**JMeter Distributed Load Testing with an Active Firewall**

how to configure JMETER and firewall to perform distributed load testing.

Distributed load testing is a mechanism which allows us to perform very high user load performance testing using multiple systems. Few months back I had a requirement to perform stress testing using distributed mode, I have searched the same over the internet, but I did not find any single article which explained all these things step by step. Many articles were available but most of them were performing the distributed load testing by disabling the firewall. But as we all know, most of the organizations does not allow to disable their firewalls. I was looking such an article where I can find how to set up my master and slave machine with an active firewall to perform distributed load testing but unfortunately, I did not find anything. Here we will see all these things step by step.

Before doing this, first let’s understand few terminology and JMeter distributed testing architecture.

**Terminology: -**

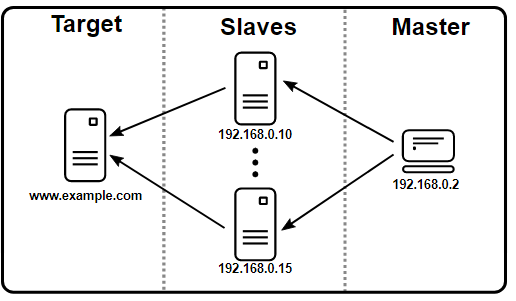
1. **Client (Master):-** The system running JMeter GUI, which controls the test.

2. **Server (Slave) :-** The system running jmeter-server, which takes commands from the client and send requests to the target system(s).

3. **Target: -** Webserver(s) or Application Under test

4. **Remote Method Invocation (RMI) :-** The RMI provides remote communication between the java applications. JMeter uses Remote Method Invocation (RMI) as the remote communication mechanism.

**JMETER Distributed Testing Architecture: -**



**Prerequisites: -**

1. All Master and Slave machines should at same subnet. RMI cannot communicate across subnets without a proxy; therefore, neither can JMeter without a proxy.

2. Make sure you use the same version of JMeter and Java on all the systems. Mixing versions will not work correctly.

**Client (Master) Engine Configuration: -**

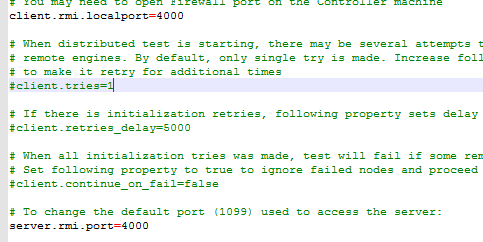
1. Open jmeter.properties present in the bin directory on the Master machine

2. Set ***server.rmi.ssl.disable=true***, since we don’t want to use SSL for RMI

3. Set ***server.rmi.port=4000***. This is the port that will be used to communicate with the server.

4. Set ***client.rmi.localport=4000***. JMeter/RMI also requires a reverse connection in order to return sample results from the server to the client.

5. remote\_hosts=X.X.X.X( Slave Machine IP address).



**Server (Slave) Engine Configuration**

1. Open jmeter.properties file present in the bin directory on the Slave machine

2. Set ***server.rmi.ssl.disable=true***, since we don’t want to use SSL for RMI

3. Set server\_port=4000.

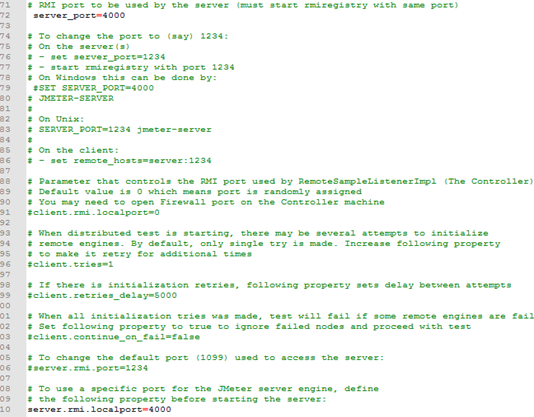
4. Set server.rmi.localport=4000.

5. We need to add JDK file path in environment variables.

6. For Setup RMI open command prompt and we have to paste

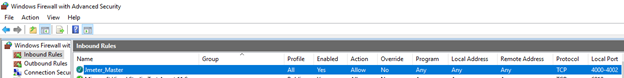
**set server\_port=4000** click on enter and next need to paste below commandand click on enter

**start rmiregistry with port 4000**



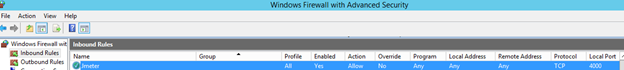
**Firewall Configurations**

**Master Machine: —**Open Windows Firewall with Advanced Security in master machine and add inbound rule for the port range 4000–4002.



Note: — If client.rmi.local port is non-zero, it will be used as the base for local port numbers for the client engine. At the moment JMeter will open up to three ports beginning with the port defined in client.rmi. local port. So, to avoid any traffic blockage by firewall, we need to open port range 4000–4002.

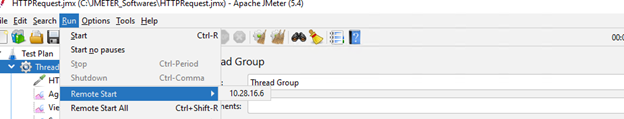
**Slave Machine: —**Open Windows Firewall with Advanced Security and add inbound rule for the port 4000. This will allow inbound traffic in the port 4000.



Note: — If you are using cloud hosted machines (Azure or AWS VMs) then allow port 4000 in azure/aws portal also.

Once all these steps are accomplished, navigate to jmeter bin folder of slave machine and start jmeter-server.bat

Now navigate to the master machine and launch the jmeter from jmeter.bat file and execute the test



**How to run scripts in Non\_GUI mode using remote testing?**

Step 1: Add JMETER\_HOME in environment variables in client and in all server machines.

Step 2: Syntax:

jmeter -n -t "Path jo JMX.jmx" -l "Path to store results.csv" -e -o "Path to store report.HTML" -r

jmeter -n -t "Path jo JMX.jmx" -l "Path to store results.csv" -e -o "Path to store report.HTML" -R server1, server2,....

Note: In csv data config, we have to give full path (not only the file name but also full path)