# How to Speed up Test Execution Using Selenium Grid

**What is need of Selenium Grid?**

As you go through entire Selenium WebDriver modules you will find out WebDriver will execute your test cases in a single machine.

**Here are few problems with such a setup:**

1. What if you want to execute your test cases for different Operating Systems?
2. How to run your test cases in different version of same browser?
3. How to run your test cases for multiple browsers?
4. Why a scenario should wait for execution of other test cases even if it does not depend upon any test cases?

**All these problems are addressed in Selenium GRID.**

As we proceed with the Selenium course, we will get the idea about how we can overcome to these problems. Basically Grid architecture is based on master slave architecture. Master machine distributes test cases to different slave machines.

There are 2 versions of Grid available. Selenium Grid 2.0 is the latest from Selenium. Selenium 1.0 was the earlier version. Most of the Selenium experts prefer using Selenium Grid 2.0 as it is packed with new features. Selenium Grid 2.0 supports both Selenium RC and Selenium WebDriver scripts.

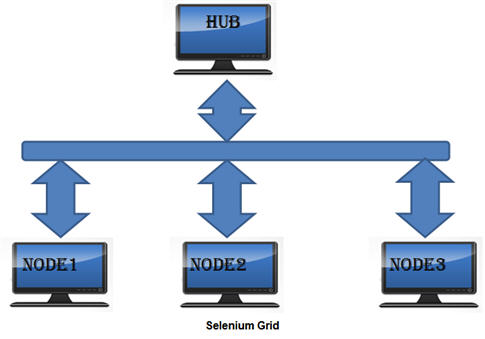
**Benefits of Selenium Grid:**

1. Selenium Grid gives the flexibility to distribute your test cases for execution.
2. Reduces batch processing time.
3. Can perform multi browser testing.
4. Can perform multi OS testing.

**Basic terminology of Selenium Grid:**

**Hub**: Hub is central point to the entire GRID Architecture which receives all requests. There is only one hub in the selenium grid. Hub distributes the test cases across each node.

**Node**: There can be multiple nodes in Grid. Tests will run in nodes. Each node communicates with the Hub and performs test assigned to it.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Selenium-grid-Tutorial-1.jpg)

**Install Selenium GRID**

**Step 1**: Download Selenium Server jar file from Selenium’s official website which is formerly known as Selenium RC Server and save it at any location on the local disk.

URL of selenium HQ: <http://www.seleniumhq.org/download/>

**Step 2**: Open command prompt and navigate to folder where the server is located. Run the server by using below command

*java -jar selenium-server-standalone-2.41.0.jar -role hub*

The hub will use the port 4444 by default. This port can be changed by passing the different port number in command prompt provided the port is open and has not been assigned a task.

Status can be checked by using the web interface:*http://localhost:4444/grid/console*

**Step 3**: Go to the other machine where you intend to setup Nodes. Open the command prompt and run the below line.

|  |  |  |
| --- | --- | --- |
| 1 | java -jar selenium-server-standalone-2.41.0.jar -role node -hub | |
| 2 | http://localhost:4444/grid/register -port 5556 |

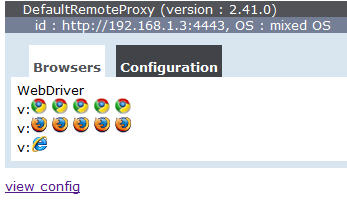
Run the selenium server in other machines to start nodes.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Selenium-grid-Tutorial-2.jpg)

**Browser and Nodes:**

After starting hub and nodes in each machine when you will navigate to GRID Console

You will find 5 Chrome, 5 Firefox and 1 IE browser under Browser section like below.

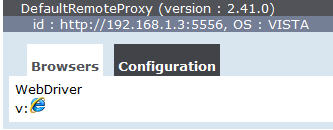
[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Selenium-grid-Tutorial-3.jpg)

This indicates that by default you can use 5 Chrome, 5 Firefox and 1 IE browser.

For Instance if you want to use only IE you can start the node by using below command:

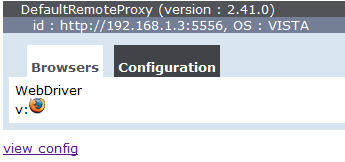
|  |  |
| --- | --- |
| 1 | java -jar selenium-server-standalone-2.41.0.jar -role webdriver -hub |
| 2 | http://localhost:4444/grid/register -port 5556 -browser browserName=iexplore | |

Verify the browser Type along with other details in GRID Console by clicking on *view config*.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Selenium-grid-Tutorial-4.jpg)

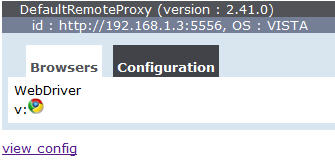
**Similarly for Firefox:**

|  |  |
| --- | --- |
| 1 | java -jar selenium-server-standalone-2.41.0.jar -role webdriver -hub |
| 2 | http://localhost:4444/grid/register -port 5556 -browser browserName=firefox | |

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Selenium-grid-Tutorial-5.jpg)

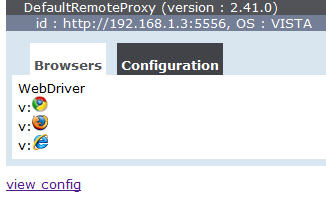
**For Chrome:**

|  |  |
| --- | --- |
| 1 | java -jar selenium-server-standalone-2.41.0.jar -role webdriver -hub |
| 2 | http://localhost:4444/grid/register -port 5556 -browser browserName=chrome | |

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Selenium-grid-Tutorial-6.jpg)

There are few scenarios where you may need browser from each type i.e.: IE, Chrome and Firefox.

For instance you may need to use 1 IE and 1 Firefox and 1 Chrome browser

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Selenium-grid-Tutorial-7.jpg)

|  |  |
| --- | --- |
| 1 | java -jar selenium-server-standalone-2.41.0.jar -role webdriver -hub |
| 2 | http://localhost:4444/grid/register -port 5556 -browser browserName=iexplore | |

|  |  |
| --- | --- |
| 3 | -browser browserName=firefox -browser browserName=chrome |

**maxInstances:**

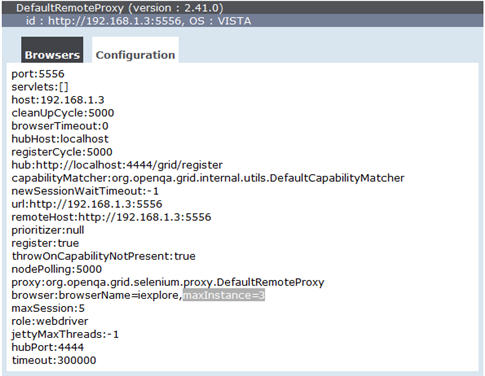
maxInstance is used to limit the number of browser initialization in a node.

For example if you want to work with 2 Firefox and 2 IE then you can start the node using maxInstance.

|  |  |
| --- | --- |
| 1 | java -jar selenium-server-standalone-2.41.0.jar -role webdriver -hub |
| 2 | http://localhost:4444/grid/register -port 5556 -browser browserName=firefox,maxInstance=3 | |

Maximum instance can be verified under configuration tab.

------------

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Selenium-grid-Tutorial-8.jpg)

Similarly other browser instances can be configured using maxInstances.

**maxSession:**

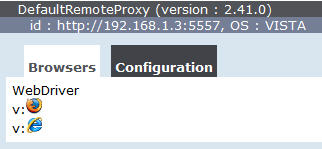
maxSession is used to configure how many number of browsers can be used parallel in the remote system.

|  |  |
| --- | --- |
| 1 | java -jar selenium-server-standalone-2.41.0.jar -role webdriver -hub |
| 2 | http://localhost:4444/grid/register -port 5556 -browser browserName=chrome,maxInstance=3 | |

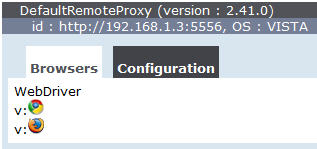
|  |  |
| --- | --- |
| 3 | -browser browserName=firefox,maxInstance=3 –maxSession 3 |

Similarly you can start multiple nodes and configuration can be verified in the console.

**NODE1:**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Selenium-grid-Tutorial-9.jpg)

**NODE2:**

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/11/Selenium-grid-Tutorial-10.jpg)

**Sample Grid Code:**

Here I have used [TestNG](http://www.softwaretestinghelp.com/testng-framework-selenium-tutorial-12/" \o "TestNG Framework) to run sample GRID test case.

**Prerequisite**: Create Hub and nodes as explained earlier and TestNG should be configured in eclipse.

Here I have taken sample test to login to Gmail and enter username and password

|  |  |  |
| --- | --- | --- |
| 1 | public class GridExample { | |
| 2 | @Test |

|  |  |  |
| --- | --- | --- |
| 3 | public void mailTest() throwsMalformedURLException{ | |
| 4 | DesiredCapabilities dr=null; |

|  |  |  |
| --- | --- | --- |
| 5 | if(browserType.equals("firefox")){ | |
| 6 | dr=DesiredCapabilities.firefox(); |

|  |  |
| --- | --- |
| 7 | dr.setBrowserName("firefox"); |
| 8 | dr.setPlatform(Platform.WINDOWS); | |

|  |  |  |
| --- | --- | --- |
| 9 | }else{ | |
| 10 | | dr=DesiredCapabilities.internetExplorer(); | |

|  |  |
| --- | --- |
| 11 | dr.setBrowserName("iexplore"); |
| 12 | dr.setPlatform(Platform.WINDOWS); | |

|  |  |
| --- | --- |
| 13 | } |
| 14 | RemoteWebDriver driver=newRemoteWebDriver(new    URL("http://localhost:4444/wd/hub"), dr); | |

|  |  |
| --- | --- |
| 15 | driver.navigate().to("http://gmail.com"); |
| 16 | driver.findElement(By.xpath("//input[@id='Email']")) .sendKeys("username"); | |

|  |  |  |
| --- | --- | --- |
| 17 | driver.findElement(By.xpath("//input[@id='Passwd']")) .sendKeys("password"); | |
| 18 | driver.close(); |

|  |  |
| --- | --- |
| 19 | } |

As in the example you have to use RemoteWebDriver if you are using GRID and you have to provide capabilities to the browser. You have to set the browser and platform as above.

In this example I have used platform as WINDOWS. You can use any platform as per your requirement.

Version of browser can also be set by using dr.setVersion(“version”)

For Instance you need to run this test serially in multiple browsers you have to configure your testng.xml .Below is the testng.xml suite for above test to run your test serially.

|  |  |
| --- | --- |
| 1 | <?xml version="1.0" encoding="UTF-8"?> |
| 2 | <suite name="GRID SAMPLE TEST" thread-count="2"> | |

|  |  |  |
| --- | --- | --- |
| 3 | <test name="GRID TEST WITH SERIAL EXECTION WITH BROWSER IE"> | |
| 4 | <parameter name ="browserType" value="IE"/> |

|  |  |
| --- | --- |
| 5 | <classes> |
| 6 | <class name ="GridExample"/> | |

|  |  |  |
| --- | --- | --- |
| 7 | </classes> | |
| 8 | </test> |

|  |  |  |
| --- | --- | --- |
| 9 |  | |
| 10 | | <test name="GRID TEST WITH SERIAL EXECTION WITH BROWSER FF "> | |

|  |  |  |
| --- | --- | --- |
| 11 | <parameter name ="browserType" value="firefox"/> | |
| 12 | <classes> |

|  |  |  |
| --- | --- | --- |
| 13 | <class name ="GridExample"/> | |
| 14 | </classes> |

|  |  |  |
| --- | --- | --- |
| 15 | </test> | |
| 16 | </suite> |

To run the test parallel, you have to change your testng.xml like below.

|  |  |
| --- | --- |
| 1 | <?xml version="1.0" encoding="UTF-8"?> |
| 2 | <suite name="GRID SAMPLE TEST" parllel="tests" thread-count="3"> | |

|  |  |  |
| --- | --- | --- |
| 3 | <test name="GRID TEST WITH SERIAL EXECTION WITH BROWSER FF"> | |
| 4 | <parameter name ="browserType" value="firefox"/> |

|  |  |
| --- | --- |
| 5 | <classes> |
| 6 | <class name ="GridExample"/> | |

|  |  |  |
| --- | --- | --- |
| 7 | </classes> | |
| 8 | </test> |

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | <test name="GRID TEST WITH SERIAL EXECTION WITH BROWSER IE"> | | |
| 10 | | <parameter name ="browserType" value="IE"/> |

|  |  |
| --- | --- |
| 11 | <classes> |
| 12 | <class name ="GridExample"/> | |

|  |  |  |
| --- | --- | --- |
| 13 | </classes> | |
| 14 | </test> |

|  |  |
| --- | --- |
| 15 | </suite> |

Here in the testng.xml you have to specify parameter as*parllel=“tests” and thread-count=“3”*describes the maximum number of threads to be executed in parallel.

**Configuration using JSON file:**

Grid can also be launched along with its configuration by using json configuration file.

Create a json file for having below configuration. Here I have created a json file named as grid\_hub.json

|  |  |
| --- | --- |
| 1 | { |
| 2 | "host": null, | |

|  |  |
| --- | --- |
| 3 | "port": 4444, |
| 4 | "newSessionWaitTimeout": -1, | |

|  |  |
| --- | --- |
| 5 | "servlets" : [], |
| 6 | "prioritizer": null, | |

|  |  |  |
| --- | --- | --- |
| 7 | "capabilityMatcher":   "org.openqa.grid.internal.utils.DefaultCapabilityMatcher", | |
| 8 | "throwOnCapabilityNotPresent": true, |

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | "nodePolling": 5000, | | |
| 10 | |  |

|  |  |  |
| --- | --- | --- |
| 11 | "cleanUpCycle": 5000, | |
| 12 | "timeout": 300000, |

|  |  |  |
| --- | --- | --- |
| 13 | "maxSession": 5 | |
| 14 | } |

**Start the hub using below command**

*java -jar selenium-server-standalone-2.41.0.jar -role hub –hubConfig grid\_hub.json*

Similarly create different json file for different nodes as per required configuration.

Here is an example of json configuration file for node named as grid\_node.json

|  |  |
| --- | --- |
| 1 | { |
| 2 | "capabilities": | |

|  |  |
| --- | --- |
| 3 | [ |
| 4 | { | |

|  |  |  |
| --- | --- | --- |
| 5 | "browserName": "chrome", | |
| 6 | "maxInstances": 2 |

|  |  |  |
| --- | --- | --- |
| 7 | }, | |
| 8 | { |

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | "browserName": "firefox", | | |
| 10 | | "maxInstances": 2 |

|  |  |  |
| --- | --- | --- |
| 11 | }, | |
| 12 | { |

|  |  |  |
| --- | --- | --- |
| 13 | "browserName": "internet explorer", | |
| 14 | "maxInstances": 1 |

|  |  |  |
| --- | --- | --- |
| 15 | } | |
| 16 | ], |

|  |  |  |
| --- | --- | --- |
| 17 | "configuration": | |
| 18 | { |

|  |  |  |
| --- | --- | --- |
| 19 | "nodeTimeout":120, | |
| 20 | "port":5555, |

|  |  |
| --- | --- |
| 21 |  |
| 22 | "hubPort":4444, | |

|  |  |  |
| --- | --- | --- |
| 23 | "hubHost":"localhost", | |
| 24 |  |

|  |  |  |
| --- | --- | --- |
| 25 | "nodePolling":2000, | |
| 26 |  |

|  |  |  |
| --- | --- | --- |
| 27 | "registerCycle":10000, | |
| 28 | "register":true, |

|  |  |  |
| --- | --- | --- |
| 29 | "cleanUpCycle":2000, | |
| 30 | "timeout":30000, |

|  |  |  |
| --- | --- | --- |
| 31 | "maxSession":5, | |
| 32 | } |

|  |  |
| --- | --- |
| 33 | } |

**To start the node**

*java -jar selenium-server-standalone-2.41.0.jar -role rc –nodeConfig grid\_node.json*

You can change all the configuration of browser, maxInstances, port, maxSession etc. in the json file.

You can provide browser version, platform in the json config file like below

*{*  
*“browserName”: “chrome”,”version”:”8”,”platform”:”Windows”*  
*}*

**Conclusion:**

It is advisable to **use Selenium Grid when you have to perform multi browser testing and you have large number of test cases.**

In this module we covered how to setup Grid hub and nodes along with how to run Grid test cases using testng.xml and json file.