**What is need of Selenium Grid?**

**Selenium Grid –  a distributed test execution environment to speed up the execution of a test pass.**

**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.**

**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.

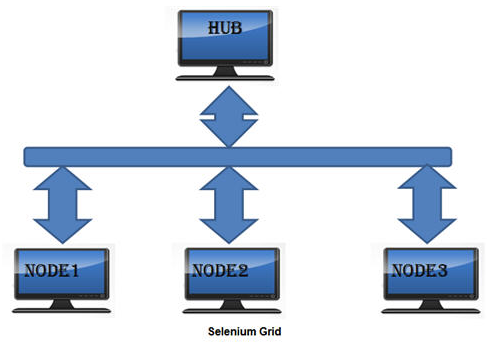
**Benefits of Selenium Grid:**

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.

****

### ****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**

**http://localhost:4444/grid/console --- To check the node status**

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:

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

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

[http://localhost:4444/grid/register -port 5556](http://localhost:4444/grid/register%20-port%205556)

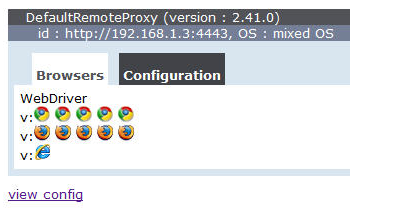
Run the selenium server in other machines/another command prompt to start nodes.



### ****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.



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

Example :

**package** com.parallel.execution;

**import** java.net.MalformedURLException;

**import** java.net.URL;

**import** org.openqa.selenium.Platform;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.remote.DesiredCapabilities;

**import** org.openqa.selenium.remote.RemoteWebDriver;

**public** **class** Gr

{

**public** **static** **void** main(String[] args) **throws** Exception

{

DesiredCapabilities caps = DesiredCapabilities.*firefox*();

//caps.setVersion("20");

caps.setPlatform(Platform.*WINDOWS*);

URL urlHub = **null**;

**try**

{

urlHub = **new** URL("http://localhost:4444/wd/hub");

}

**catch** (MalformedURLException e)

{

e.printStackTrace();

}

RemoteWebDriver driver = **new** RemoteWebDriver(urlHub, caps);

driver.navigate().to("http://selenium-suresh.blogspot.com");

Thread.*sleep*(2000);

System.*out*.println(driver.getTitle());

driver.quit();

}

}

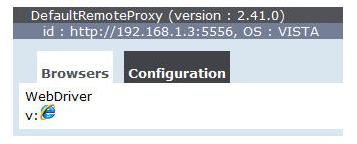
-->Run this program you will be able to see that Firefox browser has been opened.

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

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

<http://localhost:4444/grid/register> -port 5556 -browser browserName=iexplore

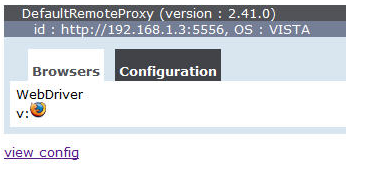
Verify the browser Type along with other details in GRID Console by clicking on view config.



**Similarly for Firefox:**

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

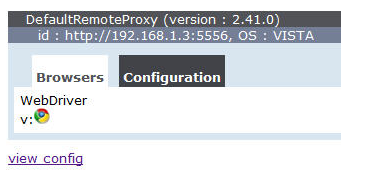
<http://localhost:4444/grid/register> -port 5556 -browser browserName=firefox



**For Chrome:**

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

<http://localhost:4444/grid/register> -port 5556 -browser browserName=chrome



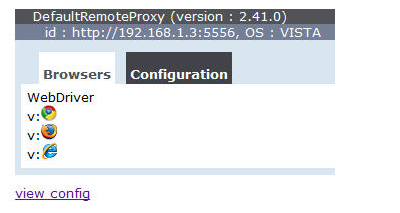
There are few scenarios where you may need browser from each type i.e.: IE, Chrome and Firefox.

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

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

<http://localhost:4444/grid/register> -port 5556 -browser browserName=iexplore

-browser browserName=firefox -browser browserName=chrome

****

### ****Sample Grid Code:****

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

public class GridExample {

          @Test

          public void mailTest() throws MalformedURLException{

                   DesiredCapabilities dr=null;

                   if(browserType.equals("firefox")){

                   dr=DesiredCapabilities.firefox();

                   dr.setBrowserName("firefox");

                   dr.setPlatform(Platform.WINDOWS);

                   }else{

                            dr=DesiredCapabilities.internetExplorer();

                            dr.setBrowserName("iexplore");

                            dr.setPlatform(Platform.WINDOWS);

                   }

                   RemoteWebDriver driver=new RemoteWebDriver(new     URL("<http://localhost:4444/wd/hub>"), dr);

                   driver.navigate().to("[http://gmail.com](http://gmail.com/)");

                   driver.findElement(By.xpath("//input[@id='Email']")) .sendKeys("username");

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

                   driver.close();

}

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”)

**Serially Execution in multiple browsers**

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.

<?xml version="1.0" encoding="UTF-8"?>

<suite name="GRID SAMPLE TEST" thread-count="2">

    <test name="GRID TEST WITH SERIAL EXECTION WITH BROWSER IE">

    <parameter name ="browserType" value="IE"/>

        <classes>

            <class name ="GridExample"/>

        </classes>

    </test>

    <test name="GRID TEST WITH SERIAL EXECTION WITH BROWSER FF ">

    <parameter name ="browserType" value="firefox"/>

        <classes>

            <class name ="GridExample"/>

        </classes>

    </test>

</suite>

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

<?xml version="1.0" encoding="UTF-8"?>

<suite name="GRID SAMPLE TEST" parllel="tests" thread-count="3">

    <test name="GRID TEST WITH SERIAL EXECTION WITH BROWSER FF">

    <parameter name ="browserType" value="firefox"/>

        <classes>

            <class name ="GridExample"/>

        </classes>

    </test>

    <test name="GRID TEST WITH SERIAL EXECTION WITH BROWSER IE">

    <parameter name ="browserType" value="IE"/>

        <classes>

            <class name ="GridExample"/>

        </classes>

    </test>

</suite>

Here in the testng.xml you have to specify parameter asparllel=“tests”andthread-count=“3”describes the maximum number of threads to be executed in parallel.