图文 07 生产经验：如何对生产环境中的数据库进行360度无死角压测?

生产经验：如何对生产环境中的数据库进行360度无死角压测?

如何提问：每篇文章都有评论区，大家可以尽情留言提问，我会逐一答疑
如何加群：购买狸猫技术窝专栏的小伙伴都可以加入狸猫技术交流群，一个非常纯粹的技术交流的地方

具体加群方式，请参见目录菜单下的文档：**《MySQL专栏付费用户如何加群》**（购买后可见）

1、昨日思考题解答

先给大家分析昨天的第一个小思考题：给你一台4核8G的机器，他可以抗到每秒几千甚至每秒几万的并发请求吗?

其实这个是不一定的，因为一台机器到底每秒钟可以抗下多少并发请求，跟CPU、内存、磁盘IO、网络带宽，都是有关系的。

举个例子，之前在我们的一个项目的生产环境中，据我们观察，一台4核8G的机器如果每秒抗下500+的请求，那么他的CPU负载就已经很高了，基本上最多可能也就是去抗下每秒1000+的请求，而且那个时候CPU负载基本会打满，机器有挂掉的风险。

另外如果你的系统的业务逻辑特别的吃内存，也许你一台4核8G的机器跑到每秒几百的请求，内存使用率就很高了，而且JVM GC频率可能会非常的高，所以此时也很难继续提升并发请求了。

所以其实你一台机器是不可能无限制的让他增加可以抗下的并发请求的。

再来看下一个思考题：关于QPS和TPS的。我上次问大家了，如果一个交易系统拆分为了很多服务，那么每个服务每秒接收的并发请求是QPS还是TPS呢?

这个明显是QPS，因为每个服务就负责干自己的一些事儿，其实对他来说，每秒并发请求数量就是QPS。

那么对于多个服务组成的一个大的交易系统而言，这个交易系统每秒可以完成多少笔交易，这是QPS还是TPS呢?

其实这个你可以认为是TPS的概念，因为一笔交易需要调用多个服务来完成，所以一笔交易的完成其实就类似数据库里的一个事务，他涵盖了很多服务的请求调用，所以每秒完成多少笔交易，你可以用TPS来形容。

比如你说交易系统的TPS是300，就是说每秒可以完成300笔交易。那么比如交易系统中的服务A的QPS是500，就是交易系统中的一个服务A每秒可以处理500个请求。

2、一款非常好用的数据库压测工具

上一篇文章给大家讲解了我们在压测的过程中要关注哪些东西，这一篇文章就来带着大家一步一步的利用一个工具进行数据库压测。

先给大家介绍一个非常好用的数据库压测工具，就是**sysbench**，这个工具可以自动帮你在数据库里构造出来大量的数据，你想要多少数据，他就自动给你构造出来多少条数据。

然后这个工具接着可以模拟几千个线程并发的访问你的数据库，模拟使用各种各样的SQL语句来访问你的数据库，包括模拟出来各种事务提交到你的数据库里去，甚至可以模拟出几十万的TPS去压测你的数据库。

所以一般来说，如果你要进行数据库的压测，就是直接使用sysbench工具就可以了，这一篇文章我来带着大家学习一下这个压测工具的使用，大家学习完这一讲，完全可以自己本地装一个MySQL数据库，然后自己压测一下试一试。

3、在linux上安装sysbench工具

首先你需要有一台linux机器，如果你只有一个windows笔记本电脑，可以在里面装一个linux的虚拟机，然后你可以用如下的命令设置一下yum repo仓库，接着基于yum来安装sysbench就可以了，安装完成以后验证一下是否成功。

curl -s https://packagecloud.io/install/repositories/akopytov/sysbench/script.rpm.sh | sudo bash

sudo yum -y install sysbench

sysbench --version

如果上面可以看到sysbench的版本号，就说明安装成功了。

4、数据库压测的测试用例

接着我们需要在自己的数据库里创建好一个测试库，我们可以取个名字叫做test_db，同时创建好对应的测试账号，可以叫做test_user，密码也是test_user，让这个用户有权限可以访问test_db。

然后我们将要基于sysbench构建20个测试表，每个表里有100万条数据，接着使用10个并发线程去对这个数据库发起访问，连续访问5分钟，也就是300秒，然后对其进行压力测试。

5、基于sysbench构造测试表和测试数据

sysbench --db-driver=mysql --time=300 --threads=10 --report-interval=1 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=test_user --mysql-password=test_user --mysql-db=test_db --tables=20 --table_size=1000000 oltp_read_write --db-ps-mode=disable prepare

上面我们构造了一个sysbench命令，给他加入了很多的参数，现在我们来 解释一下这些参数，相信很多参数大家自己看到也就大致明白什么意思了：

--db-driver=mysql：这个很简单，就是说他基于mysql的驱动去连接mysql数据库，你要是oracle，或者sqlserver，那自然就是其他的数据库的驱动了
--time=300：这个就是说连续访问300秒
--threads=10：这个就是说用10个线程模拟并发访问
--report-interval=1：这个就是说每隔1秒输出一下压测情况
--mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=test_user --mysql-password=test_user：这一大串，就是说连接到哪台机器的哪个端口上的MySQL库，他的用户名和密码是什么
--mysql-db=test_db --tables=20 --table_size=1000000：这一串的意思，就是说在test_db这个库里，构造20个测试表，每个测试表里构造100万条测试数据，测试表的名字会是类似于sbtest1，sbtest2这个样子的
oltp_read_write：这个就是说，执行oltp数据库的读写测试
--db-ps-mode=disable：这个就是禁止ps模式

最后有一个prepare，意思是参照这个命令的设置去构造出来我们需要的数据库里的数据，他会自动创建20个测试表，每个表里创建100万条测试数据，所以这个工具是非常的方便的。

6、对数据库进行360度的全方位测试

测试数据库的综合读写TPS，使用的是oltp_read_write模式（大家看命令中最后不是prepare，是run了，就是运行压测）：

sysbench --db-driver=mysql --time=300 --threads=10 --report-interval=1 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=test_user --mysql-password=test_user --mysql-db=test_db --tables=20 --table_size=1000000 oltp_read_write --db-ps-mode=disable run

测试数据库的只读性能，使用的是oltp_read_only模式（大家看命令中的oltp_read_write已经变为oltp_read_only了）：

```
sysbench --db-driver=mysql --time=300 --threads=10 --report-interval=1 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=test_user --mysql-password=test_user --mysql-db=test_db --tables=20 --table_size=1000000 oltp_read_only --db-ps-mode=disable run
```

测试数据库的删除性能，使用的是oltp_delete模式：

```
sysbench --db-driver=mysql --time=300 --threads=10 --report-interval=1 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=test_user --mysql-password=test_user --mysql-db=test_db --tables=20 --table_size=1000000 oltp_delete --db-ps-mode=disable run
```

测试数据库的更新索引字段的性能，使用的是oltp_update_index模式：

```
sysbench --db-driver=mysql --time=300 --threads=10 --report-interval=1 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=test_user --mysql-password=test_user --mysql-db=test_db --tables=20 --table_size=1000000 oltp_update_index --db-ps-mode=disable run
```

测试数据库的更新非索引字段的性能，使用的是oltp_update_non_index模式：

```
sysbench --db-driver=mysql --time=300 --threads=10 --report-interval=1 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=test_user --mysql-password=test_user --mysql-db=test_db --tables=20 --table_size=1000000 oltp_update_non_index --db-ps-mode=disable run
```

测试数据库的更新非索引字段的性能，使用的是oltp_update_non_index模式：

```
sysbench --db-driver=mysql --time=300 --threads=10 --report-interval=1 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=test_user --mysql-password=test_user --mysql-db=test_db --tables=20 --table_size=1000000 oltp_update_non_index --db-ps-mode=disable run
```

测试数据库的插入性能，使用的是oltp_insert模式：

```
sysbench --db-driver=mysql --time=300 --threads=10 --report-interval=1 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=test_user --mysql-password=test_user --mysql-db=test_db --tables=20 --table_size=1000000 oltp_insert --db-ps-mode=disable run
```

测试数据库的写入性能，使用的是oltp_write_only模式：

```
sysbench --db-driver=mysql --time=300 --threads=10 --report-interval=1 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=test_user --mysql-password=test_user --mysql-db=test_db --tables=20 --table_size=1000000 oltp_write_only --db-ps-mode=disable run
```

使用上面的命令，sysbench工具会根据你的指令构造出各种各样的SQL语句去更新或者查询你的20张测试表里的数据，同时监测出你的数据库的压测性能指标，最后完成压测之后，可以执行下面的cleanup命令，清理数据。

```
sysbench --db-driver=mysql --time=300 --threads=10 --report-interval=1 --mysql-host=127.0.0.1 --mysql-port=3306 --mysql-user=test_user --mysql-password=test_user --mysql-db=test_db --tables=20 --table_size=1000000 oltp_read_write --db-ps-mode=disable cleanup
```

7、压测结果分析

按照我们上面的命令，我们是让他每隔1秒都会输出一次压测报告的，此时他每隔一秒会输出类似下面的一段东西：

```
[ 22s ] thds: 10 tps: 380.99 qps: 7312.66 (r/w/o: 5132.99/1155.86/1321.35) lat (ms, 95%): 21.33 err/s: 0.00 reconn/s: 0.00
```

我来给大家解释一下这是什么意思，首先他说的这是第22s输出的一段压测统计报告，然后是其他的一些统计字段：

thds: 10，这个意思就是有10个线程在压测
tps: 380.99，这个意思就是每秒执行了380.99个事务
qps: 7610.20，这个意思就是每秒可以执行7610.20个请求

(r/w/o: 5132.99/1155.86/1321.35)，这个意思就是说，在每秒7610.20个请求中，有5132.99个请求是读请求，1155.86个请求是写请求，1321.35个请求是其他的请求，就是对QPS进行了拆解

lat (ms, 95%): 21.33，这个意思就是说，95%的请求的延迟都在21.33毫秒以下

err/s: 0.00 reconn/s: 0.00，这两个的意思就是说，每秒有0个请求是失败的，发生了0次网络重连

这个压测结果会根据每个人的机器的性能不同有很大的差距，你要是机器性能特别高，那你可以开很多的并发线程去压测，比如100个线程，此时可能会发现数据库每秒的TPS有上千个，如果你的机器性能很低，可能压测出来你的TPS才二三十个，QPS才几百个，这都是有可能的。

另外在完成压测之后，最后会显示一个总的压测报告，我把解释写在下面了：

```
SQL statistics:
    queries performed:
        read: 1480084 // 这就是说在300s的压测期间执行了148万多次的读请求
        write: 298457 // 这是说在压测期间执行了29万多次的写请求
        other: 325436 // 这是说在压测期间执行了30万多次的其他请求
        total: 2103977 // 这是说一共执行了210万多次的请求
    // 这是说一共执行了10万多个事务，每秒执行350多个事务
    transactions: 105180( 350.6 per sec. )
    // 这是说一共执行了210万多次的请求，每秒执行7000+请求
    queries: 2103977 ( 7013.26 per sec. )
    ignored errors: 0 (0.00 per sec.)
    reconnects: 0 (0.00 per sec.)

// 下面就是说，一共执行了300s的压测，执行了10万+的事务
General staticstics:
    total time: 300.0052s
    total number of events: 105180

Latency (ms):
    min: 4.32 // 请求中延迟最小的是4.32ms
    avg: 13.42 // 所有请求平均延迟是13.42ms
    max: 45.56 // 延迟最大的请求是45.56ms
    95th percentile: 21.33 // 95%的请求延迟都在21.33ms以内
```

8、今日作业

今天希望大家可以完成一个作业，自己准备一台linux机器或者虚拟机，然后装一个mysql数据库，接着使用sysbench工具尝试一下数据库的压测，自己分析一下压测的报告和结果，感受一下你的数据库到底能抗多高的并发。

同时接下来我们还会给大家讲解在压测的过程中，如何去观察机器的其他重要的性能指标，比如说CPU、网络、内存、磁盘IO，等等。

End

狸猫技术窝精品专栏及课程推荐：

- 《从零开始带你成为消息中间件实战高手》
- 《21天互联网Java进阶面试训练营》（分布式篇）
- 《互联网Java工程师面试突击》（第1季）
- 《互联网Java工程师面试突击》（第3季）
- 《从零开始带你成为JVM实战高手》