

图文 08 生产经验：在数据库的压测过程中，如何360度无死角观察机器性能？

882 人次阅读 2020-02-04 08:08:55

[返回](#)
[前进](#)
[重新加载](#)
[打印](#)[详情](#) [评论](#)

生产经验：在数据库的压测过程中，如何360度无死角观察机器性能？

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）



狸猫技术窝

[进店逛](#)

1、除了QPS和TPS以外，我们还需要观察机器的性能

上一篇文章我们给大家讲解了如何使用sysbench这个工具非常方便的去对数据库进行压测，压测过后其实大家就会看到自己的数据库大概能抗下多少QPS和TPS了。

但是这里还得给大家说另外一个压测时的技巧，就是上一篇文章里我们是使用了10个线程去压测数据库，如果你的机器性能很高，然后你觉得10个线程没法压测出来数据库真实的最高负载能力，你其实可以在sysbench中不停的增加线程的数量，比如使用20个线程，甚至100个线程去并发的访问数据库，直到发现数据库的QPS和TPS上不去。

当然，这个不停的提高线程数量，不停的让数据库承载更高的QPS的过程，还需要配合我们对机器性能表现的观察来做，不能盲目的不停的增加线程去压测数据库。

这篇文章，我们就是要讲解在压测过程中，如何同时观察机器的性能表现，从而来决定是否要继续增加线程数量去压测数据库。

2、为什么在不停增加线程数量的时候，要密切关注机器性能？

我们先来解答一个问题，就是在压测的时候我们需要不停的增加线程的数量去让数据库承载更高的QPS，一直到最后看看数据库到底最高可以承载多高的QPS。

那么在这个过程中，为什么必须要密切的关注机器的性能呢？

我们来给大家举两个例子，相信大家看完就明白这个问题了。

首先，假设数据库当前抗下了每秒2000的QPS，同时这个时候机器的CPU负载、内存负载、网络负载、磁盘IO负载，都在正常的范围内，负载相对较高一些，但是还没有达到这些硬件的极限，那么我们可以认为这台数据库在高峰期抗到每秒2000的QPS，是没有问题的。

但是如果你一直不停的在压测过程中增加sysbench的线程数量，然后数据库此时勉强抗到了每秒5000的QPS了，但是这个时候你发现机器的CPU已经满负荷运行了，内存使用率特别高，内存都快要不行了，然后网络带宽几乎被打满了，磁盘IO的等待时间特别长，这个时候说明机器已经到了极致了，再搞下去，机器都快挂了。

所以这个时候你压测出来的5000 QPS是没什么代表性的，因为在生产环境根本不可能让数据库抗下这么高的QPS，因为到这么高的QPS就说明你的数据库几乎已经快要挂掉了，这是不现实的。

相关频道

从零开始
实战优化
已更新3

所以说，在压测的过程中，必须是不停的增加sysbench的线程数量，持续的让数据库承载更高的QPS，同时密切关注机器的CPU、内存、磁盘和网络的负载情况，在硬件负载情况比较正常的范围内，哪怕负载相对较高一些，也还是可以继续增加线程数量和提高数据库的QPS的。

然后当你不停的增加线程数量，发现在数据库抗下一个QPS的数值的同时，机器的CPU、内存、网络和磁盘的负载已经比较高了，到了一个有一定风险的临界值的了，此时就不能继续增加线程数量和提高数据库抗下的QPS了。

接着我们今天就给大家分析一下，在压测的过程中，需要使用哪些linux命令去观察机器的性能情况，以及机器的CPU、内存、磁盘和网络在什么样的负载下是正常的，在什么样的负载下就是比较危险的了。

3、压测时如何观察机器的CPU负载情况？

先来看一个最最常用的监测linux机器性能的命令，就是top命令，直接在linux命令行只能够输入top指令就可以了，然后我们这里来给大家解释一下，top指令展示出来的各种信息都是什么意思。

首先我们会看到如下一行信息：

```
top - 15:52:00 up 42:35, 1 user, load average: 0.15, 0.05, 0.01
```

先来解释一下这行信息，这行信息是最直观可以看到机器的cpu负载情况的，首先15:52:00指的是当前时间，up 42:35指的是机器已经运行了多长时间，1 user就是说当前机器有1个用户在使用。

最重要的是load average: 0.15, 0.05, 0.01这行信息，他说的是CPU在1分钟、5分钟、15分钟内的负载情况。

这里要给大家着重解释一下这个CPU负载是什么意思，假设我们是一个4核的CPU，此时如果你的CPU负载是0.15，这就说明，4核CPU中连一个核都没用满，4核CPU基本都很空闲，没啥人在用。

如果你的CPU负载是1，那说明4核CPU中有一个核已经被使用的比较繁忙了，另外3个核还是比较空闲一些。要是CPU负载是1.5，说明有一个核被使用繁忙，另外一个核也在用，但是没那么繁忙，还有2个核可能还是空闲的。

如果你的CPU负载是4，那说明4核CPU都被跑满了，如果你的CPU负载是6，那说明4核CPU被繁忙的使用还不够处理当前的任务，很多进程可能一直在等待CPU去执行自己的任务。

这个就是CPU负载的概念和含义。

所以大家现在知道了，上面看到的load average实际上就是CPU在最近1分钟，5分钟，15分钟内的平均负载数值，上面都是0.15之类的，说明CPU根本就没怎么用。

但是如果你在压测的过程中，发现4核CPU的load average已经基本达到3.5，4了，那么说明几个CPU基本都跑满了，在满负荷运转，那么此时你就不要再继续提高线程的数量和增加数据库的QPS了，否则CPU负载太高是不合理的。

4、压测时如何观察机器的内存负载情况？

在你执行top命令之后，中间我们跳过几行内容，可以看到如下一行内容：

```
Mem: 33554432k total, 20971520k used, 12268339 free, 307200k buffers
```

这里说的就是当前机器的内存使用情况，这个其实很简单，明显可以看出来就是总内存大概有32GB，已经使用了20GB左右的内存，还有10多G的内存是空闲的，然后有大概300MB左右的内存用作OS内核的缓冲区了。

对于内存而言，同样是要在压测的过程中紧密的观察，一般来说，如果内存的使用率在80%以内，基本都能接受，在正常范围内，但是如果你的机器的内存使用率到了70%~80%了，就说明有点危险了，此时就不要继续增加压测的线程数量和QPS了，差不多就可以了。

5、压测时如何观察机器的磁盘IO情况？

接着我们说说如何在压测的时候观察机器的磁盘IO的情况？

这里会使用dstat命令，我们之前给大家讲过几个磁盘IO相关的指标，包括存储的IO吞吐量、IOPS这些，我们下面就看看这里是如何查看的。

使用dstat -d命令，会看到如下的东西：

```
-dsk/total -
```

```
read writ
103k 211k
0 11k
```

在上面可以清晰看到，存储的IO吞吐量是每秒钟读取103kb的数据，每秒写入211kb的数据，像这个存储IO吞吐量基本上都不算多的，因为普通的机械硬盘都可以做到每秒钟上百MB的读写数据量。

使用命令：`dstat -r`，可以看到如下的信息

```
--io/total-
read writ
0.25 31.9
0 253
0 39.0
```

他的这个意思就是读IOPS和写IOPS分别是多少，也就是说随机磁盘读取每秒钟多少次，随机磁盘写入每秒钟执行多少次，大概就是这个意思，一般来说，随机磁盘读写每秒在两三百次都是可以承受的。

所以在这里，我们就需要在压测的时候密切观察机器的磁盘IO情况，如果磁盘IO吞吐量已经太高了，都达到极限的每秒上百MB了，或者随机磁盘读写每秒都到极限的两三百次了，此时就不要继续增加线程数量了，否则磁盘IO负载就太高了。

6、压测时观察网卡的流量情况

接着我们可以使用`dstat -n`命令，可以看到如下的信息：

```
-net/total-
recv send
16k 17k
```

这个说的就是每秒钟网卡接收到流量有多少kb，每秒钟通过网卡发送出去的流量有多少kb，通常来说，如果你的机器使用的是千兆网卡，那么每秒钟网卡的总流量也就在100MB左右，甚至更低一些。

所以我们在压测的时候也得观察好网卡的流量情况，如果网卡传输流量已经到了极限值了，那么此时你再怎么提高sysbench线程数量，数据库的QPS也上不去了，因为这台机器每秒钟无法通过网卡传输更多的数据了。

7、今天的一点总结和作业

今天的文章我们来做一点总结，今天给大家介绍了在数据库压测的过程中，你必须不停的增加sysbench的线程数量，增加数据库抗下的QPS，同时通过各种命令观察机器的CPU、内存、磁盘和网络的负载情况，如果你发现某个硬件负载已经很高了，此时就可以不再提高数据库的QPS了。

在硬件的一定合理的负载范围内，把数据库的QPS提高到最大，这就是数据库压测的时候最合理的一个极限QPS值，而不是不管机器的各个硬件的负载，盲目的不停的增加sysbench的线程数量，不停的让数据库增加可以抗下的QPS的数值。

今天同样给大家布置一个小作业，大家可以自己本地笔记本电脑装一个数据库，然后基于sysbench去压测，不停的提高线程数量，不停的提高数据库抗下的QPS，同时观察你的机器的CPU、内存、磁盘和网络的各项负载

然后希望大家对自己压测的情况做一个总结，发布到评论区跟其他同学进行交流。

End

专栏版权归公众号狸猫技术窝所有

未经许可不得传播，如有侵权将追究法律责任

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》](#)

返回
前进
重新加载
打印

返回

前进

重新加载

打印

认准一手QQ3195303913微信wxywd8