

Documentație tema 2.

Simulator de cozi.

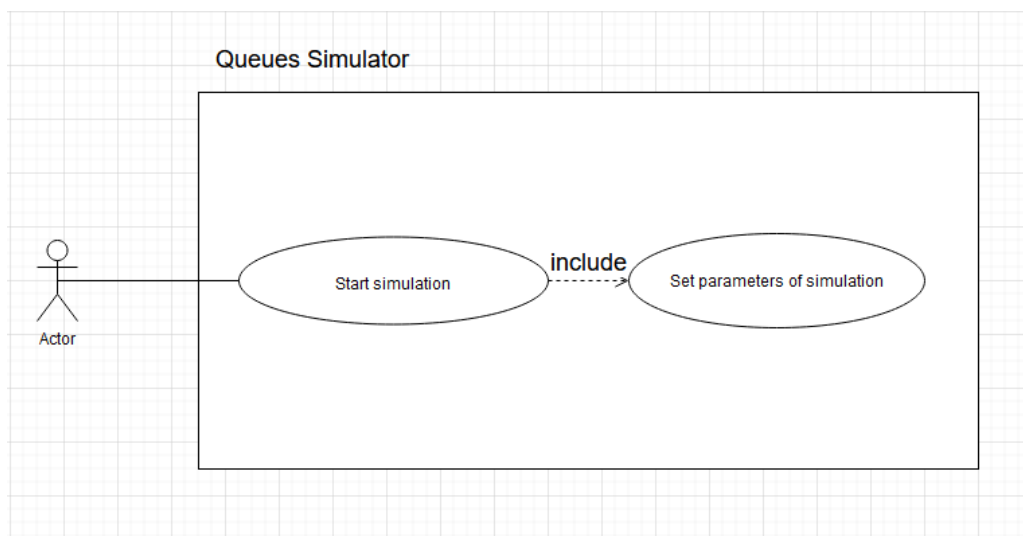
1. Obiectivul temei

Obiectivul principal al temei este simularea unui scenariu în care avem un număr de clienți care sosesc la momente diferite de timp unul față de celălalt și care fiecare are un timp aleatoriu de servire, o cantitate de timp care descrie cât timp ar dura să se realizeze task-ul aferent clientului respectiv, fiecare client trebuie plasat la una dintre cozile disponibile sau în care are cel mai puțin de așteptat.

Această problemă modelează un set de probleme din viața reală în care primim asincron mai multe task-uri pe care trebuie să le deservim în cel mai eficient mod cu resursele disponibile. Una dintre soluții este folosirea tuturor resurselor disponibile pentru rezolvarea problemei, în principal să ne folosim de toată puterea de procesare a sistemului pe care este găzduit aplicația. Acest lucru se realizează de obicei prin concurență.

2. Analiza problemei

Pentru crearea concurenței se folosește paradigma multi-threading, se consideră că pentru o simulare dată avem un număr fix de cozi, iar pentru fiecare dintre acestea există un thread separat care să faciliteze clienții respectivei cozi. Pentru a simula avem nevoie de un set de date pe care aplicația să ruleze, așadar trebuie să implementăm un generator de clienți random pe baza a unor seturi de date primite de la utilizator. Pentru a facilita interacțiunea dintre utilizator și aplicație se implementează o interfață grafică în care se introduc parametrii simulării curente și se afișează de asemenea pașii simulării. De asemenea pentru o analiză mai exactă a modului în care aplicația rulează, se realizează la un interval fix de timp câte un snapshot intern al aplicației care este scris într-un fișier.



Se pot determina 2 **use case**-uri:

Use case 1: Pornire simulare

Actorul principal : Utilizatorul

Scenariu principal de succes:

- Utilizatorul setează parametrii simulării
- După setare apasă pe butonul **Start Simulation** din interfața grafică
- Aplicația citește parametrii, creează clienți aleatori bazați pe acești parametri
- Se pornește simularea

Alternativă:

- Datele din interfața grafică nu sunt setate sau nu sunt relevante pentru simulare

Use case 2: Repornire simulare

Actorul principal : Utilizatorul

Scenariu principal de succes:

- Utilizatorul setează noii parametri a simulării sau nu
- Utilizatorul apasă din nou pe butonul **Start Simulation** din interfața grafică
- Aplicația citește din nou parametrii, creează noii clienți aleatori bazați pe acești parametri
- Se repornește simularea

Alternativă:

- Datele din interfața grafică nu sunt setate sau nu sunt relevante pentru simulare

3. Proiectare

S-a creat o clasă numită **SimulatorController** care are rolul de a controla întreaga simulare și a face legătura dintre starea internă a simulării și interfața grafică, se ocupă de asemenea de gestionarea acționării butonul **Start Simulation**, moment în care creează un nou obiect de tipul **QueuesSimulator** care gestionează simularea cu parametrii colectați din interfața grafică.

SimulatorController		
f	view	SimulatorView
f	simulator	QueuesSimulator
f	simulatorThread	Thread
f	fileWriter	FileWriter
m	SimulatorController(SimulatorView)	
m	main(String[]) void	

S-a mai creat o clasă **SimulatorView** care este clasa ce realizează interfața grafică și funcționalitatea acesteia.

SimulatorView		
f	content	JPanel
f	timeLimit	TextField
f	nrClients	TextField
f	nrServers	TextField
f	minArrival	TextField
f	maxArrival	TextField
f	minProcessing	TextField
f	maxProcessing	TextField
f	startButton	Button
f	simulationTime	TextField
f	averageWaitingTime	TextField
f	averageServiceTime	TextField
f	peakHour	TextField
f	simulationTextArea	TextArea
m	SimulatorView()	
m	printSimulationStage(int, String)	void
m	printSimulationResults(float, float, int)	void
m	createInputPanel()	void
m	createRow(String[], JComponent[], Dimension)	JPanel
m	createJLabel(String, Dimension)	JLabel
m	createSimulationInfoPanel()	void

Pentru realizarea simulării a fost necesar de o clasă care să controleze simularea, aceasta clasă se numește **QueuesSimulator** și are ca scop generarea de clienți aleatori în funcție de parametri primit prin constructor, a ține evidența clienților creați și a clienților care sunt în așteptare. De asemenea implementează metoda **run()** din interfața **Runnable** care reprezintă funcția de pornire a threadului principal care se ocupă de gestionarea repartiției clienților la cozi, realizarea snapshot-urilor stării interne și comunicarea cu interfața grafică pentru a afișa utilizatorului în timp real evoluția simulării.

Realizarea repartiției clienților fiind o acțiune mai specifică s-a decis a fi delegată unei alte clase care să se ocupe de aceasta, clasa este numită **Scheduler**, pe lângă funcționalitatea de gestionarea a clienților creează și threadurile specifice fiecărei cozi și reținerea referinței spre acestea, de asemenea are metode pentru interogarea

stărilor threadurilor și pentru terminarea threadurilor la sfârșitul simulării cât se ocupă și de realizarea unor statistici legate de simularea curentă cum ar fi timpul mediu de așteptare, timpul mediu de servire, ora de vârf.

QueuesSimulator		Scheduler	
f	view	SimulatorView	
f	timeLimit	int	
f	N	int	
f	Q	int	
f	minArrivalTime	int	
f	maxArrivalTime	int	
f	minProcessingTime	int	
f	maxProcessingTime	int	
f	scheduler	Scheduler	
f	clients	ArrayList<Client>	
f	fileWriter	FileWriter	
m	QueuesSimulator()		
m	QueuesSimulator(SimulatorView, int, int, int, int, int, int, int, int, FileWriter)		
m	generateNRandomClients()	void	
m	run()	void	
m	getViewSnapshot()	String	
m	writeLog(String)	void	
m	getSnapshot(int)	String	
m	createResults(float, float, int)	String	
m	shutdown()	void	

Scheduler	
f	numberOfServers int
f	queues ArrayList<Queue>
f	threads ArrayList<Thread>
f	totalWaitingTime int
f	totalServiceTime int
f	maxClientsAtTime int
f	peakTime int
m	Scheduler(int)
m	getQueues() ArrayList<Queue>
m	getTotalWaitingTime() int
m	getTotalServiceTime() int
m	getPeakTime() int
m	dispatchClient(Client, int) void
m	serversStillWorking() boolean
m	shutdown() void

Pentru modelarea unei cozi s-a creat o clasă specifică numită **Queue** care are ca scop reținerea clienților într-o listă de clienți care sunt aferenți cozii respective și timpul pentru care coada respectivă este ocupată cu clienți existenți. De asemenea implementează metoda **run()** executată de thread în care deservește clienții și deservește getter prin care clasa **Scheduler** poate interoga starea cozii respective.

De asemenea mai este nevoie de o clasă care să modeleze obiectele de tip clienti, clasă numită **Client** și care are proprietăți specifice clientului respectiv.

Queue		Client
f 🔒 clients <code>LinkedListQueue<Client></code>		f 🔒 ID <code>int</code>
f 🔒 waitingPeriod <code>AtomicInteger</code>		f 🔒 timeArrival <code>int</code>
		f 🔒 timeService <code>int</code>
m 📄 Queue()		m 📄 Client(int, int, int)
m 📄 getWaitingPeriod() <code>AtomicInteger</code>		m 📄 getArrivalTime() <code>int</code>
m 📄 run() <code>void</code>		m 📄 getProcessingTime() <code>int</code>
m 📄 addClient(Client) <code>int</code>		m 📄 toString() <code>String</code>
m 📄 getNumberOfClients() <code>int</code>		m 📄 decrementTimeService() <code>void</code>
m 📄 getClients() <code>Client[]</code>		

4. Implementare

Clasa SimulatorController:

Variable:

- **view** – referință la interfața grafică
- **simulator** – referință la obiectul care realizează simularea
- **simulatorThread** – thread-ul principal al simulării
- **fileWriter** – referință la fisierul de log

Metode:

- **SimulatorController(SimulatorView)** – constructor care creează obiectul și care primește referință la interfața grafică
- **Main()** – funcția de început

Clasa SimulatorView:

Variable:

- **content** – panelul principal al interfeței
- **timeLimit** – referință la JTextField-ul în care se scrie timpul de sfârșit al simulării
- **nrClients** – referință la JTextField-ul în care se scrie numărul de clienți
- **nrServers** – referință la JTextField-ul în care se scrie numărul de servere
- **minArrival** – referință la JTextField-ul în care se scrie timpul minim la care pot ajunge clienții
- **maxArrival** – referință la JTextField-ul în care se scrie timpul maxim la care pot ajunge clienții
- **minProcessing** – referință la JTextField-ul în care se scrie timpul minim de procesare a unui client
- **maxProcessing** – referință la JTextField-ul în care se scrie timpul maxim de procesare a unui client
- **startButton** – referință la butonul de start al simulării
- **simulationTime** – referință la JTextField-ul în care se scrie timpul simulării
- **averageWaitingTime** – referință la JTextField-ul în care se scrie timpul mediu de așteptare a clienților la sfârșitul simulării
- **averageServingTime** – referință la JTextField-ul în care se scrie timpul mediu de servire a clienților la sfârșitul simulării
- **peakHour** – referință la JTextField-ul în care se scrie timpul de vârf în care au fost cei mai mulți clienți care așteptau la coadă
- **simulationTextArea** – referință la JTextArea-ul în care se scrie fiecare pas al execuției

Metode:

- **SimulatorView()** – constructor care creează obiectul interfeței grafice apelând metodele ajutoare pentru construirea ei
- **createInputPanel()** – creează o parte din interfața grafică
- **createRow()** - creează o linie de elemente aliniate
- **createJLabel(String)** – creează un JLabel cu un string dat și cu o anumită dimensiunea
- **printSimulationStage(int, String)** – metodă apelată de threadul principal în momentul în care se dorește să se afișeze starea curentă
- **printSimulationResults(float, float, int)** – metodă apelată la sfârșitul simulării pentru a afișa în interfața grafică mediile calculate și ora de vârf

Clasa QueuesSimulator :

Variabile:

- **view** – referință la interfața grafică
- **timeLimit** – timpul la care se termină simularea
- **N** – numărul de clienți
- **Q** – numărul de cozi, servere
- **minTimeArrival** – timpul minim la care pot să ajungă clienții, folosit pentru generare
- **maxTimeArrival** – timpul maxim la care pot să ajungă clienții, folosit pentru generare
- **minProcessingTime** - timpul minim de procesare a clienților, folosit pentru generare
- **maxProcessingTime** - timpul maxim de procesare a clienților, folosit pentru generare
- **schedule** – referință la obiectul scheduler
- **fileWriter** – referință la fișierul în care se dorește scrierea log-ului

Metode:

- **QueueSimulator()** – constructor care initializează obiectul cu valorile default ale parametrilor
- **QueueSimulator(SimulatorView, int, ..., FileWriter)** – constructor care inițializează obiectul cu valorile alese din interfața grafică
- **generateNRandomClients()** – generează N clienți aleatori pe care îi stochează în lista de clienți
- **run()** – metoda de start a threadului
- **getViewSnapshot()** – creează un snapshot specific interfeței grafice
- **writeLog()** – scrie în fișier starea actuală
- **getSnapshot()** – creează un snapshot specific fișierului de log
- **createResults()** – calculează rezultatele simulării, valoarea medie de așteptare a clienților, valoare medie de servire a clienților și timpul de vârf al simulării
- **shutdown()** – închide toate thread-uri worker

Clasa Scheduler :

Variabile:

- **numberOfServers** – reprezintă numărul de servere pe care îl creăm
- **queues** – este o listă de obiecte de tipul **Queue** care modelează o coadă
- **threads** – este o listă de threaduri create pentru fiecare obiect de tip Queue
- **totalWaitingTime** – se acumulează timpul total de așteptare al tuturor clienților
- **totalServingTime** – se acumulează timpul total de servire al tuturor clienților
- **maxClientAtTime** – reprezintă maximumul de clienți care au fost în cozi până în momentul respectiv al simulării
- **peakTime** – reprezintă momentul de timp în care a fost maximumul de clienți în cozi

Metode:

- **Scheduler(int)** – creează Q obiecte de tipul Queue si câte un thread pentru fiecare
- **getQueues()** – returneaza lista de cozi
- **getTotalWaitingTime()** - getter la totatWaitingTime
- **getTotalServingTime()** - getter la totatServingTime
- **getPeakTime()** - getter la peakTime
- **dispatchClient(Client, int)** – este metoda care primește un client care trebuie introdus într-una dintre cozi și îl introduce în funcției de timpul minim pe care trebuie să-l aștepte
- **serversStillWorking()** – este o metodă care verifică daca există cel puțin un server care mai are de servit clienți
- **shutdown()** – este metoda folosita pentru a închide toate thread-urile workeri

Clasa Queue:

Variabile:

- **clients** – este o listă care reprezintă referințele la clienții care se află în coada respectivă
- **waitingPeriod** – timpul pentru a procesa clienții existenți in coadă

Aceste variabile sunt de tipuri care asigură **Thread Safe**, **BlockingQueue**, respectiv **AtomicInteger**, pentru a ne asigura că datele nu sunt corupte, fiind accesate de 2 thread-uri concurent, thread-ul principal pentru a citi starea cozii și de thread-ul actual care actualizează informațiile despre el.

Metode:

- **Queue()** – constructor pentru crearea unei cozi
- **getWaitingPeriod()** – getter pentru waitingPeriod
- **run()** – funcția cu care începe execuția thread-ul și care se ocupă de deservirea clienților
- **addClient(Client)** – adaugarea unui client nou in coada respectivă
- **getNumberOfClients()** – returnează numărul de clienți din coada actuală
- **getClients()** – returnează un array de clienti care este o copie al listei de clienti din interiorul obiectului, array folosit pentru afișare

Clasa Client:

Variabile:

- **ID** – identificatorul clientului respectiv
- **timeArrival** – timpul la care acesta ajunge sa fie distribuit la una dintre cozi
- **timeService** – timpul de servire al clientului

Metode:

- **Client(int, int, int)** – contstructor care creează un obiect de tipul Client
- **getArrivalTime()** – getter la timeArrival
- **getProcessingTime()** – getter la timeService
- **toString()** – returnează un String care reprezintă obiectul
- **decrementTimeService()** – decrementează timeService-ul, apelat repetitiv de thread-ul care servește clientul

5. Rezultate

S-au realizat 3 teste, cu parametrii diferiți pentru a se testa cum reacționează aplicația atât cu număr mic de clienți cât și cu un număr mare de clienți. Pentru testul cu cel mai mare număr de clienți durează ceva timp pentru generarea clienților aleatori.

Pentru fiecare test s-a realizat un fișier specific testului care corespunde aceluiași șablon de afișare, pe prima linie se afișează toți clienții care urmează a fi distribuiți, iar pe următoarele linii pentru fiecare coadă este afișată numărul cozii și în ordine secvențială clienții care așteaptă la coada respectivă

Fiecare client este afișat prin tripletul (x,y,z), unde x este identificatorul clientului, y este timpul la care acesta ajunge să fie distribuit și z timpul de servire a acestuia, în momentul în care acesta este primul la coadă și thread-ul respectiv îl servește timpul de servire descrește.

Mai jos sunt afișate rezultatele testelor:

1. Testul 1

Pentru $N = 4$, $Q = 2$, $tMax = 60$; $[tMinArrival, tMaxArrival] = [2, 30]$;

$[tMinService, tMaxService] = [2, 4]$;

Time: 0
Waiting clients: (2,4,2); (1,9,2); (3,15,3); (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 1
Waiting clients: (2,4,2); (1,9,2); (3,15,3); (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 2
Waiting clients: (2,4,2); (1,9,2); (3,15,3); (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 3
Waiting clients: (2,4,2); (1,9,2); (3,15,3); (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 4
Waiting clients: (1,9,2); (3,15,3); (4,25,2);
Queue 1: (2,4,2);
Queue 2: closed

Time: 5
Waiting clients: (1,9,2); (3,15,3); (4,25,2);
Queue 1: (2,4,1);
Queue 2: closed

Time: 6
Waiting clients: (1,9,2); (3,15,3); (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 7
Waiting clients: (1,9,2); (3,15,3); (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 8
Waiting clients: (1,9,2); (3,15,3); (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 9
Waiting clients: (3,15,3); (4,25,2);
Queue 1: (1,9,2);
Queue 2: closed

Time: 10
Waiting clients: (3,15,3); (4,25,2);
Queue 1: (1,9,1);
Queue 2: closed

Time: 11
Waiting clients: (3,15,3); (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 12
Waiting clients: (3,15,3); (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 13
Waiting clients: (3,15,3); (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 14
Waiting clients: (3,15,3); (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 15
Waiting clients: (4,25,2);
Queue 1: (3,15,3);
Queue 2: closed

Time: 16
Waiting clients: (4,25,2);
Queue 1: (3,15,2);
Queue 2: closed

Time: 17
Waiting clients: (4,25,2);
Queue 1: (3,15,1);
Queue 2: closed

Time: 18
Waiting clients: (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 19
Waiting clients: (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 20
Waiting clients: (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 21
Waiting clients: (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 22
Waiting clients: (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 23
Waiting clients: (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 24
Waiting clients: (4,25,2);
Queue 1: closed
Queue 2: closed

Time: 25
Waiting clients:
Queue 1: (4,25,2);
Queue 2: closed

Time: 26
Waiting clients:
Queue 1: (4,25,1);
Queue 2: closed

Average waiting time: 2,25
Average servint time: 2,25
Peak hour: 4

2. Testul 2

Pentru $N = 50$, $Q = 5$, $t_{\text{Max}} = 60$; $[t_{\text{MinArrival}}, t_{\text{MaxArrival}}] = [2, 40]$;
 $[t_{\text{MinService}}, t_{\text{MaxService}}] = [1, 7]$;

Time: 0
Waiting clients: (12,3,3); (19,3,5); (38,3,1); (35,4,7); (24,6,6); (27,7,7); (50,9,1); (7,10,4); (36,10,2); (48,11,4); (11,12,3); (17,13,5); (25,14,2); (29,15,1); (40,16,2); (21,17,4); (31,17,3);
Queue 1: closed
Queue 2: closed
Queue 3: closed
Queue 4: closed
Queue 5: closed

Time: 1
Waiting clients: (12,3,3); (19,3,5); (38,3,1); (35,4,7); (24,6,6); (27,7,7); (50,9,1); (7,10,4); (36,10,2); (48,11,4); (11,12,3); (17,13,5); (25,14,2); (29,15,1); (40,16,2); (21,17,4); (31,17,3);
Queue 1: closed
Queue 2: closed
Queue 3: closed
Queue 4: closed
Queue 5: closed

Time: 2
Waiting clients: (12,3,3); (19,3,5); (38,3,1); (35,4,7); (24,6,6); (27,7,7); (50,9,1); (7,10,4); (36,10,2); (48,11,4); (11,12,3); (17,13,5); (25,14,2); (29,15,1); (40,16,2); (21,17,4); (31,17,3);
Queue 1: closed
Queue 2: closed
Queue 3: closed
Queue 4: closed
Queue 5: closed

Time: 3
Waiting clients: (35,4,7); (24,6,6); (27,7,7); (50,9,1); (7,10,4); (36,10,2); (48,11,4); (11,12,3); (17,13,5); (25,14,2); (29,15,1); (40,16,2); (21,17,4); (31,17,3);
Queue 1: (12,3,3);
Queue 2: (19,3,5);
Queue 3: (38,3,1);
Queue 4: closed
Queue 5: closed

Time: 4
Waiting clients: (24,6,6); (27,7,7); (50,9,1); (7,10,4); (36,10,2); (48,11,4); (11,12,3); (17,13,5); (25,14,2); (29,15,1); (40,16,2); (21,17,4); (31,17,3);
Queue 1: (12,3,2);
Queue 2: (19,3,4);
Queue 3: (35,4,7);
Queue 4: closed
Queue 5: closed

Time: 5
Waiting clients: (24,6,6); (27,7,7); (50,9,1); (7,10,4); (36,10,2); (48,11,4); (11,12,3); (17,13,5); (25,14,2); (29,15,1); (40,16,2); (21,17,4); (31,17,3);
Queue 1: (12,3,1);
Queue 2: (19,3,3);
Queue 3: (35,4,6);
Queue 4: closed
Queue 5: closed

Time: 6
Waiting clients: (27,7,7); (50,9,1); (7,10,4); (36,10,2); (48,11,4); (11,12,3); (17,13,5); (25,14,2); (29,15,1); (40,16,2); (21,17,4); (31,17,3);
Queue 1: closed
Queue 2: (19,3,2);
Queue 3: (35,4,5);
Queue 4: (24,6,6);
Queue 5: closed

Time: 7
Waiting clients: (50,9,1); (7,10,4); (36,10,2); (48,11,4); (11,12,3); (17,13,5); (25,14,2); (29,15,1); (40,16,2); (21,17,4); (31,17,3);
Queue 1: (27,7,7);
Queue 2: (19,3,1);

Time: 34
Waiting clients: (37,35,3); (2,36,3); (20,36,1); (33,36,1); (39,36,7); (42,36,1); (26,38,7); (28,38,1); (30,38,5); (5,39,5);
Queue 1: closed
Queue 2: (41,32,5);
Queue 3: (4,31,2);
Queue 4: closed
Queue 5: (14,32,5);

Time: 35
Waiting clients: (2,36,3); (20,36,1); (33,36,1); (39,36,7); (42,36,1); (26,38,7); (28,38,1); (30,38,5); (5,39,5); (9,39,4); (
Queue 1: (37,35,3);
Queue 2: (41,32,4);
Queue 3: (4,31,1);
Queue 4: closed
Queue 5: (14,32,4);

Time: 36
Waiting clients: (26,38,7); (28,38,1); (30,38,5); (5,39,5); (9,39,4); (15,39,1); (10,40,7);
Queue 1: (37,35,2); (39,36,7);
Queue 2: (41,32,3);
Queue 3: (2,36,3);
Queue 4: (20,36,1); (33,36,1); (42,36,1);
Queue 5: (14,32,3);

Time: 37
Waiting clients: (26,38,7); (28,38,1); (30,38,5); (5,39,5); (9,39,4); (15,39,1); (10,40,7);
Queue 1: (37,35,1); (39,36,7);
Queue 2: (41,32,2);
Queue 3: (2,36,2);
Queue 4: (33,36,1); (42,36,1);
Queue 5: (14,32,2);

Time: 38
Waiting clients: (5,39,5); (9,39,4); (15,39,1); (10,40,7);
Queue 1: (39,36,7);
Queue 2: (41,32,1); (26,38,7);
Queue 3: (2,36,1); (28,38,1);
Queue 4: (42,36,1); (30,38,5);
Queue 5: (14,32,1);

Time: 39
Waiting clients: (10,40,7);
Queue 1: (39,36,6);
Queue 2: (26,38,7);
Queue 3: (28,38,1); (9,39,4); (15,39,1);
Queue 4: (30,38,5);
Queue 5: (5,39,5);

Time: 40
Waiting clients:
Queue 1: (39,36,5);
Queue 2: (26,38,6);
Queue 3: (9,39,4); (15,39,1);
Queue 4: (30,38,4); (10,40,7);
Queue 5: (5,39,4);

Time: 45
Waiting clients:
Queue 1: closed
Queue 2: (26,38,1);
Queue 3: closed
Queue 4: (10,40,6);
Queue 5: closed

Time: 46
Waiting clients:
Queue 1: closed
Queue 2: closed
Queue 3: closed
Queue 4: (10,40,5);
Queue 5: closed

Time: 47
Waiting clients:
Queue 1: closed
Queue 2: closed
Queue 3: closed
Queue 4: (10,40,4);
Queue 5: closed

Time: 48
Waiting clients:
Queue 1: closed
Queue 2: closed
Queue 3: closed
Queue 4: (10,40,3);
Queue 5: closed

Time: 49
Waiting clients:
Queue 1: closed
Queue 2: closed
Queue 3: closed
Queue 4: (10,40,2);
Queue 5: closed

Time: 50
Waiting clients:
Queue 1: closed
Queue 2: closed
Queue 3: closed
Queue 4: (10,40,1);
Queue 5: closed

Time: 51
Waiting clients:
Queue 1: closed
Queue 2: closed
Queue 3: closed
Queue 4: closed
Queue 5: closed

Average waiting time: 3,66
Average servint time: 3,66
Peak hour: 28

3. Testul 3

Pentru N = 1000, Q = 20, tMax = 200; [tMinArrival, tMaxArrival] = [10, 100];
[tMinService, tMaxService] = [3, 9];

```
Time: 0
Waiting clients: (63,10,8); (74,10,5); (146,10,4); (357,10,9); (374,10,6); (379,10,9); (423,10,5); (430,10,4); (596,10,8); (687,10,7); (707,10,9); (744,10,6); (749,10,6); (830,10,7,9); (386,17,4); (388,17,3); (403,17,9); (411,17,4); (529,17,8); (532,17,4); (550,17,8); (625,17,4); (656,17,5); (711,17,6); (855,17,3); (856,17,5); (929,17,7); (978,17,5); (45,1,416,24,9); (436,24,8); (777,24,3); (888,24,8); (102,25,3); (114,25,3); (173,25,9); (213,25,3); (376,25,4); (456,25,8); (706,25,7); (721,25,6); (967,25,6); (152,26,9); (537,26,8,7); (281,34,8); (399,34,3); (471,34,7); (545,34,9); (714,34,6); (738,34,5); (810,34,5); (885,34,8); (89,35,8); (160,35,6); (162,35,6); (278,35,5); (305,35,4); (364,35,4); (377,350,7); (708,40,9); (741,40,3); (969,40,5); (25,41,8); (224,41,4); (231,41,9); (431,41,8); (441,41,3); (562,41,7); (753,41,7); (789,41,9); (816,41,6); (887,41,7); (971,41,8); (24,4247,9); (688,47,5); (692,47,4); (824,47,7); (994,47,6); (65,48,3); (134,48,6); (157,48,7); (257,48,8); (335,48,4); (355,48,7); (384,48,9); (461,48,7); (484,48,5); (570,48,4); (572,54,7); (807,54,4); (133,55,9); (192,55,9); (271,55,5); (274,55,9); (387,55,6); (468,55,3); (808,55,8); (843,55,3); (865,55,9); (905,55,9); (999,55,8); (11,56,7); (18,56,3); (215,56,6); (845,61,6); (848,61,6); (871,61,6); (67,62,8); (249,62,6); (250,62,9); (511,62,3); (593,62,5); (661,62,3); (919,62,8); (93,63,7); (298,63,4); (439,63,7); (483,63,7); (573,63,4,6); (368,70,9); (503,70,5); (653,70,6); (697,70,9); (951,70,3); (137,71,9); (361,71,5); (398,71,5); (450,71,8); (481,71,4); (645,71,6); (731,71,9); (745,71,7); (105,72,3); (284,7113,79,3); (268,79,7); (307,79,4); (373,79,9); (425,79,5); (460,79,6); (549,79,4); (698,79,6); (803,79,8); (890,79,8); (893,79,4); (914,79,5); (989,79,3); (66,80,9); (171,80,3); (508,87,4); (839,87,5); (922,87,7); (253,88,4); (302,88,4); (342,88,4); (363,88,4); (369,88,9); (435,88,9); (492,88,4); (512,88,9); (520,88,7); (577,88,5); (582,88,7); (723,88,3); (92,94,3); (950,94,3); (54,95,4); (148,95,5); (474,95,9); (548,95,3); (607,95,8); (757,95,4); (835,95,5); (867,95,8); (886,95,4); (20,96,7); (84,96,5); (97,96,8); (123,96,4); (308,
Queue 1: closed
Queue 2: closed
Queue 3: closed
Queue 4: closed
Queue 5: closed
Queue 6: closed
Queue 7: closed
Queue 8: closed
Queue 9: closed
Queue 10: closed
Queue 11: closed
Queue 12: closed
Queue 13: closed
Queue 14: closed
Queue 15: closed
Queue 16: closed
Queue 17: closed
Queue 18: closed
Queue 19: closed
Queue 20: closed

Time: 1
Waiting clients: (63,10,8); (74,10,5); (146,10,4); (357,10,9); (374,10,6); (379,10,9); (423,10,5); (430,10,4); (596,10,8); (687,10,7); (707,10,9); (744,10,6); (749,10,6); (830,10,7,9); (386,17,4); (388,17,3); (403,17,9); (411,17,4); (529,17,8); (532,17,4); (550,17,8); (625,17,4); (656,17,5); (711,17,6); (855,17,3); (856,17,5); (929,17,7); (978,17,5); (45,1,416,24,9); (436,24,8); (777,24,3); (888,24,8); (102,25,3); (114,25,3); (173,25,9); (213,25,3); (376,25,4); (456,25,8); (706,25,7); (721,25,6); (967,25,6); (152,26,9); (537,26,8,7); (281,34,8); (399,34,3); (471,34,7); (545,34,9); (714,34,6); (738,34,5); (810,34,5); (885,34,8); (89,35,8); (160,35,6); (162,35,6); (278,35,5); (305,35,4); (364,35,4); (377,350,7); (708,40,9); (741,40,3); (969,40,5); (25,41,8); (224,41,4); (231,41,9); (431,41,8); (441,41,3); (562,41,7); (753,41,7); (789,41,9); (816,41,6); (887,41,7); (971,41,8); (24,4247,9); (688,47,5); (692,47,4); (824,47,7); (994,47,6); (65,48,3); (134,48,6); (157,48,7); (257,48,8); (335,48,4); (355,48,7); (384,48,9); (461,48,7); (484,48,5); (570,48,4); (572,54,7); (807,54,4); (133,55,9); (192,55,9); (271,55,5); (274,55,9); (387,55,6); (468,55,3); (808,55,8); (843,55,3); (865,55,9); (905,55,9); (999,55,8); (11,56,7); (18,56,3); (215,56,6); (845,61,6); (848,61,6); (871,61,6); (67,62,8); (249,62,6); (250,62,9); (511,62,3); (593,62,5); (661,62,3); (919,62,8); (93,63,7); (298,63,4); (439,63,7); (483,63,7); (573,63,4,6); (368,70,9); (503,70,5); (653,70,6); (697,70,9); (951,70,3); (137,71,9); (361,71,5); (398,71,5); (450,71,8); (481,71,4); (645,71,6); (731,71,9); (745,71,7); (105,72,3); (284,7113,79,3); (268,79,7); (307,79,4); (373,79,9); (425,79,5); (460,79,6); (549,79,4); (698,79,6); (803,79,8); (890,79,8); (893,79,4); (914,79,5); (989,79,3); (66,80,9); (171,80,3); (508,87,4); (839,87,5); (922,87,7); (253,88,4); (302,88,4); (342,88,4); (363,88,4); (369,88,9); (435,88,9); (492,88,4); (512,88,9); (520,88,7); (577,88,5); (582,88,7); (723,88,3); (92,94,3); (950,94,3); (54,95,4); (148,95,5); (474,95,9); (548,95,3); (607,95,8); (757,95,4); (835,95,5); (867,95,8); (886,95,4); (20,96,7); (84,96,5); (97,96,8); (123,96,4); (308,
Queue 1: closed
Queue 2: closed
Queue 3: closed
Queue 4: closed
Queue 5: closed
Queue 6: closed
Queue 7: closed
Queue 8: closed
Queue 9: closed
Queue 10: closed
Queue 11: closed
Queue 12: closed
Queue 13: closed

Time: 12
Waiting clients: (96,13,7); (185,13,9); (240,13,9); (463,13,7); (652,13,8); (685,13,8); (857,13,7); (944,13,6); (447,14,8); (476,14,5); (507,14,5); (557,143,20,5); (647,20,4); (720,20,3); (790,20,5); (838,20,7); (879,20,4); (906,20,3); (966,20,4); (6,21,9); (167,21,6); (283,21,3); (292,21,4); (643,21,9); (659,29,3); (704,29,8); (925,29,4); (153,30,8); (193,30,4); (222,30,5); (236,30,6); (243,30,9); (513,30,7); (542,30,4); (599,30,8); (627,30,8); (730,30,4); (83,7); (598,37,5); (606,37,4); (620,37,8); (644,37,7); (673,37,7); (737,37,5); (939,37,9); (986,37,3); (180,38,6); (214,38,6); (258,38,7); (402,38,3); (519,34,9); (272,44,7); (412,44,9); (597,44,6); (747,44,4); (754,44,8); (765,44,6); (771,44,9); (862,44,4); (866,44,7); (942,44,5); (336,45,7); (353,45,3); (383,322,51,6); (331,51,3); (389,51,9); (517,51,5); (527,51,5); (684,51,9); (739,51,3); (34,52,9); (51,52,8); (124,52,4); (315,52,7); (393,52,5); (637,52,4); (71,58,7); (434,58,7); (454,58,4); (478,58,9); (841,58,9); (907,58,7); (917,58,6); (930,58,4); (19,59,9); (72,59,3); (147,59,9); (234,59,9); (327,59,9); (395,65,8); (860,65,6); (9,66,4); (29,66,3); (41,66,6); (117,66,7); (188,66,3); (211,66,3); (248,66,9); (466,66,6); (491,66,5); (514,66,3); (701,66,6); (718,666); (1,75,3); (30,75,3); (158,75,4); (245,75,3); (262,75,5); (275,75,4); (449,75,5); (631,75,7); (875,75,4); (189,76,6); (242,76,7); (247,76,4); (269,76,4); (769,83,5); (826,83,4); (959,83,8); (156,84,5); (233,84,6); (235,84,5); (238,84,7); (404,84,4); (736,84,9); (752,84,9); (756,84,8); (834,84,6); (873,84,8726,90,6); (763,90,9); (911,90,3); (947,90,3); (953,90,6); (962,90,9); (62,91,6); (326,91,5); (328,91,4); (366,91,4); (488,91,7); (499,91,5); (506,91,3); (312,99,8); (365,99,4); (728,99,6); (22,100,4); (320,100,4); (337,100,5); (552,100,7); (568,100,3); (651,100,3); (667,100,3); (940,100,8);
Queue 1: (63,10,6); (650,12,3);
Queue 2: (74,10,3); (963,11,8);
Queue 3: (146,10,2); (624,11,6);
Queue 4: (357,10,7); (683,12,9);
Queue 5: (374,10,4); (295,12,6);
Queue 6: (379,10,7); (699,12,7);
Queue 7: (423,10,3); (127,12,4); (759,12,7);
Queue 8: (430,10,2); (773,11,9);
Queue 9: (596,10,6); (680,12,6);
Queue 10: (687,10,5); (348,12,9);
Queue 11: (707,10,7); (811,12,5);
Queue 12: (744,10,4); (297,12,8);
Queue 13: (749,10,4); (347,12,4);
Queue 14: (830,10,1); (314,11,7);
Queue 15: (891,10,7); (854,12,8);
Queue 16: (954,10,5); (490,12,3);
Queue 17: (8,11,2); (820,11,3); (533,12,7);
Queue 18: (13,11,7); (864,12,7);
Queue 19: (53,11,3); (228,12,4); (884,12,4);
Queue 20: (220,11,2); (936,11,6);
```


Time: 17

Waiting clients: (45,18,5); (202,18,3); (294,18,5); (504,18,4); (609,18,6); (621,18,9); (727,18,9); (767,18,7); (814,18,4); (152,26,9); (537,26,8); (608,26,6); (878,26,5); (208,27,9); (267,27,9); (291,27,5); (306,27,7); (397,27,5); (470,27,4); (364,35,4); (377,35,6); (479,35,8); (524,35,3); (556,35,3); (675,35,7); (696,35,5); (770,35,7); (829,35,5); (842,35,9); (971,41,8); (24,42,7); (52,42,3); (98,42,4); (260,42,5); (344,42,9); (475,42,6); (700,42,7); (748,42,4); (825,42,3); (15,5); (570,48,4); (572,48,3); (725,48,8); (788,48,5); (806,48,8); (851,48,7); (992,48,9); (998,48,7); (81,49,8); (108,49,5); (207,49,3); (380,49,6); (588,49,3); (694,49,9); (793,49,5); (859,6,7); (18,56,3); (215,56,7); (300,56,4); (354,56,6); (358,56,3); (372,56,6); (498,56,9); (502,56,5); (554,56,8); (772,56,3); (863,56,8); (901,56,5); (915,56,8); (86,5); (483,63,7); (573,63,4); (587,63,4); (677,63,3); (779,63,8); (955,63,8); (960,63,4); (1000,63,7); (85,64,8); (109,64,6); (105,72,3); (284,72,8); (581,72,8); (605,72,5); (847,72,3); (889,72,8); (931,72,7); (956,72,6); (7,73,9); (44,73,6); (1,80,9); (171,80,3); (583,80,9); (970,80,4); (972,80,6); (40,81,4); (101,81,3); (141,81,4); (142,81,6); (501,81,4); (715,582,88,7); (723,88,3); (740,88,9); (795,88,9); (800,88,4); (852,88,7); (900,88,8); (988,88,3); (47,89,6); (61,89,4); (77,8); (123,96,4); (308,96,5); (391,96,3); (509,96,7); (551,96,5); (602,96,4); (612,96,7); (618,96,9); (626,96,7); (46,97,4); Queue 1: (63,10,1); (650,12,3); (685,13,8); (844,15,7); (550,17,8); Queue 2: (963,11,6); (944,13,6); (928,15,8); (929,17,7); Queue 3: (624,11,3); (96,13,7); (159,15,4); (796,16,8); Queue 4: (357,10,2); (683,12,9); (564,15,7); (388,17,3); (978,17,5); Queue 5: (295,12,5); (857,13,7); (106,16,7); (625,17,4); Queue 6: (379,10,2); (699,12,7); (623,14,6); (221,17,3); (403,17,9); Queue 7: (127,12,2); (759,12,7); (751,14,9); (411,17,4); Queue 8: (773,11,6); (447,14,8); (174,17,3); (311,17,9); Queue 9: (596,10,1); (680,12,6); (507,14,5); (194,16,7); (656,17,5); Queue 10: (348,12,9); (766,14,9); (529,17,8); Queue 11: (707,10,2); (811,12,5); (557,14,5); (277,16,7); (711,17,6); Queue 12: (297,12,7); (586,14,3); (197,15,3); (547,16,8); Queue 13: (347,12,3); (185,13,9); (325,16,4); (255,17,5); Queue 14: (314,11,3); (240,13,9); (528,16,5); (323,17,9); Queue 15: (891,10,2); (854,12,8); (276,15,3); (669,16,4); (386,17,4); Queue 16: (490,12,3); (463,13,7); (394,15,3); (709,16,8); Queue 17: (533,12,7); (590,14,7); (210,17,8); Queue 18: (13,11,2); (864,12,7); (827,14,9); (532,17,4); Queue 19: (228,12,2); (884,12,4); (476,14,5); (710,15,8); (855,17,3); Queue 20: (936,11,3); (652,13,8); (764,15,8); (856,17,5);

Time: 41

Waiting clients: (24,42,7); (52,42,3); (98,42,4); (260,42,5); (344,42,9); (475,42,6); (700,42,7); (748,42,4); (825,42,3); (831,42,7); (975,42,3); (288,43,7); (371,43,8); (437,43,3); (473,43,8,5); (570,48,4); (572,48,3); (725,48,8); (788,48,5); (806,48,8); (851,48,7); (992,48,9); (998,48,7); (81,49,8); (108,49,5); (207,49,3); (380,49,6); (588,49,3); (694,49,9); (793,49,5); (859,6,7); (18,56,3); (215,56,7); (300,56,4); (354,56,6); (358,56,3); (372,56,6); (498,56,9); (502,56,5); (554,56,8); (772,56,3); (863,56,8); (901,56,5); (915,56,8); (86,5); (483,63,7); (573,63,4); (587,63,4); (677,63,3); (779,63,8); (955,63,8); (960,63,4); (1000,63,7); (85,64,8); (109,64,6); (105,72,3); (284,72,8); (581,72,8); (605,72,5); (847,72,3); (889,72,8); (931,72,7); (956,72,6); (7,73,9); (44,73,6); (55,73,3); (163,73,9); (204,73,5); (565,73,9); (743,73,8); (3,74,5); (66,80,9); (171,80,3); (583,80,9); (970,80,4); (972,80,6); (40,81,4); (101,81,3); (141,81,4); (142,81,6); (501,81,4); (715,81,4); (732,81,6); (837,81,6); (846,81,7); (183,82,6); (186,82,4); (582,88,7); (723,88,3); (740,88,9); (795,88,9); (800,88,4); (852,88,7); (900,88,8); (988,88,3); (47,89,6); (61,89,4); (77,89,9); (544,89,8); (663,89,9); (668,89,9); (671,89,8); (815,89,5); (123,96,4); (308,96,5); (391,96,3); (509,96,7); (551,96,5); (602,96,4); (612,96,7); (618,96,9); (626,96,7); (46,97,6); (78,97,5); (144,97,3); (594,97,8); (785,97,9); (881,97,7); (909,97,5); Queue 1: (550,17,3); (908,19,9); (455,23,3); (902,23,5); (213,25,3); (397,27,5); (664,28,7); (730,30,4); (48,32,8); (885,34,8); (977,36,6); (519,38,3); (495,39,9); (441,41,3); Queue 2: (929,17,3); (912,19,6); (643,21,9); (416,24,9); (338,28,6); (236,30,6); (64,32,6); (281,34,8); (516,36,5); (606,37,4); (979,38,8); (741,40,3); (562,41,7); Queue 3: (609,18,3); (196,20,6); (823,21,8); (436,24,8); (103,28,8); (542,30,4); (850,31,7); (399,34,3); (162,35,6); (629,36,5); (673,37,7); (424,40,8); (887,41,7); Queue 4: (978,17,2); (244,19,9); (927,22,3); (768,23,6); (376,25,4); (470,27,4); (746,28,4); (153,30,8); (198,32,6); (738,34,5); (696,35,5); (256,37,8); (742,39,4); (546,40,8); Queue 5: (861,18,3); (26,20,7); (933,21,8); (777,24,3); (152,26,9); (535,29,7); (477,31,4); (494,32,4); (471,34,7); (83,36,7); (737,37,5); (805,39,7); (969,40,5); Queue 6: (483,17,3); (76,20,6); (659,21,8); (23,24,5); (608,26,6); (813,28,7); (832,30,7); (523,32,6); (278,35,5); (611,36,4); (598,37,5); (990,38,7); (632,40,9); Queue 7: (621,18,7); (906,20,3); (445,22,9); (888,24,8); (341,28,8); (895,30,5); (237,32,3); (734,33,9); (90,36,7); (939,37,9); (591,40,4); (231,41,9); Queue 8: (311,17,2); (420,19,4); (790,20,5); (2,23,6); (75,24,3); (456,25,8); (896,28,7); (904,30,7); (758,32,7); (556,35,3); (99,36,8); (214,38,6); (426,40,8); (971,41,8); Queue 9: (946,18,5); (543,20,5); (678,22,3); (633,23,9); (878,26,5); (518,28,3); (561,29,3); (243,30,9); (797,32,6); (305,35,4); (111,36,9); (610,38,3); (817,39,9); (753,41,7); Queue 10: (529,17,1); (443,19,3); (647,20,4); (686,21,4); (642,23,8); (537,26,8); (266,29,8); (487,31,5); (60,33,6); (364,35,4); (409,36,4); (343,37,7); (319,39,6); (646,40,7); Queue 11: (711,17,1); (122,19,9); (689,22,9); (102,25,3); (208,27,9); (704,29,8); (187,32,8); (89,35,8); (14,37,3); (620,37,8); (438,40,8); Queue 12: (45,18,2); (452,19,5); (966,20,4); (206,23,4); (910,23,7); (267,27,9); (925,29,4); (68,31,5); (270,32,7); (160,35,6); (622,36,8); (937,38,4); (821,39,4); (660,40,3); (431,41,8); Queue 13: (949,18,3); (172,20,5); (283,21,3); (405,23,5); (16,24,9); (941,27,3); (121,29,9); (569,31,8); (545,34,9); (657,36,6); (258,38,7); (539,40,6); (789,41,9); Queue 14: (323,17,1); (467,19,9); (422,23,9); (706,25,7); (617,28,6); (513,30,7); (351,32,3); (762,33,3); (810,34,5); (770,35,7); (644,37,7); (828,39,9); (52,42,3); (831,42,7); Queue 15: (294,18,1); (485,19,9); (422,23,4); (920,23,5); (721,25,6); (139,28,8); (599,30,8); (648,33,6); (377,35,6); (918,36,5); (986,37,3); (88,39,4); (459,40,5); (25,41,8); Queue 16: (504,18,1); (230,19,6); (6,21,9); (21,24,6); (291,27,5); (655,28,5); (193,30,4); (392,31,8); (926,33,4); (479,35,8); (465,37,9); (219,40,6); (224,41,6); Queue 17: (727,18,7); (167,21,6); (672,23,4); (87,24,9); (199,28,6); (222,30,5); (849,31,8); (714,34,6); (829,35,5); (560,37,6); (118,39,6); (601,40,7); Queue 18: (767,18,5); (720,20,3); (292,21,4); (505,23,5); (131,24,3); (967,25,6); (218,28,9); (128,31,5); (381,32,9); (675,35,7); (566,37,7); (385,39,3); (500,40,7); (816,41,6); Queue 19: (814,18,2); (567,19,4); (838,20,7); (693,23,6); (114,25,3); (306,27,7); (339,29,5); (627,30,8); (705,33,6); (524,35,3); (842,35,9); (402,38,3); (406,39,6); (695,40,7); Queue 20: (965,18,6); (879,20,4); (691,22,5); (958,23,4); (173,25,9); (254,29,7); (155,31,9); (32,34,7); (31,36,3); (935,36,5); (180,38,6); (225,40,4); (708,40,9);

Time: 42

Waiting clients: (288,43,7); (371,43,8); (437,43,3); (473,43,7); (530,43,4); (595,43,5); (616,43,7); (654,43,3); (781,43,8); (784,43,3); (868,43,5); (883,43,4); (934,43,7); (945,43,7); (35,49,3); (380,49,6); (588,49,3); (694,49,9); (793,49,5); (859,49,5); (869,49,9); (923,49,4); (190,50,4); (209,50,8); (229,50,5); (733,50,3); (778,50,6); (818,50,4); (898,50,6); (924,50,6); (1,56,6); (772,56,3); (863,56,8); (901,56,5); (915,56,8); (86,57,3); (95,57,5); (130,57,3); (303,57,4); (382,57,3); (407,57,3); (457,57,9); (464,57,5); (510,57,8); (619,57,6); (682,57,4); (874,9); (145,64,8); (282,64,6); (316,64,8); (641,64,4); (729,64,9); (760,64,5); (786,64,7); (812,64,6); (858,64,6); (932,64,8); (264,65,9); (285,65,3); (317,65,4); (472,65,4); (400,65,5); (482,3); (163,73,9); (204,73,5); (565,73,9); (743,73,8); (3,74,5); (33,74,9); (120,74,7); (154,74,7); (217,74,4); (329,74,6); (346,74,5); (534,74,9); (571,74,4); (603,74,3); (666,74,6); (724,74,3); (732,81,6); (837,81,6); (846,81,7); (183,82,6); (186,82,4); (419,82,5); (553,82,7); (558,82,9); (799,82,4); (819,82,7); (94,83,8); (104,83,8); (151,83,7); (181,83,5); (263,83,4); (318,83,8); (544,89,8); (663,89,9); (668,89,9); (671,89,8); (815,89,5); (882,89,8); (921,89,9); (10,90,6); (49,90,7); (80,90,5); (110,90,8); (241,90,3); (279,90,8); (321,90,4); (340,90,3); (469,90,5); (97,5); (144,97,3); (594,97,8); (785,97,9); (881,97,7); (909,97,9); (12,90,4); (17,90,7); (50,90,7); (59,90,7); (129,90,5); (370,90,8); (408,90,5); (522,90,5); (574,90,4); (43,99,6); (57,99,8); Queue 1: (550,17,2); (908,19,9); (455,23,3); (902,23,5); (213,25,3); (397,27,5); (664,28,7); (730,30,4); (48,32,8); (885,34,8); (977,36,6); (519,38,3); (495,39,9); (441,41,3); (344,42,9); Queue 2: (929,17,2); (912,19,6); (643,21,9); (416,24,9); (338,28,6); (236,30,6); (64,32,6); (281,34,8); (516,36,5); (606,37,4); (979,38,8); (741,40,3); (562,41,7); Queue 3: (609,18,3); (196,20,6); (823,21,8); (436,24,8); (103,28,8); (542,30,4); (850,31,7); (399,34,3); (162,35,6); (629,36,5); (673,37,7); (424,40,8); (887,41,7); Queue 4: (978,17,1); (244,19,9); (927,22,3); (768,23,6); (376,25,4); (470,27,4); (746,28,4); (153,30,8); (198,32,6); (738,34,5); (696,35,5); (256,37,8); (742,39,4); (546,40,8); (475,42,6); Queue 5: (861,18,2); (26,20,7); (933,21,8); (777,24,3); (152,26,9); (535,29,7); (477,31,4); (494,32,4); (471,34,7); (83,36,7); (737,37,5); (805,39,7); (969,40,5); (700,42,7); Queue 6: (483,17,2); (76,20,6); (659,21,8); (23,24,5); (608,26,6); (813,28,7); (832,30,7); (523,32,6); (278,35,5); (611,36,4); (598,37,5); (990,38,7); (632,40,9); (975,42,3); Queue 7: (621,18,6); (906,20,3); (445,22,9); (888,24,8); (341,28,8); (895,30,5); (237,32,3); (734,33,9); (90,36,7); (939,37,9); (591,40,4); (231,41,9); Queue 8: (311,17,1); (420,19,4); (790,20,5); (2,23,6); (75,24,3); (456,25,8); (896,28,7); (904,30,7); (758,32,7); (556,35,3); (99,36,8); (214,38,6); (426,40,8); (971,41,8); Queue 9: (946,18,4); (543,20,5); (678,22,3); (633,23,9); (878,26,5); (518,28,3); (561,29,3); (243,30,9); (797,32,6); (305,35,4); (111,36,9); (610,38,3); (817,39,9); (753,41,7); Queue 10: (529,17,1); (443,19,3); (647,20,4); (686,21,4); (642,23,8); (537,26,8); (266,29,8); (487,31,5); (60,33,6); (364,35,4); (409,36,4); (343,37,7); (319,39,6); (646,40,7); (748,42,4); Queue 11: (712,19,9); (689,22,9); (102,25,3); (208,27,9); (704,29,8); (187,32,8); (89,35,8); (14,37,3); (620,37,8); (438,40,8); (24,42,7); Queue 12: (45,18,1); (452,19,5); (966,20,4); (206,23,4); (910,23,7); (267,27,9); (925,29,4); (68,31,5); (270,32,7); (160,35,6); (622,36,8); (937,38,4); (821,39,4); (660,40,3); (431,41,8); Queue 13: (949,18,2); (172,20,5); (283,21,3); (405,23,5); (16,24,9); (941,27,3); (121,29,9); (569,31,8); (545,34,9); (657,36,6); (258,38,7); (539,40,6); (789,41,9); Queue 14: (323,17,1); (467,19,9); (422,23,9); (706,25,7); (617,28,6); (513,30,7); (351,32,3); (762,33,3); (810,34,5); (770,35,7); (644,37,7); (828,39,9); (52,42,3); (831,42,7); Queue 15: (294,18,1); (485,19,9); (422,23,4); (920,23,5); (721,25,6); (139,28,8); (599,30,8); (648,33,6); (377,35,6); (918,36,5); (986,37,3); (88,39,4); (459,40,5); (25,41,8); Queue 16: (504,18,6); (6,21,9); (21,24,6); (291,27,5); (655,28,5); (193,30,4); (392,31,8); (926,33,4); (479,35,8); (465,37,9); (219,40,6); (224,41,6); (98,42,4); Queue 17: (727,18,6); (167,21,6); (672,23,4); (87,24,9); (199,28,6); (222,30,5); (849,31,8); (714,34,6); (829,35,5); (560,37,6); (118,39,6); (601,40,7); (260,42,5); Queue 18: (767,18,4); (720,20,3); (292,21,4); (505,23,5); (131,24,3); (967,25,6); (218,28,9); (128,31,5); (381,32,9); (675,35,7); (566,37,7); (385,39,3); (500,40,7); (816,41,6); Queue 19: (814,18,1); (567,19,4); (838,20,7); (693,23,6); (114,25,3); (306,27,7); (339,29,5); (627,30,8); (705,33,6); (524,35,3); (842,35,9); (402,38,3); (406,39,6); (695,40,7); (825,42,3); Queue 20: (965,18,5); (879,20,4); (691,22,5); (958,23,4); (173,25,9); (254,29,7); (155,31,9); (32,34,7); (31,36,3); (935,36,5); (180,38,6); (225,40,4); (708,40,9);


```

Time: 198
Waiting clients:
Queue 1: (960,63,1); (858,64,6); (514,66,3); (833,67,7); (368,70,9); (33,74,9); (446,76,5); (589,78,7); (66,80,9); (769,83,5); (3
Queue 2: (786,64,7); (701,66,6); (313,69,3); (984,69,9); (565,73,9); (414,76,6); (649,78,4); (373,79,9); (94,83,8); (410,85,8); (
Queue 3: (93,63,2); (264,65,9); (168,68,8); (361,71,5); (163,73,9); (631,75,7); (774,78,4); (425,79,5); (101,81,3); (799,82,4); (
Queue 4: (145,64,6); (117,66,7); (349,69,6); (398,71,5); (204,73,5); (571,74,4); (875,75,4); (280,77,9); (890,79,8); (263,83,4); (
Queue 5: (812,64,6); (188,66,3); (195,67,4); (415,69,5); (951,70,3); (284,72,8); (603,74,3); (245,75,3); (536,76,8); (70,79,6); (
Queue 6: (641,64,3); (480,65,5); (775,66,4); (717,68,4); (226,70,5); (581,72,8); (666,74,6); (538,76,9); (460,79,6); (715,81,4); (
Queue 7: (1000,63,4); (540,65,9); (448,69,9); (605,72,5); (217,74,4); (158,75,4); (578,76,6); (899,78,7); (972,80,6); (585,83,7); (
Queue 8: (919,62,2); (285,65,3); (29,66,3); (107,67,4); (170,69,4); (293,70,7); (956,72,6); (724,74,3); (262,75,5); (350,77,8); (
Queue 9: (439,63,2); (317,65,4); (211,66,3); (432,67,8); (503,70,5); (847,72,3); (743,73,8); (189,76,6); (299,78,8); (171,80,3); (
Queue 10: (729,64,8); (164,67,6); (872,69,4); (137,71,9); (329,74,6); (242,76,7); (780,78,7); (583,80,9); (156,84,5); (995,85,8); (
Queue 11: (779,63,4); (662,65,3); (718,66,6); (493,69,9); (889,72,8); (1,75,3); (247,76,4); (440,77,7); (549,79,4); (40,81,4); (8
Queue 12: (85,64,5); (41,66,6); (296,68,9); (731,71,9); (938,74,8); (776,77,6); (113,79,3); (893,79,4); (837,81,6); (233,84,6); (
Queue 13: (282,64,4); (802,65,8); (201,69,7); (450,71,8); (346,74,5); (275,75,4); (100,77,6); (980,78,4); (914,79,5); (419,82,5); (
Queue 14: (955,63,4); (860,65,6); (42,68,9); (481,71,4); (7,73,9); (449,75,5); (801,77,5); (5,79,9); (553,82,7); (752,84,9); (112
Queue 15: (483,63,2); (472,65,4); (248,66,9); (957,69,5); (745,71,7); (534,74,9); (200,77,3); (352,78,3); (69,79,7); (142,81,6); (
Queue 16: (109,64,3); (482,65,6); (702,67,3); (261,69,4); (310,70,6); (931,72,7); (961,74,6); (628,76,8); (268,79,7); (846,81,7); (
Queue 17: (760,64,4); (9,66,4); (169,67,5); (716,69,4); (653,70,6); (44,73,6); (987,74,6); (750,76,8); (307,79,4); (970,80,4); (5
Queue 18: (115,64,6); (466,66,6); (304,69,8); (105,72,3); (55,73,3); (120,74,7); (269,76,4); (973,77,9); (989,79,3); (501,81,4); (
Queue 19: (250,62,1); (932,64,8); (783,67,8); (697,70,9); (154,74,7); (309,76,6); (375,78,5); (698,79,6); (183,82,6); (238,84,7); (
Queue 20: (316,64,6); (491,66,5); (670,68,8); (645,71,6); (3,74,5); (30,75,3); (332,76,8); (913,78,9); (186,82,4); (635,83,8); (1

Time: 199
Waiting clients:
Queue 1: (858,64,6); (514,66,3); (833,67,7); (368,70,9); (33,74,9); (446,76,5); (589,78,7); (66,80,9); (769,83,5); (362,85,8); (5
Queue 2: (786,64,6); (701,66,6); (313,69,3); (984,69,9); (565,73,9); (414,76,6); (649,78,4); (373,79,9); (94,83,8); (410,85,8); (
Queue 3: (93,63,1); (264,65,9); (168,68,8); (361,71,5); (163,73,9); (631,75,7); (774,78,4); (425,79,5); (101,81,3); (799,82,4); (
Queue 4: (145,64,5); (117,66,7); (349,69,6); (398,71,5); (204,73,5); (571,74,4); (875,75,4); (280,77,9); (890,79,8); (263,83,4); (
Queue 5: (812,64,5); (188,66,3); (195,67,4); (415,69,5); (951,70,3); (284,72,8); (603,74,3); (245,75,3); (536,76,8); (70,79,6); (
Queue 6: (641,64,2); (480,65,5); (775,66,4); (717,68,4); (226,70,5); (581,72,8); (666,74,6); (538,76,9); (460,79,6); (715,81,4); (
Queue 7: (1000,63,3); (540,65,9); (448,69,9); (605,72,5); (217,74,4); (158,75,4); (578,76,6); (899,78,7); (972,80,6); (585,83,7); (
Queue 8: (919,62,1); (285,65,3); (29,66,3); (107,67,4); (170,69,4); (293,70,7); (956,72,6); (724,74,3); (262,75,5); (350,77,8); (
Queue 9: (439,63,1); (317,65,4); (211,66,3); (432,67,8); (503,70,5); (847,72,3); (743,73,8); (189,76,6); (299,78,8); (171,80,3); (
Queue 10: (729,64,7); (164,67,6); (872,69,4); (137,71,9); (329,74,6); (242,76,7); (780,78,7); (583,80,9); (156,84,5); (995,85,8); (
Queue 11: (779,63,3); (662,65,3); (718,66,6); (493,69,9); (889,72,8); (1,75,3); (247,76,4); (440,77,7); (549,79,4); (40,81,4); (8
Queue 12: (85,64,4); (41,66,6); (296,68,9); (731,71,9); (938,74,8); (776,77,6); (113,79,3); (893,79,4); (837,81,6); (233,84,6); (
Queue 13: (282,64,3); (802,65,8); (201,69,7); (450,71,8); (346,74,5); (275,75,4); (100,77,6); (980,78,4); (914,79,5); (419,82,5); (
Queue 14: (955,63,3); (860,65,6); (42,68,9); (481,71,4); (7,73,9); (449,75,5); (801,77,5); (5,79,9); (553,82,7); (752,84,9); (112
Queue 15: (483,63,1); (472,65,4); (248,66,9); (957,69,5); (745,71,7); (534,74,9); (200,77,3); (352,78,3); (69,79,7); (142,81,6); (
Queue 16: (109,64,2); (482,65,6); (702,67,3); (261,69,4); (310,70,6); (931,72,7); (961,74,6); (628,76,8); (268,79,7); (846,81,7); (
Queue 17: (760,64,3); (9,66,4); (169,67,5); (716,69,4); (653,70,6); (44,73,6); (987,74,6); (750,76,8); (307,79,4); (970,80,4); (5
Queue 18: (115,64,5); (466,66,6); (304,69,8); (105,72,3); (55,73,3); (120,74,7); (269,76,4); (973,77,9); (989,79,3); (501,81,4); (
Queue 19: (932,64,8); (783,67,8); (697,70,9); (154,74,7); (309,76,6); (375,78,5); (698,79,6); (183,82,6); (238,84,7); (378,86,4); (
Queue 20: (316,64,5); (491,66,5); (670,68,8); (645,71,6); (3,74,5); (30,75,3); (332,76,8); (913,78,9); (186,82,4); (635,83,8); (1

Average waiting time: 6,01
Average serving time: 6,01
Peak hour: 100

```

6. Concluzii

Se poate observa cum folosind paradigma de programare multi-threading putem minimiza cu mult timpul de așteptare și de servire a clienților sau a oricărui tip de task, dacă totul ar fi fost secvențial și realizat pe un singur thread timpii ar fi fost imenși comparativ cu timpii obținuți prin această metodă.

De asemenea un mod foarte eficient prin care putem folosi resursele unui sistem, care în zilele noastre au mai multe procesoare sau chiar pe un sistem distribuit sau pe un cluster de procesoare, la care dacă folosim algoritmi multi threading castigam foarte multă putere de procesare într-un timp scurt, fiind mai ales benefic pentru lucrul cu seturi de date mari sau imense, unde probabil de multe ori este singura modalitate de procesare într-un timp rezonabil.

7. Bibliografie

ASSIGNMENT_2_SUPPORT_PRESENTATION

<https://docs.oracle.com/javase/8/docs/api/java/util/concurrent/atomic/AtomicInteger.html>

<https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/BlockingQueue.html>

