Level 5

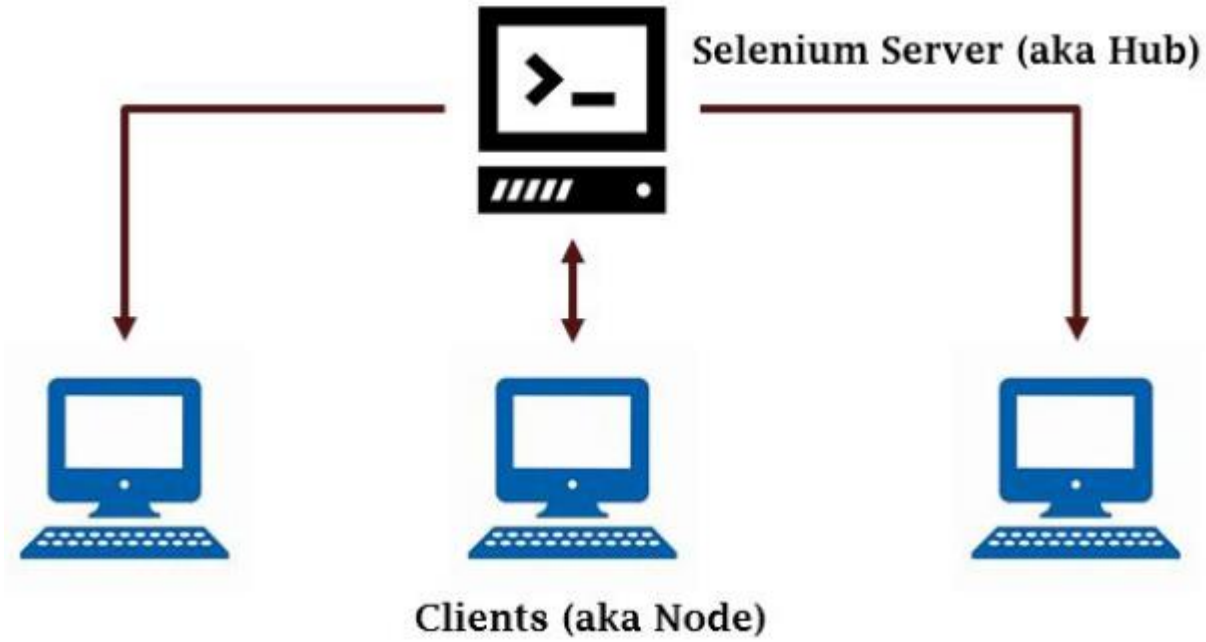
Parallelization

5.1 - ¿What is Selenium Grid?

Selenium Grid allows us to run our test cases in different nodes in parallel, either in one or in different browsers. With this, we can run multiple test cases in parallel, reducing execution times linearly.

5.2 - Configuring Nodes

In order to run our tests in parallel, we need to configure first a server, and then the nodes that are waiting for the execution to run the test cases. We can have a server and as many connected nodes as we want.



5.2.1 - Configuring the Server

The first thing we have to do is download Selenium from here:

http://docs.seleniumhq.org/download/

That will download a file named:

selenium-server-standalone-X.X.X.jar

Once the file is downloaded, from a console, standing in the folder where the file is located, we run the command:

java -jar selenium-server-standalone-X.X.X.jar -role hub

This will give us a console output indicating that the hub is running.

This console output will give us some interesting information:

10:23:39.691 INFO - Nodes should register to http://192.168.58.1:4444/grid/register/

This line tells us that the nodes must register to that address.

5.2.2 - Configuring Nodes

Let’s now configure the nodes:

In another PC (it may be a virtual one) or in the same PC where the server is located, we are going to download the same file, and we will execute it indicating that its role will be a node, and that the server is in the previously mentioned URL:

java -jar selenium-server-standalone-3.0.1.jar -role node - hub http://localhost:4444/grid/register

As we can see, we indicate him that his role is a node and the server URL

On another PC we will repeat the command, thus we register another node. If you choose to use the same PC, you have to change the port since there is already a node listening in the default port. This is done by adding **-port XXXX** at the end of the command.

On another PC we will repeat the command, thus we register another node. If you choose to use the same PC, you have to change the port since there is already a node listening in the default port. This is done by adding -port XXXX at the end of the command.

We now go to a browser and type in the following URL:

http://localhost:4444/grid/console

Where localhost: 4444 is the server's IP and port. There we should see an image as the following:

This page shows the active nodes, as well as the capacity of each one of them, in this case it indicates that each of them is capable of running 5 Firefox, 5 Chromes and 1 Explorer.

We already have two nodes up and ready to run test cases!

5.3 - Pointing to our Server

Running our cases on selenium grid is very simple: with only a couple of changes in our framework we can achieve it. We will modify the MyDriver.java class as follows:

```java
public class MyDriver {

    private WebDriver driver;

    public MyDriver(String browser) {
        switch (browser) {
        case "remoteFirefox":
            try {
                driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"),DesiredCapabilities.firefox());
            } catch (MalformedURLException e) {
                e.printStackTrace();
            }
        case "firefox":
            driver = new FirefoxDriver();
            break;
        case "chrome":
            System.setProperty("webdriver.chrome.driver", "/Users/santiago.hernandez/Documents/chromedriver");
            driver = new ChromeDriver();
            break;
        default:
            break;
        }
    }

    public WebDriver getDriver() {
        return this.driver;
    }
}
```

As we can see, now if in the parameters of the suite, we pass **“remoteFirefox”** the driver will run against the previously configured grid.

Note that in this case the URL is hard-coded, and that the way to go here would be to have it as another parameter in the suite, but for practical purposes we are going to have it that way.

5.4 - Parallelizing our Tests

We already have our test framework pointing to our grid, now we only have to configure testng so that it runs the test cases in parallel.

This configuration will be done in the Suite.xml file, where we will indicate that the run will be parallel and the number of threads to be used.

Let’s see how to do so.

```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >

<suite name="My test suite" verbose="1" parallel="methods" thread-count="5" >
    <parameter name="browser" value="chrome" />
    <test name="My first test">
        <classes>
            <class name="com.automation.training.tests.WikiTests"></class>
        </classes>
    </test>
</suite>
```

As we can see, we tell TestNG that we want to run the methods in parallel, and that the number of threads to be picked up will be 5.

Parallel can have different values, such as:

- Methods
- Tests
- Classes
- Instances

Summary of the Level

In this level we have learned how we can run our test cases in parallel using Selenium Grid.