Python+Selenium笔记（十一）：配置selenium Grid

（一） 前言

Selenium Grid可以将测试分布在若干个物理或虚拟机器上，从而实现分布方式或并行方式执行测试。

这个链接是官方的相关说明。

https://github.com/SeleniumHQ/selenium/wiki/Grid2

（二） Selenium Grid

大概就是这个意思（一个中心节点（HUB），N个子节点（NODE，操作系统+浏览器））

（三） 环境配置

1. 前提：已经配置相应的 JDK环境（LINUX自带JDK环境(我装的LINUX自带的是1.8)，WINDOWS要自己配置JDK环境（我装的是1.9））

2. https://docs.seleniumhq.org/download/ 下载 Selenium Standalone Server

3. 启动Selenium Grid server（hub）

Selenium Grid server(hub,作为中心节点的电脑),切换到Selenium Standalone所在的目录（直接在Selenium Standalone所在的文件夹shift+鼠标右键，选择在此处打开命令窗口，或者CD 路径），然后执行下面的命令

4. 配置node（节点）

（1）   Node(也就是其他电脑或虚拟机环境，也可以直接将hub所在的那台电脑添加为node)，在系统变量path中加上相应的驱动文件的路径。（例如：将chromedriver所在的路径添加到path中，之前已经说过火狐、IE、谷歌浏览器环境的配置）

（2）   执行下面的命令

java -jar selenium-server-standalone-3.9.1.jar -role node -browser "browserName=firefox,version=62,maxSession=3,platform=WINDOWS" -hub http://192.168.4.196:4444/grid/register -port 5555

说明：

browserName：浏览器名称

Version：浏览器版本

maxSession：支持并发浏览器实例的数量

platform:操作系统

-hub：http:// (Selenium Grid server（hub）的IP)+启动时设置的端口号/ grid/register

-port ：指定端口号

（3）   要在同一电脑（或虚拟机）添加其他节点，再打开一个CMD窗口，运行上面的命令就行了（记得修改浏览器信息），记得同一电脑上端口号别重复。

（四） 环境配置好后如下图所示（我电脑开着一个linux的虚拟机太卡了，有点浪费时间的感觉，不想弄了，不过应该和windows上的差距不会太大，有环境的可以试下添加path变量、在终端执行相关命令添加节点）

（五） 示例（运行脚本会发现，直接在相匹配的环境中运行测试）

（六） 未解决的问题（单单配置环境是不够的）

1、 实现多环境并行运行同一测试脚本

2、 实现多个测试脚本并行运行（例如几百个测试脚本要是一个个运行的话，要用很长时间）

java -jar selenium-server-standalone-<version>.jar -role hub

例如：java -jar selenium-server-standalone-3.9.1.jar -role hub

可以加 -port 指定端口号，默认4444

http://localhost:4444/grid/console 启动后用这个地址访问

1 import sys  
 2 import unittest  
 3 from selenium import webdriver  
 4 from selenium.webdriver.common.action\_chains import ActionChains  
 5 class SearchTest(unittest.TestCase):  
 6 #定义2个全局属性，没有外部参数时，使用默认值  
 7 PLATFORM = "WINDOWS"  
 8 BROWSER = "firefox"  
 9 @classmethod  
10 def setUpClass(cls):  
11 #设置操作系统和浏览器  
12 desired\_caps = {}  
13 desired\_caps['platform'] = cls.PLATFORM  
14 desired\_caps['browserName'] = cls.BROWSER  
15 #这里的IP就是HUB所在电脑的ip  
16 cls.driver = webdriver.Remote('http://192.168.3.2:4444/wd/hub',desired\_caps)  
17 cls.driver.implicitly\_wait(10)  
18 cls.driver.maximize\_window()  
19 cls.driver.get("https://www.cnblogs.com/")  
20   
21 def test\_search\_by\_look(self):  
22 seach\_class = self.driver.find\_element\_by\_xpath('//li/a[@href="/cate/2/"]')  
23 #定位编程语言下的小类Python  
24 seach\_small =self.driver.find\_element\_by\_xpath('//li/a[@href="/cate/python/"]')  
25 ActionChains(self.driver).move\_to\_element(seach\_class).perform()  
26 seach\_small.click()  
27 #检查打开的网页标题是不是 Python - 网站分类 - 博客园  
28 self.assertEqual(self.driver.title,"Python - 网站分类 - 博客园" )  
29   
30 @classmethod  
31 def tearDownClass(cls):  
32 cls.driver.quit()  
33   
34 if \_\_name\_\_ == '\_\_main\_\_':  
35 #使用命令行运行脚本时，如果添加了参数，PLATFORM和BROWSER使用外部参数  
36 if len(sys.argv)>1:  
37 SearchTest.PLATFORM = sys.argv.pop()  
38 SearchTest.BROWSER = sys.argv.pop()  
39 #加verbosity=2参数，在命令行中显示具体的测试方法  
40 unittest.main(verbosity=2)













