

云数据库RDS MySQL性能 深度测评与对比

周振兴@ninedata.cloud

ACMUG 2024-01

目录

关于我

云数据库性能测试总览

ARM-Based的RDS MySQL的性能

各云数据库厂商性能详情

总结

关于 NineData



关于我：周振兴



www.ninedata.cloud

2021~至今

NineData 联合创始人



2009~2021

淘宝/阿里云数据库资深专家
曾用花名“苏普”



2021~至今

Oracle ACE (MySQL方向)



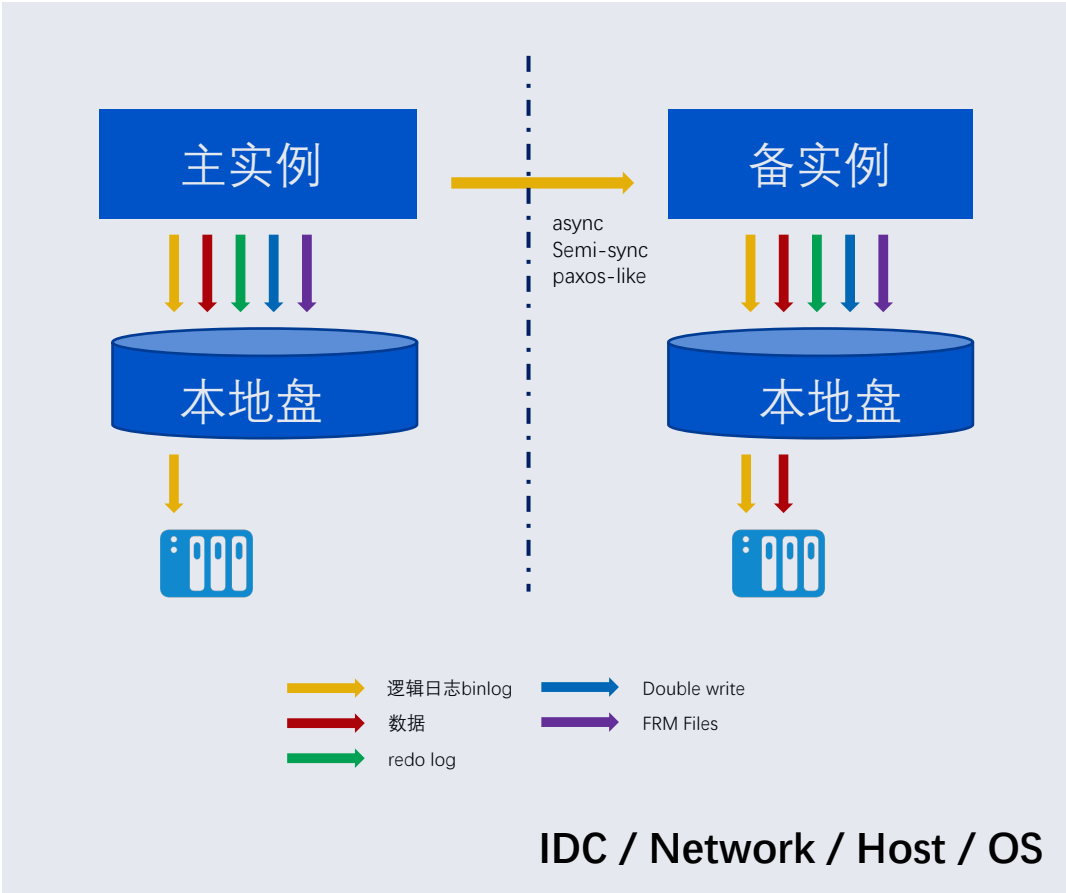
2013、2022

《高性能MySQL》
第三、四版译者

个人博客：<https://www.orczhou.com>

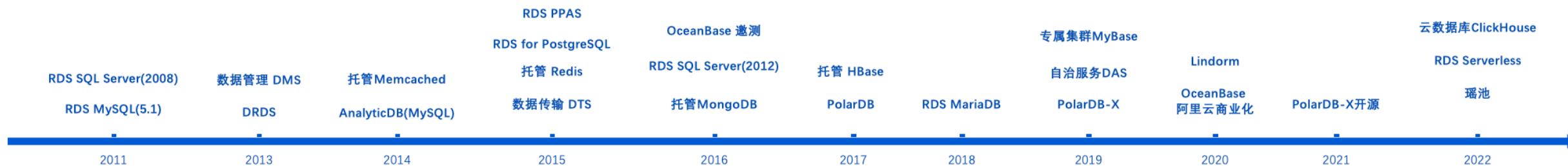
自建数据库架构

典型的自建MySQL数据库架构

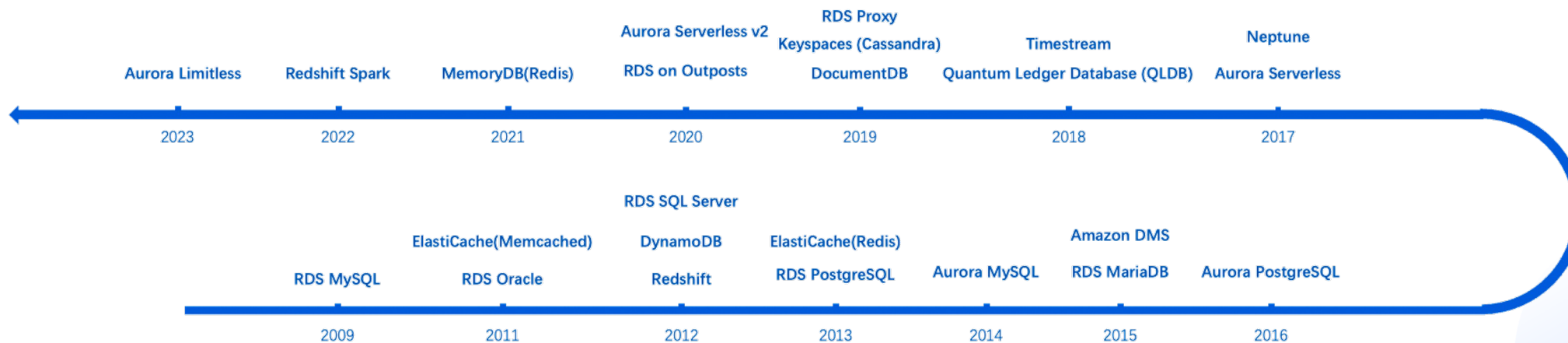


生产变更发布	在线数据访问管理	数据流动	
性能管理	容量管理	故障排查	高可用
数据库生命周期管理	监控告警	备份恢复	
数据库版本与架构			
OS与配置			
IDC与硬件			

云数据库的发展-阿里云/AWS

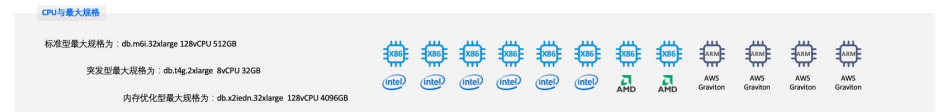


阿里云数据库主要产发布历史



AWS数据库主要产发布历史

Amazon RDS MySQL架构图



存储层：gp2		存储层：gp2		存储层：预留OPS SSD 实例：io1		存储：HD/磁盘	
定位：	测试环境	定位：	性价比高、OLTP	定位：	性能高、稳定性高的生产环境OLTP	定位：	过时的、向前兼容而保留
SLA：	99% 毫秒级	SLA：	99% 毫秒级	SLA：	99.9% 毫秒级	SLA：	/
最大容量：	64TB	最大容量：	64 TB	最大容量：	64 TB	最大容量：	3 TB
最大IOPS：	64000 (根据存储空间换算)	最大IOPS：	64000 (与存储空间分开, 提供基线IOPS)	最大IOPS：	250000 (与存储空间分开)	最大IOPS：	1000
计费：	存储空间	计费：	存储空间 底层基线的IOPS 超出基线的带宽	计费：	存储空间 预留IOPS	计费：	存储空间 IO请求总数



PART 01

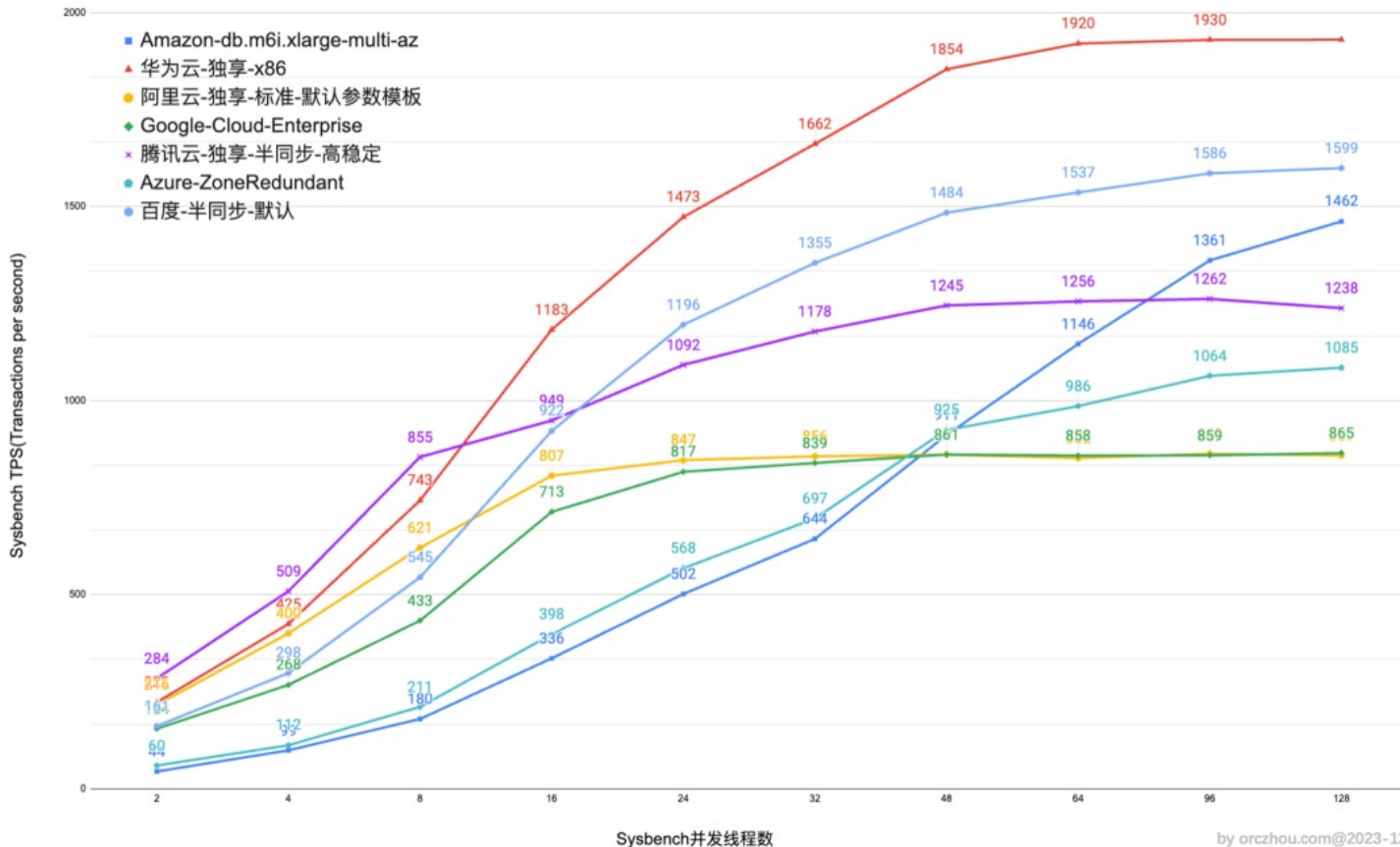
云数据库性能测试总览

通过这个性能测试，尝试回答如下问题

- 不同厂商的**相同**规格（例如 4vCPU16GB），他们的性能相同吗？
- ARM-Based实例总是**宣称**有更高的性价比，真的是这样吗？
- 很多厂商RDS都提供了**不同的存储选择**，例如ESSD PL0/1/2/3，gp2/gp3/io1，他们性能差别大吗？
- “**高性能**”、“**高可靠**”参数模板，他们性能差别大吗？不同的选择我们让渡了什么？
- “**通用**”、“**独享**”规格，他们性能差别大吗？不同的选择我们让渡了什么？
- 部分云厂商提供了“**专有日志盘**”功能，是否值得选择，付出了额外的成本，我们会收获什么？

云数据库性能对比（4vCPU16g“企业级规格”）

云数据库性能对比（4vCPU16g “企业级规格”）



HUAWEI 华为云数据库（红色），在中高并发度时（ ≥ 16 ），性能最强，且高于第二名约18%；

百度云 百度云数据库（淡蓝色），在中高并发时（ > 16 ），性能跃居第二，仅次于华为云；

腾讯云 腾讯云数据库（紫色），在低并发时（ ≤ 8 ），性能最强，高于第二名约15%；

阿里云 **Google Cloud**

阿里云数据库（黄色）和谷歌云数据库（绿色）在低并发时也都有不错的表现，高并发之后性能则持续稳定在850 tps。这两家云数据库的响应延迟，表现得也非常接近。



亚马逊云数据库（Amazon RDS，蓝色）和微软Azure云在中低并发时，表现出了较为相近的性能趋势，且性能较低。但是在高并发时，两者都表现出了非常强的扩展性，AWS RDS在96和128并发时，性能超过阿里云、腾讯云、谷歌云等，跃居第三；同样的，微软Azure云的数据库在高并发时也超过了阿里云、谷歌云，也有不错的性能表现。



限制与说明

- 本次测试选择了 sysbench 的 oltp_read_write 模型，这是一个标准、简洁的数据库测试，但并不能反应所有复杂的生产环境的实际性能情况
- 本次测试，地域选择较为随机，例如阿里云选择了杭州、百度云选择北京、AWS/谷歌云选择了东京、Azure云选择了美东等。这里的一个假设是，各个区域的性能应该是相同或者接近的。但是，似乎这个假设并不总是成立
- 受限于时间与资源，这里仅对4vCPU16GB的规格进行测试，单次并发测试持续时间为300秒
- 虽然都是用MySQL 8.0版本，但是不同的厂商的数据库小版本也会不同。不同厂商做得改造也不同
- 不同厂商的CPU、磁盘类型、价格等各有不同，所以这不是一个完全对等的测试
- 不同厂商的RDS实例的默认参数模板也各有不同，甚至同一个厂商在不同阶段的参数也可能会有调整
- 有的云厂商的磁盘存储有着多种不同的选择，例如阿里云支持ESSD PL1/2/3，AWS也支持gp3/io1类型的存储选择，不同的存储选择也会对应这个不同的性能
- 关于可用区的选择：在测试中，我尽量会将测试的ECS/EC2/VM等与数据库主节点放在同一个可用区。但依旧有一些情况做不到这一点。例如在这次的测试过程中，GCP在东京地区b可用区，虽然可以创建数据库，但是却没有资源创建VM节点（“A n2-highcpu-8 VM instance is currently unavailable in the asia-northeast1-b zone.”）
- 由于Oracle Cloud国内无法注册，故没有测试；火山云由于不支持个人账号购买使用，故未测试

“企业级”规格的选取

为了方便对比，对横向对比的规格做了统一。在众多配置选项中，“企业级规格”需满足如下条件：

- **具备敏捷的高可用能力。**需要是双节点或更多，在其中一个节点出现故障时，能够快速地进行切换（Switchover或failover）；
- **需要具备非常高的数据可靠性保障。**至少需要跨可用区（同城）的容灾，且数据在不同可用区同步持久化。即需要使用同步或“接近同步”（如半同步）的复制；
- **具备较好的性能一致性。**核心数据库对于稳定性有着更高的要求，需要使用独享的计算、存储、内存等资源。不允许因为资源争抢导致的，故数据库的性能表现需要较为一致，不能有太大的波动，或者未知的不可预测的资源争抢。
- **关于规格的大小，**这里选择中等规格4vCPU16GB内存/100GB存储/3000 IOPS。是企业业务的起步规格；另一个主要考量是，降低测试的执行成本。

测试方法说明

目标：易于实现，易于重现；有较强的参考价值

测试工具：sysbench

测试模型：oltp_read_write

测试时长：300秒

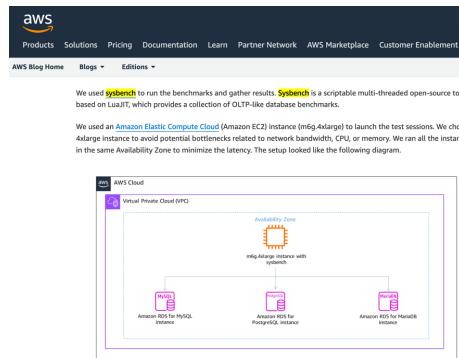
单表大小：1000000

总表数量：10

测试并发：2、4、8、16、24、32、48、64、96、128 线程

Sysbench被业界广泛使用：

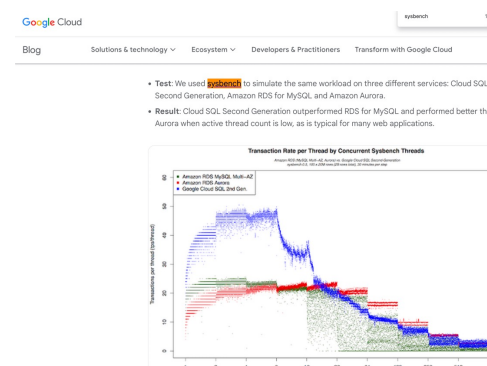
```
sysbench oltp_read_write --threads=$conthread --time=$run_time \
--report-interval=3 --percentile=95 --histogram=on \
--table_size=$table_size --tables=$tables run
```



[AWS官方博客：参考](#)



[腾讯云性能白皮书：参考](#)

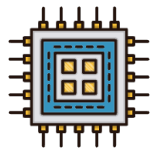


[GCP官方博客：参考](#)



[阿里云官方文档：参考](#)

整体性能结果分析 (1)



CPU性能（代际）是本次测试中性能表现的主导因素，其次是存储的IO能力



非常明显的，有更强的、更多的CPU资源，这在独享规格中并不常见。



单核的计算性能确实更强，在CPU代际上有着非常明显的优势



(测试区域: Tokyo)

访问延迟较大，但是，在高并发的时候，才能够充分发挥吞吐量的优势



访问延迟较大，但是，在高并发的时候，能够充分发挥吞吐量的优势



访问延迟非常低，所以在低并发时优势明显。但是高并发由于处理能力受限，较为明显的是CPU代差所导致的。

整体性能结果分析 (2)



存储架构和“持久性”考量对于性能影响明显，尤其是低并发的时候



Google Cloud

云盘 / 本地SSD

均为“半同步”复制

拥有更低的延迟

低并发有更高的性能表现

性能 vs 持久性 / Semi sync



使用“云盘”方案

均为“云盘”同步复制

延迟较大

但整体上，吞吐量都有很好的扩展性

性能 vs 持久性存储同步复制



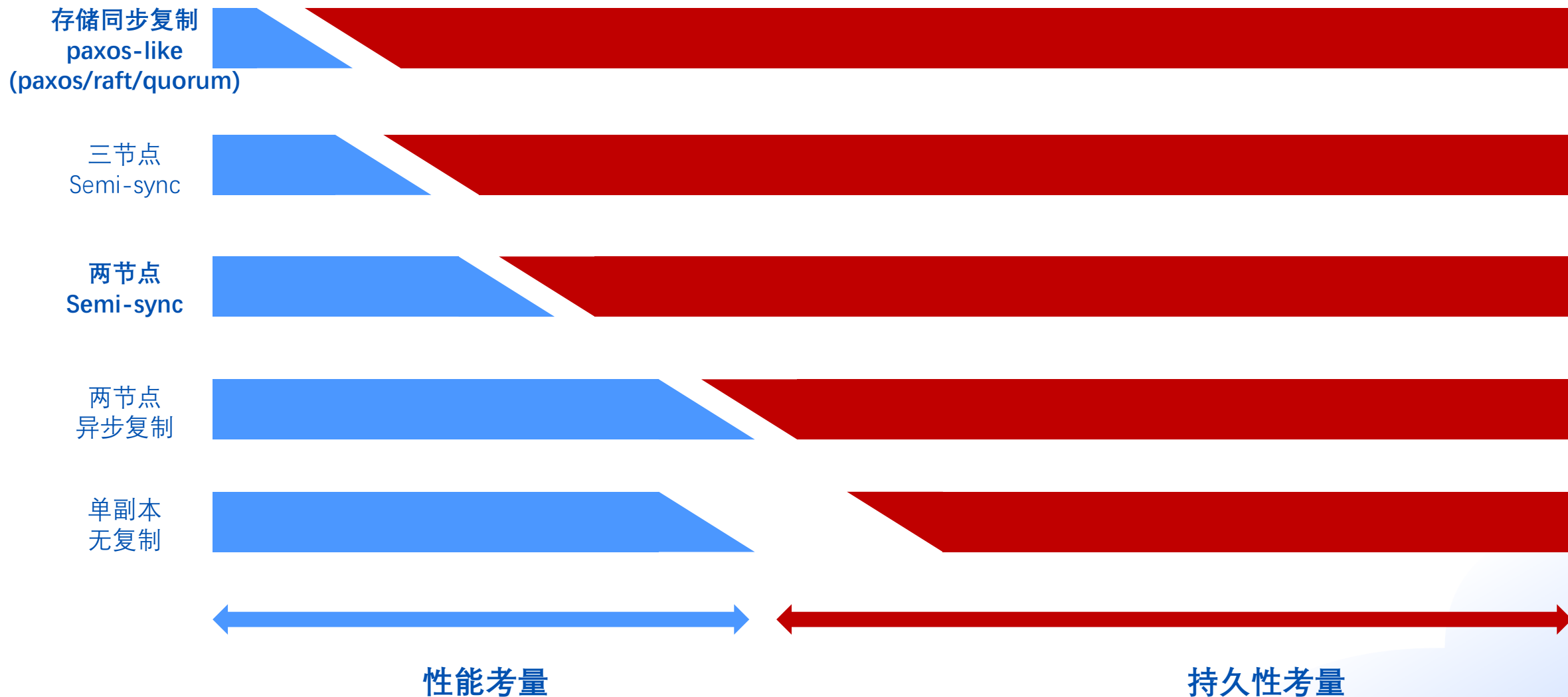
整体性能结果分析（2） - Availability && Durability

- Availability: 表示服务是否可用。出现故障，服务是否可用，或者多长时间可以恢复
- Durability: 表示数据是否持久化。出现故障，数据/事务是否可以恢复。
 - 通常翻译作/用作：“持久性”。也经常被 翻译作/用作“可靠性”，这可能是一个有一定歧义，但是也被用得比较多的说法。

其他：

Reliability: 狭义的，表示服务“一致性”。可能包括性能（延迟、吞吐量等）的一致性

RDS MySQL持久性(durability)分类



整体性能结果分析 (3)



在这个测试中，低并发的性能看延迟；高并发的性能看处理能力与吞吐量

单位时长 = $\text{TPS} \times \text{average latency}$

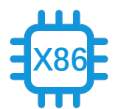
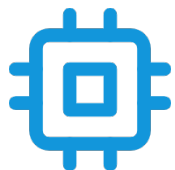
平均延迟和95%延迟都有较强的参考价值，本次测试后续也会公布这些数据

国内云厂商可用区之间延迟更低，整体上在低并发的表现更好。AWS则总是在教育用户，要选择跨可用区实例，要容忍可用区之间的延迟。

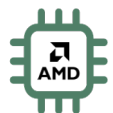
PART 02

ARM vs x86 如何影响云数据库性能

云数据库的主要选项与测试



Graviton



倚天芯片



鲲鹏



本地SSD

云盘

ESSD PL0/1/2/3

gp3 / io1 / io2



高性能

高可靠/持久性



通用

独享

ARM vs x86

arm

Advanced RISC Machines
Acorn RISC Machine

low costs

low power consumption

low heat generation

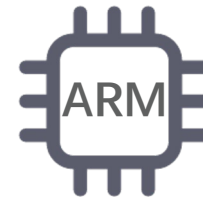
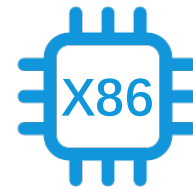
1985年 ARM1: first samples of ARM

1986年 ARM2

1995年 StrongARM / Xscale

2011年 ARMv8 新增64位支持

2016年 日本软银集团约合311亿美元收购ARM



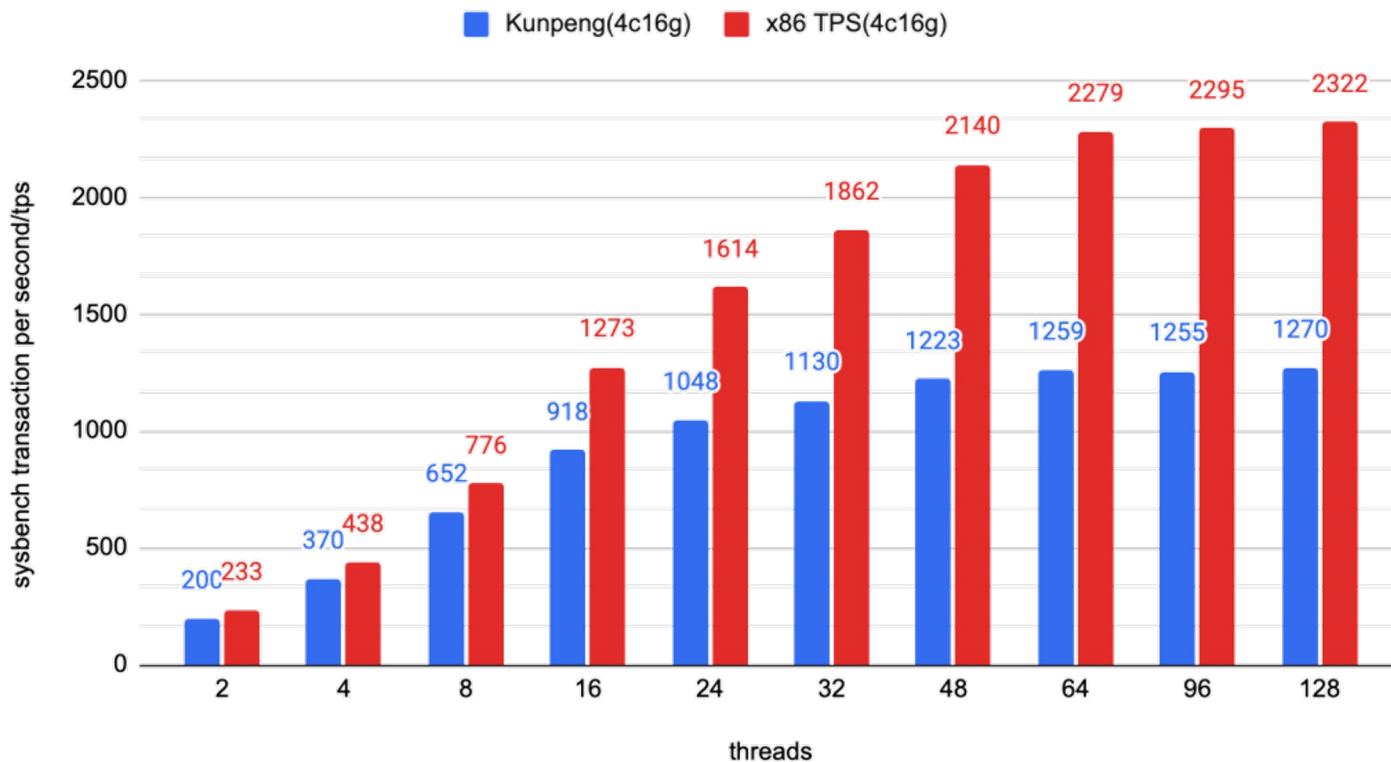
参考:

- [ARM@wikipedia](#)
- [Where to Encode: A Performance Analysis of x86 and Arm-based Amazon EC2 Instances](#)
- [On Power Efficiency: x86 vs. ARM](#)

ARM vs x86 (intel)



华为云RDS 鲲鹏 vs x86规格性能对比 TPS



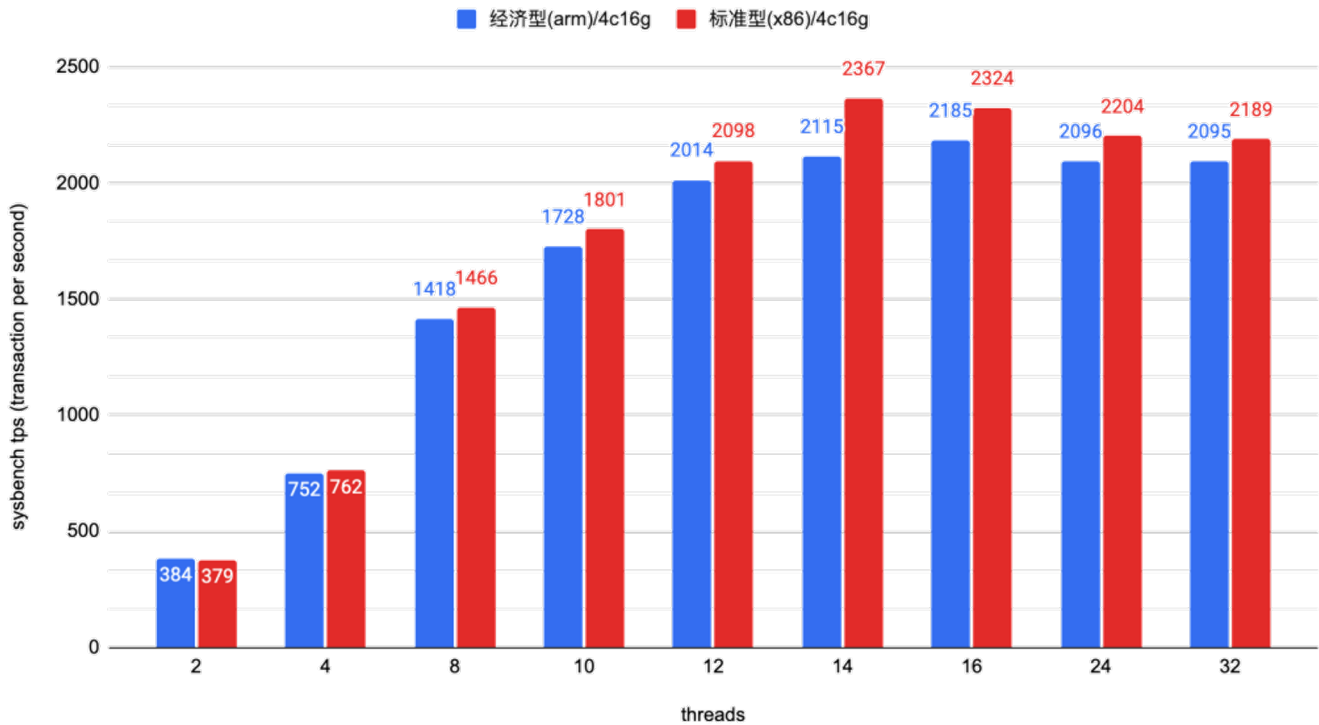
在华为云上

- x86和鲲鹏架构实例价格是相同的
- 鲲鹏版本相比x86约有15~45%的性能差距
- 考虑到自研鲲鹏芯片在中国自主可控芯片中的地位，在国内大量无法使用x86的场景中，这个性能下降通常都是可以接受

ARM vs x86 (intel)



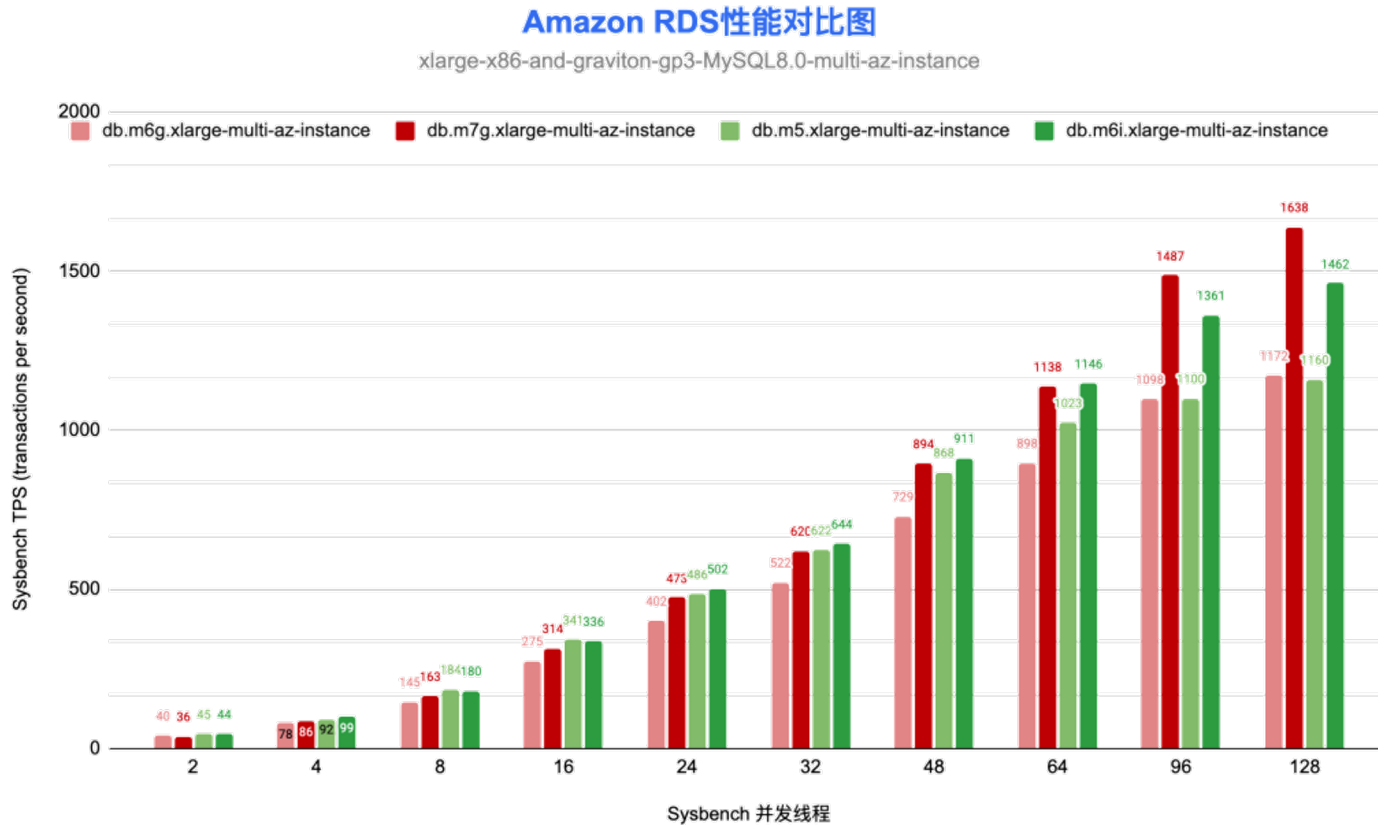
阿里云RDS经济型(ARM) vs 标准型(x86) 性能对比 / TPS



在阿里云上

- 经济版（ARM）比标准版（x86）性价比要高出**32%**
- 具体的：选取16并发，ARM版TPS为2185，x86版TPS为2324。价格上，ARM版价格为1.61元/时，x86版价格为2.52元/时，那么对应每1000个TPS的价格分别为：0.74元与1.08元。从性价比的角度来看，经济版提升了31.5%

ARM vs x86 (intel)



Graviton (ARM)

m7g实例 (Graviton 3)

m6g实例 (Graviton 2)

Intel (x86)

m6i实例 (第三代Intel Xeon/Ice Lake)

m5实例 (Intel Xeon)

在AWS上

- Graviton 2实例相比x86几乎没有什么优势，与宣称的52%性价比提升结果相悖
- Graviton 3实例性能相比Graviton有非常大提升
- Graviton 3实例性能在高并发时，相对x86有较明显的性价比优势，以128并发为例，m7g vs m6i高出10%
- Graviton 3 相比Graviton 2有非常明显的性能优势，这与宣称的27%性价比提升是较为一致的

结论： ARM vs x86 (intel)

在华为云上

- x86和鲲鹏架构实例价格是相同的
- 鲲鹏版本相比x86约有15~45%的性能差距
- 考虑到自研鲲鹏芯片在中国自主可控芯片中的地位，在国内大量无法使用x86的场景中，这个性能下降通常都是可以接受

在阿里云上

- 经济版（ARM）比标准版（x86）性价比要高出**32%**
- 具体的：选取16并发，ARM版TPS为2185，x86版TPS为2324。价格上，ARM版价格为1.61元/时，x86版价格为2.52元/时，那么对应每1000个TPS的价格分别为：0.74元与1.08元。从性价比的角度来看，经济版提升了31.5%

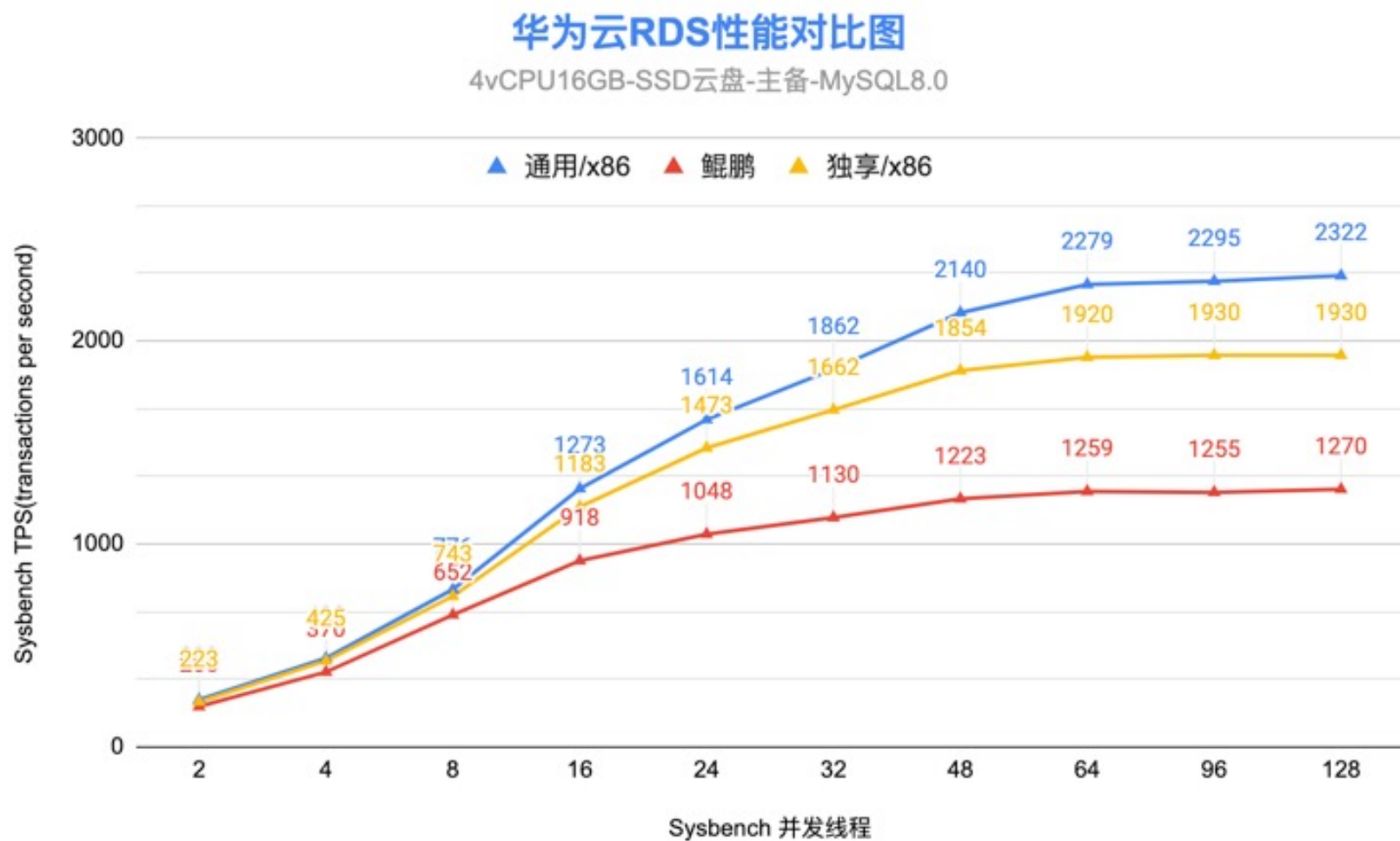
在AWS上

- Graviton 2实例相比x86几乎没有什么优势，与宣称的52%性价比提升结果相悖
- Graviton 3实例性能相比Graviton有非常大提升
- **Graviton 3实例性能在高并发时**，相对x86有较明显的性价比优势，以128并发为例，m7g vs m6i
- **Graviton 3 相比Graviton 2**有非常明显的性能优势，这与宣称的27%性价比提升是较为一致的

PART 03

各云数据库性能详情

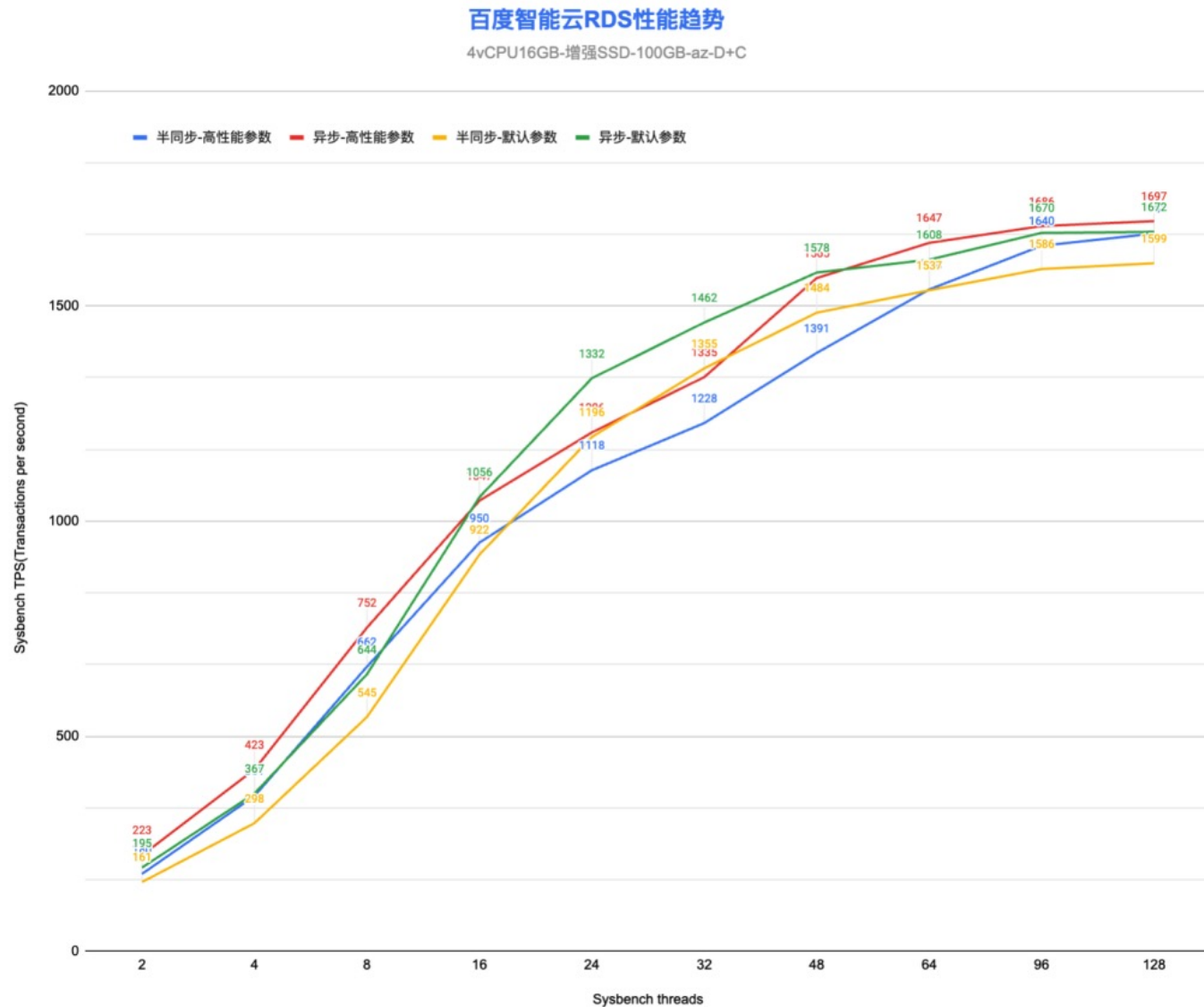
华为云RDS MySQL性能详情



相比于独享规格，华为云通用规格，在高并发时，用户又可以获得约15%~20%的性能提升，代价是牺牲一定的性能的一致性

鲲鹏增强型，相比Intel的x86有着约20%~30%的性能下降。考虑到鲲鹏芯片在国产芯片中的地位，在国产化场景中，采用该规格是非常不错的选择。

百度云数据库（RDS）性能



选项:

- “异步/半同步复制”
- “高性能/默认参数模板”

整体上，百度云RDS性能是非常高的

百度云RDS的选项，并没有表现出预期的差异

高性能:

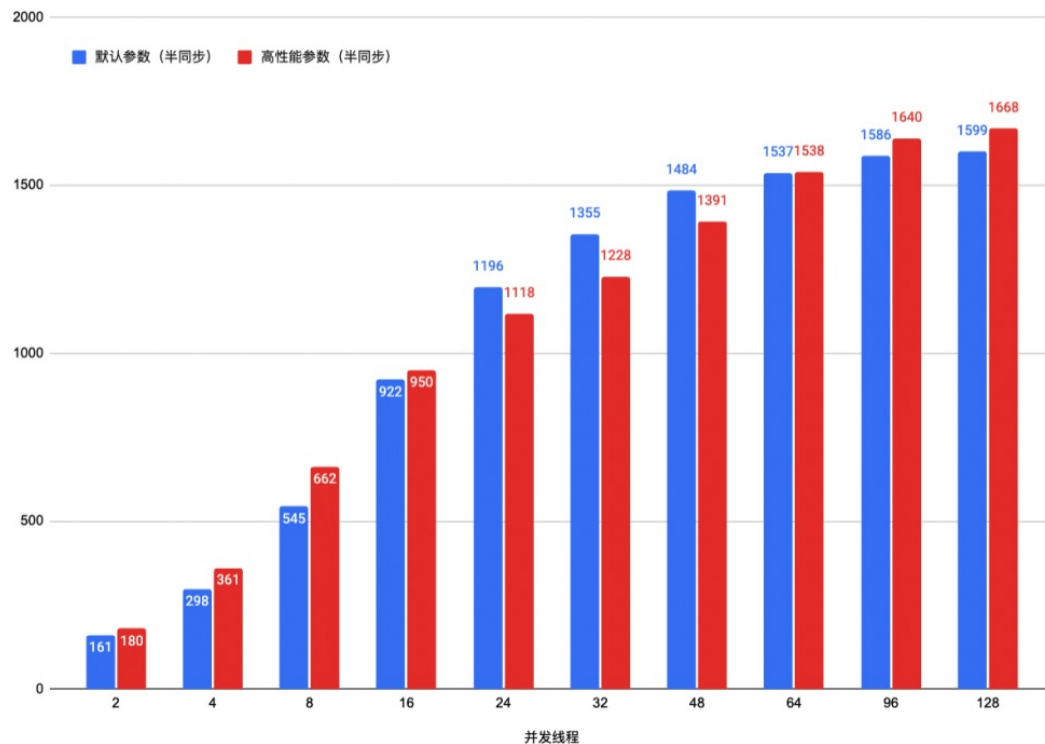
`sync_binlog=1000、`
`innodb_flush_log_at_trx_commit=2`

默认参数模板:

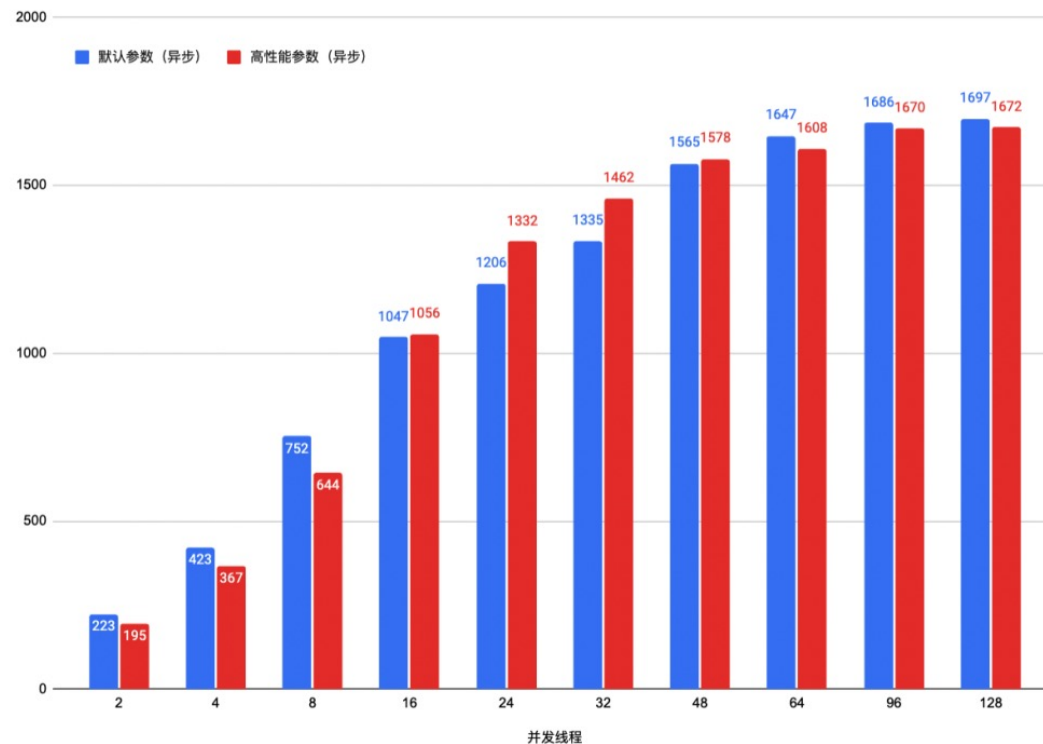
`sync_binlog=1000、`
`innodb_flush_log_at_trx_commit=1`

百度云数据库 (RDS) 性能

默认参数模板 vs 高性能参数模板 (半同步)

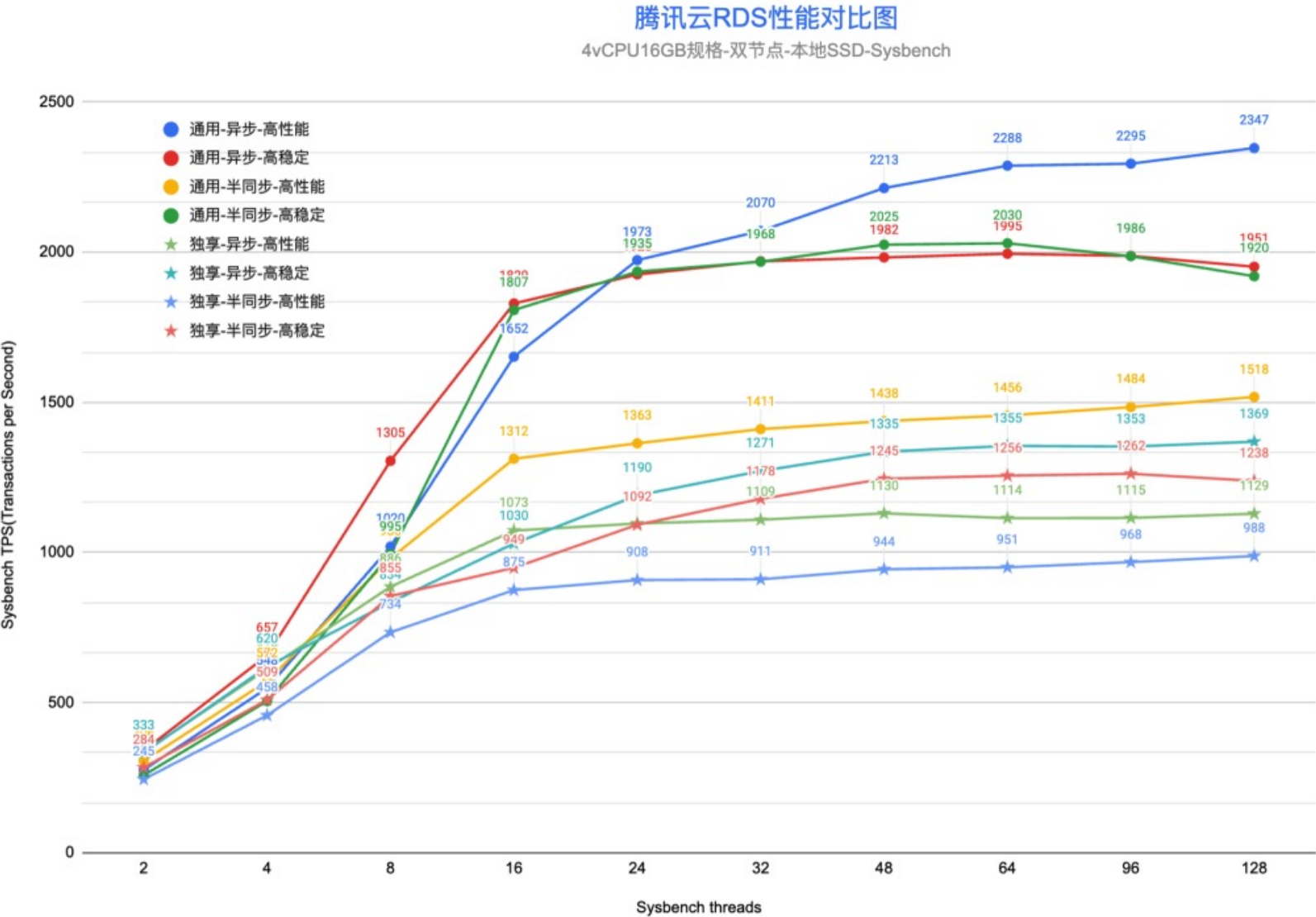


默认参数模板 vs 高性能参数模板 (异步)



无论是在半同步，还是异步同步下，高性能参数模板，并没有总是表现出更强的性能

腾讯云RDS性能详情

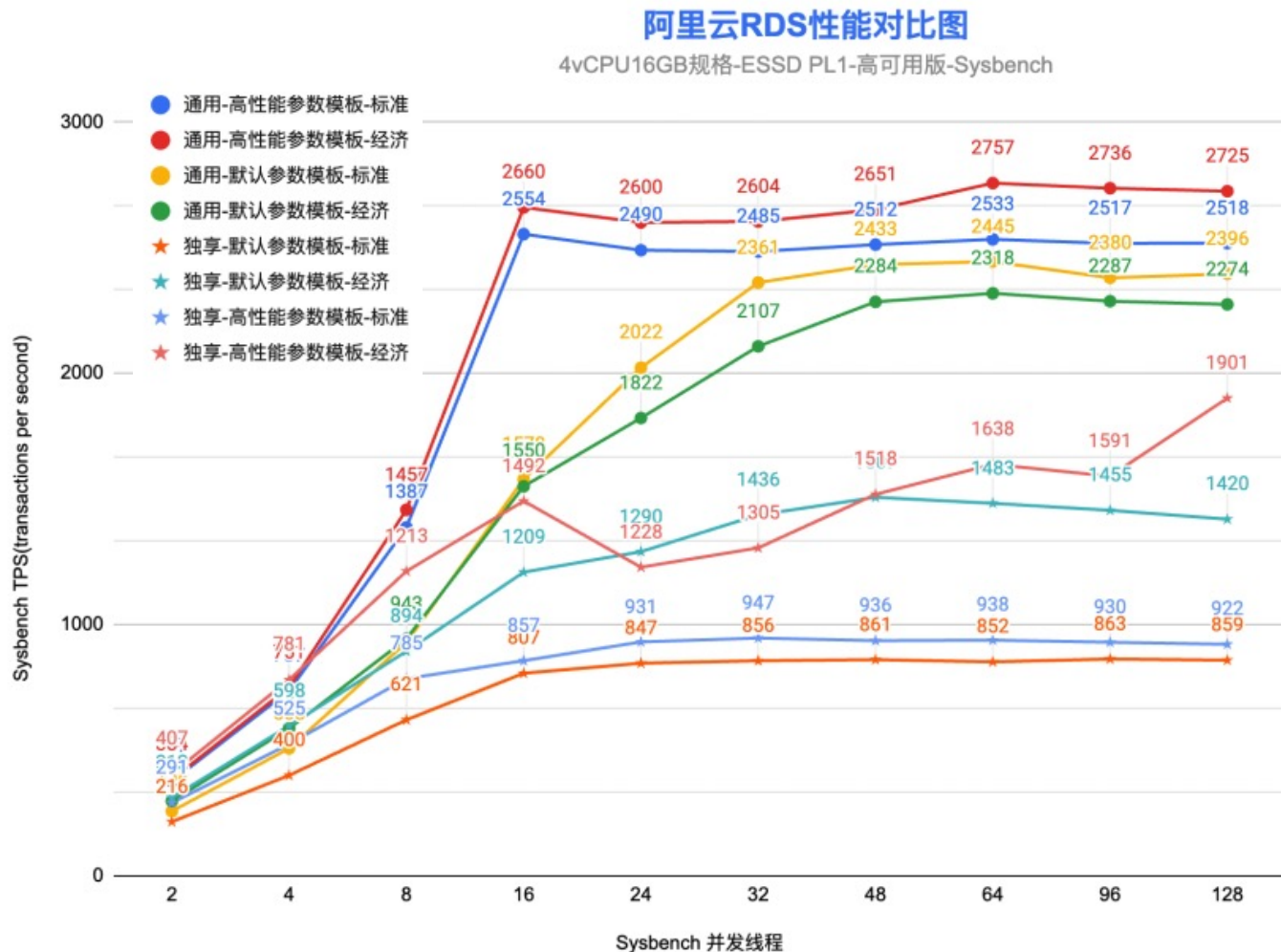


相比于独享规格，腾讯云通用规格，普遍的表现出了更好的性能。在较高的并发压力之下，甚至高出约100~200%。

在独享实例的测试中，也有一些不一致的地方，性能（高稳定）> 性能（高性能）：即

- “独享-异步-高稳定” > “独享-异步-高性能”
- “独享-半同步-高稳定” > “独享-半同步-高性能”

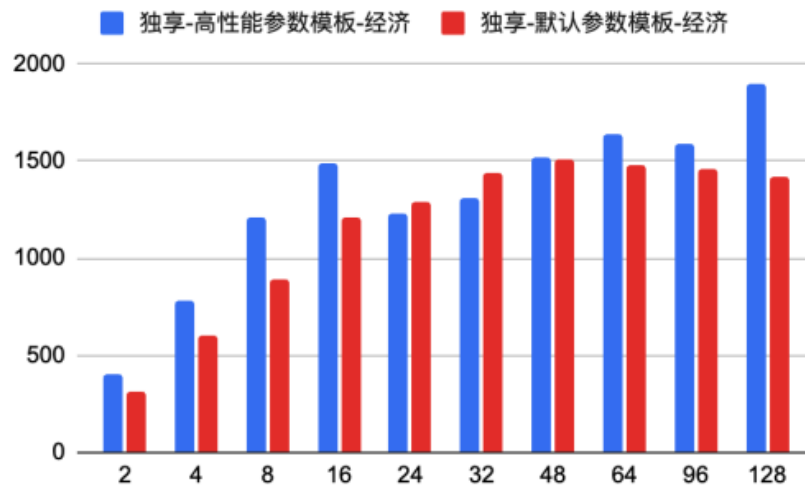
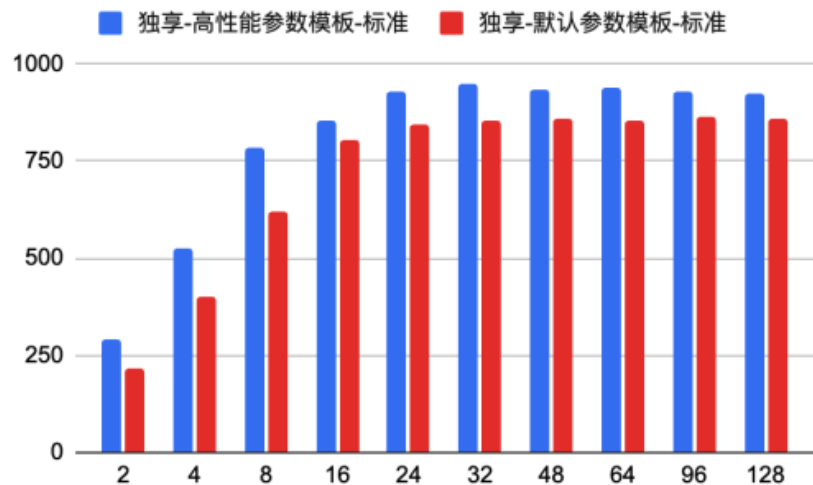
阿里云RDS性能详情



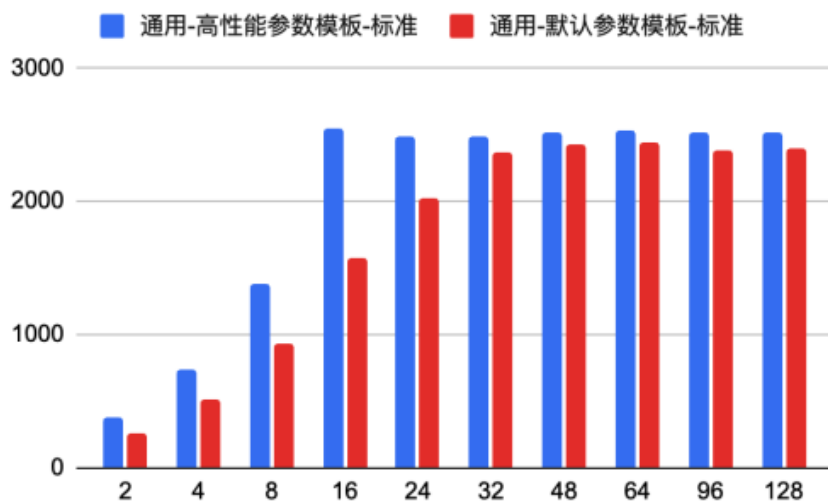
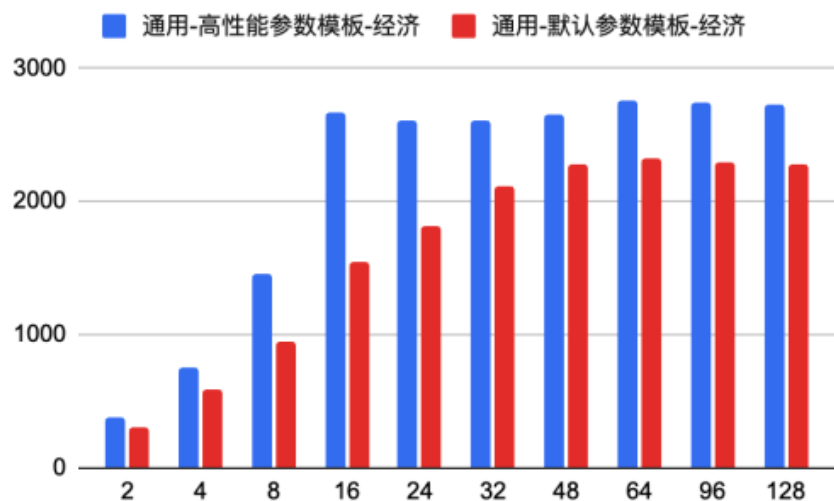
相比于独享规格，阿里云通用规格，普遍的表现出了更好的性能。在较高的并发压力之下，高出约50%~150%。

参数模板“高性能”、“默认参数模板”的对比中，也符合预期，高性能参数模板总是表现出更好的性能。

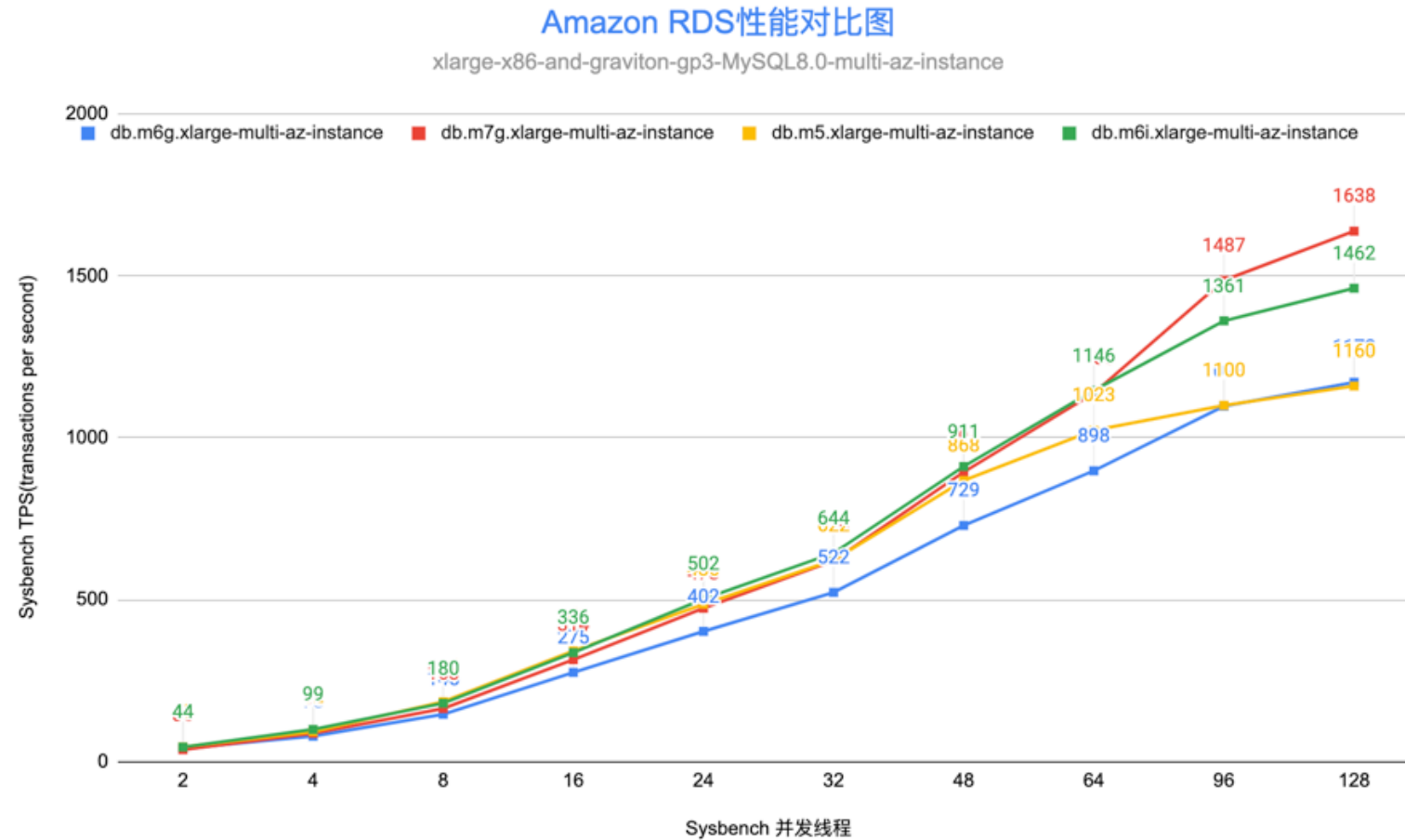
阿里云RDS性能详情 (2)



在“独享-标准”、“独享-经济”、“通用-标准”、“通用-经济”四种组合下，参数模板“高性能”、“默认参数模板”的对比中，如预期，高性能参数模板总是表现出更好的性能。



Amazon RDS性能详情



AWS提供了非常清晰的CPU型号与架构选项

不同的规格与型号，表现出了非常一致的性能：

- Intel(x86) 性能总是高于 Graviton(ARM)
- 新一代总是高于旧一代

在本次测试中，Graviton并没有表现出优于x86的性价比，这与AWS的宣传是不符的。

选项参考：

region & zone: ap-northeast-1c / ap-northeast-1a

Availability and durability: Multi-AZ DB instance

Instance configuration: db.m6i.xlarge

(supports optimized writes)

Storage type: io1 (Provisioned IOPS (io1))

Allocated storage 100gb

Provisioned IOPS: 3000

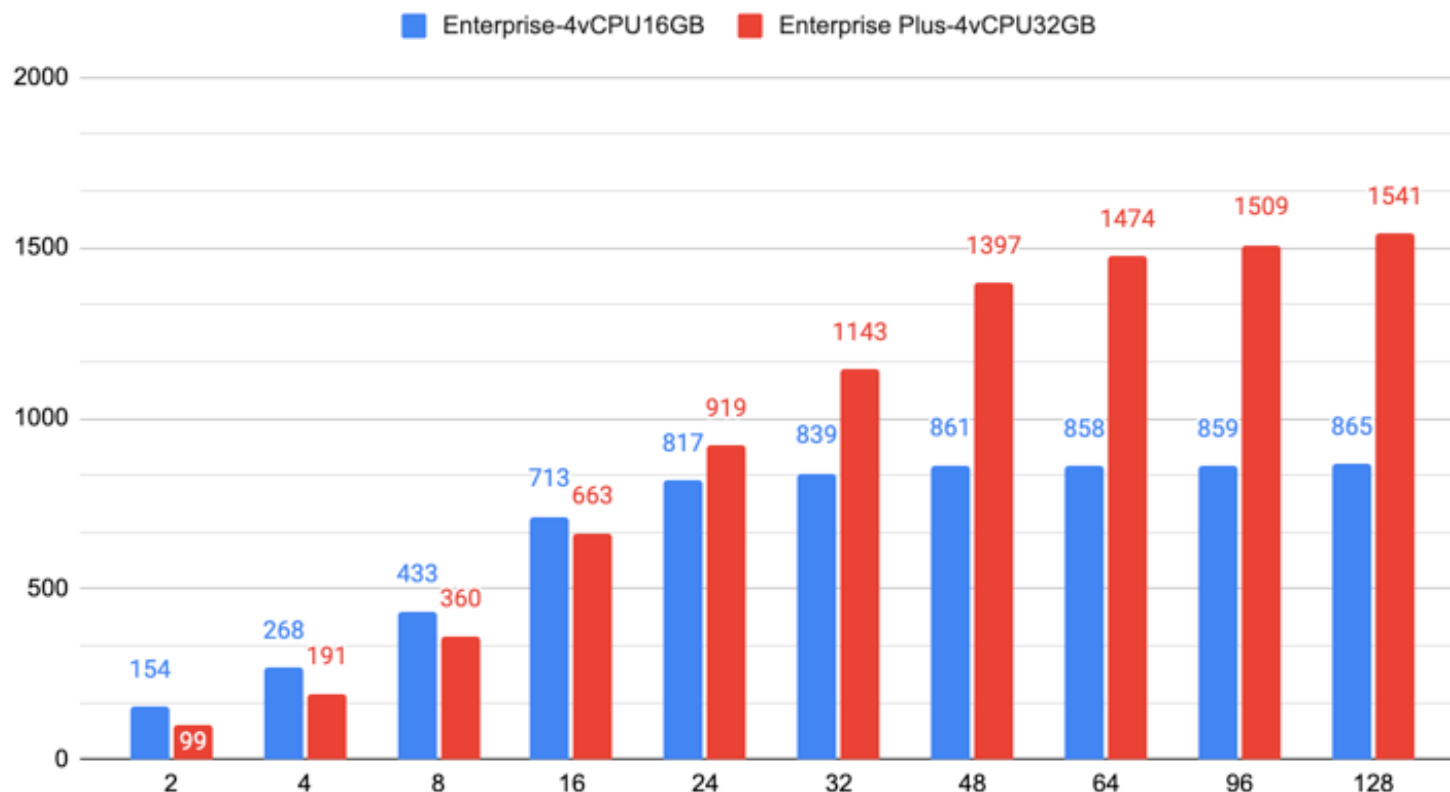
Dedicated Log Volume on/off

Performance Insights : on (default)

innodb_flush_log_at_trx_commit=1 sync_binlog=1

Google云Cloud SQL(RDS)的性能

谷歌云 (GCP) “RDS” (Cloud SQL) 性能趋势



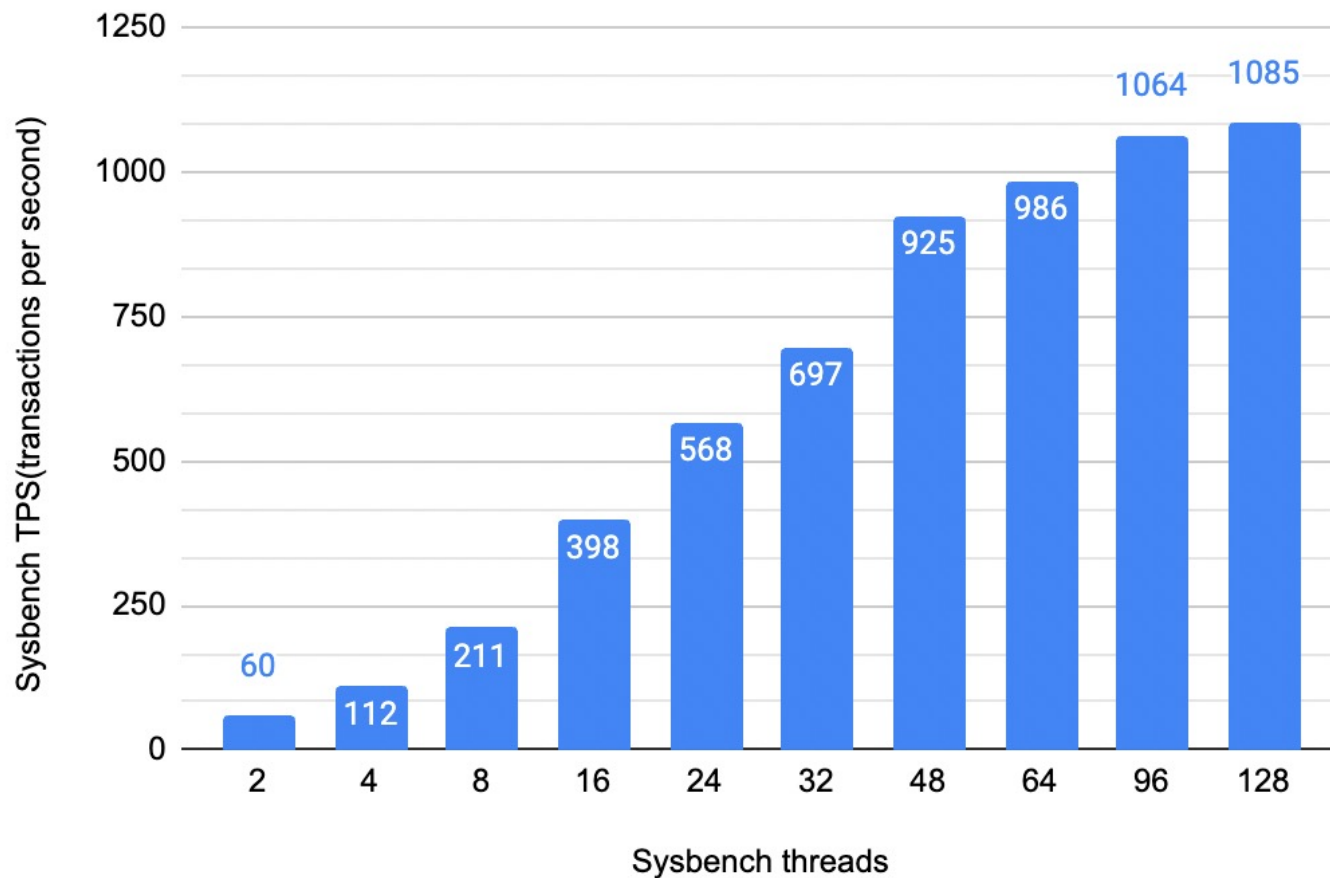
选项较为简单，但是产品命名却把简单的选项变复杂了

- “Enterprise”和“Enterprise Plus”版本
- “Plus”版提供了额外的“Data Cache能力”，
价格高30%
- “plus”版，在本次测试中，在高并发时，
表现出了非常好的性能，高出约75%以上

Azure Database for MySQL (RDS) 的性能

Azure Database for MySQL (FS) 性能趋势

GP,4vCPU16GB,3000 IOPS,Zone-Redundant



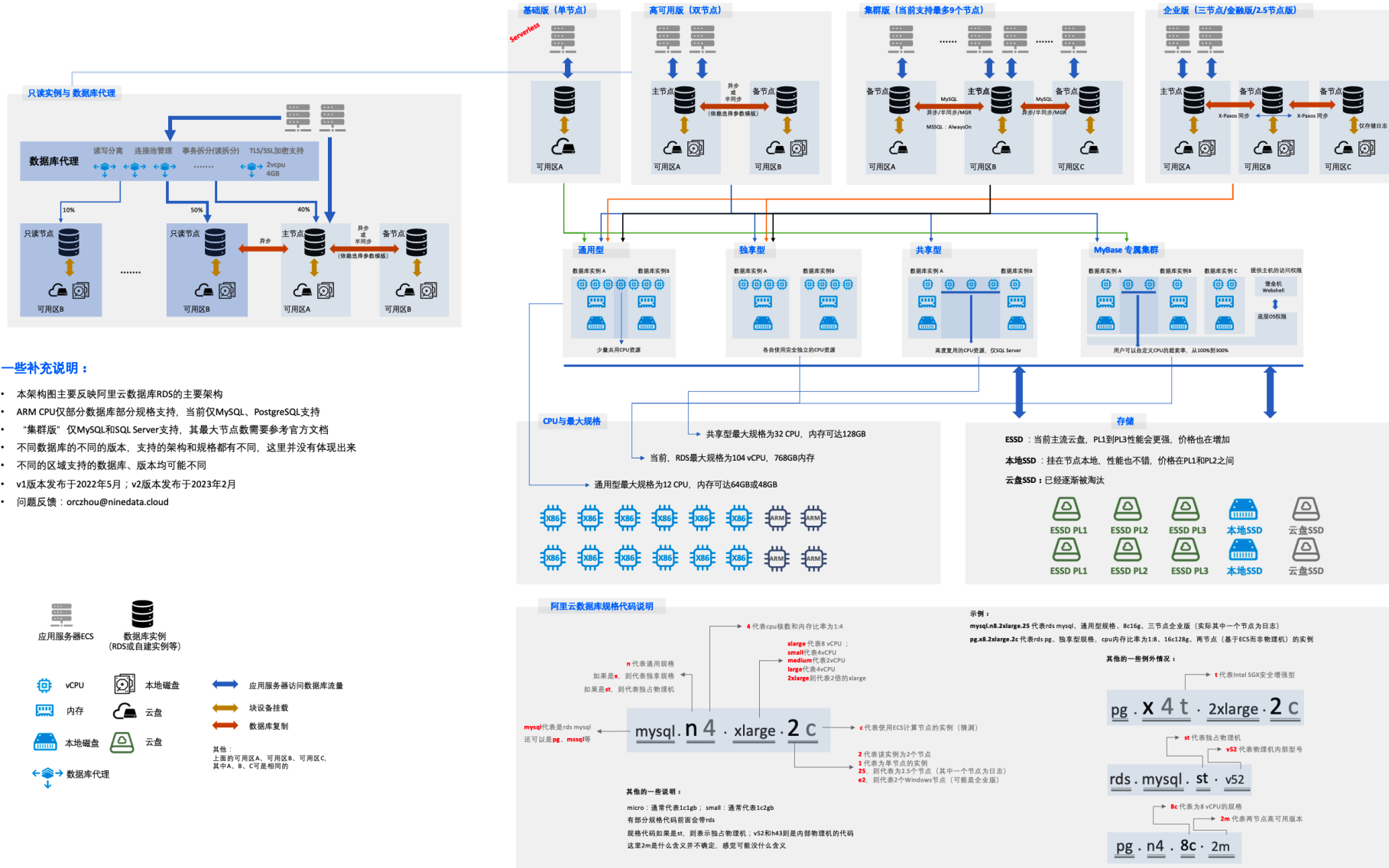
选项较为简单, "Flexible Server" 这个名字大概是让很多人一头雾水的

Azure是少有提供AMD CPU的托管RDS平台, 不过本次测试中, 并没有进行测试

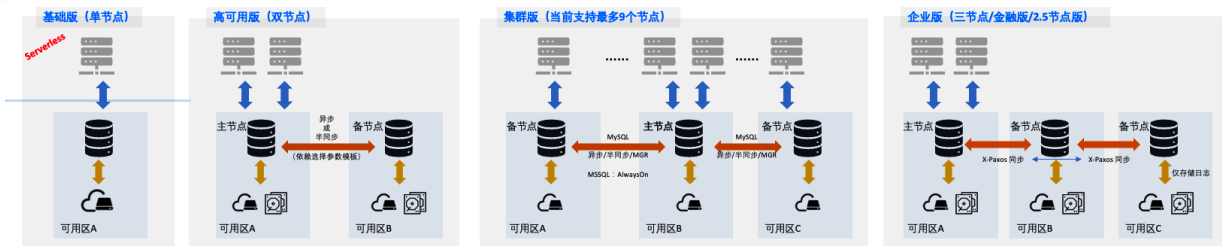
PART 04 (附)

云数据库架构概述 (阿里云、AWS)

阿里云RDS数据库架构



阿里云RDS高可用架构



单节点基础版

定位： 小规模的测试环境

数据可靠性： ★★★

可用性： ★★★

性能： ★★ (规格小)

价格： 低

两节点高可用版本

定位： 生产环境使用

数据可靠性： ★★★★★

可用性： ★★★★★

性能： ★★★★★★

价格： 中

集群版

定位： 生产环境使用

数据可靠性： ★★★★★★

可用性： ★★★★★★

性能： ★★★★★★

价格： 略高

经济版 (vs 标准版)

定位： 生产环境小范围

数据可靠性： ★★★★★★

可用性： ★★★★★★

性能： ★★★★★★

价格： 较低

阿里云数据库规格代码说明

CPU与最大规格

- 通用型：数据实例A, 数据实例B, 少量共用CPU资源
- 独享型：数据实例A, 数据实例B, 各自使用完全独立的CPU资源
- 共享型：数据实例A, 数据实例B, 高度复用的CPU资源, 仅SQL Server
- MyBase 专属集群：数据实例A, 数据实例B, 数据实例C, 提供主机的访问权限, 兼容OS权限, 用户自定义CPU的规格, 从100%到500%

存储

- ESSD：当前主流云盘, PL1到PL3性能会更强, 价格也在增加
- 本地SSD：挂在节点本地, 性能也不错, 价格在PL1和PL2之间
- 云盘SSD：已经逐渐被淘汰

规格代码示例

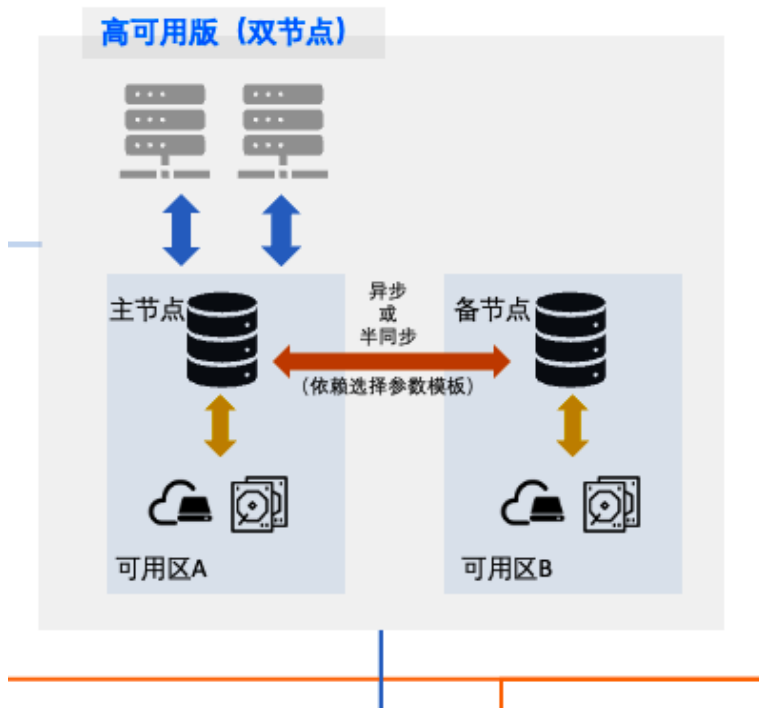
mysql.n4.xlarge.2c

- mysql: 代表是mysql数据库
- n4: 代表通用规格
- xlarge: 代表8 vCPU, 16c128g
- 2c: 代表2个节点

其他的一些说明

- micro: 通常代表1c1gb
- small: 通常代表1c2gb
- 有部分规格代码前面会带rds
- 规格代码如果是st, 则表示独占物理机
- v52和h43则是内部物理机的代码
- 这里2m是什么意思并不确定, 感觉可能没什么意义

阿里云RDS双节点高可用版： 内置参数模板



定位： 生产环境使用

数据可靠性： ★★★★★

可用性： ★★★★★

性能： ★★★★★★

价格： 中

MySQL_InnoDB_8.0_高可用版_高性能参数模版

系统参数模版

MySQL_InnoDB_8.0_高可用版_高性能参数模版

MySQL_InnoDB_8.0_高可用版_异步参数模版

MySQL_InnoDB_8.0_高可用版_默认参数模版

自定义参数模版

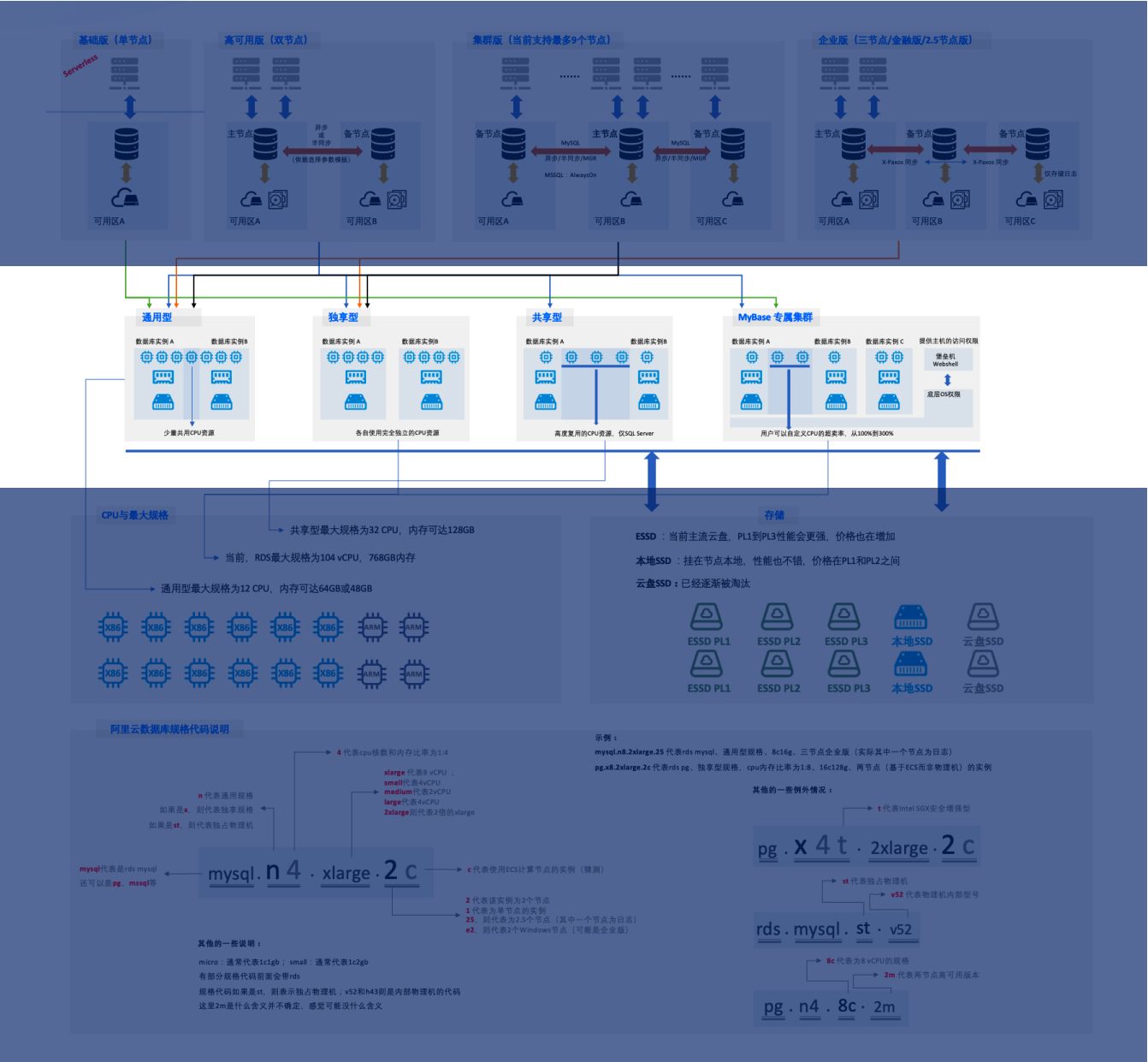
高性能

异步模式

默认

参数模板	高性能	异步模式	默认
同步模式	async	async	semi-sync
sync_binlog	1000	1	1
innodb_flush_log_at_trx_commit	2	1	1
定位	“最大性能”	“最大可用”	“最大保护”

阿里云RDS 规格类型



通用型

定位: 生产环境使用

最大规格: 12vCPU / 64GB

说明: 少量CPU共享

性能: 比较稳定

价格: 中

独享型

定位: 核心生产环境使用

最大规格: 104vCPU / 768GB

说明: 独享CPU

性能: 非常稳定

价格: 高

共享型

定位: 过时的, 少量规格

最大规格: 32vCPU / 128GB

说明: 较高CPU共享

性能: 比较稳定

价格: 低

MyBase

定位: 可定制的

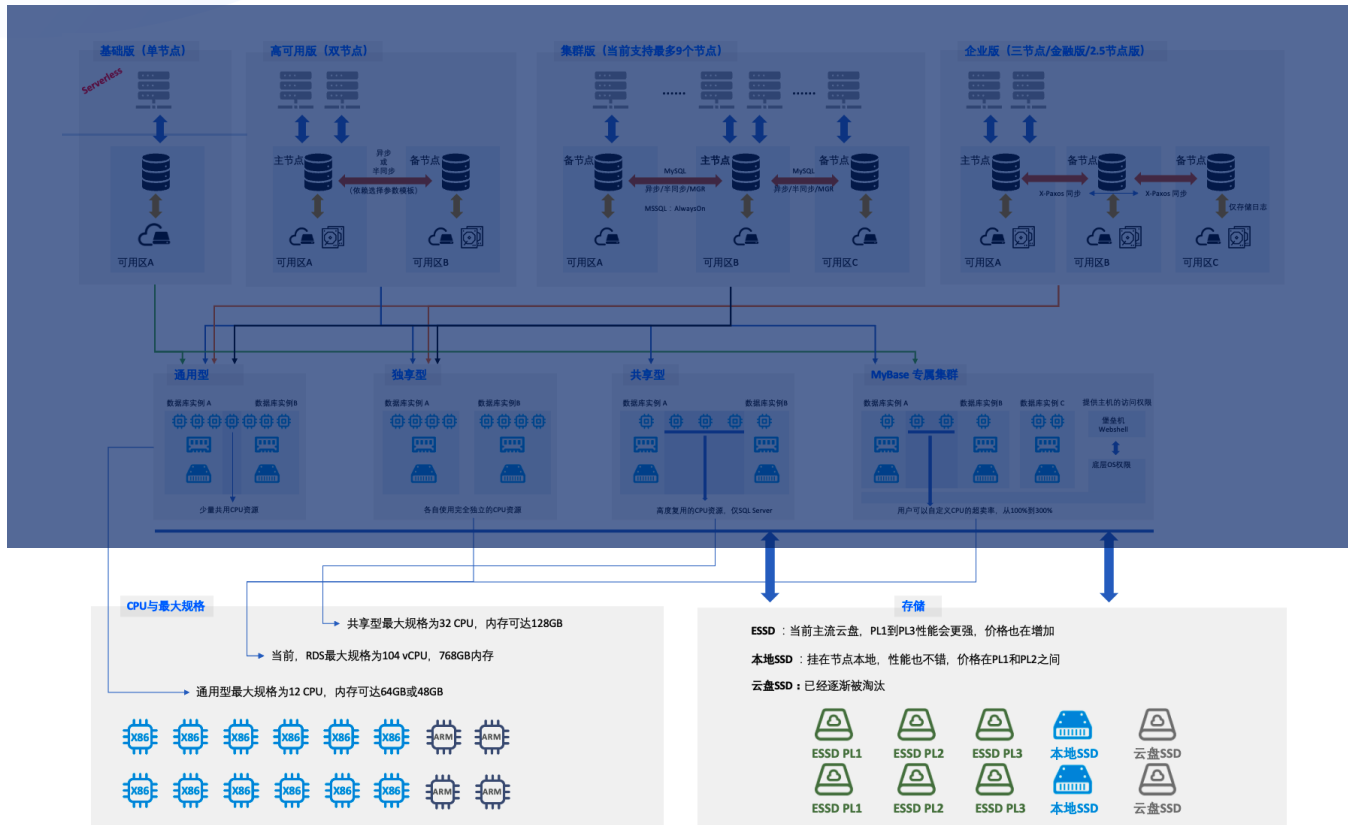
最大规格: 单机最大规格

说明: 独享主机

性能: 稳定

价格: 高

阿里云RDS CPU与存储



X86

定位: 主流架构

ARM架构

定位: 高性价比、自主可控

ESSD

PL1	PL2	PL3
最大IOPS: 50000	最大IOPS: 100000	最大IOPS: 1000000
最大存储: 32TB	最大存储: 32TB	最大存储: 32TB

本地 SSD

定位: 延迟低, 性能也很高

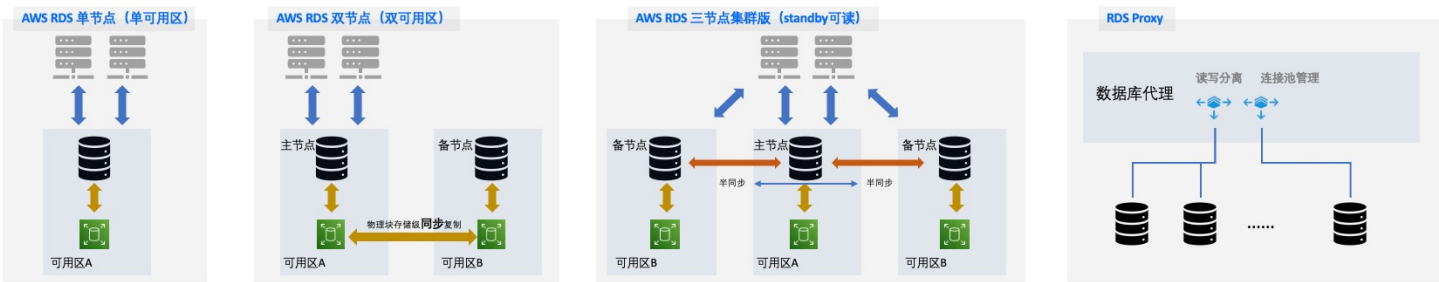
其他: 大空间扩展慢 (迁移)

SSD云盘 *

定位: 过时的, 逐步下线

- SSD云盘逐步停售的过程中: [参考](#)
- 云数据库存储类型官方说明: [参考](#)

Amazon RDS架构与选型(v0.5)

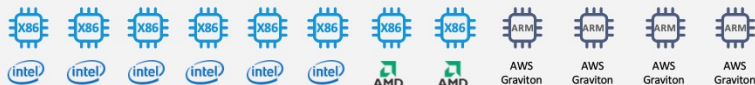


CPU与最大规格

标准型最大规格为：db.m6i.32xlarge 128vCPU 512GB

突发型最大规格为：db.t4g.2xlarge 8vCPU 32GB

内存优化型最大规格为：db.x2iedn.32xlarge 128vCPU 4096GB



存储层：gp2

定位：测试环境

SLA：99% 毫秒级

最大容量：64TB

最大IOPS：64000
(根据存储空间换算)

计费：存储空间

存储层：gp2

定位：性价比，OLTP

SLA：99% 毫秒级

最大容量：64 TB

最大IOPS：64000
(与存储空间分开，提供基线IOPS)

计费：存储空间
超出基线的IOPS
超出基线的带宽

存储层：预留IOPS SSD 简称：io1

定位：性能高、稳定性高的生产环境OLTP

SLA：99.9% 毫秒级

最大容量：64 TB

最大IOPS：256000
(与存储空间分开)

计费：存储空间
预留IOPS

存储：HDD磁盘

定位：过时的、向前兼容而保留

SLA：/

最大容量：3 TB

最大IOPS：1000

计费：存储空间
IO请求总数

标准型

M 代表通用规格 CPU内存比为1:4
 可能是medium/Moderate的缩写

6 代表第六代实例
 i 代表 Intel 处理器

db 代表这是用于数据库的实例

db.M6i.xlarge → xlarge 代表4 vCPU 16GB

内存优化型

R 代表内存优化型规格
 CPU内存比通常为1:8
 当前，最大规格为128c1024gb

5 代表第五代实例
 存储优化，通常支持 NVMe SSD

db 代表这是用于数据库的实例

db.R5d.2xlarge → 2xlarge 代表2倍xlarge大小18c64g

突发性能型

T 规格较小，具备一定性能突增能力
 可能是tiny的缩写
 最小规格为1c0.5g，最大为8c32g

6 代表第六代实例

db 代表这是用于数据库的实例

db.T3.medium → medium 代表2 vCPU 4GB

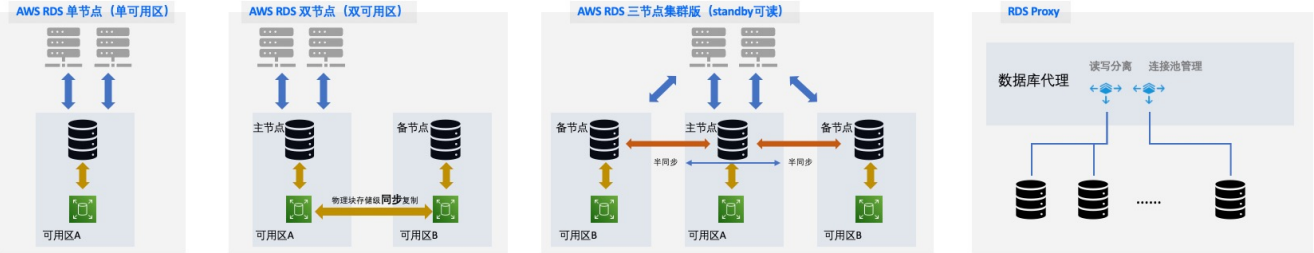
X 代表更大的内存优化型规格
 CPU内存比通常为1:16 / 1:30.5
 当前，最大规格为64c1024gb

1 代表第一代实例
 e 代表额外的内存或者存储空间

db 代表这是用于数据库的实例

db.X1e.4xlarge → 4xlarge 代表4倍xlarge大小16c488g

Amazon RDS: 高可用架构



CPU与最大规格

标准型最大规格为: db.m6i.32xlarge 128vCPU 512GB

突发型最大规格为: db.t4g.2xlarge 8vCPU 32GB

内存优化型最大规格为: db.x2iedn.32xlarge 128vCPU 4096GB

存储层: gp2

定位:	测试环境	性价比, OLTP	性能高、稳定性高的生产环境OLTP	过时的、向前兼容而保留
SLA:	99% 毫秒级	99% 毫秒级	99.9% 毫秒级	/
最大容量:	64TB	64 TB	64 TB	3 TB
最大IOPS:	64000 (根据存储空间换算)	64000 (与存储空间分开, 提供基线IOPS)	256000 (与存储空间分开)	1000
计费:	存储空间	存储空间 超出基线的IOPS 超出基线的带宽	存储空间 预留IOPS	存储空间 IO请求总数

存储: 预留IOPS SSD 简称: io1

存储: HDD磁盘

标准型

M 代表通用规格 CPU内存比为1:4
可能是medium/Moderate的缩写
db 代表这是用于数据库的实例

db.M6i.xlarge → xlarge 代表4 vCPU 16GB

内存优化型

R 代表内存优化型规格
CPU内存比通常为1:8
当前, 最大规格为128vCPU 512GB
存储优化, 通常支持 NVMe SSD

db.R5d.2xlarge → 2xlarge 代表2倍xlarge大小 8c64g

突发性性能

T规格较小, 具备一定性能突增能力
可能是tiny的缩写
最小规格为1vCPU 512MB, 最大为8vCPU 32GB

db.T3.medium → medium 代表2 vCPU 4GB

X 代表更大的内存优化型规格
CPU内存比通常为1:16 / 1:30.5
当前, 最大规格为64vCPU 1024GB

db.X1e.4xlarge → 4xlarge 代表4倍xlarge大小 16c488g

单节点

定位: 可靠性/可用性略低

数据可靠性: ★★★

可用性: ★★★

性能: ★★★★★

价格: 低

多可用区

定位: 生产环境使用

数据可靠性: ★★★★★

可用性: ★★★★★

性能: ★★★★★

价格: 中

多可用区集群版

定位: 生产环境使用

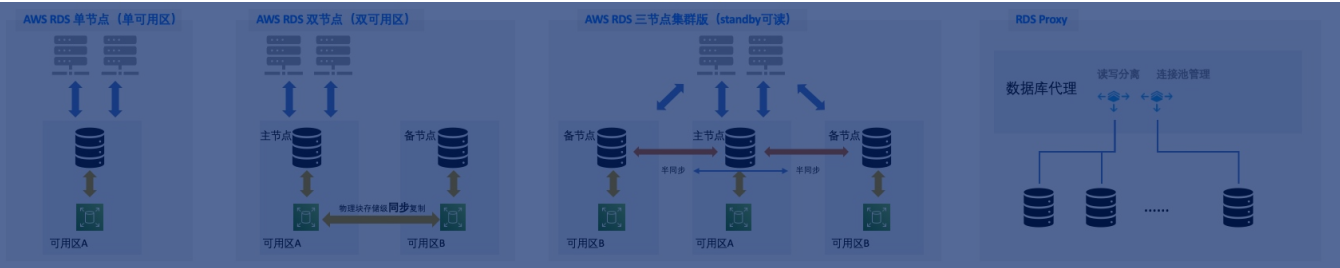
数据可靠性: ★★★★★

可用性: ★★★★★

性能: ★★★★★

价格: 较高

Amazon RDS: CPU架构



CPU与最大规格

标准型最大规格为: db.m6i.32xlarge 128vCPU 512GB

突发型最大规格为: db.t4g.2xlarge 8vCPU 32GB

内存优化型最大规格为: db.x2iedn.32xlarge 128vCPU 4096GB



X86/Intel

定位: 更通用/更广泛

性能: /

价格: /

ARM/Graviton

定位: 更高性价比

性能: /

价格: /

关于性能的一些观点:

“Graviton实例性价比高出53%”
([参考](#) from aws.amazon.com)

“Graviton在超高并发时性能更好, Intel在低/中并发性能更好”
([参考](#) from Percona)

存储层: gp2

定位: 测试环境

SLA: 99% 毫秒级

最大容量: 64TB

最大IOPS: 64000 (根据存储空间换算)

计费: 存储空间

存储层: gp2

定位: 性价比, OLTP

SLA: 99% 毫秒级

最大容量: 64 TB

最大IOPS: 64000 (与存储空间分开, 提供基线IOPS)

计费: 存储空间
超出基线的IOPS
超出基线的带宽

存储层: 预留IOPS SSD 简称: io1

定位: 性能高、稳定性高的生产环境OLTP

SLA: 99.9% 毫秒级

最大容量: 64 TB

最大IOPS: 256000 (与存储空间分开)

计费: 存储空间
预留IOPS

存储: HDD磁盘

定位: 过时的、向前兼容而保留

SLA: /

最大容量: 3TB

最大IOPS: 1000

计费: 存储空间
IO请求总数

标准型

M 代表通用规格 CPU内存比为1:4
可能是medium/Moderate的缩写

db 代表这是用于数据库的实例

db . **M6i** . **xlarge**

xlarge 代表4 vCPU 16GB

6 代表第六代实例

内存优化型

R 代表更大的内存优化规格
CPU内存比通常为1:8
当前, 最大规格为128c1024gb

db 代表这是用于数据库的实例

db . **R5d** . **2xlarge**

2xlarge 代表2倍xlarge大小8c64g

5 代表第五代实例

存储优化, 通常支持 NVMe SSD

突发性能型

T规格较小, 具备一定性能突增能力
可能是tiny的缩写
最小规格为1c0.5g, 最大为8c32g

db 代表这是用于数据库的实例

db . **T3** . **medium**

medium 代表2 vCPU 4GB

6 代表第六代实例

内存优化型

X 代表更大的内存优化规格
CPU内存比通常为1:16 / 1:30.5
当前, 最大规格为64c1024gb

db 代表这是用于数据库的实例

db . **X1e** . **4xlarge**

4xlarge 代表4倍xlarge大小16c64gb

1 代表第一代实例

e 代表额外的内存或者存储空间

Amazon RDS: 存储选型



存储层: gp2	存储层: gp2	存储层: 预留IOPS SSD 简称: io1	存储: HDD磁盘
定位: 测试环境	定位: 性价比高, OLTP	定位: 性能高、稳定性高的生产环境OLTP	定位: 过时的、向前兼容而保留
SLA: 99% 毫秒级	SLA: 99% 毫秒级	SLA: 99.9% 毫秒级	SLA: /
最大容量: 64TB	最大容量: 64 TB	最大容量: 64 TB	最大容量: 3 TB
最大IOPS: 64000 (根据存储空间换算)	最大IOPS: 64000 (与存储空间分开, 提供基线IOPS)	最大IOPS: 256000 (与存储空间分开)	最大IOPS: 1000
计费: 存储空间	计费: 存储空间 超出基线的IOPS 超出基线的带宽	计费: 存储空间 预留IOPS	计费: 存储空间 IO请求总数



gp2

定位: 测试环境

SLA: 99% 毫秒级

最大容量: 64 TB

最大IOPS: 64000
(据存储空间计算)

计费: 存储空间

gp3

定位: 性价比高, OLTP

SLA: 99% 毫秒级

最大容量: 64 TB

最大IOPS: 64000
(相对独立选择)

计费: 存储空间
超出基线的IOPS
超出基线的带宽

预留IOPS SSD/io1

定位: 性能高 / 稳定性高
生产环境OLTP

SLA: 99.9% 毫秒级

最大容量: 64 TB

最大IOPS: 256000
(与存储空间分开)

计费: 存储空间
预留IOPS

HDD磁盘

定位: 过时的
因兼容而保留

SLA: /

最大容量: 3 TB

最大IOPS: 1000

计费: 存储空间
IO请求总数

阿里云 RDS vs Amazon RDS

最大规格受限，仅小场景使用

基础版



单可用区版

实例大小不受限制，可达32xlarge

数据库层复制；性能/数据保护更灵活

高可用版本



多可用区版

EBS物理复制、架构统一；同步复制

较为稳定，可以考虑扩大验证范围
现更名为“经济版”

ARM



Graviton

更加成熟，对比X86性价比明显，
但似乎并没有性价比

阿里云 ESSD的IOPS上限更高
(pl3 百万 / PL-X 数百万)

ESSD



gp2/gp3 io1

Amazon的SLA更加明确，如io1 99.9% 毫秒级
计费上，Amazon更加复杂，包括存储空间、
IOPS基线等

提供共享、通用、独享、MyBase给
客户较强的定制能力

资源共享



突发型

仅在小规格提供；并提供了非常明确的规则

更加复杂；并不太强调

规格代码



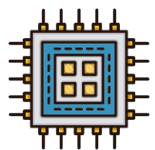
规格代码

规则清晰、连续；体现架构、大小、类型等

PART 05

总结

总结 (1)



CPU性能（代际）是本次测试中性能表现的主导因素，其次是存储的IO能力



更好的CPU通常能够获得更好的性能，IO表现在该系列测试中，占次要位置



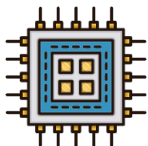
在性能的“一致性”上，AWS是典范，有着非常产品化的CPU代际管理



AWS通过实例型号，例如m5.xlarge、m6i、m6g、m7g等明确标识了CPU代际与架构



由于CPU代际的差异，阿里云、腾讯云都有较明显的性能不一致的情况发生

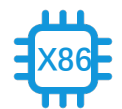


部分厂商会通过赠送更多的CPU core以获得更好的性能



可能并不是长远之计，但短期，开发者和企业依旧会受益

总结 (2)



VS



整体上，x86 vs ARM在不同的平台上的性价比会不同



在AWS， m6i vs m7g， m5 vs m6g， 低并发时x86实例都有更好的表现； 高并发m7g性能优势也比较明显



阿里云的RDS的经济型（ARM） 性价比要高32%



目前， 鲲鹏架构的优势并不在于性能



理论上， AWS/Azure/GCP， 更偏向于“持久性”， 华为/阿里/腾讯/百度云更偏向于性能



AWS/Azure/GCP都选择了“同步的存储架构”， 以保障持久性



华为/阿里/腾讯/百度云都选择了semi-sync作为同步架构， 以保障持久性

NineData云原生智能数据管理平台



数据开发



BI开发



业务运营



应用开发者



DBA



系统管理

数据仓库/大数据平台

应用消息队列

搜索平台

云厂商A

托管数据库 (开源、自研)

NineData SQL开发

数据查询

权限与审计

数据库变更发布

敏感数据管理

NineData 数据复制

同/异构支持

可观测/干预

NineData 数据对比

结构对比

数据对比

快速对比

NineData
数据备份

1分钟配置

跨区域容灾

RPO ~ 0

备份数据实时查询

按量付费

自建 IDC

自建数据库

云厂商B

托管数据库

丰富的数据源、平台支持: RDBMS、NoSQL、实时分析

MySQL

PostgreSQL

Microsoft SQL Server

ORACLE

Amazon Aurora

OCEANBASE

IBM DB2

PolarDB

DORIS

MongoDB

elasticsearch

redis

ClickHouse

SELECTDB

kafka

阿里云

aws

腾讯云

天翼云 E Cloud

HUAWEI

5G 移动云



IDC

Q & A