

Modelli di prestazioni di sistemi e reti

Prof.ssa Vittoria De Nitto Personè
a.a 2015-2016

UNIVERSITA' DEGLI STUDI DI ROMA TOR VERGATA
FACOLTA' DI INGEGNERIA INFORMATICA

Agostini Stefano	0234240
Belli Gabriele	0229621
Salomè Paolo	0233502

Parte prima

Test di uniformità del
generatore di Lehmer

Test di Uniformità del generatore di Lehmer

- E' un test di ipotesi non parametrico;
- Verifica che l'output di ripetute chiamate a random costituisca un campione distribuito come una $\text{Uniform}(0,1)$;
- Adattamento dei dati misurati con la distribuzione teorica $\text{Uniform}(0,1)$.

Test di Uniformità del generatore di Lehmer

Modalità: suddivisione dell'intervallo (0,1) in k bins. Successivamente n chiamate a Random e conteggio delle osservazioni che cadono nell'intervallo x-esimo, per ogni intervallo. Calcolo della statistica v :

$$v = \sum_{x=0}^{k-1} \frac{(o[x] - n/k)^2}{n/k}$$

Se il campione generato si adatta ad una distribuzione Uniform(0,1) allora $v \sim \chi^2_{k-1}$.

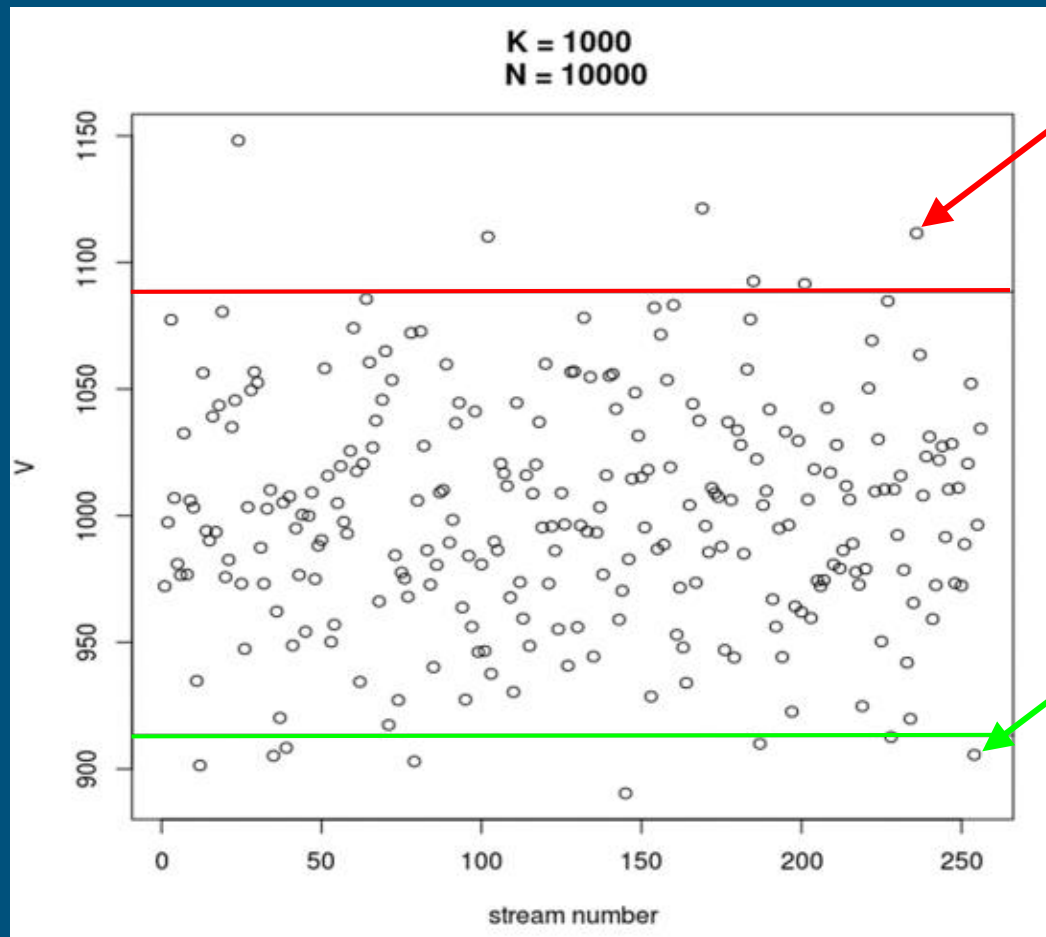
Quindi il test si costruisce nel seguente modo:

$P(v_1^* \leq v \leq v_2^*) \cong 1 - \alpha$ con $v_1^* = \chi^2_{k-1, \alpha/2}$ $v_2^* = \chi^2_{k-1, 1-\alpha/2}$

Risultato del test effettuato sul generatore di Lehmer con
moltiplicatore e modulo rispettivamente:

$(a,m) = (48271, 2^{31} - 1)$ a 256 streams.

$k = 1000$ bins e 10000 osservazioni per stream ($n = 10 \cdot k$).



Poca
uniformità

Troppo
uniformità

Parte seconda

Simulatore di traffico in
un sistema multi-tier

Panoramica

Lo scopo del progetto è quello di analizzare le prestazioni di un'architettura three-tier che fornisce un servizio web.

Un'applicazione realizzata con questa architettura è suddivisa in tre componenti:

- web server;

- application server;

- database di back-end.

L'accesso da parte di un utente al servizio web costituisce l'inizio di una sessione, che può essere formata da più richieste.

Panoramica (2)

Nel momento in cui viene generata una sessione, le richieste del client transitano più volte tra front-end server e back-end server prima di ritornare al client.

Quando la richiesta torna al client, che l'ha generata, quest'ultimo attende un tempo chiamato think time, prima di generare una nuova richiesta relativa alla sessione corrente.

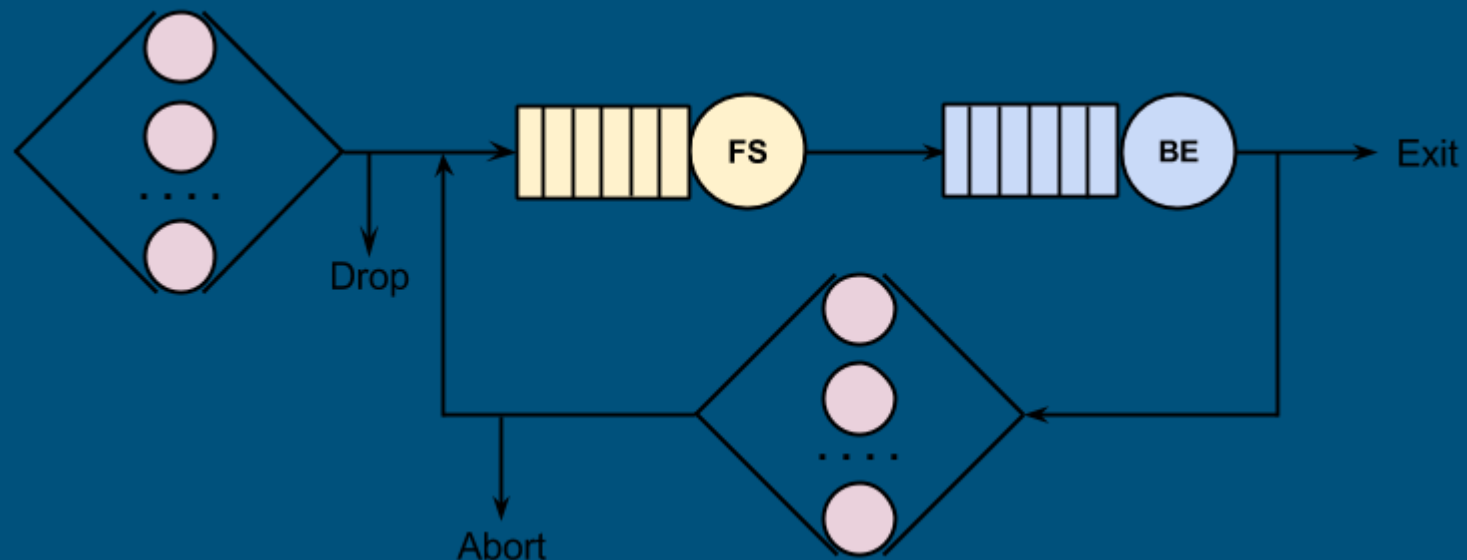
Qualora, invece, vengano esplicate tutte le richieste della sessione questa esce dal sistema.

La simulazione del sistema reale è stata condotta attraverso i seguenti passaggi:

1. Definizione obiettivi
2. Costruzione del modello concettuale
3. Costruzione delle specifiche del modello concettuale
4. Modello computazionale
5. Verifica
6. Validazione

Definizione degli obiettivi

L'obiettivo principale è quello di progettare e sviluppare un simulatore di traffico per il seguente sistema web Multi-Tier . Inoltre per gestire le situazioni di congestione si adotta un meccanismo di Overload Management o anche detto OMM.



Modello Concettuale:

Variabili di stato

Il sistema è descritto completamente dalle seguenti variabili di stato:

numero di richieste in servizio al tempo t nel front-end

numero di richieste in servizio al tempo t nel back-end

numero di richieste nella coda del front-end server al tempo t

numero di richieste nella coda del back-end server al tempo t

il numero di sessioni attive nel nodo di think.

Modello Specifiche

Nello sviluppo del modello delle specifiche, l'attenzione è stata rivolta alla definizione dei modelli di input da utilizzare nel modello di simulazione. Tali modelli sono stati definiti in base alle specifiche fornite:

tempo di servizio del front-end, distribuito esponenzialmente con media 0,00456 s;

tempo di servizio del back-end, distribuito esponenzialmente con media 0,00117 s;

tempo di interarrivo delle nuove sessioni, distribuito esponenzialmente di parametro 35 sessioni/s;

lunghezza delle sessioni, uniforme discreta di parametri $a=5$, $b=35$;

tempo di thinking, distribuito esponenzialmente di parametro 7 s.

Modello Specifiche

Generazione degli input condotta per mezzo del generatore di Lehmer con moltiplicatore e modulo rispettivamente $(a,m) = (48271, 2^{31} - 1)$.

Per le simulazioni viene utilizzato il modello next-event time-advance. In questo modello le variabili del sistema avanzano in maniera discreta secondo una sequenza di eventi programmata di volta in volta durante l'esecuzione.

Modello Specifiche:

Design del simulatore next-event

Nella costruzione del modello di simulazione next-event sono stati eseguiti principalmente 3 step:

- costruzione delle variabili di stato, che forniscono una descrizione completa del sistema;

- identificazione degli eventi;

- costruzione di un insieme di algoritmi che definiscono i cambiamenti di stato che devono essere eseguiti all'occorrenza di un dato evento.

Modello Specifiche:

Design del simulatore next-event (**Metriche**)

Per analizzare il sistema sono state calcolate le seguenti metriche:

- il tempo di risposta e il throughput del sistema;

- il numero di sessioni presenti, tempo di risposta, il throughput e l'utilizzazione del front-end;

- il tempo di risposta, il ritardo medio, il throughput e l'utilizzazione del back-end;

- la popolazione, il numero delle sessioni *droppate* e il numero delle richieste abortite del nodo di think;

- il tempo totale di esecuzione della simulazione.

Modello Specifiche:

Design del simulatore next-event (**Gestione Eventi**)

Gli eventi che caratterizzano il sistema sono quattro:

arrivo di una nuova sessione: un utente sta richiedendo al sistema di poter iniziare una nuova sessione. Se il meccanismo OMM non è attivo oppure non lo è quello di DROP la nuova sessione sarà accettata e accodata nel nodo di Front End. Qualora il meccanismo di DROP sia attivo, invece, per poter accettare una nuova sessione bisognerà attendere che l'utilizzazione sia inferiore al 75%.

fine servizio del Front End Server: la sessione esce dal nodo di Front End e viene accodata al nodo di Back End.

Modello Specifiche:

Design del simulatore next-event (Gestione Eventi (2))

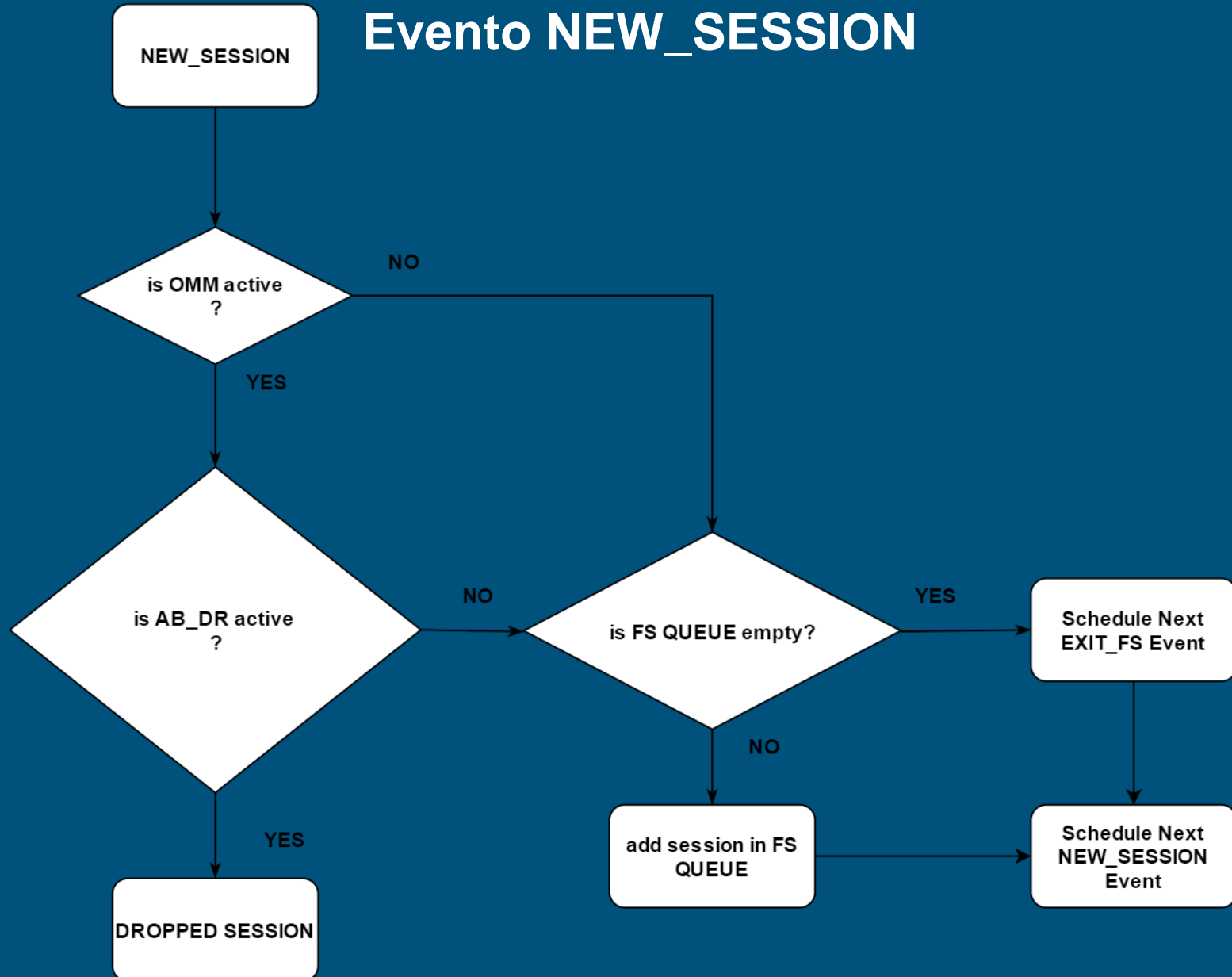
fine servizio del Back End Server: la sessione esce dal nodo di Back End e se ha completato tutte le sue richieste esce dal sistema altrimenti passa al nodo di think.

fine tempo di think: il comportamento del sistema durante questo evento cambia se il meccanismo di OMM è attivo o meno. Se non è attivo la sessione esce dal nodo di think e viene accodata al nodo di Front End. Se è attivo il comportamento del sistema dipende dal fatto che il meccanismo di abort sia attivo o meno: nel primo caso la sessione viene scartata altrimenti no.

Oltre ai quattro eventi che descrivono il comportamento del sistema considerato è stato generato un **evento di sampling** per monitorare lo stato del sistema senza influenzarlo.

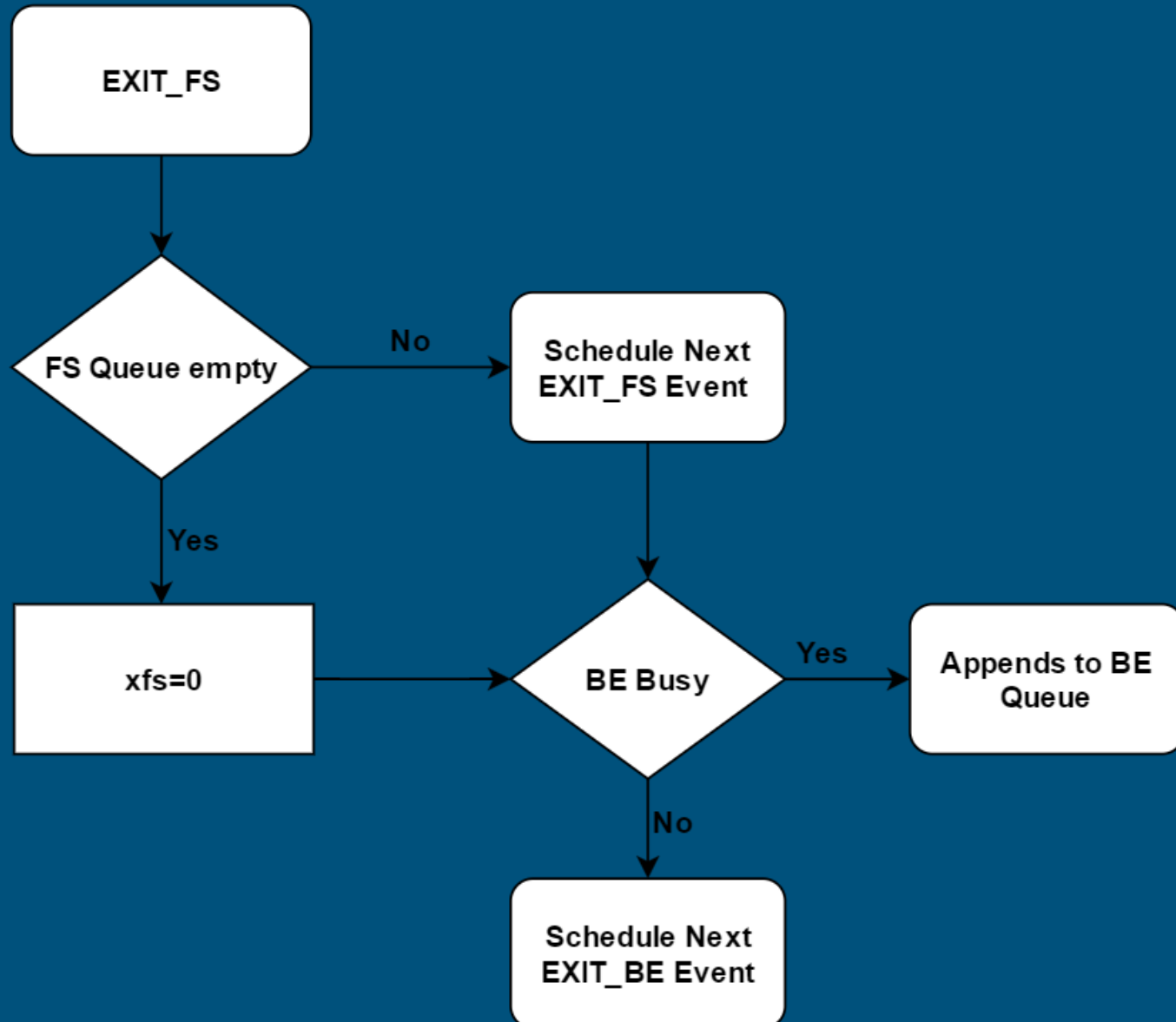
Modello Specifiche:

Evento NEW_SESSION



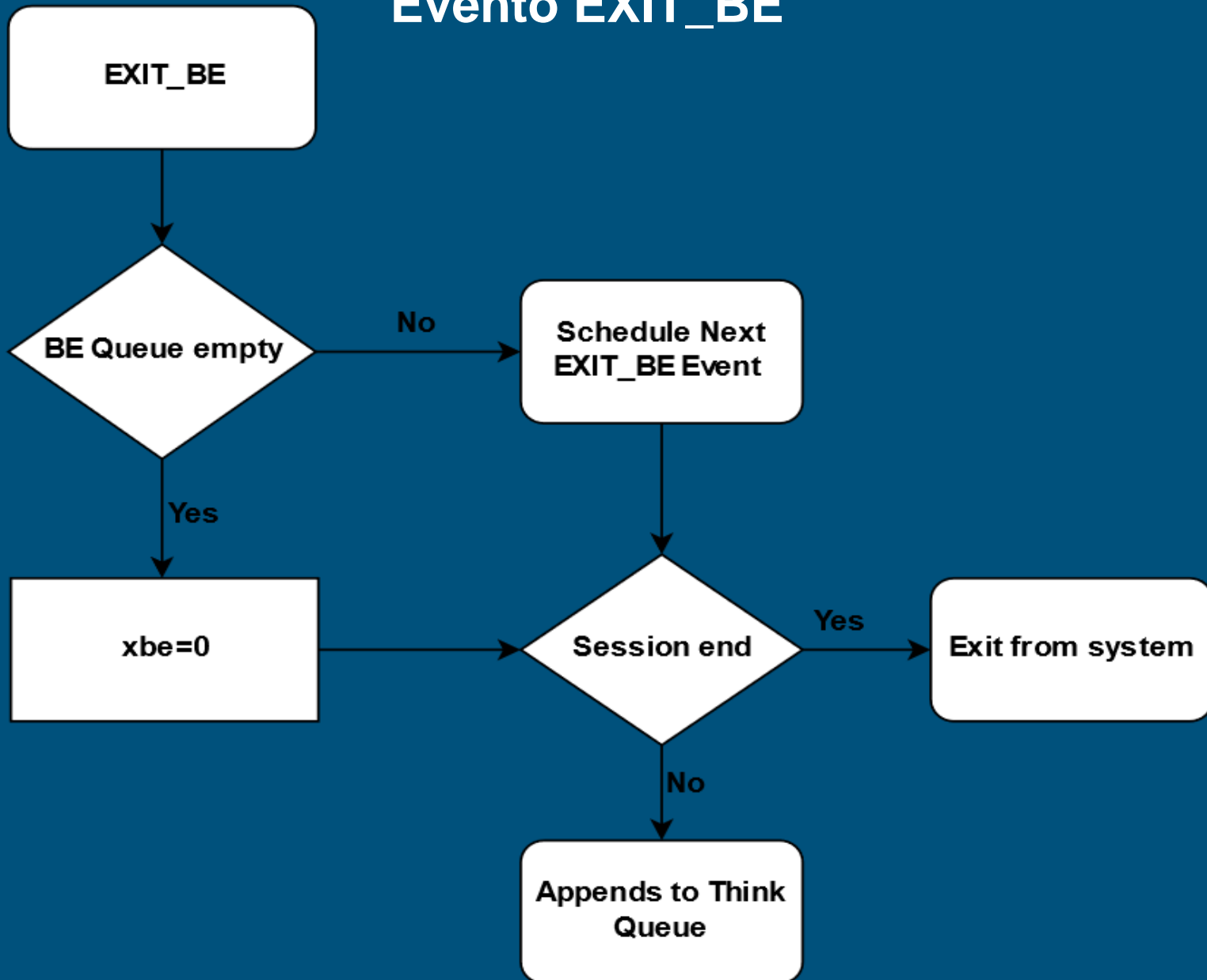
Modello Specifiche:

Evento EXIT_FS



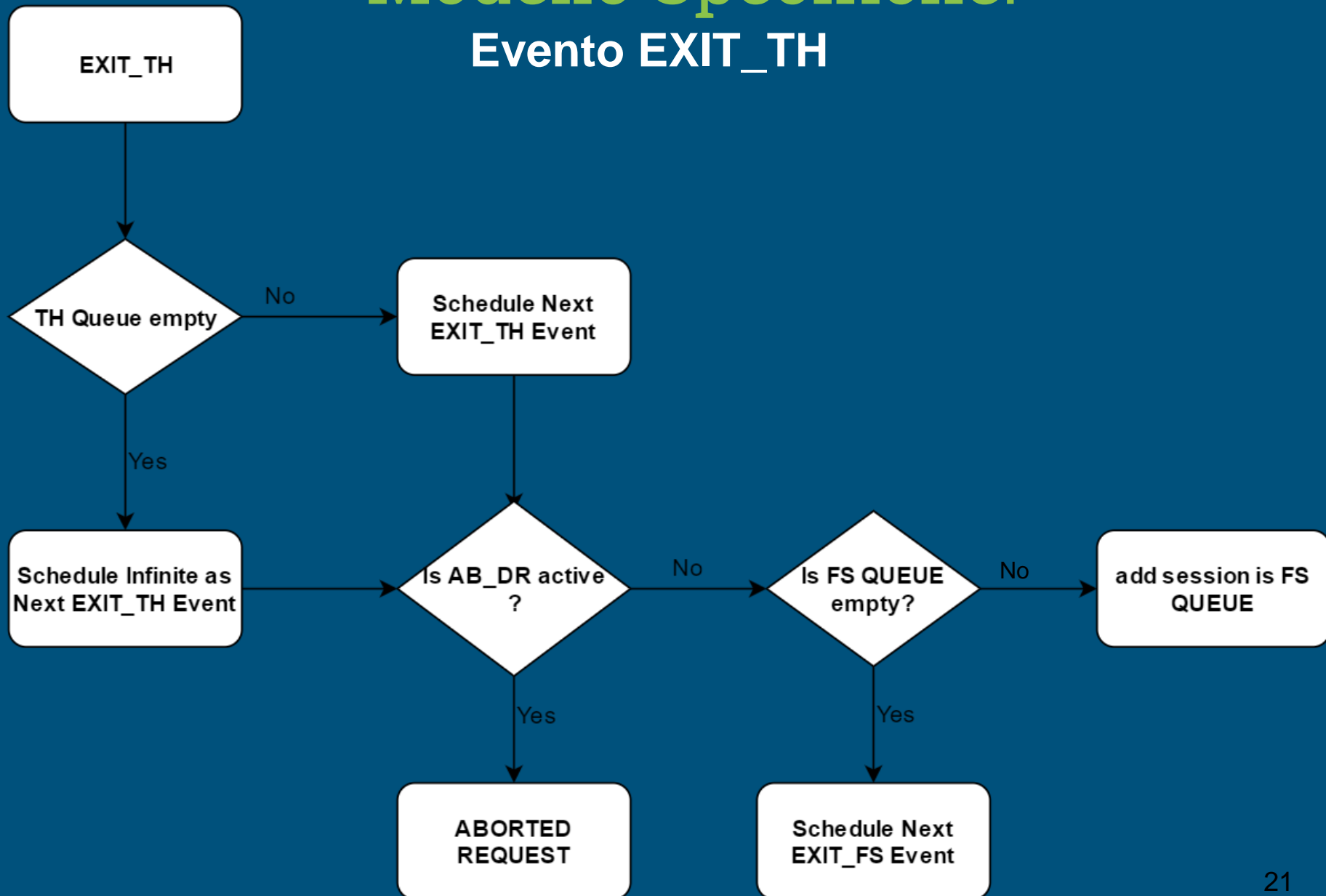
Modello Specifiche:

Evento EXIT_BE



Modello Specifiche:

Evento EXIT_TH



Modello Specifiche:

Design del simulatore next-event (Indici di prestazione)

Gli indici di prestazione calcolati sono:

Useful Throughput: indica il numero di sessioni completate dal sistema in un'unità di tempo.

Tempo di risposta del sistema: indica il tempo che intercorre tra l'istante in cui una richiesta entra nel front-server e l'istante in cui la stessa esce dal back-end server.

Aborted ratio: indica la percentuale di richieste abortite dal sistema rispetto al totale delle richieste processate.

Drop ratio: indica la percentuale delle sessioni rifiutate dal sistema rispetto al totale delle sessioni generate.

Modello Computazionale

Il simulatore è stato sviluppato utilizzando il linguaggio C

E' stato usato un approccio Next Event Time Advance per gestire l'avanzamento temporale

All'avvio dell'applicazione sono richiesti dei parametri di configurazione, quali la scelta dell'utilizzo del OMM e la scelta della metrica da analizzare della quale si vuole avere il trend temporale

Finita la simulazione verranno visualizzati gli intervalli di confidenza al 95% per ogni indice di prestazione

Modello Computazionale

Esempio di inizio e fine simulazione

```
TYPE 1 TO ACTIVATE THE OMM MECHANISM, 0 OTHERWISE [ENTER]:  
1  
ENTER THE METRIC TO MEASURE (ITS TIME TREND) [ENTER]:  
1.sys_resp  
2.sys_thr  
3.fs_resp  
4.fs_thr  
5.fs_population  
6.5.fs_util  
7.be_resp  
8.be_delay  
9.be_thr  
10.th_population  
11.drop_ratio  
12.abort_ratio
```

Inizio simulazione

```
System Response Time:  
Mean: 122.145654  
Var: 1.659639  
Confidence Interval (level 0.95): (+/-)0.366122  
  
System Useful Throughput:  
Mean: 3.691496  
Var: 0.001668  
Confidence Interval (level 0.95): (+/-)0.011606  
  
Fs Response Time:  
Mean: 122.144105  
Var: 1.659637  
Confidence Interval (level 0.95): (+/-)0.366122  
  
Fs Throughput:  
Mean: 174.519414  
Var: 2.705855  
Confidence Interval (level 0.95): (+/-)0.467489  
  
Popolation FS:  
Mean: 55888.100000  
Var: 85352243.765306  
Confidence Interval (level 0.95): (+/-)2625.588972
```

Fine simulazione

Modello Computazionale:

OMM

Il meccanismo di Overload Management analizza l'utilizzazione del front-end ed in base al valore riscontrato attiva o disattiva il meccanismo di AB_DR:

se AB_DR è attivo le nuove sessioni entranti al nodo di front-end vengono droppate mentre le nuove richieste da parte delle sessioni già accettate dal sistema vengono abortite.

se AB_DR è attivo e l'utilizzazione del FS scende sotto il 75% viene disattivato

se AB_DR è disattivo e l'utilizzazione del FS sale fino all' 85% viene attivato

Modello Computazionale:

Strutture utilizzate

Il modello è stato implementato utilizzando diverse strutture:

La struttura *node* rappresenta una sessione in coda nel centro.

La struttura *list* una coda dei centri che compongono il sistema

La struttura *server* rappresenta un servente del sistema

La struttura *area* tiene traccia del tempo cumulativo trascorso all'interno di un nodo, in coda e in servizio da parte di tutte le sessioni che attraversano il nodo stesso.

La struttura *clock* modella il tempo di simulazione memorizzando i tempi assoluti relativi all'evento corrente e al prossimo evento

La struttura *calendar* per tener traccia degli eventi successivi del sistema

Per misurare gli intervalli di confidenza sono stati seguiti i seguenti passi

Modello Computazionale:

Intervalli di confidenza

1. raccolta dati
2. aggiornamento valore medio e varianza con l'algoritmo di Welford one-pass

a. media:
$$\bar{x}_i = \frac{1}{i} (x_i + \bar{x}_{i-1})$$

b. varianza:
$$v_i = v_{i-1} + \left(\frac{i-1}{i} \right) (x_i - \bar{x}_{i-1})^2$$

3. calcolo del valore critico
$$t^* = \text{idfStudent}\left(n-1, 1 - \frac{\alpha}{2}\right)$$

$$\bar{x} \pm \frac{t^* s}{\sqrt{n-1}}$$
4. calcolo degli estremi dell'intervallo di confidenza

dove s rappresenta la deviazione standard ed n la dimensione del campione

Verifica

La fase di verifica consente di dimostrare l'effettiva consistenza del programma con il modello delle specifiche.

Si è utilizzata la scrittura sul file per verificare il corretto flusso delle sessioni all'interno del sistema; si è notato e verificato inoltre che la lista contenente la sessione si riempia e si svuoti in modo corretto.

Come da specifiche, il simulatore parte per ogni singolo run da e termina in uno stato di quiete: il numero delle sessioni è nullo e tutte le variabili, di stato e di supporto, utilizzate tornano ai valori di partenza.

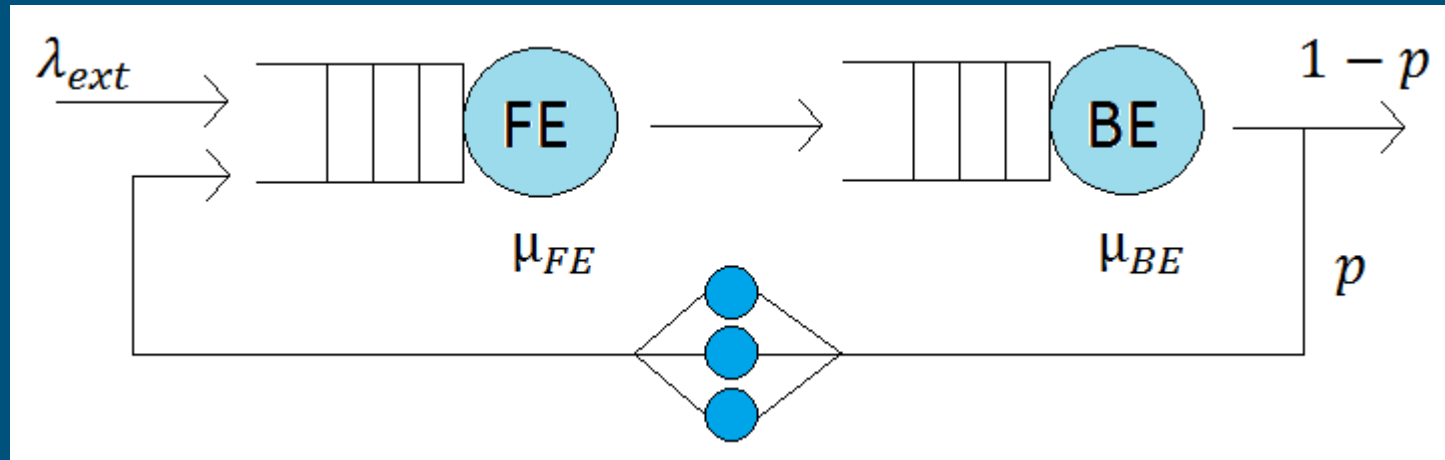
Inoltre il numero delle sessioni rifiutate cresce consistentemente con la presenza del meccanismo OMM e con il cambiamento del flusso in entrata.

Verifica (2)

Infine, come ultima verifica, è stato dimostrato che, superato il tempo di STOP_SIMULATION , nessuna nuova sessione fosse accettata dal sistema e, successivamente a tale istante, si assisteva allo svuotamento del sistema in modo consistente.

Non sono stati riscontrati errori in fase di compilazione ed esecuzione.

Validazione (*OMM* Attivo)



{ $X_{FE} = \rho_{FE} * \mu_{FE}$, dove ρ_{FE} è da considerare pari a 0.8 con OMM attivo.

Throughput del FE osservato (media tra 400 run ciascuno di 10000 secondi) risulta pari a 174.6 rich/s.

Throughput del FE teorico è pari ad $X_{FE} = 0.8 * \mu_{FE} = 175.44$ rich/s.

Quindi, come ci aspettavamo: $X_{FE \text{ OSSERVATO}} \cong X_{FE}$

Validazione (*OMM* Attivo) (2)

Simulazione

- Utilizzazione del FS misurata attraverso la struttura *Area*
- Il calcolo cumulativo non rende il meccanismo *OMM* sufficientemente sensibile alla rilevazione di picchi di carico
- Ciò porta comunque ad una crescita indefinita della coda del FS, seppur molto rallentata rispetto a quella che avrebbe il sistema senza meccanismo di *OMM*
- Tempo di risposta medio del FS cresce molto lentamente ma illimitatamente.

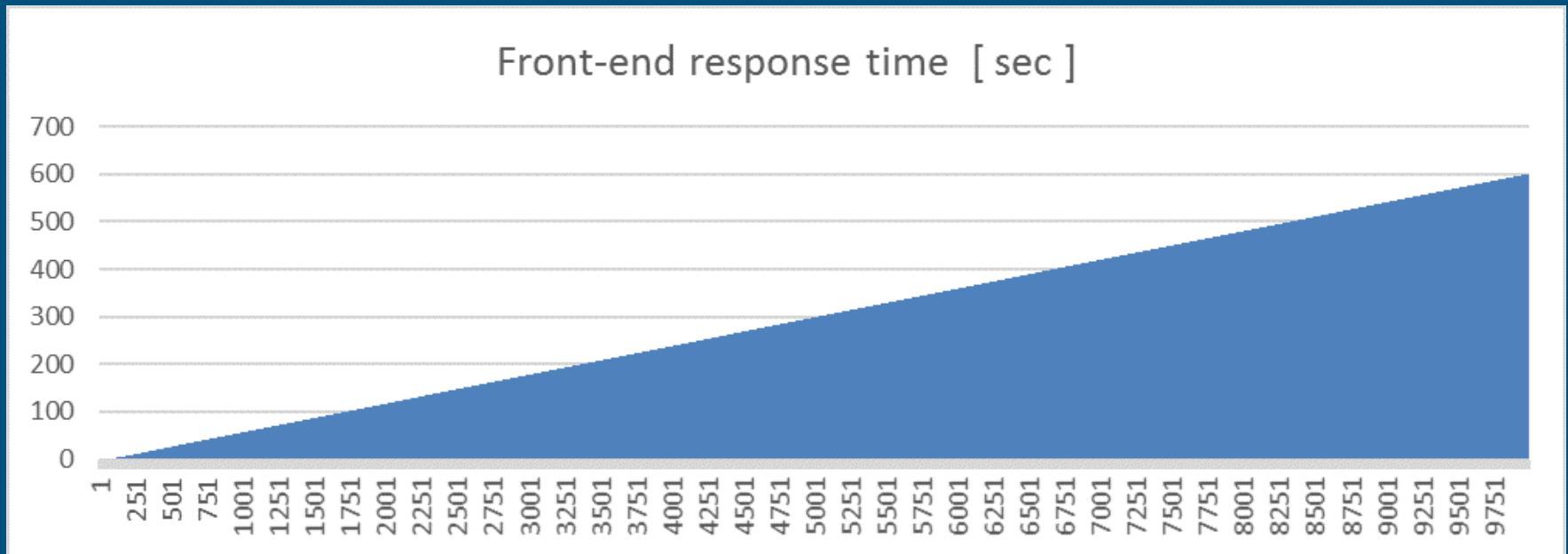
Modello Teorico

- Nel modello teorico è sempre soddisfatta $X_{FE} = 0.8 * \mu_{FE}$
- Quindi teoricamente il sistema raggiunge la stabilità in quanto:

$$\lambda_{FE} = 0.8 * \mu_{FE} < \mu_{FE}$$

Trend Temporale FS Response Time

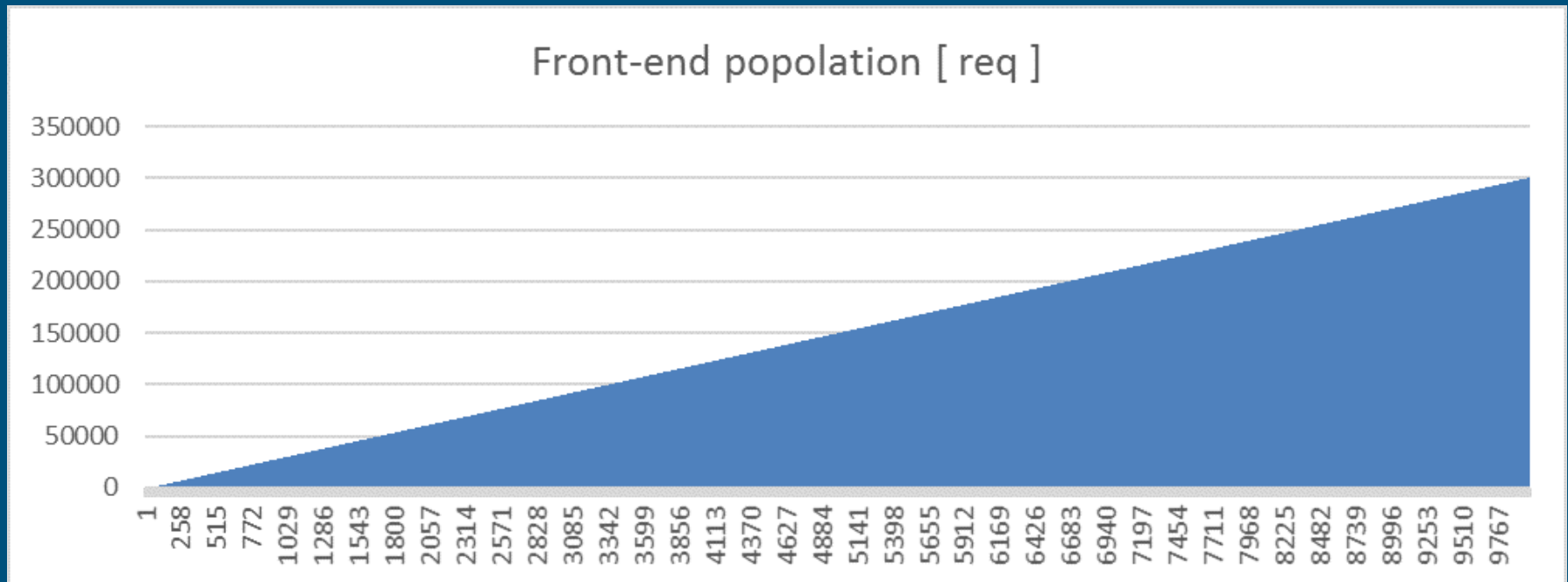
OMM disattivato



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Trend Temporale FS Population

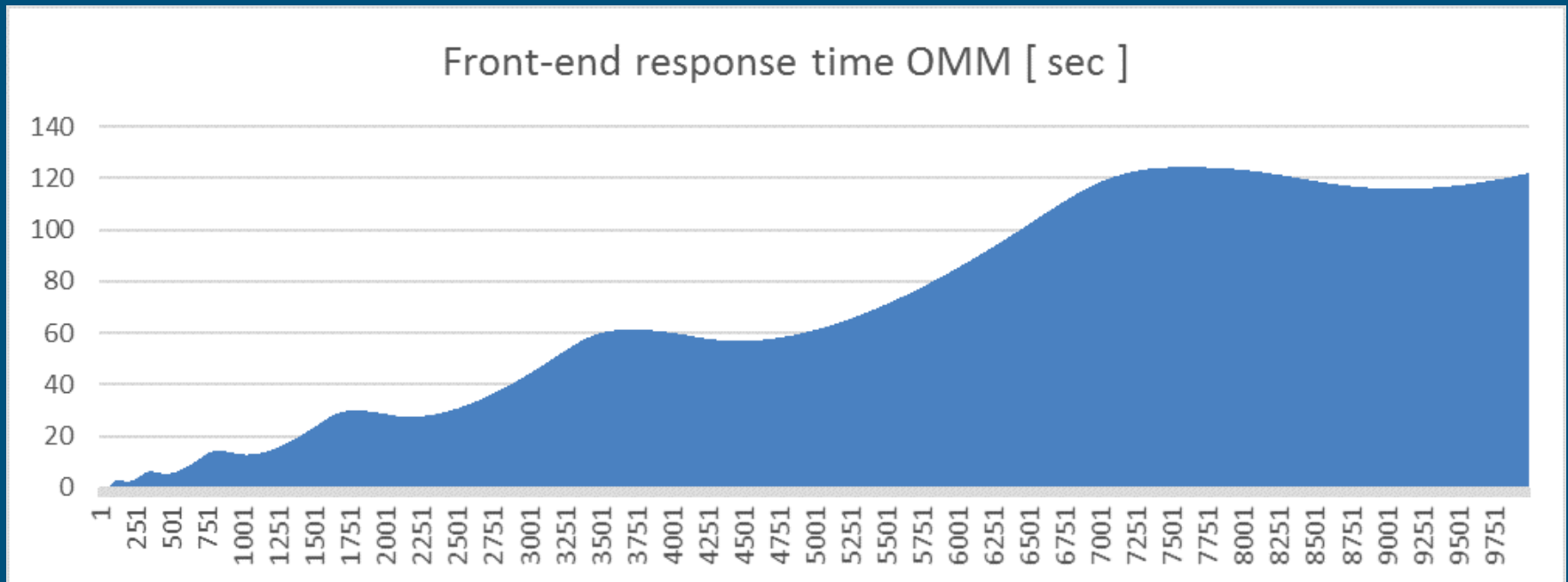
OMM disattivato



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Trend Temporale FS Response Time

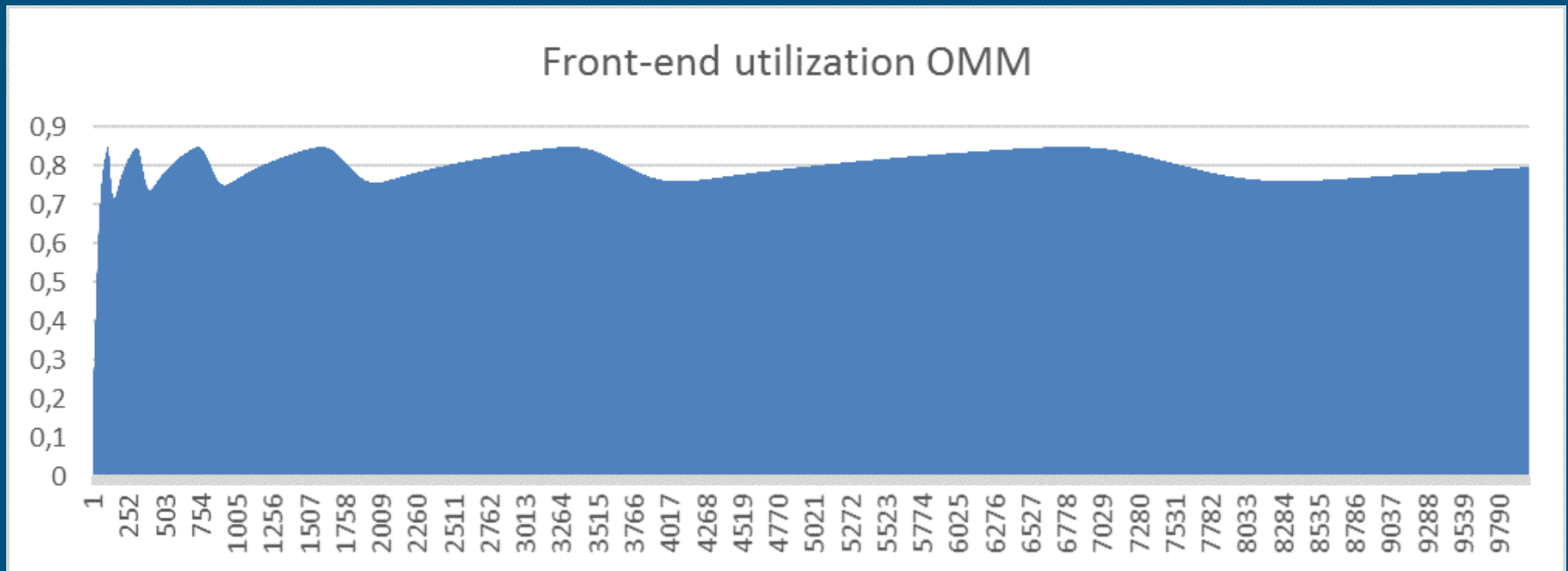
OMM attivo



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Trend Temporale FS Utilization

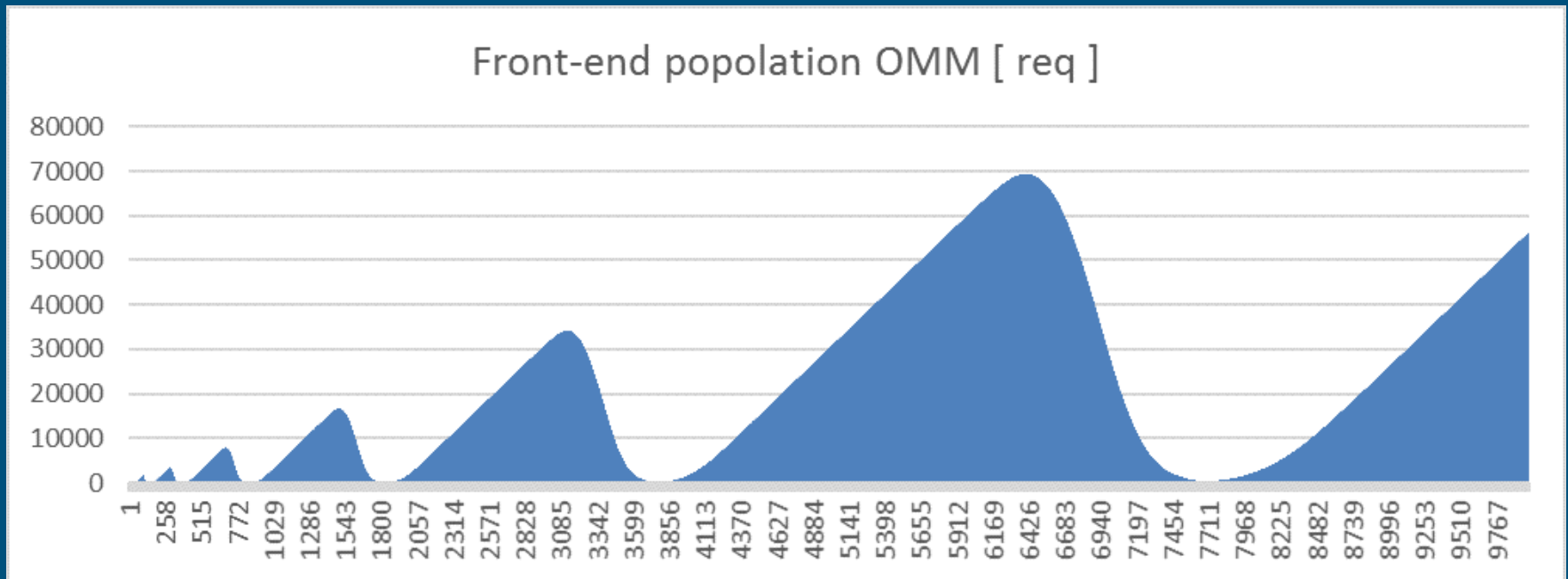
OMM attivo



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Trend Temporale FS Population

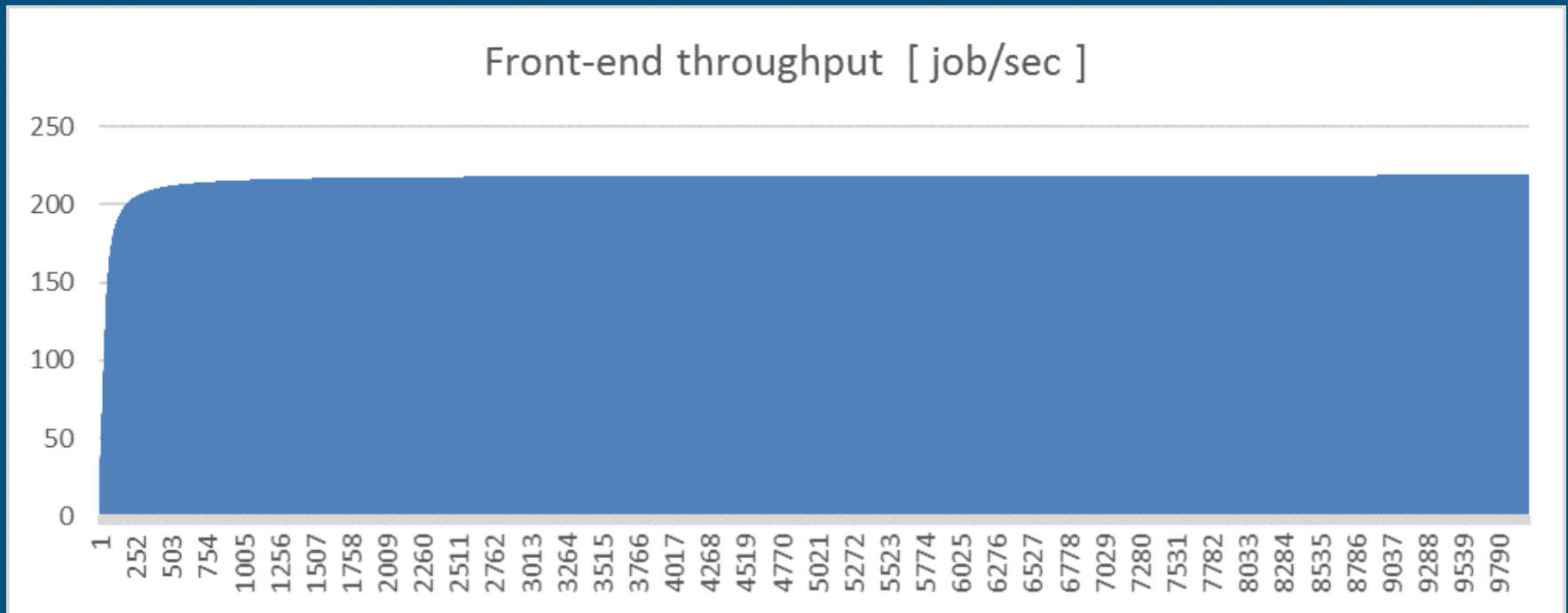
OMM attivo



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Trend Temporale FS Throughput

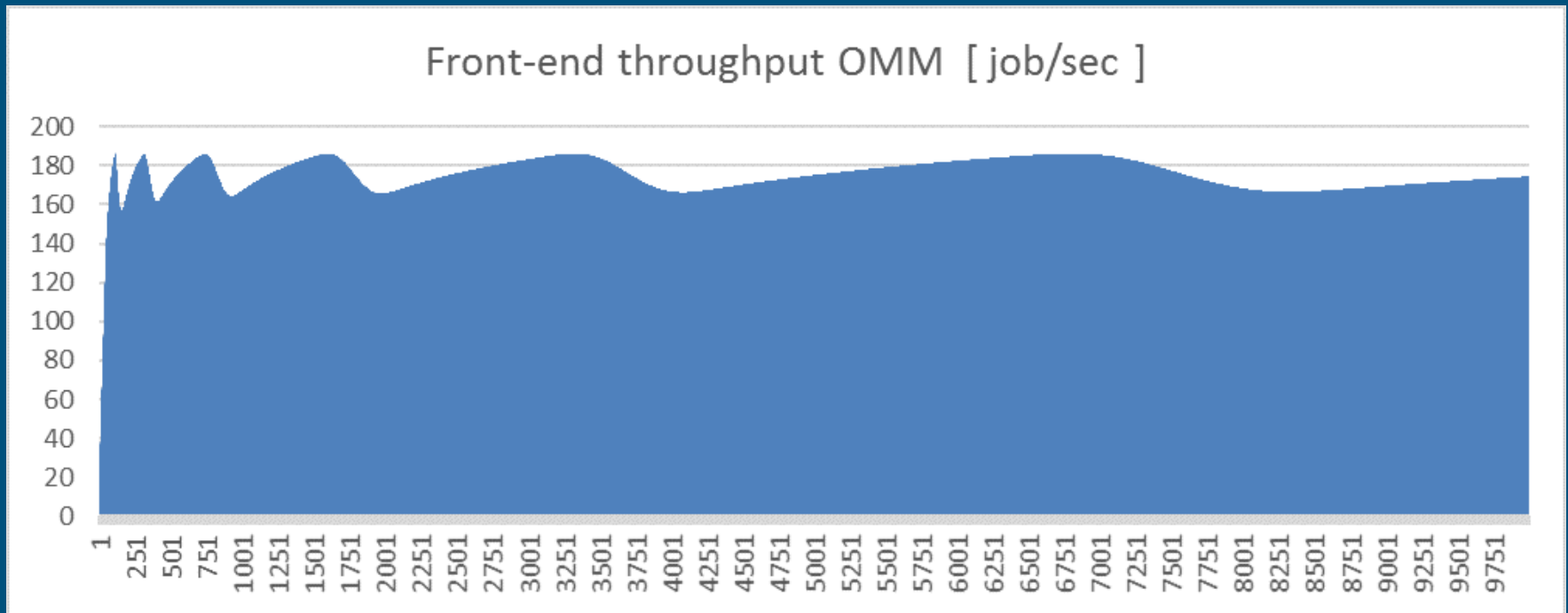
OMM disattivato



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Trend Temporale FS Throughput

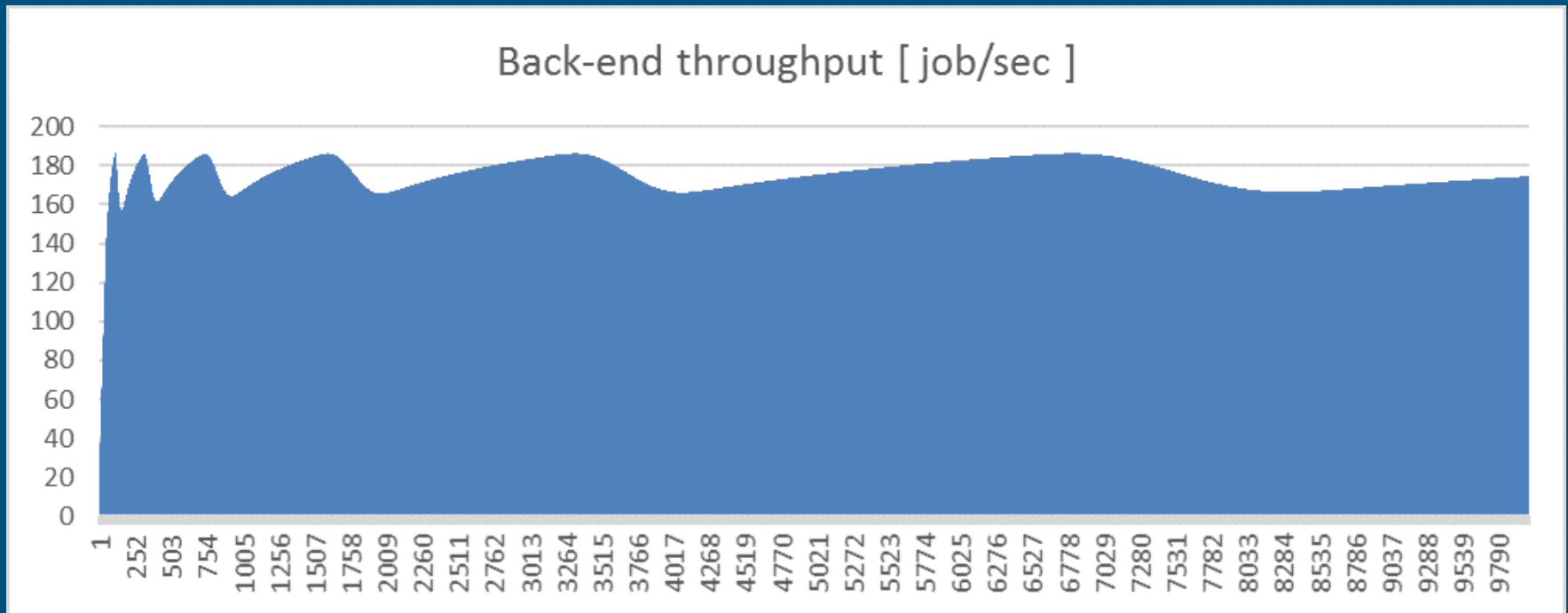
OMM attivo



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

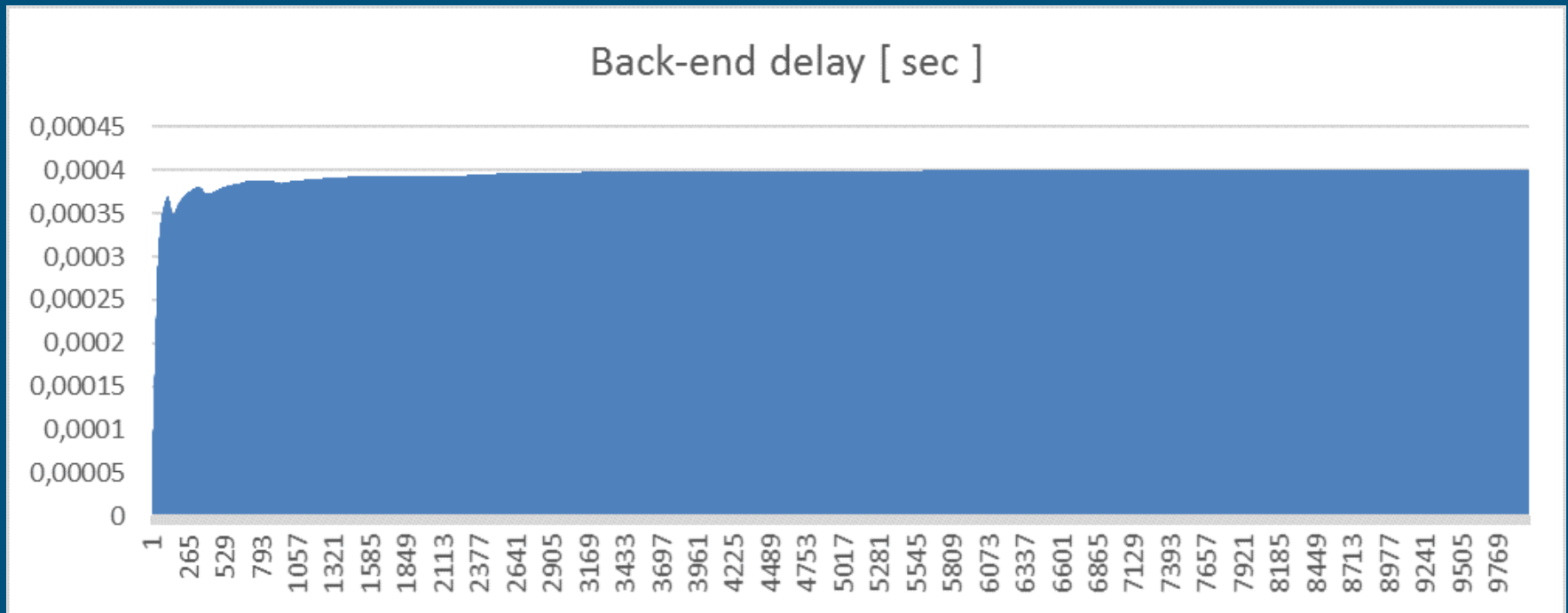
Trend Temporale BE Throughput

OMM attivo



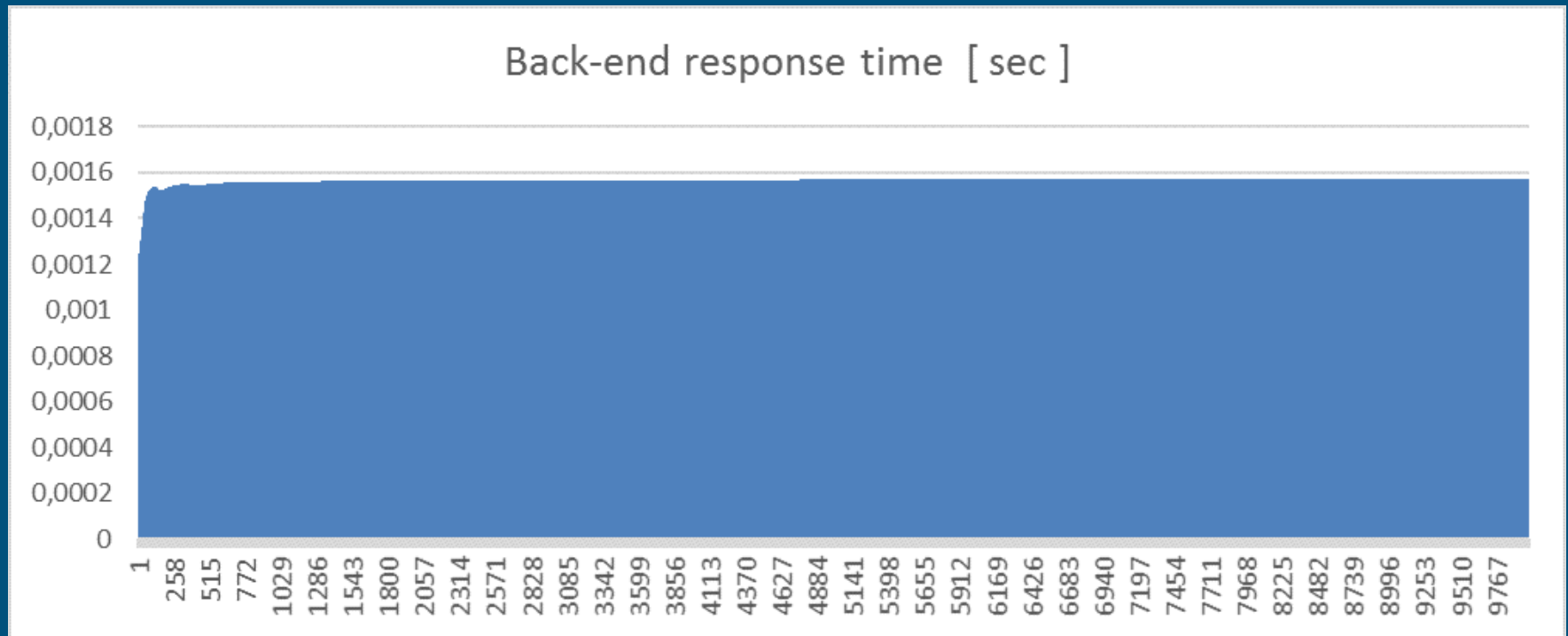
Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Trend Temporale BE Delay



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Trend temporale BE Response Time



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Correlazione

Il grado di correlazione fra due variabili viene espresso mediante i cosiddetti indici di correlazione. Questi assumono valori compresi tra $[-1, +1]$, un indice di correlazione pari a zero indica un'assenza di correlazione.

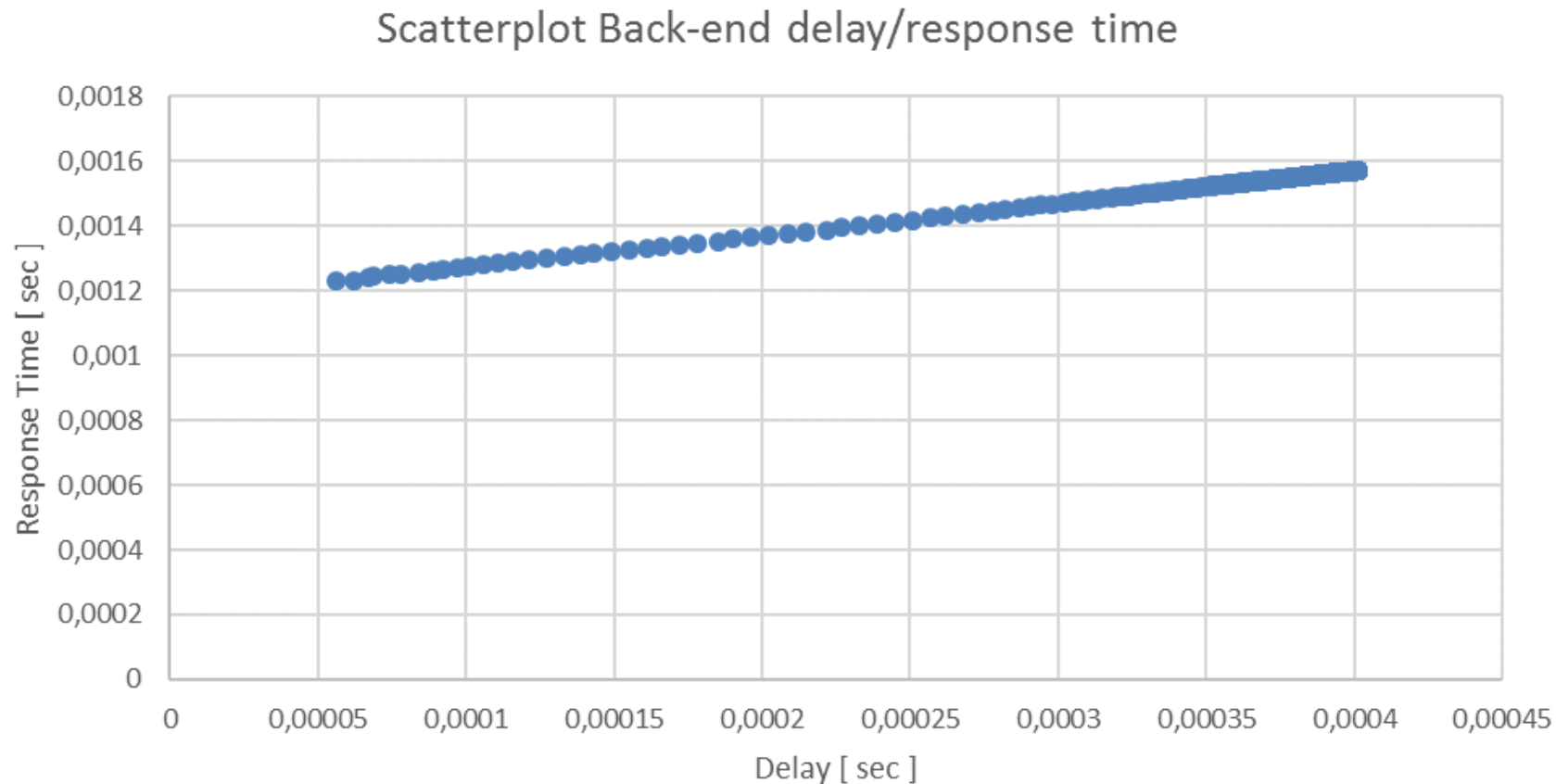
Due variabili indipendenti hanno sicuramente un indice di correlazione pari a 0, ma al contrario un valore pari a 0 non implica necessariamente che le due variabili siano indipendenti. Dai dati raccolti è possibile calcolare il grado di correlazione tra il delay ed il response time del back end.

Correlazione (2)

Il calcolo è stato effettuato a posteriori utilizzando il coefficiente di correlazione di Pearson-Bravais il quale è calcolato come rapporto tra la covarianza delle due variabili ed il prodotto delle loro deviazioni standard.

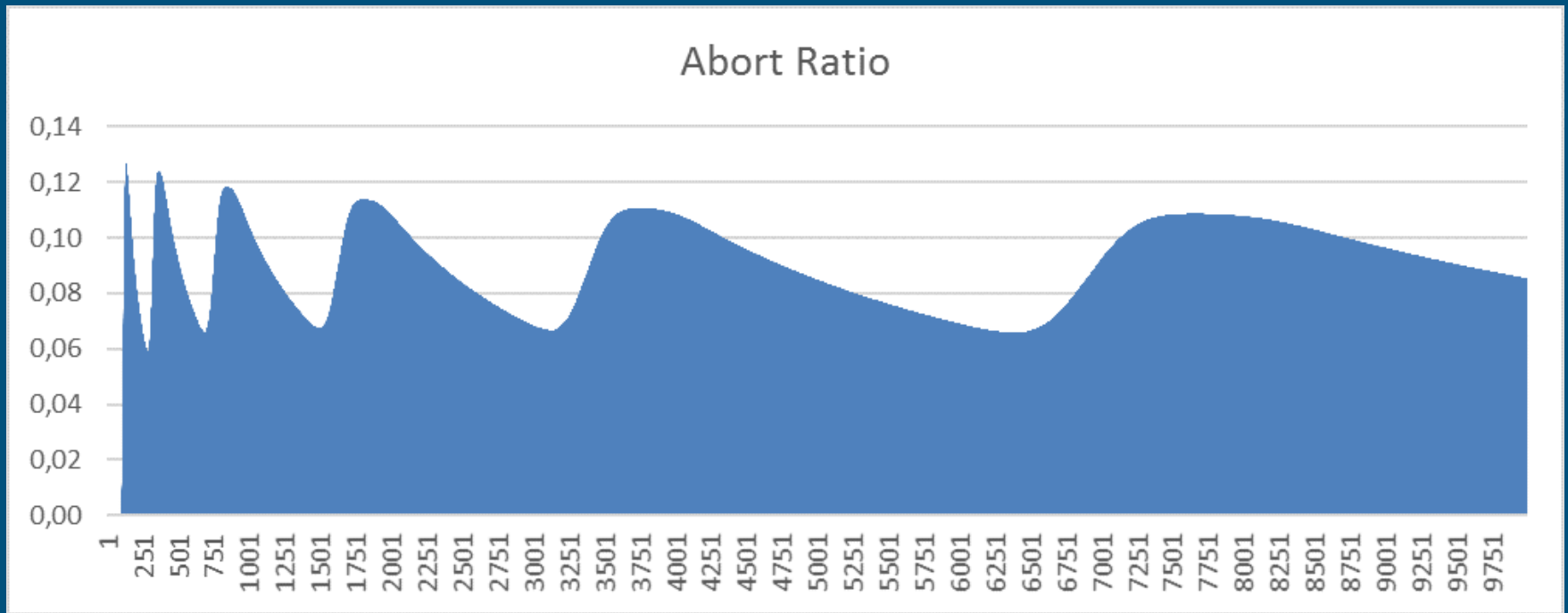
$$-1 \leq \rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^n (x_i - \mu_x)^2} \sqrt{\sum_{i=1}^n (y_i - \mu_y)^2}} \leq +1$$

Correlazione (3)



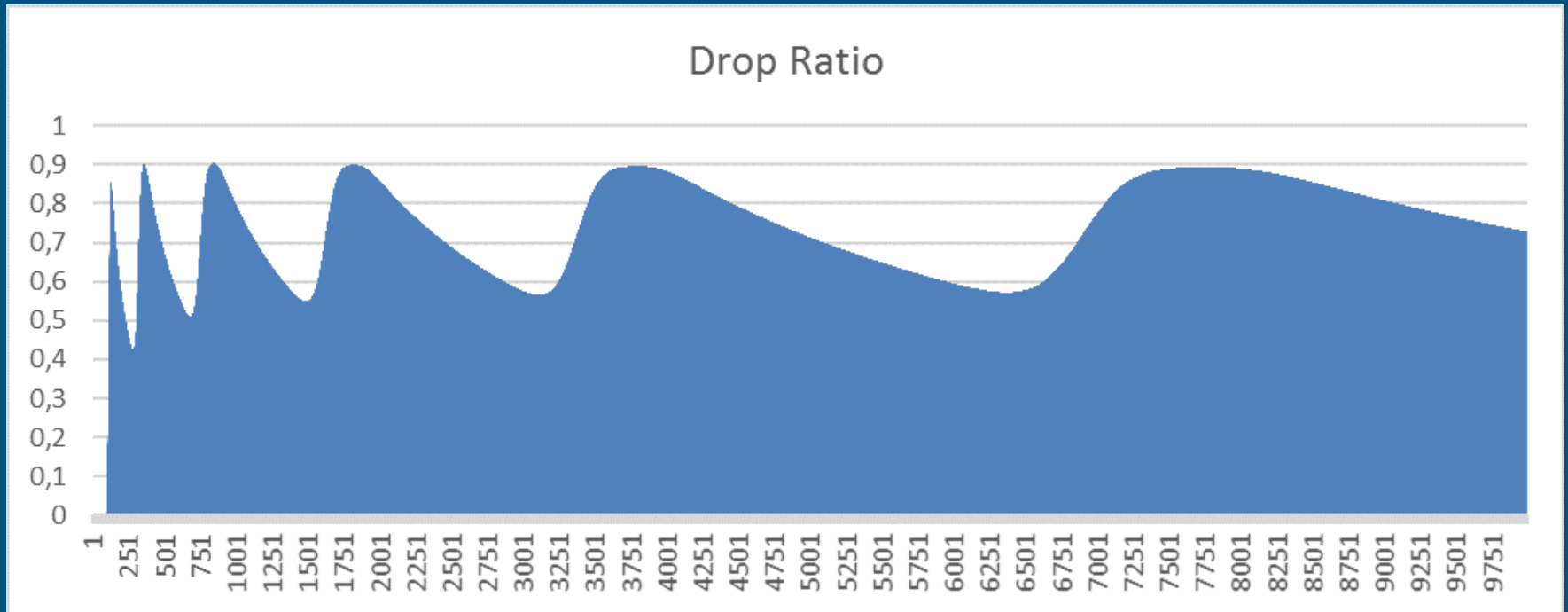
Dai calcoli effettuati il coefficiente è pari a 0,999808, il che implica forte correlazione tra il delay ed il response time del back end.

Trend temporale Abort Ratio



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

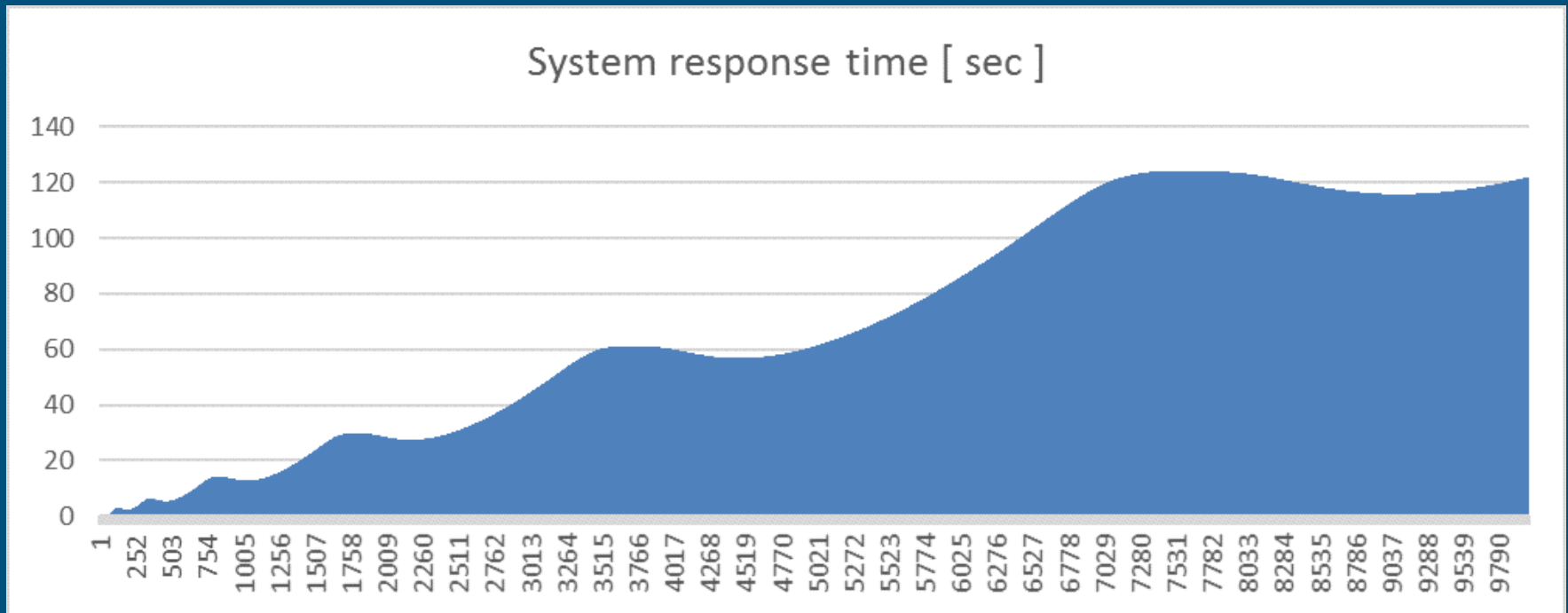
Trend temporale Drop Ratio



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Trend temporale System Response Time

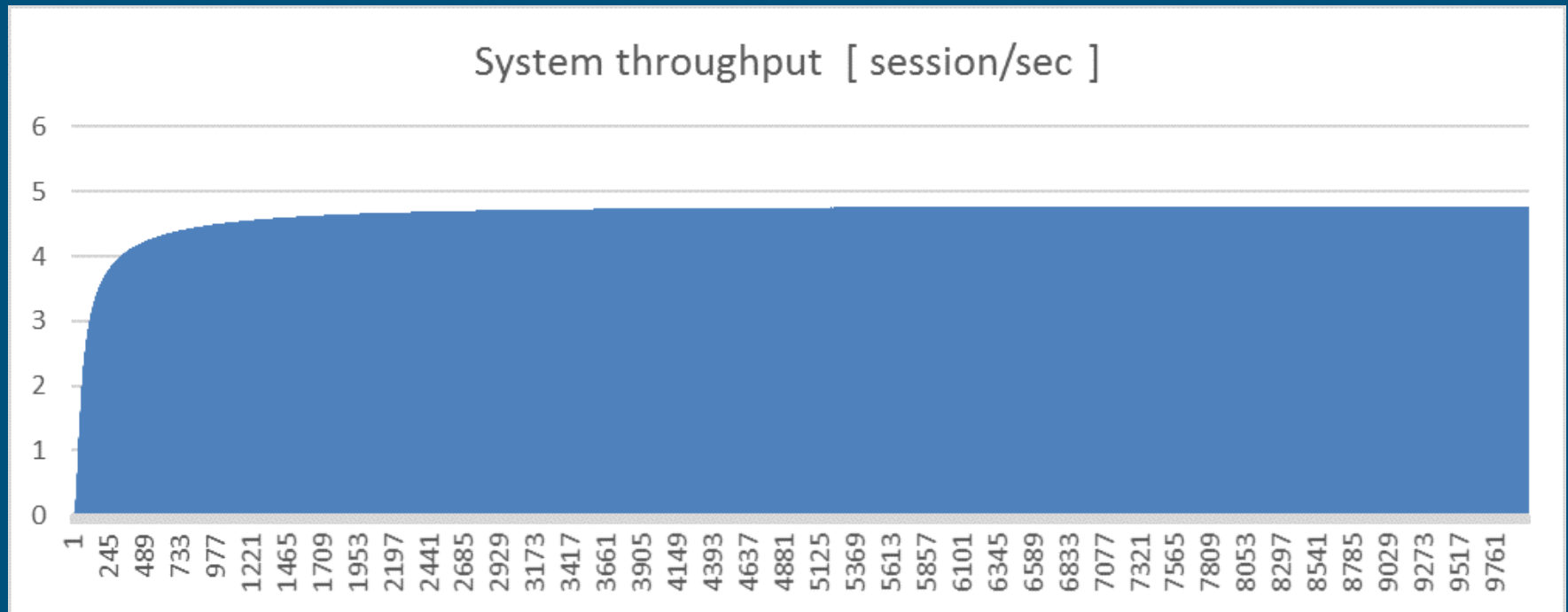
OMM attivo



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Trend temporale System Throughput

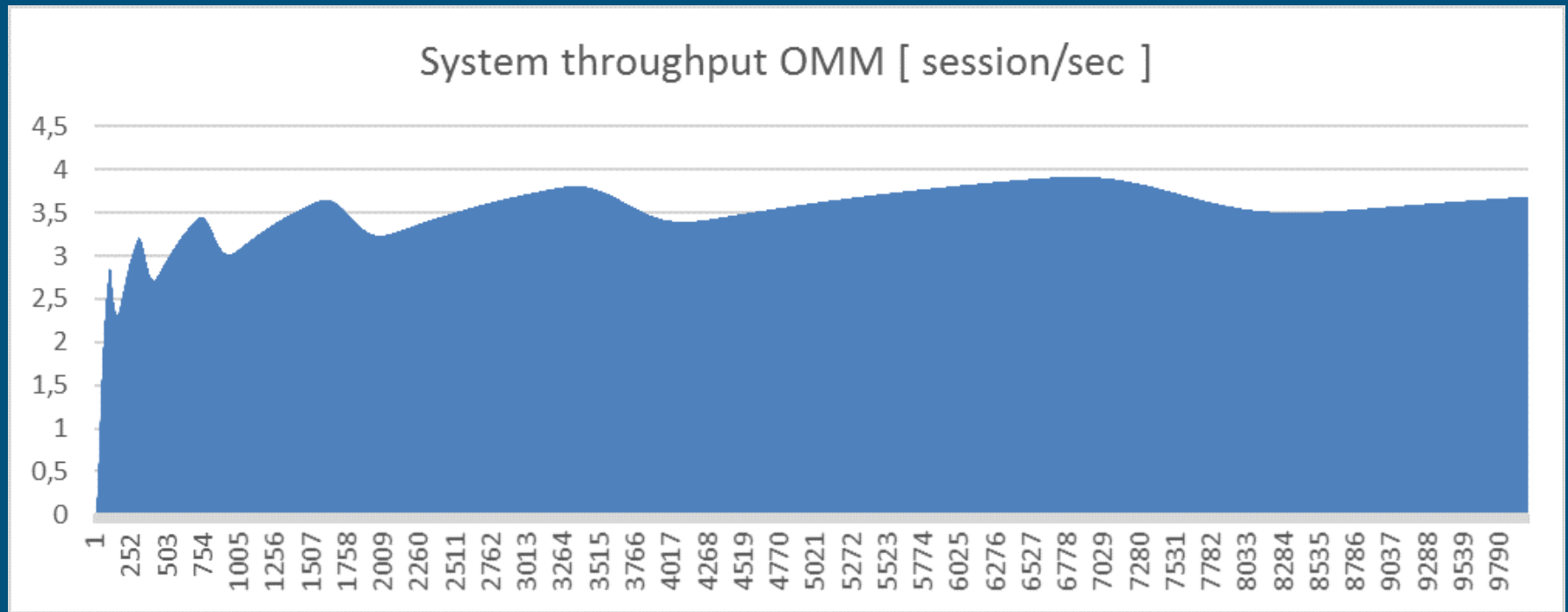
OMM disattivato



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Trend temporale System Throughput

OMM attivo



Trend mediato tra 400 simulazioni tagliate a 10000 secondi

Conclusioni

L'introduzione di un meccanismo di blocco causa dei vantaggi in termini di response time ma conseguentemente comporta un'alta percentuale di connessioni rifiutate (drop ratio 75% circa) ed una percentuale notevole di richieste rigettate (abort ratio 8% circa).

C'è da dire però che questi dati derivano dal calcolo cumulativo dell'utilizzazione, e che quindi con un calcolo intervallare non è da escludere il raggiungimento di prestazioni migliori in termini di stazionarietà del sistema.

Conclusioni (2)

Bisogna fare attenzione alla larghezza degli intervalli:

con un intervallo troppo stretto c'è rischio di stare troppo frequentemente con il meccanismo attivo e quindi di scartare tantissime connessioni ma si avrebbe una diminuzione dei tempi di attesa;

con intervalli troppo larghi si tende alla situazione attuale e quindi ad attese infinite ma con percentuale di richieste rigettate più bassa.

Intervalli di confidenza degli indici delle prestazioni

OMM attivo

Tabella 2.1: TABELLA RIASSUNTIVA DELLE METRICHE DEL SISTEMA CON OMM ALL'ISTANTE 10000 sec

SYSTEM					
RESPONSE TIME			USEFUL THROUGHPUT		
MEAN	VAR	CONF.INTERVAL	MEAN	VAR	CONF.INTERVAL
122,1798	1,445578	0,118184	3,69119	0,00165	0,003993
ABORT RATIO			DROP RATIO		
MEAN	VAR	CONF.INTERVAL	MEAN	VAR	CONF.INTERVAL
0,085226	0,000013	0,00035	0,729227	0,00073	0,002655

FRONT END					
RESPONSE TIME			THROUGHPUT		
MEAN	VAR	CONF.INTERVAL	MEAN	VAR	CONF.INTERVAL
122,1782	1,445576	0,118184	174,5794	2,70406	0,161639

Intervalli di confidenza degli indici delle prestazioni

OMM attivo

BACK END					
RESPONSE TIME			THROUGHPUT		
MEAN	VAR	CONF.INTERVAL	MEAN	VAR	CONF.INTERVAL
0,001571	0	0	174,5793	2,70406601	0,161639
DELAY					
MEAN		VAR		CONF.INTERVAL	
0,000401		0		0	

Intervalli di confidenza degli indici delle prestazioni

Tabella 2.2: **TABELLA RIASSUNTIVA DELLE METRICHE DEL SISTEMA SENZA OMM ALL'ISTANTE 10000 sec**

SYSTEM					
RESPONSE TIME			USEFUL THROUGHPUT		
MEAN	VAR	CONF.INTERVAL	MEAN	VAR	CONF.INTERVAL
601,1137	2,512203	0,155799	4,777278	0,000452	0,002089

FRONT END					
RESPONSE TIME			THROUGHPUT		
MEAN	VAR	CONF.INTERVAL	MEAN	VAR	CONF.INTERVAL
601,1122	2,512205	0,155799	218,9553	0,261108	0,0155883

Intervalli di confidenza degli indici delle prestazioni

BACK END					
RESPONSE TIME			THROUGHPUT		
MEAN	VAR	CONF.INTERVAL	MEAN	VAR	CONF.INTERVAL
0,001574	0	0	218,9953	0,026109011	0,015883
DELAY					
MEAN		VAR		CONF.INTERVAL	
0,000404		0		0	

Live simulation