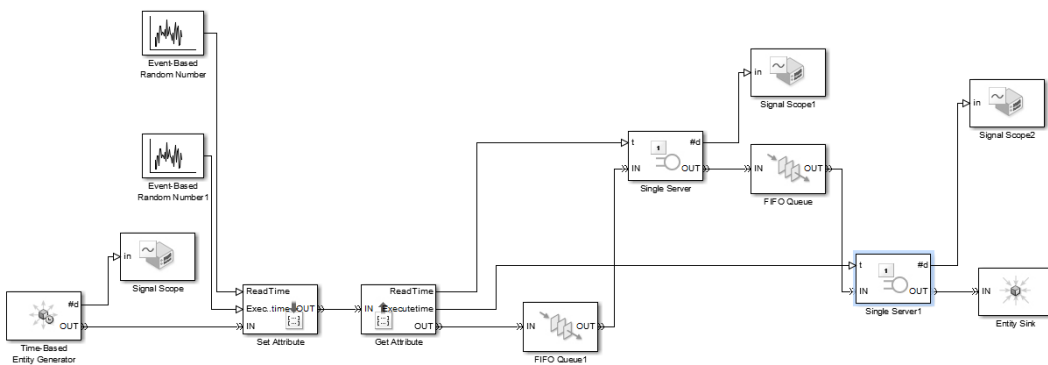


Laboratory 3 MS

Problem 1:

1. Modelarea unui sistem de calcul. La un sistem de calcul sosesc programe la intervale de 2 ± 1 minute cu distribuție uniformă. Ele sunt citite de pe disc cu timpul de citire 30 ± 10 s cu distribuție uniformă. Timpul de execuție este de 3 ± 1 minute cu distribuție uniformă. Să se simuleze modelul pe o durată de 2 ore. Unitatea de timp se alege secunda. Modelul va conține două blocuri Single Server, unul pentru citirea programelor de pe disc și altul pentru execuția programelor și cozile corespunzătoare. Se vor afișa numărul de programe intrate în model, timpul de citire de pe disc și timpul de execuție.

We will use the following schema:



To simulate program arrival we used the following entity generator with these parameters:

Distribution:	Uniform
Initial seed:	12333
Minimum:	60
Maximum:	180
Generation event priority:	300
<input checked="" type="checkbox"/> Generate entity at simulation start	

IN

t

IN

Sin

We randomly generated the reading time:

Parameters

Distribution:

Minimum:

Maximum:

Initial seed:

We also randomly generated the execution time:

Parameters

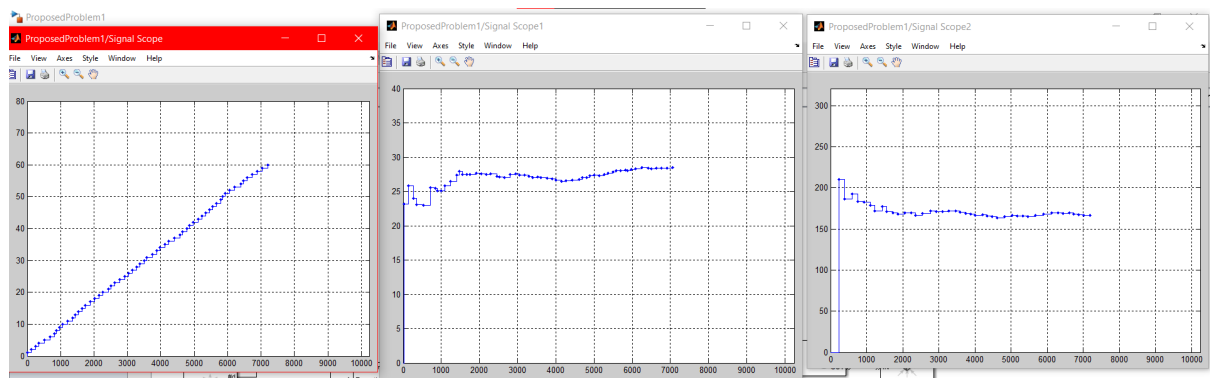
Distribution:

Minimum:

Maximum:

Initial seed:

We got the following result:



LEFT: Number of generated entities

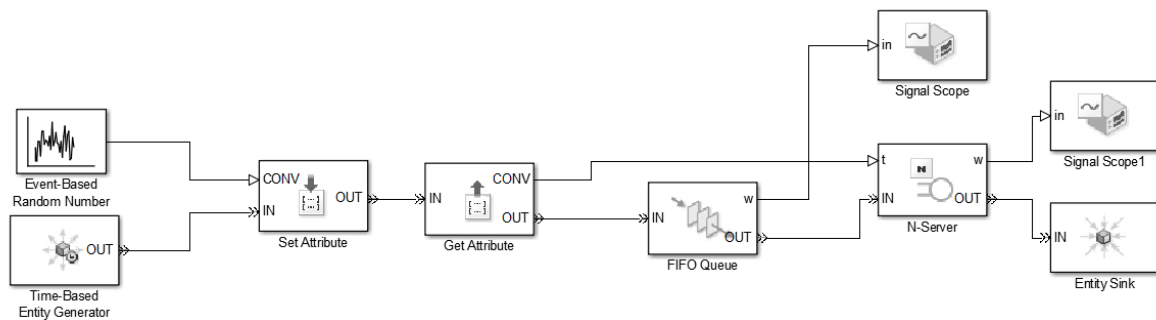
MIDDLE: Average reading time

RIGHT: Average execution time

Problem 2:

2. La o companie telefonică există un serviciu de informații cu cinci operatori. Orice operator poate prelua orice cerere. Cererile apar la intervale cu distribuție exponențială cu valoarea medie 0.2 minute. Durata unei convorbiri este 1 ± 0.2 minute cu distribuție uniformă. Se va utiliza un multiserver cu capacitatea cinci. Să se modeleze sistemul pe o durată de 8 ore. Se vor afișa timpul mediu de așteptare în coadă și durata medie de utilizare a serverului.

We designed the following schema:



We generate the requests using an entity generator:

Distribution: **Exponential**

Initial seed:

12345

Mean:

12

Generation event priority:

300

☒ Generate entity at simulation start

We generated the conversation time using a random number generator:

Minimum:

48

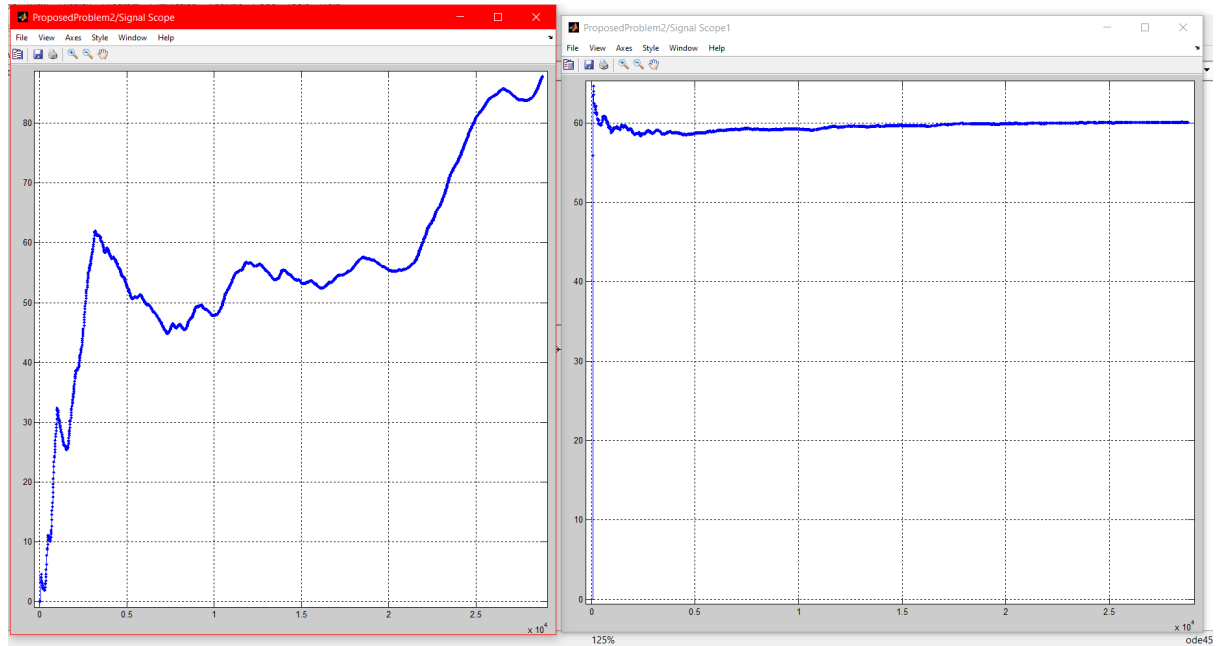
Maximum:

72

Initial seed:

12346

After generation we get the following result:



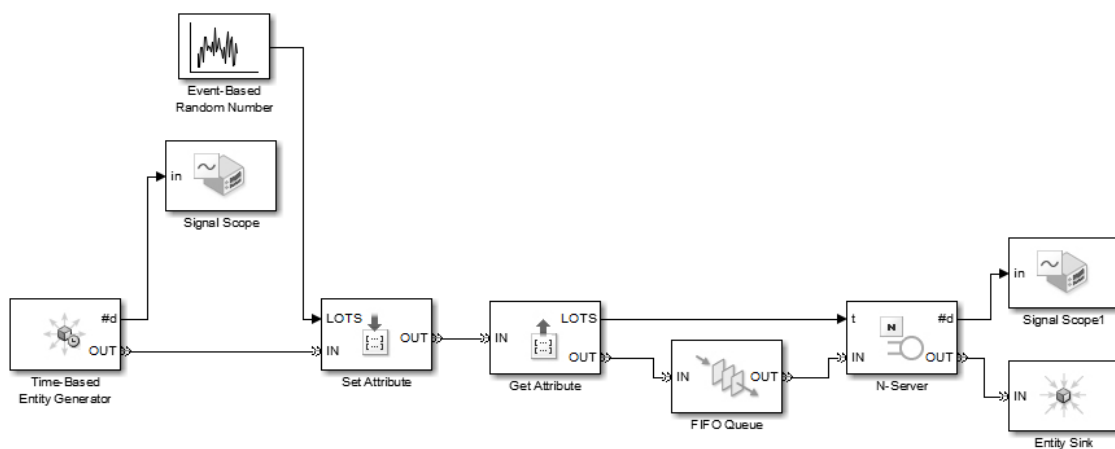
LEFT: Waiting time in queue

RIGHT: Waiting time in server

Problem 3:

- La o mașină unealtă sosesc loturi de piese pentru prelucrare. Loturile sosesc la intervale de 18 ± 6 minute cu distribuție uniformă. Prelucrarea unui lot durează 32 ± 4 minute cu distribuție uniformă. Loturile sunt prelucrate la o stație cu capacitatea 2. Să se simuleze modelul pe o durată de 960 minute. Se vor afișa numărul de loturi generate și numărul de loturi prelucrate. Pentru stația de prelucrare se va utiliza un multiserver cu capacitatea doi.

We designed the following schema:



We set the lots using an entity generator:

Distribution: **Uniform**

Initial seed:
12367

Minimum:
720

Maximum:
1440

Generation event priority:
300

☒ Generate entity at simulation start

We set the processing time using a random number generator:

Parameters

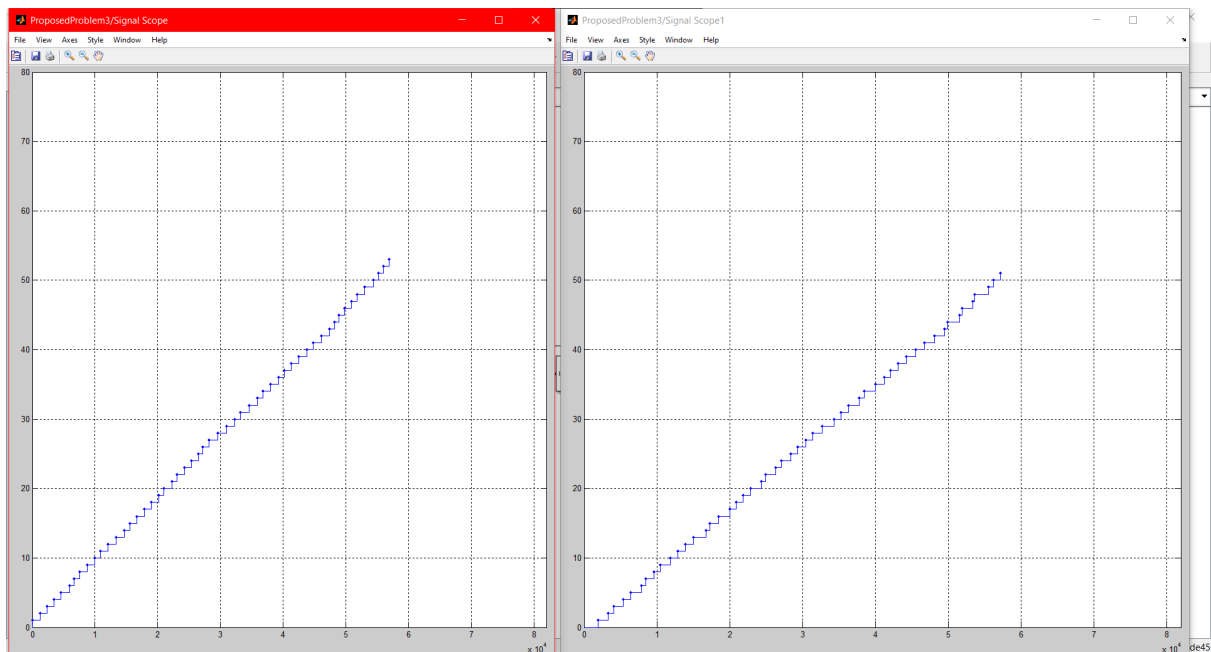
Distribution: **Uniform**

Minimum:
1680

Maximum:
2160

Initial seed:
12332

We got the following result:



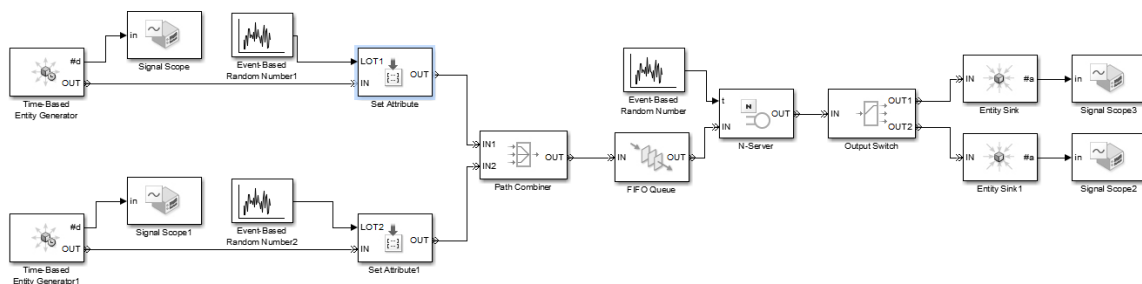
RIGHT: Processed lots

LEFT: Generated lots

Problem 4:

4. La o mașină unealtă sosesc două tipuri de loturi de piese pentru prelucrare. Loturile de primul tip sosesc la intervale de 35 ± 10 minute cu distribuție uniformă. Prelucrarea unui lot durează 18 ± 4 minute cu distribuție uniformă. Loturile de al doilea tip sosesc la intervale de 45 ± 7 minute cu distribuție uniformă. Prelucrarea unui lot durează 18 ± 4 minute cu distribuție uniformă. Loturile de ambele tipuri intră în aceeași coadă printr-un bloc Path Combiner. Să se simuleze modelul pe o durată de 960 minute. Se vor afișa numărul de loturi generate și numărul de loturi prelucrate de fiecare tip.

We designed the following schema:



The parameters set for the first type of lot:

Distribution:	Uniform
Initial seed:	12343
Minimum:	1500
Maximum:	2700
Generation event priority:	300
<input checked="" type="checkbox"/> Generate entity at simulation start	

The parameters set for the second type of lot:

Distribution: **Uniform**

Initial seed:
12342

Minimum:
2280

Maximum:
3120

Generation event priority:
300

☒ Generate entity at simulation start

We used two identical random number generators to simulate the processing time:

Parameters

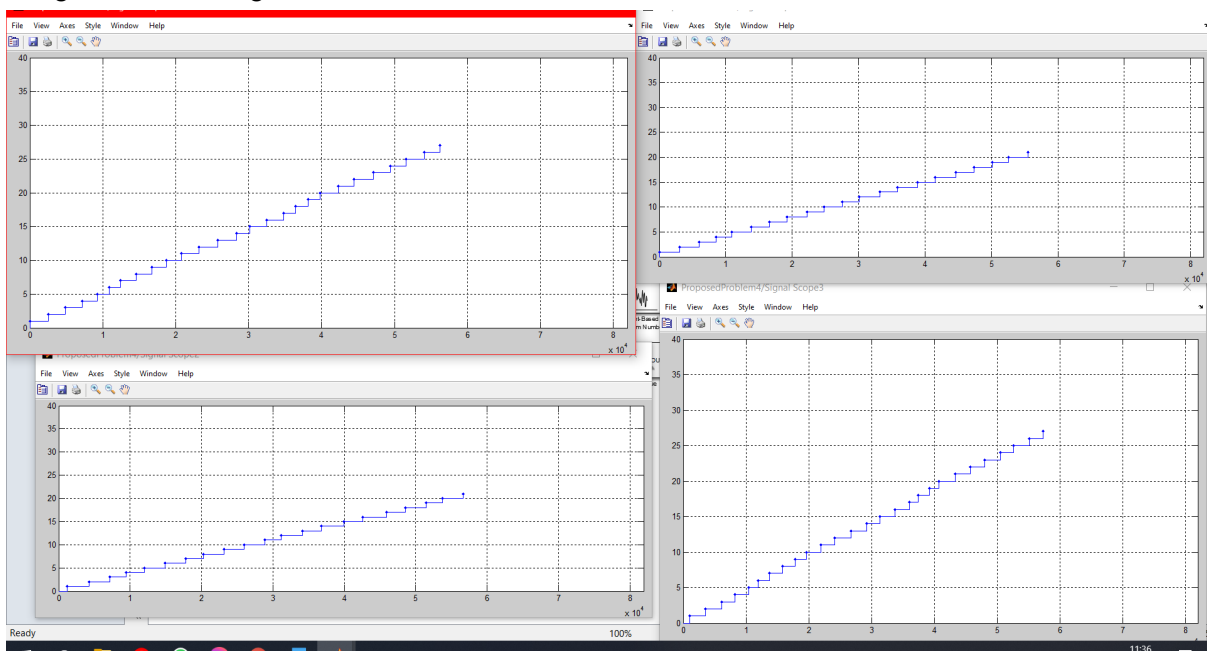
Distribution: **Uniform**

Minimum:
840

Maximum:
1320

Initial seed:
13225

We get the following result:



TOP LEFT: Type one lots generated

TOP RIGHT: Type two lots generated

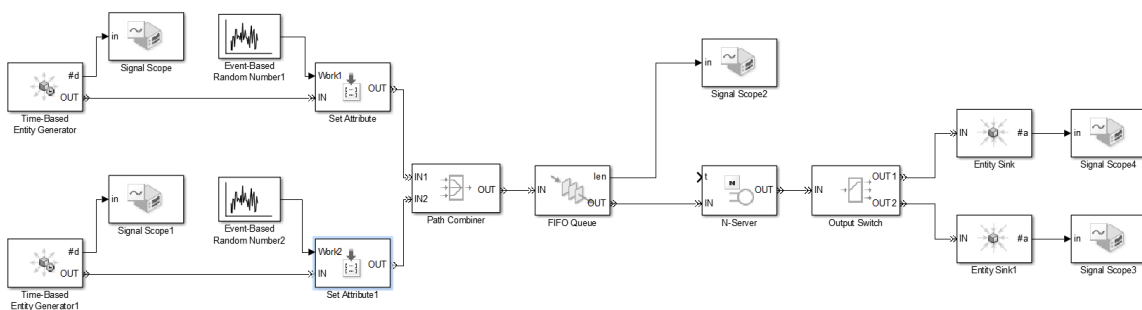
BOTTOM LEFT: Type two lots processed

BOTTOM RIGHT: Type one entities processed

Problem 5:

5. La o magazie pentru aprovizionarea cu materiale vin două tipuri de muncitori. Cei de primul tip vin la intervale de 420 ± 360 s cu distribuție uniformă, iar timpul de deservire este 300 ± 90 s cu distribuție uniformă. Muncitorii de tipul doi la intervale de 360 ± 240 s cu distribuție uniformă, iar timpul de deservire este 100 ± 30 s cu distribuție uniformă. Muncitorii de ambele tipuri așteaptă în aceeași coadă să fie deserviți. Să se simuleze modelul pe o durată de 28800s. Se vor afișa numărul de muncitori de fiecare tip care vin pentru aprovizionare, lungimea cozii și numărul de muncitori deserviți de fiecare tip.

We designed the following schema:



We set the first entity generator to model the first type of worker:

Distribution:	Uniform
Initial seed:	12343
Minimum:	60
Maximum:	780
Generation event priority:	300
<input checked="" type="checkbox"/> Generate entity at simulation start	

We set the second entity generator to model the second type of worker:

Distribution: Uniform

Initial seed:

Minimum:

Maximum:

Generation event priority:

☒ Generate entity at simulation start

Serving time for the first worker type:

Parameters

Distribution: Uniform

Minimum:

Maximum:

Initial seed:

Serving time for second worker type:

Parameters

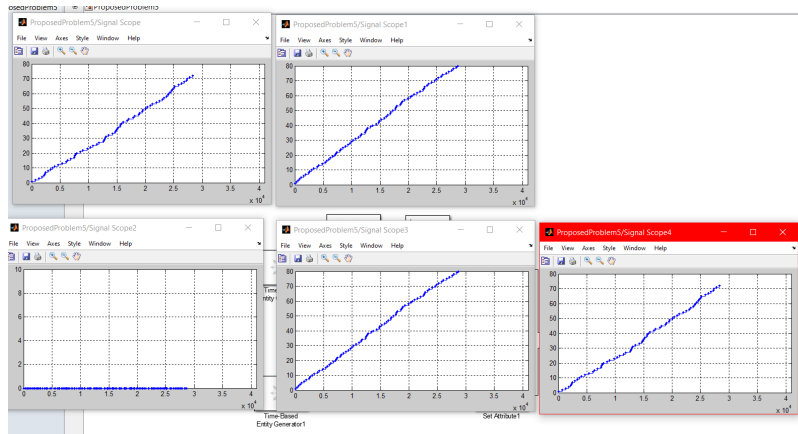
Distribution: Uniform

Minimum:

Maximum:

Initial seed:

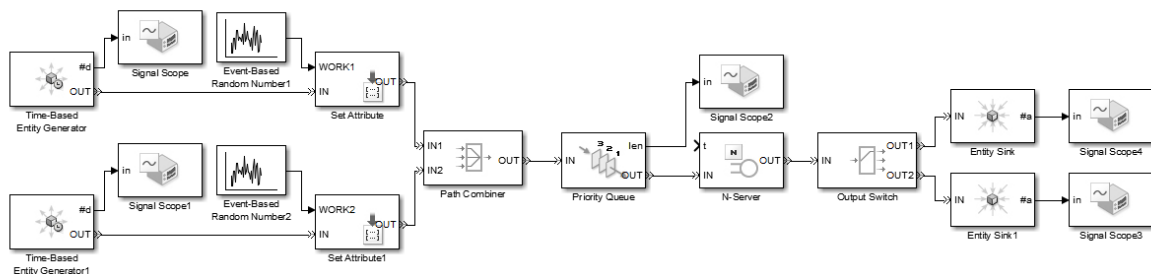
We get the following result:



Problem 6:

6. Se va rezolva problema de mai sus când muncitorii de tipul doi sunt deserviți cu prioritate față de cei de tipul unu.

We used the following schema:



In order to achieve the order we set an importance attribute:

Set Attribute				
Statistics				
Available Attributes				
Filter by name				
No attributes ...				
Set Attribute				
tribute Name	Value From	Value	Vector Is 1-D	
1 WORK1	Signal pc	unused	--	
2 IMP	Dialog	1	<input checked="" type="checkbox"/>	

Set Attribute
Statistics

Available Attributes

Filter by name

No attributes ...

Set Attribute

	Attribute Name	Value From	Value	Vector Is 1-D
1	WORK2	Signal processor	unused	--
2	IMP	Dialog	2	<input checked="" type="checkbox"/>

We also used a priority queue to sort by that attribute:

Priority Queue

Store entities in sorted sequence for an undetermined length of time. The Capacity parameter is the number of entities the queue can hold. The queue sorts entities according to the values of the specified attribute, in either ascending or descending order.

Priority Queue

Timeout

Statistics

Capacity:

25

Sorting attribute name:

IMP

Sorting direction:

Descending

We got the following result

