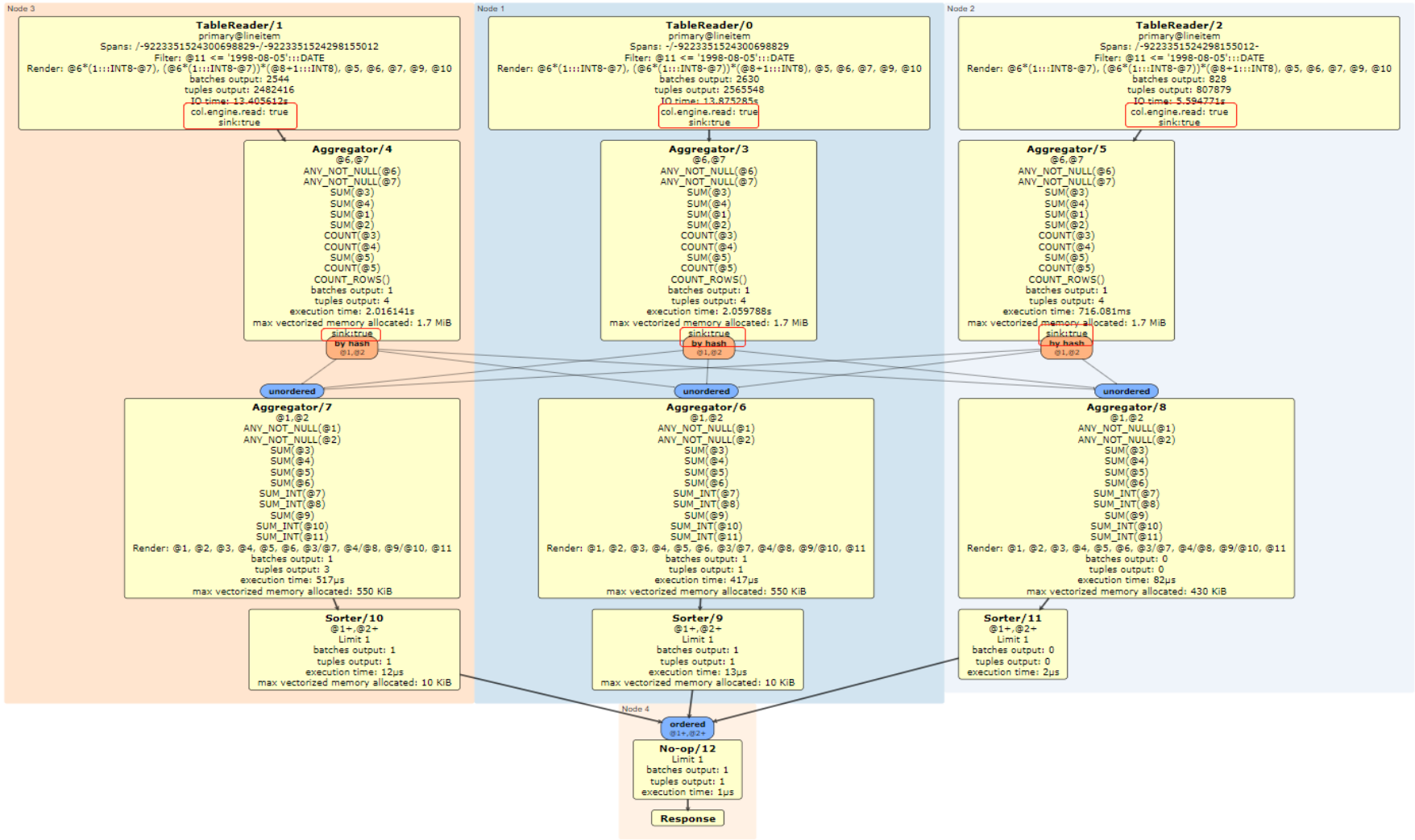


减少内存拷贝

优势

- 利用CKDB的向量算子和动态代码生成技术
- 使用Arrow格式分批返回列式数据
- 降低内存拷贝数据量、减少GRPC中PB的序列化数据量

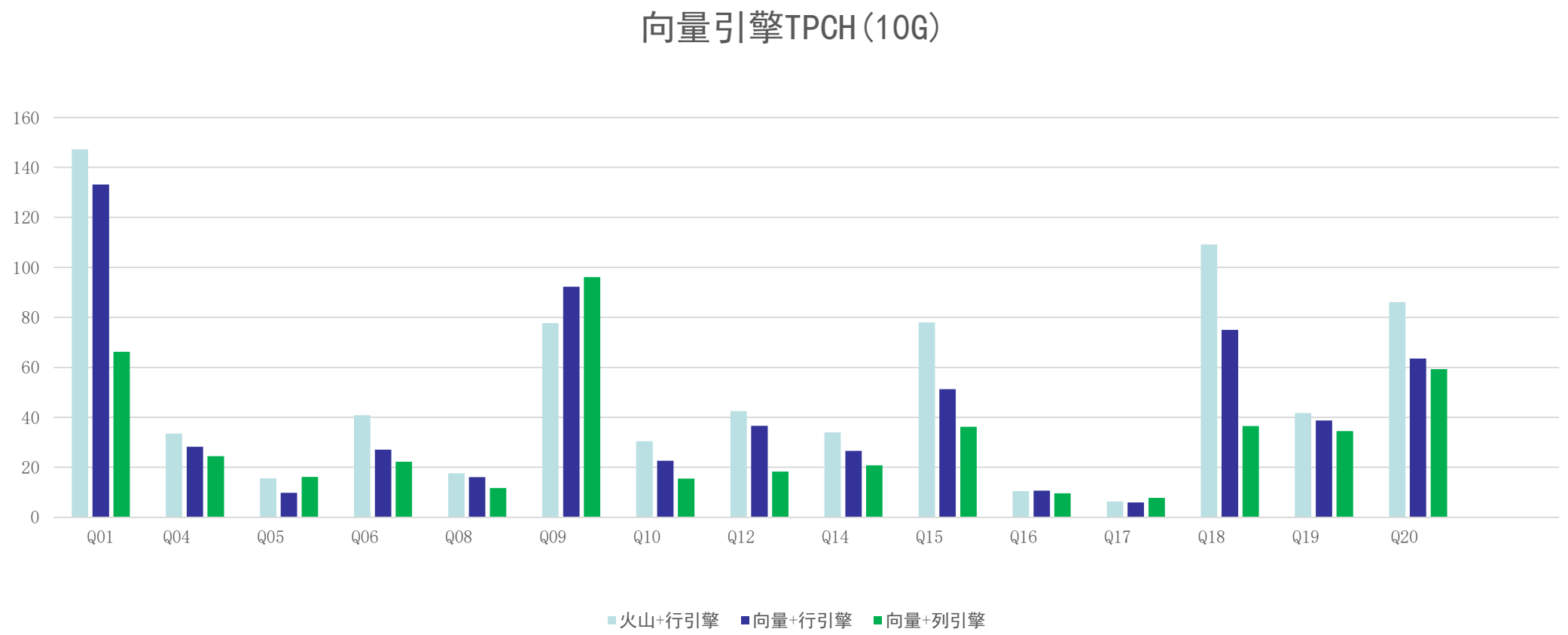
SQL执行分析图



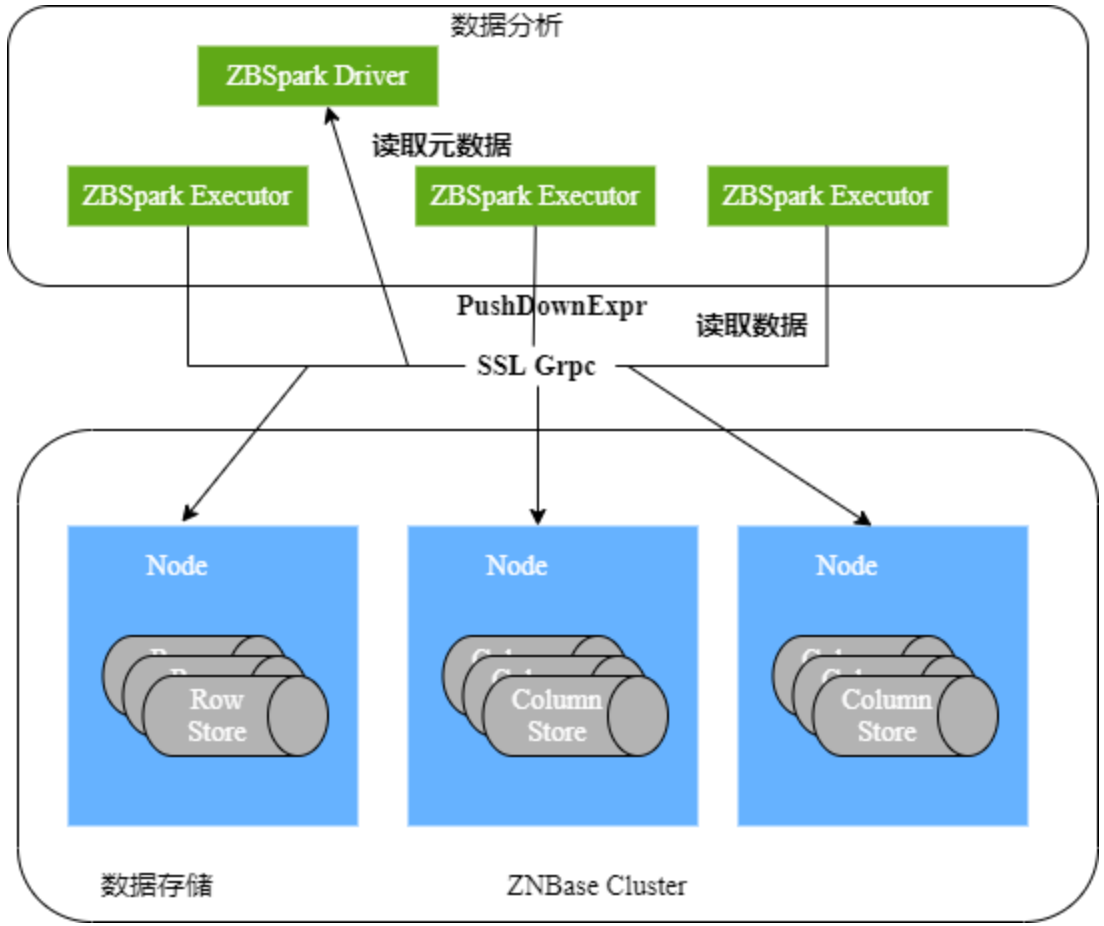
下沉算子

- 过滤器Filter
- Render表达式
- 聚合器Aggregator

TPCH对比


向量引擎TPCH(10G)
160
140
120
100
80
60
40
20
0
Q01
Q04
Q05
Q06
Q08
Q09
Q10
Q12
Q14
Q15
Q16
Q17
Q18
Q19
Q20
火山+行引擎
向量+行引擎
向量+列引擎

计算引擎ZBSpark



Spark计算集成

- 支持Spark2.x和Spark3.x版本
- 直接连接ZNBase节点的底层存储(行存、列存)接口
- 支持过滤算子、聚合算子下推至ZNBase
- 智能选择Range所在的ZNBase节点进行计算
- 数据统计信息支撑Spark Catalyst引擎CBO优化
- 可以使用Spark生态系统中的工具来在ZNBase上进行进一步的数据处理和操作。

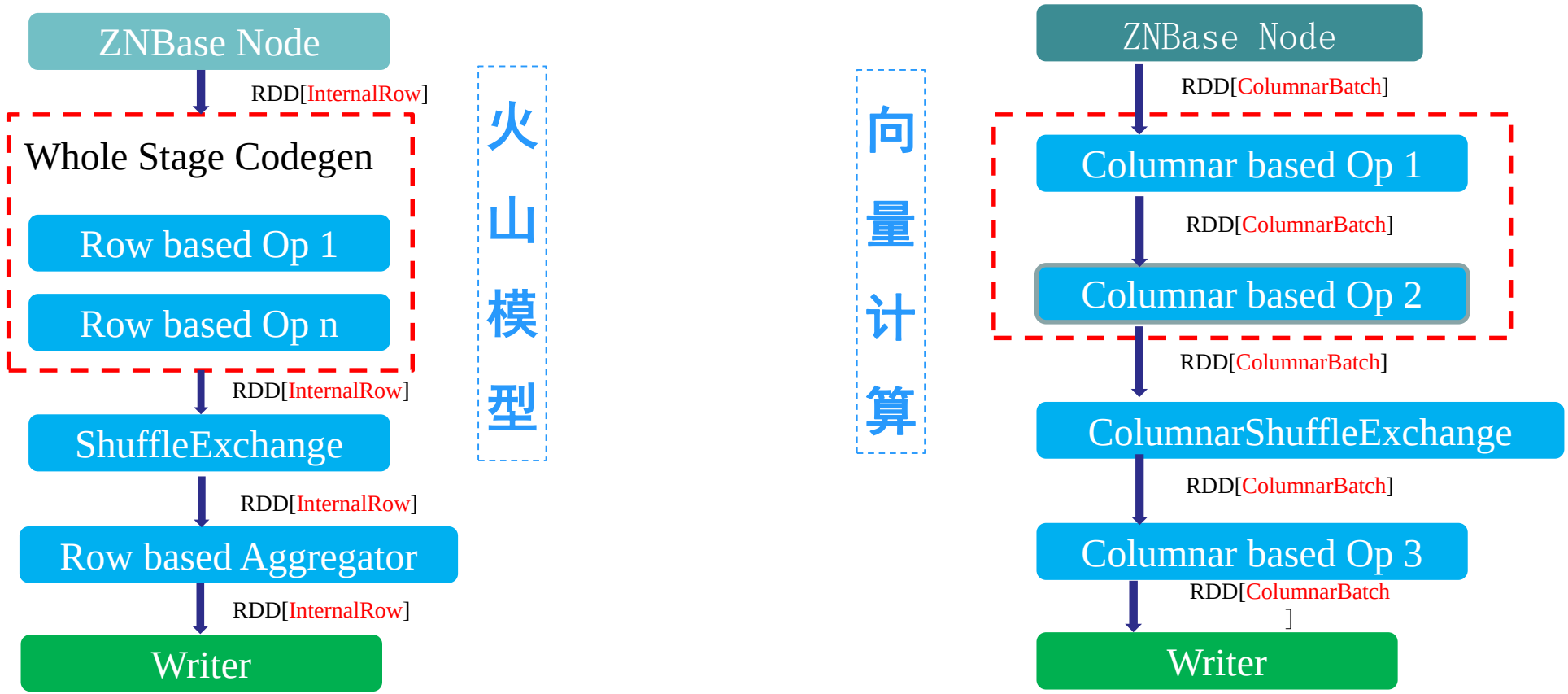
Spark WSCG

- 全称Whole Stage Codegen技术
- 大量虚函数调用，生成的实际代码不再需要执行表达式系统中统一定义的虚函数(compute, execute等)
- 判断数据类型和操作算子等内容的大型分支选择语句
- 常量传播限制，生成的代码中能够确定性的折叠常量

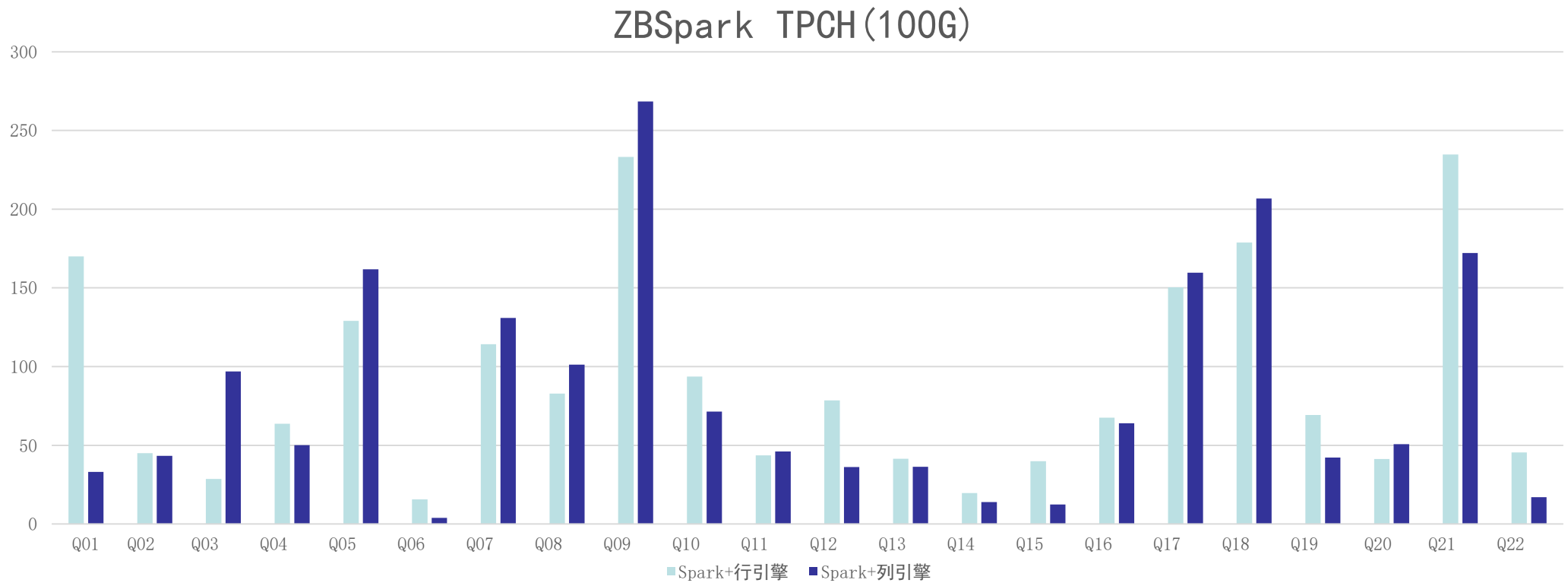
  编译优化时，能够计算出结果的变量直接替换为常量

  多个变量进行计算时，能够直接计算出结果，常量直接替换变量

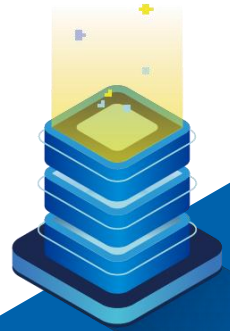
Spark火山&向量计算



TPCH对比


ZBSpark TPCH(100G)
300
250
200
150
100
50
0
Q01
Q02
Q03
Q04
Q05
Q06
Q07
Q08
Q09
Q10
Q11
Q12
Q13
Q14
Q15
Q16
Q17
Q18
Q19
Q20
Q21
Q22
Spark+行引擎
Spark+列引擎

目 录



未来规划

- 列存数据压缩算法优化
  - 针对特定类型数据的专用编解码器(字典压缩、序列编码等)
- Join算子下沉
  - 副本放置策略(亲和)，自适应选择hash连接 or merge连接
- 行列混合查询
  - 子计划自适应选择行计算引擎(火山模型)或向量计算引擎(分批)
- AI赋能逻辑计划、物理计划生成、算子是否下沉
- 随需共享

THANKS

关注我们 / 了解更多