

Progetto PMCSN
Gestione di un parco divertimenti
Anno accademico 2022/2023

Francesco Bernardini
Matricola: 0338264

Indice

1) Introduzione e Obiettivi	2
2) Modello Concettuale	3
2.1) Stati del sistema	3
2.2) Eventi.....	3
2.3) Assunzioni.....	4
3) Modello Delle Specifiche.....	5
4) Modello Computazionale.....	6
4.1) station.....	6
4.2) time	6
4.3) outputValues.....	7
4.4) event_type.....	7
4.5) multiserver	8
4.6) Altri dati.....	8
5) Verifica	9
5.1) Biglietteria	9
5.2) Coda Veloce Jumanji	10
5.3) Coda Lenta Jumanji	11
5.4) Coda Veloce Atlantide	12
5.5) Coda Lenta Atlantide	13
6) Validazione	14
7) Progettazione degli esperimenti.....	15
7.1) Simulazione a orizzonte infinito.....	15
7.2) Simulazione a orizzonte finito.....	20
8) Esecuzione simulazione	24
8.1) Analisi QoS.....	24
8.2) Analisi collo di bottiglia.....	24
9) Conclusioni	25

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.

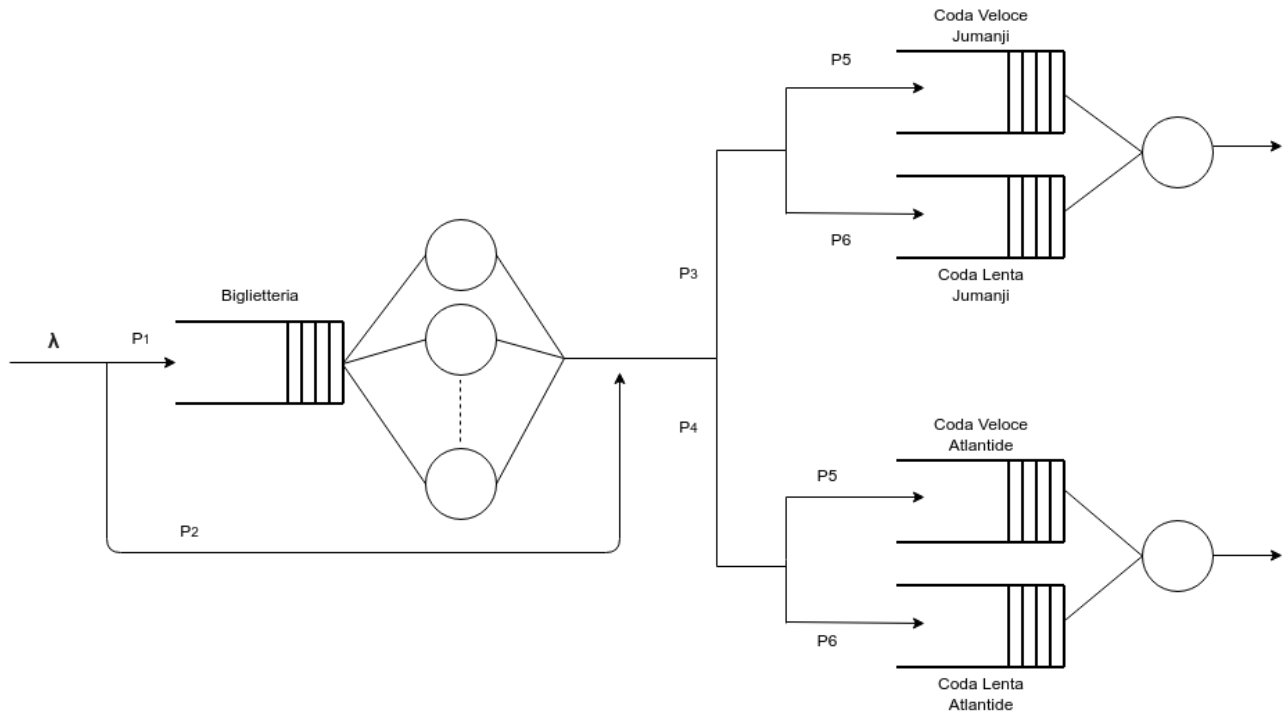
Grazie a siti online, in cui sono raccolti alcuni tempi di attesa dei parchi divertimento, è stato possibile notare e prendere in studio alcune statistiche relative al parco e le sue attrazioni in cui è risultato che, ad esempio, l'attesa media solo per la coda per accedere al gioco di Jumanji era di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

L'obiettivo è creare un modello il più simile possibile alla realtà, minimizzando il numero di serventi in biglietteria così da portare un risparmio di costi di mantenimento al parco.

Inoltre, si vuole mantenere il seguente QoS:

- Per la fascia oraria con affluenza maggiore, avere un tempo totale di risposta, per un utente che passa in biglietteria e va nella coda lenta di Jumanji, minore dei 46 minuti di attesa media che ha l'attrazione. Mentre per gli utenti che passano per la biglietteria e vanno nella coda lenta di Atlantide un tempo di risposta minore di 20 minuti.

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 server 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 server, se occupato o meno.

2.2) Eventi

Gli eventi che si hanno nel sistema sono sostanzialmente di due tipi, arrivi e partenze.

Si hanno arrivi e partenze per ogni centro, in particolare si ha che:

- Per la biglietteria, quando si ha un arrivo si controlla se c'è un server 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 server e quest'ultimo tornerà disponibile per un altro utente che attendeva.
- Per la coda veloce di entrambi i giochi, quando si ha un arrivo se il server è libero entra subito in servizio e cioè entra nel gioco, altrimenti attenderà in coda. Alla fine del servizio ci sarà la partenza e di conseguenza l'uscita dal sistema.
- Per la coda lenta di entrambi i giochi, si ha lo stesso funzionamento di quella veloce.

2.3) Assunzioni

- La coda della biglietteria è una coda M/M/K, con capacità infinita, con politica di scheduling FIFO non preemptive.
- Le code dei giochi sono entrambe M/M/1, con capacità infinita, politica di scheduling FIFO e non preemptive.
- Non conoscendo i costi di mantenimento o attivazione delle casse della biglietteria non possiamo definire una cifra di risparmio. Per questo non verrà citata in seguito una somma di denaro risparmiata ma ci limiteremo a dire solamente il numero di serventi minimi.
- 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

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 mattutina. Mediamente sono stati calcolati intorno ai 7150 arrivi nelle 3 ore ed è stato ricavato un valore di $\lambda = 0,66$ job/s.
- Fascia oraria pomeridiana. Qui il valore degli arrivi aumenta, sono intorno ai 9000 nelle 3 ore e si ricava un valore di $\lambda = 0,83$ job/s.

I valori di tassi di servizio sono fissi e valgono:

- Il servizio E(S) dell'attrazione Jumanji: The Adventure è di 8,3 secondi, questo valore è stato ottenuto poiché il gioco ha una durata circa di 10 minuti e in questo lasso di tempo entrano in servizio 72 persone e per questo sono state divise le 72 persone nei 10 minuti così che all'interno della simulazione quando passeranno 10 minuti saranno state serviti almeno 72 job, così da ottenere gli stessi utenti serviti nella realtà.
- Il servizio E(S) dell'attrazione Fuga da Atlantide è di 3,6 secondi, questo valore è ottenuto poiché la durata del gioco è di circa 12 minuti e in questo tempo partecipano circa 200 persone. Così facendo nella simulazione dopo che passeranno 12 minuti saranno servite 200 persone così da rispecchiare la realtà.
- Il servizio E(S) della biglietteria è di 15 secondi, questo valore è stato scelto come approssimazione di un ipotetico tempo che si spende in una biglietteria per pagare e prendere il biglietto.

Le percentuali sono state scelte approssimativamente nel seguente modo:

- Accessi in biglietteria = 65%. I pagamenti online stanno aumentando sempre di più e per questo si è deciso di far sì che poco più della metà degli utenti pagassero l'entrata fisicamente nel parco.
- Accesso nell'attrazione Jumanji = 60%. Il seguente gioco è quello più gettonato nel parco, per questo si è deciso di dare una maggiore affluenza in questa attrazione piuttosto che nell'altra.
- Accesso nell'attrazione Fuga da Atlantide = 40%. Il resto della clientela che non accede a Jumanji parteciperà a questo gioco.
- Accesso alla coda veloce = 10%. Il servizio di salta fila è molto utile ma allo stesso tempo molto meno utilizzato rispetto alla coda normale. Questo è stato possibile dirlo grazie a testimonianze di persone da me conosciute che hanno confermato che una piccola percentuale di persone utilizza il servizio, creando un enorme divario tra chi è in possesso del salta fila e chi no. Per questo si è deciso di dare il 10% dei job alla coda veloce, così da marcare questa grande differenza. Inoltre, la percentuale di accesso degli utenti nelle due code è uguale in tutte e due le attrazioni poiché il salta fila non è rivolto a un singolo gioco ma a un sottogruppo, di conseguenza una persona che può accedere alla coda veloce di Jumanji, può farlo anche per Atlantide.
- Accesso alla coda lenta = 90%. Qui sarà presente la clientela che non dispone di salta fila.

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 in biglietteria.
- 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).

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).

Questa simulazione svolta riguarda la fascia oraria mattutina ed inoltre sono stati calcolati alcuni valori teorici, del tempo di risposta e di jobs nel nodo, che dovremmo ottenere con l'arrivo, il servizio e le percentuali che sono state indicate.

Di seguito qui ci sono i vari risultati stampati a schermo.

5.1) Biglietteria

Valori Biglietteria			

Interarrival		2.43 +/-	0.03
Wait		24.06 +/-	2.58
Delay		10.11 +/-	2.44
Service		2.01 +/-	0.06
Num Node		9.87 +/-	1.05
Num Queue		4.15 +/-	1.00
Utilization		0.82 +/-	0.02

Da questi valori possiamo vedere come:

$$E[T_Q] = 10.11 \text{ s}$$

$$E[S] = 2.01 \text{ s}$$

$$E[T_S] = 24.06 \text{ s}$$

$$E[T_Q] + E[S_i] = 10.11 + 2.01 * 7 = 24.18 \text{ s}$$

E notiamo come il valore di $E[T_S]$ e $E[T_Q] + E[S]$ sono molto simili.

Lo stesso ragionamento facciamo per gli utenti medi nel nodo:

$$E[N_Q] = 4.15$$

$$\rho = 0.82$$

$$E[N] = 9.87$$

$$E[N_Q] + m * \rho = 4.15 + 7 * 0.82 = 9.89$$

Di seguito si hanno anche i valori teorici del tempo di risposta e del numero di utenti nel nodo, calcolati sempre all'interno della simulazione. Notiamo come questi sono molto simili ai valori ottenuti.

Valori teorici Biglietteria	

Wait	25.112497
Num Node	10.773261

5.2) Coda Veloce Jumanji

Valori Coda Veloce Jumanji			

Interarrival		24.91 +/-	0.63
Wait		13.08 +/-	0.81
Delay		4.32 +/-	0.63
Service		8.76 +/-	0.26
Num Node		0.53 +/-	0.04
Num Queue		0.18 +/-	0.03
Utilization		0.36 +/-	0.02

Da questi valori possiamo vedere come:

$$E[T_Q] = 4.32 \text{ s}$$

$$E[S] = 8.76 \text{ s}$$

$$E[T_S] = 13.08 \text{ s}$$

$$E[T_Q] + E[S] = 4.32 + 8.76 = 13.08 \text{ s}$$

E notiamo come in questo nodo il valore di $E[TS]$ e $E[TQ]+E[S]$ sono gli stessi.

Lo stesso ragionamento facciamo per gli utenti medi nel nodo:

$$E[N_Q] = 0.18$$

$$\rho = 0.36$$

$$E[N] = 0.53$$

$$E[N_Q] + \rho = 0.18 + 0.36 = 0.54$$

Qui di seguito sono riportati i valori teorici di questo nodo, calcolati sempre all'interno della simulazione. Anche qua si hanno dei valori teorici molto vicini ai valori ottenuti.

Valori teorici Coda Veloce Jumanji	

Wait	13.577680
Num Node	0.537676

5.3) Coda Lenta Jumanji

Valori Coda Lenta Jumanji			
Interarrival		8.96 +/-	0.16
Wait		1847.10 +/-	74.25
Delay		1838.14 +/-	74.12
Service		8.96 +/-	0.16
Num Node		205.12 +/-	5.49
Num Queue		204.12 +/-	5.49
Utilization		1.00 +/-	0.00

Da questi valori possiamo vedere come:

$$E[T_Q] = 1838.14 \text{ s} \approx 30.63 \text{ min}$$

$$E[S] = 8.96 \text{ s}$$

$$E[T_S] = 1847.1 \text{ s} \approx 30.79 \text{ min}$$

$$E[T_Q] + E[S] = 1838.14 + 8.96 = 1847,1 \text{ s} \approx 30.79 \text{ min}$$

E notiamo come anche in questo nodo il valore di $E[TS]$ e $E[TQ]+E[S]$ è uguale.

Lo stesso ragionamento facciamo per gli utenti medi nel nodo:

$$E[N_Q] = 204.12$$

$$\rho = 1$$

$$E[N] = 205.12$$

$$E[N_Q] + \rho = 204.12 + 1 = 205.12$$

Della coda lenta non sono stati riportati i valori teorici dalla simulazione, poiché in questa coda si ha un'utilizzazione =1, voluta appositamente per rispecchiare la realtà, dal momento che le attese, mediamente, per l'attrazione sono lunghe.

5.4) Coda Veloce Atlantide

Valori Coda Veloce Atlantide			

Interarrival		37.17 +/-	1.68
Wait		4.06 +/-	0.21
Delay		0.34 +/-	0.08
Service		3.72 +/-	0.17
Num Node		0.11 +/-	0.01
Num Queue		0.01 +/-	0.00
Utilization		0.10 +/-	0.01

Da questi valori possiamo vedere come:

$$E[T_Q] = 0.34 \text{ s}$$

$$E[S] = 3.72 \text{ s}$$

$$E[T_S] = 4.06 \text{ s}$$

$$E[T_Q] + E[S] = 0.34 + 3.72 = 4.06 \text{ s}$$

E notiamo come in questo nodo il valore di $E[T_S]$ e $E[T_Q] + E[S]$ è uguale.

Lo stesso ragionamento facciamo per gli utenti medi nel nodo:

$$E[N_Q] = 0.01$$

$$\rho = 0.1$$

$$E[N] = 0.11$$

$$E[N_Q] + \rho = 0.01 + 0.1 = 0.11$$

Qui di seguito sono riportati i valori teorici di questo nodo, calcolati sempre all'interno della simulazione. Anche qua si hanno dei valori teorici molto simili ai valori ottenuti.

Valori teorici Coda Veloce Atlantide	

Wait	3.978076
Num Node	0.105021

5.5) Coda Lenta Atlantide

Valori Coda Lenta Atlantide			

Interarrival		4.36 +/-	0.06
Wait		21.20 +/-	2.63
Delay		17.58 +/-	2.58
Service		3.62 +/-	0.06
Num Node		4.89 +/-	0.62
Num Queue		4.06 +/-	0.61
Utilization		0.83 +/-	0.02

Da questi valori possiamo vedere come:

$$E[T_Q] = 17.58 \text{ s}$$

$$E[S] = 3.62 \text{ s}$$

$$E[T_S] = 21.2 \text{ s}$$

$$E[T_Q] + E[S] = 17.58 + 3.62 = 21.2 \text{ s}$$

E notiamo come in questo nodo il valore di $E[T_S]$ e $E[T_Q] + E[S]$ sono gli stessi.

Vediamo anche gli utenti medi nel nodo:

$$E[N_Q] = 4.06$$

$$\rho = 0.83$$

$$E[N] = 4.89$$

$$E[N_Q] + \rho = 4.06 + 0.83 = 4.89$$

Qui di seguito sono riportati i valori teorici di questo nodo, calcolati sempre all'interno della simulazione. Anche qua si hanno dei valori teorici vicini ai valori ottenuti.

Valori teorici Coda Lenta Atlantide	

Wait	24.889381
Num Node	5.913717

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 partendo dal valore inserito di lambda e modificandolo.

Fascia mattutina: (con 7 serventi in biglietteria)

Lambda	Biglietteria	Coda Veloce Jumanji	Coda Lenta Jumanji	Coda Veloce Atlantide	Coda Lenta Atlantide
0.66 job/s	24.06 s \pm 2.58 s	13.08 s \pm 0.81 s	1847.1 s \pm 74.25 s (\approx 30.79 min \pm 1.24 min)	4.06 s \pm 0.21 s	21.2 s \pm 2.63 s
0,83 job/s	62.93 s \pm 6.76 s	14.33 s \pm 0.83 s	2200.58 s \pm 96.82 s (\approx 36.68 min \pm 1.61 min)	4.06 s \pm 0.24 s	35.77 s \pm 4.82 s
0,55 job/s	16.69 s \pm 0.94 s	12.28 s \pm 0.63 s	1436 s \pm 64.17 s (\approx 24 min \pm 1.07 min)	3.67 s \pm 0.18 s	11.98 s \pm 1 s

Fascia pomeridiana: (con 9 serventi in biglietteria)

Lambda	Biglietteria	Coda Veloce Jumanji	Coda Lenta Jumanji	Coda Veloce Atlantide	Coda Lenta Atlantide
0.83 job/s	18.68 s \pm 1.79 s	14.99 s \pm 0.93 s	2395.38 s \pm 86.35 s (\approx 39.92 min \pm 1.44 min)	3.93 s \pm 0.19 s	60.17 s \pm 7.86 s
0,66 job/s	13.25 s \pm 0.68 s	13.71 s \pm 0.9 s	1841.04 s \pm 86.34 s (\approx 30.79 min \pm 1.44 min)	4.12 s \pm 0.23 s	21.18 s \pm 2.66 s
0,94 job/s	31.07 s \pm 3.16 s	17.11 s \pm 1.31 s	2561.01 s \pm 103.68 s (\approx 42.69 min \pm 1.73 min)	4.07 s \pm 0.22 s	91.84 s \pm 10.97 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 sono stati ricavati con la formula $\frac{\lambda}{m*\mu} < 1$, e si ottiene:

Fascia oraria affollata:

$$- \frac{0,54}{m*0,067}, m > 8,06 \text{ di conseguenza } m = 9.$$

Fascia oraria media:

$$- \frac{0,43}{m*0,067}, m > 6,42 \text{ di conseguenza } m = 7.$$

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.

Così facendo questa simulazione ci dà delle statistiche dello stato stazionario del sistema 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.

La scelta di B e K è stata effettuata prendendo una delle più comuni coppie di valori delle due variabili, indicate dal libro Discrete Event Simulation - A First Course - Lemmis Park, e cioè B=256 e K=64.

Quando un'esecuzione finisce vengono calcolati i vari valori medi di ogni nodo e si fa l'intervallo di confidenza. Il livello di confidenza scelto è pari a 0,95 ed è stato utilizzato l'algoritmo di Welford per calcolare i tempi medi e gli intervalli.

Questi valori poi sono salvati nella struttura apposita di output.

Ovviamente, come detto, non si ha il problema del bias dello stato iniziale ed 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 di ogni centro al variare del valore B con K fisso a 64.

La simulazione presa in considerazione per ottenere queste statistiche è stata quella con fascia oraria mattutina.

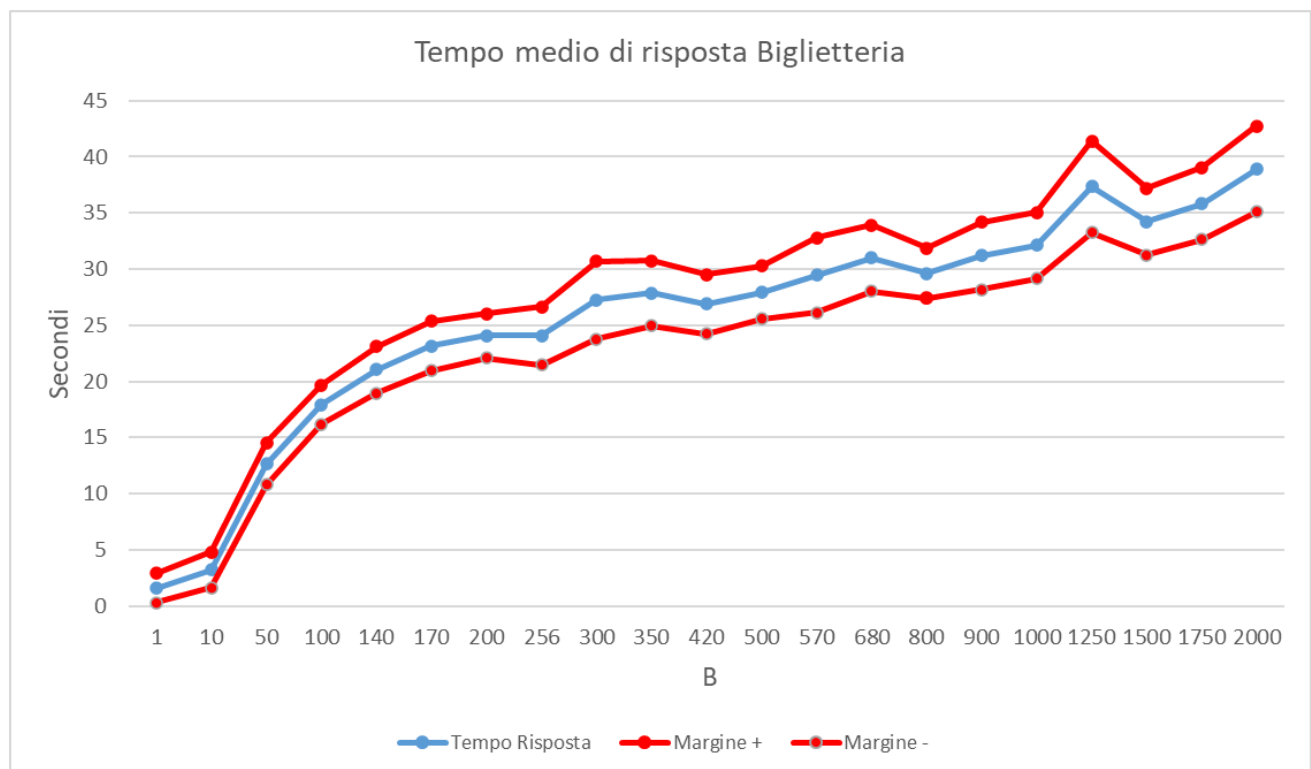
Inoltre, nei grafici la linea di mezzo (quella in blu) rappresenta il valore del tempo di risposta medio, mentre le altre due linee (quelle rosse) rappresentano gli estremi dell'intervallo di confidenza.

Biglietteria

Si può notare come nella biglietteria all'aumentare delle Batch aumenti anche il tempo di risposta. L'aumento non è esponenziale ma più lineare e in alcuni tratti è anche costante.

Il fatto che il tempo non si stabilizzi del tutto ma aumenti di poco volta per volta è normale dal momento che la biglietteria è il primo nodo che si incontra e la probabilità di accedervi è del 65%, per questo ci si aspettava che, all'aumentare dei job (il numero delle Batch), sarebbero aumentati i tempi di risposta dato l'afflusso sempre maggiore.

Una soluzione per rendere più costante il tempio medio di risposta è aumentare il numero dei server, ma dato che l'obiettivo principale dello studio è ridurli al minimo si è deciso di mostrare l'andamento con la configurazione minima.



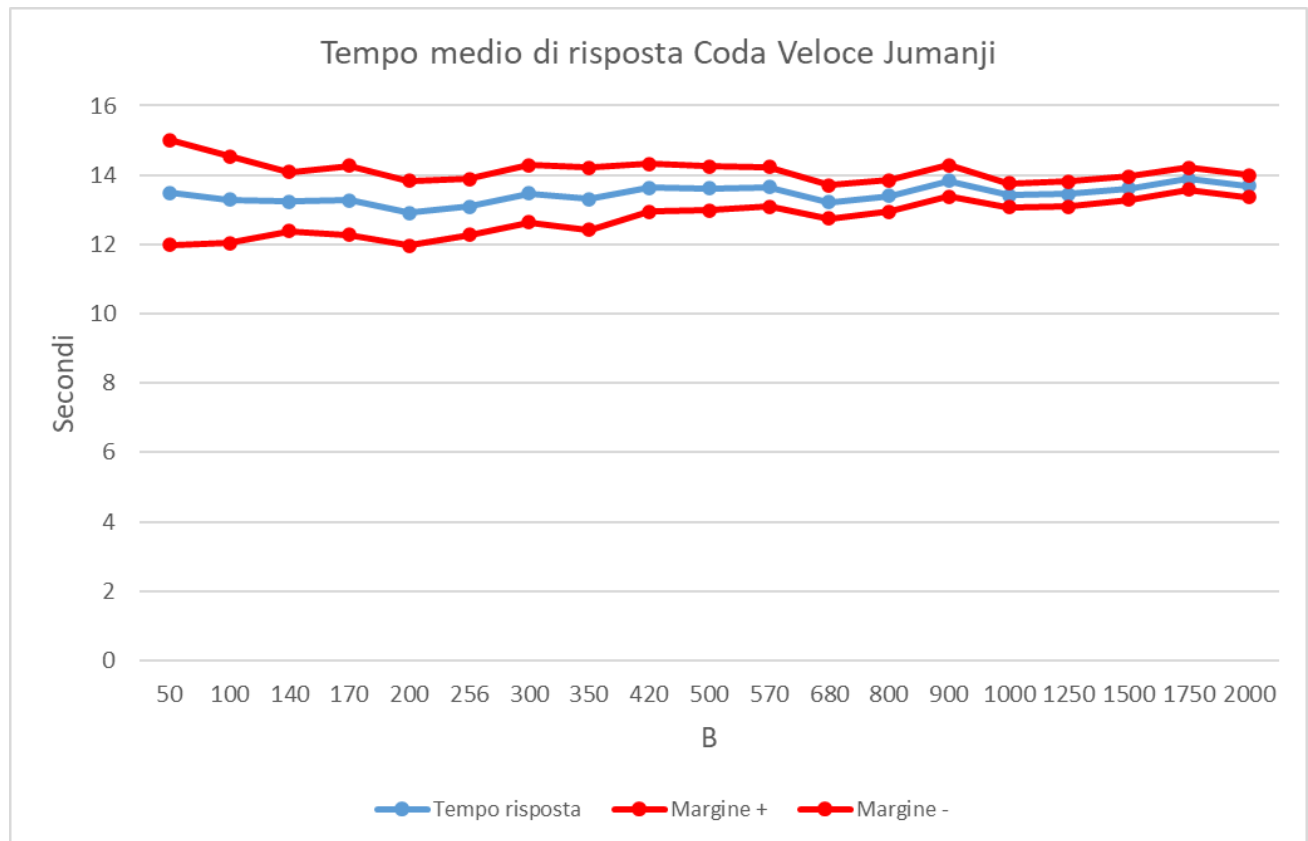
Coda Veloce Jumanji

In questa coda si ha sin da subito un valore costante, il tempo di risposta medio è quasi sempre sotto i 14 secondi.

Questo è dovuto soprattutto dal fatto che una ristretta cerchia di job accede a questa coda.

Nell'attrazione di Jumanji si accede con una probabilità del 60% e di questi solo il 10% entra in questa coda.

È possibile notare, quindi, come all'aumentare delle Batch non ci sia alcun problema sul tempo di risposta medio per questa coda.

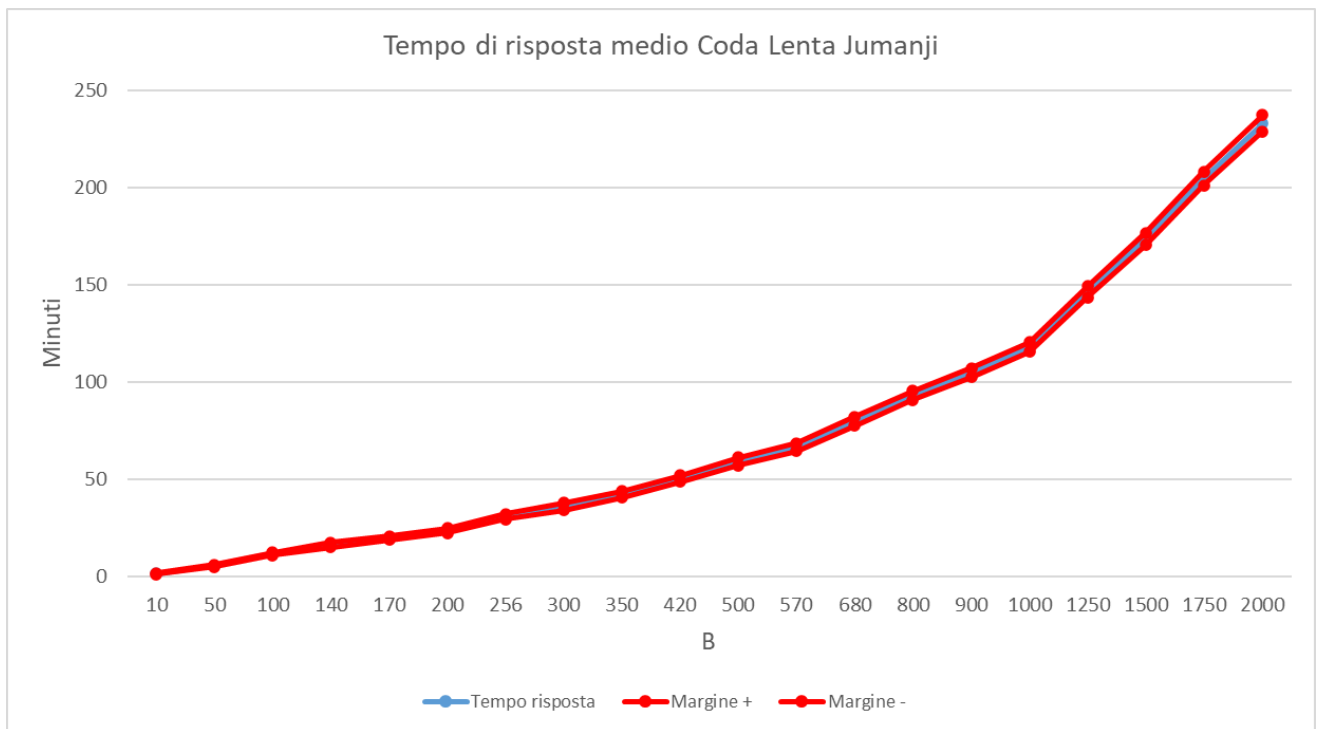


Coda Lenta Jumanji

Questa coda è il problema principale della simulazione del parco (oltre che nella realtà).

All'aumentare del numero di job si avrà sempre un tempo di risposta peggiore e con una crescita sempre più elevata, a differenza della biglietteria dove la crescita è molto più lenta.

In questo grafico, tra l'altro, non si vede bene la linea blu che corrisponde al tempo di risposta medio, poiché i secondi dei tempi ottenuti sono stati riportati in minuti e per questo l'intervallo di confidenza risulta molto più piccolo.

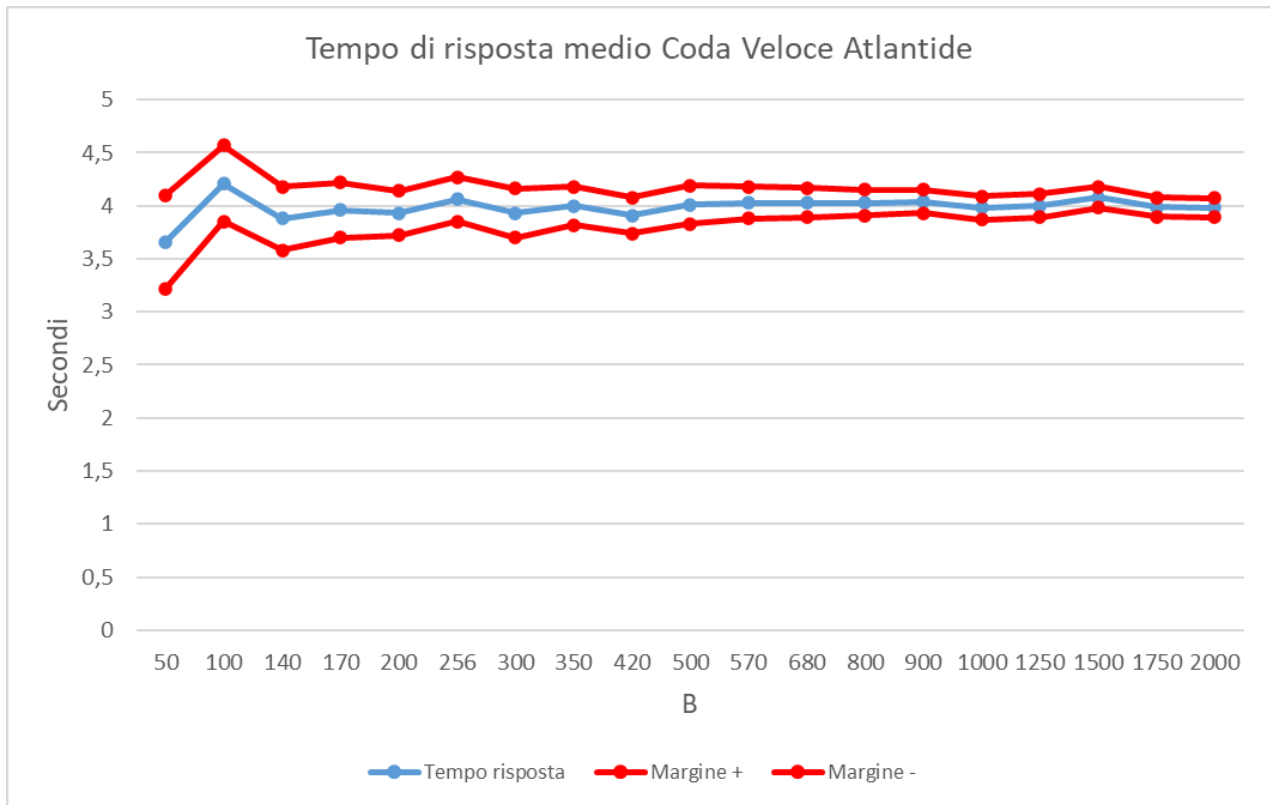


Coda Veloce Atlantide

In questa coda si ha la stessa situazione con la coda veloce di Jumanji.

Qui il tempo è ancora più basso visto che un numero minore di job vi accede (la probabilità di andare ad Atlantide è del 40% e di questo solo il 10% accede a questa coda) e, inoltre, il tempo di servizio qui è minore rispetto a Jumanji.

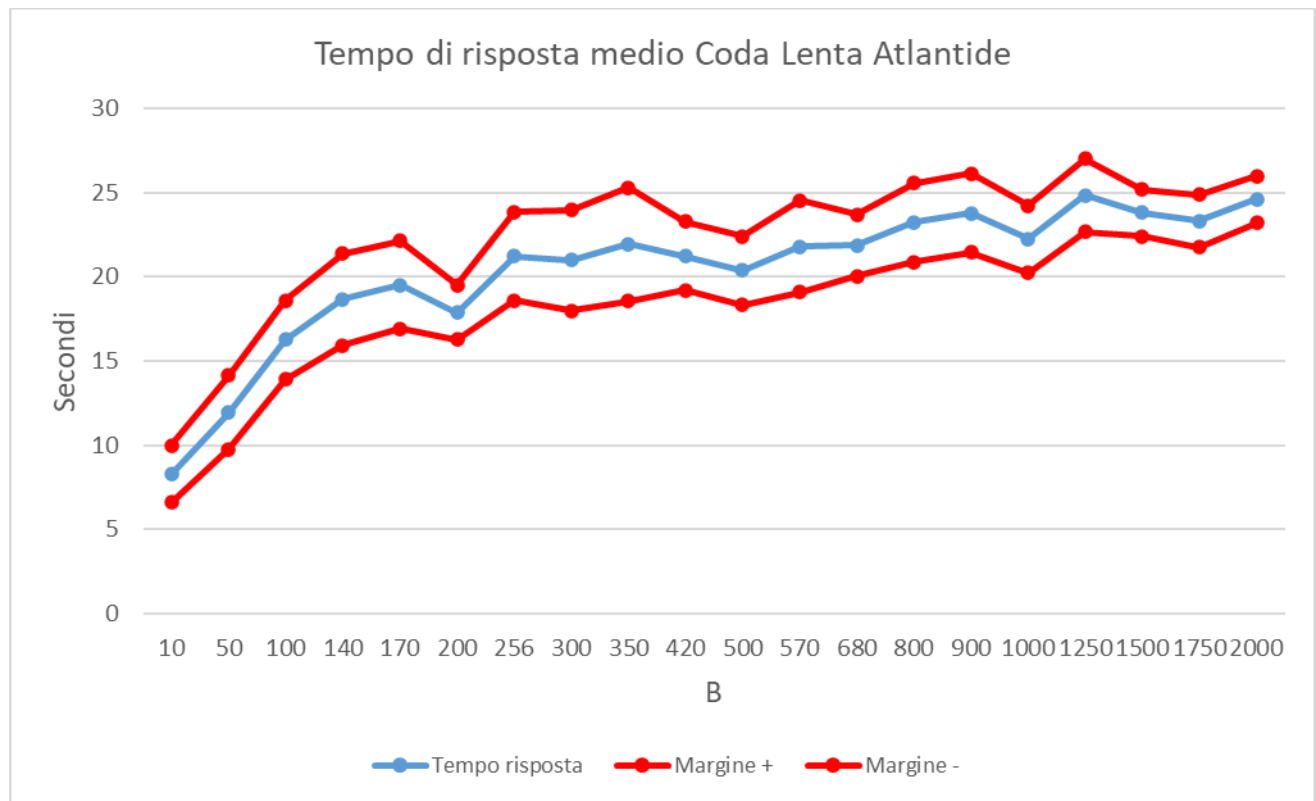
All'aumentare delle Batch possiamo vedere anche come l'intervallo si restringe sempre di più.



Coda Lenta Atlantide

Questa coda è molto simile alla biglietteria. Si ha un aumento del tempo di risposta all'aumentare delle Batch.

La crescita dei tempi qui è lenta, questo sempre grazie al tasso di servizio.



7.2) Simulazione a orizzonte finito

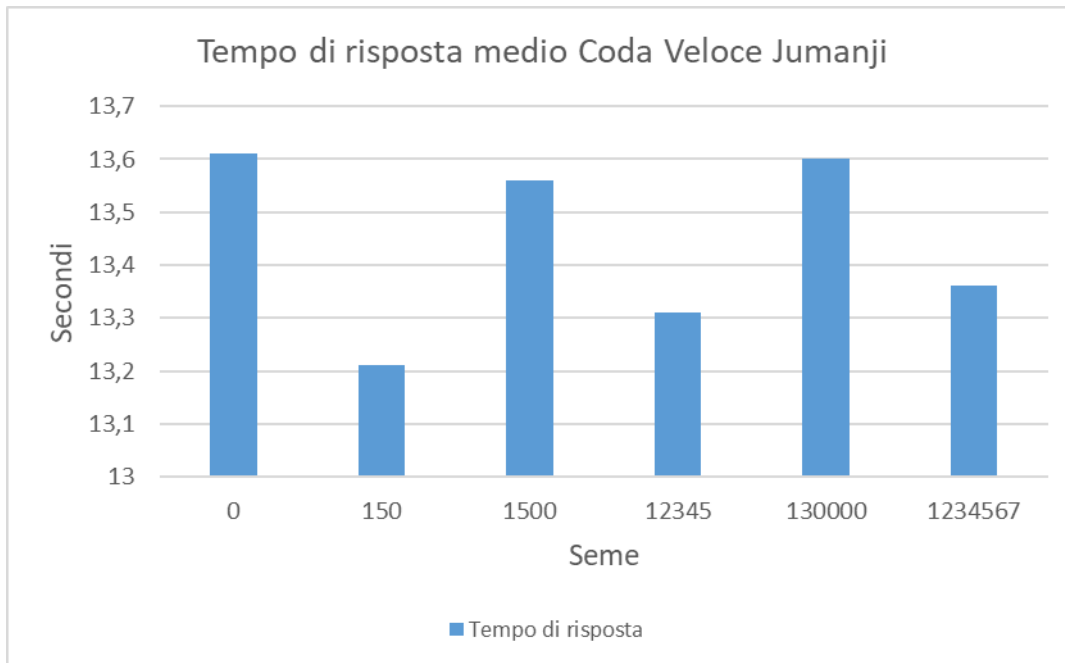
In questo tipo di simulazione si ha un tempo di 3 ore, che corrispondono alla durata di una fascia oraria.

In questo caso la simulazione viene effettuata più volte, di preciso 64. Tra l'altro ogni replica è indipendente dall'altra e per questo la funzione PlantSeed(SEED) è chiamata una volta sola prima di far partire le varie repliche, così da non avere sovrapposizioni.

Anche qui si salvano i valori medi ottenuti all'interno di una struttura per la gestione degli output. Ai fini del raggiungimento degli obiettivi posti precedentemente, questa simulazione non viene considerata. Comunque, sono state fatte delle run per studiare il comportamento del sistema in queste condizioni.

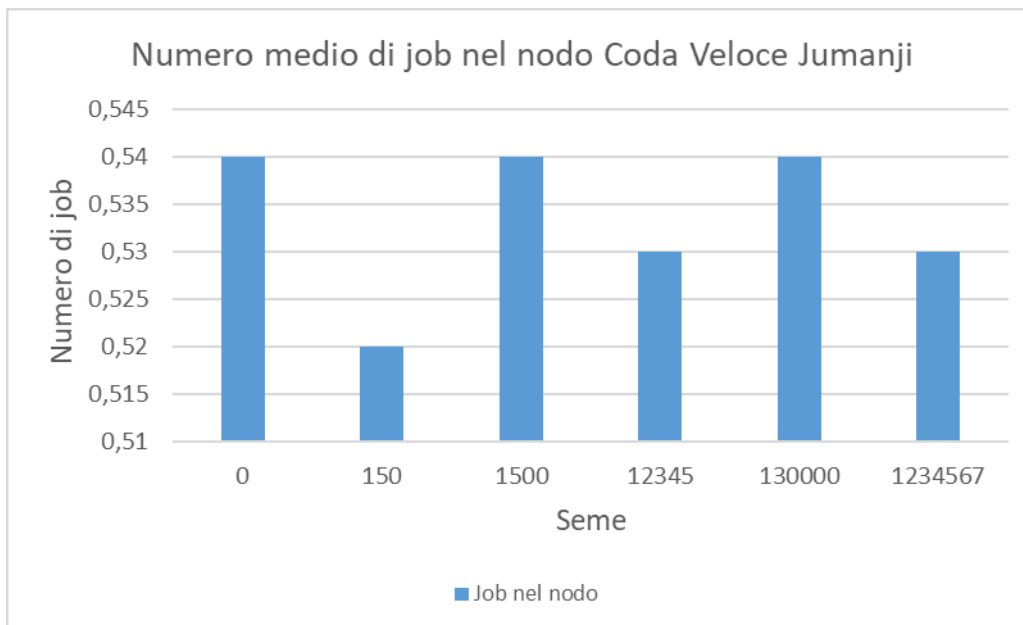
Qui di seguito troveremo un focus sull'attrazione Jumanji: The Adventure. In particolare, inseriremo un numero di fisso di job pari a 1000 e vedremo, al variare del seme, come varieranno i tempi medi di risposta, il numero di job nel nodo e l'utilizzazione.

Tempo di risposta Coda Veloce Jumanji



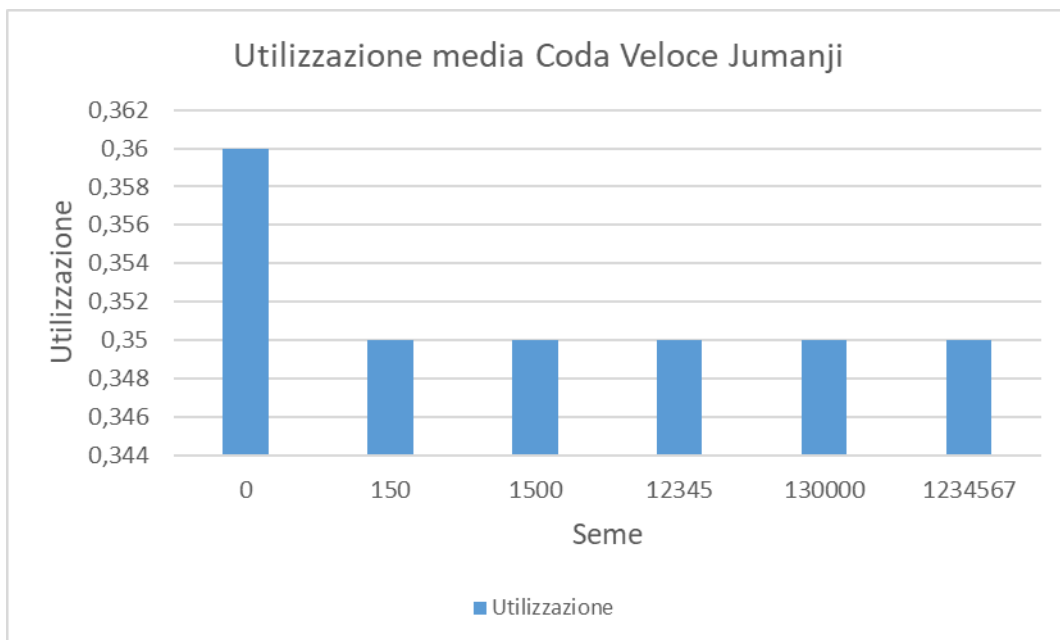
Nella coda veloce possiamo notare come i tempi di risposta medi sono diversi l'uno dall'altro ma si aggirano intorno allo stesso valore. Difatti sono compresi tra 13,2 secondi e 13,6 secondi, si può notare come ci sia una situazione stabile nel nodo indipendentemente dal seme.

Numero medio di job nel nodo Coda Veloce Jumanji



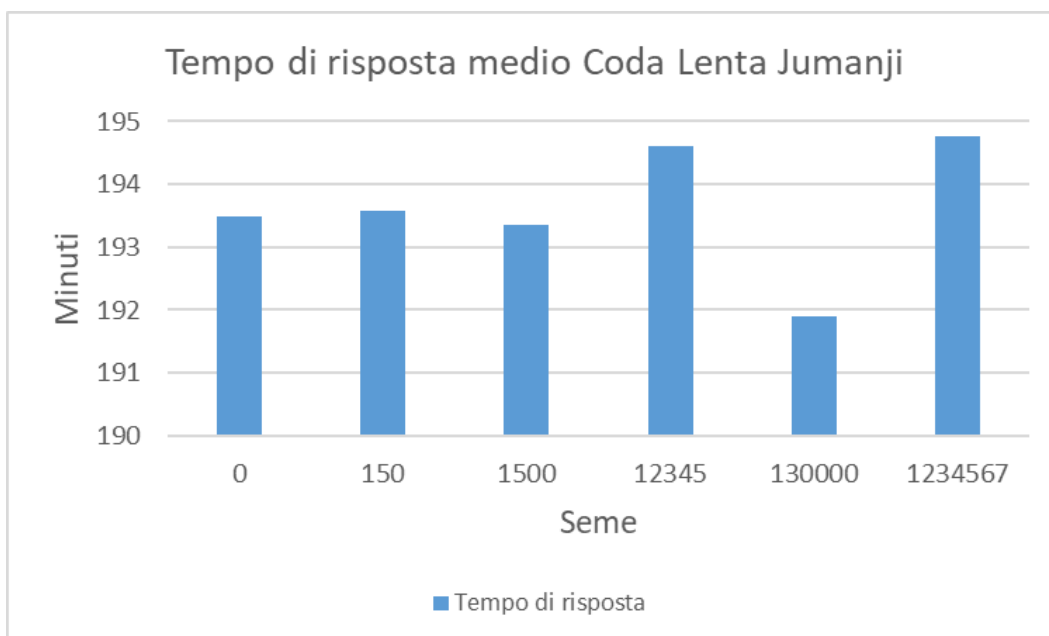
Anche per il numero medio troviamo una situazione stabile. I valori si aggirano tra 0,52 e 0,54 job nel nodo.

Utilizzazione Coda Veloce Jumanji



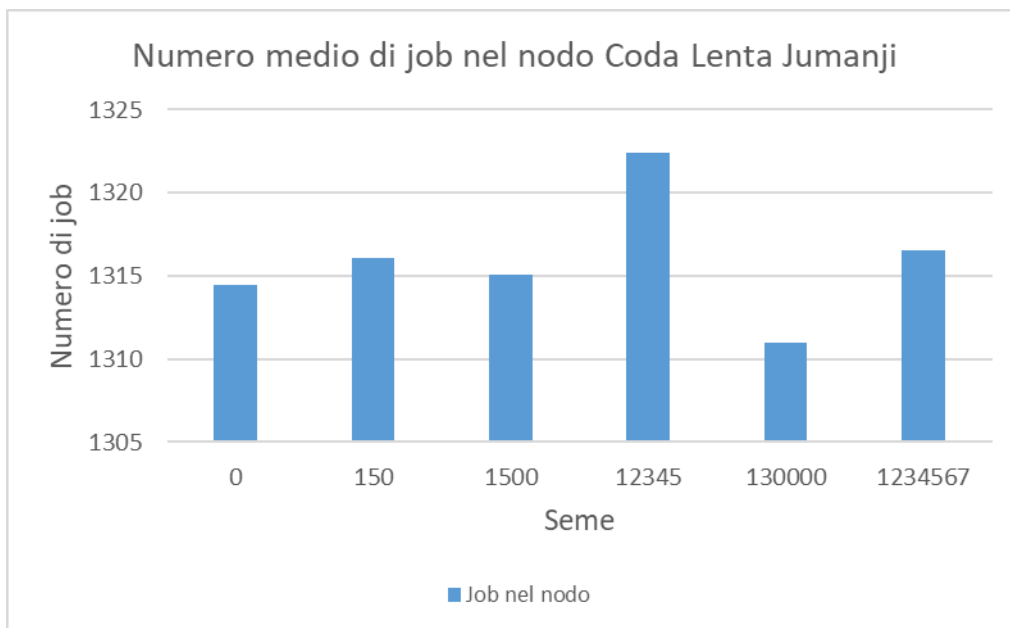
L'utilizzazione è sempre uguale tranne in un caso. Questi valori uguali sono coerenti con gli altri valori ottenuti su questa coda e marca ancora di più la stabilità del nodo.

Tempo di risposta Coda Lenta Jumanji



Questa coda, pur avendo dei tempi di risposta molto lunghi, mantiene valori simili indipendentemente dal seme utilizzato. Ci aggiriamo di un valore tra quasi 192 minuti e quasi 195 minuti, valori molto alti ma ribadiscono il fatto che questa coda è un collo di bottiglia.

Numero medio di job nel nodo Coda Lenta Jumanji



Anche in questo caso si hanno dei valori alti ma comunque simili.

Utilizzazione Coda Lenta Jumanji

Non è stata riportata graficamente poiché questa coda ha un'utilizzazione sempre pari a 1.

8) Esecuzione simulazione

8.1) Analisi QoS

L'obiettivo in questo QoS era di avere un tempo totale di risposta sotto i 46 minuti, nella fascia con più affluenza, per un utente che va in biglietteria e poi nella coda lenta di Jumanji ed inoltre avere un tempo totale di risposta sotto i 20 minuti, sempre nella stessa fascia, per un utente che va in biglietteria e poi nella coda lenta di Atlantide.

Effettuando la simulazione con la fascia oraria pomeridiana possiamo dire che:

- Per Jumanji, si riesce ad avere un tempo di risposta in biglietteria di 18,68 secondi e un tempo di risposta nella coda lenta di 39,92 minuti. Sommando questi due tempi di risposta medi otteniamo un totale di 40,23 minuti.
- Per Atlantide, abbiamo sempre un tempo di risposta in biglietteria di 18,8 secondi e un tempo di risposta nella coda lenta di 60,17 secondi. In totale si ha un tempo di risposta 78,97 secondi, un valore molto più basso rispetto ai 20 minuti che si hanno nella realtà. Questa differenza è data molto probabilmente dal fatto che nella realtà non vengono utilizzati tutti i posti disponibili dell'attrazione, ma vengono fatte entrare un numero ristretto rispetto alla capienza, per questo si ha nella realtà un servizio $E(S)$ maggiore e di conseguenza un'attesa maggiore.

Nel complesso, vedendo i risultati ottenuti il QoS è stato rispettato.

8.2) Analisi collo di bottiglia

Il collo di bottiglia di questo sistema è la coda lenta di Jumanji.

Questo è un problema che purtroppo per come è stata strutturata l'attrazione e per l'afflusso costante di persone che vi transitano è inevitabile.

D'altronde, in ogni simulazione, pur cambiando i valori dei tassi di arrivo, all'aumentare del numero di job nel sistema, aumenta in modo significativo anche il delay di questo nodo.

L'unica cosa che è possibile fare è ottimizzare gli altri nodi (come la biglietteria) in modo da avere complessivamente dei tempi di risposta migliori.

9) Conclusioni

L'analisi effettuata ha cercato di ricreare il più possibile il sistema reale.

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.

Per quanto riguarda invece l'attrazione Fuga da Atlantide, i valori che sono stati ricavati dalle simulazioni sono migliori rispetto ai valori reali. Pensiamo ai 60,17 secondi ricavati per il tempo di risposta nella Coda Lenta contro i 20 minuti che si hanno nella realtà per poter accedere al gioco. Questa differenza, come detto prima, sarà data dal fatto che nella realtà non si utilizza la capienza massima di persone che partecipano al gioco ma solamente un numero ristretto. Questa considerazione è stata fatta documentandosi online tramite video YouTube dove viene mostrata l'attrazione, ma soprattutto grazie a persone da me conosciute che sono state nel parco divertimenti ed hanno partecipato a questo gioco.