

# 如何模拟超过 5 万的并发用户

转载

公众号:方志朋 于 2020-03-08 10:28:00 发布 48 收藏  
文章标签: [java](#) [大数据](#) [python](#) [数据库](#) [分布式](#)

点击上方“方志朋”，选择“设为星标”

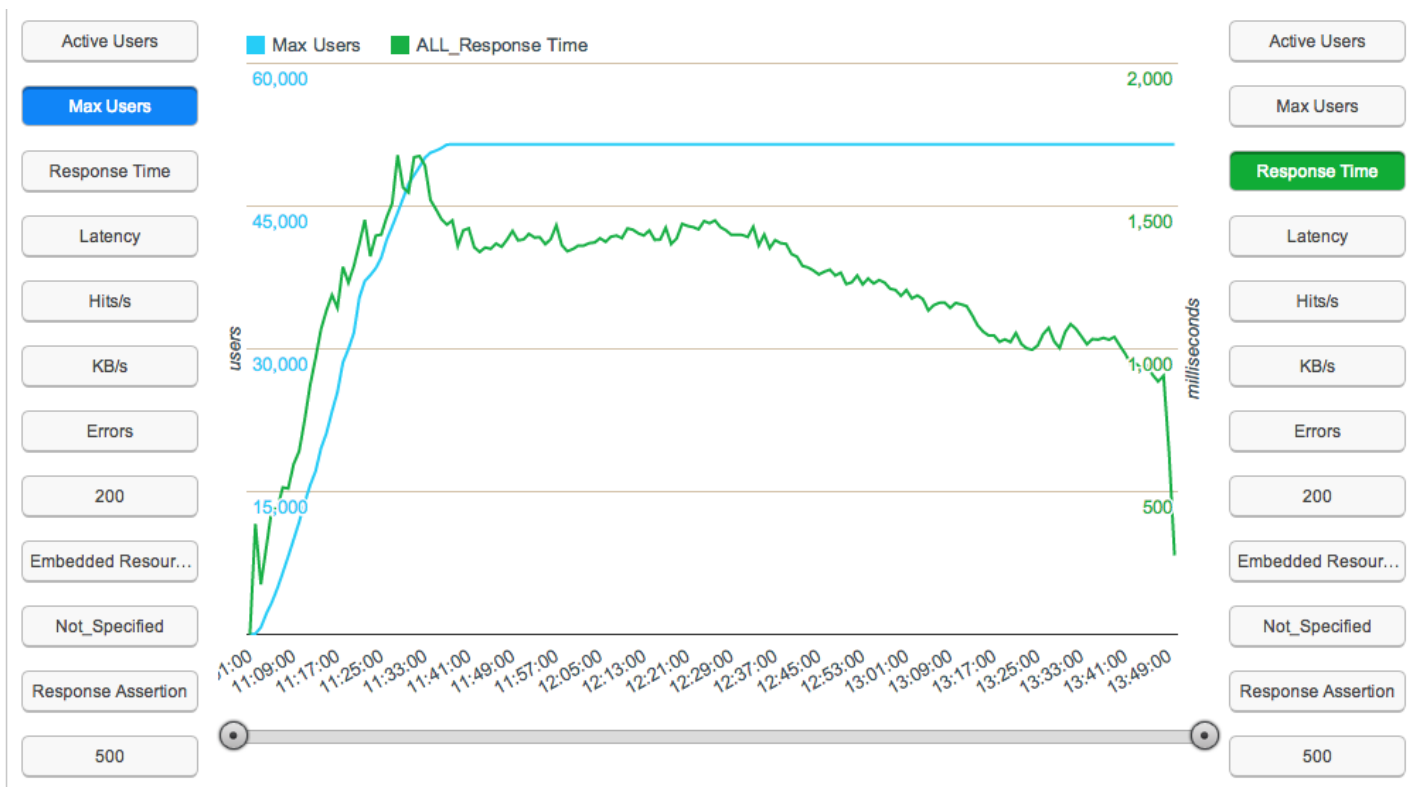
回复“666”获取新整理的面试文章



来源: [www.oschina.net/translate/how-run-load-test-50k-concurrent-users](http://www.oschina.net/translate/how-run-load-test-50k-concurrent-users)

本文将从负载测试的角度，描述了做一次流畅的5万用户并发测试需要做的事情。

你可以在本文的结尾部分看到讨论的记录。



## 步骤一：编写你的脚本

开始之前，请确定从JMeter的Apache社区jmeter.apache.org 获得了最新的版本。

你也会要下载这些附加的插件，因为它们可以让你的工作更轻松。

有许多方法可以获得脚本：

- 1、使用 BlazeMeter 的 Chrome 扩展 来记录你的方案
- 2、使用 JMeter HTTP(S) 测试脚本记录器 来设置一个代理，那样你就可以运行你的测试并记录下所有的东西
- 3、从头开始全部手工构建(可能是功能/QA测试)

如果你的脚本是一份记录的结果(像步骤1&2), 请牢记:

- 1、你需要改变诸如Username & Password这样的特定参数，或者你也许会想要设置一个CSV文件，有了里面的值每个用户就可以是不同的。
- 2、为了完成诸如“添加到购物车”，“登录”还有其它这样的请求，你也许要使用正则表达式，JSON路径提取器，XPath提取器，来提取诸如Token字符串，表单构建ID还有其它要素
- 3、保持你的脚本参数化，并使用配置元素，诸如默认HTTP请求，来使得在环境之间切换时你的工作更轻松。

## 步骤2：使用JMeter进行本地测试

在1个线程的1个迭代中使用查看结果树要素，调试样本，虚拟样本还有打开的日志查看器（一些JMeter的错误会在里面报告），来调试你的脚本。

遍历所有的场景(包括True 或者 False的回应) 来确保脚本行为确如预期...

在成功使用一个线程测试之后——将其提高到10分钟10到20个线程继续测试:

- 1、如果你想要每个用户独立——是那样的么？

2、有没有收到错误?

3、如果你在做一个注册过程, 那就看看你的后台 - 账户是不是照你的模板创建好了? 它们是不是独立的呢?

4、从总结报告中, 你可以看到对测试的统计 - 它们有点用么? (平均响应时间, 错误, 每秒命中率)

一旦你准备好了脚本:

1、通过移除任何调试和虚拟样本来清理脚本, 并删除你的脚本侦听器

2、如果你使用了侦听器(诸如 "将响应保存到一个文件"), 请确保你没有使用任何路径!, 而如果他是一个侦听器或者一个CSV数据集配置——请确保你没有使用你在本地使用的路径 - 而只要文件名(就好像跟你的脚本在同一个文件夹)

3、如果你使用了自己专有的JAR文件, 请确保它也被上传了.

4、如果你使用了超过一个线程组 (不是默认的那个) - 请确保在将其上传到BlazeMeter之前设置了这个值.

### 步骤3 : BlazeMeter沙箱测试

如果那时你的第一个测试——你应该温习一下 这篇 有关如何在BlazeMeter中创建测试的文章.

将沙箱的测试配置设置成, 用户300, 1个控制台, 时间50分钟.

对沙箱进行这样的配置让你可以在后台测试你的脚本, 并确保上的BlazeMeter的一切都运行完好.

为此, 先按下灰色的按钮: 告诉JMeter引擎我想要完全控制! - 来获得对你的测试参数的完全控制

通常你将会遇到的问题:

1、防火墙 - 确保你的环境对BlazeMeter的CIDR 列表 (它们会实时更新)开发, 并把它们放入白名单中

2、确保你所有的测试文件, 比如: CSVs, JAR, JSON, User.properties 等等.. 都可以使用

3、确保你没有使用任何路径

如果仍然有问题, 那就看看错误日志吧(你应该可以把整个日志都下载下来).

一个沙箱的配置可以是这样的:

引擎: 是能使控制台(1 个控制台, 0 个引擎)

线程: 50-300

产能提升: 20 分钟

迭代: 一直测试下去

时间: 30-50 分钟

这可以让你在产能提升期间获得足够多的数据(以防你遇到问题), 而你将对结果进行分析, 以确保脚本的执行确如预期.

你应该观察下Waterfall / WebDriver 选项卡来看看请求是否正常, 你不应该在这点上出任何问题 (除非你是故意的).

你应该盯着监控选项卡, 观察期内存和CPU消耗 - 这对你在步骤4中尝试设置每一个引擎的用户数量.

#### 步骤4：使用1个控制台和1个引擎来设置每个引擎用户的数量

现在我们可以肯定脚本能在BlazeMeter中完美运行了——我们需要计算出要多少用户放到一个引擎中。

如果你能用用户沙箱中的数据来做这个决定，那就太棒了！

在这里，我会给出一种不用回头去查看沙箱测试数据就能计算出这个数的方法。

设置你的测试配置：

线程数：500

产能提升：40分钟

迭代：永久

时长：50分钟

使用一个控制台和一个引擎。

运行测试并(通过监视选项卡)对你的测试引擎进行监视。

如果你的引擎对于75%的CPI使用率和85%的内存使用率都没有达到(一次性的峰值可以忽略)的话：

将线程数调整到700在测试一次

提交线程的数量直到线程数达到1000或者60%的CPU或内存使用

如果你的引擎过了75%的CPU使用率或者85%的内存使用率(一次性的峰值可以忽略)：

看看你第一次达到75%的点，在那个点有多少并发用户。

在运行一次测试，而不是提高你之前500个用户数量的产能

这一次将产能提升放到真实的测试中(5-15分钟是一个好的开始)并将时长设置为50分钟。

确保整个测试过程中没有超过75%的CPU使用率或者85%的内存使用率...

为安全起见，你可以把每个引擎的线程数降低**10%**的。

#### 步骤5：安装并测试集群

我们现在知道了从一个引擎中我们得到了多少线程，在该章节的最后，我们将会知道一个集群能给我们提供多少用户。

一个集群是指具有一个控制台（仅有一个）和0-14个引擎的逻辑容器。

即使你可以创建一个使用超过14个引擎的测试案例——但实际上是创建了两个集群（你可以注意到控制台的数量增加了），并且克隆了你的测试案例.....

每个集群具有最多14个引擎，是基于BlazeMeter自己本身的测试，以确保控制台可以控制这14台引擎对新建的大量数据处理的压力。

所以在这一步骤中，我们会用步骤4种的测试，并且仅仅修改引擎数量，将其增加到14。

将该测试按照最终测试的全部时长运行。当测试在运行时，打开监听标签，并且检验：

1. 没有一个引擎超过CPU75%的占有率和内存85%占有率的上限；

2. 定位你的控制台标签（你可以通过一次点击Logs Tab->Network Information，查看控制台私有IP地址来找到它的名字）——它不应该达到CPU75%占有率和内存85%占有率的上限。如果你的控制台达到了该上限——减少引擎数量并重新运行直到控制台在该上限之下。

在这个步骤的最后，你会发现：

1. 每个集群的用户数量；
2. 每个集群的命中率。

查看Aggragate Table中的其他统计信息，并找到本地结果统计图来获得有关你集群吞吐量的更多信息。

## 步骤 6：使用 Master / Slave 特性来达成你的最大CC目标

我们到了最后一步了。

我们知道脚本正在运行，我们也知道一个引擎可以支持多少用户以及一个集群可以支持多少用户。

让我们做一下假设：

一个引擎支持500用户

一个集群可以用12个引擎

我们的目标是5万用户测试

因此为了完成这些，我们需要8.3 个集群..

我们可以用8个12台引擎的集群和一个4台引擎的集群 - 但是像下面这样分散负载应该会更好：

每个集群我们用10台引擎而不是12，那么每个集群可以支持  $10 * 500 = 5K$  用户并且我们需要10个集群来支持5万用户。

这样可以得到如下好处：

- 1、不用维护两个不同的测试类型
- 2、我们可以通过简单的复制现有集群来增加5K用户（5K比6K更常见）
- 3、只要需要我们可以一直增加

现在，我们已经准备好创建最终的5万用户级别的Master / Slave测试了：

- 1、将测试的名称从"My prod test" 改为"My prod test - slave 1"。
- 2、我们回到步骤5，将高级测试属性(Advanced Test Properties)下的Standalone修改为Slave。
- 3、按保存按钮——现在我们有了一个Master和9个Slave中的一个。
- 4、返回你的 "My prod test -slave 1".
- 5、按复制按钮
- 6、接下来重复步骤1-5直到你创建了9个slave。
- 7、回到你的 "My prod test -salve 9" 并按复制按钮.
- 8、将测试的名称改为 "My prod test -Master".

9、将高级测试属性(Advanced Test Properties) 下的Slave改为Master。

10、检查我们刚才创建的所有的Slave(My prod test -salve 1..9)并按保存。

你的5万用户级别的Master-Slave测试已经准备好了。通过按master上的开始按钮来运行10个测试，每个测试5千用户。

你可以修改任意一个测试（salve或master），让它们来自不同的区域，有不同的脚本/csv/以及其他文件，使用不同的网络模拟器，不同的参数等。

你可以在一个叫“Master load results”的master报告中的一个新tab页中找到生成的聚合结果的报告，你还可以通过打开单个的报告来独立的查看每一个测试结果。

**热门内容：硬核干货：一位码农的架构师封神之路！**

**阿里问题定位神器 Arthas 的骚操作，定位线上BUG，超给力**

**用好idea这几款插件，可以帮你少写30%的代码！浅谈 CAP 和 Paxos 共识算法一个丧心病狂的Github项目：东北话编程，大写的**

**最近面试BAT，整理一份面试资料《Java面试BAT通关手册》，覆盖了Java核心技术、JVM、Java并发、SSM、微服务、数据库、数**  
**获取方式：点“在看”，关注公众号并回复 666 领取，更多内容陆续奉上。**

**明天见(。·ω·。**