



212. What is Selenium Grid? Its advantages on bringing down execution time

What is Selenium Grid?

Selenium Grid is a smart proxy server that makes it easy to run tests in parallel on multiple machines.

So there is one disadvantage earlier when we ran three tests in parallel, all the three tests triggered in the same machine.

what if you have 15 test cases and if you trigger all them parallelly?

what if all start on the same machine? Your machine will run out of memory.

 when 15 Chrome browsers open at a time to execute? So then obviously your system will eat up. Resources and test will start failing because of slowness in your system.(flakiness error)

So instead, those 15 test cases have about distributing them into multiple machines, like running three per machine.

So that's where selenium grid comes into picture. It will help you to run your test in multiple missions. So that means you will control your execution from one single machine only.

so you will just say run this test, but you automation will distribute the execution.

So how does Grid works internally to distribute your execution from your local system to different machine?

And it should also report as the output right, whether a tester pass, it should consolidated again output from different machines, and it should report in your eclipse console.

But overall, from high level, the hub could be in your local system like where you are running and writing. You are selenium code so that you can treat it as a hub. You can start this execution system in your local machine.

And when you run your test, you are a test will first reach to this hub.

And then this execution system hub will decide on how to distribute tests in to different machines.

different physical machines, we call them as nodes.

So everything this exclusion system can take it at runtime.

It will read script properties to know what browser you want to run and what operating system you want to run.

there are a lot of components inside the hub.

So client is nothing but you a local system, selenium java, which sends the request.

So here we are executing and when we are when you run your test first, a request will go to this router.

So there is one processor internally in this grid, you can call it as a processor or component. router will receive you a request and then it will immediately ask a distributor to create a new session and assign that to the one physical mission.

So all your requests first router only will receive if it is first time request.

If Root Router is receiving brand new request for a first time, it does not know which physical machine it has to root.

so basically router job here is to router that physical machines. But first time we don't know to which physical machine I want to root.

So what it does is it sent its request to distributor. So this distributor is a guy who will read about a script of properties and understand what is the right physical mission for you to execute.

So it will check what operating system you are expecting, and it will see what physical node missions it have to send a request out of our browser to about executing out of what is a common windows and Chrome. Then it will see like load balancing, like what his machines are already occupied with test and what else I have to say. So like that distributor do?

Well, some work.

what distributor does is it well, I'll send a request to node to execute that particular

client code That means a physical machine.

So once it sends a session to one physical machine to execute at the same time, what distributor does is it will send information to session map. Also, it will tell you that, Hey, I received one session. This is the Session HD. So whenever you create a new session, every time you are new, session will have an I.D..

So this session map guy will store the seesion information, and distributor will also give that on which physical machine I assign the session.

So duty of session map here is to simply create one map, which consists of what test it is executing and in what physical machine it is. Executing that information, it will store.

Session map – The test which is executing is nothing but session ID and in the physical mission, which it is executing nothing but node URL. That information it will store here now.

Java file there 10 line of selenium code write first line of selenium code will get a request.

And then all of this distributor will see and some physical machine next subsequently.

If you are getting the code from the same session, from same test.

what does router do?

Is again, it will not ask the distributor to figure out which physical mission because distributor

is already figured it out earlier for that test, where to execute, which mission it execute.

that information is already there in seesion map so far a subsequent request.

What does router do is it will simply talk to session map. Hey, I got another session.

If it is already existing session, well, yeah, it's already existing test.

And what is the physical mission? OK, this is IP address. OK, let me route it.

So it will directly related to that particular physical machine.

Only for the 1st time it will take the distributor help

So that's how internally your grid is design.

create one hub and let's create some nodes, which are

physical machines, and we will register them and we will see how the real time execution happens by running in the different machines.

213. Getting Started with Grid Infrastructure Setup – Create components

[Grid | Selenium](https://www.selenium.dev/documentation/grid/)

Hub & Nodes

There is one command which will trigger you to start the hub. So once you start the hub,

automatically the components inside this hub will start automatically.

hub is the union of the following components.

So it is a set of Router, which will take your incoming request from your Selenium test;

Distributor, which decides which node it has to execute;

Session Map, which can store the session ID and also the node physical machine address mapping related.

New Session Queue - when you initiate multiple sessions from your Eclipse Selenium(client)

distributor can solve only one session at a time, so all the remaining sessions

which are waiting for distributor to check,they will go in the queue. So this comes b/w distributor and router. There will be another component. So that component will actually store

all the new sessions in a queue. So that distributor, one by one, will retrieve the item from the session queue and it resolve to check to which physical machine it has to provide.

Event bus - is a virtual here. So all this communication, whatever you are seeing in the arrows, this is all, they're communicating internally. So they are making communication through event bus channel here. So to enable this communication, there is another component that also you have to start.

So all this, separately, you have to start if you choose to set up Selenium Grid in your system

on Distributed mode.

before you set up all this Hub and Nodes, first you need to download one separate Selenium jar which supports this Selenium Grid architecture.

Download the Selenium Server

Download Browser drivers and place in the same path where Selenium server is located

Open the command prompt – cd Grid

Once your on the path

Start the Hub - which eventually Starts Router, Distributor, Session Map , New Session Queue, Event Bus

• java -jar <seleniumjarname> hub

Check the status of running hub – localhost:4444

How would I know that my Grid Hub started at this port number?"- Simply read your log.

Without registering your node to this hub, your hub will never know what nodes it has to delegate. goal is to start a node and register to your hub, and then hub will start listening to that node, and whenever you receive any request, then it will assign it to that node physical machine.

Hub and Node can be your local system also. So in the same system you started Hub,

and Node also you can start in the same system. So that means when hub receives request,

then it will assign to the node, which is, again, your machine only.

Hub and Node can be in the same machine also. A hub can act, also node, to execute test.

First, I'll tell you how to start node in this local system only.

Start the Node in Same Machine where Hub is running

java -jar <seleniumjarname> node --detect-drivers true

214. Create Node Physical machines and register with Grid-Check Grid health

I have opened another physical machine, and I connected to that machine through TeamViewer.

now I'm going to start another node in different machine.

So you have to repeat some things in the node machine before you start the node.

Download the Selenium Server

Download Browser drivers and place in the same path where Selenium server is located

So you have to repeat the step in all the machines where those machines are acting

as node because those machines are ultimately responsible to execute your test cases,

 if you don't have browser at all in that machine how does test execute there? -no it won’t

215. Create Selenium TestNG tests with Desired capabilities & remote webdriver class

 we created hub which will automatically start session map, distributor, router.

So these three are already started when you have created a hub

ll’y we created two nodes, one node we started in the local system and one node we created in another physical machine.

So these are the two nodes and we have registered these two nodes to our hub.

So we have given the enough commands to register with subscribe events and publish events.

So one of the node is installed in the same machine where hub is there. So that's why there we don't need to give any publish and subscribe events.

Right click->new->java project->next->seleniumGrid ->Finish

add selenium JARs into this project

rightclick on the project->properties->java buildpath-.libraries-under class path->slect add external jars->add the selenium jar->Apply&close

create one new class(GoogleTest)

whenever you are running using SeleniumGrid, there is a different way of initializing your driver. So you have to use the class called new RemoteWebDriver.

 RemoteWebDriver class take two arguments.One argument is the information

where your hub is listening, So that's where you have to send this request. So you have to make a request to your router.

when you are dealing with this RemoteWebDriver class, you have to create one more class

called DesiredCapabilities. Like what browser you want to execute, everything

and finally that object you are going to send as a 2nd argument to your RemoteWebDriver class.

On last just convert the project in to testNG.

216. Demo on executing the Selenium scripts in Node machines from Grid hub

we are asking them to run parallelly. So that means two sessions will start parallelly but both these sessions will report to this IP address.

As these are the initial requests, so router cannot get that information from session map

to send it to the nodes to execute. So it will give it to the distributor. Now, distributor will open this capability object. and it will see that, one of the session is requesting to run on Chrome."

real time, all the hub process can be in one machine. You're Eclipse Java code can be another machine and your nodes can be again, another physical machines.

See, no matter from which machine you are running your test, ultimately, you are asking to reach this hub IP address.

So it doesn't matter if the hub and your actual client is in same machine or a different machine.

Big Projects - they create multiple nodes. They'll bring at least five to six physical machines.

In every physical machine, they will go and set up the node just like how we did now through TeamViewer.

You just need to ask your management, "Give me four or five node machines because I want to run my Selenium test parallelly in multiple machines so that way we will get faster feedback."

If you run them sequence in one machine, it will take two, three hours to get result.

But if you distribute them running in five to 10 machines, just imagine, the results will be out in 10 minutes.

So that's how you need to talk to your management. Get at least five different servers,

five different physical servers, login into them and set up node in all those machines.

And then set up hub in one system. Maybe you can do in your local system also and just trigger. That's it.

Just create all the desired capabilities and route them to your main hub and bring all tests parallely in your TestNG and trigger it from here. That's it.

So all the tests will run parallely in all different physical machines. And it will finally report

to your local machine, output like this. So that's the beauty of Selenium Grid.

Guys, so no matter you write two lines of code or 10 lines of code.