

OceanBase赋能SaaS场景探索与思考

冯光普 @ 多点DMALL

2023.06

个人介绍

冯光普

- 大规模MySQL、TiDB、Redis、MongoDB等集群**安全、稳定、高效**
- **DB PaaS平台建设**，为研发体系输出标准化流程、快速交付能力
- 同城双活，异地多活**数据库架构、数据同步**方案落地

更早，在阿里巴巴负责：AliSQL分支维护、特性开发、Bugfix

主要内容

- 零售SaaS场景特点
- 为什么要使用分布式数据库？
- OceanBase有什么不一样？
- 为什么OceanBase性能好？
- 为什么OceanBase成本更低？

零售SaaS场景特点

零售SaaS场景特点

业务规模

120+
连锁零售商¹

跨国零售商	  
超市	  
便利店	   
专营零售	 
新业态	 
	 

467
国内品牌²

国内品牌	      
国内品牌	      
国际品牌	     
国际品牌	     

备注1：截至2021年6月30日数据
备注2：截至2021年3月31日数据

零售SaaS场景特点

技术挑战

微服务架构

- 应用拆分细
- 数据库多 (500+)
- 持续增加

复杂度

多租户设计

- 共用数据库
- 数据增长快
- 单租户大

扩展性

新环境部署

- 资源成本高
- 初始分配难
- 后期扩容难

成本

为什么要使用分布式数据库？

为什么要使用分布式数据库

扩展性

- **MySQL单库朝着10TB飞奔，DBA很慌**
 - 催研发：清理/归档
 - 扩容、迁移（以后更慌）
 - 分库分表（耗时数月、SQL有损）



By Bing Chat

✓使用分布式数据库 -> 透明扩展

数据库	数据量	月增长
l_prd_push	29775	1.61%
l_sale	21075	1.16%
l_accounting	18790	0.67%
l_financial_engine	15377	0.8%
l_inventory_voucher	14658	0.59%
l_supplier	13781	-3.37%
l_payment	13726	0.52%
l_physical_stock_tracking	9992	0.44%
l_order	9378	1.3%
l_order_product	9252	0.67%
l_inventory	8900	0.29%
l_delivery_online_tracking	8329	1.19%
l_sales	6860	2.25%
l_inventory_center_mq	6611	0.88%
l_rdc_buiche	6452	1.31%
l_project_log_record_01	5305	0.11%
l_order_product	4658	1.87%
l_inventory_center	4598	0.47%
l_intention	4481	0.61%

为什么要使用分布式数据库

成本

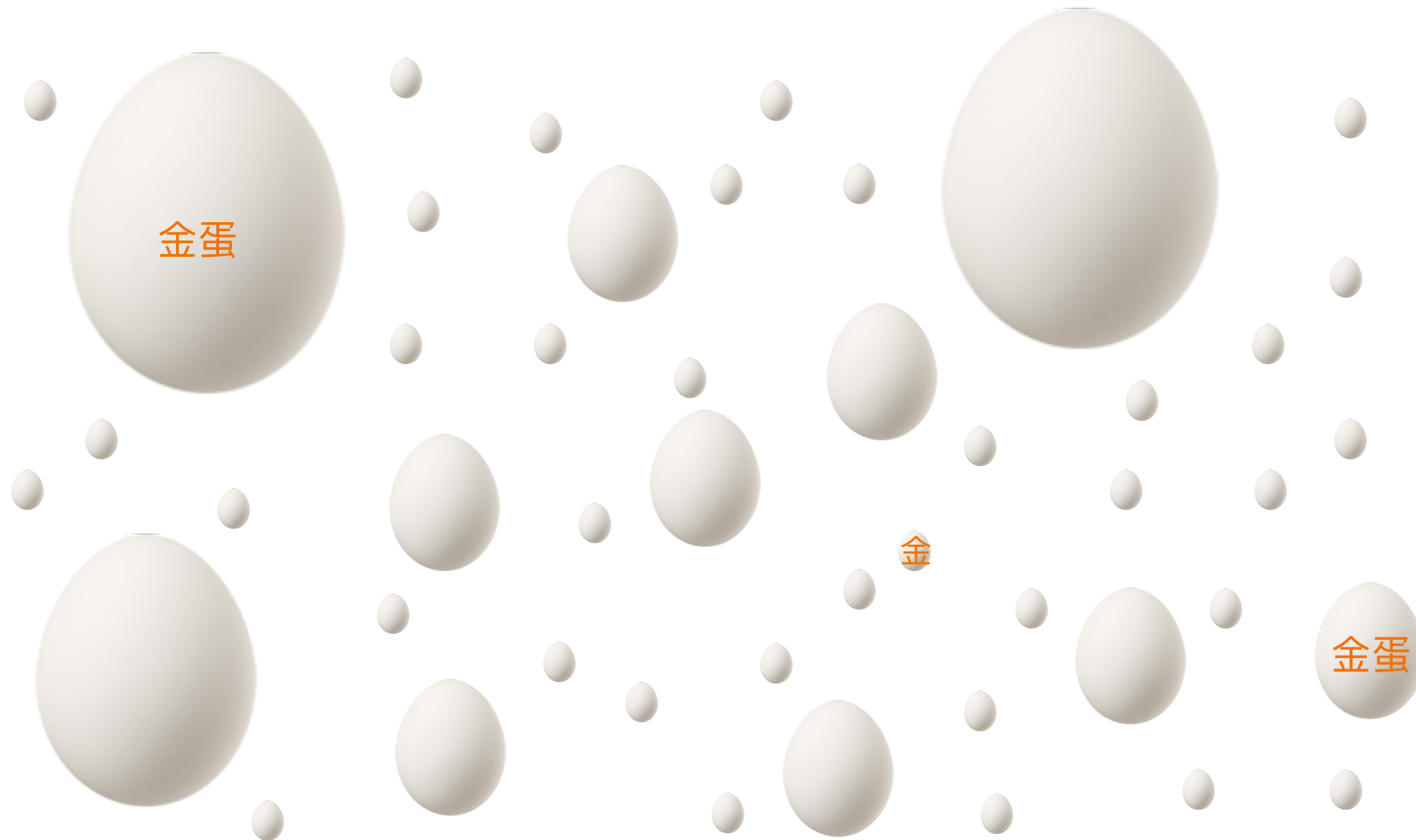
- 鸡蛋 = 数据库
- 篮子 = 集群

DBA：挪蛋是我的工作，蛋不能碎！

😊 1个篮子1个鸡蛋甚好
😂 篮子比蛋贵！

运维成本

资源成本



✓使用分布式数据库 -> 自动挪蛋

为什么要使用分布式数据库

高可用 (RPO=0, RTO<30s很难)

- 外挂方式 无法彻底避免脑裂

MHA

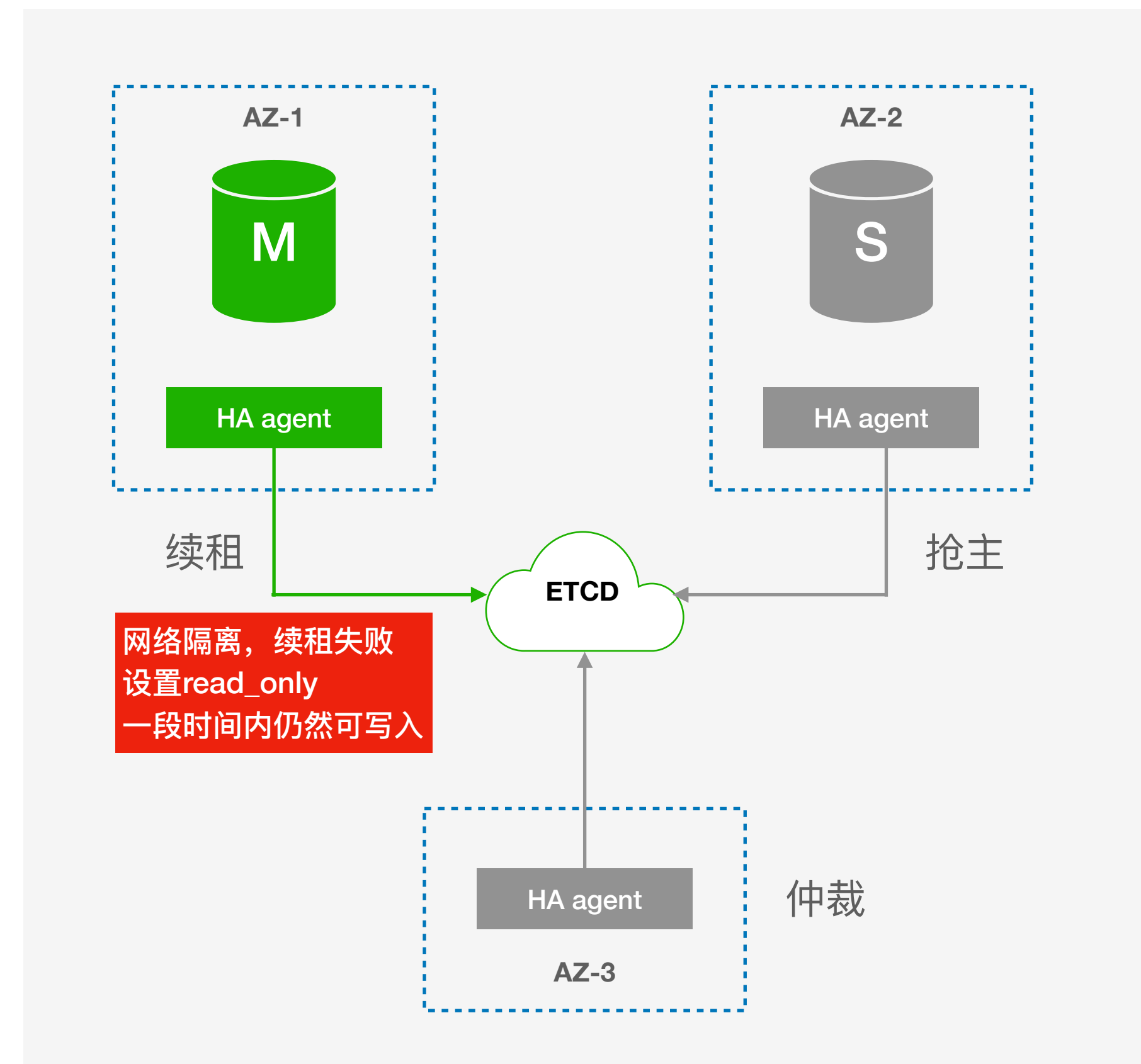
Orchestrator

- 内置实现

Group Replication

OceanBase
TiDB

✓使用分布式数据库 -> 金融级高可用



同城双活，
我们的高可用方案

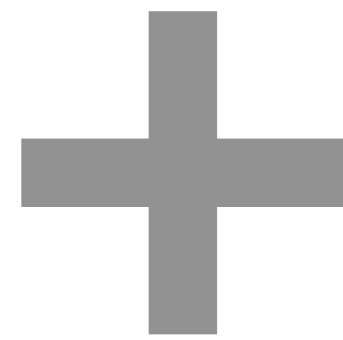
OceanBase有什么不一样？

OceanBase有什么不一样

单机分布式一体化，在说啥？

单机

低延迟，高性能



分布式

易扩展，大容量

- 业务初期：数据量不超过单机，享受MySQL般极致性能
- 业务壮大：可透明扩展，几乎无限容量
- 全生命周期：不改代码、不停机迁移、依然高性能

OceanBase有什么不一样

可扩展、高性能、低成本、高可用

可扩展

透明扩缩容：分片自动迁移，负载均衡

高性能

最小化分布式开销的架构设计

低成本

相对MySQL节省60%+、多租户能力

高可用

极致追求：RPO = 0, **RTO < 8s**

为什么OceanBase性能好？

为什么OceanBase性能好

计算存储不分离

网络是性能杀手

OBProxy

数据路由，SQL转发

OBServer

对等架构：计算 + 存储

内存延迟
~0.1us

内存吞吐
> 100GB/s

SSD延迟
~0.1ms

SSD吞吐
~1GB/s

网络延迟
LAN: ~0.1ms
同城: ~3ms

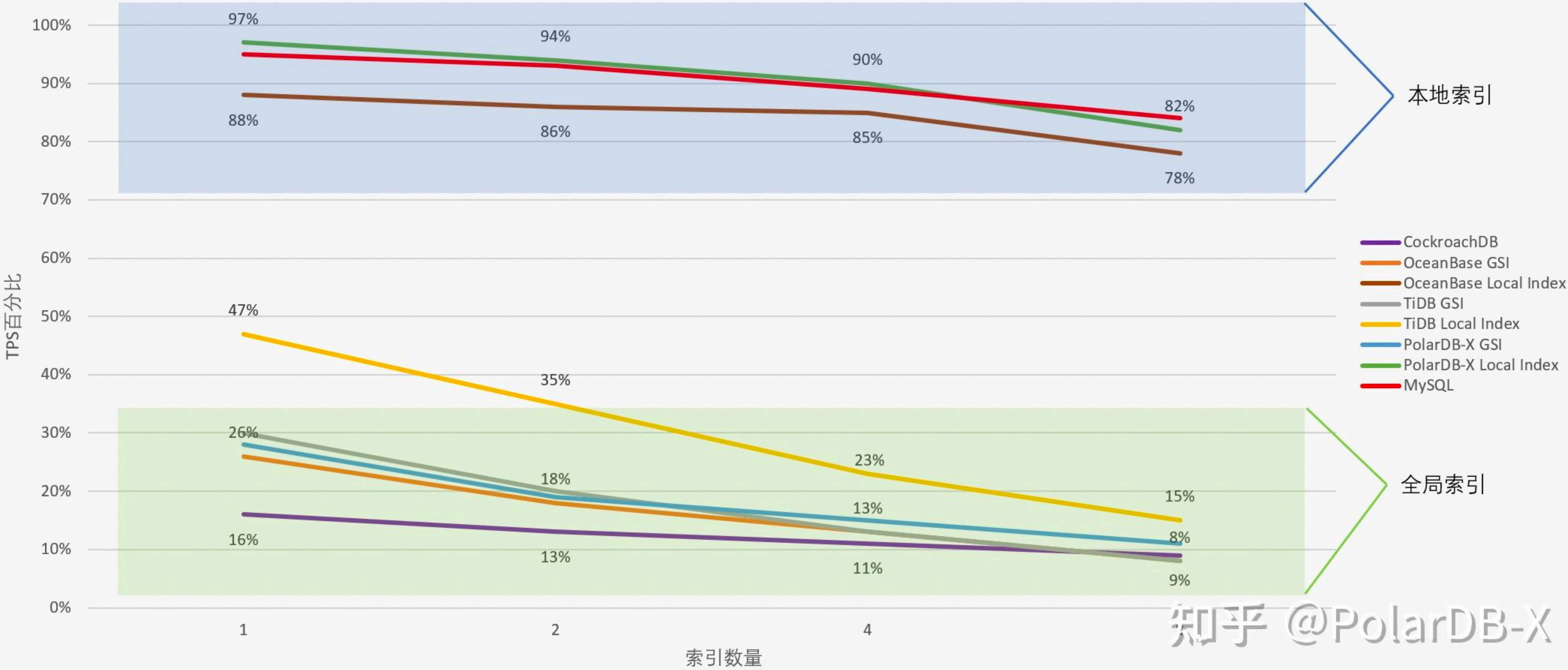
网络吞吐
万兆: 1.2GB/s

慢~1000倍

小~100倍



不同数据库、不同索引类型、数目的TPS百分比



为什么OceanBase性能好

单机低延迟、分布式高吞吐

分片粒度：1个表、或者1个分区

- 单机事务占比，一般可达80%以上
 - 单表（非分区）
 - 单个分区
 - 同一个日志流



为什么OceanBase性能好

Table group优化跨机join

跨机事务可优化成单机事务

```
obclient> CREATE TABLEGROUP myTableGroup1;
Query OK, 0 rows affected

obclient> CREATE TABLE t1 (...) TABLEGROUP = myTableGroup1;
Query OK, 0 rows affected

obclient> CREATE TABLE t5 (...) TABLEGROUP = myTableGroup1;
Query OK, 0 rows affected
```

```
obclient> ALTER TABLEGROUP myTableGroup1 ADD t1;
Query OK, 0 rows affected

obclient> ALTER TABLEGROUP myTableGroup1 ADD t5;
Query OK, 0 rows affected
```



为什么OceanBase性能好

Query优先级区分

- 大货车，靠右行驶
 - 小查询优先，大查询最多占用 30% 工作线程
 - 可通过 `large_query_worker_percentage` 配置
 - 没小查询时，大查询可用到 100% 工作线程



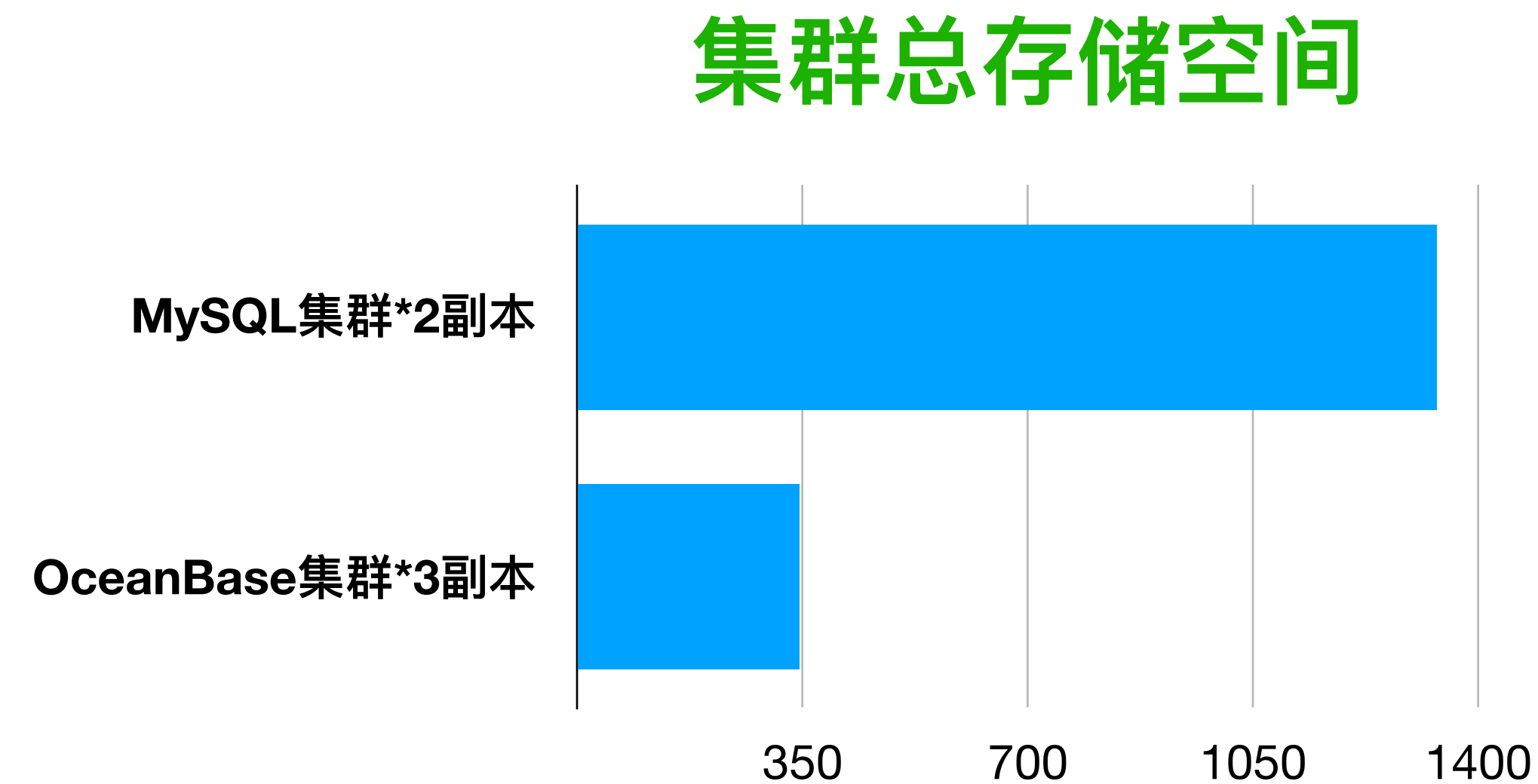
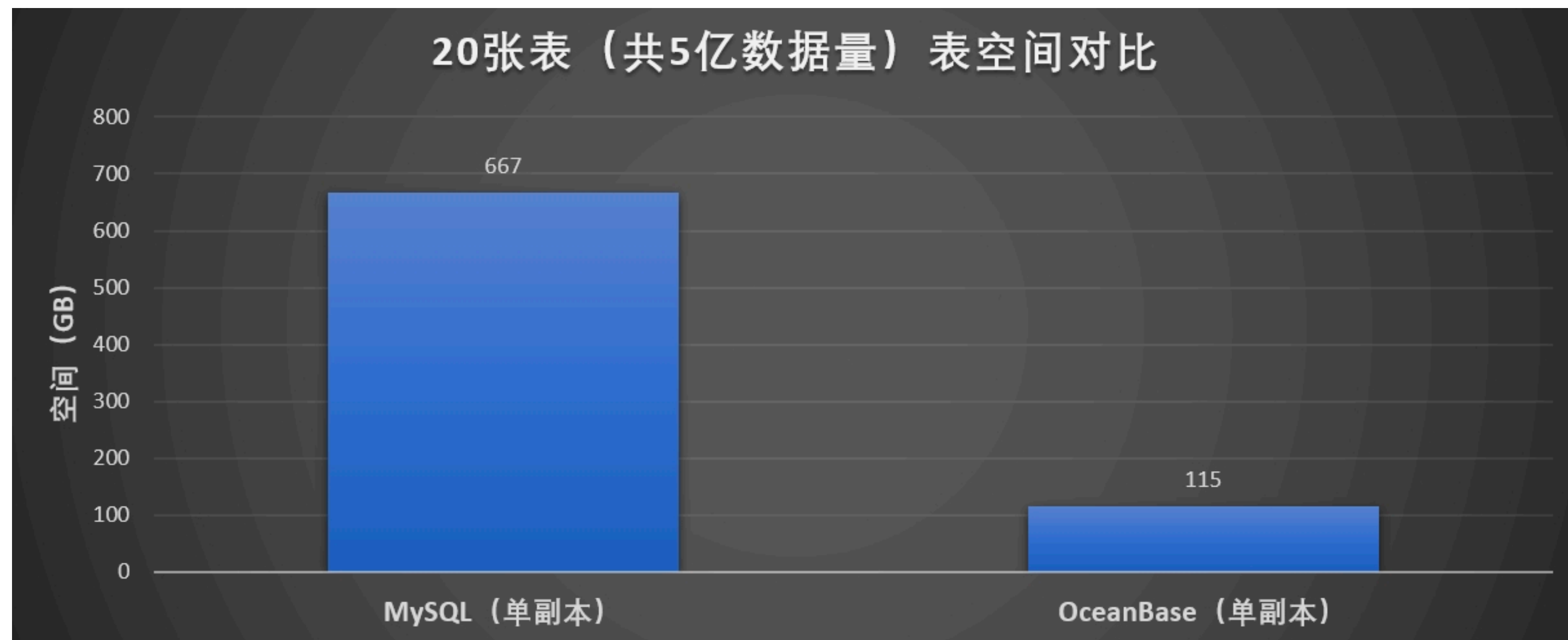
为什么OceanBase成本更低?

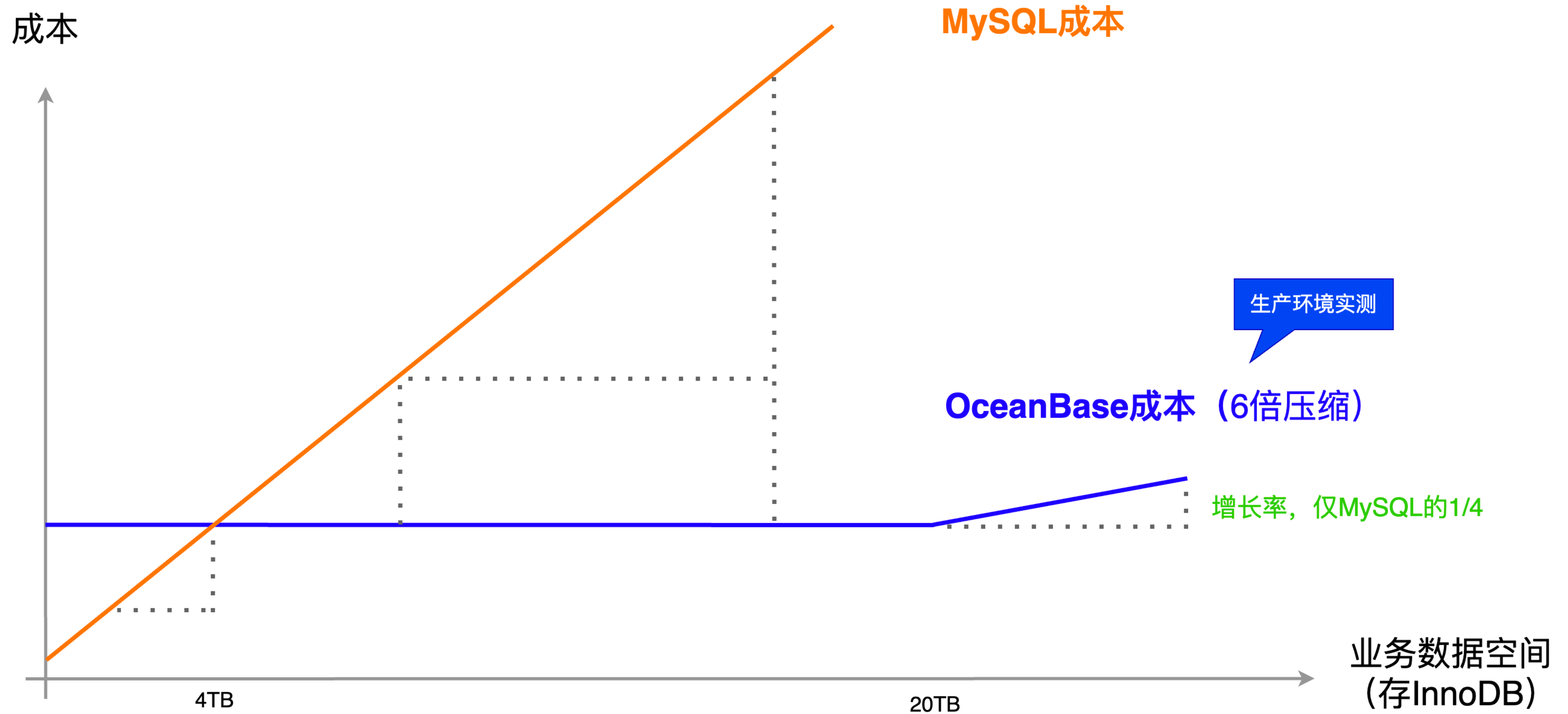
为什么OceanBase成本更低

高压压缩

- LSM-Tree存储引擎，高压压缩

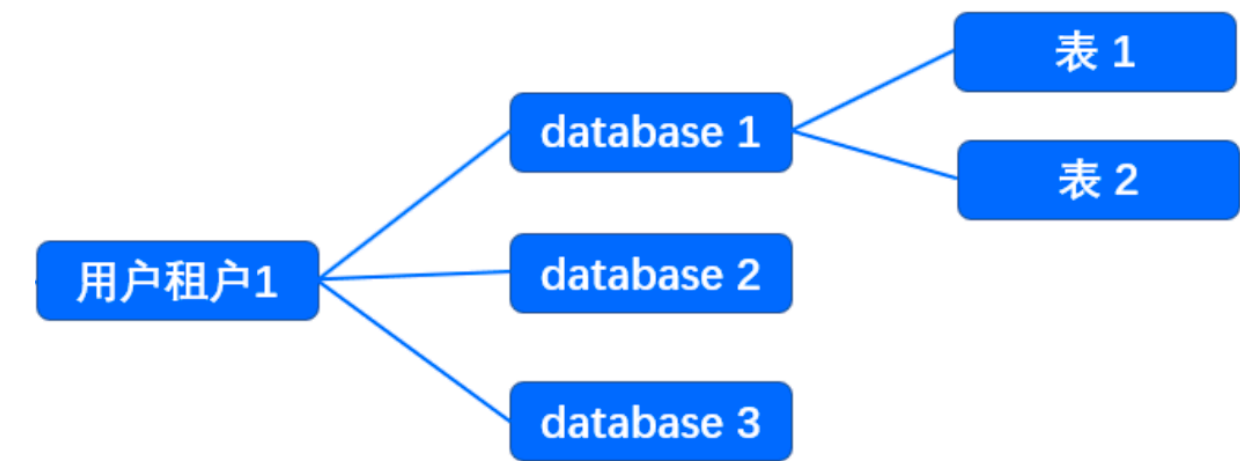
省75%





为什么OceanBase成本更低

多租户契合SaaS场景

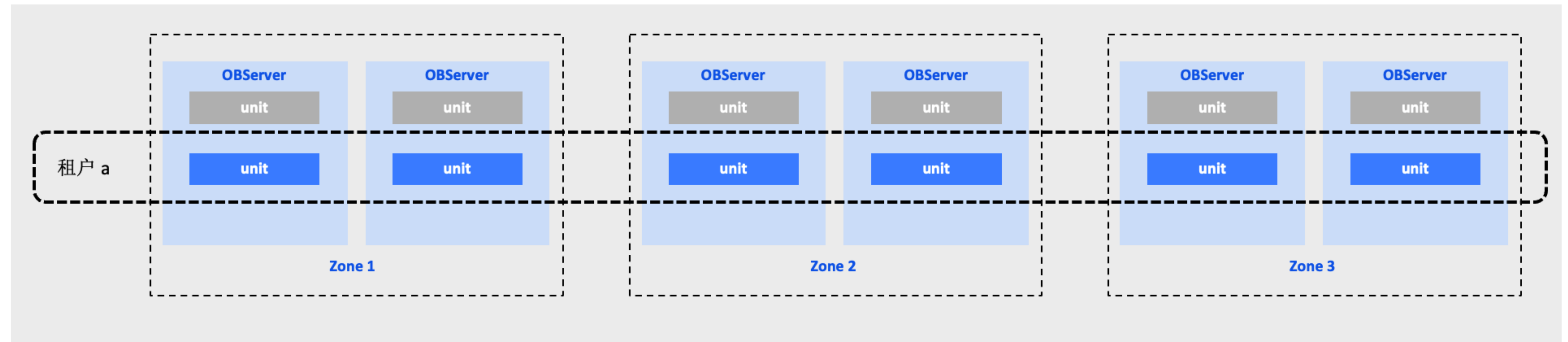


租户资源隔离

按需定义规格

动态扩缩容

自动均衡



水平扩缩容：在2个zone中各增加2个资源单元；减少50%资源

垂直扩展资源：升级资源单元规格，更多CPU、内存、IO

```

obclient> ALTER RESOURCE POOL pool_a
UNIT = unit_a, UNIT_NUM = 2,
ZONE_LIST = ('zone-1', 'zone-2', 'zone-3', 'zone-4', 'zone5');

obclient> ALTER RESOURCE POOL pool_a
UNIT = unit_a, UNIT_NUM = 1, ZONE_LIST = ('zone-1', 'zone-2', 'zone-3');
  
```

按需
弹性
无损

```

obclient> CREATE RESOURCE UNIT new_unit_a
MAX_CPU = 6, MIN_CPU = 6,
MEMORY_SIZE = '12884901888',
MAX_IOPS = 30000, MIN_IOPS = 30000;

obclient> ALTER RESOURCE POOL pool_a unit='new_unit_a';
  
```

总结

- **SaaS场景**：数据库多、数据增长快、资源成本高、运维难度大
- **分布式数据库**：支撑业务增长、提升开发效率、解放DBA
- **OB高性能**：一体化架构，单机低延迟、分布式高吞吐
- **OB低成本**：高压缩能力、更低存储成本
- **OB多租户**：灵活动态扩展能力，契合SaaS场景

Thanks!