Progetto PMCSN
Gestione di un parco divertimenti

Anno accademico 2022/2023

Francesco Bernardini

Matricola: 0338264

Indice

1) Introduzione e Obiettivi

Il sistema preso in considerazione ricrea lo scenario che si ha quando si entra dentro Gardaland e si partecipa a una delle attrazioni che il parco offre.
I giochi presi in considerazione sono Jumanji: The Adventure e Fuga da Atlantide. Sono state scelte queste due attrazioni poiché risultano due delle più gettonate e affollate all'interno del parco, in particolar modo la prima.

Gli utenti che entrano nel sistema possono passare per la biglietteria se devono acquistare il biglietto giornaliero del parco o andare direttamente in uno dei due giochi se hanno effettuato l'acquisto del biglietto online.

Una volta entrati nel parco e scelto il gioco, gli utenti si mettono in coda per poter accedere all'attrazione.
Abbiamo due tipi di code, una lenta e una veloce. La coda lenta rappresenta la coda standard dove qualsiasi persona può accedervi. Mentre la coda veloce rappresenta la coda salta fila, un tipo di coda che Gardaland concede agli utenti che pagano un'aggiunta al biglietto di entrata.

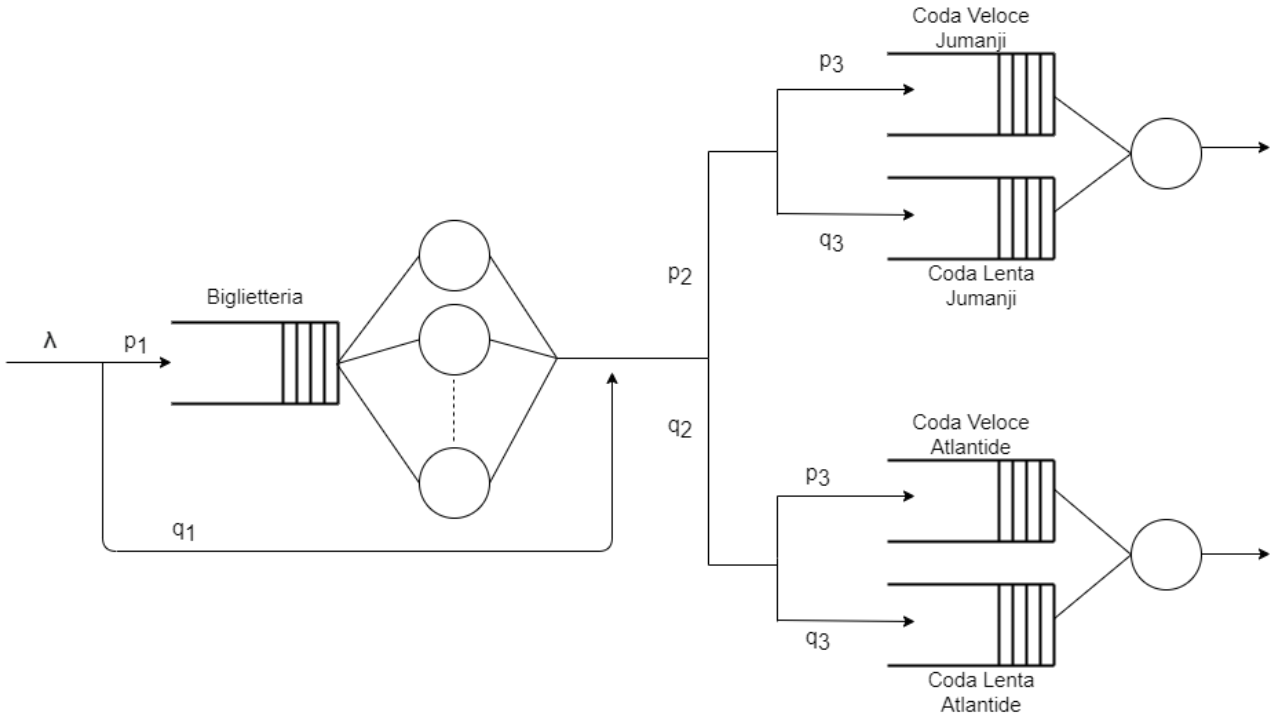
Abbiamo inoltre 2 fasce orarie diverse con cui gli utenti accedono al sistema. Una in cui c'è un afflusso medio e rispecchia le prime ore in cui il parco è aperto e le ultime ore, l'altra fascia rispecchia le ore centrali della giornata in cui l'affluenza di persone è maggiore.
Entrambe le fasce orarie sono di 3 ore, la prima dalle 10 alle 13, la seconda dalle 16 alle 19.

Grazie a siti online, in cui sono raccolti alcuni tempi di attesa dei parchi divertimento, è stato possibile notare e prendere in studio alcune statistiche relative al parco e le sue attrazioni in cui è risultato che, ad esempio, l'attesa media in coda per accedere al gioco di Jumanji era di 46 minuti mentre quella di Atlantide era di 20 minuti.

L'obiettivo è creare un modello il più simile possibile alla realtà mantenendo i seguenti QoS:

- Per la biglietteria, avere un tempo di risposta medio sotto i 20 secondi per entrambe le fasce orarie.
- Per l'attrazione Jumanji: The Adventure, avere un tempo di risposta medio sotto i 46 minuti, per la fascia mattutina, e sotto i 60 minuti per la fascia pomeridiana.
- Per l'attrazione Fuga da Atlantide, avere un tempo di risposta medio sotto i 20 minuti per entrambe le fasce.

2) Modello Concettuale



La figura qui sopra descrive il modello che si vuole creare.
La biglietteria è un multiserver, dal momento che il parco dispone di diverse casse per poter acquistare il biglietto giornaliero.
I due giochi invece sono stati modellati con due code, per dividere chi usufruisce della coda veloce e chi no, e un unico servente dal momento che il gioco è lo stesso per entrambe le file.

2.1) Stati del sistema

Ad ogni istante di tempo si hanno i seguenti stati:

- Numero di persone nel sistema.
- Numero di persone nelle varie code.
- Numero di persone nei vari servizi.
- Stato di un servente, se occupato o meno.

2.2) Eventi

Gli eventi che si hanno nel **sistema** sono:

- Arrivo nel sistema.
- Uscita dal sistema che consiste nell'uscire da uno dei due giochi.

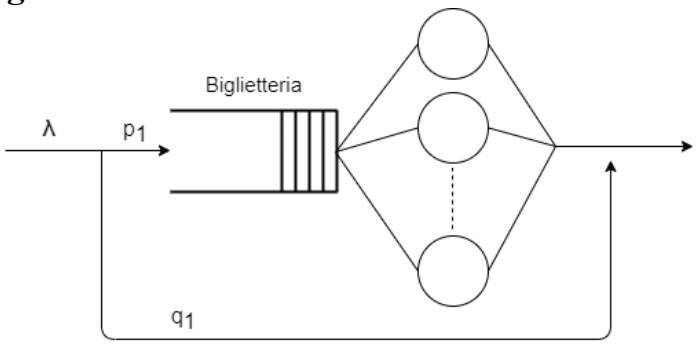
Per la **biglietteria** si ha:

- Arrivo di un job nella coda per acquistare il biglietto.
- Uscita di un job, che consiste nell'acquistato del biglietto.

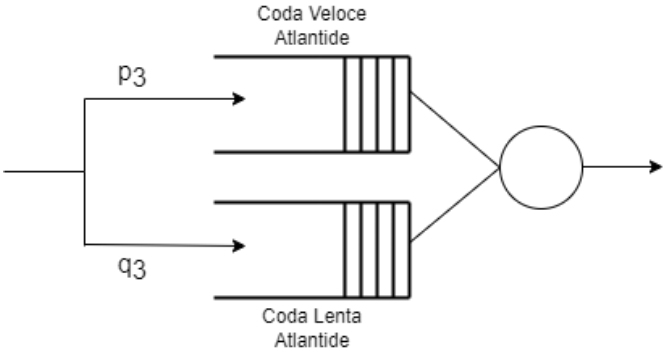
Per **Jumanji: The Adventure** e **Fuga da Atlantide** si ha:

- Arrivo di un job in una delle due code dell'attrazione.
- Uscita di un job, che consiste nell'aver terminato il gioco.

2.3) Descrizione degli eventi



Quando si ha un arrivo in **biglietteria** si controlla se c'è un servente libero, in caso di risposta affermativa l'utente andrà subito in servizio, o sennò dovrà attendere. Per la partenza, l'utente che ha finito lascia il servente e quest'ultimo tornerà disponibile per un altro utente che attendeva.

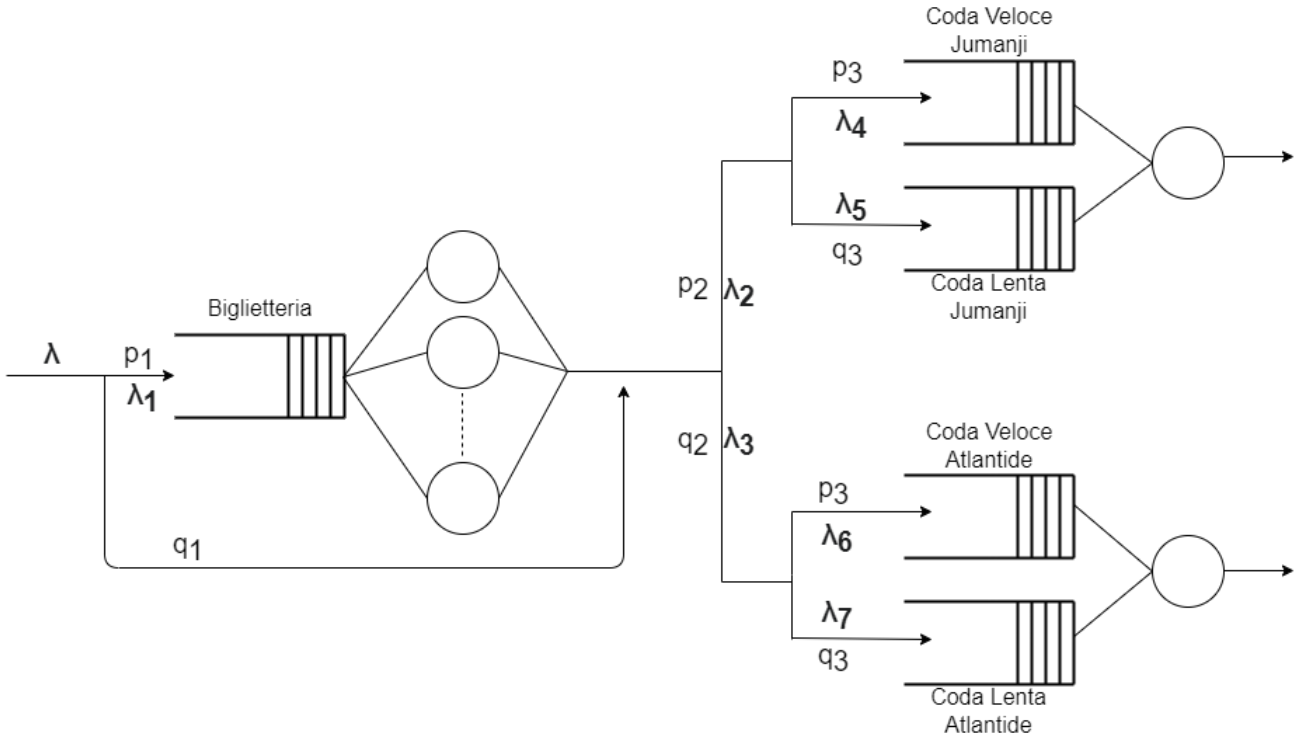


Qui sopra è riportata solo una delle due attrazioni ma il funzionamento è lo stesso per entrambe. Per entrambe le code quando si ha un arrivo, se il servente è libero il job entra subito in servizio e cioè inizia a giocare, altrimenti attenderà. Finito il gioco si uscirà dal nodo e di conseguenza si uscirà dal sistema.

2.4) Assunzioni

- Tutte le percentuali utilizzate nel modello (come quella di chi va in biglietteria) sono state ipotizzate.
- Il servizio qui è il pagamento alla cassa, per quanto riguarda la biglietteria, mentre per le due attrazioni corrisponde allo svolgimento dei giochi stessi.
- L'arrivo di un utente nel parco divertimenti corrisponde ad un nuovo job.

3) Modello Delle Specifiche



$\lambda_1 = \lambda * p_1$
$\lambda_2 = \lambda * p_2$
$\lambda_3 = \lambda * (1-p_2)$
$\lambda_4 = \lambda_2 * p_3$
$\lambda_5 = \lambda_2 * (1-p_3)$
$\lambda_6 = \lambda_3 * p_3$
$\lambda_7 = \lambda_3 * (1-p_3)$

3.1) Modellazione centri

Biglietteria

Questo centro è modellato con un multiserver con coda M/G/N.

- La distribuzione dei tempi di interarrivo è Esponenziale di parametro $\frac{1}{\lambda 1}$.
- La distribuzione di servizio è una Normale troncata, con un tempo medio E[Si] = 15 s. il servizio oscilla tra i 10 s e i 20 s.
- Inoltre, non c'è una possibilità di abbandono poiché il sistema è stato pensato privo di ciò.

Jumanji: The Adventure

Questo centro è modellato con un servente con coda M/G/1.

- La distribuzione dei tempi di interarrivo è Esponenziale di parametro $\frac{1}{\lambda 2}$.
- La distribuzione di servizio è una Normale troncata, con un tempo medio E[Si] = 6,67 s. il servizio oscilla tra i 5 s e i 8,33 s.
- Inoltre, non c'è una possibilità di abbandono poiché il sistema è stato pensato privo di ciò.

Fuga da Atlantide
Questo centro è modellato con un servente con coda M/G/1.

- La distribuzione dei tempi di interarrivo è Esponenziale di parametro $\frac{1}{\lambda 3}$.
- La distribuzione di servizio è una Normale troncata, con un tempo medio E[Si] = 3 s. il servizio oscilla tra i 2,4 s e i 3,6 s.
- Inoltre, non c'è una possibilità di abbandono poiché il sistema è stato pensato privo di ciò.

3.2) Matrice di routing

	Esterno	Biglietteria	Coda Veloce Jumanji	Coda Lenta Jumanji	Coda Veloce Atlantide	Coda Lenta Atlantide
Esterno	0	p_1	$(1-p_1)*p_{jv}$	$(1-p_1)*p_{jl}$	$(1-p_1)*p_{av}$	$(1-p_1)*p_{al}$
Biglietteria	0	0	p_{jv}	p_{jl}	p_{av}	p_{al}
Coda Veloce Jumanji	1	0	0	0	0	0
Coda Lenta Jumanji	1	0	0	0	0	0
Coda Veloce Atlantide	1	0	0	0	0	0
Coda Lenta Atlantide	1	0	0	0	0	0

Con

- $p_1 = 0,65$
- $p_{jv} = p_2 * p_3 = 0,6 * 0,1 = 0,06$
- $p_{jl} = p_2 * (1-p_3) = 0,6 * 0,9 = 0,54$
- $p_{av} = (1-p_2) * p_3 = 0,4 * 0,1 = 0,04$
- $p_{al} = (1-p_2) * (1-p_3) = 0,4 * 0,9 = 0,36$

3.3) Fasce orarie (e scelta probabilità)

Tramite alcune ricerche e articoli su internet e grazie anche a siti, come Gardaland Plus e Queue Times, è stato deciso di assegnare i seguenti tassi di arrivo alle due fasce orarie di tre ore:

- Fascia oraria 10-13 ⟶ $\lambda = 0,51$ job/s.
- Fascia oraria 16-19 ⟶ $\lambda = 0,65$ job/s.

4) Modello Computazionale

Per quanto riguarda il modello computazionale è stato scelto il linguaggio C per la scrittura del codice del simulatore.
Inoltre, per la visualizzazione dei dati ottenuti, è stato deciso di salvare in parte alcuni di essi su file csv e il resto stampati sul terminale da cui si lancia il simulatore.
Tra l'altro il salvataggio dei dati su file, o comunque la stampa su terminale, avviene prelevando queste informazioni da strutture presenti all'interno del codice sorgente.
In particolare, tra le strutture create, segnalo:

4.1) station

```
typedef struct{
  double node;
  double queue;
  double service;
  double index;
  double number;
  double servers;
  double firstArrival;
  double lastArrival;
  double lastService;
}station;
```

Questa struttura **station** mi serve per inserire i valori di ogni nodo che ho nel sistema, quindi la biglietteria e i giochi con le loro code.
I valori di *node*, *queue* e *service* mantengono rispettivamente il numero di jobs presenti nel nodo, in coda e in servizio, tutti e 3 rispettivamente integrati nel tempo.
Il valore di *index* mi dà il numero di utenti che hanno completato il servizio nel nodo, mentre con *number* si ha un aiuto a mantenere il numero attuale di jobs nel nodo.
Grazie a *servers* ho il numero di serventi che il nodo dispone.
Infine, *firstArrival*, *lastArrival* e *lastService* tengono conto rispettivamente dell'istante di tempo in cui arriva il primo job, l'istante di tempo in cui arriva l'ultimo job e l'istante di tempo in cui viene servito l'ultimo job.

4.2) time

```
typedef struct {
  double arrival;
  double completion;
  double current;
  double next;
  double last;
} time;
```

In questa struttura **time** si vanno a memorizzare e ad aggiornare in tempo reale i vari tempi che si hanno nell'esecuzione della simulazione.
Con *arrival* andiamo a scrivere il tempo di arrivo di un job, con *completion* andiamo a segnare il tempo di completamento di questo job, con *current* aggiorniamo il tempo corrente, con *next* andiamo a scrivere il tempo del prossimo evento che avverrà ed infine con *last* si scrive il tempo di arrivo dell'ultimo job.

4.3) outputValues

```
typedef struct {
  int jobs;
  double interarrival;
  double wait;
  double delay;
  double service;
  double Ns;
  double Nq;
  double utilization;
} outputValues;
```

Questa struttura contiene i valori di output, in ordine si ha: numero di jobs, interarrivo medio, attesa media (tempo di risposta), ritardo medio, servizio medio, numero medio di utenti nel nodo, numero medio di utenti in coda e infine l'utilizzazione del nodo.

4.4) event_type

```
typedef struct{
        double time;
        int status;
} event_type;
```

Grazie a questa struttura gestiamo i vari eventi nel sistema. Abbiamo 9 eventi in totale:

- Evento 0, è un arrivo nel sistema.
- Evento 1, è una partenza dalla biglietteria.
- Evento 2, è un arrivo nella coda veloce di Jumanji.
- Evento 3, è un arrivo nella coda lenta di Jumanji.
- Evento 4, è una partenza dal gioco dopo l'accesso nella coda veloce di Jumanji.
- Evento 5, è una partenza dal gioco dopo l'accesso nella coda lenta di Jumanji.
- Evento 6, è un arrivo nella coda veloce di Atlantide.
- Evento 7, è un arrivo nella coda lenta di Atlantide.
- Evento 8, è una partenza dal gioco dopo l'accesso nella coda veloce di Atlantide.
- Evento 9, è una partenza dal gioco dopo l'accesso nella coda lenta di Atlantide.

Con il valore *time* indico il tempo in cui questo evento si sta svolgendo e con *status* indico se questo evento si sta verificando (pongo status pari a 1) o meno (pongo status pari a 0). Ogni partenza da un gioco equivale a un'uscita dal sistema.

4.5) multiserver

```
typedef struct {
   double service;
   int served;
   int occupied;
} multiserver;
```

Questa struttura è stata fatta principalmente per il multiserver della biglietteria. Il valore *service* è lo stesso della struttura station, il valore *served* mi dà il numero di job serviti da un server in particolare e infine *occupied* mi dà lo stato del server, se uguale a 1 è occupato mentre se è uguale a 0 è libero.

4.6) Altri dati

Altro aspetto da considerare è la generazione di numeri pseudocasuali, è stata di aiuto la libreria rngs.h, tramite la quale sono state utilizzate funzioni come PlantSeed(SEED), per scegliere il seme con cui far partire la simulazione, Random(), per la generazione casuale di numeri, GetArrival() per quando si ha un arrivo, Exponential() usata nei servizi e negli arrivi per renderli esponenziali e SelectStream(stream), utile quando si processa il servizio o anche un arrivo nel sistema.
Inoltre, per SelectStream viene specificato un valore stream che cambia a seconda di quale nodo sta chiedendo il servizio. I valori che abbiamo sono:

- 0, quando si processa un arrivo nel sistema.
- 1, quando si processa un servizio di un job in biglietteria.
- 4, quando si processa un servizio di un job che proviene dalla coda veloce di Jumanji.
- 5, quando si processa un servizio di un job che proviene dalla coda lenta di Jumanji.
- 8, quando si processa un servizio di un job che proviene dalla coda veloce di Atlantide.
- 9, quando si processa un servizio di un job che proviene dalla coda lenta di Atlantide.

Mentre PlantSeed ha questo valore SEED che vale 12345 e indica il seme da cui partirà la simulazione.

5) Verifica

Per poter vedere la veridicità dei risultati ottenuti, è stata fatta una simulazione a orizzonte infinito (stazionaria), dal momento che gli arrivi rimangono costanti all'interno di una fascia oraria.
I risultati ottenuti dalle run sono stati confrontati tra loro in modo da vedere se sono coerenti (ad esempio la somma del tempo di attesa e il servizio trovati deve dare lo stesso valore del tempo di risposta trovato).
Inoltre, vengono confrontati anche i valori teorici di λ, μ e ρ con i valori ottenuti dalle simulazioni.
Infine, viene fatto un confronto tra i valori teorici dei tempi di risposta e numero di job nel nodo teorici e ottenuti dal sistema, ovviamente questi confronti non devono essere uguali poiché i valori teorici sono calcolati con un servizio Esponenziale mentre nel simulatore il servizio è una Normale troncata, ma era interessante mostrare la differenza tra i valori con queste due distribuzioni.

5.1) Consistenza centri

Per la biglietteria l'E(S) corrisponde al singolo centro.

	$E(T_S)$	$E(T_Q)$	$E(S)$
Biglietteria	19.35 ± 0.26	4.54 ± 0.25	14.82 ± 0.06
Veloce Jumanji	7.47 ± 0.03	0.8 ± 0.03	6.66 ± 0.01
Lenta Jumanji	2378.42 ± 24.12	2371.75 ± 24.12	6.67 ± 0
Veloce Atlantide	3.09 ± 0.01	0.09 ± 0.01	3 ± 0.01
Lenta Atlantide	4.58 ± 0.03	1.58 ± 0.03	3 ± 0

	$E(N)$	$E(N_Q)$	ρ
Biglietteria	6.42 ± 0.11	1.51 ± 0.09	4.92 ± 0.06
Veloce Jumanji	0.23 ± 0	0.03 ± 0	0.21 ± 0
Lenta Jumanji	356.7 ± 3.6	355.7 ± 3.7	1 ± 0
Veloce Atlantide	0.06 ± 0	0 ± 0	0.06 ± 0
Lenta Atlantide	0.84 ± 0.01	0.29 ± 0.01	0.55 ± 0

Andando a confrontare $E(T_S)$ con $E(T_Q) + E(S)$ e anche $E(N)$ con $E(N_Q) + \rho$, possiamo vedere come i valori sono molti simili tra loro.

5.2) Confronto con λ, μ, ρ teorici

Biglietteria

	Teorico	Simulazione
λ	0,33	0,33
μ	0,066	0,067
ρ	0,825	0,82±0,01

Coda Veloce Jumanji

	Teorico	Simulazione
λ	0,0306	0,031
μ	0,15	0,15
ρ	0,204	0,21

Coda Lenta Jumanji

	Teorico	Simulazione
λ	0,2754	0,15
μ	0,15	0,15
ρ	1,83	1

Coda Veloce Atlantide

	Teorico	Simulazione
λ	0,0204	0,0205
μ	0,33	0,33
ρ	0,0618	0,06

Coda Lenta Atlantide

	Teorico	Simulazione
λ	0,1836	0,184
μ	0,33	0,33
ρ	0,55	0,55

Tutti i valori confrontati sono simili tra loro, tranne nella coda lenta di Jumanji, dove il valore di λ e l'utilizzazione della simulazione sono più bassi del teorico. Questa differenza sarà data dal fatto che questa coda è un collo di bottiglia, dal momento che nessun altro nodo presenta una situazione del genere. Vedremo anche più avanti i problemi di questa coda.

5.3) Confronto con teorici E(T_S) e E(N)

Sono stati riportati i valori di 3 nodi per mostrare questo confronto.

	E(T_S) ottenuto	E(T_S) teorico
Biglietteria	19.35 ± 0.26	17,53
Veloce Jumanji	7.47 ± 0.03	8,38
Lenta Atlantide	4.58 ± 0.03	6,67

	E(N) ottenuto	E(N) teorico
Biglietteria	6.42 ± 0.11	5,81
Veloce Jumanji	0.23 ± 0	0,25
Lenta Atlantide	0.84 ± 0.01	1,22

Possiamo notare come per la biglietteria si hanno dei valori leggermente più alti rispetto al teorico, mentre nelle due code prese in considerazione, i valori ottenuti sono più bassi rispetto al teorico.

6) Validazione

Per avere una validazione è stato visto l'andamento del simulatore a seconda del cambiamento dei valori di arrivo e numero di serventi in biglietteria.
Ad esempio, se diminuisco l'arrivo mi aspetterò meno attesa in generale nelle code, così come se io dovessi aumentare il numero di serventi in biglietteria avrei meno attese.
In particolare, vediamo come cambiano i tempi di risposta di ogni nodo variando il valore di lambda.
Per entrambe le fasce sono state utilizzate le configurazioni minime.

Fascia mattutina: (con 6 serventi in biglietteria)

Lambda	Biglietteria	Coda Veloce Jumanji	Coda Lenta Jumanji	Coda Veloce Atlantide	Coda Lenta Atlantide
0.51 job/s	19,35 s ± 0,26 s	7,47 s ± 0,03 s	2378,42 s ± 24,12 s (≈39,64 min±0,4 min)	3,09 s ± 0,01 s	4,58 s ± 0,03 s
0,65 job/s	108,49 s ± 6,87s	7,64 s ± 0,04 s	3430,38 s ± 27,63 s (≈57,17 min±0,46 min)	3,11 s ± 0,01 s	5,46 s ± 0,06 s
0,4 job/s	15,72 s ± 0,08 s	7,26 s ± 0,04 s	1300,07 s ± 22,61 s (≈21,67 min ±0,38 min)	3,07 s ± 0,01 s	4,06 s ± 0,02 s

Fascia pomeridiana: (con 7 serventi in biglietteria)

Lambda	Biglietteria	Coda Veloce Jumanji	Coda Lenta Jumanji	Coda Veloce Atlantide	Coda Lenta Atlantide
0.65 job/s	23,38 s ± 0,74 s	7,77 s ± 0,04 s	3625,89 s ± 30,69 s (≈60,43 min±0,51 min)	3,11 s ± 0,01 s	5,97 s ± 0,08 s
0,51 job/s	15,96 s ± 0,09 s	7,52 s ± 0,04 s	2385,06 s ± 23,89 s (≈39,75 min±0,4 min)	3,09 s ± 0,01 s	4,69 s ± 0,03 s
0,76 job/s	99,21 s ± 5,47 s	7,93 s ± 0,05 s	4319,76 s ± 29,95 s (≈72 min±0,5 min)	3,13 s ± 0,01 s	7,38 s ± 0,18 s

7) Progettazione degli esperimenti

Per mantenere la stazionarietà in biglietteria, conoscendo i valori di λ e la percentuale di arrivo in questo nodo, i serventi minimi sono stati ricavati con la formula $\frac{\lambda}{m*\mu} < 1$, e si ottiene:

Fascia oraria più affollata (pomeridiana):

- $\frac{0,42}{m*0,067}$, m > 6,27 di conseguenza m = 7.

Fascia oraria meno affollata (mattutina):

- $\frac{0,34}{m*0,067}$, m > 5,07 di conseguenza m = 6.

Questi sono i valori minimi di serventi che servono e ora vediamo nello specifico le simulazioni effettuate.
Il numero di serventi nella simulazione è fisso e per cambiarlo bisogna manualmente modificare nel codice il valore della variabile "serversNum" nel file "values.h".

7.1) Simulazione a orizzonte infinito

In questo tipo di simulazione il sistema viene ricreato per un tempo infinito.
Grazie a questo possiamo vedere allo stazionario il tempo di risposta dei centri e tra l'altro, poiché la simulazione è stazionaria, si toglie il problema del bias dello stato iniziale.

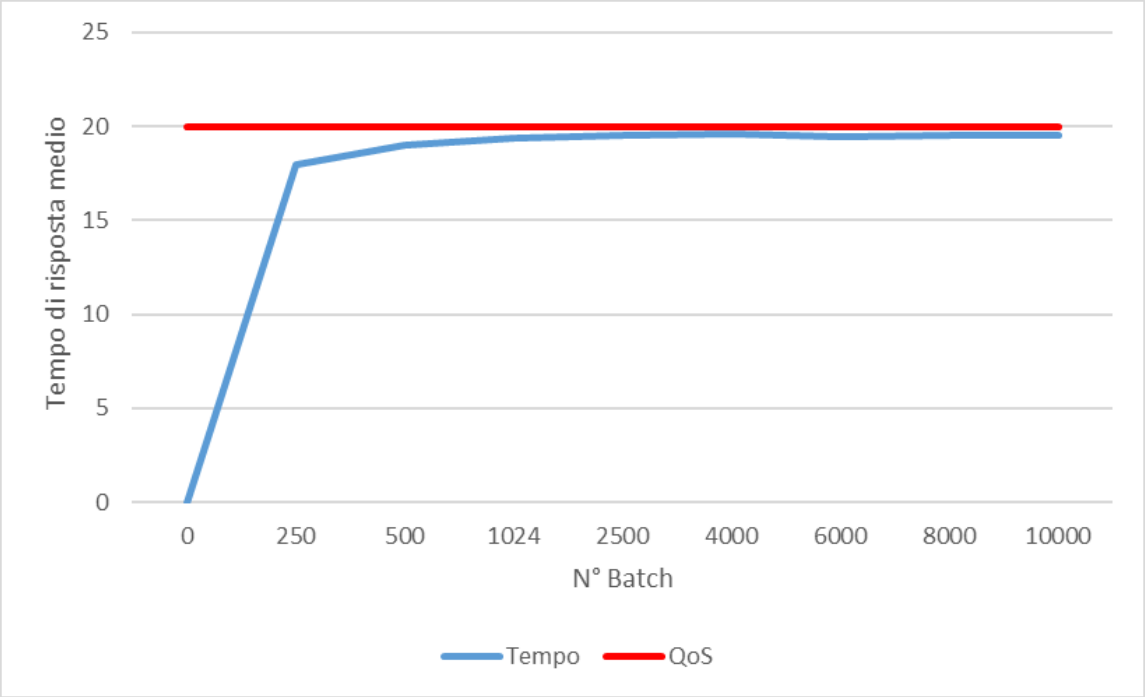
Per poter effettuare questa simulazione è stato utilizzato il metodo delle Batch Means.
Questo consiste in dividere la run in K parti e ognuna di queste ha una grandezza pari a B (la grandezza B consiste nel numero di job presenti).
Ognuna delle K parti termina solamente quando ognuno dei nodi serve esattamente B job.
I valori scelti sono B=1024 e K=128.
Inoltre, non si ha problema di autocorrelazione poiché ognuna di queste K run è indipendente l'una dall'altra. L'unica cosa che hanno in comune due run è che lo stato finale della precedente coincide con lo stato iniziale della successiva.

Qui di seguito ci sono vari grafici che mostrano il variare dei tempi di risposta della biglietteria e delle code lente al variare del valore B con K fisso a 128.
Viene mostrato l'andamento all'aumentare delle Batch poiché così facendo si riescono avere tempi di esecuzioni maggiori e di conseguenza si riesce a vedere il comportamento dei centri all'aumentare del tempo.
Sono state mostrate le statistiche di entrambe le fasce orarie e si vedrà anche se i QoS sono stati rispettati.
Nelle fasce pomeridiane vengono mostrati, su ogni grafico, i valori con 6 e 7 serventi. Questi server corrispondono a quelli della biglietteria e nei grafici degli altri centri sono state mostrate entrambe le configurazioni per vedere come questo influenza il nodo considerato.

Biglietteria

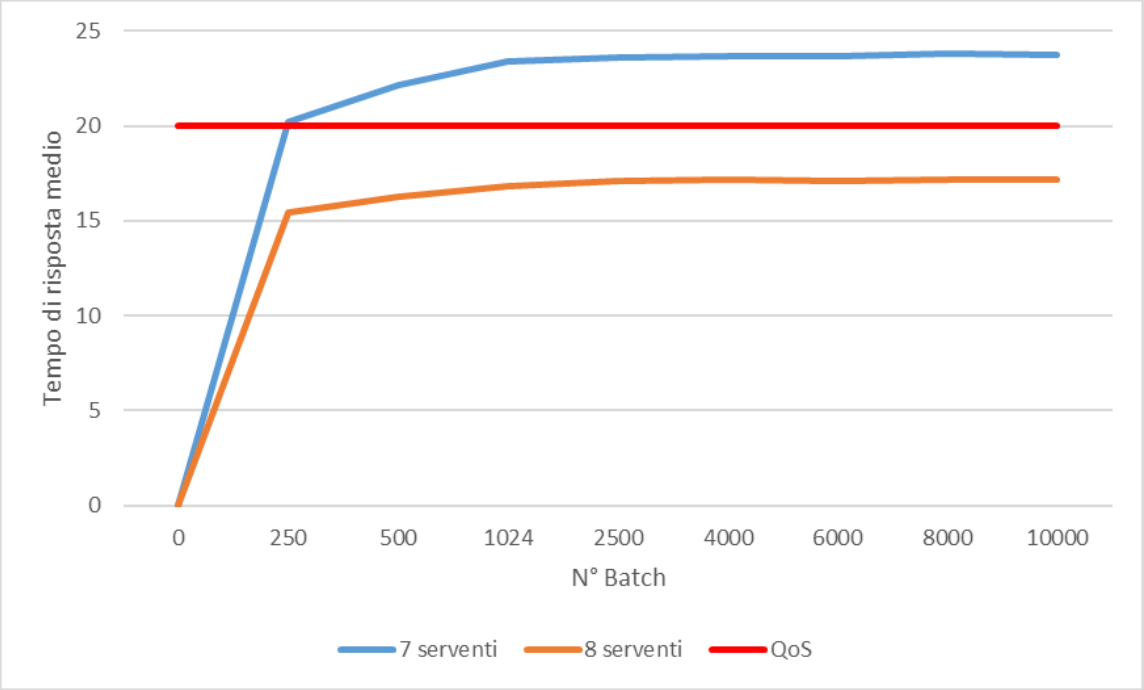
Fascia mattutina

Nella fascia meno affollata si ha un tempo di risposta sempre sotto i 20 s. Il QoS è quindi rispettato ed inoltre il tempo di risposta rimane intorno ai 19 secondi una volta raggiunti.



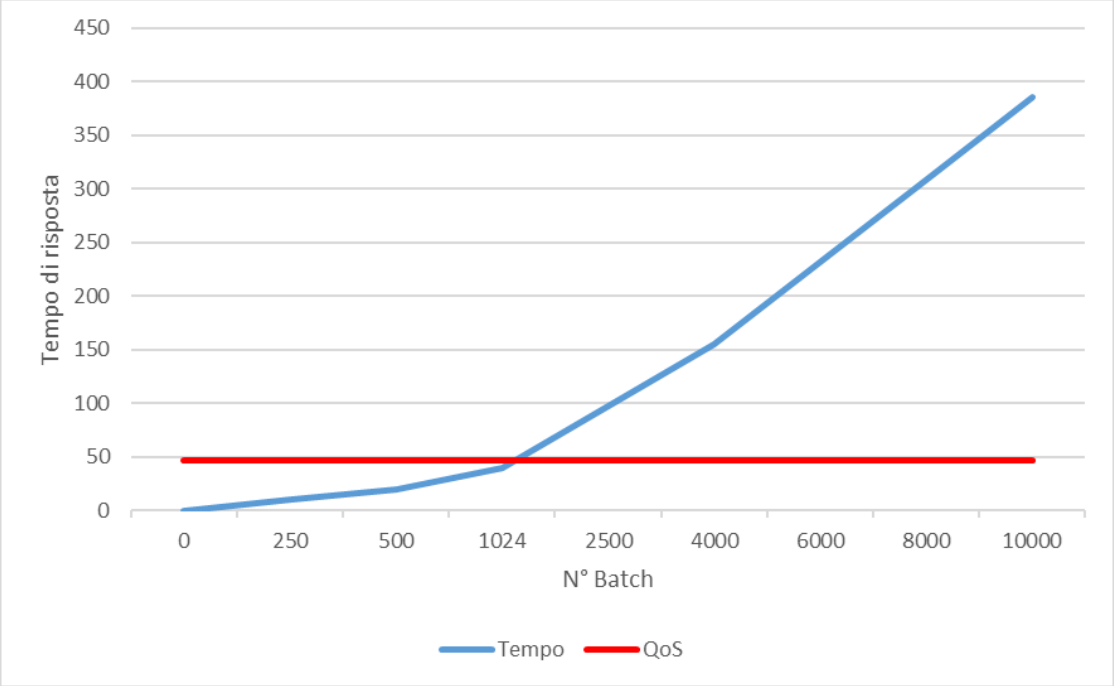
Fascia pomeridiana

Nella fascia più affollata si può vedere come il tempo di risposta medio non rispetti il QoS con la configurazione minima. Per questo è stato aggiunto un ulteriore servente in biglietteria e questo ha portato risultati nettamente migliori come si può vedere dal grafico.

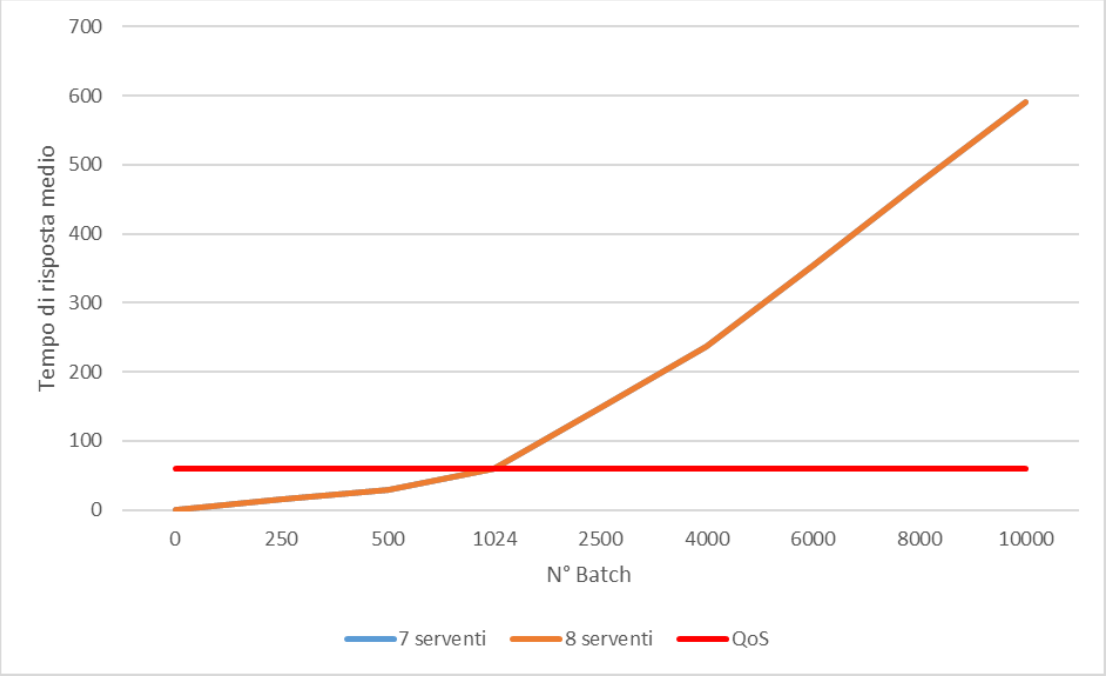


Coda Lenta Jumanji

Fascia mattutina
Questa coda purtroppo non può essere modificata ed essendo un collo di bottiglia solamente all'inizio si riesce a rispettare il QoS, ma andando avanti nel tempo l'attesa aumenta sempre di più.

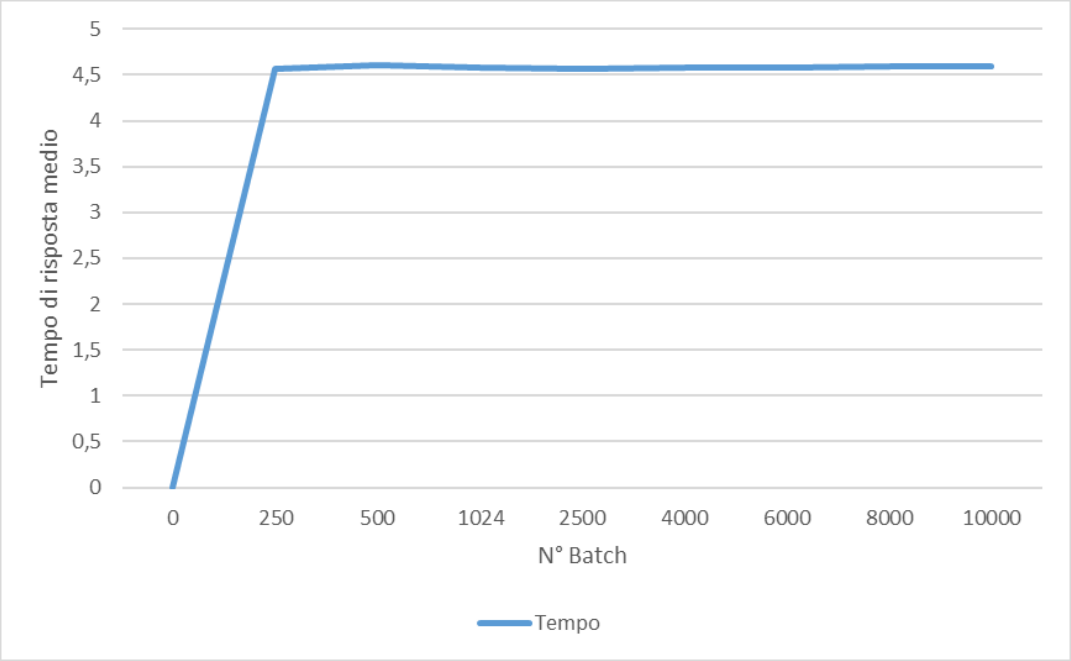


Fascia pomeridiana
Anche in questa fascia si ha una situazione analoga alla precedente. (Qui sono presenti entrambe le curve delle due configurazioni, ma dal momento che i valori sono molto simili e data anche la grande crescita dei tempi di risposta, sembrerebbe esserci una curva unica)

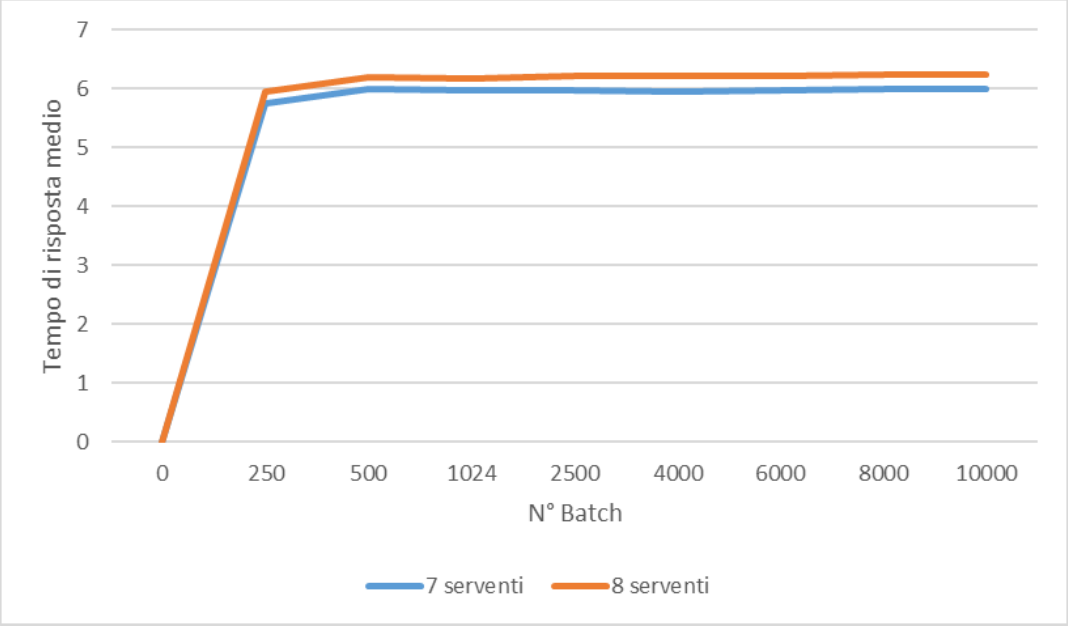


Coda Lenta Atlantide

Fascia mattutina
Con questa fascia si ha un tempo di risposta medio poco sopra i 4,5 secondi. Il QoS è rispettato, non è stato mostrato nel grafico poiché è di 20 minuti e perciò se entrambi i valori fossero stati aggiunti il grafico che si otteneva avrebbe fatto apparire il tempo di risposta quasi a 0.
Questo distacco tra tempo di risposta ottenuto e il QoS è dato dal fatto che l'attrazione non sfrutta tutta la capacità che ha a disposizione, cosa che invece è stata fatta nella simulazione.



Fascia pomeridiana
Anche in questa fascia si ha una situazione analoga a sopra.

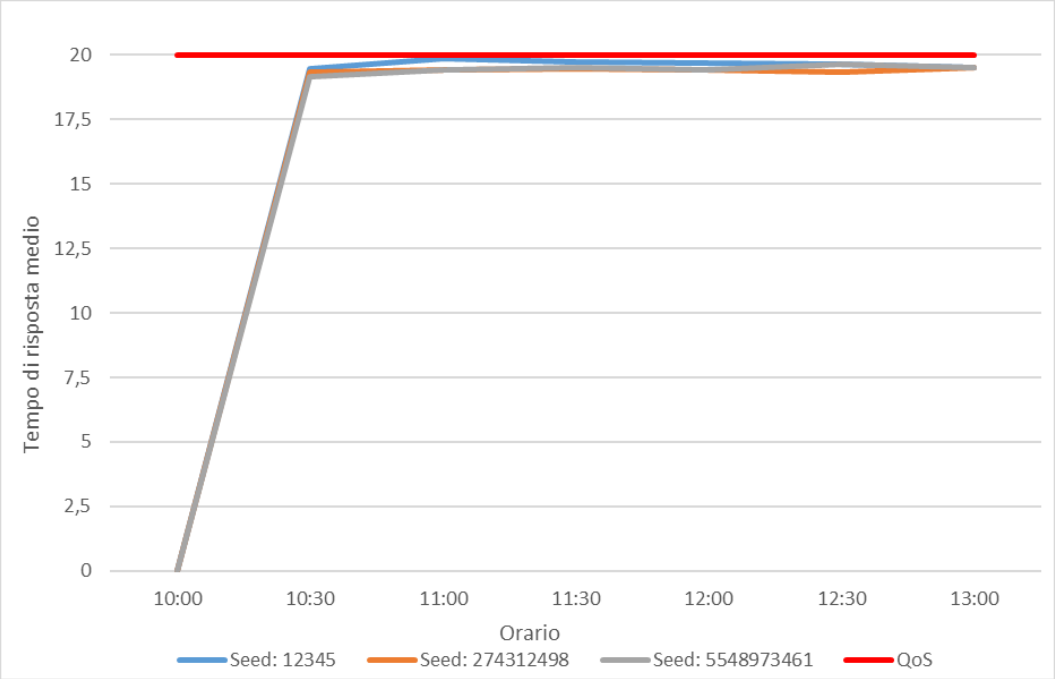


7.2) Simulazione a orizzonte finito

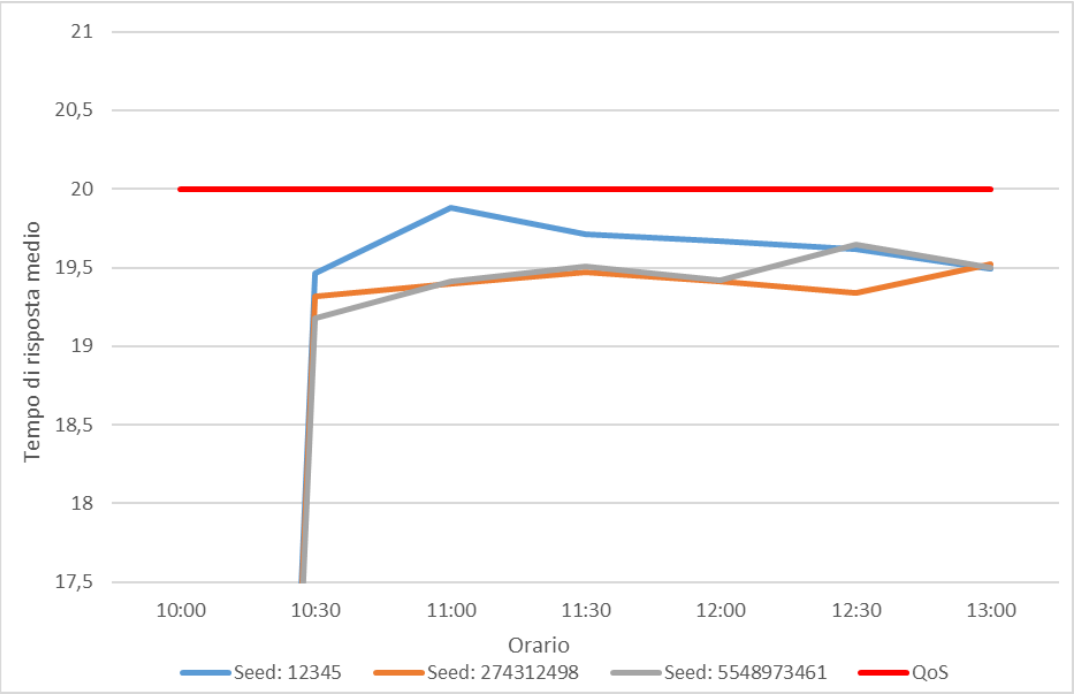
In questa simulazione transiente è stato visto il comportamento del simulatore all'interno della fascia oraria.
È stato usato il metodo delle Replicazioni, con un numero di repliche pari a 128.
Per garantire l'indipendenza tra una replica e l'altra, il seme iniziale è stato scelto prima di chiamare il ciclo delle repliche, così da non avere sovrapposizioni tra di loro, tramite la funzione PlantSeeds(). Inoltre, per calcolare la media campionaria e l'intervallo di confidenza al 95%, è stato utilizzato l'algoritmo di Welford (one-pass).
La simulazione ha una durata massima di 3 ore, che corrisponde alla durata di una fascia. Tra le due fasce si è deciso di mostrarne solo una ed è stata scelta la fascia mattutina. Delle code veloci è stata mostrata solo quella di Jumanji, poiché il comportamento delle due code veloci è molto simile tra loro.

Biglietteria

Dopo mezz'ora si raggiunge uno stato stazionario e rimane intorno ai 19 secondi per tutta la durata della simulazione.

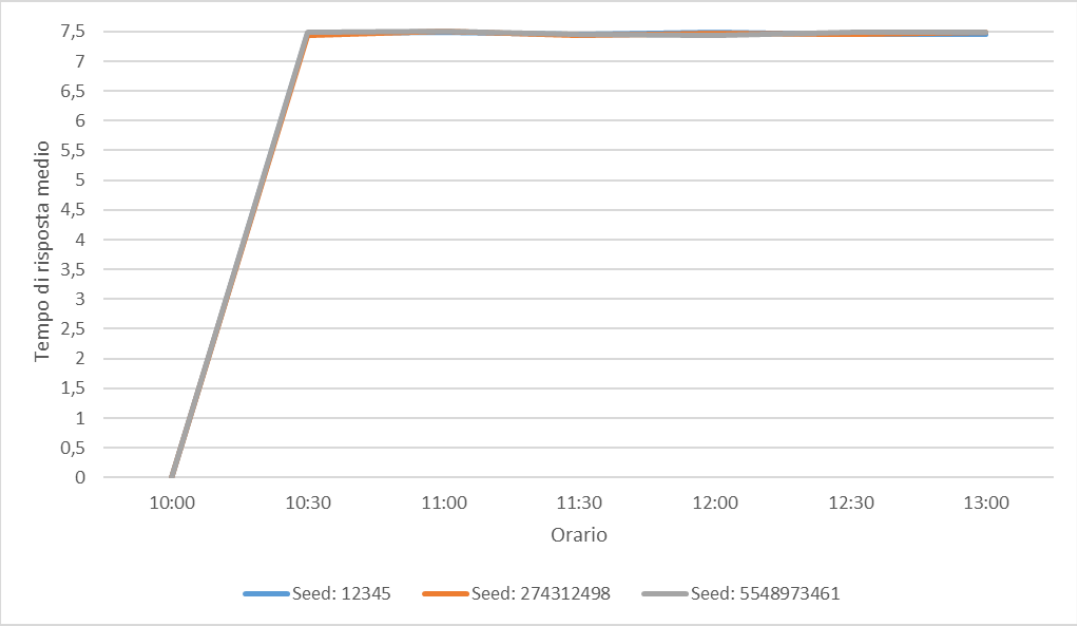


Qui sotto si ha uno zoom dei tre andamenti.

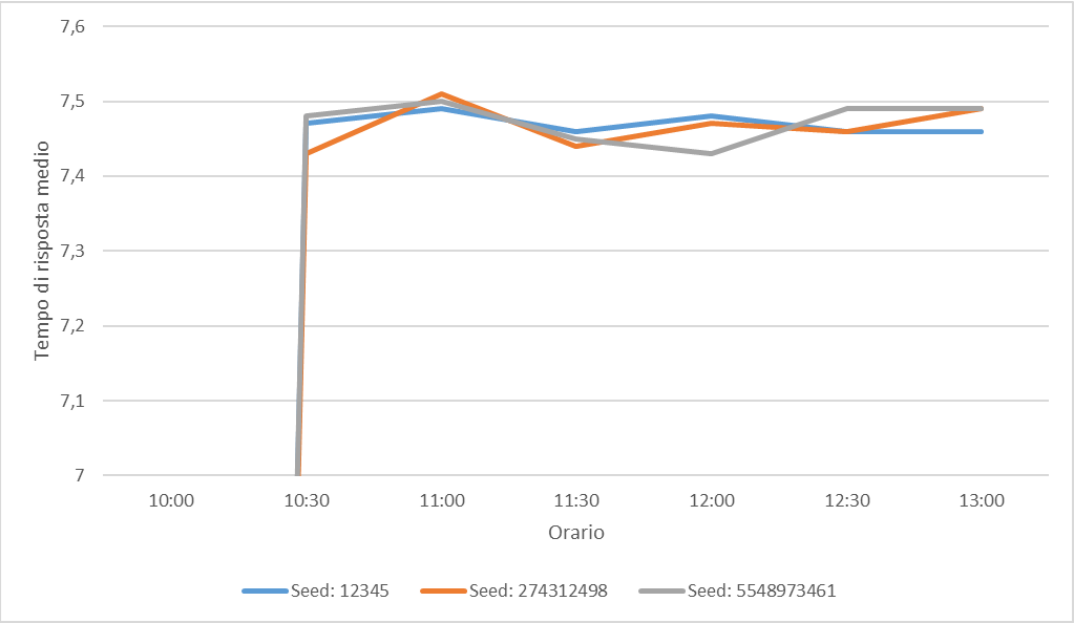


Coda Veloce Jumanji

Anche in questa coda dopo mezz'ora si raggiunge uno stato stazionario intorno ai 7,5 secondi.

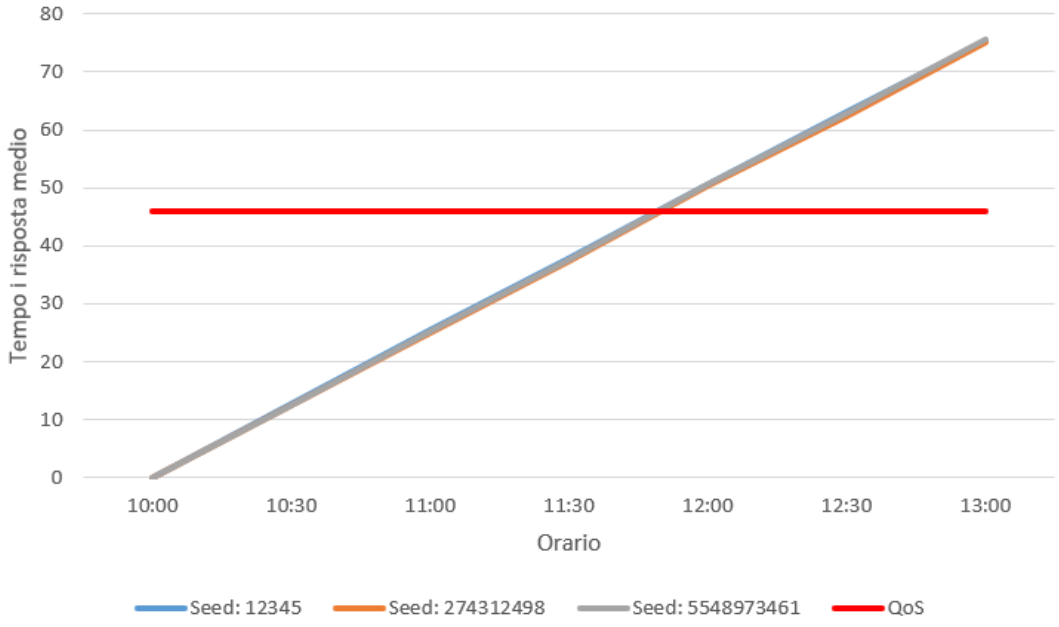


Anche qui è presente un piccolo zoom sugli andamenti.



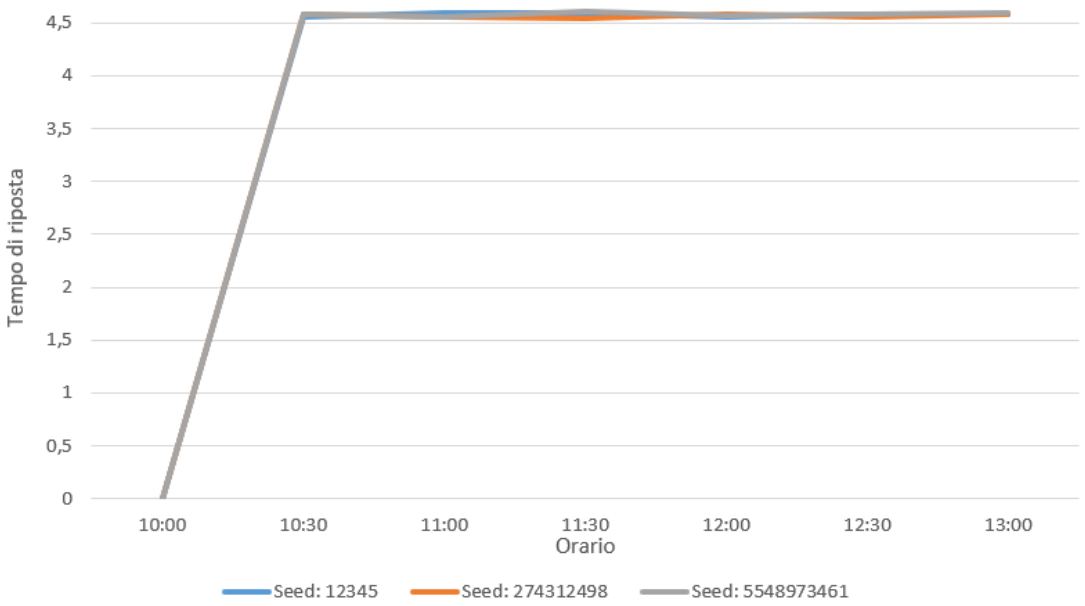
Coda Lenta Jumanji

Come è possibile vedere, anche nella fascia oraria di 3 ore non è possibile rispettare il QoS. Si riesce ad avere un tempo di risposta medio sotto il QoS fino a poco prima delle due ore, ma da lì in poi il tempo di risposta medio continua a crescere.

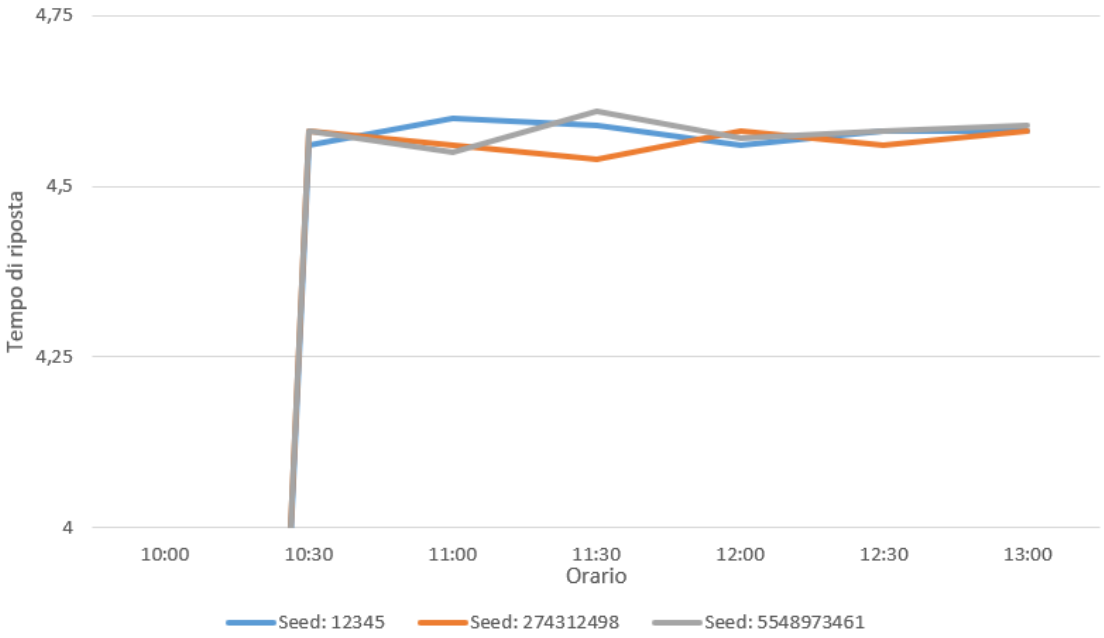


Coda Lenta Atlantide

Infine, anche per questa coda dopo mezz'ora si ha un valore poco sopra i 4,5 secondi.



Qui di seguito troviamo uno zoom per questa coda.



8) Esecuzione simulazione

8.1) Analisi QoS 1

In questo QoS l'obiettivo era avere, in entrambe le fasce, un tempo di risposta in biglietteria sotto i 20 secondi.
Come è stato possibile vedere dalle simulazioni il goal è stato raggiunto da tutte e due le fasce, con la differenza che nella prima fascia è basta la configurazione minima, mentre nella seconda fascia, quella più affollata, è stato necessario aggiungere un servente in più per soddisfare il QoS.

8.2) Analisi QoS 2

Qui l'obiettivo riguardava l'attrazione di Jumanji, in particolare bisognava avere nella coda lenta:

- Per la fascia mattutina, un tempo di risposta sotto i 46 minuti.
- Per la fascia pomeridiana, un tempo di risposta sotto i 60 minuti.

Come è stato possibile vedere, questo QoS non viene rispettato andando avanti nel tempo. Inizialmente è possibile avere un tempo di risposta medio inferiore agli obiettivi prefissati, ma dal momento che questa coda è un collo di bottiglia inevitabile, con l'andare avanti nel tempo e di conseguenza l'aumento del numero di job presenti, l'attesa diventa sempre più grande.

8.3) Analisi QoS 3

In questo terzo QoS l'obiettivo riguardava l'attrazione Fuga da Atlantide, in particolare bisognava avere nella coda lenta un tempo di risposta medio al di sotto dei 20 minuti per entrambe le fasce.
Come è stato possibile vedere dalle simulazioni, questo QoS è stato rispettato poiché i tempi di risposta sono molto bassi rispetto al goal. Questo, come già detto, è dovuto dal fatto che nella realtà il parco divertimenti non sfrutta a pieno la capacità dell'attrazione.

9)Conclusioni

L’analisi effettuata ha cercato di ricreare il più possibile il sistema reale.
In biglietteria si riescono ad avere tempi efficienti già con la configurazione minima per la fascia mattutina, mentre per la fascia pomeridiana basta aggiungere una sola cassa alla configurazione minima.
È emerso anche qui come l’attesa della coda lenta di Jumanji: The Adventure sia un problema ai fini dell’accesso all’attrazione, il divario che ha questa coda con quella veloce è smisurato. Purtroppo, però per come l’attrazione è stata ideata l’unica soluzione sarebbe modificare la capacità di persone che accedono contemporaneamente in servizio, così da ottenere un E(S) più basso e magari ridurre di più queste lunghe attese.
Si ha una situazione opposta nell’attrazione Fuga da Atlantide. Qui i tempi di risposta sono molto più bassi rispetto all’attesa che si ha mediamente nella realtà. Grazie a testimonianze che hanno partecipato a questo gioco e anche tramite video trovati online si ha una spiegazione di questo divario che consiste, come già detto, nel fatto che il parco non sfrutta la capacità massima dell’attrazione.