

# 云数据库 MySQL

## 白皮书

## 产品文档



腾讯云

**【 版权声明 】**

©2013–2022 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

**【 商标声明 】**

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

**【 服务声明 】**

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

**【 联系我们 】**

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100。

# 文档目录

## 白皮书

### 性能白皮书

性能概述

测试方法

测试环境

测试工具

测试方法

测试指标

测试结果

MySQL 8.0 测试结果

MySQL 5.7 测试结果

MySQL 5.6 测试结果

### 安全白皮书

概述

攻击防护

访问控制

网络隔离

数据存储加密

备份恢复

实例容灾

审计和治理

数据安全

数据库巡检

数据销毁

版本升级

# 白皮书

## 性能白皮书

### 性能概述

最近更新时间：2022-02-23 09:08:37

云数据库 MySQL (TencentDB for MySQL) 是腾讯云基于开源数据库 MySQL 专业打造的高性能分布式数据存储服务，让用户能够在云中更轻松地设置、操作和扩展关系数据库。云数据库 MySQL 使用了自研内核 TXSQL 以提供高性能以及高稳定性，自研内核具备众多核心特性，如企业级透明数据加密、审计、线程池等，大幅提升了云数据库的性能和稳定性。

腾讯云数据库 MySQL 经专业团队不断测试和优化，具备灵活、高效的事务处理能力以及先进和完备的合规安全防护和数据加密能力，大大保证产品的安全性和业务能力。

云数据库 MySQL 的性能测试结果请参见：

- [MySQL 8.0测试结果](#)
- [MySQL 5.7测试结果](#)
- [MySQL 5.6测试结果](#)

# 测试方法

## 测试环境

最近更新时间：2022-02-23 09:08:47

本文介绍云数据库 MySQL 性能测试所使用的环境。

- 地域/可用区：北京 - 北京七区
- 客户端：S5.8XLARGE64(标准型S5，32核64GB)
- 客户端操作系统：CentOS 8.2 64位
- 网络：云服务器 CVM 和云数据库 MySQL 实例网络类型均为私有网络（VPC）且在同一子网下

测试的云数据库 MySQL 实例信息如下：

- 存储类型：本地 SSD 盘
- 实例规格：通用型
- 参数模板：高性能模板

# 测试工具

最近更新时间：2022-02-23 09:08:51

本文为您介绍云数据库 MySQL 性能测试工具 SysBench，以及如何在云服务器 CVM 实例上安装 SysBench。

## SysBench 工具介绍

SysBench 是一个跨平台且支持多线程的模块化基准测试工具，用于评估系统在运行高负载的数据库时相关核心参数的性能表现。可绕过复杂的数据库基准设置，甚至在没有安装数据库的前提下，快速了解数据库系统的性能。

## SysBench 测试模型

- SysBench 标准 OLTP 读写混合场景中一个事务包含18个读写 SQL。
- SysBench 标准 OLTP 只读场景中一个事务包含14个读 SQL（10条主键点查询、4条范围查询）。
- SysBench 标准 OLTP 只写场景中一个事务包含4个写 SQL（2条 UPDATE、1条 DELETED、1条 INSERT）。

## SysBench 参数说明

| 参数             | 说明              |
|----------------|-----------------|
| db-driver      | 数据库引擎           |
| mysql-host     | MySQL 实例连接地址    |
| mysql-port     | MySQL 实例连接端口    |
| mysql-user     | MySQL 实例帐号      |
| mysql-password | MySQL 实例帐号对应的密码 |
| mysql-db       | MySQL 实例数据库名    |
| table_size     | 测试表大小           |
| tables         | 测试表数量           |
| events         | 测试请求数量          |
| time           | 测试时间            |

| 参数              | 说明  |
|-----------------|---|
| threads         | 测试线程数                                     |
| percentile      | 需要统计的百分比，默认值为95%，即请求在95%的情况下的执行时间         |
| report-interval | 表示 N 秒输出一次测试进度报告，0表示关闭测试进度报告输出，仅输出最终的报告结果 |
| skip-trx        | 是否跳过事务。<br>1: 跳过<br>0: 不跳过                |

## 安装方法

本压测使用SysBench 1.0.20版本。更多信息，请参见 [Sysbench 官方文档](#)。

1. 在 CVM 实例执行如下命令安装 SysBench。

```
yum install gcc gcc-c++ autoconf automake make libtool bzip2 mysql-devel git mysql
git clone https://github.com/akopytov/sysbench.git
##从 Git 中下载 SysBench
cd sysbench
##打开 SysBench 目录
git checkout 1.0.20
##切换到 SysBench 1.0.20 版本
./autogen.sh
##运行 autogen.sh
./configure --prefix=/usr --mandir=/usr/share/man
make
##编译
make install
```

2. 执行如下命令配置客户端，使内核可以使用所有的 CPU 处理数据包，同时减少 CPU 之间的上下文切换。

```
sudo sh -c 'for x in /sys/class/net/eth0/queues/rx-*; do echo ffffffff>$x/rps_cpus; done'
sudo sh -c "echo 32768 > /proc/sys/net/core/rps_sock_flow_entries"
sudo sh -c "echo 4096 > /sys/class/net/eth0/queues/rx-0/rps_flow_cnt"
sudo sh -c "echo 4096 > /sys/class/net/eth0/queues/rx-1/rps_flow_cnt"
```

② 说明：

fffffff表示使用32个 CPU（1个f表示4个 CPU）。



# 测试方法

最近更新时间：2022-02-23 09:08:56

本文为您介绍云数据库 MySQL 性能测试的方法。

## 操作步骤

使用 SysBench 测试云数据库 MySQL 实例的读写混合性能。

### 1. 准备数据。

```
sysbench --db-driver=mysql --mysql-host=XXX --mysql-port=XXX --mysql-user=XXX --mysql-  
password=XXX --mysql-db=sbtest --table_size=25000 --tables=250 --events=0 --time=600  
oltp_read_write prepare
```

### 2. 运行 workload。

```
sysbench --db-driver=mysql --mysql-host=XXX --mysql-port=XXX --mysql-user=XXX --mysql-  
password=XXX --mysql-db=sbtest --table_size=25000 --tables=250 --events=0 --time=600 -  
-threads=XXX --percentile=95 --report-interval=1 oltp_read_write run
```

### 3. 清理数据。

```
sysbench --db-driver=mysql --mysql-host=XXX --mysql-port=XXX --mysql-user=XXX --mysql-  
password=XXX --mysql-db=sbtest --table_size=25000 --tables=250 --events=0 --time=600 -  
-threads=XXX --percentile=95 oltp_read_write cleanup
```

# 测试指标

最近更新时间：2022-03-04 16:11:39

本文为您介绍云数据库 MySQL 性能测试的测试指标。

## 测试指标

- **每秒执行事务数 TPS (Transactions Per Second)**  
数据库每秒执行的事务数，以 COMMIT 成功次数为准。
- **每秒执行请求数 QPS (Queries Per Second)**  
数据库每秒执行的 SQL 数，包含 INSERT、SELECT、UPDATE、DETELE、COMMIT 等。
- **所有 event 平均耗时 avg\_lat (Average Latency)**

# 测试结果

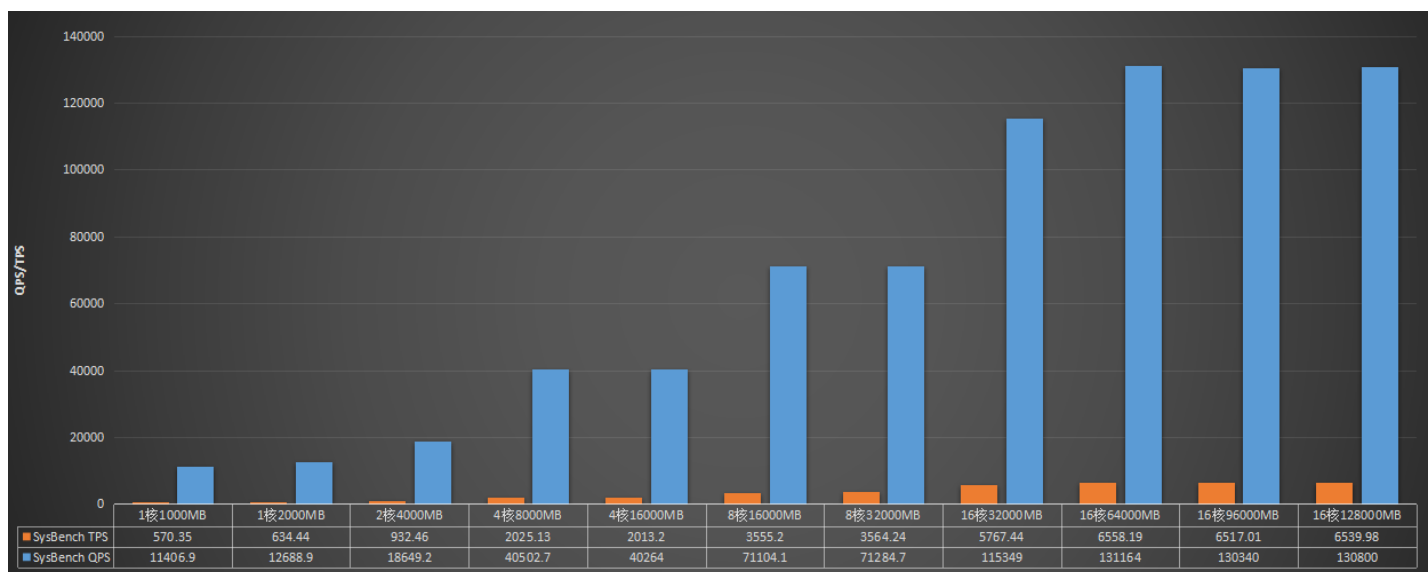
## MySQL 8.0 测试结果

最近更新时间：2022-02-23 09:10:35

本文为您介绍云数据库 MySQL 8.0 通用型实例的性能测试结果。

### 场景一：全缓存

全缓存指只有全部数据可以放到缓存里，查询过程中不需要读写磁盘更新缓存。

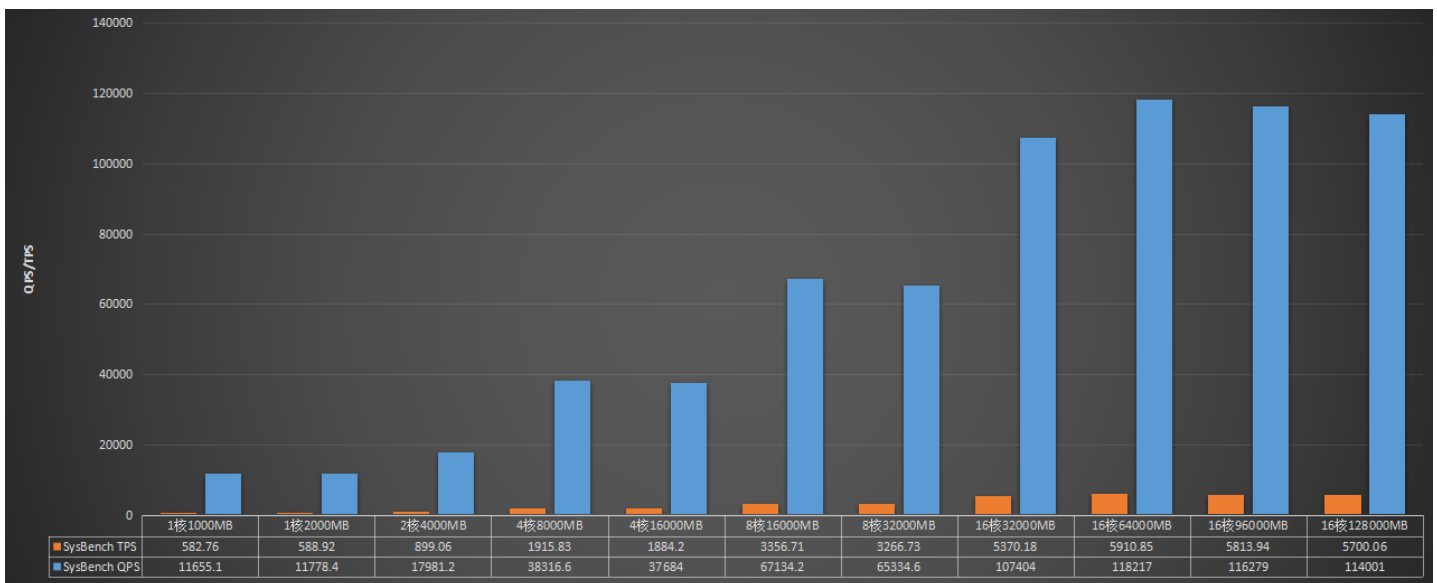


| CPU (core) | 内存 (MB) | 并发度 | 单表数据量 | 表总数 | SysBench TPS | SysBench QPS | avg_lat |
|------------|---------|-----|-------|-----|--------------|--------------|---------|
| 1          | 1000    | 8   | 25000 | 150 | 570.35       | 11406.9      | 14.03   |
| 1          | 2000    | 8   | 25000 | 150 | 634.44       | 12688.9      | 12.61   |
| 2          | 4000    | 16  | 25000 | 150 | 932.46       | 18649.2      | 17.16   |
| 4          | 8000    | 32  | 25000 | 150 | 2025.13      | 40502.7      | 15.8    |
| 4          | 16000   | 32  | 25000 | 150 | 2013.2       | 40264        | 15.89   |
| 8          | 16000   | 64  | 25000 | 150 | 3555.2       | 71104.1      | 18      |
| 8          | 32000   | 64  | 25000 | 150 | 3564.24      | 71284.7      | 17.95   |
| 16         | 32000   | 128 | 25000 | 150 | 5767.44      | 115349       | 22.19   |

| CPU (core) | 内存 (MB) | 并发度 | 单表数据量 | 表总数 | SysBench TPS | SysBench QPS | avg_lat |
|------------|---------|-----|-------|-----|--------------|--------------|---------|
| 16         | 64000   | 128 | 25000 | 150 | 6558.19      | 131164       | 19.51   |
| 16         | 96000   | 128 | 25000 | 150 | 6517.01      | 130340       | 19.63   |
| 16         | 128000  | 128 | 25000 | 150 | 6539.98      | 130800       | 19.57   |

## 场景二：磁盘 IO 型

磁盘 IO 型场景指只有部分数据可以放到缓存里，查询过程中需要读写磁盘更新缓存。



| CPU (core) | 内存 (MB) | 并发度 | 单表数据量   | 表总数 | SysBench TPS | SysBench QPS | avg_lat |
|------------|---------|-----|---------|-----|--------------|--------------|---------|
| 1          | 1000    | 8   | 800000  | 6   | 582.76       | 11655.1      | 13.73   |
| 1          | 2000    | 8   | 800000  | 12  | 588.92       | 11778.4      | 13.58   |
| 2          | 4000    | 16  | 800000  | 24  | 899.06       | 17981.2      | 17.8    |
| 4          | 8000    | 32  | 800000  | 48  | 1915.83      | 38316.6      | 16.7    |
| 4          | 16000   | 32  | 6000000 | 13  | 1884.2       | 37684        | 16.98   |
| 8          | 16000   | 64  | 6000000 | 13  | 3356.71      | 67134.2      | 19.06   |
| 8          | 32000   | 64  | 6000000 | 25  | 3266.73      | 65334.6      | 19.59   |
| 16         | 32000   | 128 | 6000000 | 25  | 5370.18      | 107404       | 23.83   |

| CPU (core) | 内存 (MB) | 并发度 | 单表数据量   | 表总数 | SysBench TPS | SysBench QPS | avg_lat |
|------------|---------|-----|---------|-----|--------------|--------------|---------|
| 16         | 64000   | 128 | 6000000 | 49  | 5910.85      | 118217       | 21.65   |
| 16         | 96000   | 128 | 6000000 | 74  | 5813.94      | 116279       | 22.01   |
| 16         | 128000  | 128 | 6000000 | 98  | 5700.06      | 114001       | 22.45   |

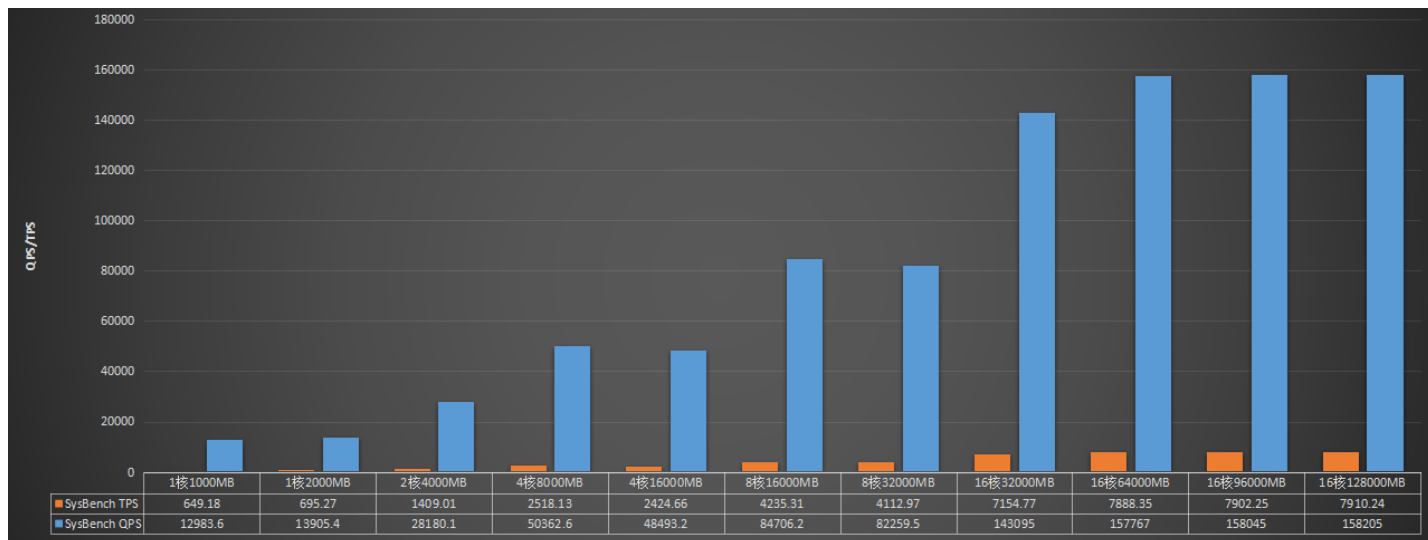
# MySQL 5.7 测试结果

最近更新时间：2022-02-23 09:10:40

本文为您介绍云数据库 MySQL 5.7 通用型实例的性能测试结果。

## 场景一：全缓存

全缓存指只有全部数据可以放到缓存里，查询过程中不需要读写磁盘更新缓存。

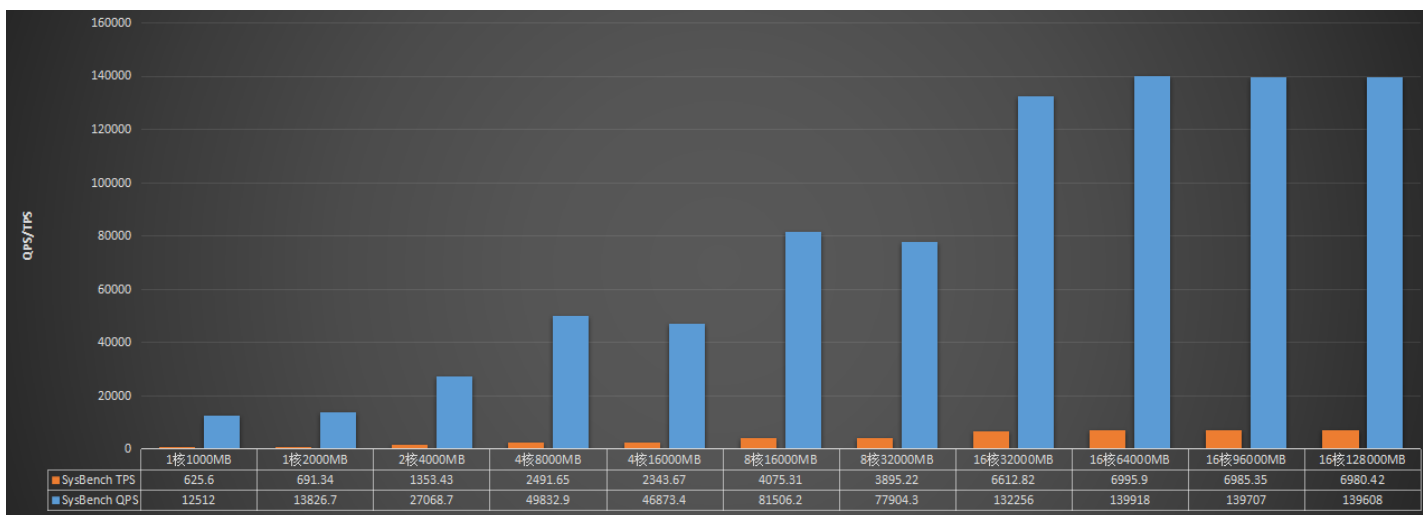


| CPU (core) | 内存 (MB) | 并发度 | 单表数据量 | 表总数 | SysBench TPS | SysBench QPS | avg_lat |
|------------|---------|-----|-------|-----|--------------|--------------|---------|
| 1          | 1000    | 8   | 25000 | 150 | 649.18       | 12983.6      | 12.32   |
| 1          | 2000    | 8   | 25000 | 150 | 695.27       | 13905.4      | 11.51   |
| 2          | 4000    | 16  | 25000 | 150 | 1409.01      | 28180.1      | 11.35   |
| 4          | 8000    | 32  | 25000 | 150 | 2518.13      | 50362.6      | 12.71   |
| 4          | 16000   | 32  | 25000 | 150 | 2424.66      | 48493.2      | 13.2    |
| 8          | 16000   | 64  | 25000 | 150 | 4235.31      | 84706.2      | 15.11   |
| 8          | 32000   | 64  | 25000 | 150 | 4112.97      | 82259.5      | 15.56   |
| 16         | 32000   | 128 | 25000 | 150 | 7154.77      | 143095       | 17.88   |
| 16         | 64000   | 128 | 25000 | 150 | 7888.35      | 157767       | 16.22   |
| 16         | 96000   | 128 | 25000 | 150 | 7902.25      | 158045       | 16.19   |

| CPU (core) | 内存 (MB) | 并发度 | 单表数据量 | 表总数 | SysBench TPS | SysBench QPS | avg_lat |
|------------|---------|-----|-------|-----|--------------|--------------|---------|
| 16         | 128000  | 128 | 25000 | 150 | 7910.24      | 158205       | 16.17   |

## 场景二：磁盘 IO 型

磁盘 IO 型场景指只有部分数据可以放到缓存里，查询过程中需要读写磁盘更新缓存。



| CPU (core) | 内存 (MB) | 并发度 | 单表数据量   | 表总数 | SysBench TPS | SysBench QPS | avg_lat |
|------------|---------|-----|---------|-----|--------------|--------------|---------|
| 1          | 1000    | 8   | 800000  | 6   | 625.6        | 12512        | 12.79   |
| 1          | 2000    | 8   | 800000  | 12  | 691.34       | 13826.7      | 11.57   |
| 2          | 4000    | 16  | 800000  | 24  | 1353.43      | 27068.7      | 11.82   |
| 4          | 8000    | 32  | 800000  | 48  | 2491.65      | 49832.9      | 12.84   |
| 4          | 16000   | 32  | 6000000 | 13  | 2343.67      | 46873.4      | 13.65   |
| 8          | 16000   | 64  | 6000000 | 13  | 4075.31      | 81506.2      | 15.7    |
| 8          | 32000   | 64  | 6000000 | 25  | 3895.22      | 77904.3      | 16.43   |
| 16         | 32000   | 128 | 6000000 | 25  | 6612.82      | 132256       | 19.35   |
| 16         | 64000   | 128 | 6000000 | 49  | 6995.9       | 139918       | 18.29   |
| 16         | 96000   | 128 | 6000000 | 74  | 6985.35      | 139707       | 18.32   |
| 16         | 128000  | 128 | 6000000 | 98  | 6980.42      | 139608       | 18.33   |

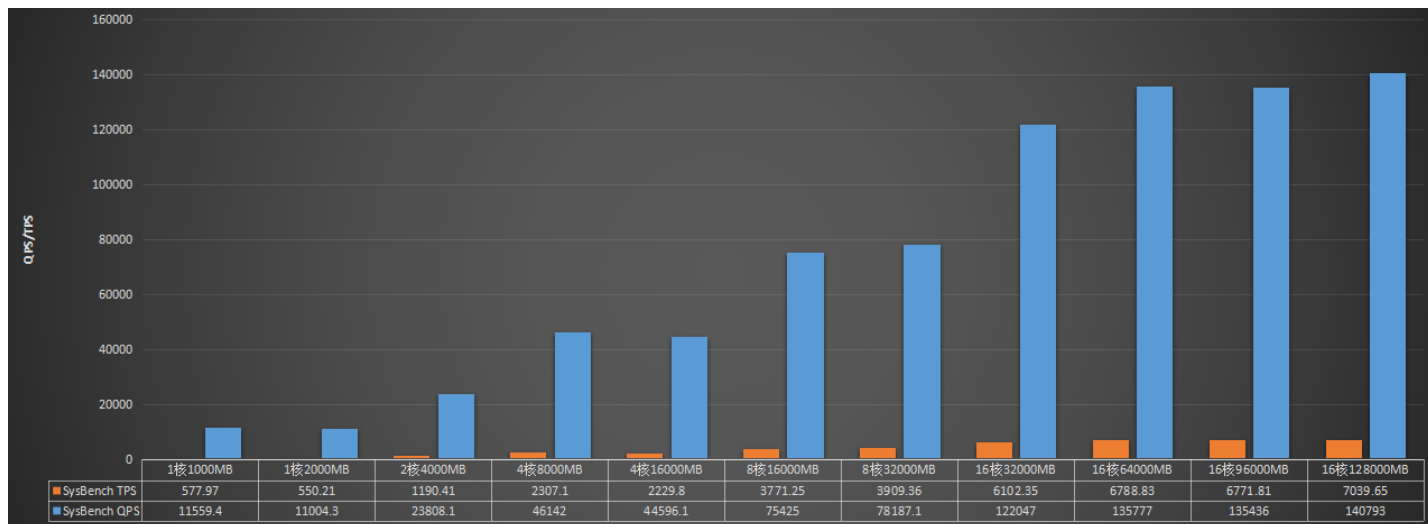
# MySQL 5.6 测试结果

最近更新时间：2022-02-23 09:10:45

本文为您介绍云数据库 MySQL 5.6 通用型实例的性能测试结果。

## 场景一：全缓存

全缓存指只有全部数据可以放到缓存里，查询过程中不需要读写磁盘更新缓存。



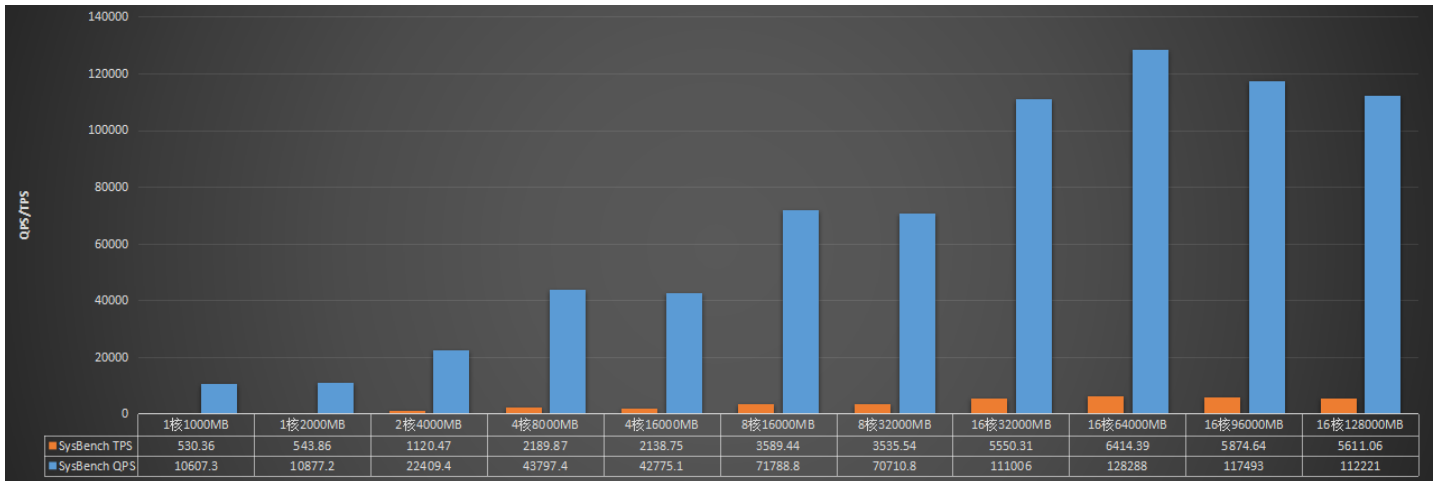
| CPU (core) | 内存 (MB) | 并发度 | 单表数据量 | 表总数 | SysBench TPS | SysBench QPS | avg_lat |
|------------|---------|-----|-------|-----|--------------|--------------|---------|
| 1          | 1000    | 8   | 25000 | 150 | 577.97       | 11559.4      | 13.84   |
| 1          | 2000    | 8   | 25000 | 150 | 550.21       | 11004.3      | 14.54   |
| 2          | 4000    | 16  | 25000 | 150 | 1190.41      | 23808.1      | 13.44   |
| 4          | 8000    | 32  | 25000 | 150 | 2307.1       | 46142        | 13.87   |
| 4          | 16000   | 32  | 25000 | 150 | 2229.8       | 44596.1      | 14.35   |
| 8          | 16000   | 64  | 25000 | 150 | 3771.25      | 75425        | 16.97   |
| 8          | 32000   | 64  | 25000 | 150 | 3909.36      | 78187.1      | 16.37   |
| 16         | 32000   | 128 | 25000 | 150 | 6102.35      | 122047       | 20.97   |
| 16         | 64000   | 128 | 25000 | 150 | 6788.83      | 135777       | 18.85   |
| 16         | 96000   | 128 | 25000 | 150 | 6771.81      | 135436       | 18.9    |



| CPU (core) | 内存 (MB) | 并发度 | 单表数据量 | 表总数 | SysBench TPS | SysBench QPS | avg_lat |
|------------|---------|-----|-------|-----|--------------|--------------|---------|
| 16         | 128000  | 128 | 25000 | 150 | 7039.65      | 140793       | 18.18   |

## 场景二：磁盘 IO 型

磁盘 IO 型场景指只有部分数据可以放到缓存里，查询过程中需要读写磁盘更新缓存。



| CPU (core) | 内存 (MB) | 并发度 | 单表数据量  | 表总数 | SysBench TPS | SysBench QPS | avg_lat |
|------------|---------|-----|--------|-----|--------------|--------------|---------|
| 1          | 1000    | 8   | 800000 | 6   | 530.36       | 10607.3      | 15.08   |
| 1          | 2000    | 8   | 800000 | 12  | 543.86       | 10877.2      | 14.71   |
| 2          | 4000    | 16  | 800000 | 24  | 1120.47      | 22409.4      | 14.28   |
| 4          | 8000    | 32  | 800000 | 48  | 2189.87      | 43797.4      | 14.61   |
| 4          | 16000   | 32  | 600000 | 13  | 2138.75      | 42775.1      | 14.96   |
| 8          | 16000   | 64  | 600000 | 13  | 3589.44      | 71788.8      | 17.83   |
| 8          | 32000   | 64  | 600000 | 25  | 3535.54      | 70710.8      | 18.1    |
| 16         | 32000   | 128 | 600000 | 25  | 5550.31      | 111006       | 23.06   |
| 16         | 64000   | 128 | 600000 | 49  | 6414.39      | 128288       | 19.95   |
| 16         | 96000   | 128 | 600000 | 74  | 5874.64      | 117493       | 21.78   |
| 16         | 128000  | 128 | 600000 | 98  | 5611.06      | 112221       | 22.81   |

# 安全白皮书

## 概述

最近更新时间：2022-01-20 15:27:33

腾讯云数据库 MySQL (TencentDB for MySQL) 让用户可以轻松在云端部署、使用 MySQL 数据库。MySQL 是世界上最流行的开源关系数据库，通过云数据库 MySQL，您在几分钟内即可部署可扩展的 MySQL 数据库实例。不仅经济实惠，而且可以弹性调整硬件容量的大小而无需停机。云数据库 MySQL 提供备份、回档、监控、快速扩容、数据传输等数据库运维全套解决方案，为您简化 IT 运维工作，让您能更加专注于业务发展。

云数据库 MySQL 提供多样化的安全加固功能，来保证用户数据的可靠性和安全性，为了保障您 MySQL 数据库具备更高的安全能力，建议您根据业务需要配合使用如下安全功能：

- 网络：[安全组](#)、[私有网络](#)。
- 存储：[数据加密](#)、[自动备份](#)。
- 容灾：[同城容灾](#)、[异地容灾](#)。
- 审计：[合规审计](#)。
- 权限：[访问控制](#)。
- 安全：[数据安全](#)、[数据库巡检](#)。

# 攻击防护

最近更新时间：2022-04-18 16:03:51

## 防 DDoS 攻击

当用户使用外网连接和访问 MySQL 实例时，可能会遭受 DDoS 攻击，腾讯云提供流量清洗和封堵处理功能，完全由系统自动触发和结束。当腾讯云大禹系统认为用户实例正在遭受 DDoS 攻击时，首先会启动流量清洗功能，如果流量清洗无法抵御攻击或者攻击达到封堵阈值，则会进行封堵处理。

### ⚠ 注意：

建议用户通过内网访问 MySQL 实例，可以避免 MySQL 实例受到 DDoS 攻击。

### 流量清洗

当 MySQL 实例公网网络流量超过设定的防护阈值时，腾讯云大禹系统将自动对该 MySQL 实例的公网入流量进行清洗。通过策略路由将流量从原始网络路径中重定向到大禹系统的 DDoS 清洗设备上，通过清洗设备对该 MySQL 实例的外网流量进行识别，丢弃攻击流量，将正常流量转发至 MySQL 实例。

### 封堵处理

当 MySQL 实例受到的攻击流量超过其封堵处理阈值时，腾讯云将通过运营商的服务屏蔽该 MySQL 实例所有外网访问，保护云平台其他用户免受影响。即当您的某个 MySQL 实例受到的攻击流量超过您所享受的最大防护峰值时，腾讯云将屏蔽该 MySQL 实例的所有外网访问。

- 封堵触发条件如下：
  - BPS ( Bits Per Second ) 达到2Gbps。
  - 流量清洗无效。
- 封堵结束条件如下：
  - 封堵在2小时后自动解除。

# 访问控制

最近更新时间：2021-04-19 10:19:56

云数据库 MySQL 通过数据库帐号管理、密码管理、访问管理、安全组等维度进行访问控制，进而保障 MySQL 的数据安全性。

## 数据库帐号管理

云数据库 MySQL 支持控制台或 API 来 [创建数据库帐号](#)，还可以为其数据库帐号授予不同粒度的管理权限，建议您采用权限最小化的授权原则，进而保证数据库的数据安全。

## 密码管理

云数据库 MySQL 支持控制台或 API 来 [重置数据库帐号密码](#)，数据库密码规格需要8 - 64个字符，至少包含英文、数字和符号 `_+&=!@#$$%^*()` 中的2种。

重置密码功能已纳入 [访问管理 CAM](#) 权限管理，建议对重置密码接口或云数据库 MySQL 实例敏感资源权限收紧，只授权给应该授权的人员。为了数据安全，建议您定期更换密码，最长间隔不超过3个月。

## 访问管理

[访问管理 CAM](#) 主要用于帮助用户安全管理腾讯云账户下资源的访问权限，通过 CAM，您可以创建、管理和销毁用户（组），并通过身份管理和策略管理控制指定用户可以使用的腾讯云资源，进而达到权限分离的目的。

## 安全组

[安全组](#) 主要用于帮助用户实现 MySQL 网络安全访问控制，安全组是一种有状态的包含过滤功能的虚拟防火墙，用于设置单台或多台云数据库的网络访问控制，是腾讯云提供的重要的网络安全隔离手段。

安全组是一个逻辑上的分组，您可以将同一地域内具有相同网络安全隔离需求的云数据库实例加到同一个安全组内。在安全组内基于规则匹配，更改安全组规则不需要重启 MySQL 实例，修改完安全组规则后立即生效。

# 网络隔离

最近更新时间：2021-04-19 10:19:41

云数据库 MySQL 支持使用私有网络来实现更程度的网络隔离控制，搭配使用 [安全组](#) 和 [私有网络](#) 将极大提升访问 MySQL 实例的安全性。

私有网络是用户在腾讯云上建立的一块逻辑隔离的网络空间。在私有网络内，用户可以自由定义网段划分、IP 地址和路由策略，进而实现资源级的网络隔离。

部署在私有网络中的 MySQL 实例默认只能被同一个私有网络中的 CVM 访问，若 CVM 与 MySQL 实例不在同一个私有网络，也可以通过申请外网的方式进行访问，考虑到网络安全的问题，不建议采用外网的方式进行数据库访问，若必须采用外网访问 MySQL 实例，请配合安全组来实现客户端的访问控制。

# 数据存储加密

最近更新时间：2021-04-19 10:19:20

云数据库 MySQL 支持 [透明数据加密功能](#)，其由腾讯云数据库团队自研。透明加密是指数据的加解密操作对用户透明。用户在创建加密表时，不用指定加密密钥，数据在写盘时加密，在读盘时解密。

透明数据加密采用国际流行的 AES 算法，密钥长度为256比特。加密密钥由腾讯云 [密钥管理系统 KMS](#) 管理，访问 KMS 服务同时需要取得用户授权，您还可以通过 KMS 控制台对密钥进行更换，进一步提升系统的安全性。

# 备份恢复

最近更新时间：2022-04-18 15:55:17

## 备份

云数据库 MySQL 支持自动备份和手动备份来保障数据可恢复性，进而保证数据的完整性和可靠性。MySQL 默认提供数据备份和日志备份功能，其中自动备份的备份周期设置一周不能少于2次，若有其他备份需求，可通过 [控制台](#) 或 API 随时发起手动备份。

另外可根据业务需要，灵活配置备份文件的保留周期，默认保留时长为7天且最大可设置为1830天，超过备份保留时长的备份文件过期自动删除。

功能使用请参见 [备份数据库](#)。

## 恢复

云数据库 MySQL 提供恢复数据的能力，您可根据业务需要通过回档功能进行数据恢复，支持数据恢复到备份保留期内的任意时间点，其中可恢复的时间点取决于备份保留时长，因此，请根据业务需要合理配置备份保留策略，进而保证业务数据的可恢复性。

功能使用请参见 [回档数据库](#)。

# 实例容灾

最近更新时间：2022-01-20 15:26:57

针对业务连续服务、数据可靠性有强需求或是监管需要的场景，云数据库 MySQL 提供跨可用区、跨地域的容灾解决方案，帮助用户以较低成本提升业务连续服务的能力，同时提升数据的可靠性。

## 同城容灾

云数据库 MySQL [双节点](#) 和 [三节点](#) 支持创建多可用区实例，多可用区实例将物理服务器部署在同一地域的不同可用区，当一个可用区故障时，业务流量可以在短时间内快速切换到另一个可用区，整个切换过程对业务透明，应用层面无需变更，进而实现同城容灾的能力。

### 说明：

多可用区实例处于不同可用区，可能会增加2ms - 3ms的同步网络延迟。

同城容灾请参见 [MySQL 多可用区部署](#)。

## 异地容灾

云数据库 MySQL 同城容灾能力局限于同地域的不同可用区之间，为了提供更高的可用性，MySQL 还支持跨地域的数据容灾。

您可以将地域 A 的 MySQL 实例通过数据传输异步复制到地域 B 的 MySQL 实例，其中灾备实例拥有独立的连接地址、帐号和权限。若 A 地域发生短期不可恢复的重大故障，您随时可以进行容灾切换，只需要修改应用程序中的数据库连接配置，便可以快速地将应用请求转发到灾备实例上，进而获得金融级数据库的可用性。

异地容灾请参见 [管理灾备实例](#)。



# 审计和治理

最近更新时间：2022-01-20 15:26:32

## 合规审计

云数据库 MySQL 提供企业级的 **合规审计** 功能，其具有合规、安全、追溯等特性，符合国家相关规定，是用户通过等保合规的刚需产品。

合规审计通过对用户访问数据库行为的记录、分析，帮助用户事后生成安全合规审计报告，通过 AI 智能分析数据库操作，帮助用户及时发现风险行为，对违反安全策略的访问行为进行及时告警，保证数据库操作满足合规性要求，助力用户通过等保合规测评。

### ② 说明：

综合功能和性能两方面的分析衡量，开启审计并进行全量审计时，对系统的性能损耗不超过5%。

## 安全治理

**安全治理** 通过 AI 和大数据分析，能够从用户与事件两个维度对日常行为进行建模，对严重偏离模型的行为进行告警，对用户数据库行为的风险情况进行可视化，提升用户数据安全治理的能力，自动化帮助用户发现恶意操作、内部泄密、数据流传失控等异常问题，在用户发生安全事件时，提供安全事件追溯，对事件进行回放、追踪、追责、定责。

# 数据安全

最近更新时间：2022-01-20 15:22:11

## 敏感数据发现

**敏感数据发现** 功能可通过识别规则自动发现实例敏感数据，并对所发现的敏感数据实现自动化分类分级保护。支持多达20种内置的敏感数据识别规则，在满足合规性的基础上，能够对数据集的所有字段进行敏感属性识别，确保隐藏在大段文本中的敏感信息能够得到妥善处理。同时，产品还支持自定义敏感数据类型的自动发现，满足用户个性化的数据保护需求。

## 数据脱敏

**数据脱敏** 功能内置多种高级脱敏算法，可智能化执行与管理脱敏任务，针对不同业务场景实现数据脱敏，进而达到企业核心数据保密的效果。数据脱敏在确保安全性的同时，处理过的数据依然保持原始数据的分布特征、数据格式，使统计分析、测试、研发、培训等用途不受影响。

- 在合规性方面，数据脱敏能够通过匿名化技术，确保脱敏后数据无法被还原。保障企业个人信息使用时严格遵守相关法律法规，满足企业客户合规性需求。
- 在系统性能方面，数据脱敏在任务开始时拉取的是用户最新的实时备份数据，不需要源数据库停机，更不会给数据库性能带来额外的影响和开销，在用户无感知的情况下即可动态在线实时完成脱敏任务。

# 数据库巡检

最近更新时间：2022-01-20 15:21:35

**数据库巡检** 用于定期自动化全实例健康巡检，用户也可以根据自己的需求个性化设置巡检，帮助用户排查实例隐患并提供解决方案。数据库巡检报告中包含介绍、基本信息、健康、实例状态、异常诊断、慢 SQL 分析、高危账号、大表分析以及性能曲线等章节。

# 数据销毁

最近更新时间：2021-04-19 10:18:31

腾讯云用户在销毁 MySQL 实例时，存储在 MySQL 数据库的所有数据（包括所有备份数据）都会被销毁，腾讯云不会保留数据，更不会主动恢复用户的数据库实例。

功能使用请参见 [销毁实例](#)。

# 版本升级

最近更新时间：2021-04-19 10:18:15

云数据库 MySQL 会为您提供最新的数据库版本，当系统出现重大 Bug 或安全漏洞时，MySQL 实例会在您的维护时间内发起数据库版本的升级，并提前推送版本升级的通知，版本升级会引起秒级别的连接闪断，请确保业务具备重连机制。