Grid2官方文档：<https://github.com/SeleniumHQ/selenium/wiki/Grid2>

本文采用selenium-server-standalone-3.X.X以上版本搭建，若采用2.53.0及以下版本，请仅作参考

一、环境准备

1、准备两台计算机（可win、可linux），一台作为hub机，另外作为node机，电脑上均安装并配置好java环境，建议1.8版本以上。

2、hub机和node机之间的网络要通畅，可以ping通，可以telnet到hub服务启动的端口。

2、作为node节点的电脑上安装好IE、Chrome、Firefox三种浏览器。具体浏览器及webdriver版本关系可参考：selenium webdriver 启动三大浏览器Firefox,Chrome,IE

3、下载selenium-server-standalone，保存到两台计算机中，位置随意。（建议下载3.x.x 以上版本，本文按照此版本进行配置）

下载地址：<http://selenium-release.storage.googleapis.com/index.html>

二、建立连接

1、在hub机上，selenium-server-standalone文件目录下，打开cmd窗口，执行语句

①默认执行：C:\>java -jar selenium-server-standalone-3.X.X.jar -role hub

参数的含义，可参考下文：<https://blog.csdn.net/xiaomin1991222/article/details/50980117>

②修改配置执行：C:\>java -jarselenium-server-standalone-2.37.0.jar -role hub -hubConfig hubconfig.json

hubconfig.json内容：

{

"port": 4444,

"newSessionWaitTimeout": -1,

"servlets" : [],

"withoutServlets": [],

"custom": {},

"capabilityMatcher": "org.openqa.grid.internal.utils.DefaultCapabilityMatcher",

"registry": "org.openqa.grid.internal.DefaultGridRegistry",

"throwOnCapabilityNotPresent": true,

"cleanUpCycle": 5000,

"role": "hub",

"debug": false,

"browserTimeout": 0,

"timeout": 1800

}

启动成功后，访问网址 http://localhost:4444/grid/console,验证网址是否出现“view config”链接，即可判断hub启动成功。默认情况下selenium使用4444作为端口。



2、在node机上执行selenium-server-standalone文件目录下，打开cmd窗口，执行语句

①默认执行：java -jar selenium-server-standalone-3.7.1.jar -role node -hub http://192.168.XX.XXX:4444/grid/register （url地址更换为hub机的地址）

参数的含义，可参考下文：<https://blog.csdn.net/xiaomin1991222/article/details/50980117>

②修改配置执行：java -jar selenium-server-standalone-3.7.1.jar -role node -hub http://192.168.XX.XXX:4444/grid/register -nodeConfig node.json（url地址更换为hub机的地址）

node.json内容：

{

"capabilities":

[

{

"browserName": "firefox",

"marionette": true,

"maxInstances": 3,

"seleniumProtocol": "WebDriver"

},

{

"browserName": "chrome",

"maxInstances": 2,

"seleniumProtocol": "WebDriver"

},

{

"browserName": "internet explorer",

"maxInstances": 1,

"seleniumProtocol": "Selenium"

}

],

"proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy",

"maxSession": 5,

"port": 5555,

"register": true,

"registerCycle": 5000,

"hub": "http://192.168.42.133:4444",

"nodeStatusCheckTimeout": 5000,

"nodePolling": 5000,

"role": "node",

"unregisterIfStillDownAfter": 60000,

"downPollingLimit": 2,

"debug": false,

"servlets" : [],

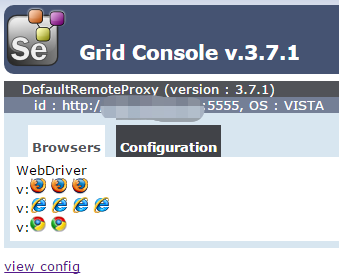
"withoutServlets": [],

"custom": {}

}

（备注：此为3.X.X以上版本，2.53.0以下版本可参考<https://github.com/SeleniumHQ/selenium/blob/selenium-2.53.0/java/server/src/org/openqa/grid/common/defaults/DefaultNode.json>）

再次访问网址 http://localhost:4444/grid/console，验证Node节点是否在Hub注册成功。



注意：①、3.X.X官方文档中"port": -1,，应改为自定义端口号，2.X.X版本默认为5555

②、3.X.X版本中，采用在命令行后面添加-Dwebdriver.XXX.driver的方式已经不能使用，可在执行启动程序中System.setProperty("webdriver.XXX.driver", "XXXDriver.exe")配置

③、安装浏览器时候，尽量选择默认安装位置，3.X.X版本不需要配置浏览器位置，若改变安装位置，不知是否有影响

三、执行脚本

注意：①、使用node节点远程执行时，需在node机上安装各浏览器的XXXdriver，需配置在环境变量path下可执行文件夹中（如：XXXdriver下载到D:\selenium中，将该地址配置到环境变量path中），且名称不能改变，必须为chromedriver、geckodriver、IEDriverServer等；②、本机执行时，需先声明浏览器路径、XXXdriver路径。

以IE为例：

1、创建RemoteDriverUtil：

public class RemoteDriverUtil {

static WebDriver driver;

static String nodeUrl = "http://192.168.XX.XXX:5555/wd/hub";

// 启用远程调用ie浏览器

public static WebDriver createRemoteIEDriver() {

// 指定调用IE进行测试

DesiredCapabilities capability = DesiredCapabilities.internetExplorer();

// 避免IE安全设置里，各个域的安全级别不一致导致的错误capability.setCapability(InternetExplorerDriver.INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS, true);

// 连接到selenium hub，远程启动浏览器

capability.setPlatform(Platform.XP);

try {

driver = new RemoteWebDriver(new URL(nodeUrl), capability);

} catch (MalformedURLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return driver;

}

}

2、创建IETest，并执行

public class IETest {

@Test

public void call() {

System.out.println("start selenium IE");

// 设置IEdriver路径

System.setProperty("webdriver.ie.driver",

System.getProperty("user.dir") + "\\extend\\IEDriverServer-3.4.0.exe");

//代码关闭IE一些配置

DesiredCapabilities dc = DesiredCapabilities.internetExplorer();

dc.setCapability(InternetExplorerDriver.INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS, true);

dc.setCapability("ignoreProtectedModeSettings", true);

// 初始化firefox浏览器实例

WebDriver driver = RemoteDriverUtil.createRemoteIEDriver();

// 最大化窗口

driver.manage().window().maximize();

// 设置隐性等待时间

driver.manage().timeouts().implicitlyWait(8, TimeUnit.SECONDS);

// 打开一个站点

driver.get("http://www.baidu.com/");

// 获取当前页面的标题

System.out.println("当前打开页面的标题是：" + driver.getTitle());

// 关闭浏览器

driver.close();

System.out.println("end selenium IE");

}

}

注意：在linux系统下运行浏览器时候，XXXdriver一定要在linux系统下进行解压缩，并同win系统一样，将所在目录配置到环境变量path中，启动必须在图形化界面下启动