

# *Simulazione PMCSN*

Luca Mastrobattista, 0292461

## Indice

<b>1</b>	<b>Traccia della Simulazione</b>	<b>2</b>
1.1	Caso di studio . . . . .	2
1.2	Obiettivi . . . . .	2
<b>2</b>	<b>Modello concettuale</b>	<b>3</b>
2.1	Visualizzazione grafica . . . . .	3
2.2	Eventi del sistema e variabili di stato . . . . .	3
2.3	Eventi . . . . .	3
2.4	Variabili di stato . . . . .	3
<b>3</b>	<b>Modello delle specifiche</b>	<b>4</b>
3.1	Assunzioni . . . . .	4
<b>4</b>	<b>Modello computazionale</b>	<b>6</b>
4.1	configurations.py . . . . .	6
4.2	simulation.py . . . . .	6
<b>5</b>	<b>Verifica e validazione</b>	<b>8</b>
5.1	Tempo di interarrivo medio . . . . .	8
5.2	Numero medio job in coda e tempo medio in coda . . . . .	9
5.3	Tempo di risposta e numero medio di job nel nodo . . . . .	12
5.4	Utilizzazione . . . . .	13
<b>6</b>	<b>Conclusioni</b>	<b>14</b>

# 1 Traccia della Simulazione

## 1.1 Caso di studio

Si vuole valutare l'idea di aprire un locale in un piccolo paese. L'attività dovrà offrire ai clienti servizi di bar e di pizzeria. Il locale è già provvisto di tutto l'arredamento e lo si affitterà per un costo 1500 € al mese. Il pizzaiolo scelto per i servizi di pizzeria ha comunicato che, nel forno presente, si possono preparare contemporaneamente al massimo 2 pizze, ognuna delle quali può essere preparata con un tempo medio di 3 minuti. Inoltre, si è già trovato un accordo con lui: lavorerà ogni giorno della settimana dalle ore 20:00 alle ore 22:00, percependo una paga di 50 € al giorno con il vincolo che tutte le ordinazioni arrivate precedentemente alle 22:00 verranno sempre completate, anche se per farlo dovrà continuare a sfornare pizze oltre questo orario. Per quanto riguarda le richieste al bar, si vuole che "l'ultimo giro" venga chiamato alle ore 02:00, senza accettare altre richieste successive a quell'orario ma completando tutte quelle ancora presenti.

Dopo un'osservazione settimanale di altri locali che offrono servizi simili, si è notato che il numero di clienti che arrivano al locale si differenzia per fasce orarie della giornata diverse. Inoltre, nel fine settimana, la frequenza delle richieste nelle fasce orarie identificate è maggiore rispetto a quella settimanale. Si riportano di seguito delle tabelle riassuntive per le frequenze di arrivo nei due casi.

Fascia oraria	$\lambda_{B,W}$	$\lambda_{P,W}$
07:00 → 13:00	10 j/h	✗
13:00 → 18:00	5 j/h	✗
18:00 → 20:00	10 j/h	✗
20:00 → 22:00	15 j/h	10 j/h
22:00 → 02:00	10 j/h	✗

*Frequenze di arrivo settimanali*

Fascia oraria	$\lambda_{B,WE}$	$\lambda_{P,WE}$
07:00 → 13:00	15 j/h	✗
13:00 → 18:00	10 j/h	✗
18:00 → 20:00	20 j/h	✗
20:00 → 22:00	15 j/h	30 j/h
22:00 → 02:00	30 j/h	✗

*Frequenze di arrivo fine-settimanali*

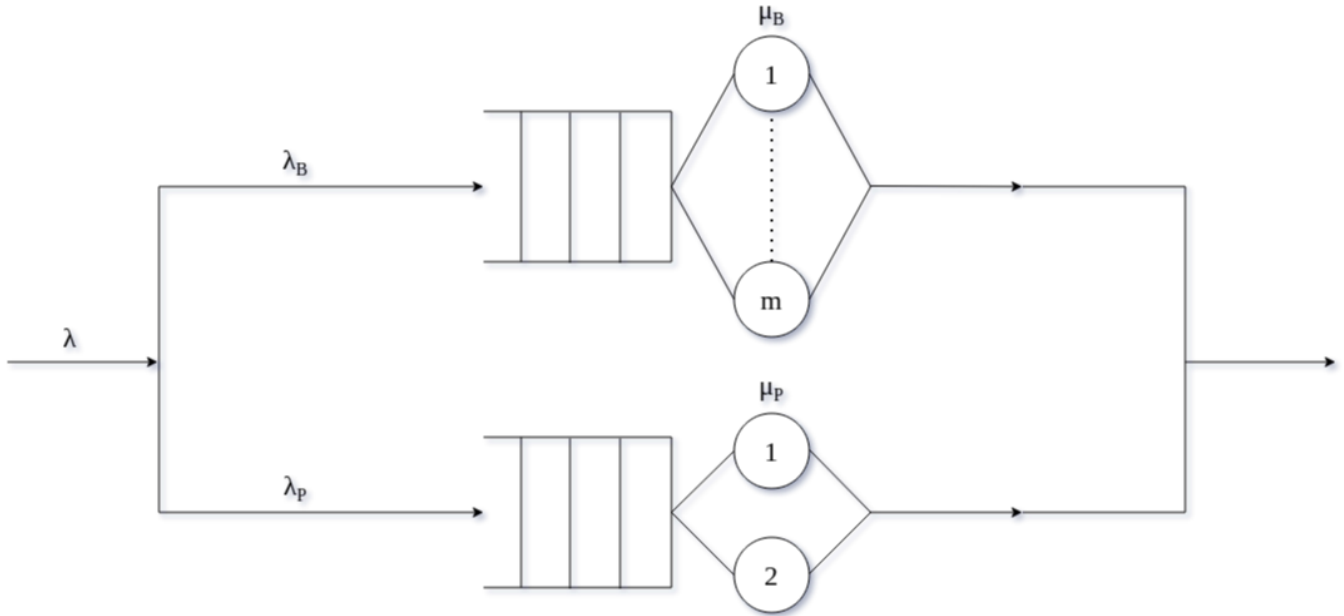
## 1.2 Obiettivi

Obiettivo dell'analisi è la valutazione del numero di baristi da assumere, col fine ultimo di massimizzare i guadagni. Si considera una paga di 40 € al giorno per ognuno di loro; si assume che ogni barista sia in grado di servire un'ordinazione in 2 minuti, durante i quali si dedica esclusivamente a quella richiesta. Si assume inoltre che il prezzo medio delle richieste di tipo B sia di 5 €, mentre quello delle richieste di tipo P sia di 7 €. Si vuole, però, che i seguenti vincoli siano sempre rispettati:

- Ogni ordinazione al bar deve essere servita in un tempo strettamente minore di 3 minuti;
- Ogni ordinazione per la pizzeria sia servita in un tempo strettamente minore di 10 minuti

## 2 Modello concettuale

### 2.1 Visualizzazione grafica



La frequenza di arrivo  $\lambda$  si compone della frequenza di arrivo  $\lambda_B$  e  $\lambda_P$ , che sono rispettivamente i tassi di arrivo per richieste al bar e alla pizzeria. Una opportuna coda per ogni tipologia rappresenta la lista di attesa della tipologia stessa. Ogni servente di tipo  $B$  rappresenta un barista assunto, che lavora con una frequenza  $\mu_B$ . Ogni servente di tipo  $P$ , invece, rappresenta una delle due richieste che il pizzaiolo è in grado di gestire contemporaneamente.

### 2.2 Eventi del sistema e variabili di stato

#### 2.3 Eventi

Indice	Descrizione	Attributo 1	Attributo 2
1	Arrivo di tipo B	t	x
2	Completamento dal server $B_1$	t	x
..	..	t	x
$m + 1$	Completamento dal server $B_m$	t	x
$m + 2$	Arrivo di tipo P	t	x
$m + 3$	Completamento dal server $P_1$	t	x
$m + 4$	Completamento dal server $P_2$	t	x
$m + 5$	Evento di campionamento	t	x

L'attributo  $t$  identifica il tempo schedulato per la successiva occorrenza dell'evento di quel tipo; l'attributo  $x$  identifica lo stato di attività dell'evento.

### 2.4 Variabili di stato

- $l_B(t)$ : numero di richieste di tipo B al centro all'istante  $t$
- $l_P(t)$ : numero di richieste di tipo P al centro all'istante  $t$

- $X_s(t)$ : stato del servente  $s$  all'istante  $t$ , con  $s \in [1, m+2]$ .  
Si è indicato con  $m+1$  e  $m+2$  i serventi relativi alla pizzeria.

$$X_s(t) = \begin{cases} 1 & \text{se servente } s \text{ è occupato} \\ 0 & \text{altrimenti} \end{cases}$$

### 3 Modello delle specifiche

#### 3.1 Assunzioni

- Stato iniziale vuoto:

$$l_B(0) + l_P(0) = P(0) + B(0) = 0$$

Come conseguenza, il primo evento deve essere necessariamente un Arrivo.

- Stato finale di ogni giorno vuoto:

$$X_s(T) = 0 \quad \forall s \in \mathcal{B} \cup \mathcal{P}$$

Con  $T$  tempo di chiusura giornaliero e  $\mathcal{B} \cup \mathcal{P}$  l'unione dell'insieme dei serventi di tipo  $B$  e  $P$ .

- Il processo degli arrivi di tipo  $B$  deve essere interrotto alle 02:00, mentre quello degli arrivi di tipo  $P$  deve essere interrotto alle 22:00.
- Alla chiusura del processo degli arrivi, le richieste in coda continuano ad essere servite fino al loro esaurimento.
- L'ultimo evento non può essere un arrivo, e sarà quindi o una partenza o un campionamento.
- Per la selezione del servente si usa una politica di Equity.
- Gli arrivi sono di Poisson e seguono una distribuzione esponenziale con media  $\frac{1}{\lambda_h}$ , con  $\lambda_h$  tasso di arrivo per quella specifica fascia oraria.
- I tempi di servizio di ognuno dei serventi si assume esponenziale. In particolare, ogni servente di tipo  $B$  lavora con frequenza media pari a:  $\frac{1}{\mu_B}$ ; ogni servente di tipo  $P$  lavora con frequenza media pari a:  $\frac{1}{\mu_P}$ .
- Il tempo di fine simulazione è  $\tau$  e si avrà che:

$$\sum_{s \in \mathcal{B} \cup \mathcal{P}} X_s(\tau) = 0$$

- I tassi di arrivo, riportati in minuti, sono i seguenti:

Fascia oraria	$\lambda_{B,W}$	$\lambda_{P,W}$
07:00 → 13:00	$\frac{1}{6}$ j/min	<b>X</b>
13:00 → 18:00	$\frac{1}{12}$ j/min	<b>X</b>
18:00 → 20:00	$\frac{1}{6}$ j/min	<b>X</b>
20:00 → 22:00	$\frac{1}{4}$ j/min	$\frac{1}{6}$ j/min
22:00 → 02:00	$\frac{1}{6}$ j/min	<b>X</b>

*Frequenze di arrivo settimanali*

Fascia oraria	$\lambda_{B,WE}$	$\lambda_{P,WE}$
07:00 → 13:00	$\frac{1}{4}$ j/min	<b>X</b>
13:00 → 18:00	$\frac{1}{6}$ j/min	<b>X</b>
18:00 → 20:00	$\frac{1}{3}$ j/min	<b>X</b>
20:00 → 22:00	$\frac{1}{4}$ j/min	$\frac{1}{2}$ j/min
22:00 → 02:00	$\frac{1}{2}$ j/min	<b>X</b>

*Frequenze di arrivo fine-settimanali*

- Le frequenze di uscita medie del singolo server sono indipendenti dalle fasce orarie e si assumono uguali tra server di tipo B e di tipo P e sono le seguenti:

$\mu_B$
$\frac{1}{2}$ j/min

$\mu_P$
$\frac{1}{3}$ j/min

## 4 Modello computazionale

Il modello computazionale è stato sviluppato in *Python* ed è il programma *simulation.py*; i parametri configurabili sono definiti invece nel file *configurations.py*.

### 4.1 configurations.py

File di configurazione che definisce le costanti per:

- Slot temporali in cui vengono cambiate le frequenze di interarrivo
- La durata di ogni slot temporale
- Tempi in cui si attivano e disattivano gli eventi di arrivo
- La durata della simulazione
- Numero di server di tipo B e P
- Tempi di arrivo per i due tipi
- Frequenze di interarrivo, per ogni tipo e per ogni fascia oraria, sia per i giorni infra-settimanali che fine-settimanali
- La paga media per ogni server di tipo B e di tipo P
- Il costo medio di ogni richiesta di tipo B e di tipo P
- Il costo mensile per l'affitto del locale
- Il tasso dell'iva
- Il costo medio mensile delle bollette

### 4.2 simulation.py

- *indexes*: array che memorizza  $[B(t), P(t)]$
- *numbers*: array che memorizza  $[l_B(t), l_P(t)]$
- *areas*: array che memorizza  $[\int_0^t l_B(\theta)d\theta, \int_0^t l_P(\theta)d\theta]$
- *sum*: array che memorizza, per ogni server  $s$ , il tempo totale di servizio e il numero di richieste completate
- *samplingEventList*: array che memorizza l'insieme di campionamenti fatti durante la simulazione
- *initializedP*: variabile Booleana usata per impostare a 1 il valore dell'attributo  $x$  relativo all'evento degli arrivi di tipo P: infatti, questo tipo di richieste iniziano ad arrivare alle ore 20:00
- metodo *changeSlot()* della classe *Time*: questo metodo serve ad aggiornare la fascia oraria in cui ci si trova. Questo valore è memorizzato in un attributo della stessa classe, chiamato *timeSlot*

- funzione *evaluation(listOfSample)*: funzione che serve a valutare i risultati della simulazione in modo complessivo. Questa viene invocata due volte:
  - la prima volta ha in input una lista di un solo elemento, che è un campione creato a fine simulazione. In questo modo, si fa quindi un'analisi a regime dei dati.
  - la seconda volta, invece, la lista è di più elementi: sono i campioni che vengono raccolti in un tempo intermedio durante la simulazione. Questo tempo è generato con una *Uniform(a, b)* con i valori di input che corrispondono alla fascia oraria in cui sono attivi entrambi i processi degli arrivi. Ciò è stato necessario perché al termine di ogni giornata si c'è sempre un equilibrio tra richieste in entrata e richieste in uscita, mentre potrebbe essere più interessante valutare anche il numero di job in coda.
 Il calcolo dei valori prevede una media su tutte le grandezze di tutti gli elementi della lista rispetto al numero di campioni raccolti.

I risultati di questa valutazione vengono stampati a schermo al termine del programma.

- funzione *FindOne(events, isP = False)*: questa funzione è stata riscritta perché bisogna tenere conto del tipo della richiesta che arriva: non si può infatti assegnare una richiesta di tipo *B* a un servente vuoto di tipo *P*. Per questo motivo, il secondo parametro in input serve a modificare opportunamente l'intervallo di ricerca dei server liberi. La funzione può ritornare un valore negativo: questo si verifica quando il prossimo evento simulato è un evento di arrivo di un qualunque tipo ma tutti i serventi in grado di gestire quella richiesta sono occupati.
- La funzione *GetService()* è stata "sdoppiata" nelle funzioni *GetServiceB()* e *GetServiceP()*. Stesso discorso vale per la funzione *GetArrival()*, divisa in *GetArrivalB()* e *GetArrivalP()*. Per queste ultime, però, c'è anche un'altra modifica: essendo i tempi di interarrivo dipendenti dalle fasce orarie e dal giorno della settimana, è necessario poter identificare la giusta media da utilizzare come parametro di *Exponential(m)*. Per fare questo, viene definita la funzione *getCorrectInterarrival(isP = False)*, che ricerca il giusto tempo di interarrivo basandosi sul *timeSlot* memorizzato nell'istanza della classe *Time* e sul parametro di input che identifica se la richiesta per cui si genera il prossimo tempo di arrivo è di tipo *B* o *P*.

## 5 Verifica e validazione

I risultati ottenuti dall'esecuzione di *simulation.py* con  $m = 2$  e inserendo come seme il valore *123* su un periodo di 365 giorni sono i seguenti:

Per le richieste di tipo B

Grandezza	Tempistica
$\frac{1}{\lambda_{\text{medio}}}$	5.22 min
$E[T_S]$	2.18
$E[N]$	0.42
$E[T_Q]$	0.18
$E[N_Q]$	0.03

Per le richieste di tipo P

Grandezza	Tempistica
$\frac{1}{\lambda_{\text{medio}}}$	3.84 min
$E[T_S]$	4.51
$E[N]$	1.17
$E[T_Q]$	1.51
$E[N_Q]$	0.39

Statistiche dei server B

Numero del server	Utilizzazione	$E[S]$	Share
1	0.190	1.98	0.500
2	0.190	1.98	0.500

Statistiche dei server P

Numero del server	Utilizzazione	$E[S]$	Share
4	0.392	2.99	0.503
5	0.387	2.99	0.497

### 5.1 Tempo di interarrivo medio

Per calcolare la frequenza di arrivo media, consideriamo come arco di tempo di riferimento una settimana.

Nei giorni settimanali, il numero medio di job che arrivano è:

$$E[N] = \frac{1}{6} \cdot 360 + \frac{1}{12} \cdot 300 + \frac{1}{6} \cdot 120 + \frac{1}{4} \cdot 120 + \frac{1}{6} \cdot 240 = 175 \text{ job}$$

Nei giorni fine-settimanali, il numero medio di job che arrivano è:

$$E[N] = \frac{1}{4} \cdot 360 + \frac{1}{6} \cdot 300 + \frac{1}{3} \cdot 120 + \frac{1}{4} \cdot 120 + \frac{1}{2} \cdot 240 = 330 \text{ job}$$

Quindi in media in una settimana arrivano:

$$E[N]_{\text{settimana}} = 175 \cdot 5 + 330 \cdot 2 = 1535 \text{ job}$$

Ora, per trovare la frequenza di arrivo media giornaliera in minuti, dividiamo per  $7 \cdot 19h \cdot 60min$ , ottenendo:

$$\lambda_{\text{medio, min}} = \frac{1535}{7d \cdot 19h \cdot 60min} = 0.19235 \text{ job/min}$$

L'inverso di questo valore è l'interarrivo medio in minuti, ed è:

$$\frac{1}{\lambda_{\text{medio, min}}} = 5.19 \text{ min}$$



## 5.2 Numero medio job in coda e tempo medio in coda

Per calcolare il tempo medio in coda per un multiserver abbiamo bisogno della formula *Erlang-C*. Si definiscono:

$$PQ = \frac{(m \cdot \rho)^m}{m! \cdot (1 - \rho)} \cdot P(0)$$

$$P(0) = \left( \sum_{i=0}^{m-1} \frac{(m \cdot \rho)^i}{i!} + \frac{(m \cdot \rho)^m}{m! \cdot (1 - \rho)} \right)^{-1}$$

$$E(T_Q) = \frac{PQ \cdot E(S)}{1 - \rho}$$

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

$$E[S_i] = \frac{1}{\mu}$$

$$E[S] = \frac{E[S_i]}{m} = \frac{1}{m \cdot \mu}$$

Quindi, ponendo  $\mu = \frac{1}{2}$ , si ha che, per ogni fascia oraria nei giorni settimanali:

- 7 → 13

$$E[S] = \frac{1}{2 \cdot \mu} = \frac{2}{2} = 1 \text{ min}$$

$$E[S_i] = \frac{1}{\mu} = 2 \text{ min}$$

$$\rho_{7 \rightarrow 13} = \frac{\lambda_{7 \rightarrow 13}}{m \cdot \mu} = \frac{\lambda_{7 \rightarrow 13}}{2 \cdot \frac{1}{2}} = \lambda_{7 \rightarrow 13} = 0.17$$

$$P(0)_{7 \rightarrow 13} = \dots = 0.71$$

$$PQ_{7 \rightarrow 13} = \dots = 0.05$$

$$E[T_{Q_{7 \rightarrow 13}}] = \dots = 0.06 \text{ min}$$

$$E[N_{Q_{7 \rightarrow 13}}] = \lambda_{7 \rightarrow 13} \cdot E[T_{Q_{7 \rightarrow 13}}] = 0.01 \text{ job}$$

- 13 → 18

$$\rho_{13 \rightarrow 18} = \frac{\lambda_{13 \rightarrow 18}}{m \cdot \mu} = \frac{\lambda_{13 \rightarrow 18}}{2 \cdot \frac{1}{2}} = \lambda_{13 \rightarrow 18} = 0.17$$

$$P(0)_{13 \rightarrow 18} = \dots = 0.71$$

$$PQ_{13 \rightarrow 18} = \dots = 0.05$$

$$E[T_{Q_{13 \rightarrow 18}}] = .. = 0.06 \text{ min}$$

$$E[N_{Q_{13 \rightarrow 18}}] = \lambda_{13 \rightarrow 18} \cdot E[T_{Q_{13 \rightarrow 18}}] = 0.01 \text{ job}$$

- 18 → 20

$$\rho_{18 \rightarrow 20} = \frac{\lambda_{18 \rightarrow 20}}{m \cdot \mu} = \frac{\lambda_{18 \rightarrow 20}}{2 \cdot \frac{1}{2}} = \lambda_{18 \rightarrow 20} = 0.17$$

$$P(0)_{18 \rightarrow 20} = .. = 0.71$$

$$PQ_{18 \rightarrow 20} = .. = 0.05$$

$$E[T_{Q_{18 \rightarrow 20}}] = .. = 0.06 \text{ min}$$

$$E[N_{Q_{18 \rightarrow 20}}] = \lambda_{18 \rightarrow 20} \cdot E[T_{Q_{18 \rightarrow 20}}] = 0.01 \text{ job}$$

- 20 → 22

$$\rho_{20 \rightarrow 22} = \frac{\lambda_{20 \rightarrow 22}}{m \cdot \mu} = \frac{\lambda_{20 \rightarrow 22}}{2 \cdot \frac{1}{2}} = \lambda_{20 \rightarrow 22} = 0.17$$

$$P(0)_{20 \rightarrow 22} = .. = 0.71$$

$$PQ_{20 \rightarrow 22} = .. = 0.05$$

$$E[T_{Q_{20 \rightarrow 22}}] = .. = 0.06 \text{ min}$$

$$E[N_{Q_{20 \rightarrow 22}}] = \lambda_{20 \rightarrow 22} \cdot E[T_{Q_{20 \rightarrow 22}}] = 0.01 \text{ job}$$

- 22 → 2

$$\rho_{22 \rightarrow 2} = \frac{\lambda_{22 \rightarrow 2}}{m \cdot \mu} = \frac{\lambda_{22 \rightarrow 2}}{2 \cdot \frac{1}{2}} = \lambda_{22 \rightarrow 2} = 0.17$$

$$P(0)_{22 \rightarrow 2} = .. = 0.71$$

$$PQ_{22 \rightarrow 2} = .. = 0.05$$

$$E[T_{Q_{22 \rightarrow 2}}] = .. = 0.06 \text{ min}$$

$$E[N_{Q_{22 \rightarrow 2}}] = \lambda_{22 \rightarrow 2} \cdot E[T_{Q_{22 \rightarrow 2}}] = 0.01 \text{ job}$$

Per calcolare il numero medio di job in coda in settimana, si sommano i risultati ottenuti in ogni fascia oraria pesati per la durata della fascia stessa e infine si divide per la durata del giorno lavorativo:

$$E[N_{Q_{\text{giorno-settimana}}}] = \frac{E[N_{Q_{7 \rightarrow 13}}] \cdot 360 + E[N_{Q_{13 \rightarrow 18}}] \cdot 300}{19 \cdot 60} + \frac{E[N_{Q_{18 \rightarrow 20}}] \cdot 120 + E[N_{Q_{20 \rightarrow 22}}] \cdot 120}{19 \cdot 60} + \frac{E[N_{Q_{22 \rightarrow 2}}] \cdot 240}{19 \cdot 60} = 0.01 \text{ job}$$

Per i giorni fine-settimanali, invece:

- $7 \rightarrow 13$

$$E[S] = \frac{1}{2 \cdot \mu} = \frac{2}{2} = 1 \text{ min}$$

$$E[S_i] = \frac{1}{\mu} = 2 \text{ min}$$

$$\rho_{7 \rightarrow 13} = \frac{\lambda_{7 \rightarrow 13}}{m \cdot \mu} = \frac{\lambda_{7 \rightarrow 13}}{2 \cdot \frac{1}{2}} = \lambda_{7 \rightarrow 13} = 0.25$$

$$P(0)_{7 \rightarrow 13} = .. = 0.60$$

$$PQ_{7 \rightarrow 13} = .. = 0.10$$

$$E[T_{Q_{7 \rightarrow 13}}] = .. = 0.13 \text{ min}$$

$$E[N_{Q_{7 \rightarrow 13}}] = \lambda_{7 \rightarrow 13} \cdot E[T_{Q_{7 \rightarrow 13}}] = 0.03 \text{ job}$$

- $13 \rightarrow 18$

$$\rho_{13 \rightarrow 18} = \frac{\lambda_{13 \rightarrow 18}}{m \cdot \mu} = \frac{\lambda_{13 \rightarrow 18}}{2 \cdot \frac{1}{2}} = \lambda_{13 \rightarrow 18} = 0.17$$

$$P(0)_{13 \rightarrow 18} = .. = 0.71$$

$$PQ_{13 \rightarrow 18} = .. = 0.05$$

$$E[T_{Q_{13 \rightarrow 18}}] = .. = 0.06 \text{ min}$$

$$E[N_{Q_{13 \rightarrow 18}}] = \lambda_{13 \rightarrow 18} \cdot E[T_{Q_{13 \rightarrow 18}}] = 0.01 \text{ job}$$

- $18 \rightarrow 20$

$$\rho_{18 \rightarrow 20} = \frac{\lambda_{18 \rightarrow 20}}{m \cdot \mu} = \frac{\lambda_{18 \rightarrow 20}}{2 \cdot \frac{1}{2}} = \lambda_{18 \rightarrow 20} = 0.33$$

$$P(0)_{18 \rightarrow 20} = .. = 0.50$$

$$PQ_{18 \rightarrow 20} = .. = 0.17$$

$$E[T_{Q_{18 \rightarrow 20}}] = .. = 0.25 \text{ min}$$

$$E[N_{Q_{18 \rightarrow 20}}] = \lambda_{18 \rightarrow 20} \cdot E[T_{Q_{18 \rightarrow 20}}] = 0.08 \text{ job}$$

- $20 \rightarrow 22$

$$\rho_{20 \rightarrow 22} = \frac{\lambda_{20 \rightarrow 22}}{m \cdot \mu} = \frac{\lambda_{20 \rightarrow 22}}{2 \cdot \frac{1}{2}} = \lambda_{20 \rightarrow 22} = 0.25$$

$$P(0)_{20 \rightarrow 22} = .. = 0.60$$

$$PQ_{20 \rightarrow 22} = .. = 0.10$$

$$E[T_{Q_{20 \rightarrow 22}}] = .. = 0.13 \text{ min}$$

$$E[N_{Q_{20 \rightarrow 22}}] = \lambda_{20 \rightarrow 22} \cdot E[T_{Q_{20 \rightarrow 22}}] = 0.03 \text{ job}$$

- $22 \rightarrow 2$

$$\rho_{22 \rightarrow 2} = \frac{\lambda_{22 \rightarrow 2}}{m \cdot \mu} = \frac{\lambda_{22 \rightarrow 2}}{2 \cdot \frac{1}{2}} = \lambda_{22 \rightarrow 2} = 0.50$$

$$P(0)_{22 \rightarrow 2} = .. = 0.33$$

$$PQ_{22 \rightarrow 2} = .. = 0.33$$

$$E[T_{Q_{22 \rightarrow 2}}] = .. = 0.67 \text{ min}$$

$$E[N_{Q_{22 \rightarrow 2}}] = \lambda_{22 \rightarrow 2} \cdot E[T_{Q_{22 \rightarrow 2}}] = 0.33 \text{ job}$$

Per calcolare il numero medio di job in coda in settimana, si sommano i risultati ottenuti in ogni fascia oraria pesati per la durata della fascia stessa e infine si divide per la durata del giorno:

$$E[N_{Q_{\text{giorno-finesettimana}}}] = \frac{E[N_{Q_{7 \rightarrow 13}}] \cdot 360 + E[N_{Q_{13 \rightarrow 18}}] \cdot 300}{10 \cdot 60} + \frac{E[N_{Q_{18 \rightarrow 20}}] \cdot 120 + E[N_{Q_{20 \rightarrow 22}}] \cdot 120}{19 \cdot 60} + \frac{E[N_{Q_{22 \rightarrow 2}}] \cdot 240}{19 \cdot 60} = 0.09 \text{ job}$$

Per calcolare la media, allora, basta fare una media pesata sulla settimana:

$$E[N_Q] = \frac{5 \cdot E[N_{Q_{\text{giorno-settimana}}}] + 2 \cdot E[N_{Q_{\text{giorno-finesettimana}}}]}{7} = 0.03 \text{ job}$$

Per calcolare il tempo medio in coda, allora, si sfrutta la legge di Little e il  $\lambda_{\text{medio}}$  calcolato prima:

$$E[T_Q] = \frac{E[N_Q]}{\lambda_{\text{medio}}} = 0.18 \text{ min}$$

### 5.3 Tempo di risposta e numero medio di job nel nodo

Una volta calcolato il tempo medio in coda, basta sommare  $\frac{1}{\mu}$  per ottenere il tempo medio di risposta:

$$E[T_S] = E[T_Q] + \frac{1}{\mu} = 2.18 \text{ min}$$

Usando Little, ci ricaviamo facilmente il numero di job nel centro:

$$E[N] = E[T_S] \cdot \lambda_{\text{medio}} = 0.41 \text{ job}$$

## 5.4 Utilizzazione

Per calcolare l'utilizzazione in un multiserver, basta fare:

$$\rho = \frac{\lambda_{\text{medio}}}{m \cdot \mu} = 0.19$$

## 6 Conclusioni

Come si nota dai risultati ottenuti, a parte per qualche errore di approssimazione, i risultati della simulazione tendono a quelli teorici.

I risultati dell'analisi mostrano che il numero migliore di baristi da assumere è 2: infatti questo è il numero minimo con cui si riesce a rispettare il vincolo sul tempo di risposta, anche se il guadagno sarebbe stato maggiore con  $m = 1$ . Continuando invece ad aumentare il numero di baristi, il guadagno diminuisce sempre di più, mentre migliorano i tempi di risposta:

$m$	$E[T_s]$	$r(\tau)$
1	5.53 min	3070.10 € al mese
2	2.18 min	1853.43 € al mese
3	2.02 min	636.76 € al mese
4	2.00 min	-579.90 € al mese

Si può notare che, al crescere di  $m$ , la differenza dei tempi con il caso  $m - 1$  è sempre minore: questo si può spiegare considerando che i tassi di arrivo non sono stati cambiati: aumentando  $m$ , quindi, si va a diminuire il tempo di coda di ogni job, che tende quindi a 0.