DTCC

第十四届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA

数智赋能 共筑未来

IT168.com ChinaUnix.net ITPUB

北京国际会议中心 | 2023/8/17-19

MatrixOne: 云原生数据库架构设计的机遇和挑战

田丰 CTO， MatrixOrigin

田丰博士

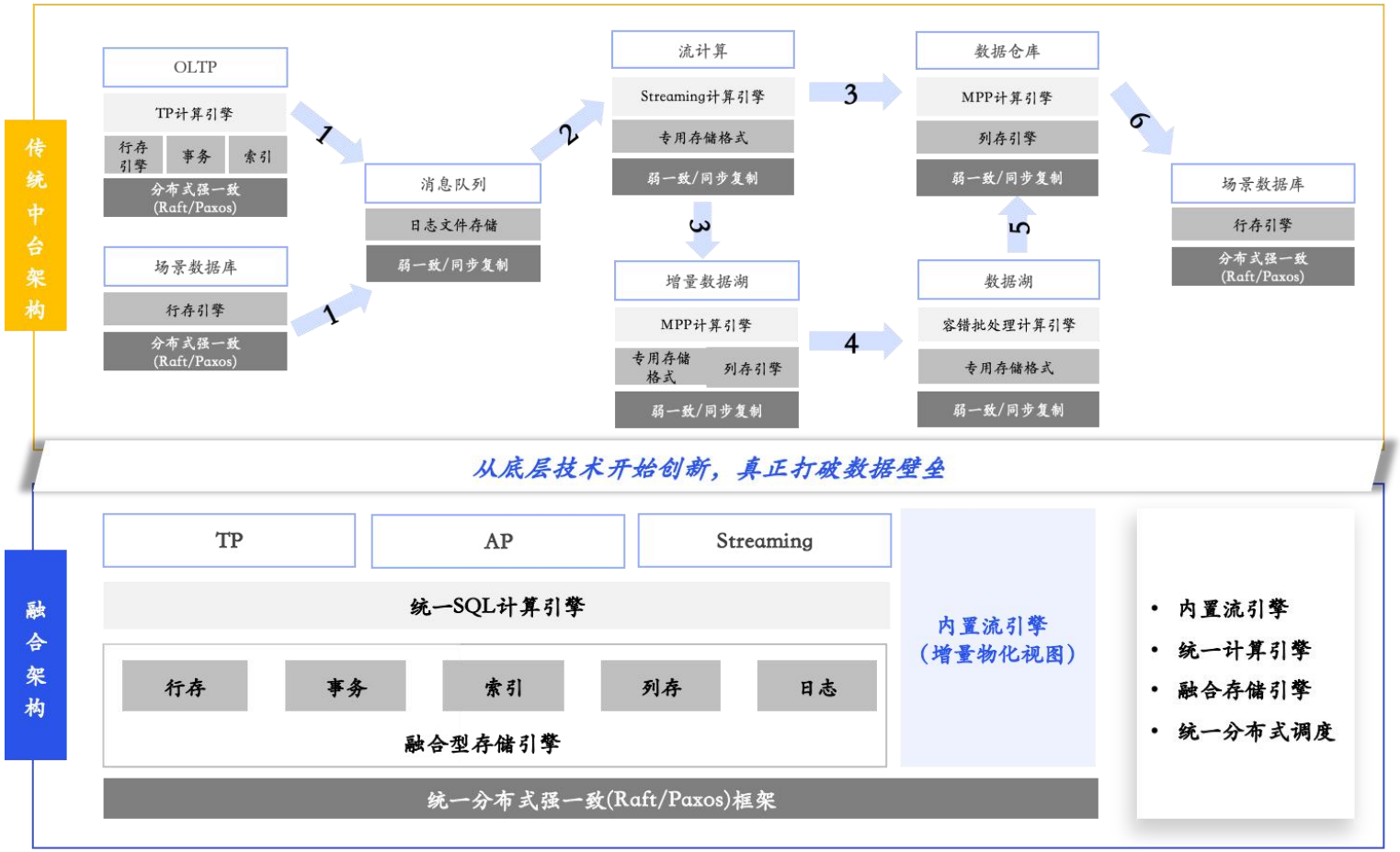
矩阵起源 CTO

- 美国威斯康星大学数据库博士（导师Dewitt）
- Snowflake Chief Engineer，20年数据库内核开发经验
- 原VitesseData/Deepgreen DB创始人CTO（业界最快最稳定的Greenplum）、前微软 / Greenplum 资深工程师/ Vmware Aurora 首席工程师
- 多篇论文入选数据库领域国际顶级会议SIGMOD、VLDB
- 2011年SIGMOD十年最佳论文作者

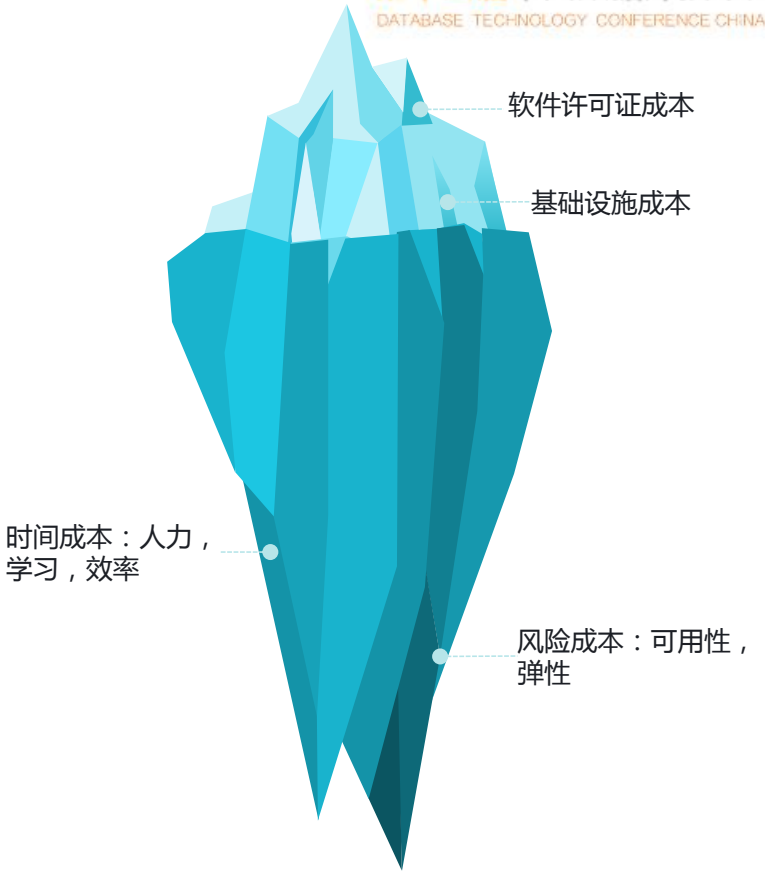
MatrixOne

- 云原生，HTSAP， 数据平台
- 从零开始

数据密集型应用复杂化 → 数据引擎越来越多 → 数据碎片化严重



混合负载需求 → 运维困难 多重学习曲线 → 数据孤岛 一致性问题



MatrixOne Cloud

1

部署自动化

无需关注数据库部署流程

根据业务负载自动扩缩容

2

配置更简单

Serverless 实例，无需关注机器与资源

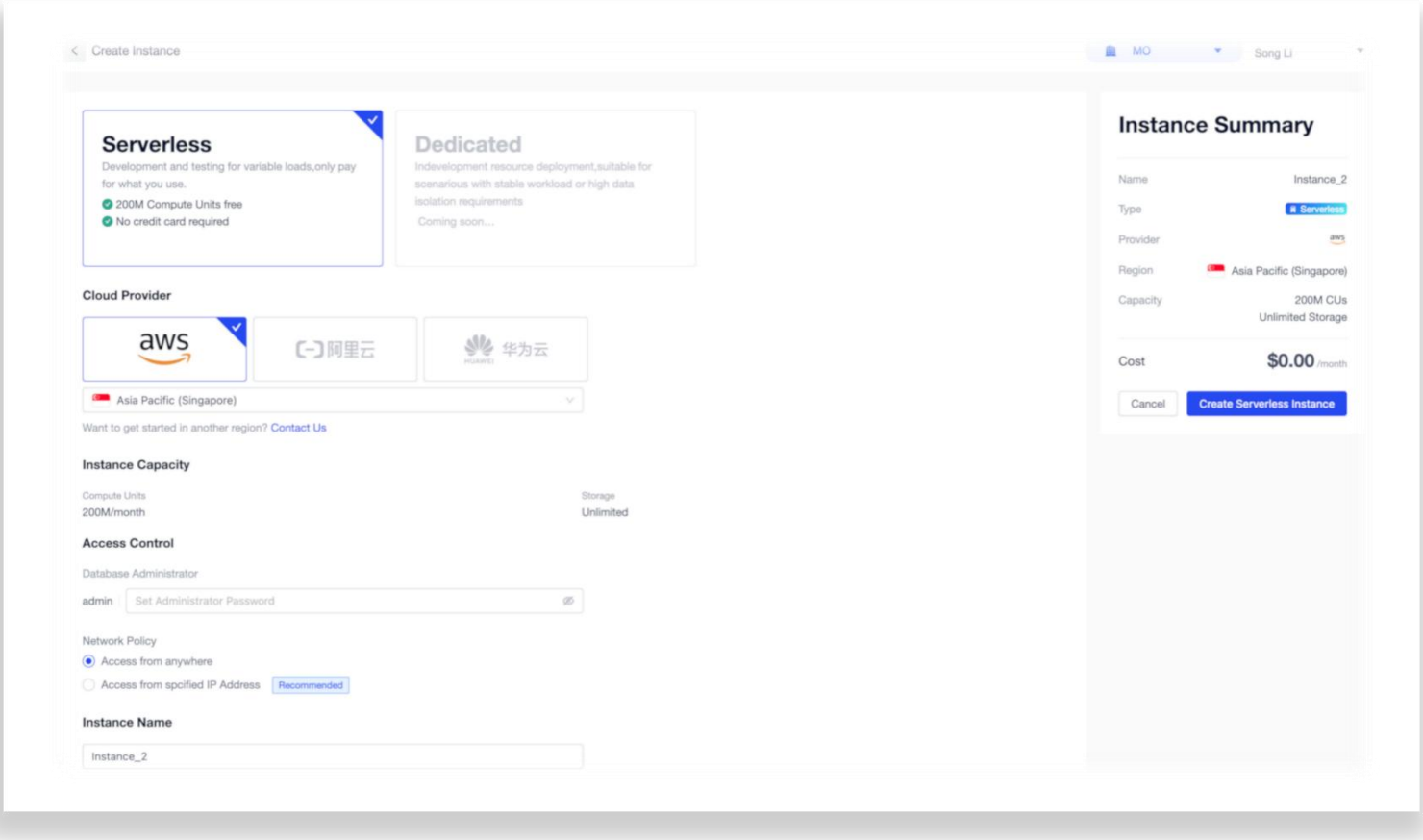
只需关注安全策略、SQL用量上限

3

实例秒级创建

秒级等待，即可完成数据库实例的部署

轻松开启数据库使用旅程



MatrixOne Cloud

1

便捷实例管理

多实例管理界面：实例基础信息和状态

支持实例的一键停止/ 恢复/ 连接

2

实时业务监控

实例使用监控：CU/ Storage/ Connections

SQL性能监控：QPS/ Query Lantency

事务监控：Transaction/ Trasaction Errors

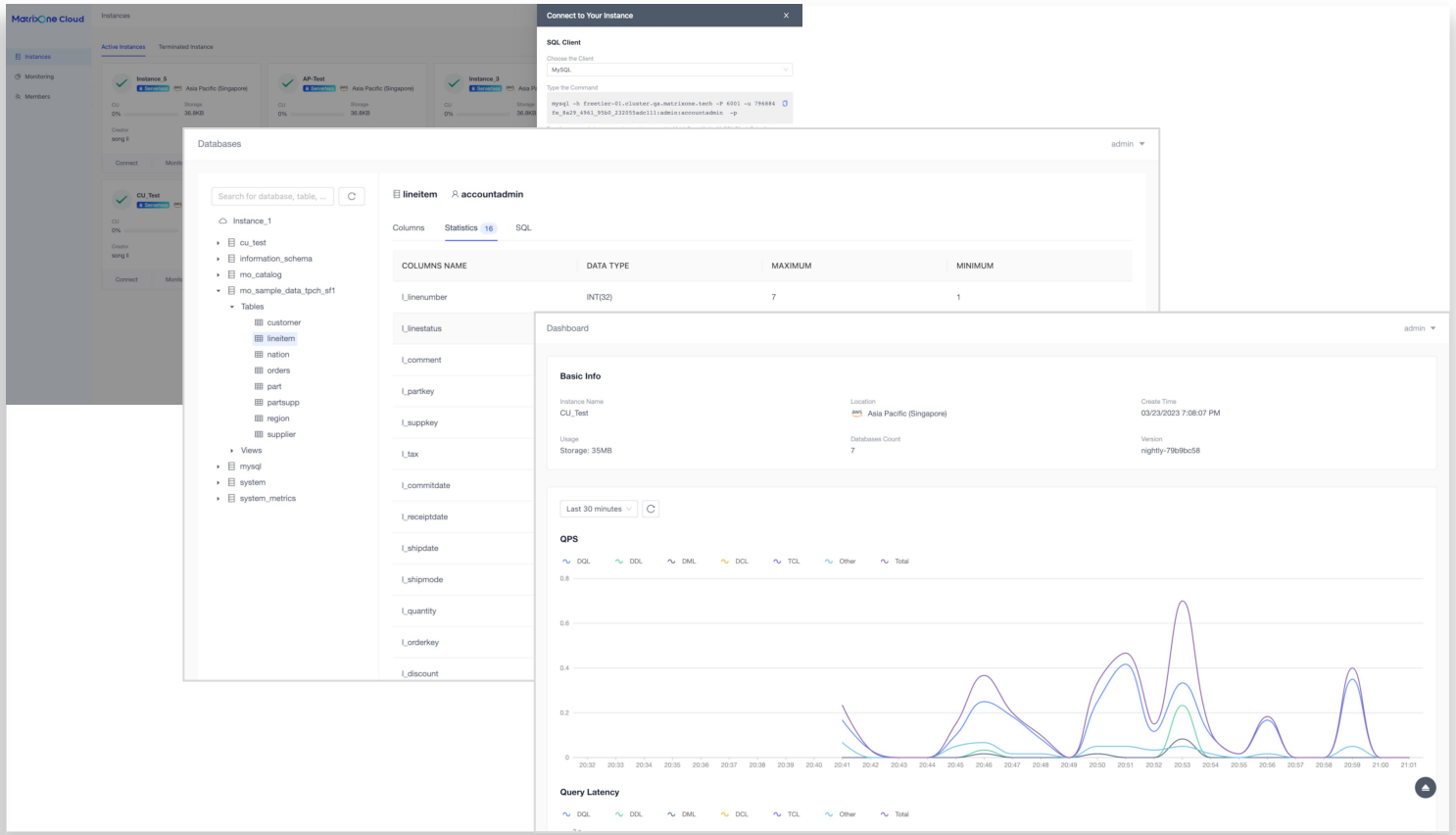
3

可视化数据对象管理

结构化信息展示：层次关系、Schema

数据表的统计和采样：Row/ Size/ Max/ Min

数据对象操作：Drop/ Alter/ Create



MatrixOne Cloud

1

极致 SQL 性能

计算资源自动扩展，高并发也可享受秒级性能

实例间资源相互隔离，SQL 性能互不影响

2

按 SQL 用量计费

以CU (Compute Unit)为计量单位，按每条SQL实际消耗的CU数计费

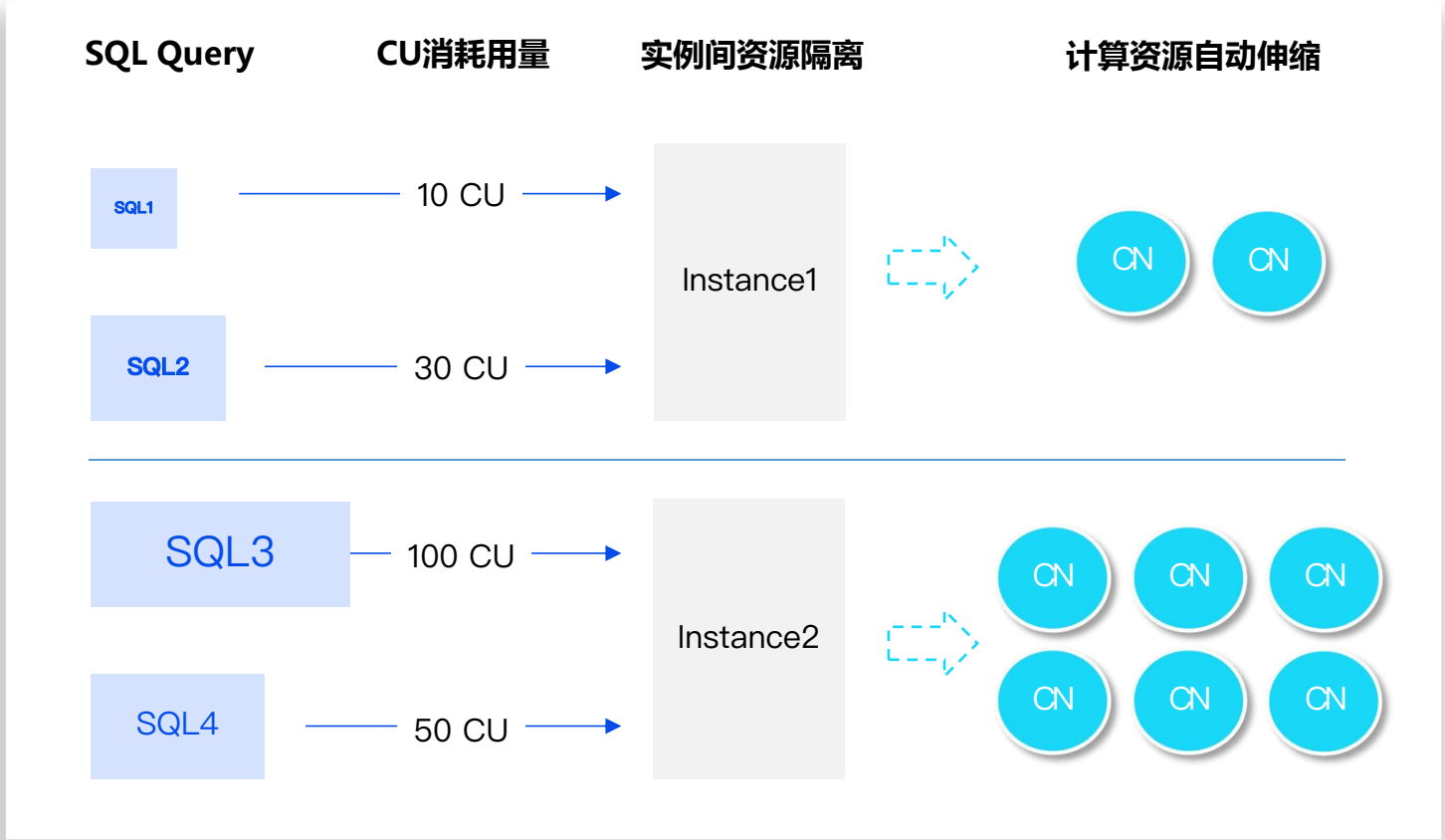
用户只需聚焦SQL本身，无需关注机器资源

3

灵活控制消费速率

支持设定日/ 月的CU消费上限，防止过度消费

精细化管理预算消费速率



MatrixOne Cloud

1

资源隔离

不同的租户间用户数据和负载彼此隔离，有效提高安全性和可靠性

每个租户可以独立扩展伸缩自己的资源

2

统一管理

支持通过系统租户对普通租户进行统一管理

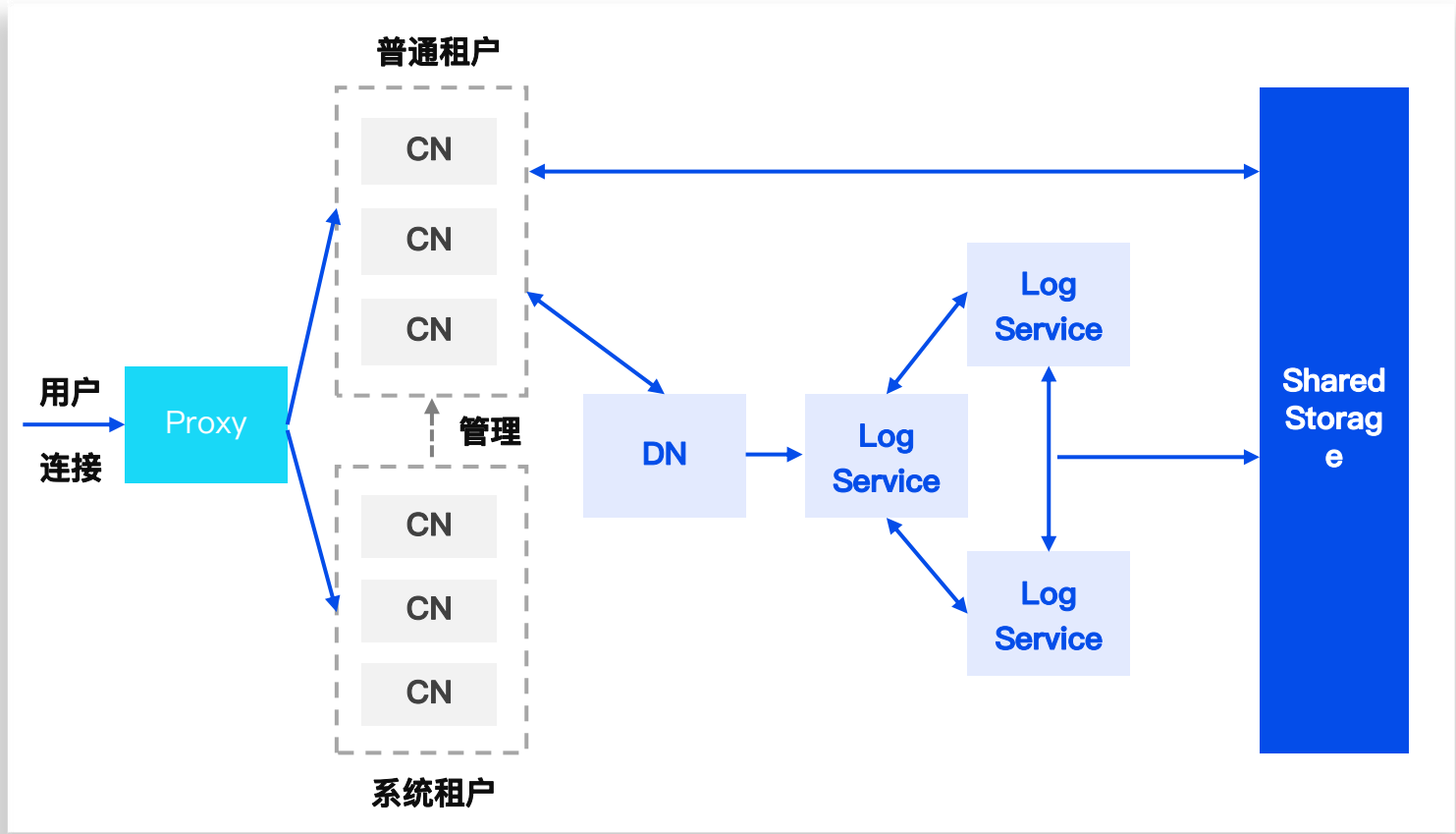
支持创建/删除租户，修改系统配置等

3

数据共享

支持数据发布订阅，快速共享关键数据

支持租户数据分享，灵活满足业务分析需求



云原生

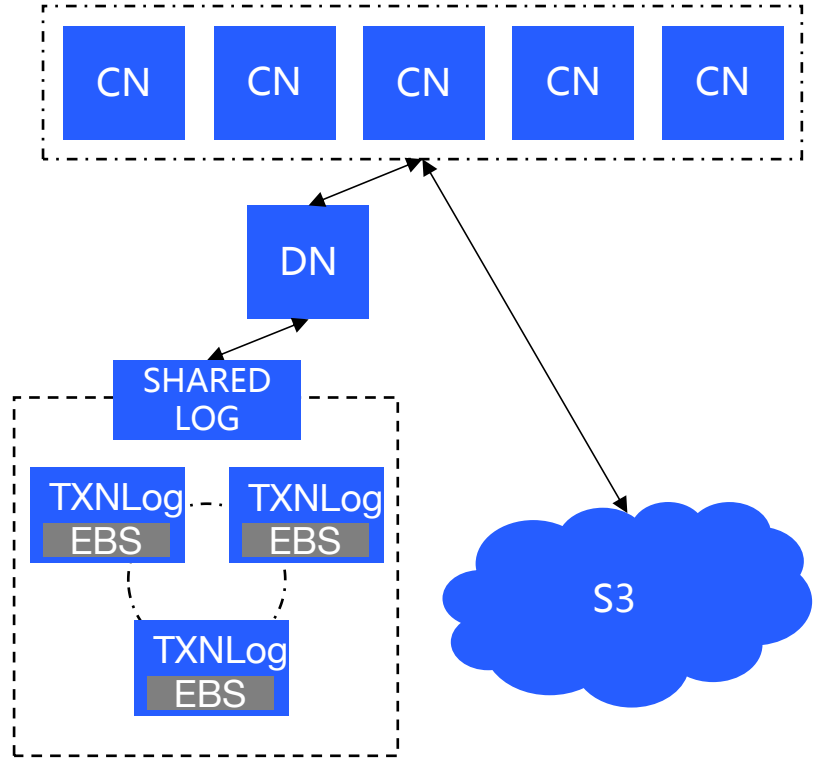
- **云服务的特点**
  - **即刻体验，极简交互，极多场景。**
  - **用户对数据库的需求是动态变化的。**
    - 可能始于TP，但发现使用中需要AP能力， 反之亦然。 所以MatrixOne一开始就定位HTSAP。 我们强调 robust performance，也就是无需客户定义软硬件配置，能够动态适应各种工作负载。
    - 可能从小应用开始，但是数据量和查询复杂度随业务增长而增长。这就要求数据库一定能满足最严苛的扩展性，无上限。云平台为此提供了可能。
    - 性能的标准不再是简单的时延/并发/吞吐，而是性价比。
    - 功能完备 （主键/外键约束）
  - **云服务意味着我们将担负绝大多数运维工作**
    - 日志，功能/性能跟踪，自动调优等

云环境的特点

- **云平台提供了各种服务**
  - **计算资源**
    - 按需使用及付费
    - 近乎无限的资源及扩展性
    - 可以根据负载选择最优的硬件配置/性价比
    - 自动弹性伸缩
  - **低成本、高扩展性的对象存储（S3或类似服务）**
    - 高可用并可以假设存储空间无上限
    - 指数级的存储费用降低和IO费用提升（相比传统块存储）/ 性价比
    - Immutable
    - 挑战： 如何在 S3 上运行 TP 数据库
    - The Design and Implementation of a Log-Structured File System, by Mendel Rosenblum and John K. Ousterhout

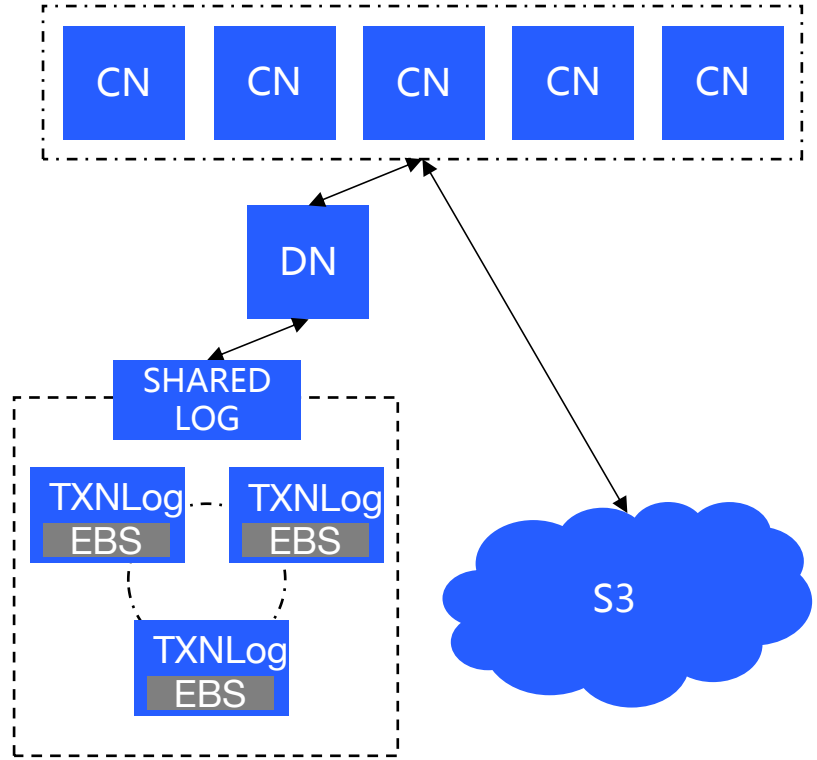
在 S3 上运行 TP 数据库

- Transaction log is database
- 大量的小数据读写
  - **注意力集中在进行写优化。但数据格式偏重读优化**
    - Transaction Log Service，Dragonboat, Raft, 3副本
      - EBS – 我们系统中唯一需要 EBS 的地方
      - 不需要太多资源
      - 及时 checkpoint，只需要很小的 EBS 容量
    - **DN， Transaction Decision Node**
      - 决定事务是否可以commit
      - 不参与复杂计算
      - 无状态
    - **Transaction Log Service 和 DN 共同解决 ACID 中的 A和D**



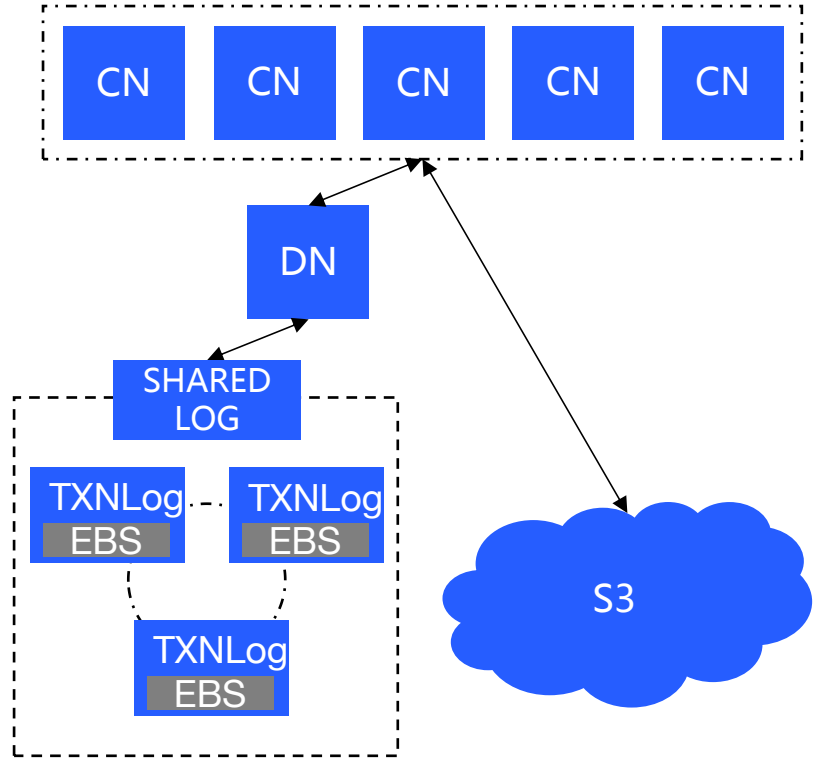
在 S3 上运行 TP 数据库

- CN， Computing Node
  - **ACID 中的 C/I**
  - **系统可以有任意多个 CN。**
  - **主键/外键/唯一性检测，以及所有的查询工作。**
  - **写路径自适应优化**
    - TXN Write
    - Batch Write
  - **读：多级缓存**
    - Cache
    - MVCC/Snapshot Read
  - **无状态**
- Garbage Collection and Storage Optimization



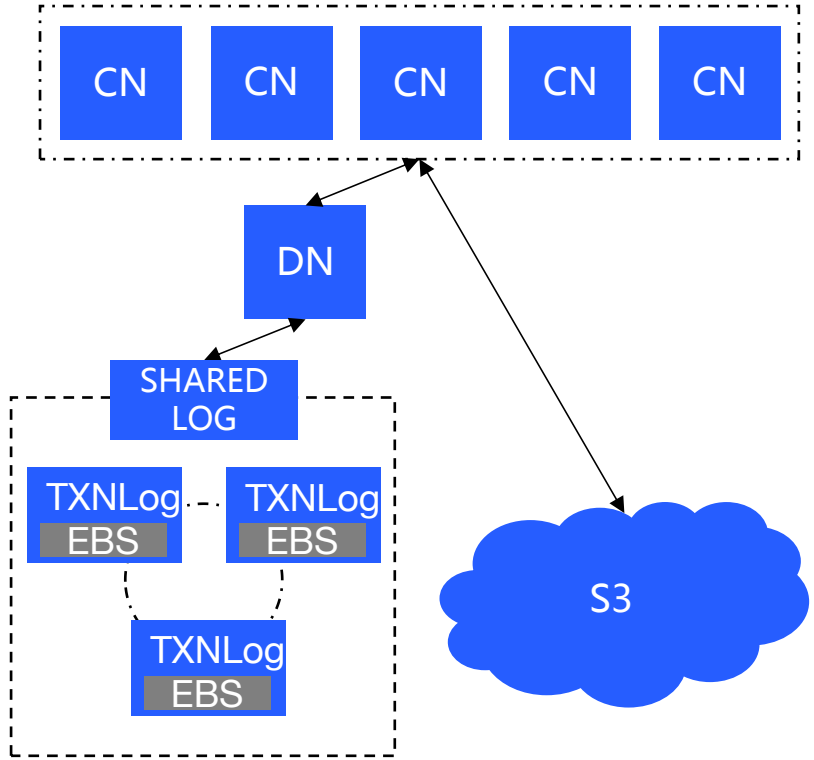
在 S3 上运行 TP 数据库

- TP 性能扩展
  - 多个 CN， 数量动态调整。
  - Proxy



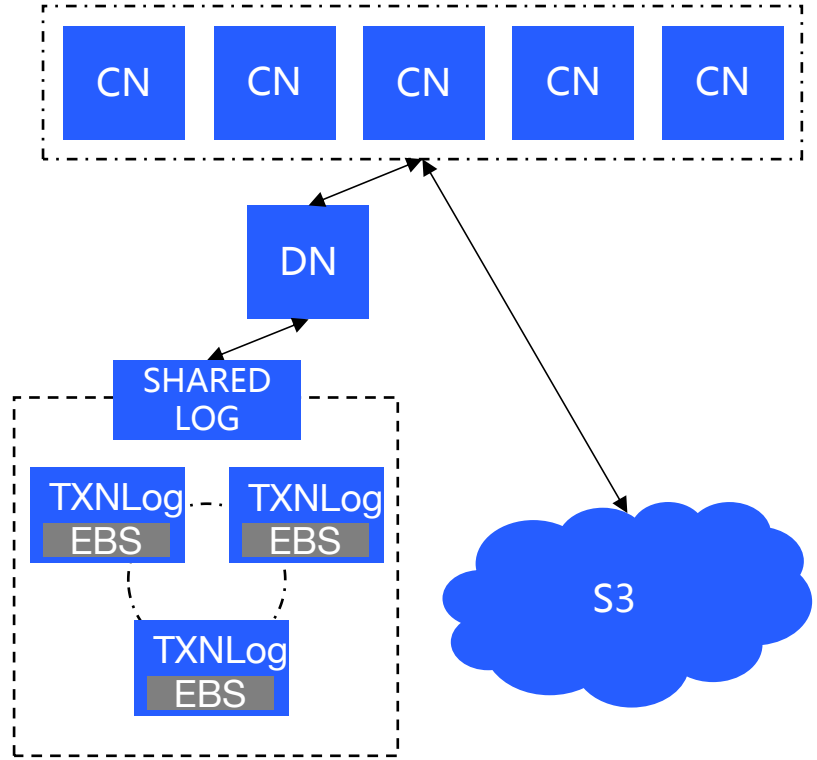
在 S3 上运行 AP 数据库

- 如果不考虑 TP 性能和一致性，相对简单。 挑战在于如果如何同时服务TP与AP
- S3 数据格式是读优化的（列存/压缩/MVCC）
- 查询可以在多个 CN 上并行执行
  - 扩展模式和TP的区别
- 大量的读是 Snapshot Read
- 但大的写操作怎么办？
  - Insert into T select * from OtherT where …
  - Update T … where



在 S3 上运行 AP 数据库

- 优化的用户数据写路径
  - 每个 CN 独立与S3交互（写入）
  - 把各种constraints检测变成查询
  - 事物结束时，通知 DN – 只需要把写入 S3 的object 路径放在事务commit
  - 把大量的数据写操作变成很少的 metadata 写操作
- GC/Storage Optimization：非用户链路上的大读写事务
  - 特殊的优化



自调优

- 大量的统计信息
  - 所有重要查询/优化相关的数据
  - 用户数据读写写路径
- 以统计为导向的产品/代码优化
- 以统计为导向的存储优化
  - Compaction
  - Merge

流数据及其他

- 我们决定不做为某种工作设计的专门的数据库
- 我们做通用数据库，但能够很好的支持各种工作。
  - 以流数据为例
  - 扩展性极强的写性能，同时保持事务强一致性。
  - 扩展性极强的计算能力
  - Snapshot
  - 预计算
    - 用metadata回答查询

AI

- **支持向量**
  - **新的数据类型**
    - Int16vec, f32vec, etc.
  - **函数**
  - **Index**
    - S3 上的数据 （不变）可以被index 加速。
    - Index 和 查询可以用不同数量的，不同配置的计算节点
      - GPU
    - 仍然可以支持事务一致性

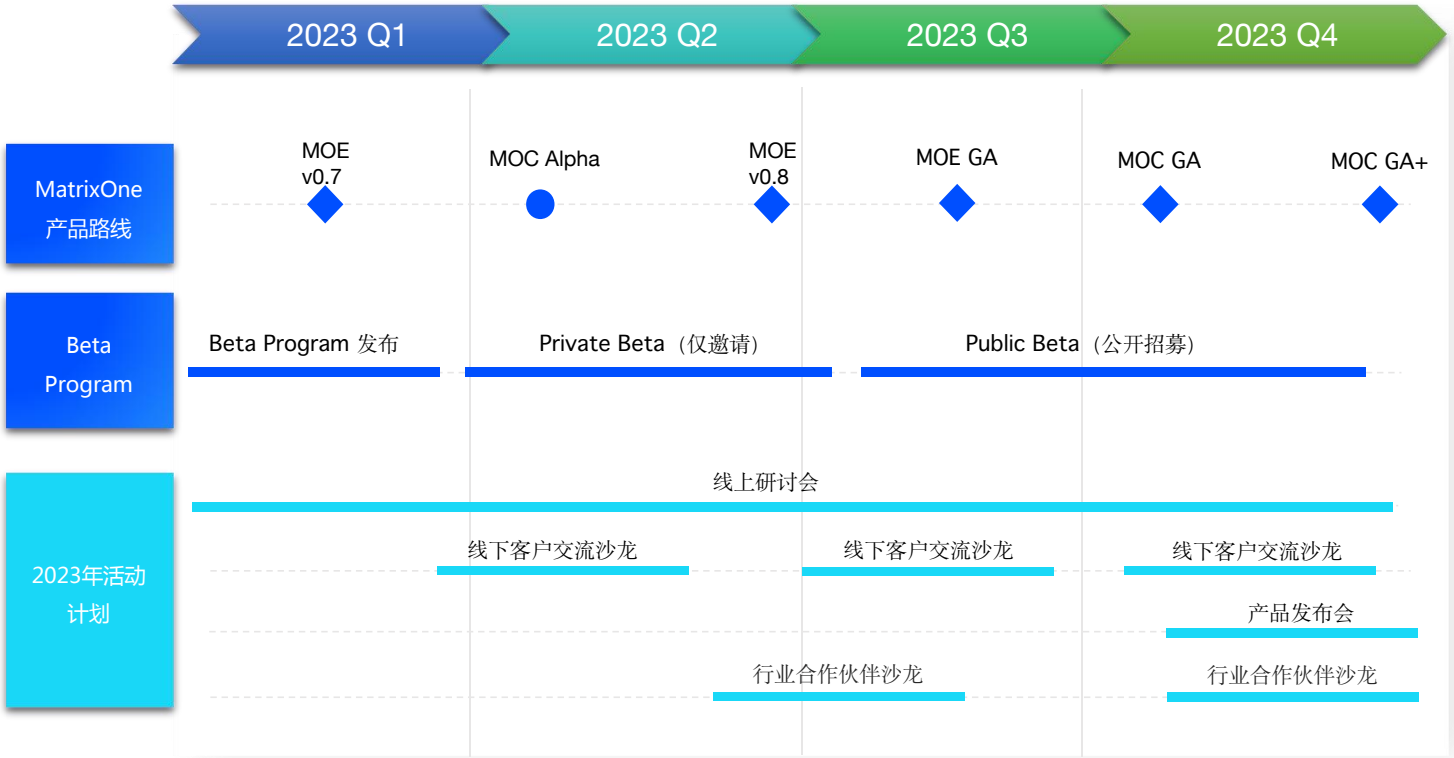
AI Next

- 数据库内部的训练
- 数据库内部对大模型的 finetune
- 原则上我们提供数据/计算平台, 客户无需关心怎样获得硬件资源，以及数据库和 AI Model 之间的数据通信

欢迎加入 MatrixOne Beta Program 用户体验计划

MatrixOne Beta Program 是矩阵起源全新推出的，与客户、用户一起持续提升 MatrixOne 产品和性能体验优化的计划。

- 新功能内测权益
- 产品设计参与权益
- 新功能本地环境优先测试权益
- 开发过程的直接发言权益
- 专家端到端专业支持权益



加入 MatrixOne Beta Program

- Step1：扫描下方小程序码提交注册
- Step2：MO架构师将会通过邮件的方式进行初步联系和沟通
- Step3：加入 Beta Program 社区，开始您和 MatrixOne 的旅程

THANKS