性能测试应该怎么做？

2016年7月6日 陈皓 3,124 人阅读 阅读评论 发表评论

偶然间看到了阿里中间件Dubbo的性能测试报告，我觉得这份性能测试报告让人觉得做这性能测试的人根本不懂性能测试，我觉得这份报告会把大众带沟里去，所以，想写下这篇文章，做一点科普。

首先，这份测试报告里的主要问题如下：

1）用的全是平均值。老实说，平均值是非常不靠谱的。

2）响应时间没有和吞吐量**TPS/QPS**挂钩。而只是测试了低速率的情况，这是完全错误的。

3）响应时间和吞吐量没有和成功率挂钩。

为什么平均值不靠谱

关于平均值为什么不靠谱，我相信大家读新闻的时候经常可以看到，平均工资，平均房价，平均支出，等等这样的字眼，你就知道为什么平均值不靠谱了。（这些都是数学游戏，对于理工科的同学来说，天生应该有免疫力）

软件的性能测试也一样，平均数也是不靠谱的，这里可以参看这篇详细的文章《Why Averages Suck and Percentiles are Great》，我在这里简单说一下。

我们知道，性能测试时，测试得到的结果数据不总是一样的，而是有高有低的，如果算平均值就会出现这样的情况，假如，测试了10次，有9次是1ms，而有1次是1s，那么平均数据就是100ms，很明显，这完全不能反应性能测试的情况，也许那1s的请求就是一个不正常的值，是个噪点，应该去掉。所以，我们会在一些评委打分中看到要去掉一个最高分一个最低分，然后再算平均值。

另外，中位数（Mean）可能会比平均数要稍微靠谱一些，所谓中位数的意就是把将一组数据按大小顺序排列，处在最中间位置的一个数叫做这组数据的中位数 ，这意味着至少有50%的数据低于或高于这个中位数。

当然，最为正确的统计做法是用百分比分布统计。也就是英文中的TP – Top Percentile ，TP50的意思在，50%的请求都小于某个值，TP90表示90%的请求小于某个时间。

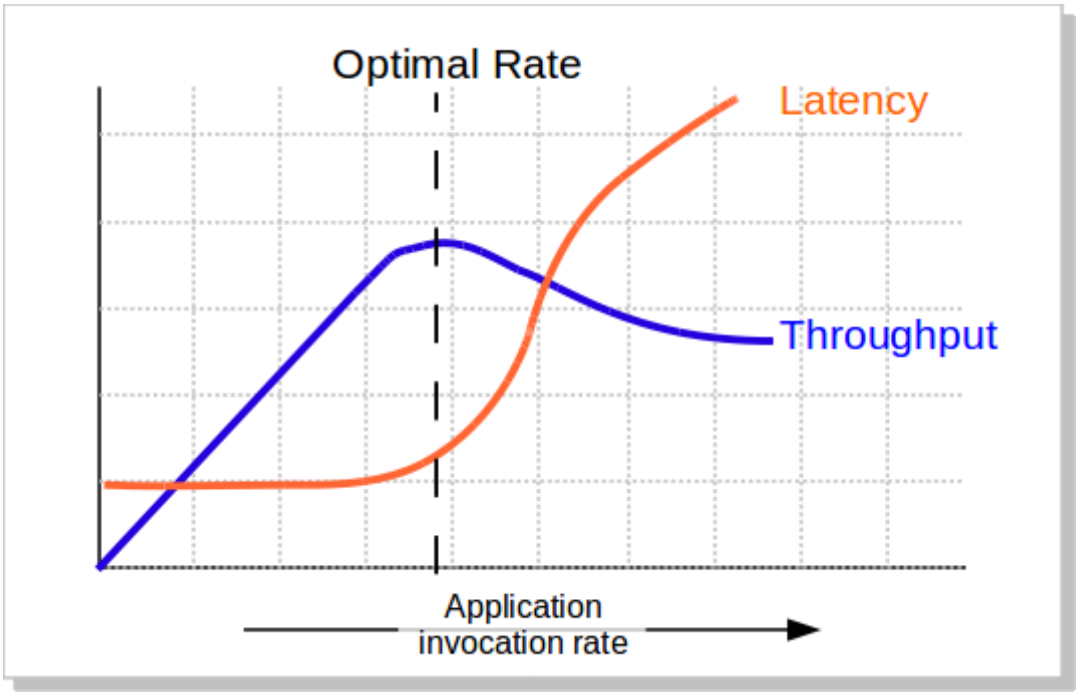
比如：我们有一组数据：[10ms, 1s, 200ms, 100ms]，我们把其从小到大排个序：[10ms, 100ms, 200ms, 1s]，于是我们知道，TP50，就是50%的请求ceil(4*0.5)=2时间是小于100ms的，TP90就是90%的请求ceil(4*0.9)=4时间小于1s。于是：TP50就是100ms，TP90就是1s。

我以前在路透做的金融系统响应时间的性能测试的要求是这样的，**99.9%**的请求必须小于**1ms**，所有的平均时间必须小于**1ms**。两个条件的限制。

为什么响应时间（latency）要和吞吐量（Thoughput）挂钩

系统的性能如果只看吞吐量，不看响应时间是没有意义的。我的系统可以顶10万请求，但是响应时间已经到了5秒钟，这样的系统已经不可用了，这样的吞吐量也是没有意义的。

我们知道，当并发量（吞吐量）上涨的时候，系统会变得越来越不稳定，响应时间的波动也会越来越大，响应时间也会变得越来越慢，而吞吐率也越来越上不去（如下图所示），包括CPU的使用率情况也会如此。所以，当系统变得不稳定的时候，吞吐量已经没有意义了。吞吐量有意义的时候仅当系统稳定的时候。



所以，吞吐量的值必需有响应时间来卡。比如：**TP99**小于**100ms**的时候，系统可以承载的最大并发数是**1000qps**。这意味着，我们要不断的在不同的并发数上测试，以找到软件的最稳定时的最大吞吐量。

为什么响应时间吞吐量和成功率要挂钩

我们这应该不难理解了，如果请求不成功的话，都还做毛的性能测试。比如，我说我的系统并发可以达到10万，但是失败率是

40%，那么，这10万的并发完全就是一个笑话了。

性能测试的失败率的容忍应该是非常低的。对于一些关键系统，成功请求数必须在100%，一点都不能含糊。

如何严谨地做性能测试

一般来说，性能测试要统一考虑这么几个因素：**Thoughput**吞吐量，**Latency**响应时间，资源利用（CPU/MEM/IO/Bandwidth...），成功率，系统稳定性。

下面的这些性能测试的方式基本上来源自我的老老东家汤森路透，一家做real-time的金融数据系统的公司。

> 一，你得定义一个系统的响应时间**latency**，建议是**TP99**，以及成功率。比如路透的定义：99.9%的响应时间必需在1ms之内，平均响应时间在1ms以内，100%的请求成功。
>
> 二，在这个响应时间的限制下，找到最高的吞吐量。测试用的数据，需要有大中小各种尺寸的数据，并可以混合。最好使用生产线上的测试数据。
>
> 三，在这个吞吐量做**Soak Test**，比如：使用第二步测试得到的吞吐量连续**7**天的不间断的压测系统。然后收集CPU，内存，硬盘/网络IO，等指标，查看系统是否稳定，比如，CPU是平稳的，内存使用也是平稳的。那么，这个值就是系统的性能
>
> 四，找到系统的极限值。比如：在成功率**100%**的情况下（不考虑响应时间的长短），系统能坚持**10**分钟的吞吐量。
>
> 五，做**Burst Test**。用第二步得到的吞吐量执行**5**分钟，然后在第四步得到的极限值执行**1**分钟，再回到第二步的吞吐量执行**5**钟，再到第四步的权限值执行**1**分钟，如此往复个一段时间，比如**2**天。收集系统数据：CPU、内存、硬盘/网络IO等，观察他们的曲线，以及相应的响应时间，确保系统是稳定的。
>
> 六、低吞吐量和网络小包的测试。有时候，在低吞吐量的时候，可能会导致latency上升，比如TCP_NODELAY的参数没有开启会导致latency上升（详见TCP的那些事），而网络小包会导致带宽用不满也会导致性能上不去，所以，性能测试还需要根据实际情况有选择的测试一下这两咱场景。

（注：在路透，路透会用第二步得到的吞吐量乘以66.7%来做为系统的软报警线，80%做为系统的硬报警线，而极限值仅仅用来扛突发的peak）

是不是很繁锁？是的，只因为，这是工程，工程是一门科学，科学是严谨的。

欢迎大家也分享一下你们性能测试的经验和方法。

（全文完）

流程方法, 程序设计　Performance, test