阿里云
云数据库 MySQL 版

性能白皮书

文档版本：20190618

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **禁止:** 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **警告:** 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 **说明:** 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
*## *	表示参数、变量。	`bae log list --instanceid` *`Instance_ID`*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *`[-all|-t]`*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *`{stand | slave}`*

目录

1 MySQL版

1.1 产品概述

云数据库RDS（Relational Database Service）是一种稳定可靠、可弹性伸缩的在线数据库服务。基于飞天分布式系统和全SSD盘高性能存储，支持MySQL、SQL Server、PostgreSQL和PPAS（高度兼容Oracle）引擎，默认部署主备架构且提供了容灾、备份、恢复、监控、迁移等方面的全套解决方案，彻底解决数据库运维的烦恼。

云数据库RDS提供了按量付费和包年包月两种付费方式，您可以根据业务压力配置RDS实例的规格。其中：

- 按量付费实例支持随时进行规格的升降级。
- 包年包月实例支持随时升降级和续费时升降级。

在面对极限的业务压力时，您还可以随时升级到RDS独占物理机规格来度过意料外的状况。

1.2 MySQL 5.7

1.2.1 规格列表

通用型

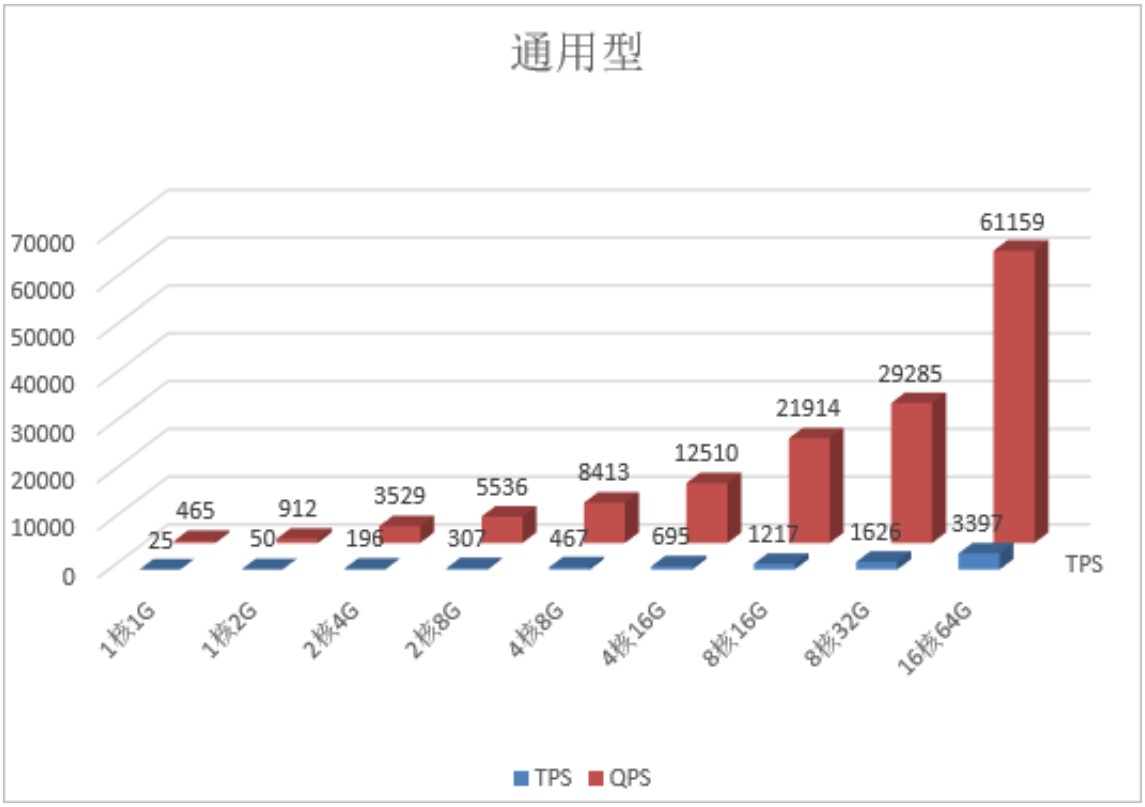
规格编号	CPU核数	内存（GB）	连接数	IOPS	TPS	QPS
rds.mysql.t1.small	1	1	300	600	25	465
rds.mysql.s1.small	1	2	600	1000	50	912
rds.mysql.s2.large	2	4	1200	2000	196	3529
rds.mysql.s2.xlarge	2	8	2000	4000	307	5536
rds.mysql.s3.large	4	8	2000	5000	467	8413
rds.mysql.m1.medium	4	16	4000	4000	695	12510
rds.mysql.c1.large	8	16	4000	8000	1217	21914

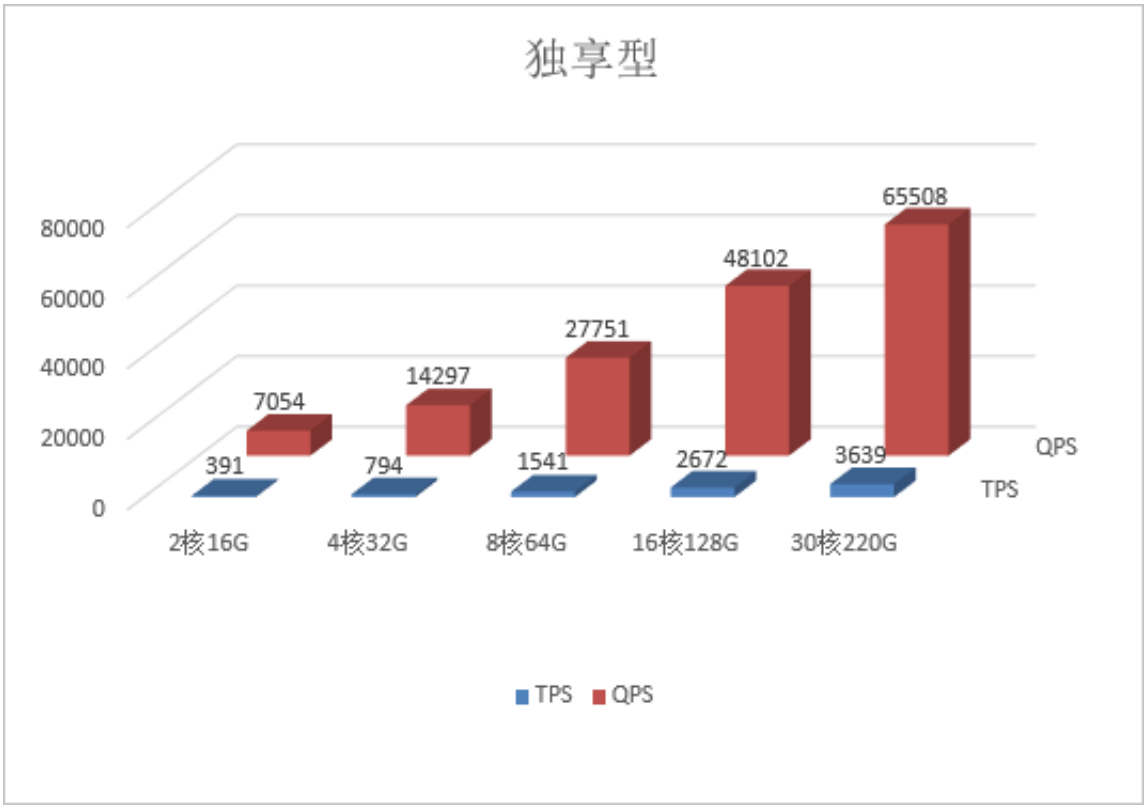
规格编号	CPU核数	内存（GB）	连接数	IOPS	TPS	QPS
rds.mysql.c1.xlarge	8	32	8000	12000	1626	29285
rds.mysql.c2.xlarge	16	64	16000	14000	3397	61159

独享型

规格编号	CPU核数	内存（GB）	连接数	IOPS	TPS	QPS
mysql.x8.medium.2	2	16	2500	4500	391	7054
mysql.x8.large.2	4	32	5000	9000	794	14297
mysql.x8.xlarge.2	8	64	10000	18000	1541	27751
mysql.x8.2xlarge.2	16	128	20000	36000	2672	48102
rds.mysql.st.d13	30	220	64000	20000	3693	65508

1.2.2 测试结果


通用型
70000
60000
50000
40000
30000
20000
10000
0
25
465
50
912
196
3529
307
5536
467
8413
695
12510
1217
21914
1626
29285
3397
61159
TPS
1核1G
1核2G
2核4G
2核8G
4核8G
4核16G
8核16G
8核32G
16核64G
TPS
QPS


独享型
80000
60000
40000
20000
0
391
7054
794
14297
1541
27751
2672
48102
3639
65508
QPS
TPS
2核16G
4核32G
8核64G
16核128G
30核220G
TPS
QPS

2 PPAS版

2.1 产品概述

云数据库RDS（Relational Database Service）是一种稳定可靠、可弹性伸缩的在线数据库服务。基于飞天分布式系统和全SSD盘高性能存储，支持MySQL、SQL Server、PostgreSQL、PPAS（高度兼容Oracle）和MariaDB引擎，默认部署主备架构且提供了容灾、备份、恢复、监控、迁移等方面的全套解决方案，彻底解决数据库运维的烦恼。

云数据库RDS提供了按量付费和包年包月两种付费方式，您可以根据业务压力配置RDS实例的规格。其中：

- 按量付费实例支持随时进行规格的升降级。
- 包年包月实例支持随时升降级和续费时升降级。

在面对极限的业务压力时，您还可以随时升级到RDS独占物理机规格来度过意料外的状况。

2.2 测试方法

2.2.1 测试环境

- 所有测试均在华北2（北京）地域完成，PPAS实例和ECS实例在同一可用区。
- ECS的实例规格：ecs.g5.16xlarge（64核 256GiB）
- ECS存储规格： SSD本地盘 200GiB。
- 网络类型：专有网络。
- 操作系统：CentOS 7.6 x64

2.2.2 测试工具

pgbench简介

PostgreSQL自带一款轻量级的压力测试工具：pgbench。pgbench是一种在PostgreSQL上运行基准测试的简单程序。它可以在并发的数据库会话中一遍一遍地运行相同的SQL命令。

安装方法

1. 执行如下命令在ECS实例中安装PostgreSQL 11。

```
yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

```
yum install -y https://download.postgresql.org/pub/repos/yum/11/
redhat/rhel-7-x86_64/pgdg-centos11-11-2.noarch.rpm
yum install -y postgresql11*

su - postgres
vi .bash_profile
export PS1="$USER@`/bin/hostname -s`-> "
export LANG=en_US.utf8
export PGHOME=/usr/pgsql-11
export LD_LIBRARY_PATH=$PGHOME/lib:/lib64:/usr/lib64:/usr/local/
lib64:/lib:/usr/lib:/usr/local/lib:$LD_LIBRARY_PATH
export DATE=`date +"%Y%m%d%H%M"`
export PATH=$PGHOME/bin:$PATH:.
export MANPATH=$PGHOME/share/man:$MANPATH
alias rm='rm -i'
alias ll='ls -lh'
unalias vi
unalias vi
```

2.2.3 测试步骤

1. **在测试前需要提交工单申请，修改PPAS实例参数，本文以PPAS 10版本为例，需要修改主备实例上的*postgresql.auto.conf*文件。**

```
edb_redwood_date = on
edb_redwood_greatest_least = on
edb_redwood_strings = on
edb_redwood_raw_names = on
edb_stmt_level_tx = off
db_dialect = 'redwood'
optimizer_mode = choose
edb_early_lock_release = on
datestyle = 'iso, ymd'
default_with_oids = off
default_with_rowids = off
vacuum_cost_delay = 0
bgwriter_delay = 10ms
bgwriter_lru_maxpages = 1000
bgwriter_lru_multiplier = 10.0
effective_io_concurrency = 0
max_worker_processes = 128
max_parallel_workers_per_gather = 0
synchronous_commit = off
wal_compression = on
wal_writer_delay = 10ms
wal_writer_flush_after = 1MB
checkpoint_timeout = 30min
max_wal_size = 64GB    # 1/2 当前PPAS实例的规格内存
min_wal_size = 16GB    # 1/8 当前PPAS实例的规格内存
checkpoint_completion_target = 0.2
hot_standby_feedback = off
random_page_cost = 1.1
log_checkpoints = on
log_statement = 'ddl'
log_autovacuum_min_duration = 0
autovacuum_freeze_max_age = 1500000000
autovacuum_multixact_freeze_max_age = 1600000000
autovacuum_vacuum_cost_delay = 0ms
vacuum_freeze_table_age = 1450000000
vacuum_multixact_freeze_table_age = 1450000000
```

```
log_min_duration_statement=5s
```

2. 修改配置后，重启PPAS实例让配置生效。
3. 根据目标库大小初始化测试数据，具体命令如下。
   - 初始化数据50亿：`pgbench -i -s 50000`
   - 初始化数据10亿：`pgbench -i -s 10000`
   - 初始化数据5亿：`pgbench -i -s 5000`
   - 初始化数据1亿：`pgbench -i -s 1000`
4. 通过以下命令配置环境变量：

```
export PGHOST=<PPAS实例内网地址>
export PGPORT=<PPAS实例端口>
export PGDATABASE=postgres
export PGUSER=<PPAS数据库用户名>
export PGPASSWORD=<PPAS对应用户的密码>
```

5. 创建只读和读写的测试脚本。
   - 创建只读脚本ro.sql内容如下：

```
\set aid random_gaussian(1, :range, 10.0)
SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
```

   - 创建读写脚本rw.sql内容如下：

```
\set aid random_gaussian(1, :range, 10.0)
\set bid random(1, 1 * :scale)
\set tid random(1, 10 * :scale)
\set delta random(-5000, 5000)
BEGIN;
UPDATE pgbench_accounts SET abalance = abalance + :delta WHERE aid
 = :aid;
SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE tid
 = :tid;
UPDATE pgbench_branches SET bbalance = bbalance + :delta WHERE bid
 = :bid;
INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VALUES
 (:tid, :bid, :aid, :delta, CURRENT_TIMESTAMP);
END;
```

6. 使用如下命令测试。
   - 只读测试：

```
rds.ppas.st.h43，总数据量50亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 480 -j 480 -T 120 -D
 scale=50000 -D range=100000000

rds.ppas.st.h43，总数据量50亿，热数据5亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 480 -j 480 -T 120 -D
 scale=50000 -D range=500000000
```

```
rds.ppas.st.h43，总数据量50亿，热数据10亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 480 -j 480 -T 120 -D
 scale=50000 -D range=1000000000

rds.ppas.st.h43，总数据量50亿，热数据50亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 480 -j 480 -T 120 -D
 scale=50000 -D range=5000000000

ppas.x8.4xlarge.2，总数据量10亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 320 -j 320 -T 120 -D
 scale=10000 -D range=100000000

ppas.x8.4xlarge.2，总数据量10亿，热数据5亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 320 -j 320 -T 120 -D
 scale=10000 -D range=500000000

ppas.x8.4xlarge.2，总数据量10亿，热数据10亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 320 -j 320 -T 120 -D
 scale=10000 -D range=1000000000

ppas.x4.4xlarge.2，总数据量10亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 320 -j 320 -T 120 -D
 scale=10000 -D range=100000000

ppas.x4.4xlarge.2，总数据量10亿，热数据5亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 320 -j 320 -T 120 -D
 scale=10000 -D range=500000000

ppas.x4.4xlarge.2，总数据量10亿，热数据10亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 320 -j 320 -T 120 -D
 scale=10000 -D range=1000000000

ppas.x8.2xlarge.2，总数据量10亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D
scale=10000 -D range=100000000

ppas.x8.2xlarge.2，总数据量10亿，热数据5亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D
scale=10000 -D range=500000000

ppas.x8.2xlarge.2，总数据量10亿，热数据10亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D
scale=10000 -D range=1000000000

ppas.x8.xlarge.2，总数据量10亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D
scale=10000 -D range=100000000

ppas.x8.xlarge.2，总数据量10亿，热数据5亿
```

```
pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D
scale=10000 -D range=500000000

ppas.x8.xlarge.2，总数据量10亿，热数据10亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D
scale=10000 -D range=1000000000

ppas.x8.large.2，总数据量5亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 16 -j 16 -T 120 -D
scale=5000 -D range=100000000

ppas.x8.large.2，总数据量5亿，热数据5亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 16 -j 16 -T 120 -D
scale=5000 -D range=500000000
```

- 读写测试：

```
rds.ppas.st.h43，总数据量50亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 480 -j 480 -T 120 -D
 scale=50000 -D range=100000000

rds.ppas.st.h43，总数据量50亿，热数据5亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 480 -j 480 -T 120 -D
 scale=50000 -D range=500000000

rds.ppas.st.h43，总数据量50亿，热数据10亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 480 -j 480 -T 120 -D
 scale=50000 -D range=1000000000

rds.ppas.st.h43，总数据量50亿，热数据50亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 480 -j 480 -T 120 -D
 scale=50000 -D range=5000000000

ppas.x8.4xlarge.2，总数据量10亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 320 -j 320 -T 120 -D
 scale=10000 -D range=100000000

ppas.x8.4xlarge.2，总数据量10亿，热数据5亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 320 -j 320 -T 120 -D
 scale=10000 -D range=500000000

ppas.x8.4xlarge.2，总数据量10亿，热数据10亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 320 -j 320 -T 120 -D
 scale=10000 -D range=1000000000

ppas.x4.4xlarge.2，总数据量10亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 320 -j 320 -T 120 -D
 scale=10000 -D range=100000000

ppas.x4.4xlarge.2，总数据量10亿，热数据5亿
```

```
pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 320 -j 320 -T 120 -D
 scale=10000 -D range=500000000

ppas.x4.4xlarge.2，总数据量10亿，热数据10亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 320 -j 320 -T 120 -D
 scale=10000 -D range=1000000000

ppas.x8.2xlarge.2，总数据量10亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D
scale=10000 -D range=100000000

ppas.x8.2xlarge.2，总数据量10亿，热数据5亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D
scale=10000 -D range=500000000

ppas.x8.2xlarge.2，总数据量10亿，热数据10亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D
scale=10000 -D range=1000000000

ppas.x8.xlarge.2，总数据量10亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D
scale=10000 -D range=100000000

ppas.x8.xlarge.2，总数据量10亿，热数据5亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D
scale=10000 -D range=500000000

ppas.x8.xlarge.2，总数据量10亿，热数据10亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D
scale=10000 -D range=1000000000

ppas.x8.large.2，总数据量5亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 16 -j 16 -T 120 -D
scale=5000 -D range=100000000

ppas.x8.large.2，总数据量5亿，热数据5亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 16 -j 16 -T 120 -D
scale=5000 -D range=500000000
```

说明:

- scale乘以10万：表示测试数据量。
- range：表示活跃数据量。
- -c：表示测试连接数，测试连接数不代表该规格的最大连接数，最大连接数请参考实例规格表。

2.2.4 测试指标

只读QPS

数据库只读时每秒执行的SQL数（仅包含select）。

读写QPS

数据库读写时每秒执行的SQL数（含insert、select、update）。

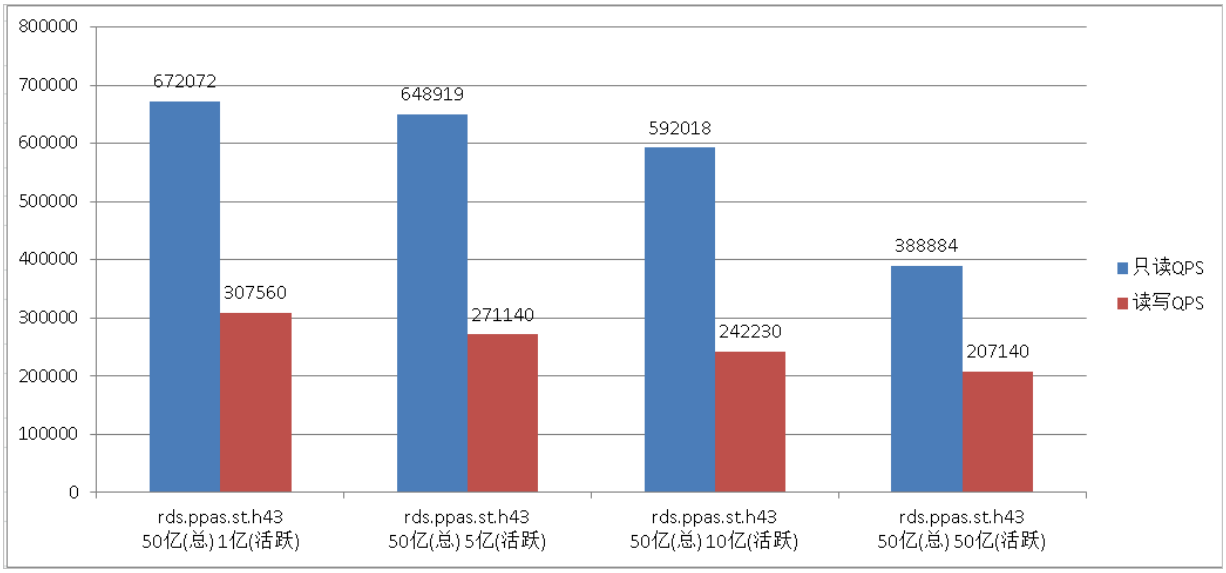
2.3 测试结果

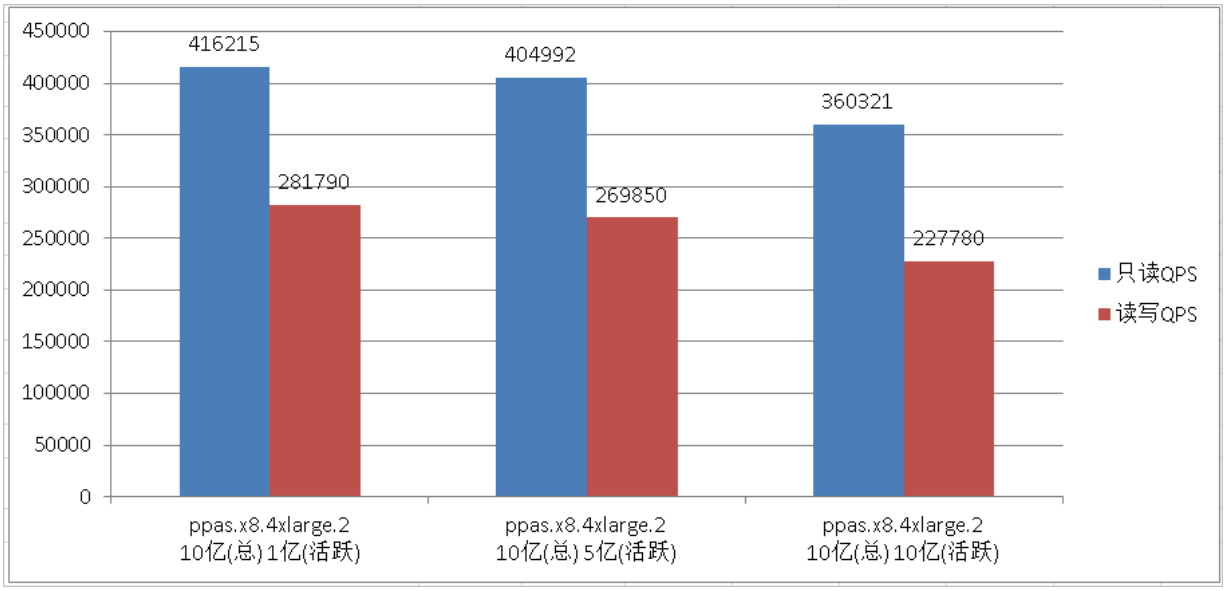
规格	预计可存储数据量	测试数据量	热(活跃)数据量	只读QPS	读写QPS
rds.ppas.st.h43 60核470G3T	150亿	50亿	1亿	672072	307560
rds.ppas.st.h43 60核470G3T	150亿	50亿	5亿	648919	271140
rds.ppas.st.h43 60核470G3T	150亿	50亿	10亿	592018	242230
rds.ppas.st.h43 60核470G3T	150亿	50亿	50亿	388884	207140
ppas.x8.4xlarge.2 32核256G2T	100亿	10亿	1亿	416215	281790
ppas.x8.4xlarge.2 32核256G 2T	100亿	10亿	5亿	404992	269850
ppas.x8.4xlarge.2 32核256G 2T	100亿	10亿	10亿	360321	227780
ppas.x4.4xlarge.2 32核128G2T	100亿	10亿	1亿	416760	253000
ppas.x4.4xlarge.2 32核128G2T	100亿	10亿	5亿	405528	233615

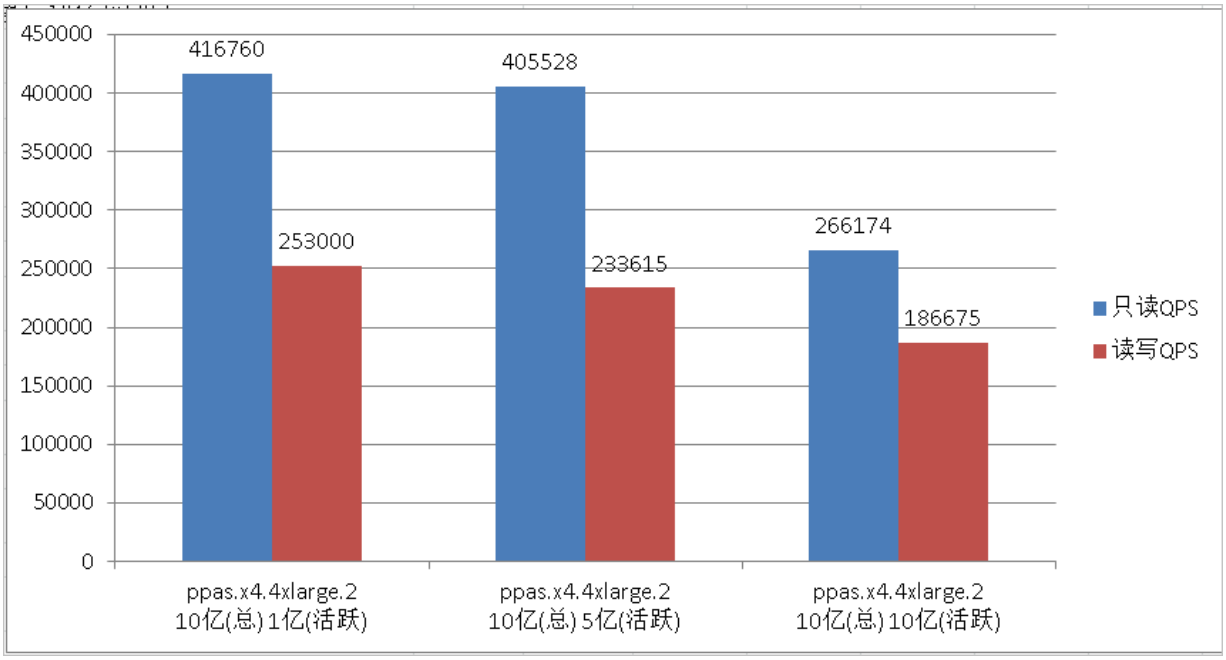
规格	预计可存储数据量	测试数据量	热(活跃)数据量	只读QPS	读写QPS
ppas.x4.4xlarge.2 32核128G2T	100亿	10亿	10亿	266174	186675
ppas.x8.2xlarge.2 16核128G2T	100亿	10亿	1亿	268319	170955
ppas.x8.2xlarge.2 16核128G2T	100亿	10亿	5亿	234482	142335
ppas.x8.2xlarge.2 16核128G2T	100亿	10亿	10亿	214005	123700
ppas.x8.xlarge.2 8核64G1T	50亿	10亿	1亿	142283	86810
ppas.x8.xlarge.2 8核64G1T	50亿	10亿	5亿	117538	69055
ppas.x8.xlarge.2 8核64G1T	50亿	10亿	10亿	101744	57500
ppas.x8.large.2 4核32G500G	25亿	5亿	1亿	70701	42625
ppas.x8.large.2 4核32G500G	25亿	5亿	5亿	57415	26255

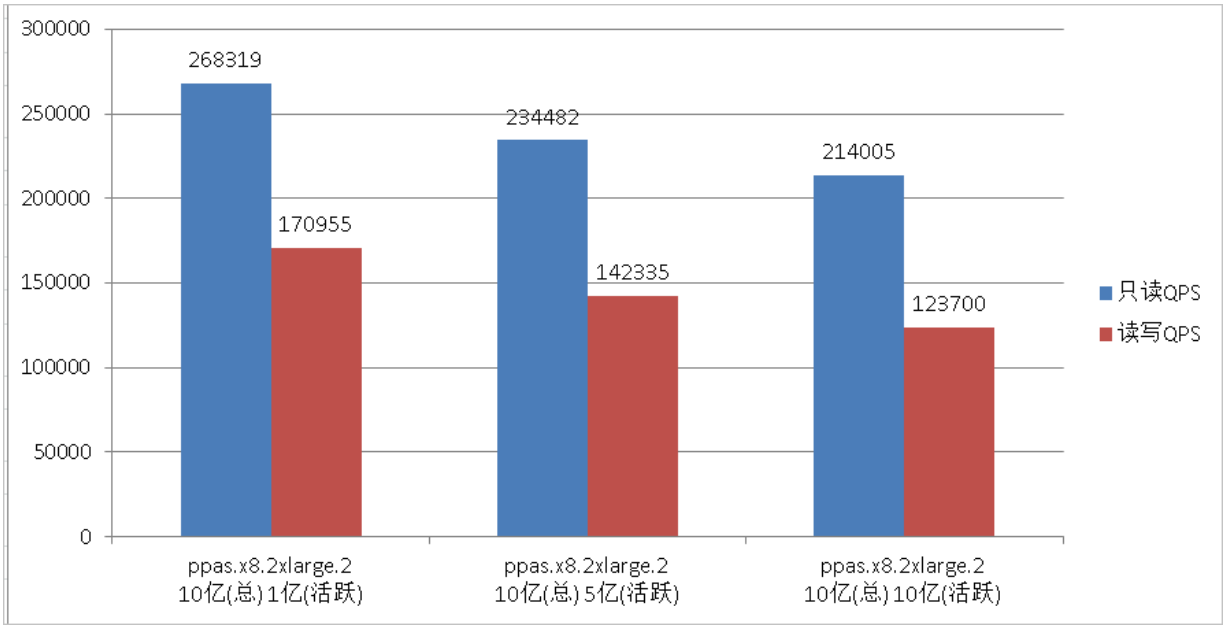
说明:

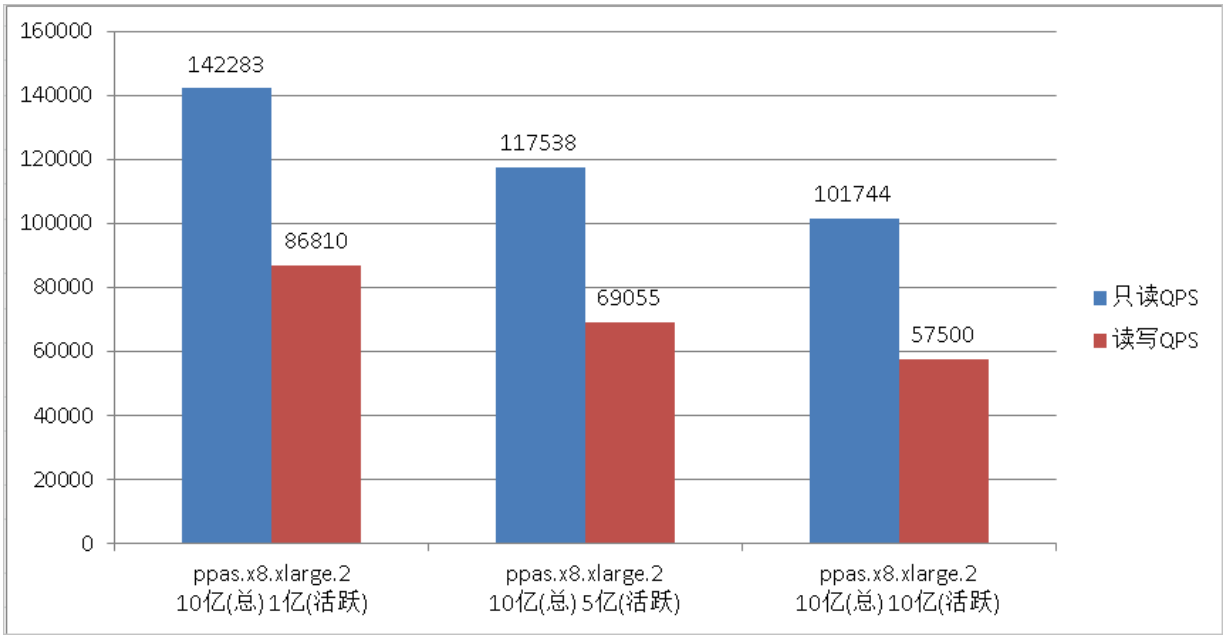
- 规格：RDS for PPAS的规格代码。
- 预计可存储数据量：预计该规格实例可以存储多少条记录。
- 测试数据量：本轮测试数据的记录数。
- 热（活跃）数据量：本轮测试的查询、更新SQL的记录数范围。
- 只读QPS：只读测试的结果，表示每秒请求数。
- 读写QPS：读写测试的结果，表示每秒请求数。

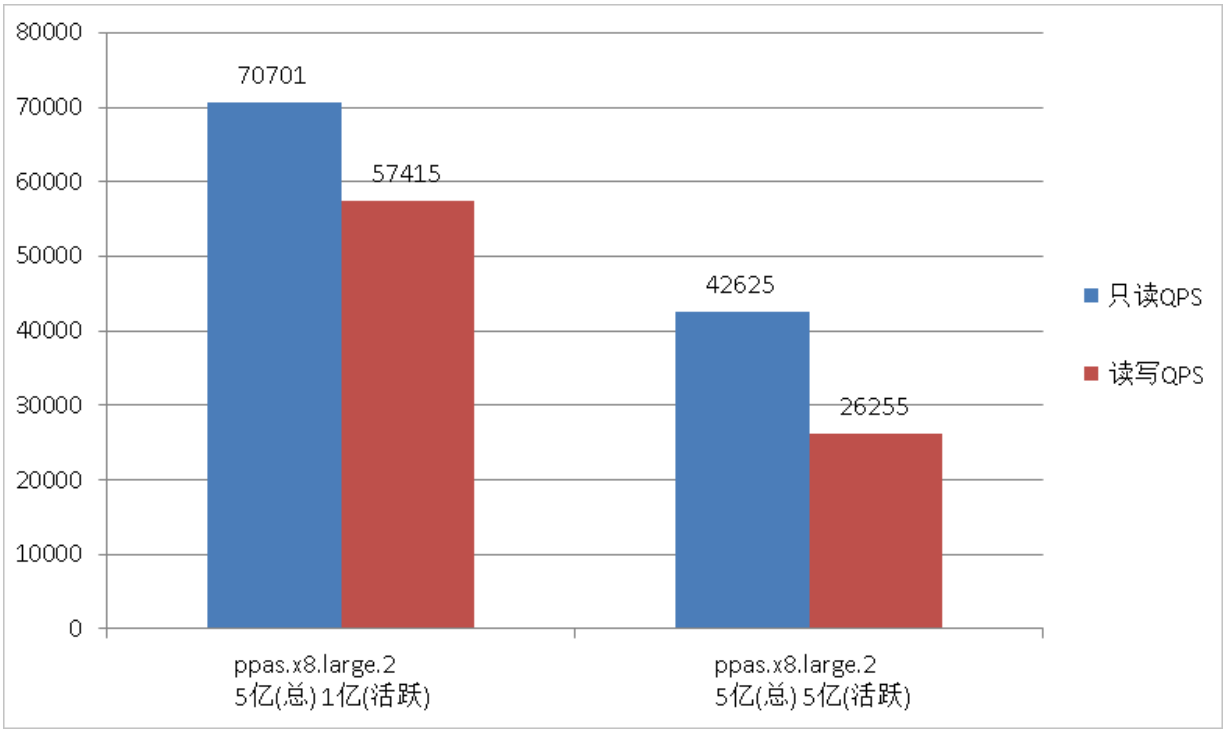

800000
700000
600000
500000
400000
300000
200000
100000
0
672072
307560
648919
271140
592018
242230
388884
207140
只读QPS
读写QPS
rds.ppas.st.h43
50亿(总.) 1亿(活跃)
rds.ppas.st.h43
50亿(总.) 5亿(活跃)
rds.ppas.st.h43
50亿(总.) 10亿(活跃)
rds.ppas.st.h43
50亿(总.) 50亿(活跃)


450000
400000
350000
300000
250000
200000
150000
100000
50000
0
416215
281790
404992
269850
360321
227780
只读QPS
读写QPS
ppas.x8.4xlarge.2
10亿(总.) 1亿(活跃)
ppas.x8.4xlarge.2
10亿(总.) 5亿(活跃)
ppas.x8.4xlarge.2
10亿(总.) 10亿(活跃)


450000
400000
350000
300000
250000
200000
150000
100000
50000
0
416760
253000
405528
233615
266174
186675
只读QPS
读写QPS
ppas.x4.4xlarge.2
10亿(总.) 1亿(活跃)
ppas.x4.4xlarge.2
10亿(总.) 5亿(活跃)
ppas.x4.4xlarge.2
10亿(总.) 10亿(活跃)


300000
250000
200000
150000
100000
50000
0
268319
170955
234482
142335
214005
123700
只读QPS
读写QPS
ppas.x8.2xlarge.2
10亿(总) 1亿(活跃)
ppas.x8.2xlarge.2
10亿(总) 5亿(活跃)
ppas.x8.2xlarge.2
10亿(总) 10亿(活跃)


160000
140000
120000
100000
80000
60000
40000
20000
0
142283
86810
117538
69055
101744
57500
只读QPS
读写QPS
ppas.x8.xlarge.2
10亿(总) 1亿(活跃)
ppas.x8.xlarge.2
10亿(总) 5亿(活跃)
ppas.x8.xlarge.2
10亿(总) 10亿(活跃)


80000
70000
60000
50000
40000
30000
20000
10000
0
70701
57415
42625
26255
只读QPS
读写QPS
ppas.x8.large.2
5亿(总.) 1亿(活跃)
ppas.x8.large.2
5亿(总.) 5亿(活跃)

3 PostgreSQL版

3.1 产品概述

云数据库RDS（Relational Database Service）是一种稳定可靠、可弹性伸缩的在线数据库服务。基于飞天分布式系统和全SSD盘高性能存储，支持MySQL、SQL Server、PostgreSQL、PPAS（高度兼容Oracle）和MariaDB引擎，默认部署主备架构且提供了容灾、备份、恢复、监控、迁移等方面的全套解决方案，彻底解决数据库运维的烦恼。

云数据库RDS提供了按量付费和包年包月两种付费方式，您可以根据业务压力配置RDS实例的规格。其中：

- 按量付费实例支持随时进行规格的升降级。
- 包年包月实例支持随时升降级和续费时升降级。

在面对极限的业务压力时，您还可以随时升级到RDS独占物理机规格来度过意料外的状况。

3.2 测试方法

3.2.1 测试环境

- 所有测试均在华北2（北京）地域完成，PostgreSQL实例和ECS实例在同一可用区。
- ECS的实例规格：ecs.g5.16xlarge（64核 256GiB）
- ECS存储规格： SSD本地盘 200GiB。
- 网络类型：专有网络。
- 操作系统：CentOS 7.6 x64

3.2.2 测试工具

pgbench简介

PostgreSQL自带一款轻量级的压力测试工具：pgbench。pgbench是一种在PostgreSQL上运行基准测试的简单程序。它可以在并发的数据库会话中一遍一遍地运行相同的SQL命令。

安装方法

1. 执行如下命令在ECS实例中安装PostgreSQL 11。

```
yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

```
yum install -y https://download.postgresql.org/pub/repos/yum/11/
redhat/rhel-7-x86_64/pgdg-centos11-11-2.noarch.rpm
yum install -y postgresql11*

su - postgres
vi .bash_profile
export PS1="$USER@`/bin/hostname -s`-> "
export LANG=en_US.utf8
export PGHOME=/usr/pgsql-11
export LD_LIBRARY_PATH=$PGHOME/lib:/lib64:/usr/lib64:/usr/local/
lib64:/lib:/usr/lib:/usr/local/lib:$LD_LIBRARY_PATH
export DATE=`date +"%Y%m%d%H%M"`
export PATH=$PGHOME/bin:$PATH:.
export MANPATH=$PGHOME/share/man:$MANPATH
alias rm='rm -i'
alias ll='ls -lh'
unalias vi
```

3.2.3 测试步骤

1. **在测试前需要提交工单申请，修改PostgreSQL实例参数，本文以PostgreSQL10.0 版本为例，需要修改主备实例上的*postgresql.auto.conf*文件。**

```
vacuum_cost_delay = 0
bgwriter_delay = 10ms
bgwriter_lru_maxpages = 1000
bgwriter_lru_multiplier = 10.0
effective_io_concurrency = 0
max_worker_processes = 128
max_parallel_workers_per_gather = 0
synchronous_commit = off
wal_compression = on
wal_writer_delay = 10ms
wal_writer_flush_after = 1MB
checkpoint_timeout = 30min
max_wal_size = 64GB    # 1/2 当前PG实例的规格内存
min_wal_size = 16GB    # 1/8 当前PG实例的规格内存
checkpoint_completion_target = 0.2
hot_standby_feedback = off
random_page_cost = 1.1
log_checkpoints = on
log_statement = 'ddl'
log_autovacuum_min_duration = 0
autovacuum_freeze_max_age = 1500000000
autovacuum_multixact_freeze_max_age = 1600000000
autovacuum_vacuum_cost_delay = 0ms
vacuum_freeze_table_age = 1450000000
vacuum_multixact_freeze_table_age = 1450000000
log_min_duration_statement=5s
```

2. **修改配置后，重启PostgreSQL实例让配置生效。**

3. 根据目标库大小初始化测试数据，具体命令如下。
   - 初始化数据50亿：`pgbench -i -s 50000`
   - 初始化数据10亿：`pgbench -i -s 10000`
   - 初始化数据5亿：`pgbench -i -s 5000`
   - 初始化数据1亿：`pgbench -i -s 1000`
4. 通过以下命令配置环境变量：

```
export PGHOST=<PostgreSQL实例内网地址>
export PGPORT=<PostgreSQL实例端口>
export PGDATABASE=postgres
export PGUSER=<PostgreSQL数据库用户名>
export PGPASSWORD=<PostgreSQL对应用户的密码>
```

5. 创建只读和读写的测试脚本。
   - 创建只读脚本ro.sql内容如下：

```
\set aid random_gaussian(1, :range, 10.0)
SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
```

   - 创建读写脚本rw.sql内容如下：

```
\set aid random_gaussian(1, :range, 10.0)
\set bid random(1, 1 * :scale)
\set tid random(1, 10 * :scale)
\set delta random(-5000, 5000)
BEGIN;
UPDATE pgbench_accounts SET abalance = abalance + :delta WHERE aid
 = :aid;
SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE tid
 = :tid;
UPDATE pgbench_branches SET bbalance = bbalance + :delta WHERE bid
 = :bid;
INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VALUES
 (:tid, :bid, :aid, :delta, CURRENT_TIMESTAMP);
END;
```

6. 使用如下命令测试。
   - 只读测试：

```
rds.pg.st.h43，总数据量50亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 240 -j 240 -T 120 -D
 scale=50000 -D range=100000000

rds.pg.st.h43，总数据量50亿，热数据5亿

pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 240 -j 240 -T 120 -D
 scale=50000 -D range=500000000

rds.pg.st.h43，总数据量50亿，热数据10亿
```

```
pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 240 -j 240 -T 120 -D
 scale=50000 -D range=1000000000
```

rds.pg.st.h43，**总数据量50亿，热数据50亿**

```
pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 240 -j 240 -T 120 -D
 scale=50000 -D range=5000000000
```

pg.x4.4xlarge.2，**总数据量10亿，热数据1亿**

```
pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 128 -j 128 -T 120 -D
 scale=10000 -D range=100000000
```

pg.x4.4xlarge.2，**总数据量10亿，热数据5亿**

```
pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 128 -j 128 -T 120 -D
 scale=10000 -D range=500000000
```

pg.x4.4xlarge.2，**总数据量10亿，热数据10亿**

```
pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 128 -j 128 -T 120 -D
 scale=10000 -D range=1000000000
```

pg.x8.2xlarge.2，**总数据量10亿，热数据1亿**

```
pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D
scale=10000 -D range=100000000
```

pg.x8.2xlarge.2，**总数据量10亿，热数据5亿**

```
pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D
scale=10000 -D range=500000000
```

pg.x8.2xlarge.2，**总数据量10亿，热数据10亿**

```
pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 64 -j 64 -T 120 -D
scale=10000 -D range=1000000000
```

pg.x8.xlarge.2，**总数据量10亿，热数据1亿**

```
pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D
scale=10000 -D range=100000000
```

pg.x8.xlarge.2，**总数据量10亿，热数据5亿**

```
pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D
scale=10000 -D range=500000000
```

pg.x8.xlarge.2，**总数据量10亿，热数据10亿**

```
pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 32 -j 32 -T 120 -D
scale=10000 -D range=1000000000
```

pg.x8.large.2，**总数据量5亿，热数据1亿**

```
pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 16 -j 16 -T 120 -D
scale=5000 -D range=100000000
```

pg.x8.large.2，**总数据量5亿，热数据5亿**

```
pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 16 -j 16 -T 120 -D
scale=5000 -D range=500000000
```

pg.x8.medium.2，**总数据量1亿，热数据5000万**

```
pgbench -M prepared -v -r -P 1 -f ./ro.sql -c 8 -j 8 -T 120 -D
scale=1000 -D range=50000000

pg.x8.medium.2，总数据量1亿，热数据1亿

pgbench -M prepared -n -r -P 1 -f ./ro.sql -c 8 -j 8 -T 120 -D
scale=1000 -D range=100000000
```

- **读写测试：**

```
rds.pg.st.h43，总数据量50亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 240 -j 240 -T 120 -D
 scale=50000 -D range=100000000

rds.pg.st.h43，总数据量50亿，热数据5亿

pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 240 -j 240 -T 120 -D
 scale=50000 -D range=500000000

rds.pg.st.h43，总数据量50亿，热数据10亿

pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 240 -j 240 -T 120 -D
 scale=50000 -D range=1000000000

rds.pg.st.h43，总数据量50亿，热数据50亿

pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 240 -j 240 -T 120 -D
 scale=50000 -D range=5000000000

pg.x4.4xlarge.2，总数据量10亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 128 -j 128 -T 120 -D
 scale=10000 -D range=100000000

pg.x4.4xlarge.2，总数据量10亿，热数据5亿

pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 128 -j 128 -T 120 -D
 scale=10000 -D range=500000000

pg.x4.4xlarge.2，总数据量10亿，热数据10亿

pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 128 -j 128 -T 120 -D
 scale=10000 -D range=1000000000

pg.x8.2xlarge.2，总数据量10亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D
scale=10000 -D range=100000000

pg.x8.2xlarge.2，总数据量10亿，热数据5亿

pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D
scale=10000 -D range=500000000

pg.x8.2xlarge.2，总数据量10亿，热数据10亿

pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 64 -j 64 -T 120 -D
scale=10000 -D range=1000000000

pg.x8.xlarge.2，总数据量10亿，热数据1亿
```

```
pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D
scale=10000 -D range=100000000

pg.x8.xlarge.2，总数据量10亿，热数据5亿

pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D
scale=10000 -D range=500000000

pg.x8.xlarge.2，总数据量10亿，热数据10亿

pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 32 -j 32 -T 120 -D
scale=10000 -D range=1000000000

pg.x8.large.2，总数据量5亿，热数据1亿

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 16 -j 16 -T 120 -D
scale=5000 -D range=100000000

pg.x8.large.2，总数据量5亿，热数据5亿

pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 16 -j 16 -T 120 -D
scale=5000 -D range=500000000

pg.x8.medium.2，总数据量1亿，热数据5000万

pgbench -M prepared -v -r -P 1 -f ./rw.sql -c 8 -j 8 -T 120 -D
scale=1000 -D range=50000000

pg.x8.medium.2，总数据量1亿，热数据1亿

pgbench -M prepared -n -r -P 1 -f ./rw.sql -c 8 -j 8 -T 120 -D
scale=1000 -D range=100000000
```

说明:

- scale乘以10万：表示测试数据量。
- range：表示活跃数据量。
- -c：表示测试连接数，测试连接数不代表该规格的最大连接数，最大连接数请参考实例规格表。

3.2.4 测试指标

只读QPS

数据库只读时每秒执行的SQL数（仅包含select）。

读写QPS

数据库读写时每秒执行的SQL数（含insert、select、update）。

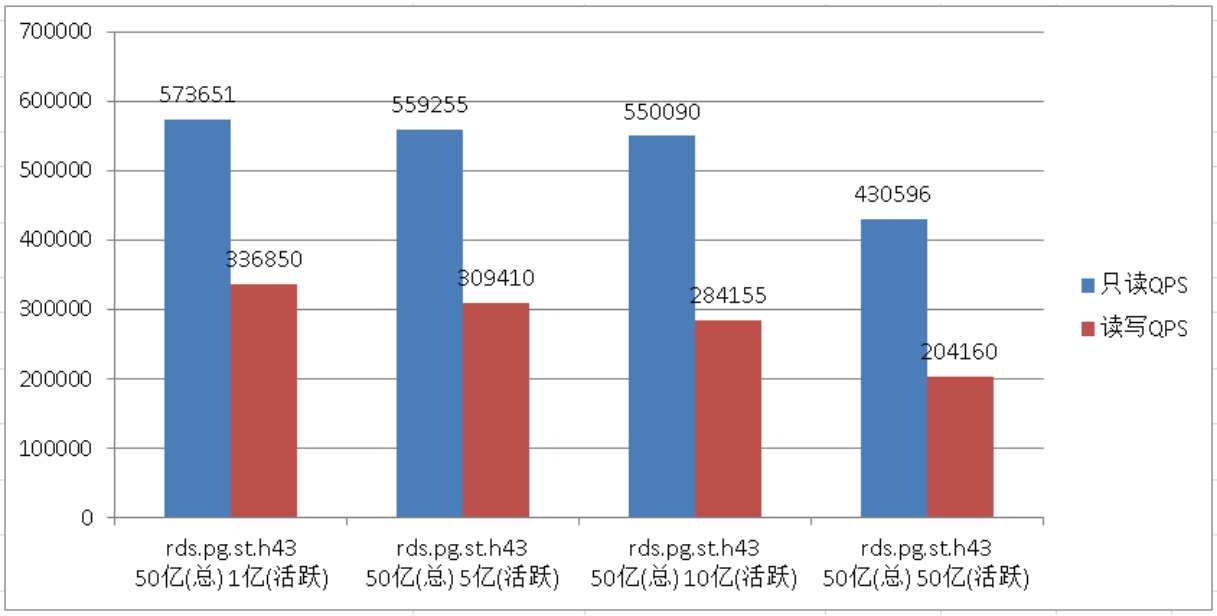
3.3 测试结果

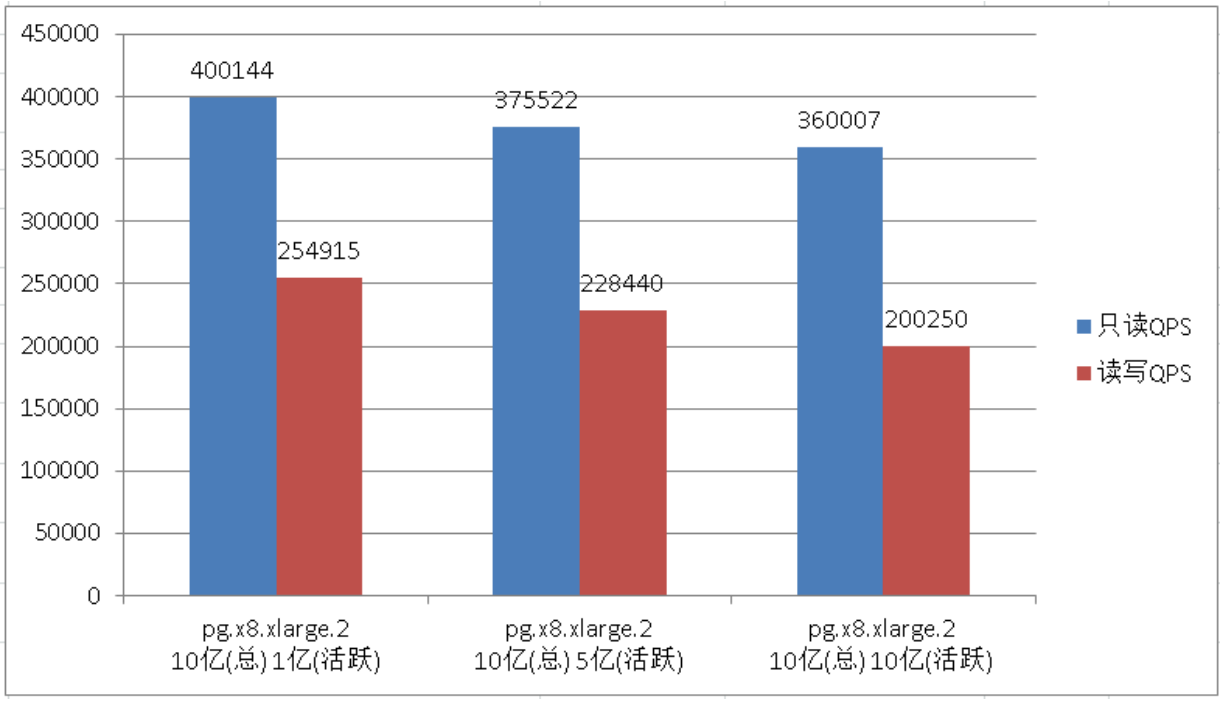
规格	预计可存储数据量	测试数据量	热(活跃)数据量	只读QPS	读写QPS
rds.pg.st.h43 60核470G3T	150亿	50亿	1亿	573651	336850
rds.pg.st.h43 60核470G3T	150亿	50亿	5亿	559255	309410
rds.pg.st.h43 60核470G3T	150亿	50亿	10亿	550090	284155
rds.pg.st.h43 60核470G3T	150亿	50亿	50亿	430596	204160
pg.x4.4xlarge.2 32核128G2T	100亿	10亿	1亿	400144	254915
pg.x4.4xlarge.2 32核128G 2T	100亿	10亿	5亿	375522	228440
pg.x4.4xlarge.2 32核128G 2T	100亿	10亿	10亿	360007	200250
pg.x8.2xlarge.2 16核128G2T	100亿	10亿	1亿	235592	156330
pg.x8.2xlarge.2 16核128G2T	100亿	10亿	5亿	221153	133125
pg.x8.2xlarge.2 16核128G2T	100亿	10亿	10亿	211268	115915
pg.x8.xlarge.2 8核64G1T	50亿	10亿	1亿	129323	71820

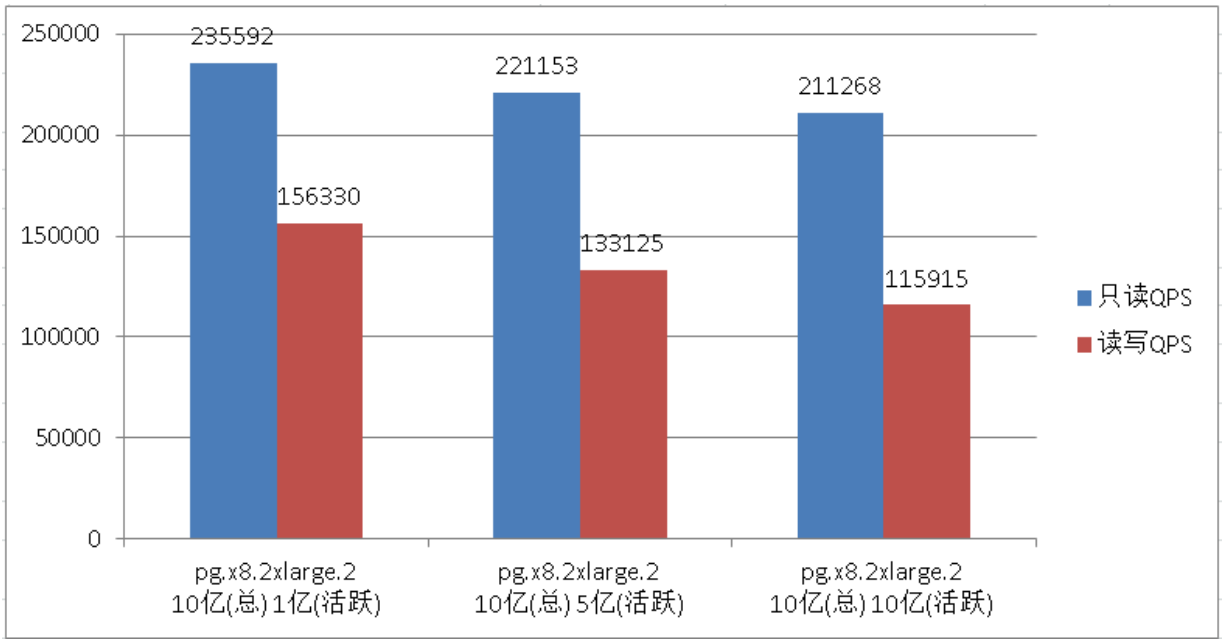
规格	预计可存储数据量	测试数据量	热(活跃)数据量	只读QPS	读写QPS
pg.x8.xlarge.2 8核64G1T	50亿	10亿	5亿	115498	58140
pg.x8.xlarge.2 8核64G1T	50亿	10亿	10亿	102735	58555
pg.x8.large.2 4核32G500G	25亿	5亿	1亿	75729	45110
pg.x8.large.2 4核32G500G	25亿	5亿	5亿	63818	36375
pg.x8.medium.2 2核16G250G	12.5亿	1亿	5000万	37245	24520
pg.x8.medium.2 2核16G250G	12.5亿	1亿	1亿	35321	19880

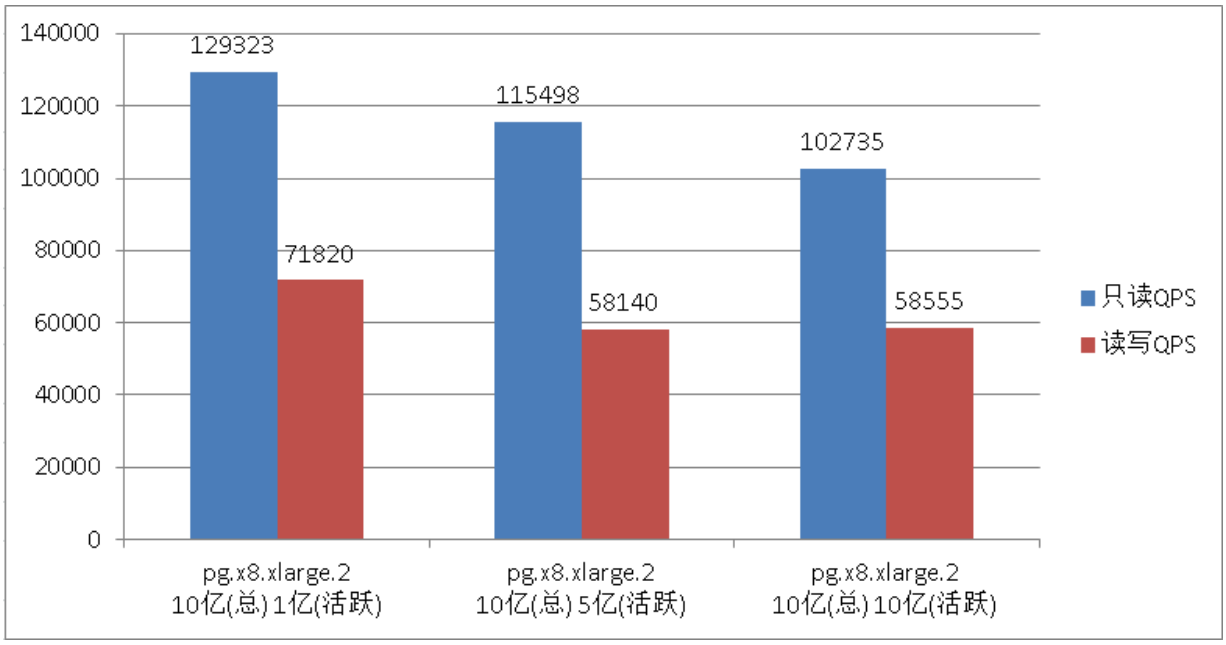
说明:

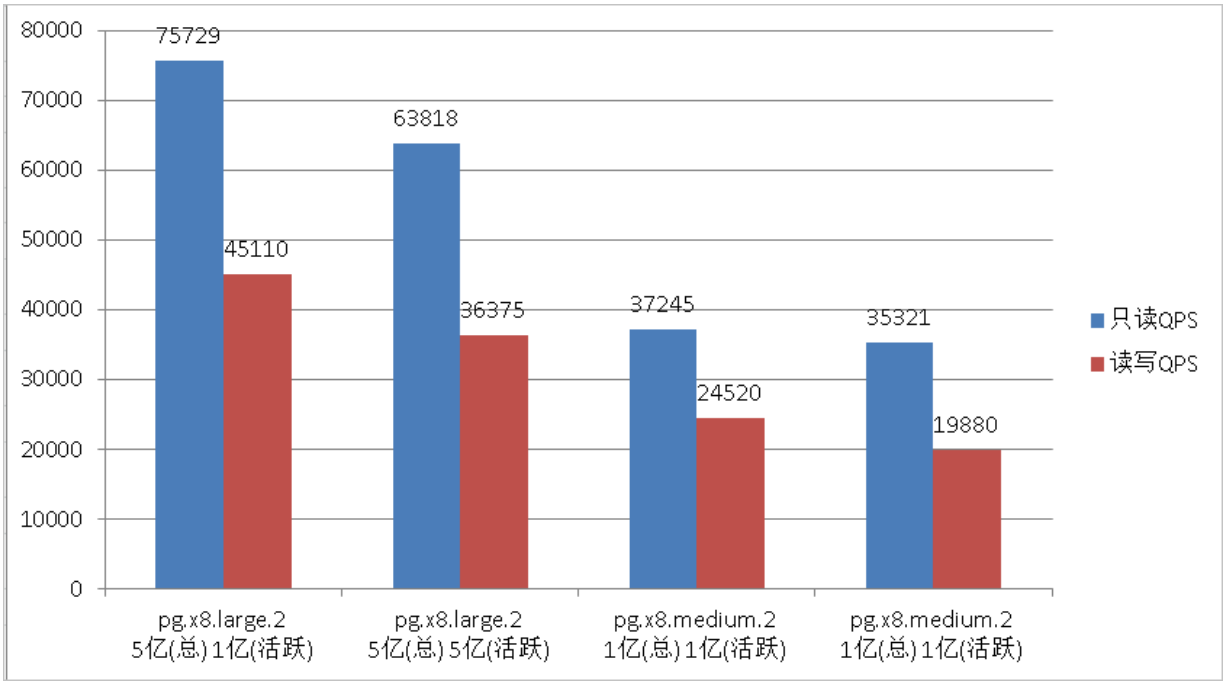
- 规格：RDS for PostgreSQL的规格代码。
- 预计可存储数据量：预计该规格实例可以存储多少条记录。
- 测试数据量：本轮测试数据的记录数。
- 热（活跃）数据量：本轮测试的查询、更新SQL的记录数范围。
- 只读QPS：只读测试的结果，表示每秒请求数。
- 读写QPS：读写测试的结果，表示每秒请求数。


700000
600000
500000
400000
300000
200000
100000
0
573651
336850
559255
309410
550090
284155
430596
204160
只读QPS
读写QPS
rds.pg.st.h43
50亿(总) 1亿(活跃)
rds.pg.st.h43
50亿(总) 5亿(活跃)
rds.pg.st.h43
50亿(总) 10亿(活跃)
rds.pg.st.h43
50亿(总) 50亿(活跃)


450000
400000
350000
300000
250000
200000
150000
100000
50000
0
400144
254915
375522
228440
360007
200250
只读QPS
读写QPS
pg.x8.xlarge.2
10亿(总) 1亿(活跃)
pg.x8.xlarge.2
10亿(总) 5亿(活跃)
pg.x8.xlarge.2
10亿(总) 10亿(活跃)


250000
200000
150000
100000
50000
0
235592
156330
221153
133125
211268
115915
只读QPS
读写QPS
pg.x8.2xlarge.2
10亿(总.) 1亿(活跃)
pg.x8.2xlarge.2
10亿(总.) 5亿(活跃)
pg.x8.2xlarge.2
10亿(总.) 10亿(活跃)


140000
120000
100000
80000
60000
40000
20000
0
129323
71820
115498
58140
102735
58555
只读QPS
读写QPS
pg.x8.xlarge.2
10亿(总.) 1亿(活跃)
pg.x8.xlarge.2
10亿(总.) 5亿(活跃)
pg.x8.xlarge.2
10亿(总.) 10亿(活跃)


80000
70000
60000
50000
40000
30000
20000
10000
0
75729
45110
63818
36375
37245
24520
35321
19880
pg.x8.large.2
5亿(总) 1亿(活跃)
pg.x8.large.2
5亿(总) 5亿(活跃)
pg.x8.medium.2
1亿(总) 1亿(活跃)
pg.x8.medium.2
1亿(总) 1亿(活跃)
只读QPS
读写QPS