Using SMARTS as a Distributed System, Version 1.0

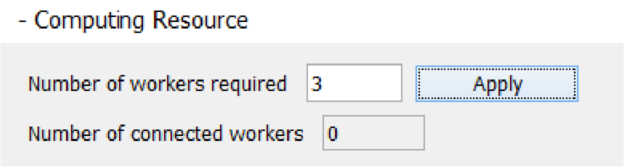
SMARTS can run as a standalone program or a distributed system. This document details how to perform distributed simulations with SMARTS.

A distributed simulation requires two types of components, **server** and **worker**. The components are compiled into **server.jar** and **worker.jar**, respectively. A distributed simulation needs to run one server instance and one or more worker instances. The components communicate using TCP. A server listens for new connection request on port 50000 by default. Theoretically, a simulation can have any number of workers. We have tested the distributed system with up to 100 workers, each of which runs on a different computing node in a research cloud.

There are two ways to run distributed simulations, depending on whether you need to visualize simulations on a Graphical User Interface (GUI) at the server. Transferring the vehicle details from the workers to the server and drawing them on the GUI can take quite a bit of time. Therefore, you may want to enable the GUI for small-scale simulations but disable it for large-scale simulations.

Running Distributed Simulations with GUI

To set up distributed simulations with GUI, you first need to launch server.jar, which will show the GUI by default. The number of workers can be specified in **Computing Resource** section in the top-right corner of the GUI. Click the **Apply** button to confirm any change of the number. In the following example, we specify that the simulation requires 3 workers. The number of connected workers is 0 in this example. This number will be incremented each time a new worker is connected.



- Computing Resource

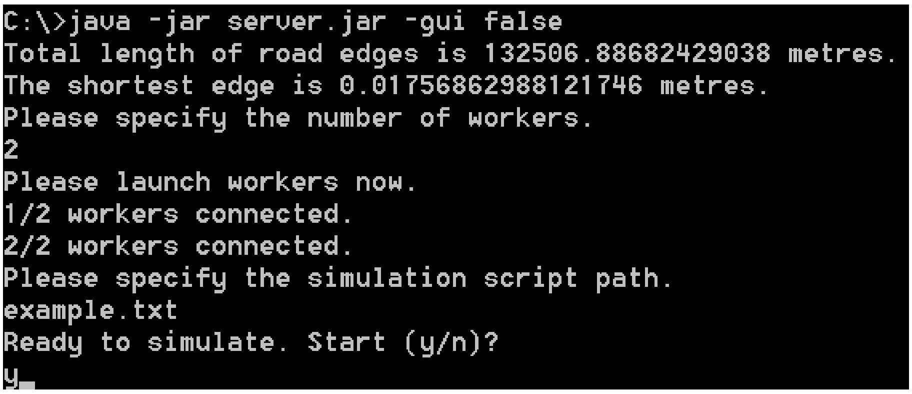
Number of workers required 3

Number of connected workers 0

Once the number of workers is specified, you can launch the worker instances. If a worker ran on the same computer as the server, you can launch worker.jar to start a worker instance without specifying the IP address of the server. If the worker ran on a different computer, you need to launch the worker from command line and provide the IP address of the server. For example, the command to launch a worker is “java -jar worker.jar 10.1.2.3”, assuming the server runs on 10.1.2.3. Once enough workers are connected, the section for configuring other simulation settings will be enabled on the GUI. You can then set up a simulation as detailed in the guide, **Using SMARTS as a Standalone Program**.

Running Distributed Simulations without GUI

To set up distributed simulations without GUI, you need to run server.jar from command line. The command line should include an option, “**-gui false**”. The server will guide you through a simple setup process as in the following example.



```
C:\>java -jar server.jar -gui false
Total length of road edges is 132506.88682429038 metres.
The shortest edge is 0.01756862988121746 metres.
Please specify the number of workers.
2
Please launch workers now.
1/2 workers connected.
2/2 workers connected.
Please specify the simulation script path.
example.txt
Ready to simulate. Start (y/n)?
y
```

- The first line shows the command that starts the server.
- The server builds a default road network when it is started. The next two lines in the screenshot show the basic statistics of the road network.
- After the default road network is built, you are prompted to enter the number of workers. In this example, the number is 2.
- The server waits for the workers to join. Once a new worker is connected to the server, the system shows how many workers have been connected. For example, "1/2 workers connected" means the number of connected workers is 1 while the system requires 2 workers.
- Once enough workers are connected with the server, you should input the path of a simulation script. A simulation script is a text file. The script can contain the setup details of an arbitrary number of simulations that will run sequentially. The script file is "example.txt" in this example. The file is in the same directory of server.jar. The format of the script is detailed in a separate document.
- As the last step, you should confirm whether to proceed with the simulations. Enter **y** to proceed with the simulations. Enter **n** to quit the system. The server and all the workers will be terminated at the hosting machines when you quit the system.
- When all the simulations specified in the script have been performed, the server will ask whether you would like to exit the program. If you choose not to exit, the server will ask you to specify the simulation script for next round of simulations.