Locust学习总结

**1.locust是一个开源性能测试工具**

**2.locust执行命令:**

*locust -f .\脚本名称 --host=测试网址*

-f 指定性能测试脚本文件。

–host 指定被测试应用的URL的地址，注意访问百度使用的HTTPS协议。

**3.locust访问地址：**

[http://localhost:8089（Locust启动网络监控器，默认为端口号为:](http://localhost:8089%EF%BC%88Locust%E5%90%AF%E5%8A%A8%E7%BD%91%E7%BB%9C%E7%9B%91%E6%8E%A7%E5%99%A8%EF%BC%8C%E9%BB%98%E8%AE%A4%E4%B8%BA%E7%AB%AF%E5%8F%A3%E5%8F%B7%E4%B8%BA:) 8089）

**4.locust no-web模式：**

*locust -f load\_test.py --host=https://www.baidu.com --no-web -c 10 -r 2 -t 1m*

**启动参数：**

–no-web 表示不使用Web界面运行测试。

-c 设置虚拟用户数。

-r 设置每秒启动虚拟用户数。

-t 设置设置运行时间。

**5.Locust 分布式运行:**

在 master 模式下启动 Locust:

*locust -f my\_loucstfile.py --master*

在每个 slave 中执行(192.168.0.14 替换为你 msater 的IP):

*locust -f my\_locustfile.py --slave --master-host=192.168.0.14*

参数说明:

*–master*

设置 Locust 为 master 模式。网页交互会在这台节点机器中运行。

*–slave*

设置 Locust 为 slave 模式。

*–master-host=X.X.X.X*

可选项，与 --slave 一起结合使用，用于设置 master 模式下的 master 机器的IP/hostname(默认设置为127.0.0.1)

*–master-port=5557*

可选项，与 --slave 一起结合使用，用于设置 master 模式下的 master 机器中 Locust 的端口(默认为5557)。注意，locust 将会使用这个指定的端口号，同时指定端口+1的号也会被占用。因此，5557 会被使用，Locust将会使用 5557 和 5558。

*–master-bind-host=X.X.X.X`*

可选项，与 --master 一起结合使用。决定在 master 模式下将会绑定什么网络接口。默认设置为\*(所有可用的接口)。

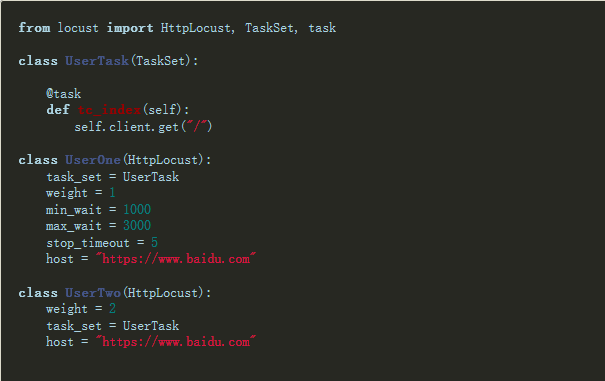
*–master-bind-port=5557*

可选项，与 --master 一起结合使用。决定哪个网络端口 master 模式将会监听。默认设置为 5557。注意 Locust 会使用指定的端口号，同时指定端口+1的号也会被占用。因此，5557 会被使用，Locust 将会使用 5557 和 5558。

*–expect-slaves=X*

在 no-web 模式下启动 master 时使用。master 将等待X连接节点在测试开始之前连接

**6.Locust 的类和方法**



**HttpLocust 类:**

**task\_set:** 指向一个 TaskSet 类，TaskSet 类定义了每个用户的行为。

**min\_wait:** 用户执行任务之间等待时间的下界，单位：毫秒。如果TaskSet类中有覆盖，以TaskSet 中的定义为准。

**max\_wait:** 用户执行任务之间等待时间的上界，单位：毫秒。如果TaskSet类中有覆盖，以TaskSet中的定义为准。

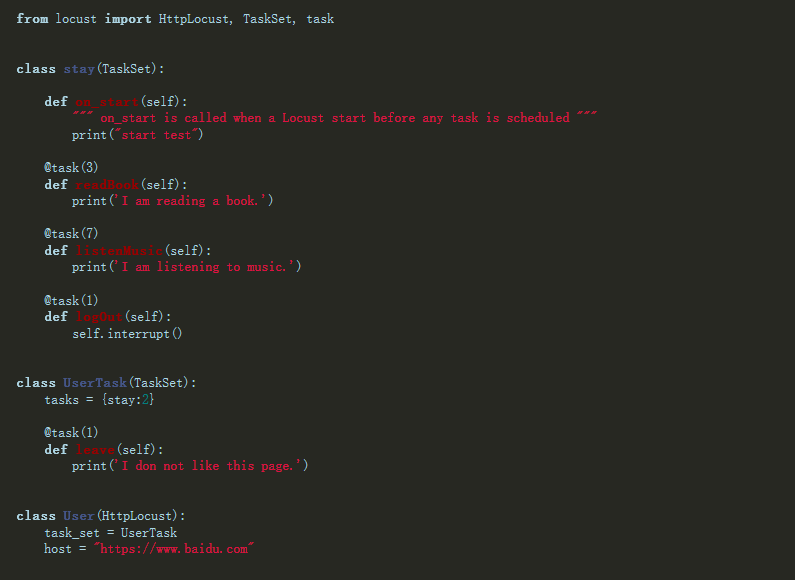
**host:** 如果是 Web 服务的测试，host 相当于是提供 URL 前缀的默认值，但是如果在命令行中指定了 --host 选项，则以命令行中指定的为准。如果不是 Web 服务测试，使用默认的 None 就好。

**stop\_timeout:** 设置 Locust 多少秒后超时，如果为 **None** ,则不会超时。

**Weight:** 一个Locust实例被挑选执行的权重，数值越大，执行频率越高。

**TaskSet 类:**

TaskSet类定义了每个用户的任务集合，测试任务开始后，每个 Locust 用户会从 TaskSet 中随机挑选一个任务执行，然后随机等待 HttpLocust 类中定义的 min\_wait和 max\_wait 之间的一段时间，执行下一个任务。



**on\_start()：**定义每个 Locust 用户开始做的第一件事。

**@task**: 通过@task()装饰的方法为一个事务。方法的参数用于指定该行为的执行权重。参数越大每次被虚拟用户执行的概率越高。如果不设置默认为1。

i**nterrupt(reschedule=True):** 顶层的TaskSet（即被绑定到某个Locust类的task\_set的第一层TaskSet）不能调用这个方法。reschedule置为True时，从被嵌套任务出来马上选择新任务执行，如果置为False，从被嵌套任务出来后，随机等待min\_wait和max\_wait之间的一段时间，再选择新任务执行。

**tasks 属性:** tasks = {stay:2} 表示每个用户执行 stay 的频率是2，也就的 UserTask 的两倍。