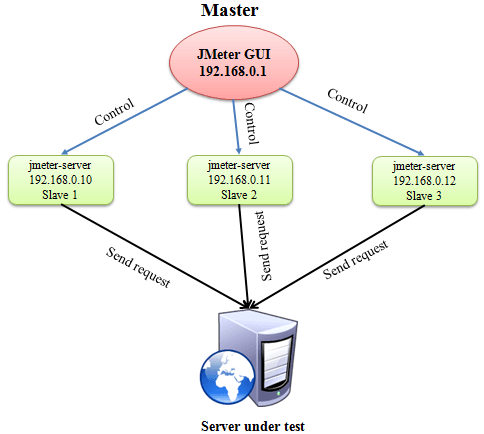
**Remote Testing:**

Distributed testing is a kind of testing which use multiple systems to perform stress testing. Distributed testing is applied for testing web sites and server applications when they are working with multiple clients simultaneously.



**Master**: the system running J Meter GUI, control each slave.

**Slave**: the system running j meter-server, receive command from the master and send a request to server under test.

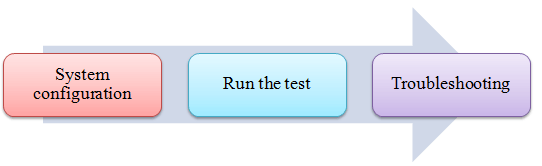
**Target**: The Web server under test, get request from slaves.

**Precondition:**

The firewalls on the systems are turned off. In some cases, the firewall may still be blocking the traffic. You should disable the Window firewall or [Linux](http://www.guru99.com/unix-linux-tutorial.html) firewall.

All the machines should be on same subnet. If machines are not on same subnet, maybe they will not recognize each other in the network.

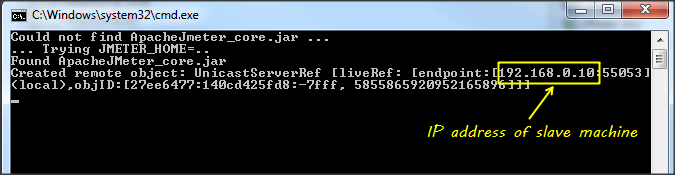
Use the same version of J Meter to avoid unanticipated errors/issues.



**Step 1) System configuration**

On the **slave** systems, go to j meter /bin directory and execute file "jmeter-server.bat".

Assume that a slave machine has IP address: 192.168.0.10. On windows, you should see a window appear like following figure:

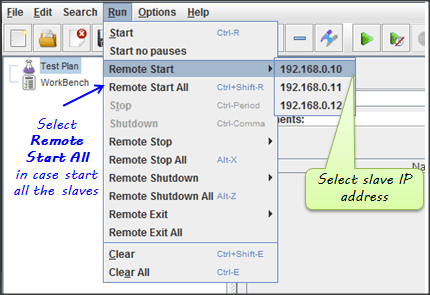


On the **master** systems, go to /bin directory and edit file j **meter .properties**, add IP slave machine as below

**Step 2) Run the test**

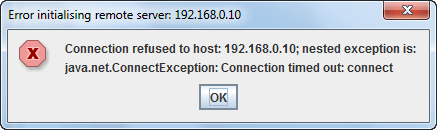
On the master machine, run J Meter GUI and open the test plan.

Click Run on the menu bar; select **Remote start** ***->***select **the IP address** of slave machine.



**Step 3) Troubleshooting**

If you are unable to run test form the above machine and see below error, simply ask owner of slave machine to run the jmeter-server.bat File.



Limitation:

There are some basic **limitations** for distributed testing. Here's list of the known items:

Server and all clients must be on **the same** subnet.

Distributed testing required target server to have large processing power. The target Server could be easily **overloaded** in case it gets too many requests by distributed J Meter tests.

A single J Meter can only handle a limited number of threads (100- 300 threads).

The distributed J Meter tests are complex, difficult for beginner to build