Departament d'Enginyeria Telemàtica
entel
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Scalev v4.1
Tutorial

Luis J. de la Cruz Llopis
Esteve Pallarès Segarra
Mónica Aguilar Igartua
Isabel V. Martín Faus

SISCOM

siscom.upc.edu

Table of Contents

1. Introduction.

The objective of this tutorial is to provide users with a simple introduction to the use of the Scalev simulator version 4.1. The information provided in this tutorial is complemented by the information available in the User Manual.

Scalev allows for two types of simulations: simple simulations and simulations with sweeping of an input parameter. The following sections will study both types of simulations and analyze the generated results.

2. Executing a single simulation.

The simulator can be executed directly from the command line:

```
java -jar scalev41.jar
```

Once started, a window like the one shown in Figure 1 will appear.

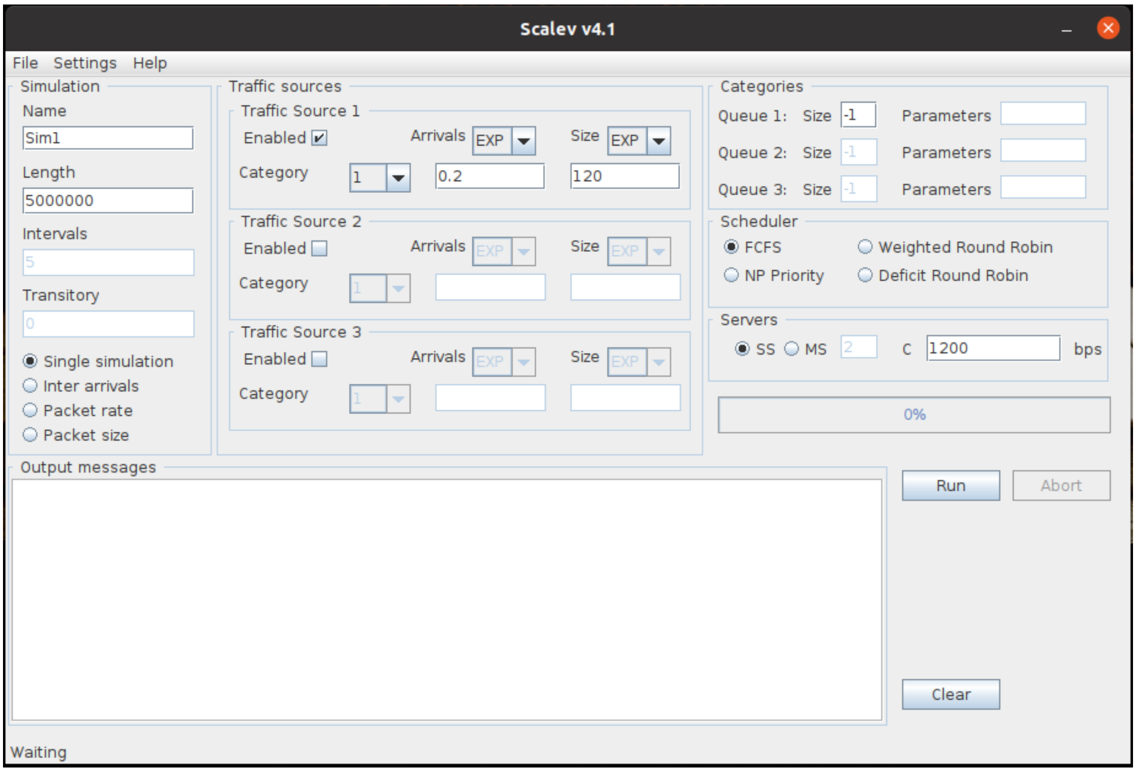

Figure 1. Scalev main window.

Using the menu option Settings → Directories will open the window shown in Figure 2, where you can select the directory and the names you want for the results files.

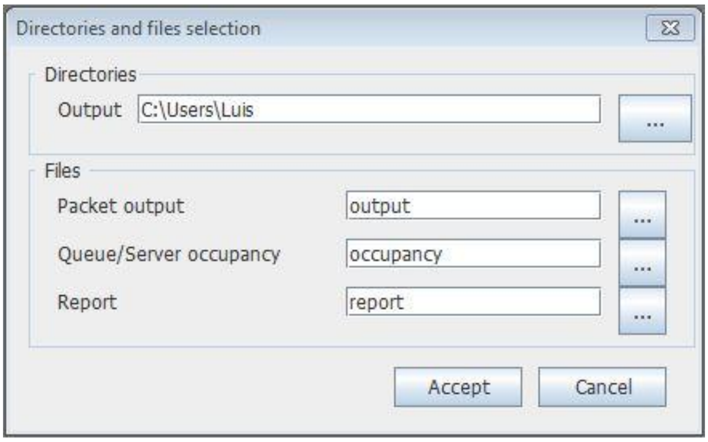

Figure 2. Directory and file names for the results files.

Next, review the default parameters that appear in the simulator's main window. Observe that only one traffic source is activated, and it is assigned to Category 1. Pay attention to the units: the mean inter-arrival time for traffic source 1 is 0.2 seconds and the mean packet size is 120 bits. Additionally, an exponential distribution has been selected for both variables. The Category 1 waiting buffer size is infinite (-1) and the selected scheduler is FCFS (First Come First Served). A single server (SS, Single Server) is used, and the channel capacity is 1200 bps. Observe that this is therefore a transmission system that can be modeled as an M/M/1 system.

Execute the simulation by clicking the Run button.

Once the simulation is finished, the following four files have been generated and saved in the selected directory (the names are the defaults if they were not changed in Settings → Directories):

- **report_Sim1.txt**: Contains the main simulation report, which provides the main mean values obtained.
- **output_Sim1_category_1.txt**: Packet output file for Category 1. Each line in this file corresponds to a packet processed by the simulator, saving the following tab-separated values:
  - Packet Identifier.
  - Arrival Time.
  - Transmission Time (Service Time).
  - Transfer Time.
  - Waiting Time.
- **output_Sim1_source_1.txt**: File similar to the previous one, with the packets from traffic source 1. In the specific case of this simulation, since there is only one traffic stream assigned to one category, the two files are identical.
- **occupancy_Sim1_1.txt**: This file is generated for each category enabled in the simulation (in this case, only Category 1). It indicates the temporal evolution of the system occupancy (queue and servers) by packets of this category. Specifically, each line of the file corresponds to an arrival or departure instant of a packet, and the information saved (tab-separated) is:
  - Time instant at which the event occurs.
  - Number of packets of this category in its waiting queue.
  - Number of packets of this category in the server or servers.

3. Analysis of Results.

3.1 Simulation report.

Open the file containing the main report. Observe and understand all the values offered in this report. Consult the Scalev 4.1 user manual, if necessary.

3.2 Packet Output File.

In MATLAB, load the packet output file into a variable:

```
>> out=load('output_Sim1_source_1.txt');
```

Observe that *out* is a matrix with 5 columns and as many rows as packets have been processed by the simulator. Display the first ten rows and all columns of this matrix:

```
>> out(1:10,:)
```

Observe the output and check that the values correspond to the parameters described previously.

Obtain the mean value of the transmission, transfer, and waiting times and compare them with those offered by the simulator report. For example, for the transmission time, first save its values, i.e., the third column of the out matrix, into a new variable:

```
>> tt=out(:,3);
```

Then obtain the mean:

```
>> mean(tt)
```

Compare this value with the one offered in the main report. If desired, increase the resolution of the values offered by MATLAB:

```
>> format long
```

Observe that in this way you can find more statistical parameters than those offered in the report, such as the variance (*var(tt)*) or the coefficient of variation (*std(tt)/mean(tt)*).

Regarding the inter-arrival time, the arrival instants are available in the second column of the file. These are the specific time instants when each packet arrived, so if you want to obtain the inter-arrival time, you need to differentiate the values that appear in that column. That is, first we obtain the arrival instant values:

```
>> ta=out(:,2);
```

and then we obtain a new vector whose values are the difference between every two consecutive values:

```
>> tia=diff(ta);
```

Obtain the mean value of tia and check that it is indeed the expected value.
Another possibility is to obtain the probability density functions (pdf) of any of the offered parameters. Although this operation can be done directly in MATLAB, we can also do it with the following steps, which helps us better understand this calculation:

```
>> t=out(:,4);
>> [ht,x]=hist(t);
>> area=(x(2)-x(1))*sum(ht);
>> pdf_t=ht/area;
>> plot(x,pdf_t)
```

If you wish to compare the obtained pdf with the theoretical one for this case (since it is an M/M/1 modelable system, the transfer time is exponential with a mean of 0.2), you can add the following:

```
>> exp_t=5*exp(-5*x);
>> plot(x,[pdf_t' exp_t'])
>> legend('simulated','theoretical');
```

3.3 System Occupancy File.

In MATLAB, load the occupancy file into a variable:

```
>> oc=load('occupancy_Sim1_1.txt');
```

Observe that oc is a matrix with 3 columns and as many rows as packet arrival or departure events have occurred during the simulation. Display the first twenty rows and all columns of this matrix:

```
>> oc(1:20,:)
```

Observe the output and check that the values correspond to the parameters described previously.

With the data stored in this file, you have the description of the system occupancy over time. A graphical representation can be obtained as follows:

```
>> t_n=[oc(:,1) oc(:,2)+oc(:,3)];
>> stairs(t_n(:,1),t_n(:,2))
```

You can also visualize only the first moments:

```
>> stairs(t_n(1:50,1),t_n(1:50,2))
```

As observed, from the data in this file, you can obtain the total time the system spends in each of the occupancy states, and therefore it is possible to obtain the probabilities of those states. Check and understand how the following script obtains and plots these probabilities:

```
oc=load('occupancy_Sim1_1.txt');
t=oc(:,1);
sim_time=t(length(t));
it=diff(t);
n=oc(:,2)+oc(:,3);
n=n(1:length(n)-1);
max_state=max(n);
states=0:max_state;
state_probs=zeros(max_state+1,1);
for i = 0:max_state
    tmp1=find(n==i);
    total_time=sum(it(tmp1));
    state_probs(i+1)=total_time/sim_time;
end
bar(states,state_probs);
```

Considering again that the simulation corresponds to an M/M/1 model, for which the expression for state probabilities is known:

$$p_k = (1-\rho)\rho^k$$

where:

$$\rho = \frac{\lambda}{\mu}$$

we can compare the state probabilities obtained through simulation with the theoretical probabilities:

```
>> rho=0.5;
>> k=[0:max_state-1]';
>> state_probs_theo=(1-rho).*(rho.^k);
>> bar(states,[state_probs state_probs_theo]);
>> legend("simulated","theoretical")
```

3.4 Unstable systems.

A system like the one simulated in the previous example, having an infinite queue, can behave unstably if the system utilization is greater than or equal to one. This fact can be checked, for example, by decreasing the inter-arrival time in the previous simulation, which would increase the offered traffic. Select a value of 0.02 seconds for the inter-arrival time, run the simulation, and observe the instability message generated by the simulator (the background color of the simulation progress bar turns red and the corresponding message appears in the Output messages textbox).

Next, modify the value of queue size 1, changing it from -1 (infinite) to any finite value (e.g., 1). In this case, the system will show packet loss but will no longer behave unstably. Run the simulation and observe that it finishes without problems.

4. Executing a simulation with parameter sweeping.

As an example of a parameter sweeping simulation, we will take the case of varying the packet rate parameter. Change the simulation name to Sim2. Select *packet rate* as the simulation type. Observe that the corresponding box changes to green. Note that in this case, packet rate values are requested, meaning packets per second (for more details, see the sweeping simulations section of the user manual).

A series of simulations will be performed, varying the arrival rate from 1 to 9 packets per second, in steps of one packet. To do this, the expression 1:9:1 (initial value: final value: increment) must be entered in the corresponding box.

Run the simulation, and observe how the simulator performs nine iterations, one for each requested value for the packet rate.

In the case of sweeping simulations, the simulator only provides mean values for the parameters of interest. These values are provided in the main report, which is now structured as a series of rows and columns. Each row corresponds to the simulation performed for each value of the variable parameter, and each column contains a result of interest (mean waiting time, mean number of elements in the system, loss probability, etc.). For a detailed description of the values offered in each column, see the user manual.

4.1 Analysis of the results report.

In MATLAB, load the packet output file into a variable:

```
>> res=load('report_Sim2.txt');
```

Observe that res is a matrix with 9 rows and 24 columns. As mentioned, each row corresponds to an iteration, meaning the simulation for each of the requested values for the variable parameter. Within each row, each of the columns provides a value of interest. For example, the first column provides the specific value of the variable parameter under study, the second provides the theoretical offered traffic value (theoretical utilization factor), and the third provides the simulated carried traffic value (simulated carried utilization factor). These last two traffic values can be graphically represented as a function of the variable packet arrival rate parameter as follows:

```
>> plot(res(:,1), [res(:,2) res(:,3)])
>> xlabel('Packet rate');
>> legend('Theoretical utilization','Simulated utilization')
```

As can be observed in the figure, both traffic values grow linearly, which is logical as the arrival rate increases. Furthermore, since this is a lossless system, the offered and carried traffic values coincide. Remember that the values for the offered traffic are calculated by the simulator based on theoretical expressions, while the values for the carried traffic are obtained from the simulation.

Other possible graphical representations that can be obtained from this simulation are the following:

- Mean number of waiting packets as a function of the offered traffic:

```
>> plot(res(:,2),res(:,17),'k-*')
>> xlabel('Offered Traffic');
>> ylabel('Waiting packets');
```

- Mean waiting time as a function of the offered traffic:

```
>> figure
>> plot(res(:,2),res(:,7),'b:+')
>> xlabel('Offered Traffic');
>> ylabel('Waiting Time');
```

These last two representations can also be compared with the known theoretical values for the M/M/1 queueing system.