COMPUTER SYSTEM MODELING AND SEMANTIC WEB

FINAL PROJECT REPORT

Amin Entezari (B34000068)
Wisemen Tinotenda J Chingombe (B34000075)

Report

ASSIGNMENT 2024/2025 – COMPUTER SYSTEMS MODELING AND S. W.

Performance of a big.LITTLE architecture

A system characterized by a big.LITTLE architecture is characterized by H high performance cores, and E energy efficient cores. It is used by N_B heavy computation tasks, and N_L low computation tasks.

The scheduler will mainly schedule heavy computation tasks on the high performance cores, while low computation tasks on the energy efficient cores. However, to better use the resources, there is also a small probability that tasks will be assigned the other way round.

The execution times of the tasks on the cores are collected in the following traces (all expressed in sec):

	High Performance Cores	**Energy Efficient Cores**
Heavy computation tasks	TraceB-HH.txt	TraceB-HE.txt
Low computation tasks	TraceB-LH.txt	TraceB-LE.txt

The system is also composed by an I/O subsystem, a Storage component and Network access. All these components can be considered working in processor sharing, with an exponential service time (different per type of job), whose average is described in these tables:

	I/O	Storage	Network
Heavy computation tasks	50 msec	200 msec	5 msec
Low computation tasks	150 msec	10 msec	120 msec

All tasks will go through a computing, a storage and either an I/O or a network communication phase.

Determine the best assignment probability distribution: test a few alternatives of probabilities of assigning a heavy computation task to an efficiency core and of assigning low computation task to a high performance core. Determine the system throughput in each scenario.

The given parameters are:

H	E	NB	NL
8	8	16	64

SOLUTION

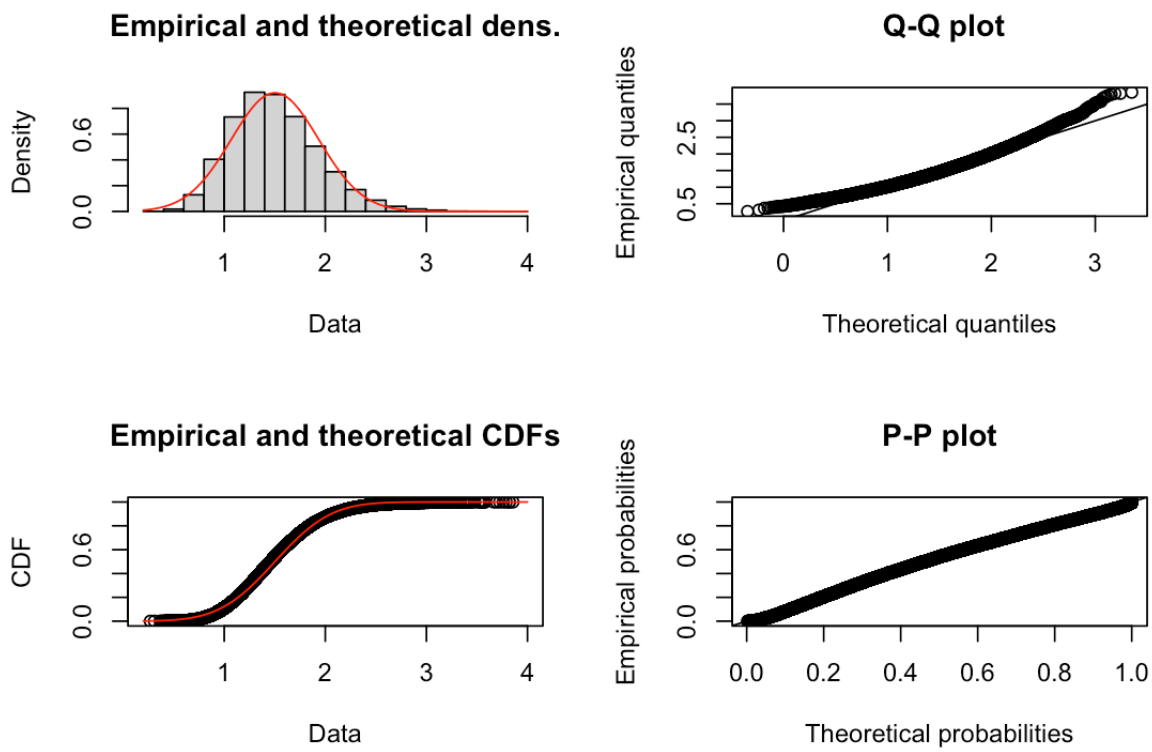
The assignment focus is the evaluation of the performance of a big.LITTLE architecture system, which includes high-performance and energy-efficient cores. A scheduler will primarily assign heavy computation tasks to high-performance cores and low computation tasks to energy-efficient cores, with a small probability of cross-assignment. With the aid of Java Modeling Tools (**JMT**) we want to model a system that incorporates the processing stages in the problem description and then evaluate system performance under a varied set of conditions paying particular attention to system throughput in each scenario.

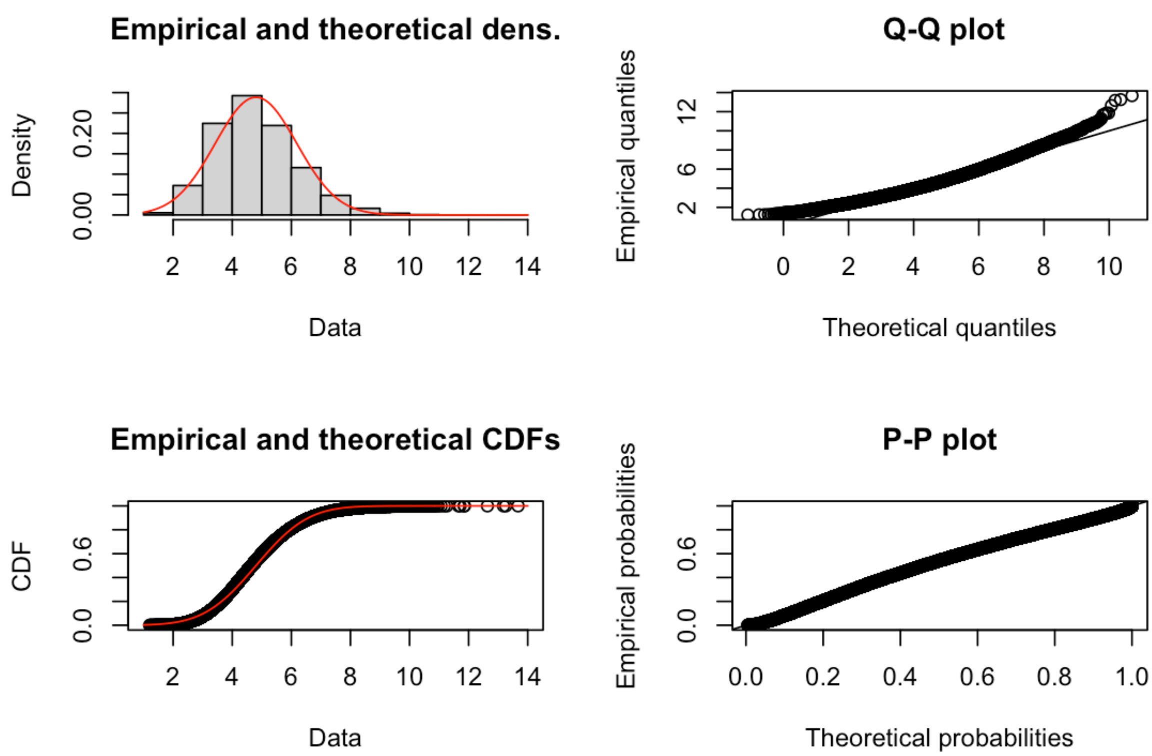
PRELIMINARY CONSIDERATIONS & MODEL

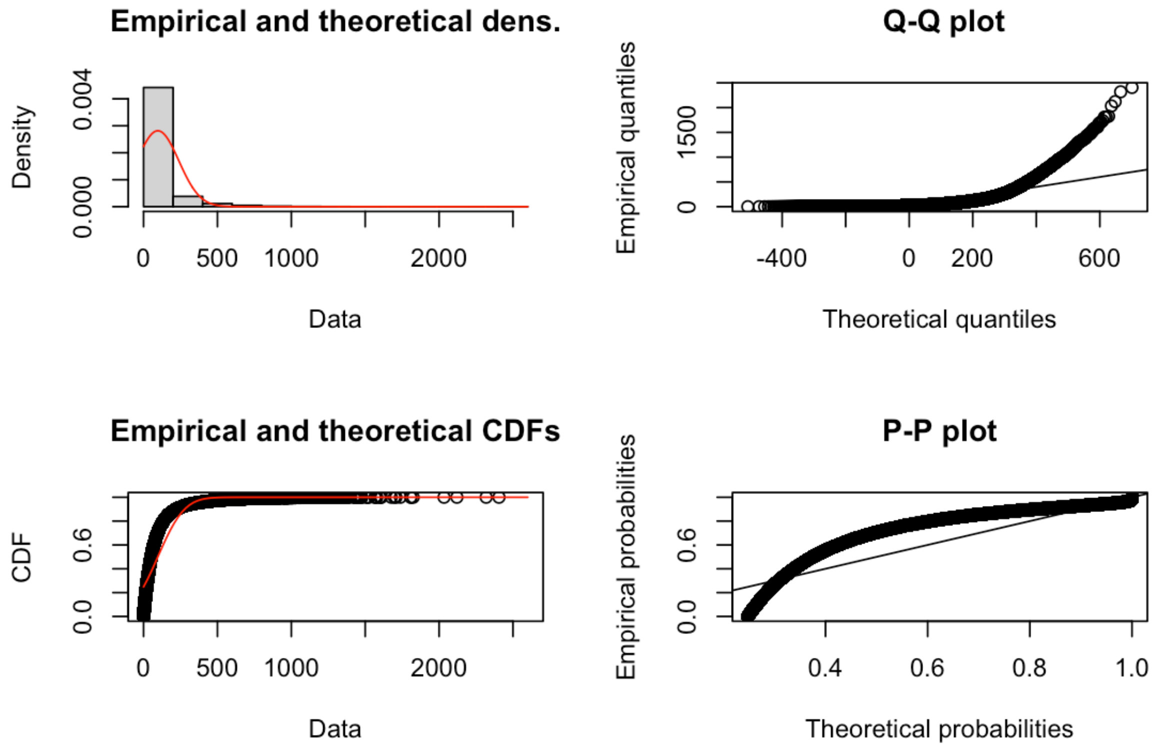
- We began by computing the coefficients of variation in R from values that were provided by the 4 trace files. We used these to guide our choice to proceed with the distributions for the model for Heavy and Low performance tasks respectively

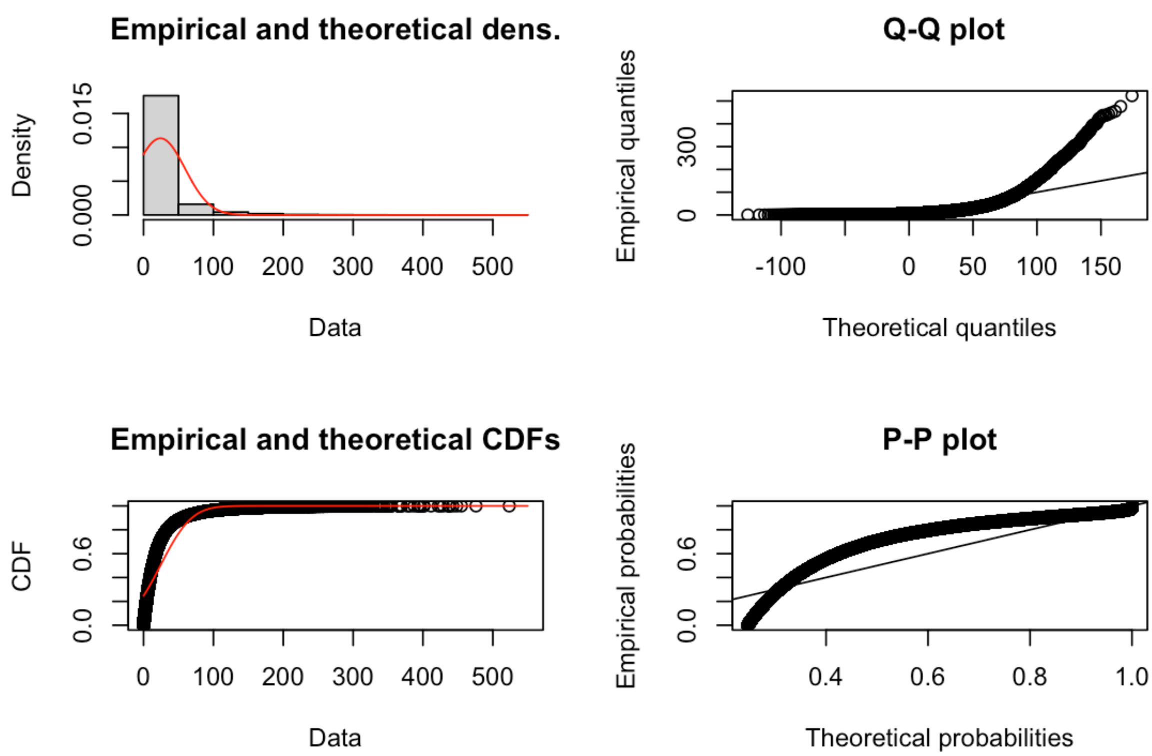
Data	
TraceB.HE	50000 obs. of 1 variable
TraceB.HH	50000 obs. of 1 variable
TraceB.LE	50000 obs. of 1 variable
TraceB.LH	50000 obs. of 1 variable
Values	
coefficient_h_e	1.4683071189541
coefficient_l_e	0.287843106155675
cofficient_h_h	1.45887646970379
cofficient_l_h	0.288480101740655
mean_h_e	96.5605314108
mean_h_h	24.1255126568
mean_l_e	4.8041312566
mean_l_h	1.5024967794
sd_h_e	141.780515680469
sd_h_h	35.1961427345465
sd_l_e	1.38283606327931
sd_l_h	0.433440423786319

Figure 11 – calculations from R.Studio


Empirical and theoretical dens.
Density
Data
Q-Q plot
Empirical quantiles
Theoretical quantiles
Empirical and theoretical CDFs
CDF
Data
P-P plot
Empirical probabilities
Theoretical probabilities


Empirical and theoretical dens.
Density
Data
Q-Q plot
Empirical quantiles
Theoretical quantiles
Empirical and theoretical CDFs
CDF
Data
P-P plot
Empirical probabilities
Theoretical probabilities


Empirical and theoretical dens.
Density
0.000
0.004
0
500
1000
2000
Data
Q-Q plot
Empirical quantiles
0
1500
-400
0
200
600
Theoretical quantiles
Empirical and theoretical CDFs
CDF
0.0
0.6
0
500
1000
2000
Data
P-P plot
Empirical probabilities
0.0
0.6
0.4
0.6
0.8
1.0
Theoretical probabilities


Empirical and theoretical dens.
Density
0.000
0.015
0
100
200
300
400
500
Data
Q-Q plot
Empirical quantiles
0
300
-100
0
50
100
150
Theoretical quantiles
Empirical and theoretical CDFs
CDF
0.0
0.6
0
100
200
300
400
500
Data
P-P plot
Empirical probabilities
0.0
0.6
0.4
0.6
0.8
1.0
Theoretical probabilities

Report

The system can be represented in **JMT** as shown in **Fig.2**, it comprises:

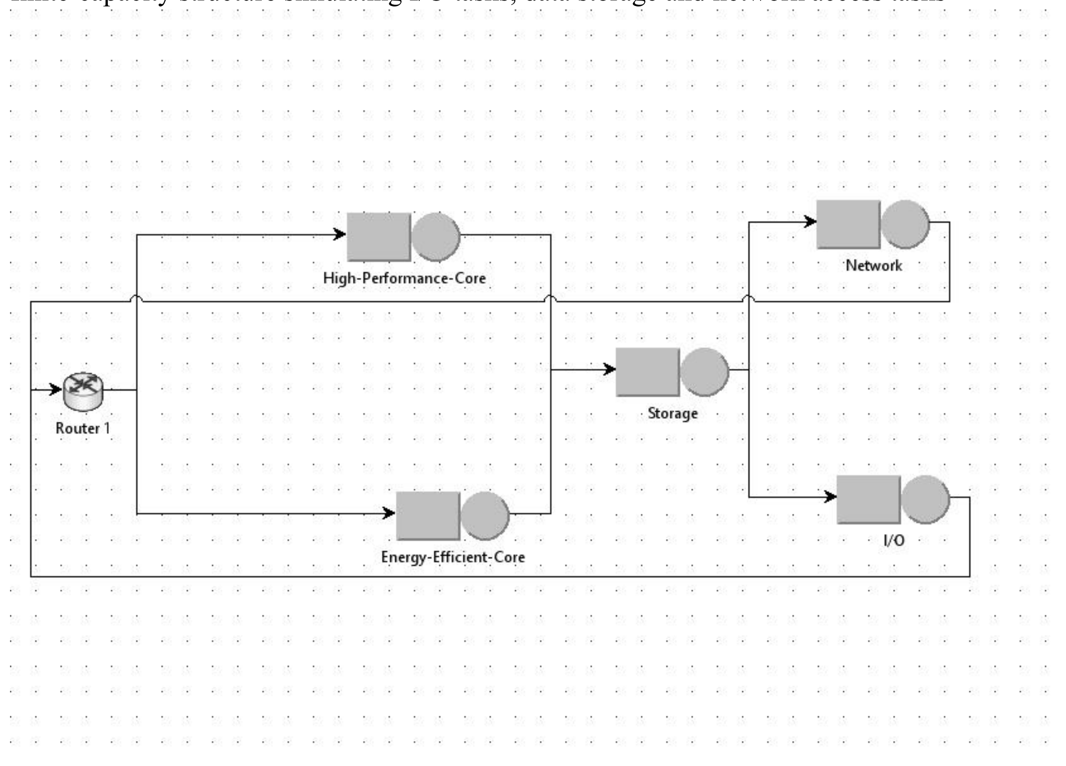
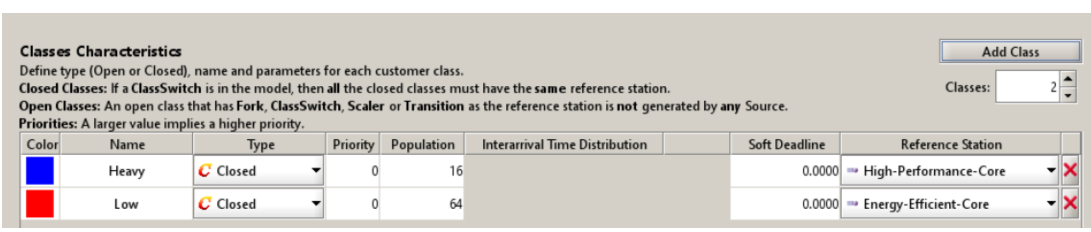
- Router 1(Scheduler) which directs requests within our subsystem
- High-Performance-Core (**HPC**) designed for tasks requiring heavy computation tasks
- Energy-Efficient-Core (**EEC**) designed for tasks requiring lower computation tasks
- I/O, Storage and Network components that simulate work within a processor sharing finite-capacity structure simulating I/O tasks, data storage and network access tasks
- 

Figure 22. – JMT model

Using JMT

Define classes for our multi-class model



Classes Characteristics

Define type (Open or Closed), name and parameters for each customer class.
Closed Classes: If a **ClassSwitch** is in the model, then **all** the closed classes must have the **same** reference station.
Open Classes: An open class that has **Fork**, **ClassSwitch**, **Scaler** or **Transition** as the reference station is **not** generated by **any** Source.
Priorities: A larger value implies a higher priority.

Add Class

Classes: 2

Color	Name	Type	Priority	Population	Interarrival Time Distribution	Soft Deadline	Reference Station
	Heavy	Closed	0	16		0.0000	High-Performance-Core
	Low	Closed	0	64		0.0000	Energy-Efficient-Core

Figure 33

Over the next series of figures we lay out the routing sections in the queues which have been set probabilistically for each class for our scheduler

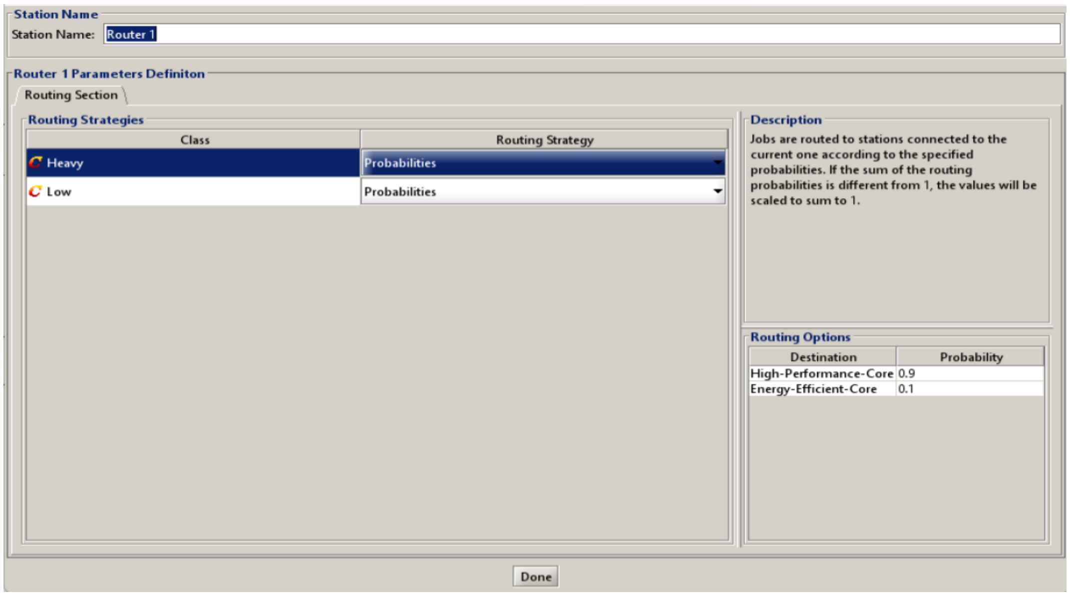


Figure 44 – scheduler parameters

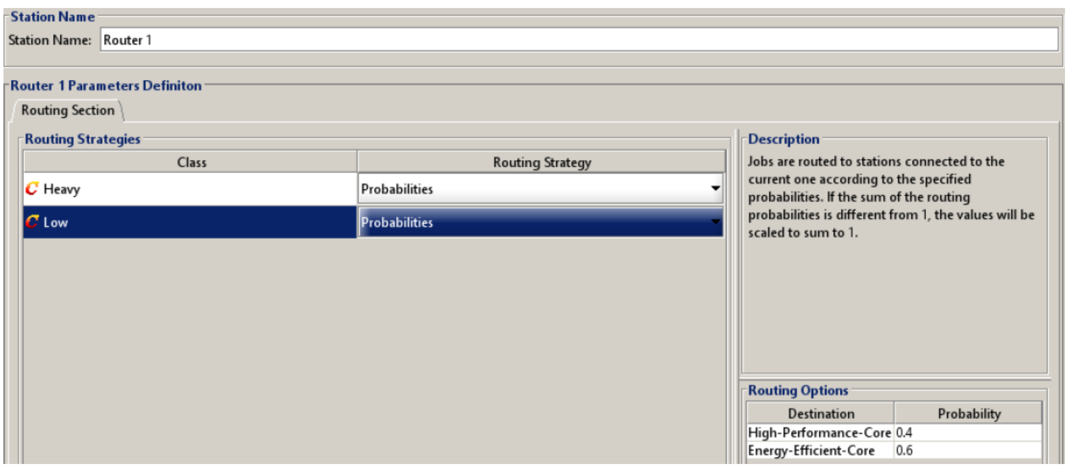


Figure 55 – scheduler parameters

For the **HPC** the queue section for is set to infinite server, and FCFS service policy for all classes. Taking a look at the service section, for heavy computation tasks the hyperexponential distribution has been chosen for this with the different service times per different job type in mind. Conversely, for low computation tasks the erlang distribution was chosen. The coefficients of variation calculations from the trace files were used as a guide for distribution selection.

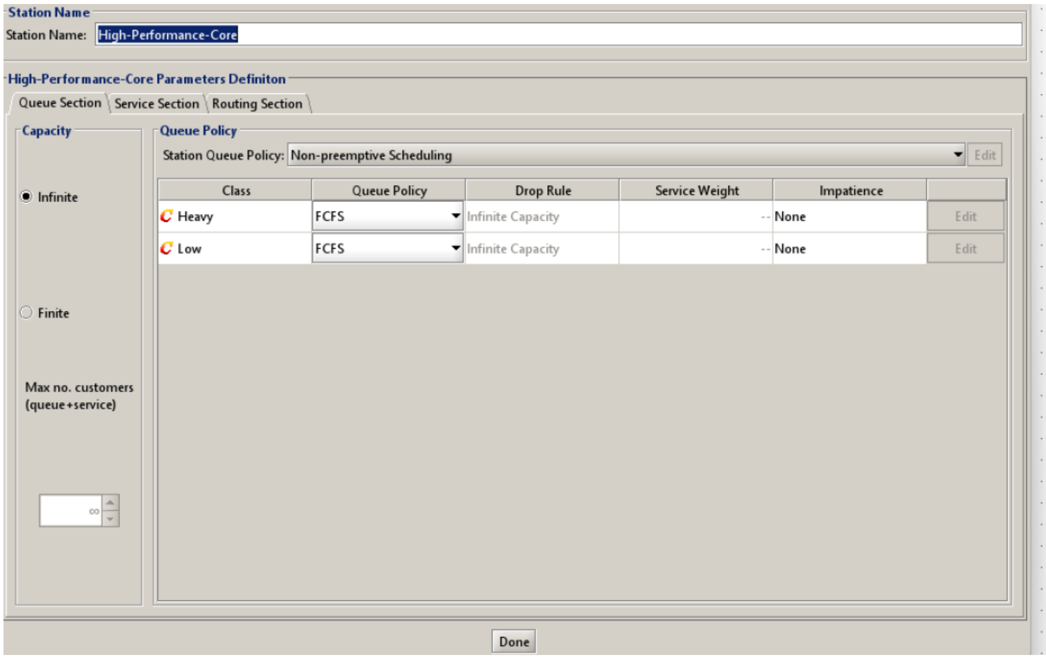


Figure 66 – HPC parameters

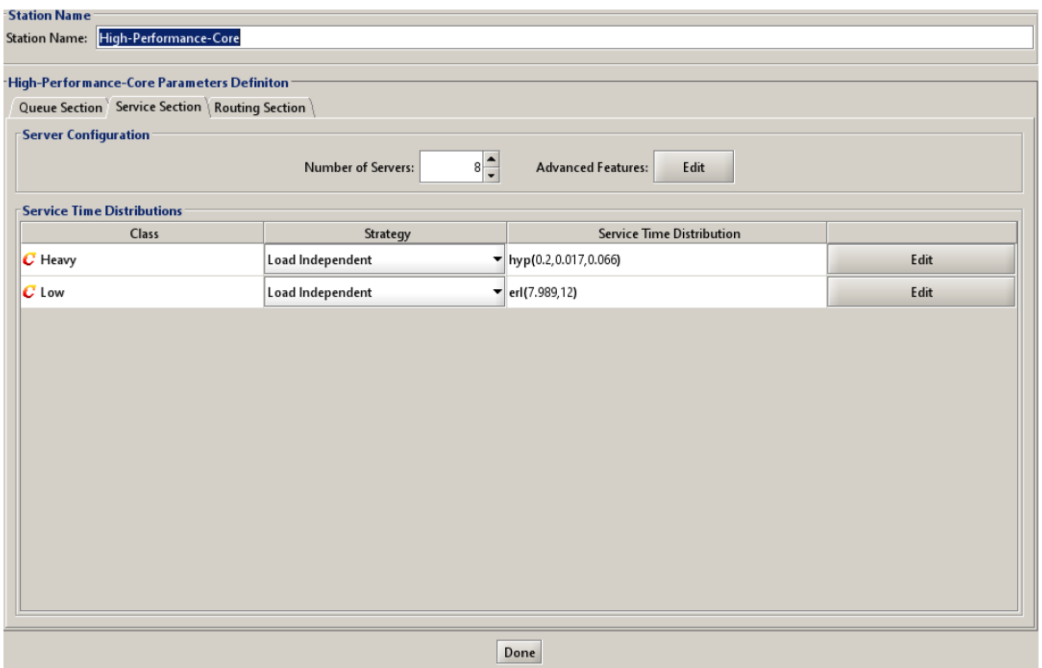


Figure 77 – HPC parameters

For the **EEC** a similar procedure is put in place with respect to queuing policy, service section and routing section.

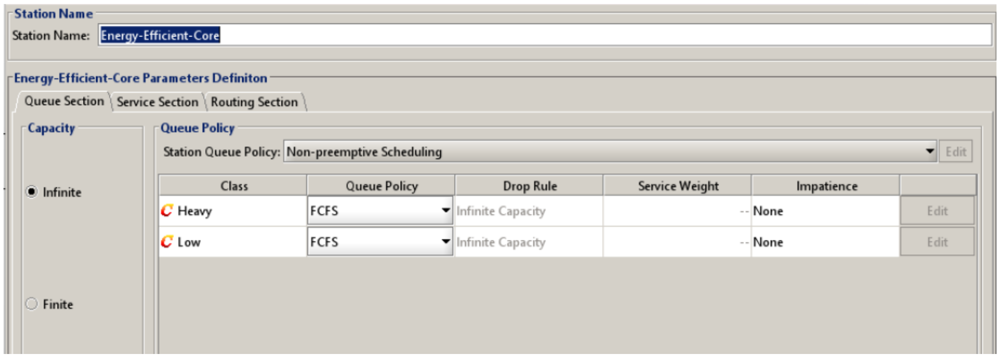


Figure 88 – EEC parameters

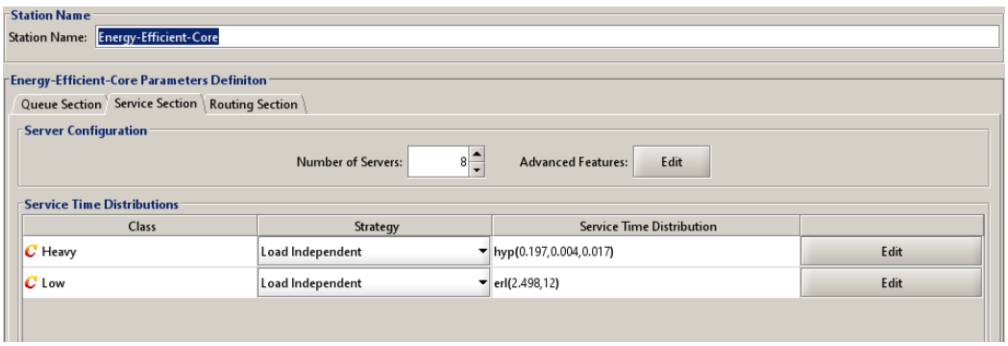


Figure 99 – EEC parameters

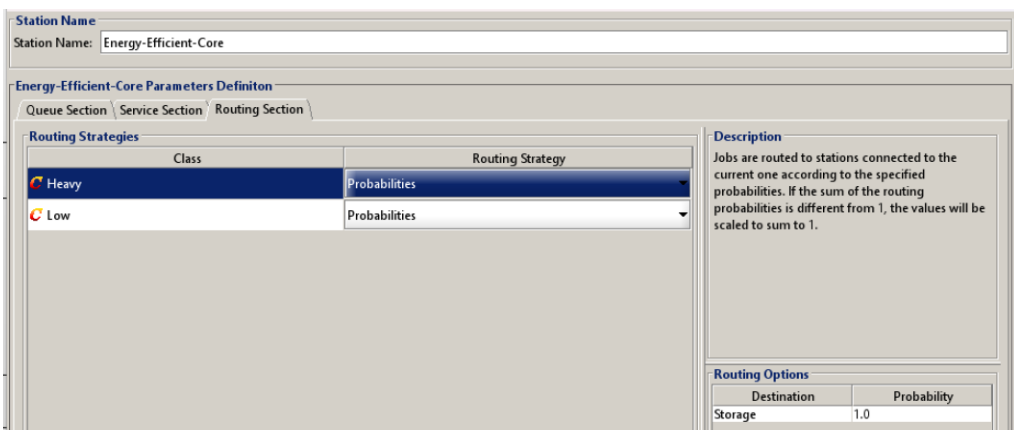


Figure 1010 – EEC parameters

For the Storage, I/O subsystem and Network tasks the considerations are slightly different as this is where the processor sharing aspect of our system model is considered. Figures 11-14 shows a queuing section for heavy and low computation tasks that adheres to processor sharing scheduling requirements. In the service section we make use of the exponential distribution and tasks are routed probabilistically

Report

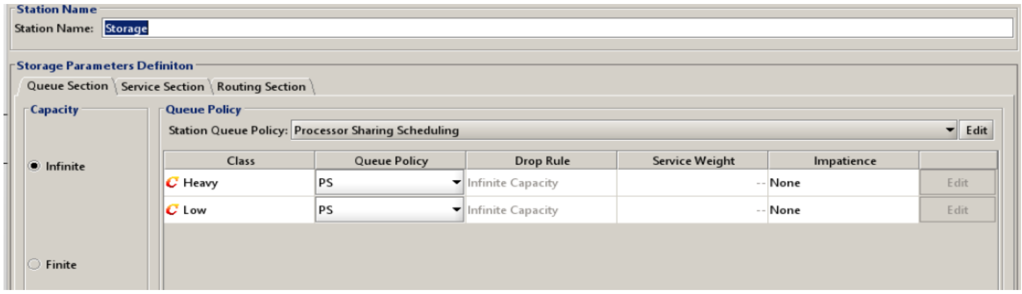


Figure 1111 – storage parameters

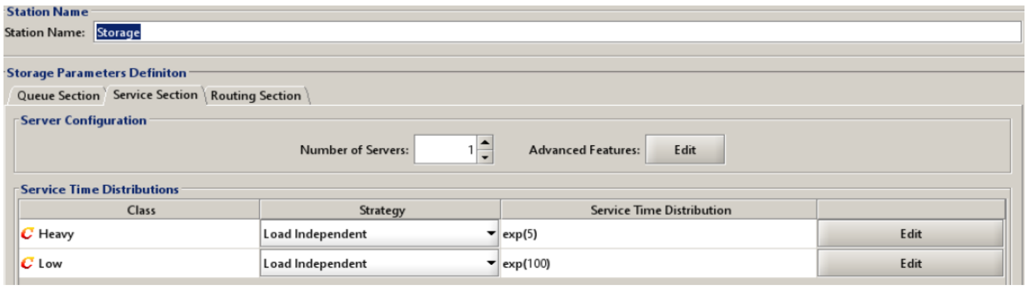


Figure 1212 – storage parameters

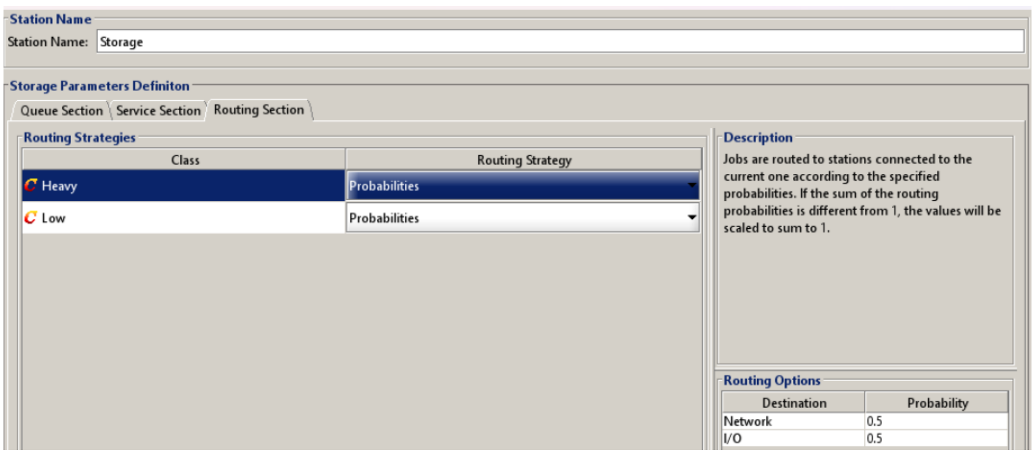


Figure 1313 – storage parameters

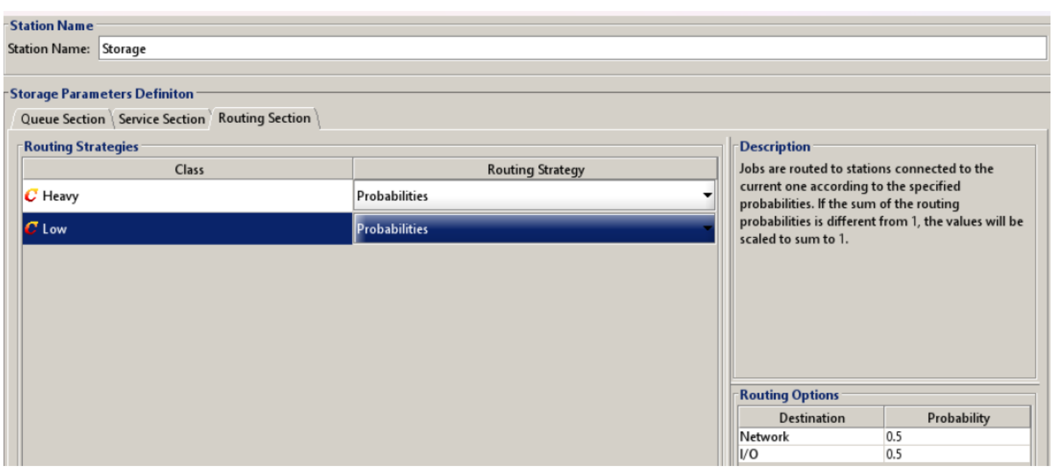


Figure 1414 – storage parameters

Report

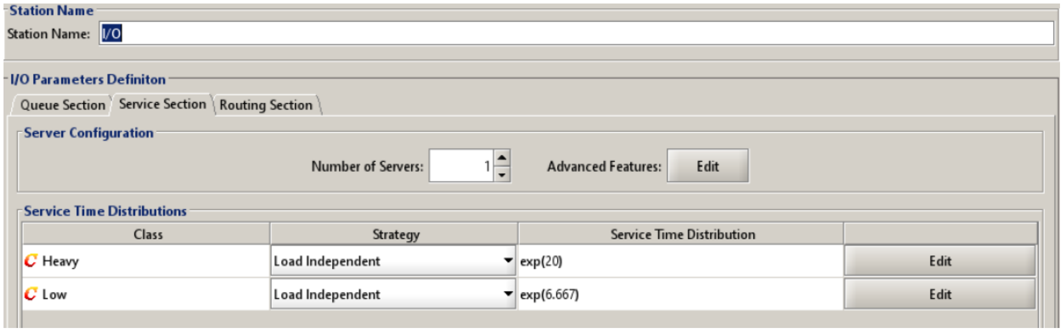


Figure 1515 - I/O parameters

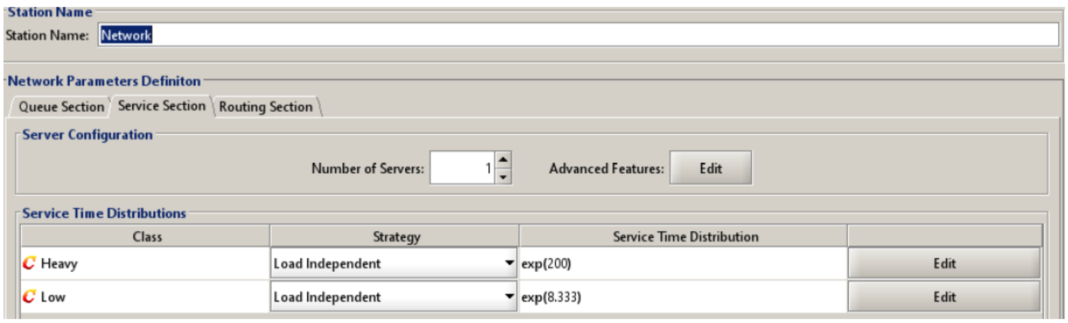


Figure 1616 – Network parameters

Figure 15,16 – show service sections of I/O and Network in processor sharing and the chosen distributions

Performance Indices

Here are the results from a run of our simulation

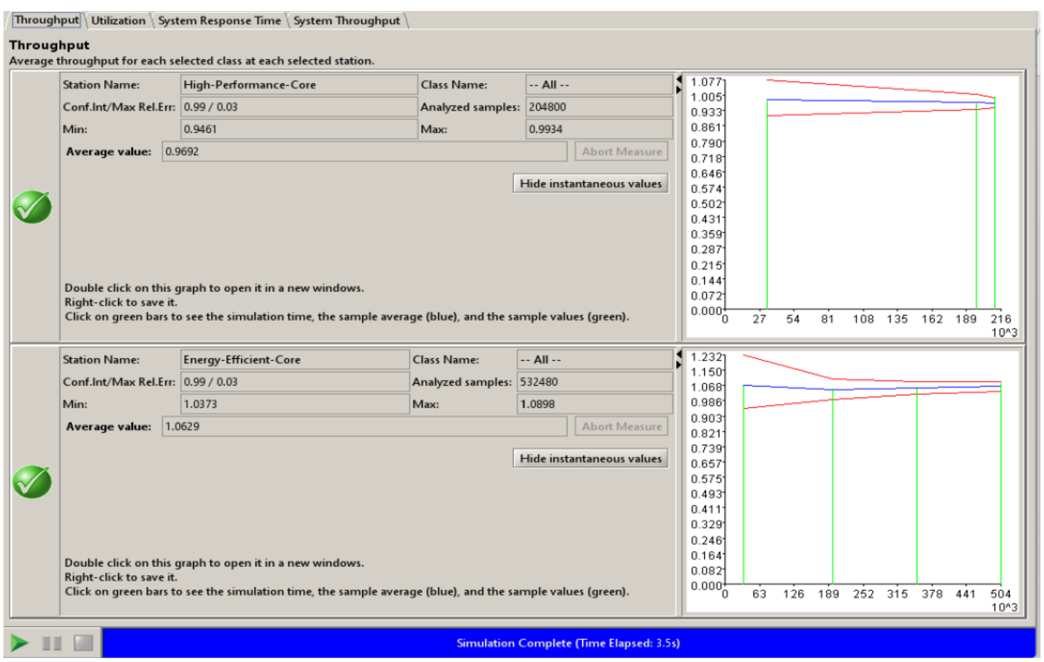


Figure 1717 – Throughput for example 1

Report

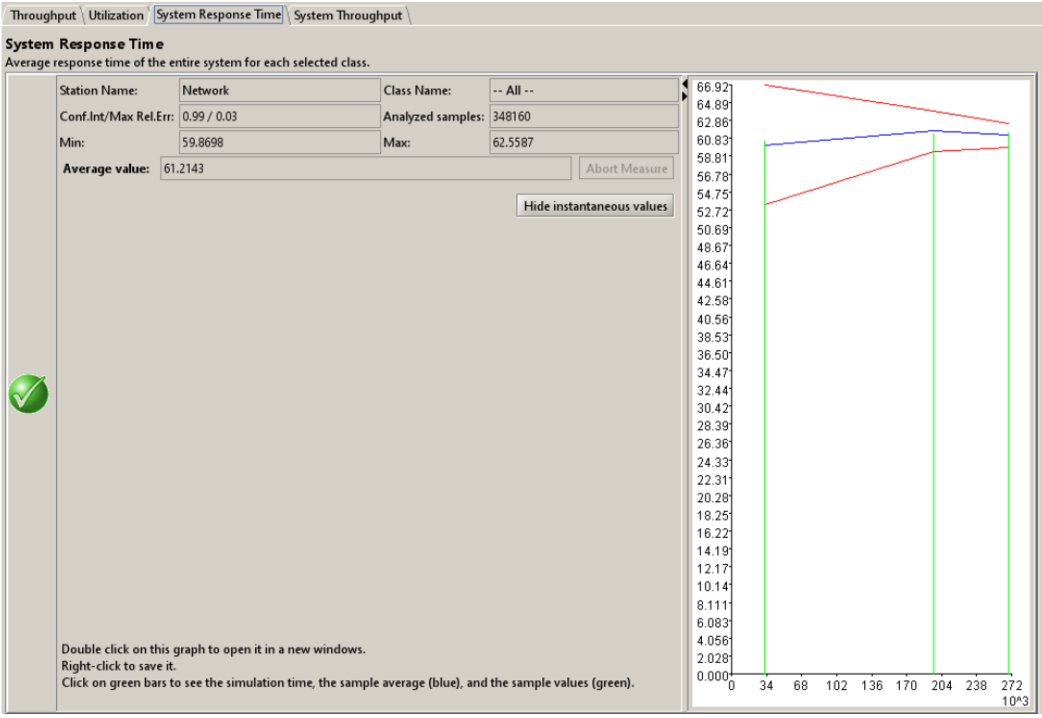


Figure 1818 – system response time for example 1

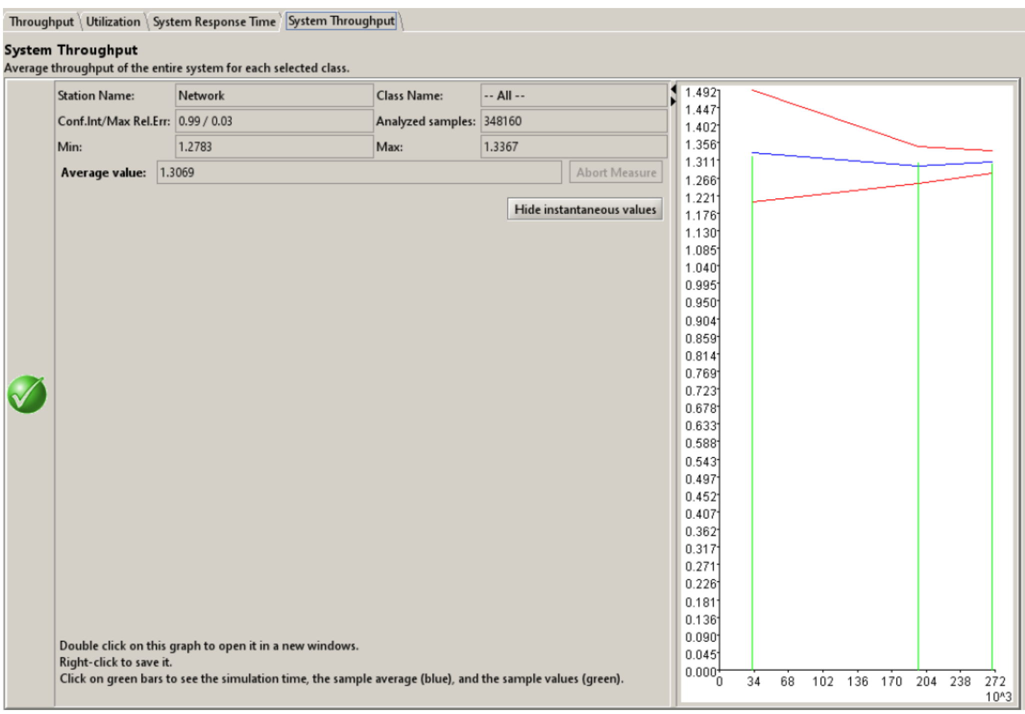


Figure 1919 system throughput for example 1

Example 2 which makes use of a different set of probabilities

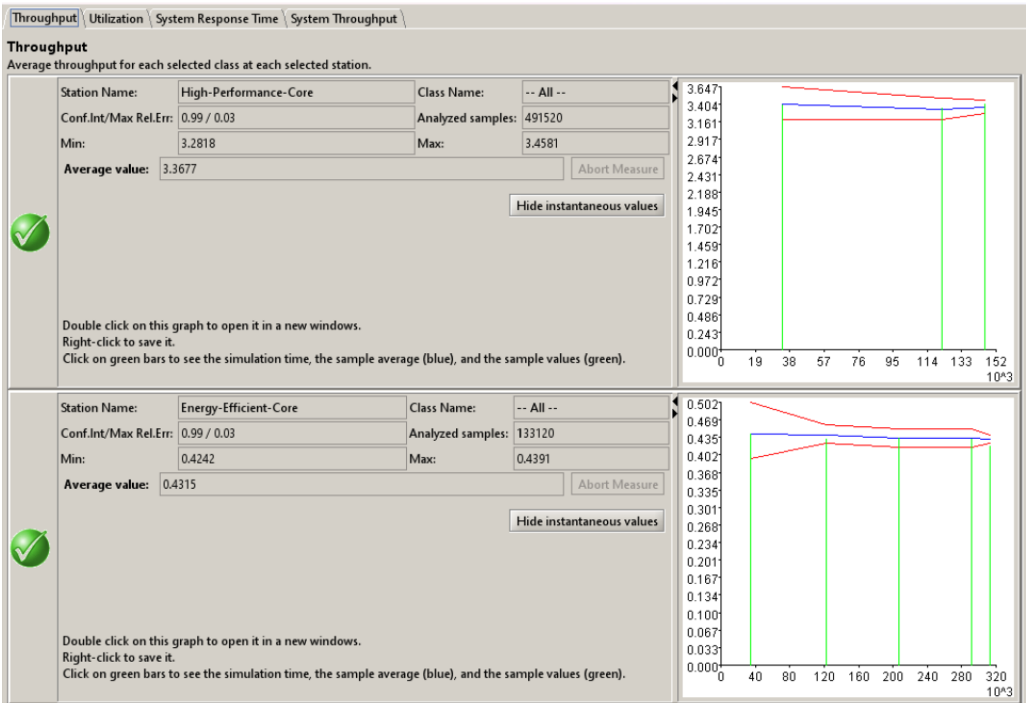


Figure 2020 – throughput for example 2

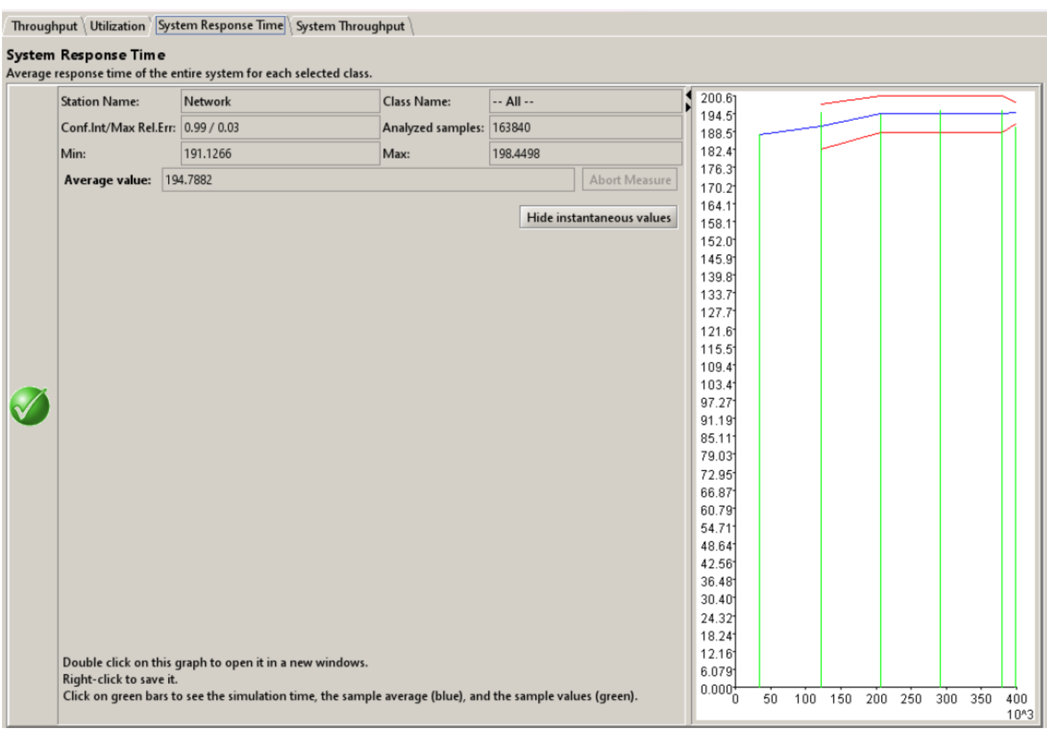


Figure 2121 – system response time for example 2

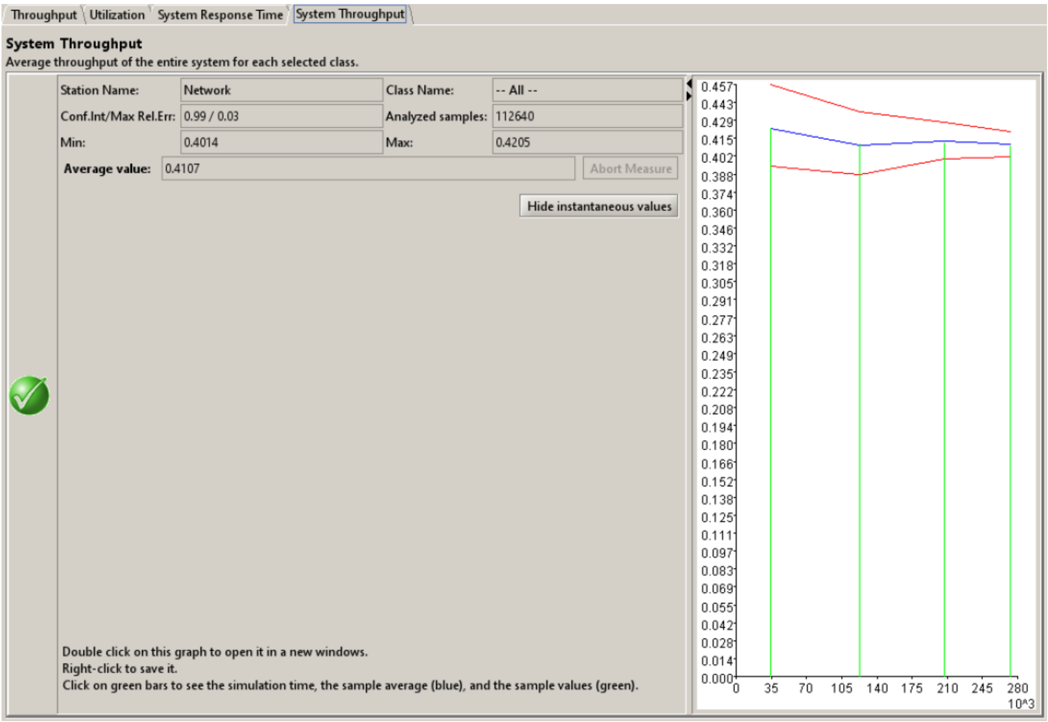


Figure 2222 system throughput for example 2

The system, as modeled, performs well in maintaining throughput and balancing high performance with energy efficiency. Analysis of our results confirms our hypothesis that the system performs much better when heavy computation tasks are assigned to a **HPC** as compared to the second scenario, in example 2, where heavy computation tasks are assigned to the **EEC**.