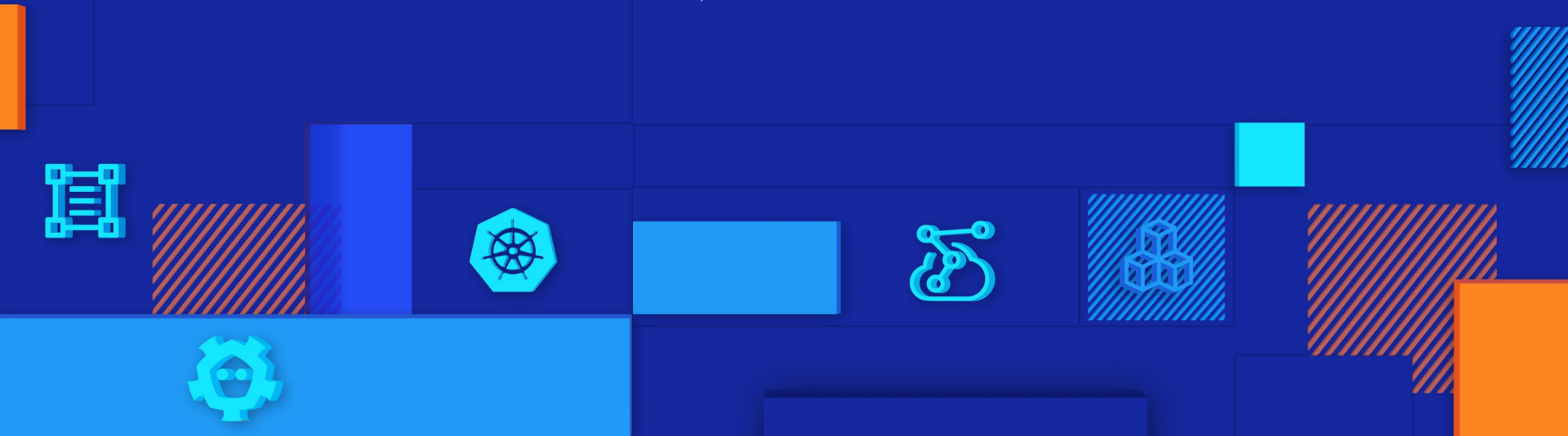
TDSQL-C PostgreSQL的发展思路

窦贤明 | 腾讯云专家工程师



自我介绍

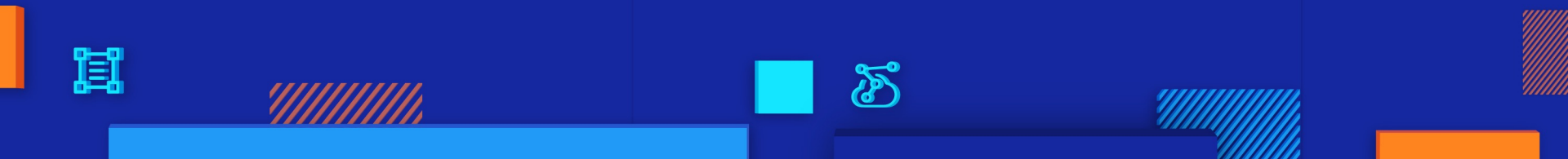
- 窦贤明
- 腾讯云 PostgreSQL/TDSQL-C PostgreSQL 产品研发负责人
- 从零到一研发多款云上数据库产品

TDSQL-C PostgreSQL的发展思路

01 云原生缘起

02 ServerlessDB

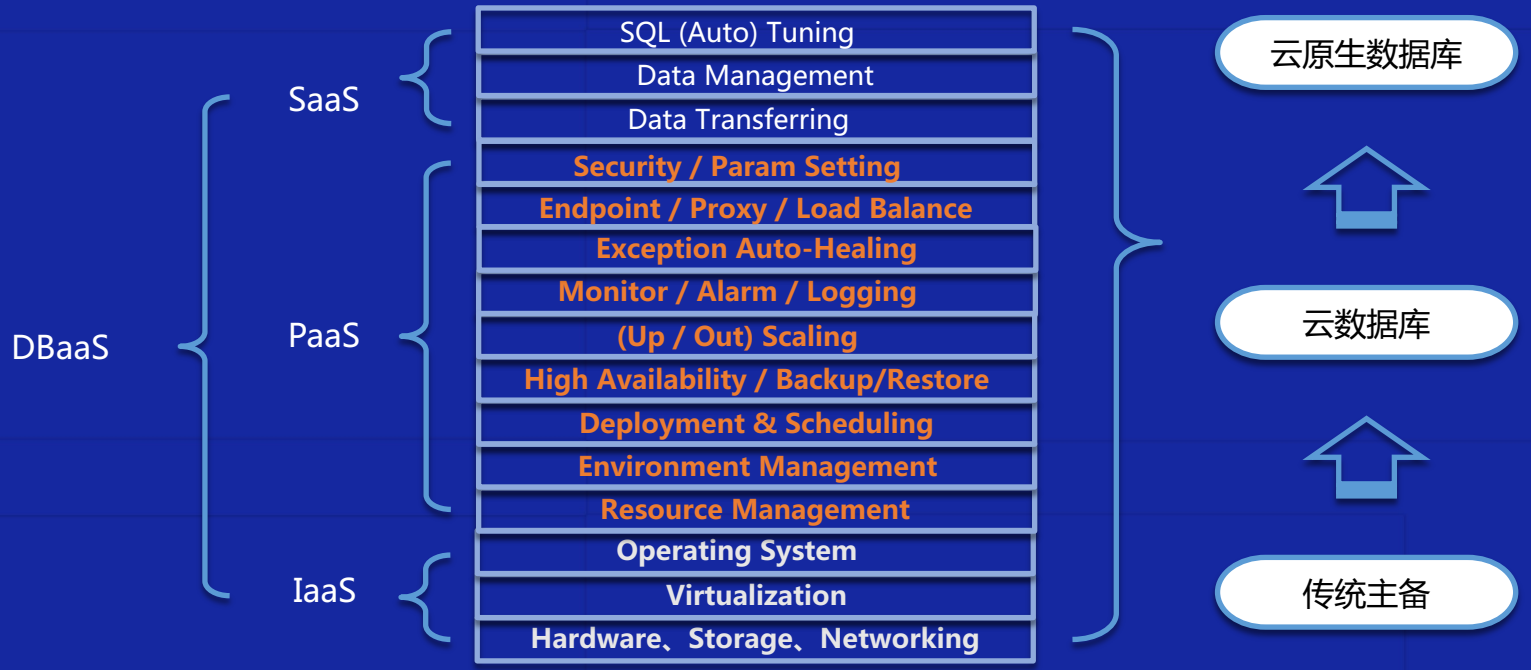
03 分级存储



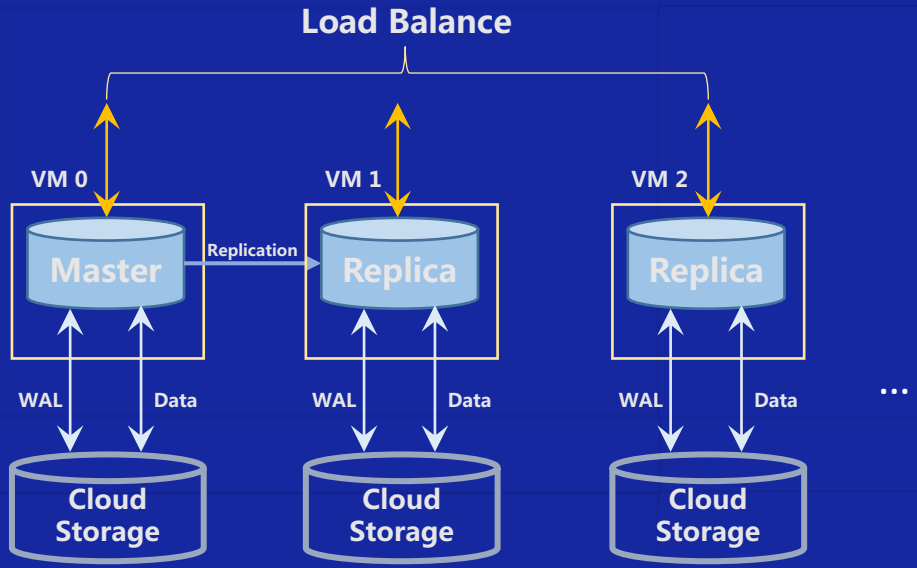
TDSQL-C PostgreSQL的发展思路

云原生缘起

云原生数据库缘起


DBaaS
SaaS
PaaS
IaaS
SQL (Auto) Tuning
Data Management
Data Transferring
Security / Param Setting
Endpoint / Proxy / Load Balance
Exception Auto-Healing
Monitor / Alarm / Logging
(Up / Out) Scaling
High Availability / Backup/Restore
Deployment & Scheduling
Environment Management
Resource Management
Operating System
Virtualization
Hardware、Storage、Networking
云原生数据库
云数据库
传统主备

云原生数据库缘起



云数据库：

- 云形态的第一阶段
- 虚拟化、托管
- 主备方式、内核架构未变

收益：

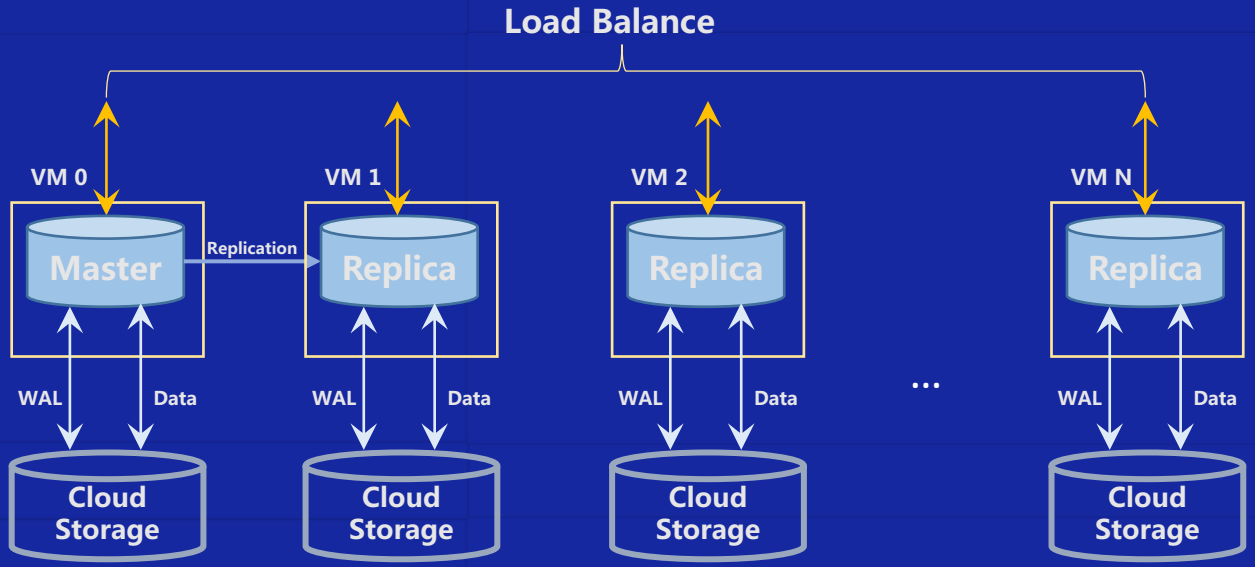
- 资源空间粒度，1vCore、1GB
- 资源时间烁度，小时
- 成本核算的模式 从CAPEX转化为OPEX
- 运维成本低、用户体验提升

云原生数据库缘起

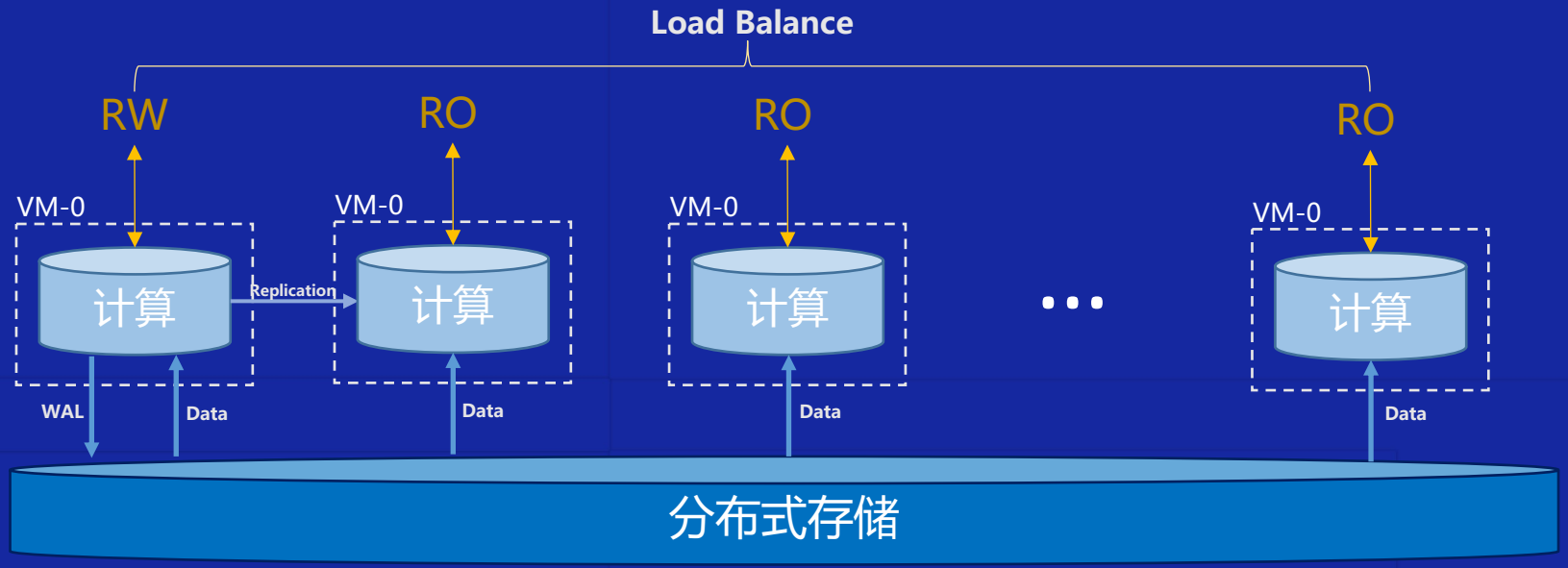
云数据库未解决的问题：

- 资源粒度较粗
  - 时间上以小时计、空间上计实例规格计
  - 存储成本线性增长
  - 网络存在浪费（WAL+Data，数据写两次）
- 调度不够灵活
  - Replica建设成本高，调度成本高
  - HA切换问题，可用性、可靠性的取舍

云原生数据库缘起


Load Balance
VM 0
VM 1
VM 2
VM N
Master
Replication
Replica
Replica
Replica
WAL
Data
WAL
Data
WAL
Data
...
WAL
Data
Cloud Storage
Cloud Storage
Cloud Storage
Cloud Storage

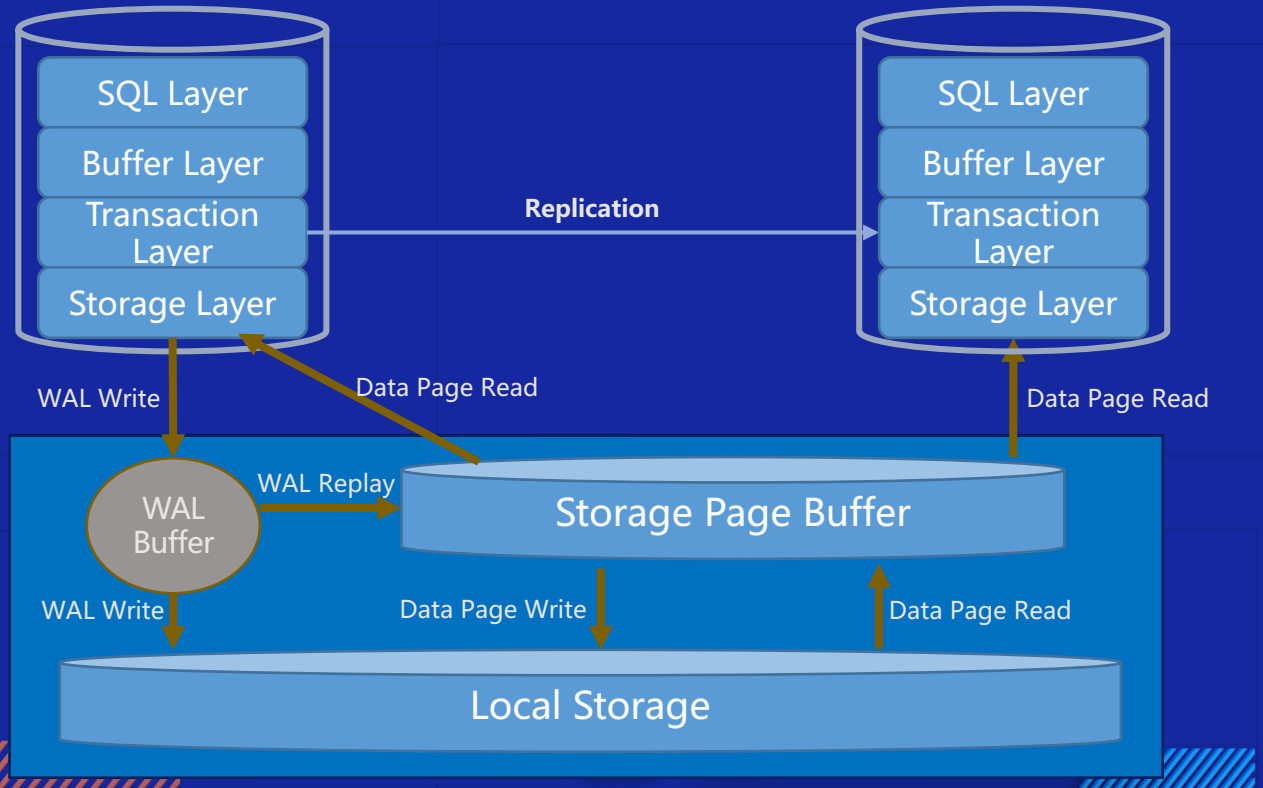
云原生数据库缘起


Load Balance
RW
RO
RO
RO
VM-0
VM-0
VM-0
VM-0
计算
计算
计算
计算
Replication
WAL
Data
Data
Data
Data
分布式存储

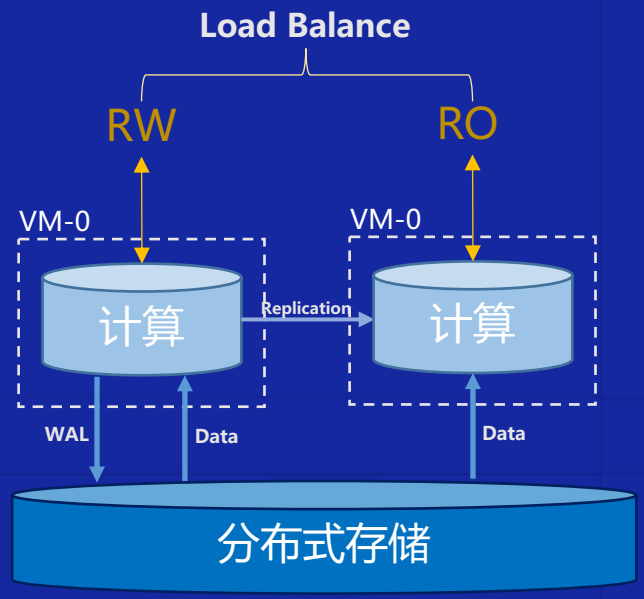
云原生数据库缘起

计算节点

分布式存储



云原生数据库缘起



云原生数据库的优势：

- 更优弹性：存储计算分离，各自动态扩缩容、分别计费
- 更优调度：计算节点秒级拉起，保证可用性
- 更优调度：状态持久化于分布式存储中，保证数据可靠性
- 更低成本存储：全局一份（三副本），RO越多存储成本相对越低

TDSQL-C PostgreSQL：

- Master 和 RO 基于一份数据，放在共享存储
- Master 仅将WAL写入共享存储、不写 **数据页**
- RO 从共享存储中读取所需 **数据页**，无须写存储
- RO 从主库接收 WAL、缓存中重放，保持缓存最新
- 共享存储接收并重放 WAL，实现存储节点上**数据页**的修改
- 存储层以 Page 为单位维护数据

云原生数据库缘起

传统主备

运维：

- 全栈运维、复杂度高
- 人工或脚本化运维
- 基本无 SaaS能力、或人工
- 或自研体系、成本高

成本：

- 机器为固定资产方式
- 采购、维护成本高
- CAPEX方式核算成本

云数据库

运维：

- 运维工作极大减少
- 充分利用SaaS能力
- 众多场景打磨，可靠性更高

成本：

- 计算费用粒度为小时级
- 根据业务需要随时升降
- 标准化、采购成本低
- 可以按OPEX方式核算成本

云原生数据库

运维：

- 运维工作极大减少
- 充分利用SaaS能力
- 可用性、可靠性更优

成本：

- 计算费用粒度为小时级
- 存储按实际用量计费
- 标准化、采购成本低
- 可以按OPEX方式核算成本

TDSQL-C PostgreSQL的发展思路

ServerlessDB

ServerlessDB

计算与存储分离，**计算更轻**、存储更重

ServerlessDB

云的本质：

- 弹性
- 自动化

演进思考：

- 计算更轻，**轻至不存在**
- 计费更细，更细时间粒度
- 用户只关心：
  - 地址
  - 计费
  - 运维

ServerlessDB：

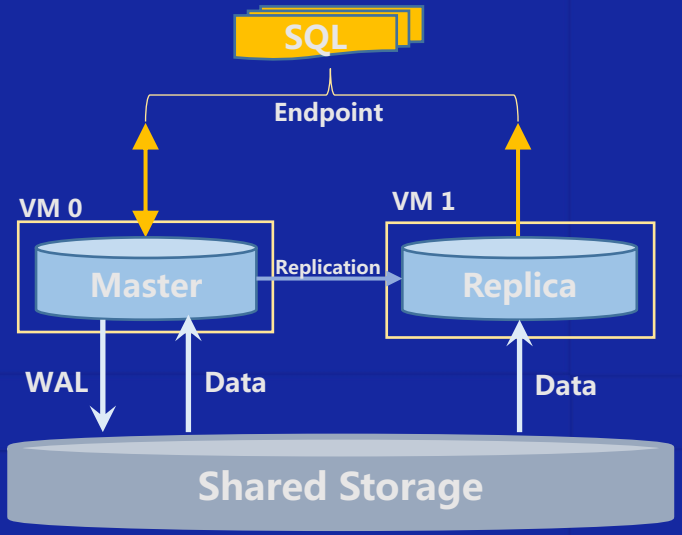
- 不再有实例，只是一个 Endpoint
- 计算资源时间粒度为秒级
- 无运维、全自动


ServerlessDB

产品形态：

- 存储、计算分别计费，且粒度更细
- 计算节点，**不用不计费、用多长计多少**
- 计算节点，依据负载自动扩缩容
- 存储空间，用多少计多少
- 全程自动化、无需“人工”（或脚本）干预
- 用户只需关心：
  - 访问地址
  - 计费
  - 业务周期

ServerlessDB



初始状态：

- 存储存在、地址存在
- 计算节点不存在
- **没有业务**

SQL 运行时：

- 存储存在
- 计算节点被拉起
- **业务被执行**

SQL 运行结束后一段时间（数秒）：

- 存储存在
- 计算节点被关闭，不再计费

ServerlessDB

云原生数据库

运维：

- 运维工作极大减少
- 充分利用SaaS能力
- 可用性、可靠性更优

成本：

- 计算费用粒度为小时级
- 存储按实际用量计费
- 标准化、采购成本低
- 可以按OPEX方式核算成本

ServerlessDB

运维：

- 无运维
- 充分利用SaaS能力
- 可用性、可靠性更优

成本：

- 自动扩缩容
- 计算费用粒度为秒级
- 存储按实际用量计费
- 标准化、采购成本低
- 可以按OPEX方式核算成本

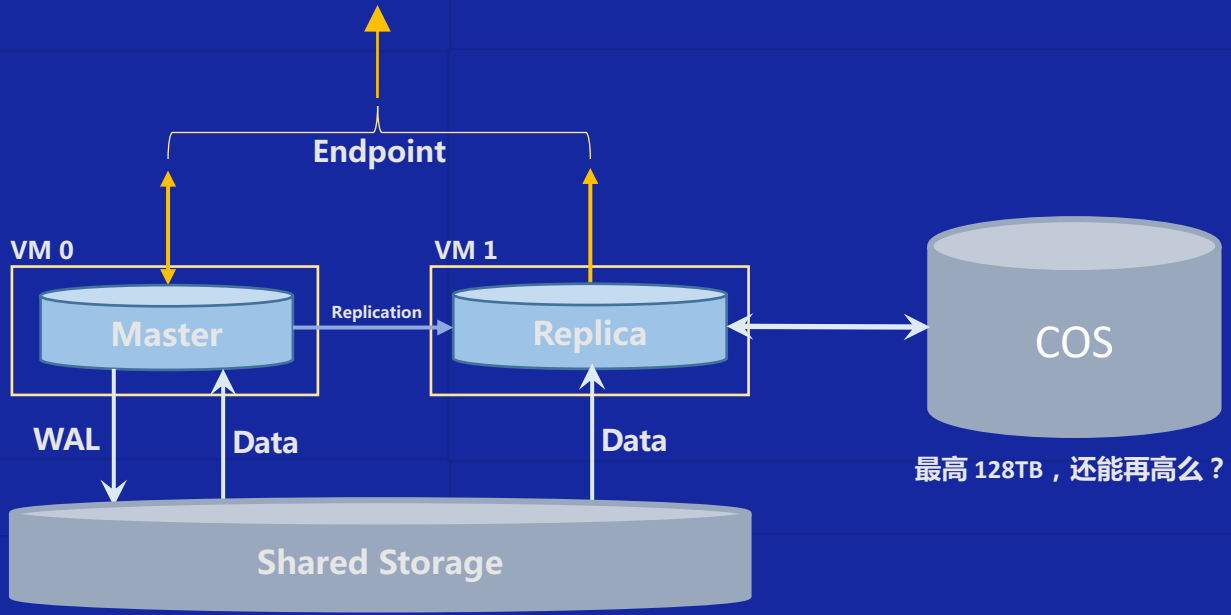
TDSQL-C PostgreSQL的发展思路

分级存储

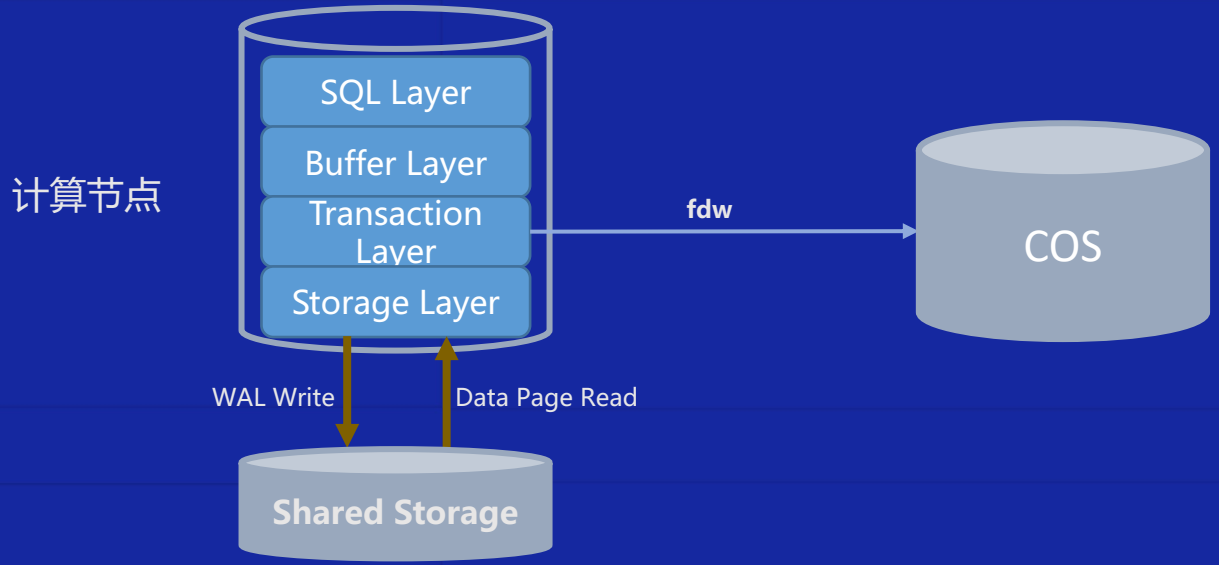
分级存储

计算与存储分离，计算更轻、**存储更重**


分级存储


Endpoint
VM 0
VM 1
Master
Replication
Replica
COS
WAL
Data
Data
Shared Storage
最高 128TB，还能再高么？

分级存储


计算节点
SQL Layer
Buffer Layer
Transaction Layer
Storage Layer
fdw
COS
WAL Write
Data Page Read
Shared Storage

分级存储

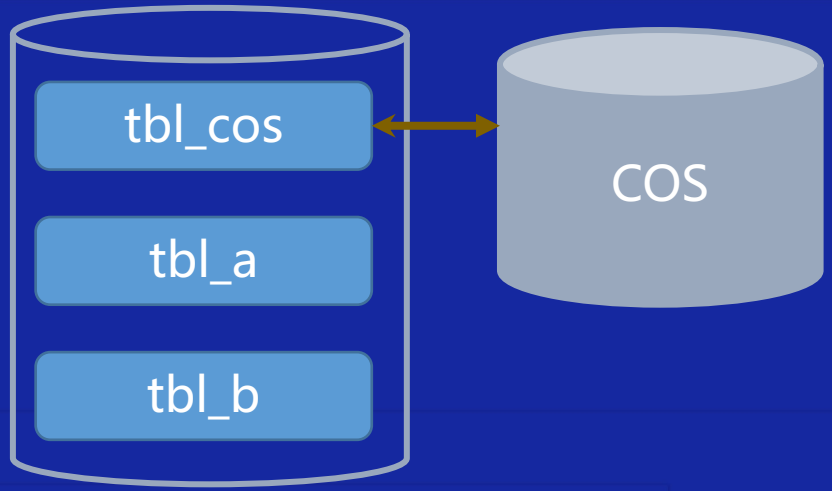
```
CREATE SERVER cos_server FOREIGN DATA WRAPPER cos_fdw OPTIONS(
                        host 'xxxxxx.cos.ap-nanjing.myqcloud.com',
                        bucket 'xxxxxxxx',
                         id 'xxxxxxxx',
                        key 'xxxxxxxxxx'
);
CREATE FOREIGN TABLE multi_csv (
                        word1 text OPTIONS (force_not_null 'true'),
                        word2 text OPTIONS (force_not_null 'off')
) SERVER cos_server OPTIONS (
                        filepath '/a.csv,/b.csv,/c.csv.2',
                        format 'csv',
                        null 'NULL'
);

postgres=# EXPLAIN SELECT * FROM multi_csv;
                                        QUERY PLAN
-------------------------------------------------------------------------------
Foreign Scan on multi_csv (cost=0.00..1.20 rows=2 width=128)
                        Foreign COS Url: https://xxxxxxxxxx.cos.ap-nanjing.myqcloud.com
                        Foreign COS File Path: /a.csv,/b.csv,/c.csv.2
                        Foreign each COS File Size(Bytes): 15,172,86
                        Foreign total COS File Size(Bytes): 273
(5 rows)
```

分级存储

APP 0

```
SELECT
                a.id, b.name, c.value
FROM
                tbl_a a, tbl_b b,
tbl_cos c
WHERE a.id = b.a_id and b.id = c.b_id
...
```



分级存储

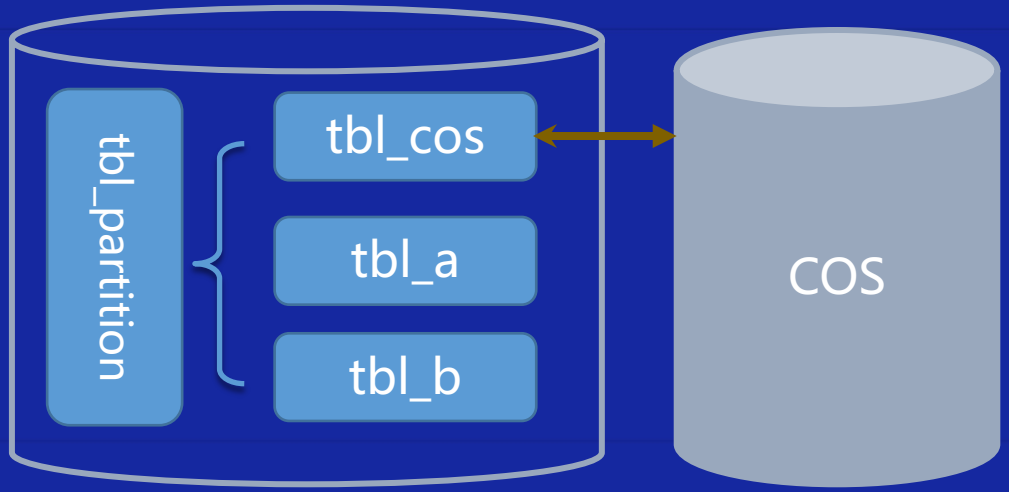
APP 0

SELECT *
FROM

WHERE

tbl_partition

date > ’ 2022.04.30’



THANKS!