Università degli Studi di Roma Tor Vergata

FACOLTÀ DI INGEGNERIA
INFORMATICA

STUDIO DELLE PRESTAZIONI DI UN'APPLICAZIONE MULTI-TIER:ANALISI DEI RISULTATI

Corso di Modelli di Prestazioni di Sistemi e Reti
Anno Accademico 2012/2013

Prof.ssa Vittoria de Nitto Personé

Emanuele Paracone
Serena Mastrogiacomo

SOMMARIO

1. INTRODUZIONE

2. RISOLUZIONE PRIMO IMPIANTO

3. RISOLUZIONE SECONDO IMPIANTO

INTRODUZIONE

OBIETTIVO: realizzazione e risoluzione di un modello analitico per lo studio delle prestazioni di un'applicazione multi-tier costituite da:

- un web server
- un application server
- un back-end server

STUDIO:
tramite modello a reti di code chiuso con soluzione analitica tramite approccio

FORMA PRODOTTO

1 . Risoluzione primo modello tramite Gordon&Newell
2. Risoluzione del secondo modello tramite algoritmo MVA

INTRODUZIONE (2)

CAC: meccanismo per il controllo delle ammissioni delle sessioni; in caso di picco di traffico deve essere in grado di selezionare un sottoinsieme delle sessioni che il sistema è in grado di completare

la scelta delle sessioni da accettare deve avvenire in maniera tale da ottimizzare una *metrica* desiderata

nel nostro caso attività di ***PERFORMANCE PREDICTION***

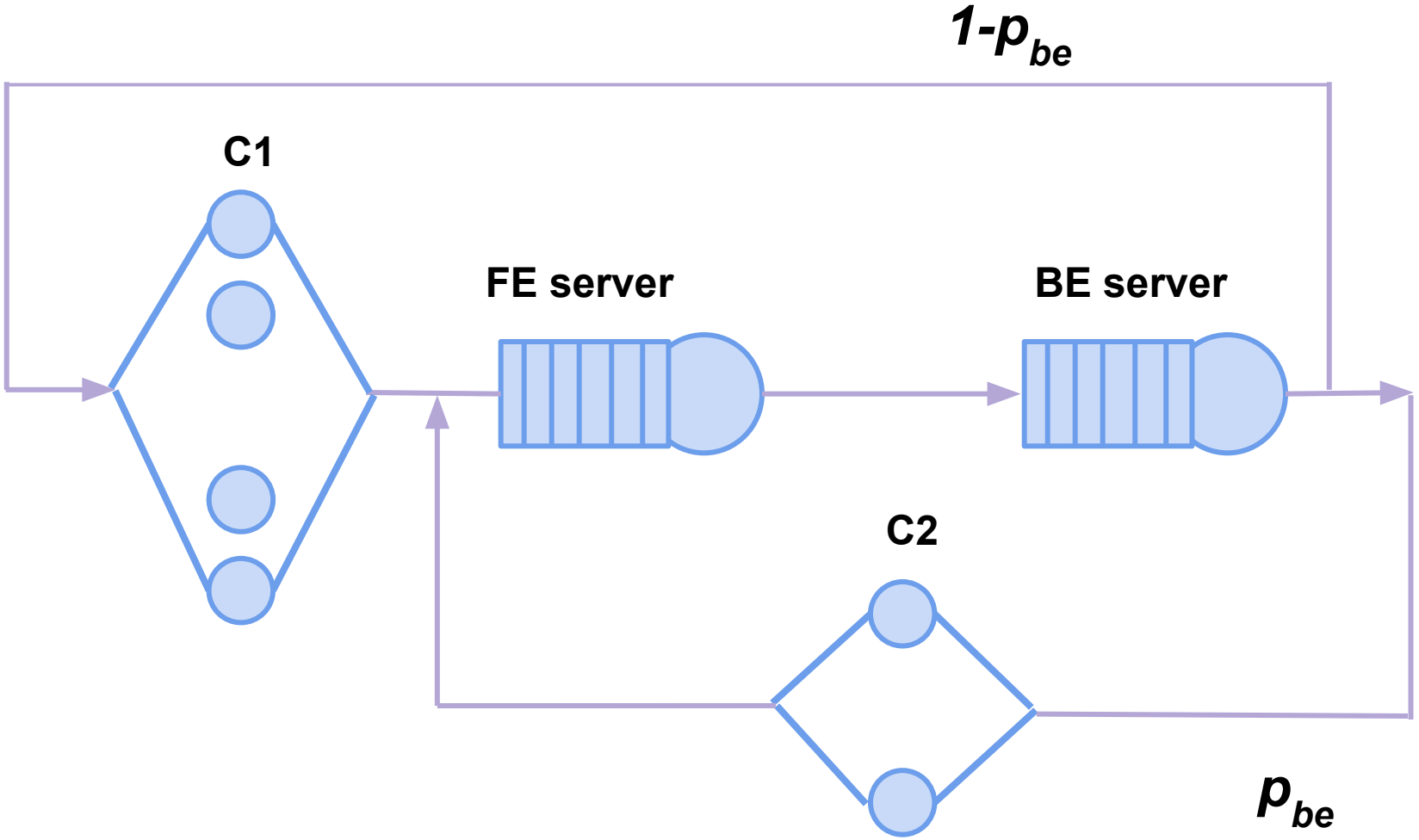
- *stima del **Tempo di risposta** che intercorre tra il primo tentativo di accesso al sistema e il completamento dell'intera sessione*.
- stima di ***indici di prestazioni locali e globali***

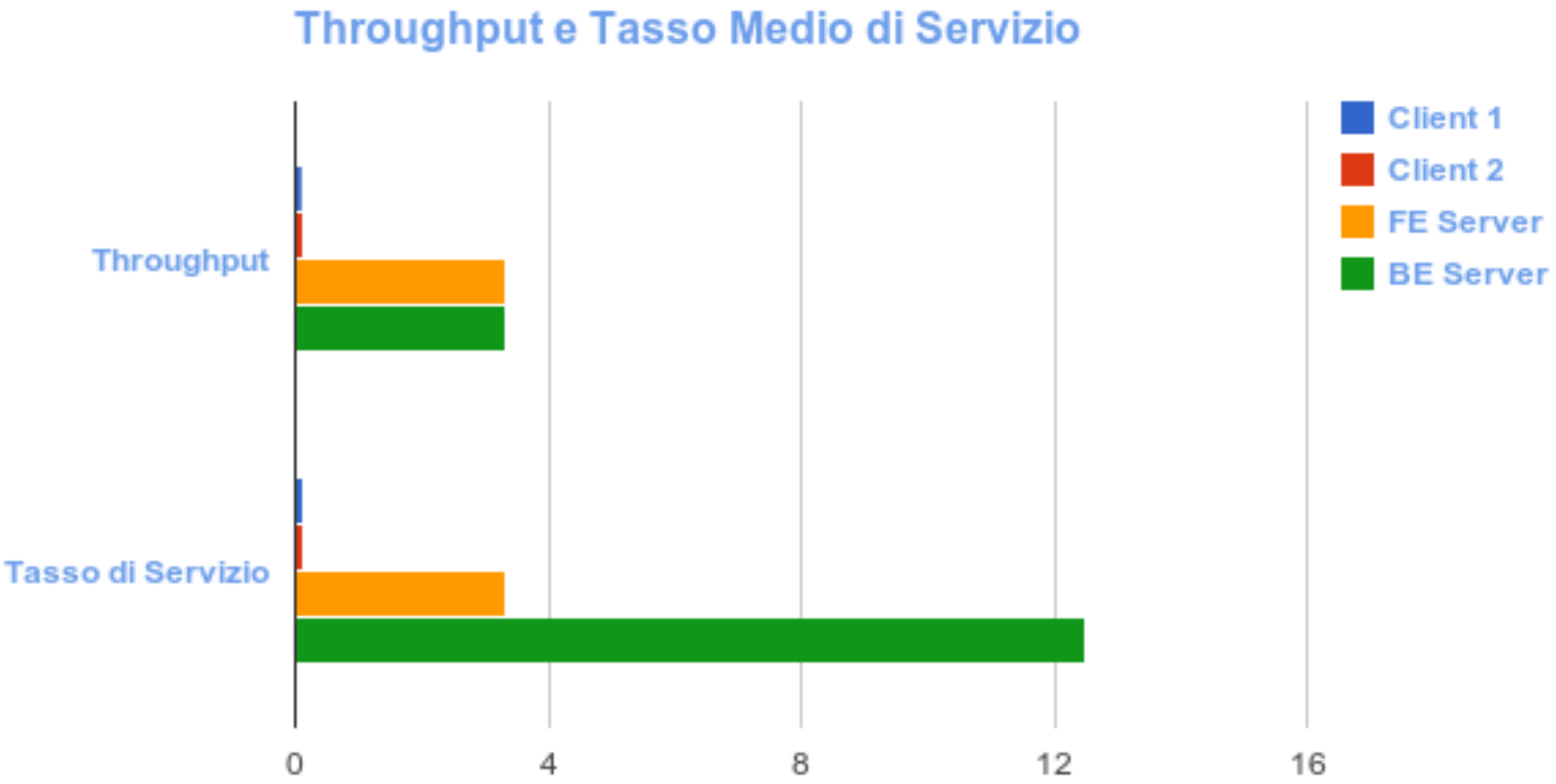
ANALISI DEI RISULTATI:SOLUZIONE PRIMO IMPIANTO

FOCUS ON: indici locali e globali ottenuti per il modello di impianto quando nel sistema sono presenti 50 utenti

- risultati riguardo la probabilità di rifiuto di un job in funzione della soglia **S** definita sulla popolazione presente presso i centri **Fe Server** e **Be Server.**

Gordon & Newell



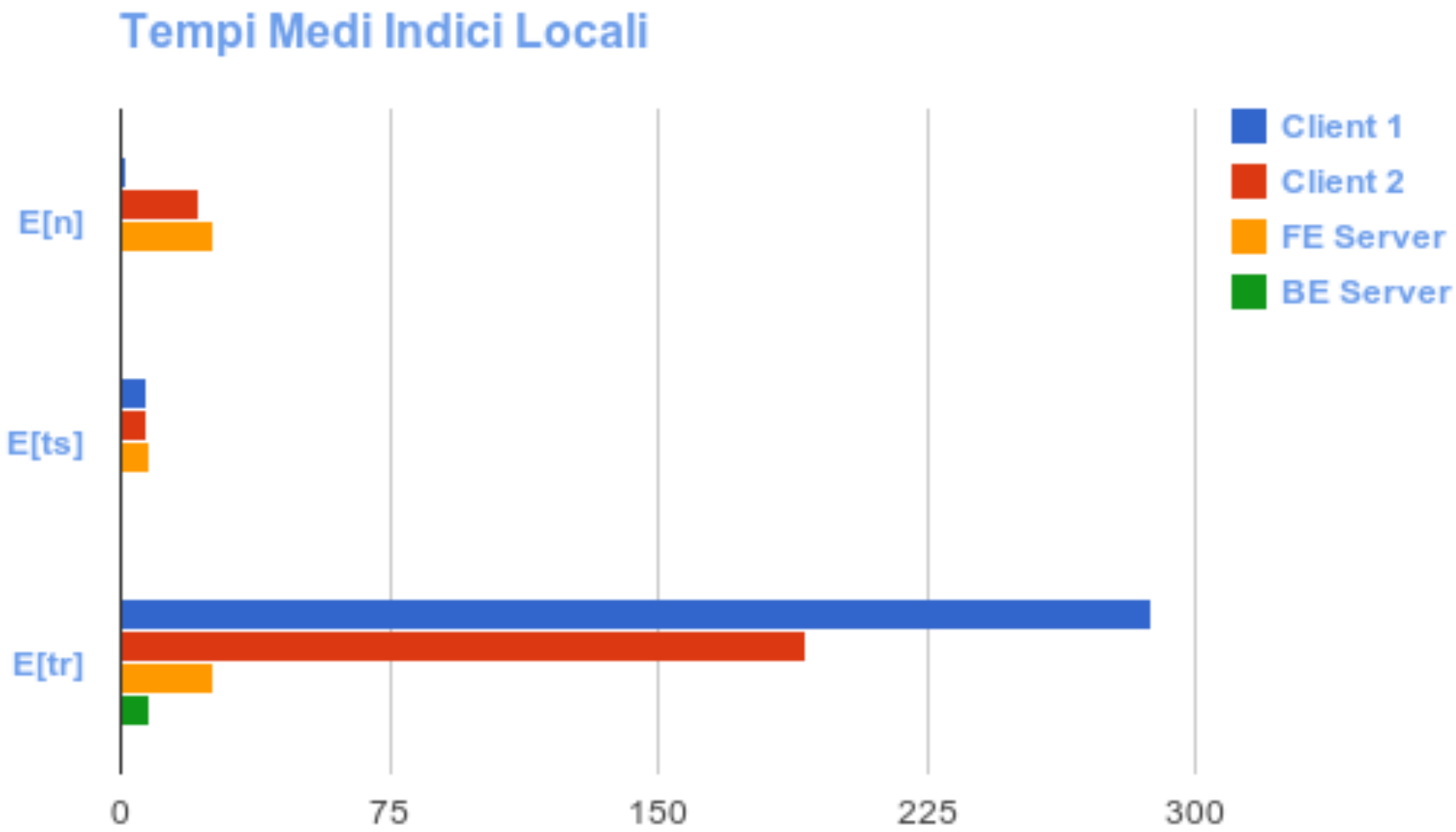


Il **throughput** dei due server è lo stesso in quanto entrambi subuscono lo stesso flusso di utenti da eseguire

Il **tasso medio di servizio** cambia radicalmente tra i due centri in quanto il tempo medio di servizio del BE server (0.08s) è nettamente inferiore rispetto a quello che caratterizza l'FE server(0.3s)

Congestione sistema server: utilizzazione FE server=1

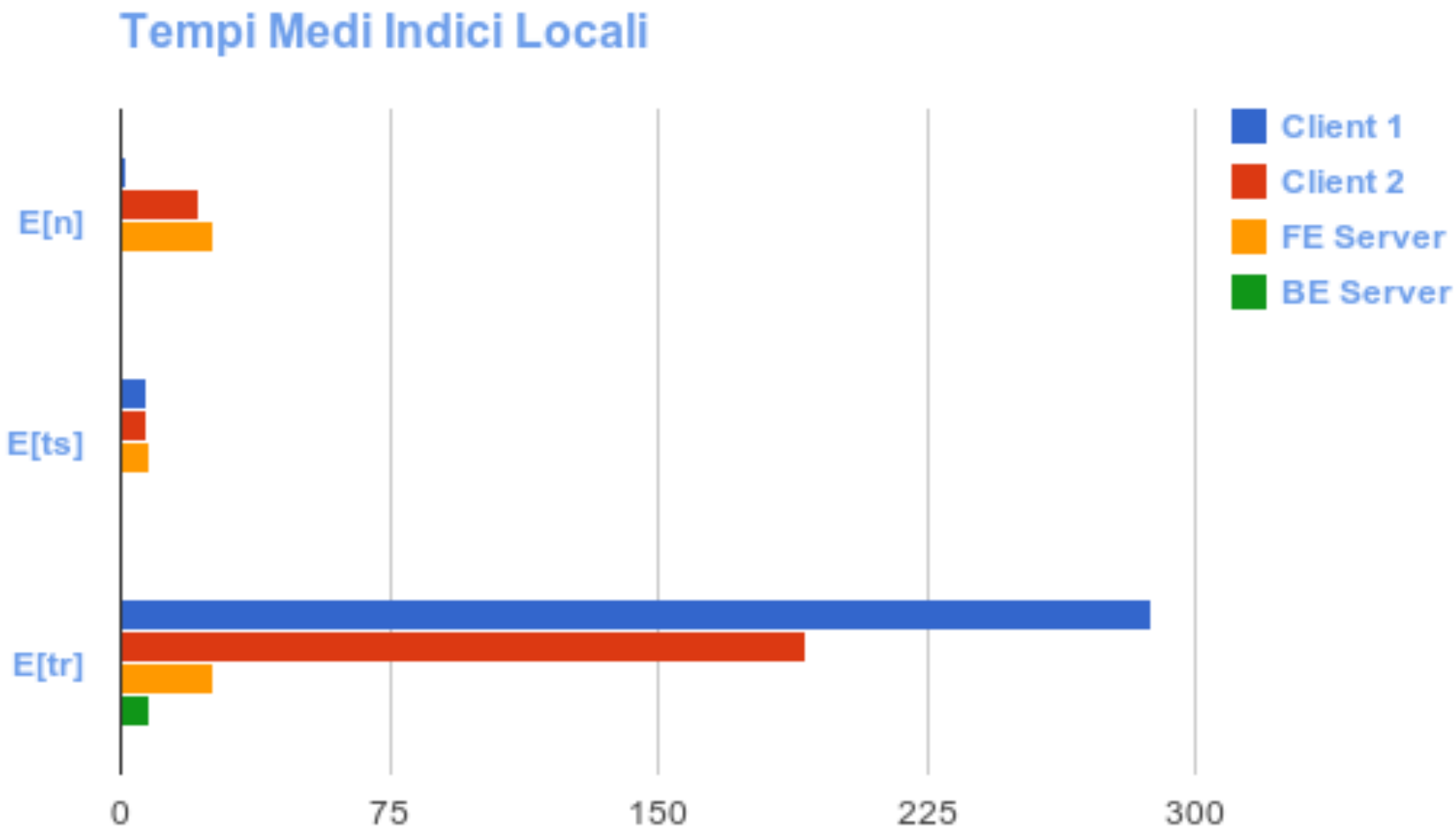
	Client 1	Client 2	FE Server	BE Server
Throughput	0.09624571579462717907	0.14285714282494874072	3.33332998587726336837	3.33332998867726937320
Tasso di Servizio	0.14285714285714284921	0.14285714285714284921	3.33333333333333348136	12.500000000000000000
Fattore di utilizzazione	-	-	0.99999899576317896610	0.26666639909418154986
E[n]	1.11999891897951897590	22.21331102696138515284	26.30305449304594489490	0.36363556101322219716
E[tq]	7.0	7.0	7.890924272270489758 80	0.10909077776530606840
E[tr]	287.4589255782569807706 7	191.3828704954807449212 2	26.30305449304594489490	8.39304077256530867146



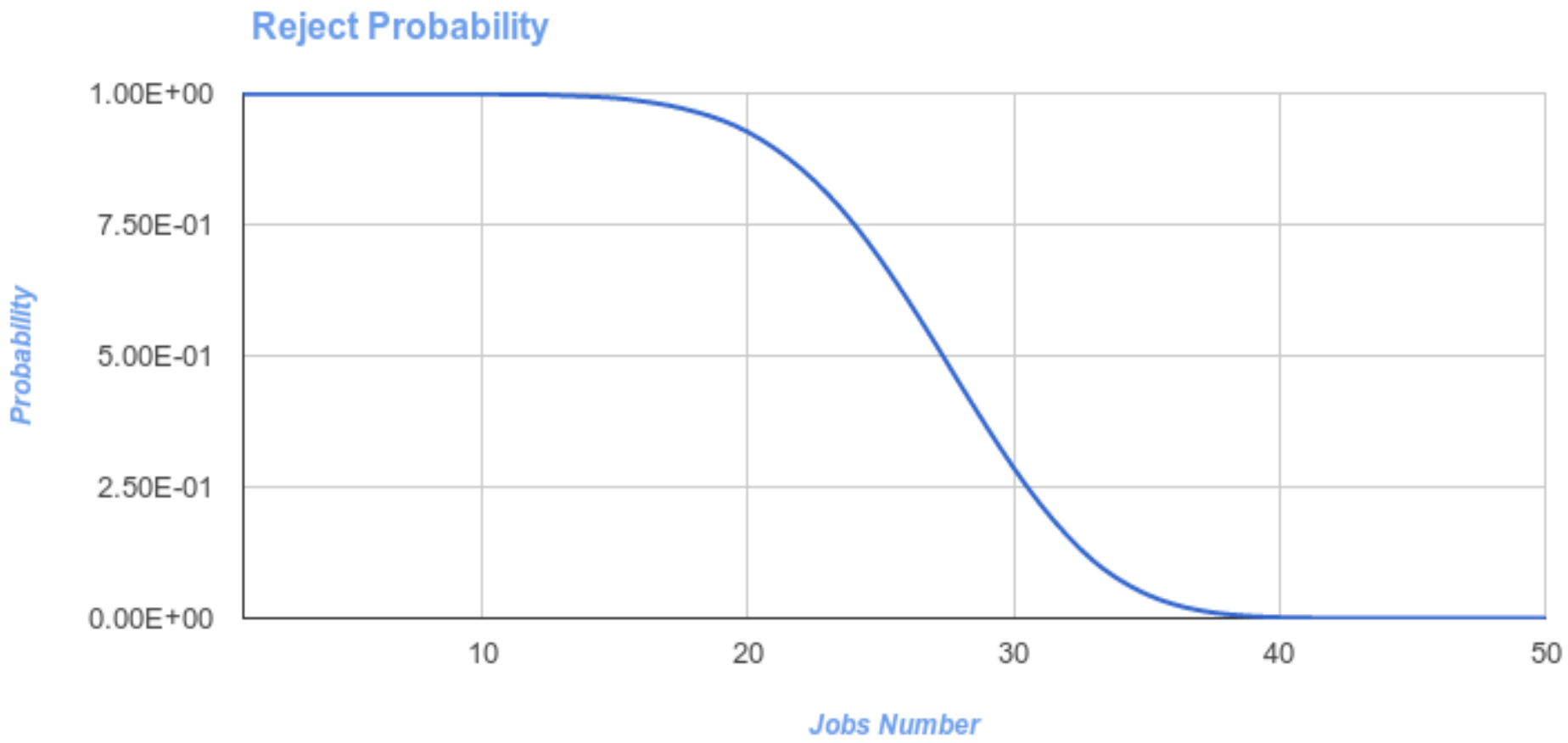
La popolazione del sistema, quando sono presenti 50 jobs, si distribuisce prevalentemente tra il **FE Server** e il **Client 2**:

- l'accumulo di jobs presso FE Server è facilmente spiegabile a partire dal fattore di utilizzazione praticamente pari a 1: la coda del centro è sempre piena, in quanto il tasso di arrivi è pari al tasso di smaltimento

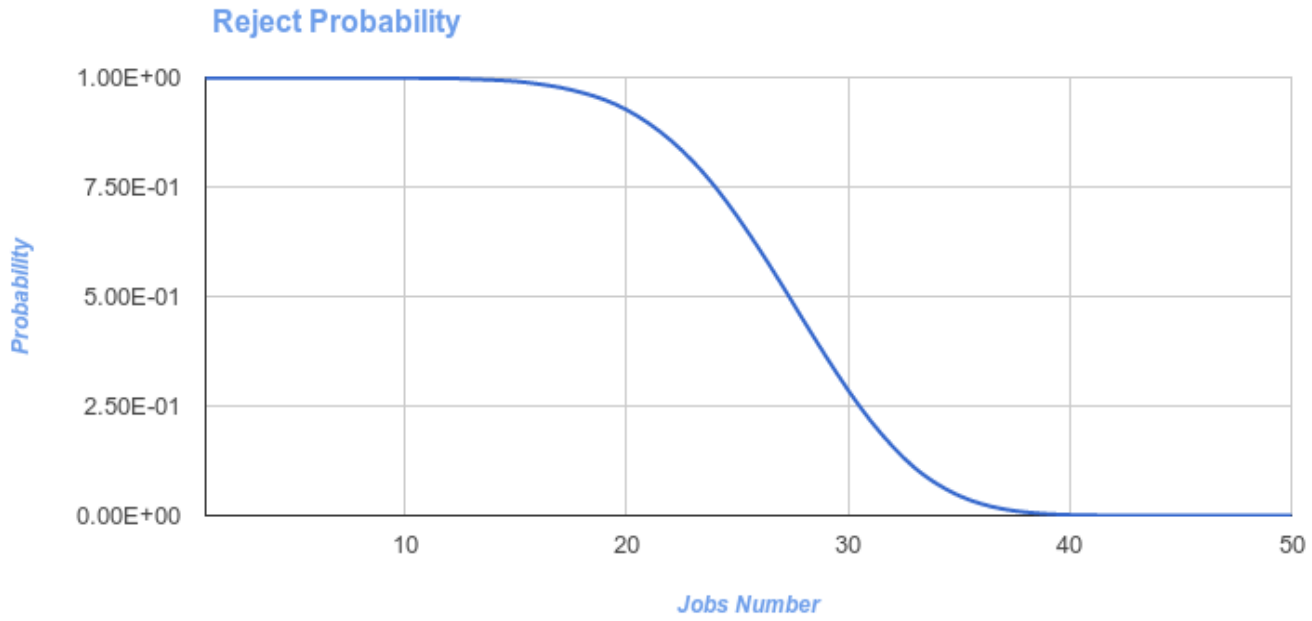
- presso Client 2 i jobs sperimentano il ritardo dovuto al tempo di think imposto dall'utente che prende visione dell'output interno ad una sessione: tale tempo è mediamente lungo (7 sec) rispetto all'ordine di grandezza dei tempi di servizio dei server (decimi di secondo).



Il tempo medio di risposta dell'impianto sperimentato dai singoli centri varia sensibilmente in funzione delle probabilità di routing. Ne consegue che i jobs provenienti dal centro con probabilità di routing più sfavorevoli, ovvero il **Client 1**, faranno ritorno allo stesso centro mediamente più tardi rispetto a quanto farebbero presso gli altri centri.



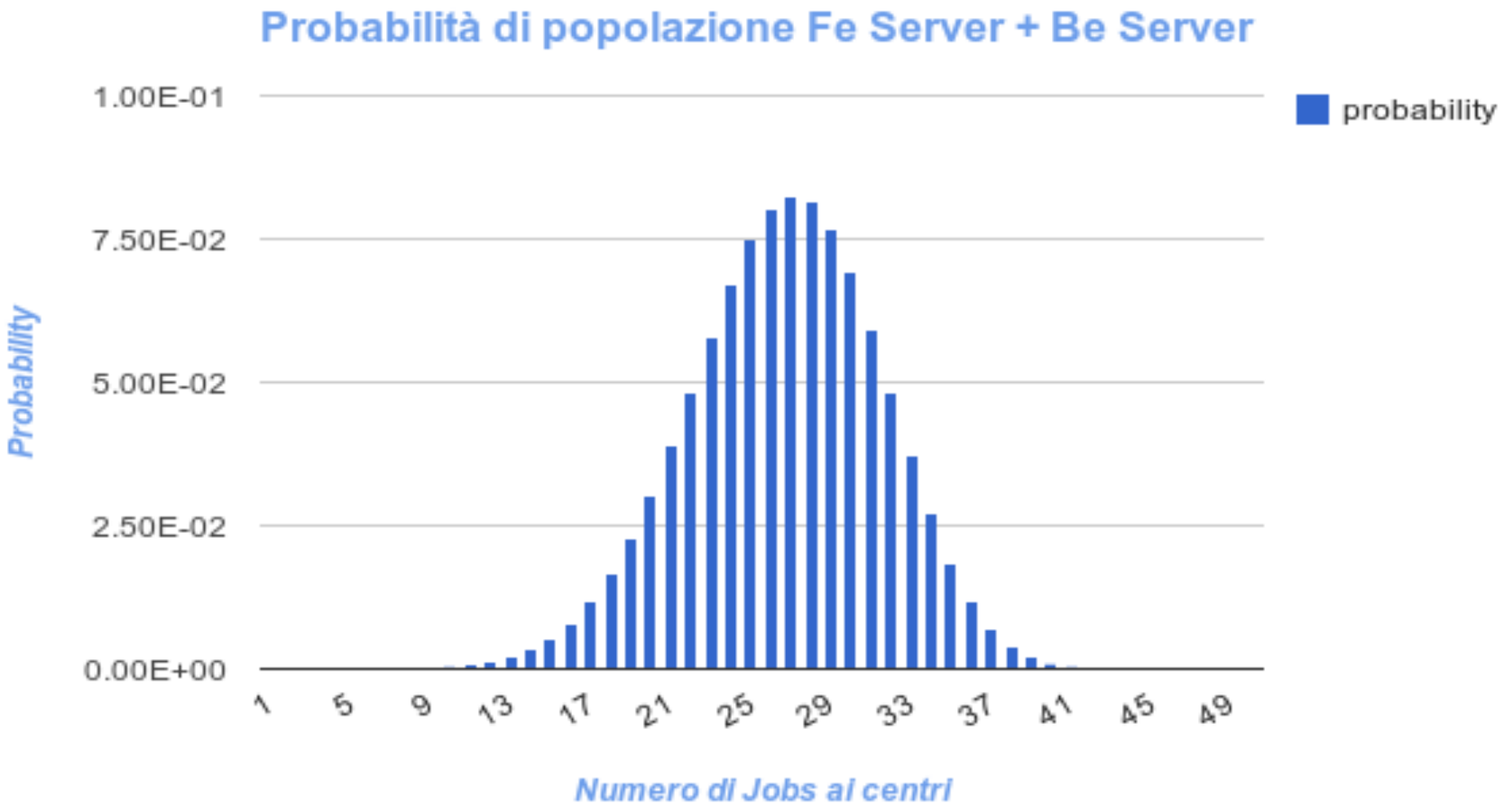
andamento della probabilità di saturazione al variare del numero di sessioni S ammissibili nel sistema. Il numero di job considerato è pari a 50, mentre il numero di sessioni varia da 1 a 50.



La figura mostra come all'aumentare della soglia S, e conseguentemente del numero di utenti ammessi nel sistema, la probabilità di saturazione tende a ridursi. La spiegazione di questo fenomeno può essere ricavata nella modalità stessa con cui viene calcolata la probabilità di saturazione: essa infatti corrisponde alla sommatoria delle probabilità di stato di tutti gli stati di saturazione, ovvero di quegli stati la cui somma di job nei centri FE+APP Server e BE Server è maggiore o uguale ad S. Di conseguenza, un aumento della soglia S corrisponde ad una ***diminuzione della cardinalità dell'insieme di stati di saturazione***, riducendo quindi il valore della sommatoria nel calcolo della probabilità di saturazione.

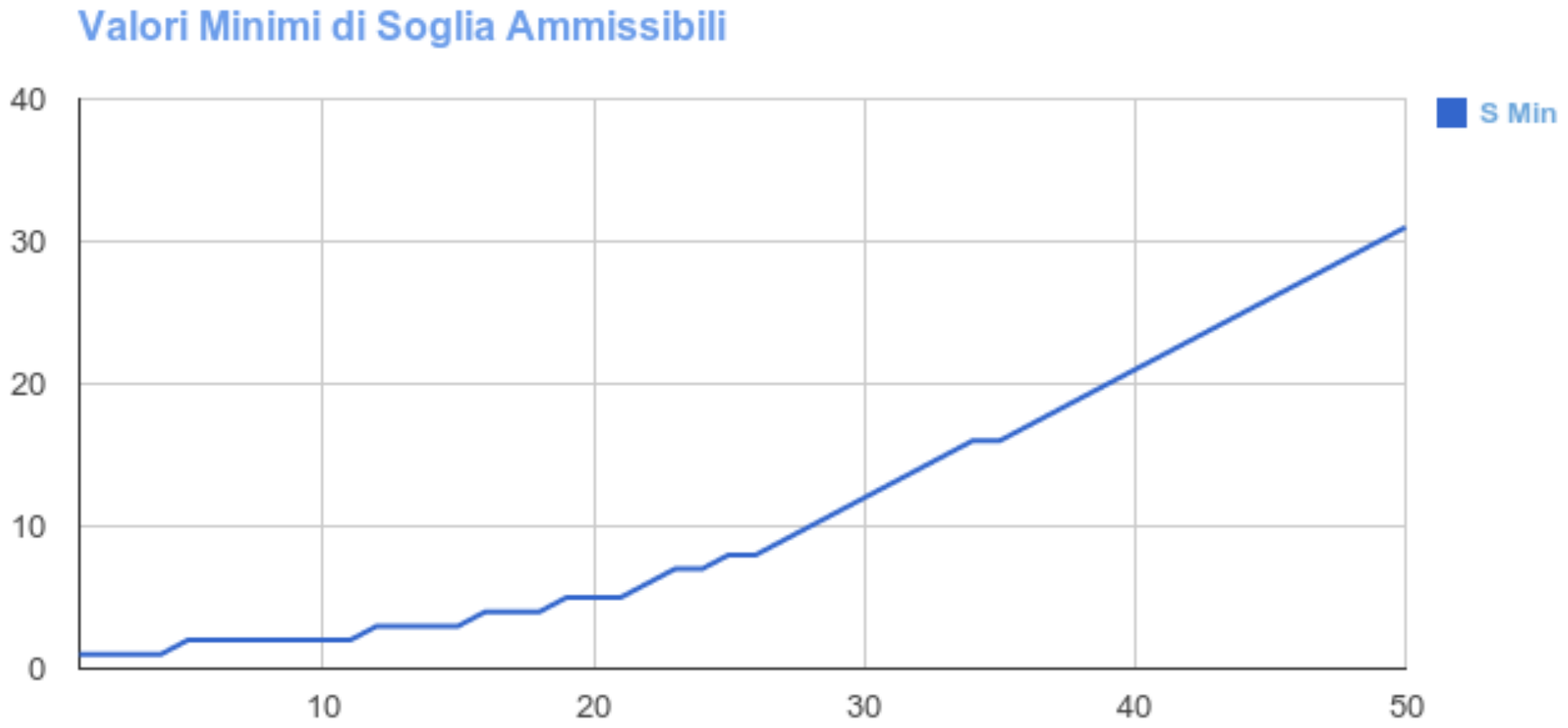
La spiegazione teorica a questo fenomeno si fonda sulla modalità con cui viene calcolata la probalità di saturazione: essa coincide con la somma delle probabilità di stato di tutti gli stati di saturazione

(stati in cui la somma dei job presenti nei centri ***FE server*** e ***BE server*** è ***maggiore o uguale a S***). Ovviamente, ad un aumento del valore di S corrisponde ***una riduzione della cardinalità dell'insieme di stati di saturazione***, con la conseguente diminuzione del valore della somma delle probabilità di stato.



Popolazioni presso i centri ***Fe Server*** e ***Be Server***
in funzione del numero di utenti presenti nel sistema

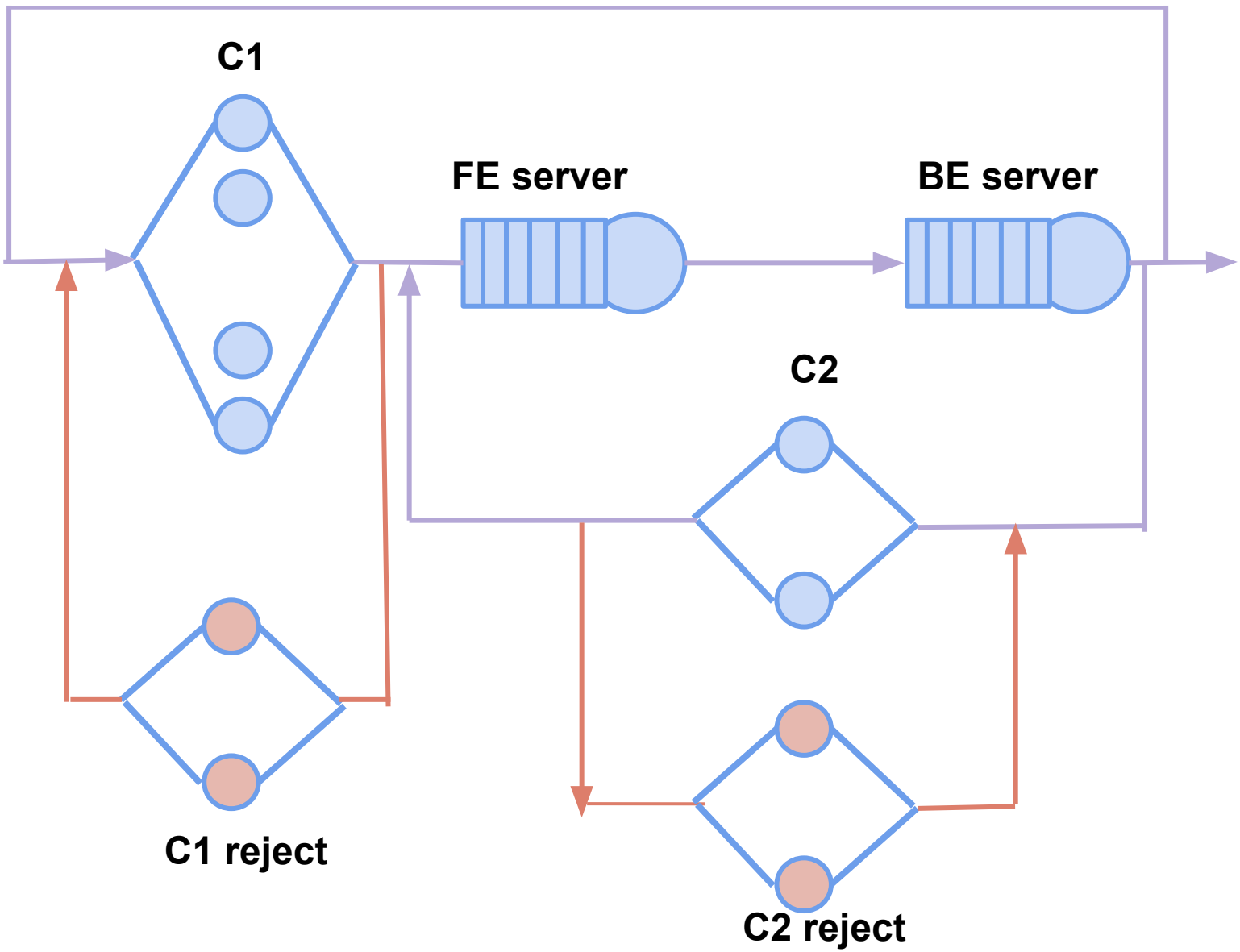
Per rispettare quanto richiesto dalla ***QoS*** dell'impianto, è possibile assumere solo quei valori di soglia S che impongono una probabilità di rifiuto non superiore a 0.22, ovvero esistono in funzione di N dei valori di S sotto ai quali non è possibile andare senza violare il requisito suddetto. Tali valori sono illustrati nella figura seguente per ***N=50***:



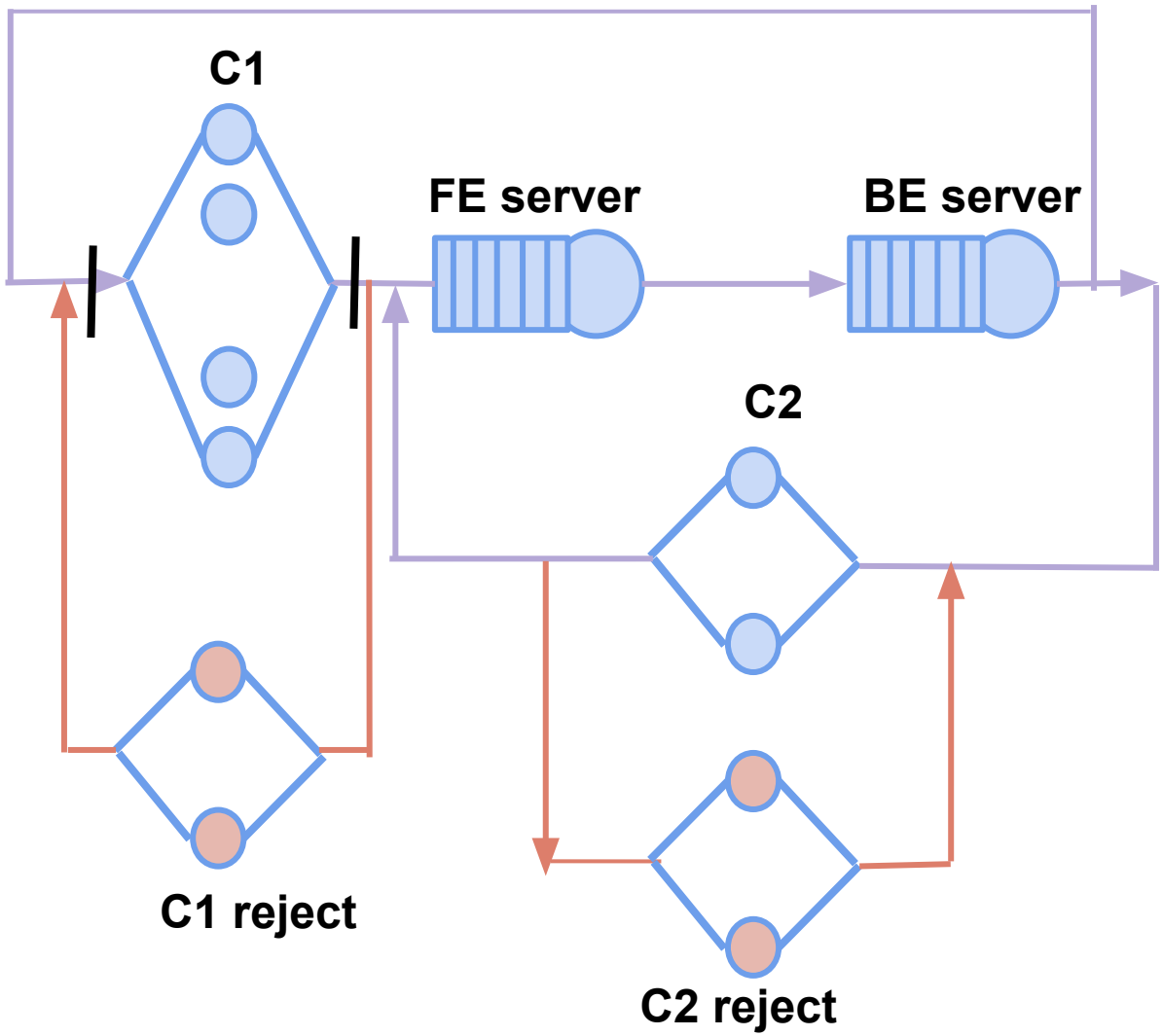
ANALISI DEI RISULTATI:SOLUZIONE SECONDO IMPIANTO

Per parametrizzare il secondo impianto, abbiamo ottenuto i valori per le soglie **S1** e **S2** che minimizzino il tempo di risposta del sistema sperimentato dai job del **Client 1** sviluppando un nuovo modello, che da ora in avanti chiameremo **Modello di Supporto SM** (**Support Model**).

MVA: SCELTE MODELLISTICHE



Calcolo del Tempo di Risposta del Sistema



Come calcolare il tempo di risposta del sistema?

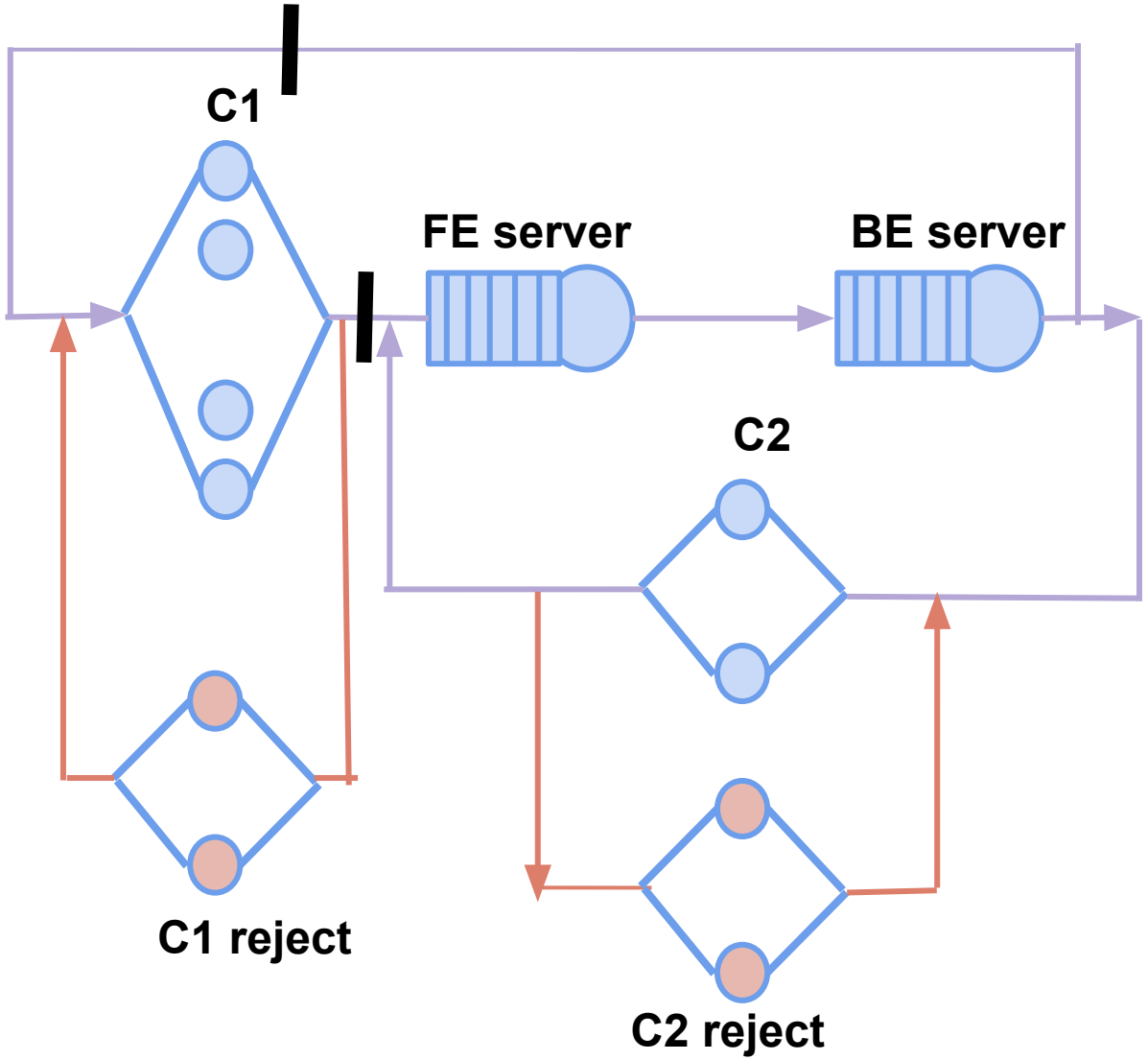
Prima opzione: Calcolo del tempo di risposta del sistema per i job in uscita dal **Client 1**.

Considerazione:

Il flusso di job in entrata presso **Client 1** è pari al flusso proveniente da **BE Server** e da **Client 1 rej**.

Viene considerato come tempo di risposta del sistema per **Client1** anche il tempo della sola rejection sperimentato dai jobs che seguono il routing
Client1 -> Client1 rej -> Client1

Calcolo del Tempo di Risposta del Sistema



Come calcolare il tempo di risposta del sistema?

Seconda opzione: Calcolo del tempo di risposta del sistema per i job in uscita dal blocco
Client 1 - **Client 1 rej**.

Considerazioni:

A. Il flusso di job in uscita dal blocco **C1 - C1 rej** è pari al *throughput* di **Client 1** meno il *throughput* di **Client 1 rej**

B. Il tempo di risposta del sistema per i job in uscita dal blocco degrada all'aumentare dei jobs presenti nel sistema

Per le osservazioni A e B, il tempo di risposta migliora al diminuire del flusso in uscita dal blocco **Client 1** - **Client 1 rej**, ovvero per minimizzare il tempo di risposta del sistema devono essere rifiutati quanti più jobs provenienti da **Client 1**.

Calcolo del Tempo di Risposta del Sistema

Col presente lavoro, ci poniamo l'obiettivo di minimizzare il tempo di risposta del sistema sperimentato da un utente generico che voglia farvi accesso.

Giustifichiamo la nostra interpretazione dei requisiti alla luce di uno studio di *Akamai Technologies*, fornitore della piattaforma *Akamai* per il Content Delivery Network:

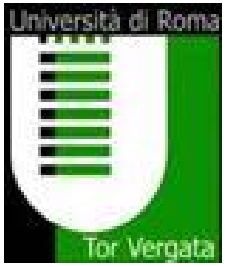
Akamai, "Boosting Online Commerce Profitability with Akamai," Akamai Technologies, 2007, http://www.akamai.com (May 30, 2008). Based on the finding that 30% to 50% of transactions above the 4-second threshold bail out, Akamai estimated that by reducing the percentage of transactions above this threshold from 40% to 10%, conversion rates will improve by 9 to 15%.

Calcolo del Tempo di Risposta del Sistema

Col presente lavoro, ci poniamo l'obiettivo di minimizzare il tempo di risposta del sistema sperimentato da un utente generico che voglia farvi accesso.

Alla luce dello studio di *Akamai*, l'adozione di un meccanismo CaC che minimizzi il tempo di risposta sperimentato da un utente aumenta il valore commerciale dell'impianto!

Calcolo del Tempo di Risposta del Sistema



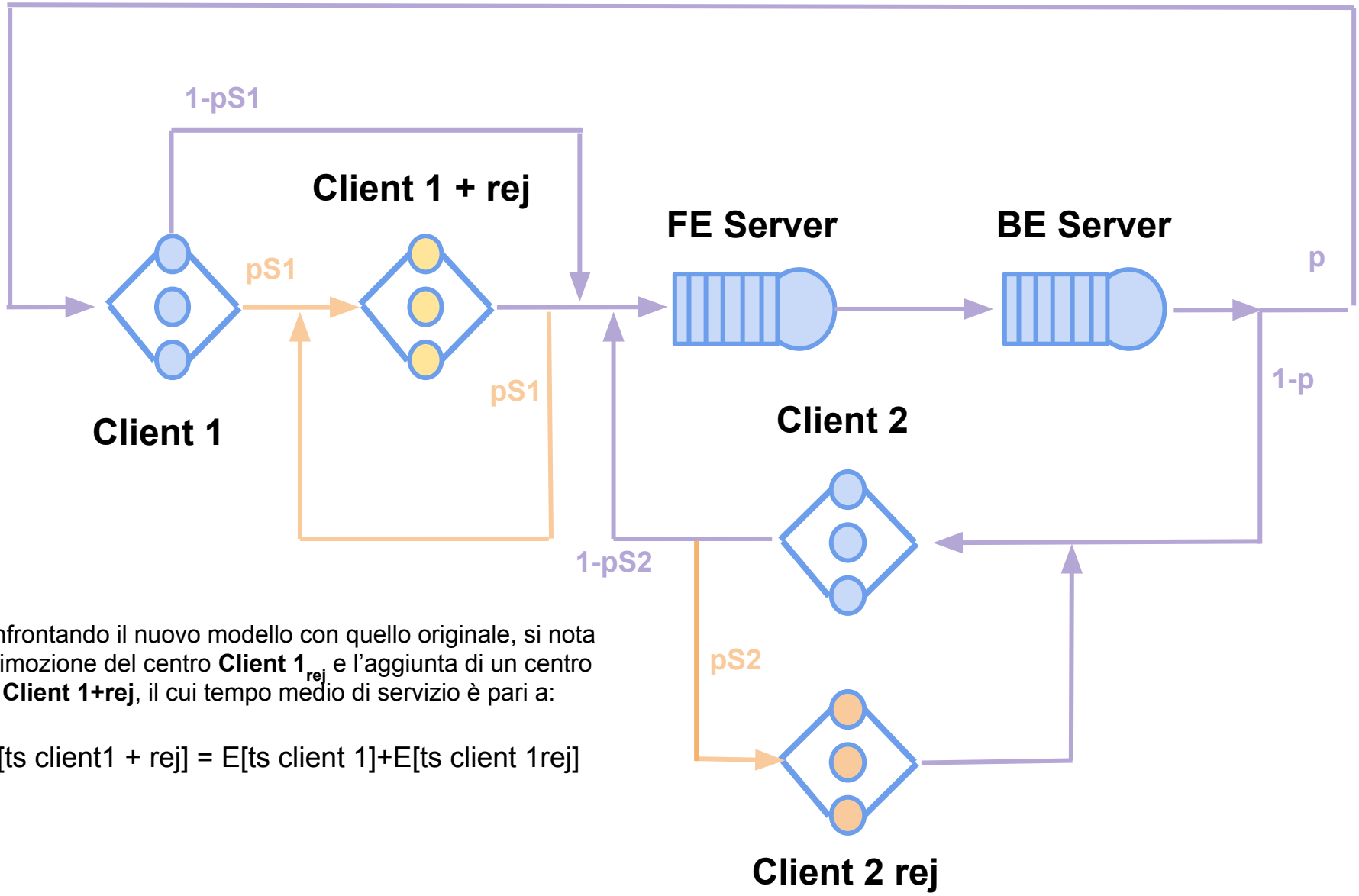
Col presente lavoro, ci poniamo l'obiettivo di minimizzare il tempo di risposta del sistema sperimentato da un utente generico che voglia farvi accesso.

Osservando il modello, per quanto detto, notiamo:

A. di non poter considerare come completamento di sessione i job che "tornano" presso il **Client 1** a causa della rejection imposta dal vincolo **S1** sul carico del sistema;

B. di dover tenere conto del ritardo introdotto dalle possibili rejection della prima connessione, comprensivo del tempo di elaborazione della rejection (ovvero il tempo di esecuzione di **Client 1 rej**) e del nuovo tempo di think presso il **Client 1**.

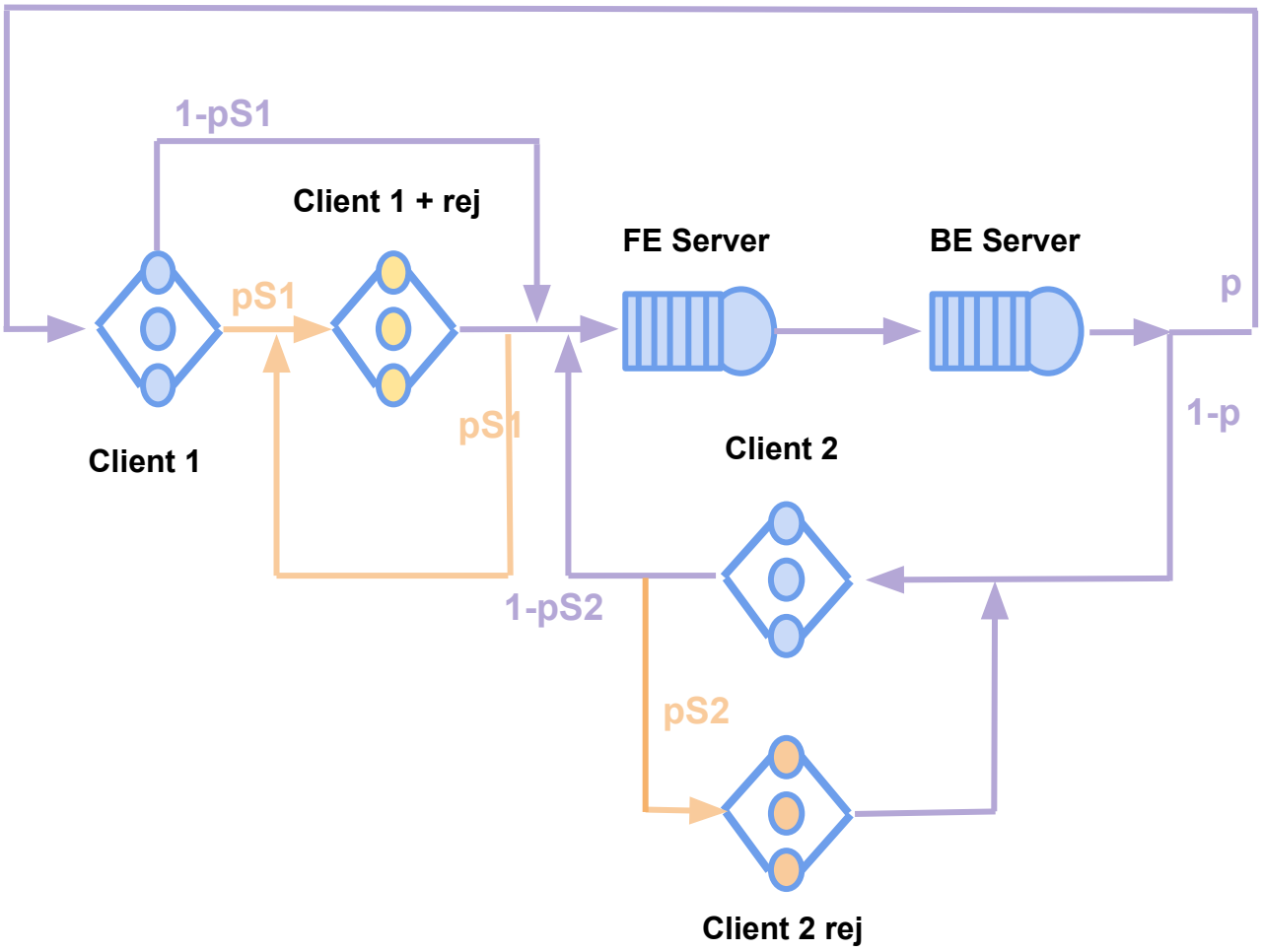
Modello di Supporto - SM



Confrontando il nuovo modello con quello originale, si nota la rimozione del centro **Client 1**$_{rej}$ e l'aggiunta di un centro **Client 1+rej**, il cui tempo medio di servizio è pari a:

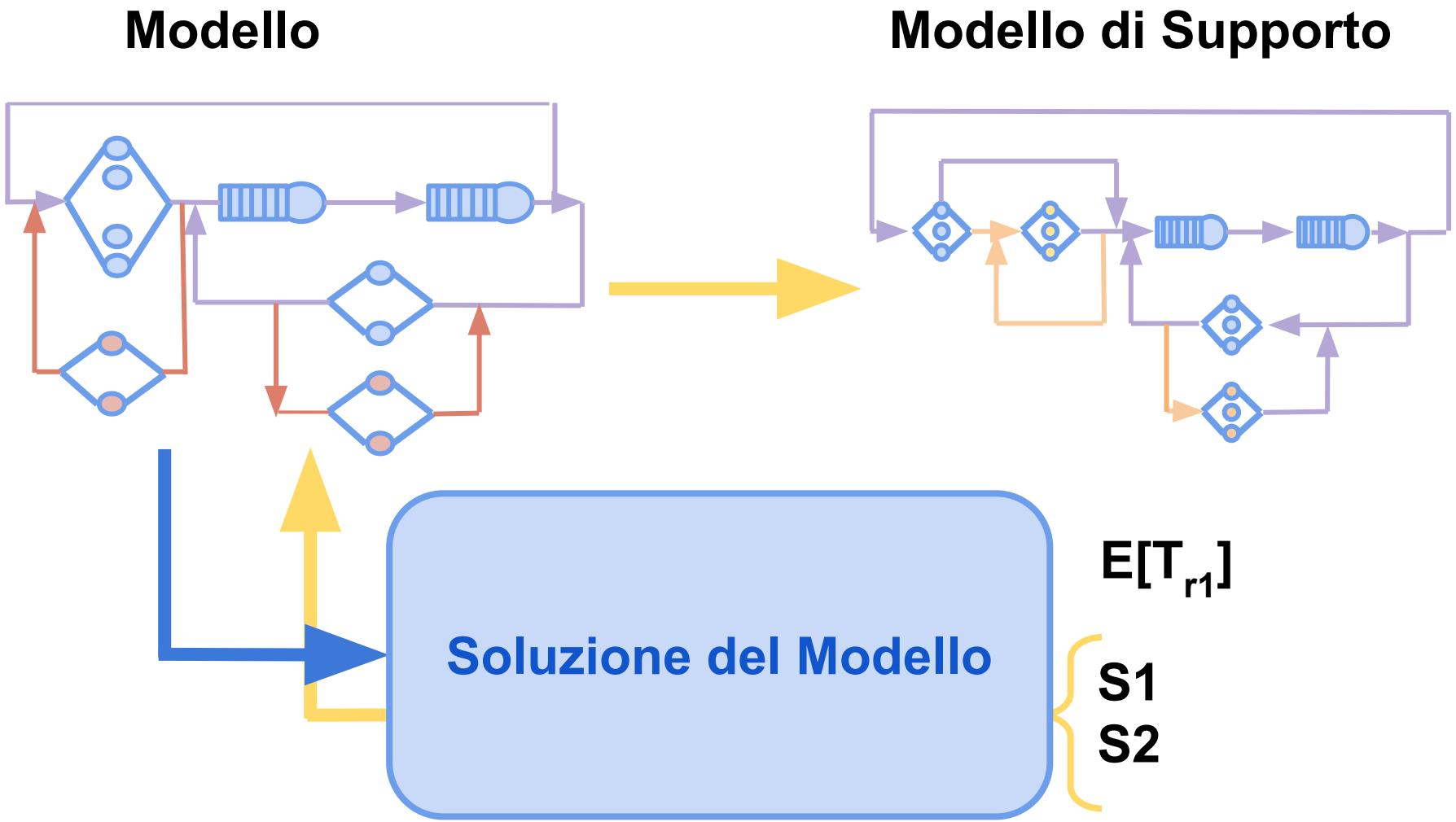
E[ts client1 + rej] = E[ts client 1]+E[ts client 1rej]

Modello di Supporto - SM

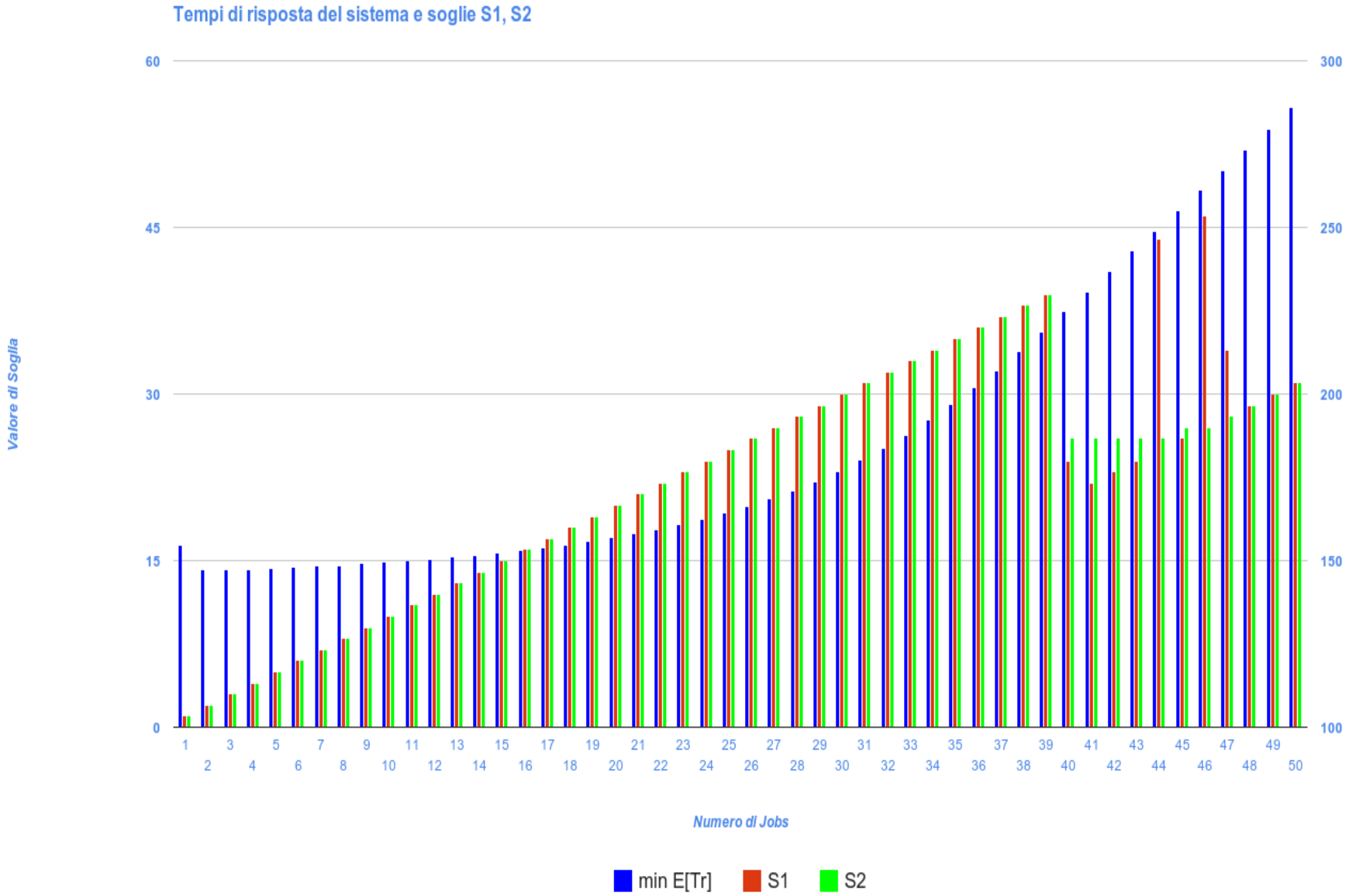


Il nuovo modello definisce il seguente routing al flusso dei jobs:

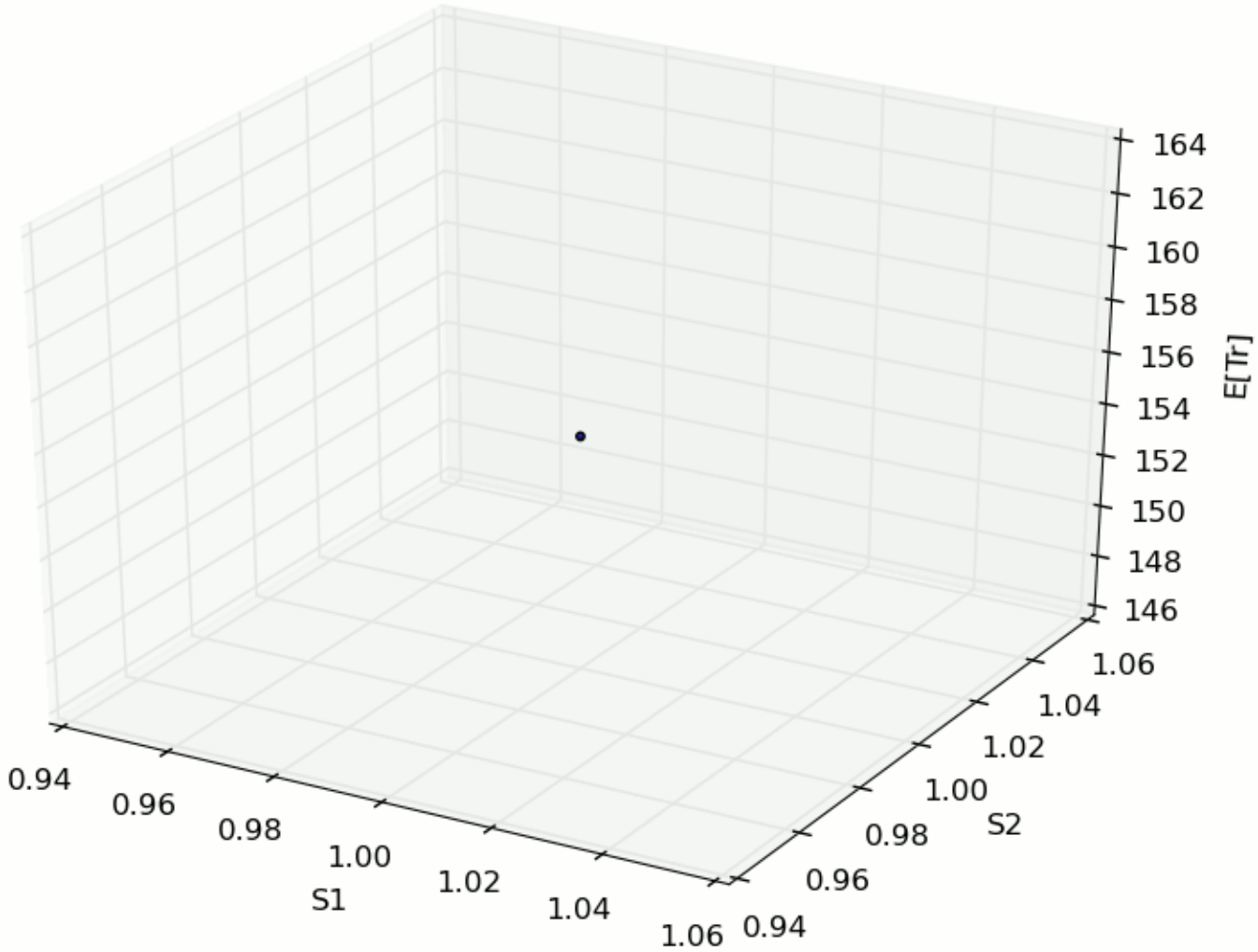
- una richiesta proveniente dal **Client 1** accede al sistema server con probabilità 1-$\mathbf{p_{S1}}$ (dove $\mathbf{p_{S1}}$ è la probabilità di rejection imposta dalla soglia **S1**);
- una richiesta proveniente dal **Client 1** accede al centro **Client 1+rej** con probabilità $\mathbf{p_{S1}}$;
- una richiesta proveniente dal **Client 1+rej** accede al sistema server con probabilità 1-$\mathbf{p_{S1}}$;
- una richiesta proveniente dal **Client 1+rej** fa ritorno al centro **Client 1+rej** con probabilità $\mathbf{p_{S1}}$;
- solo job provenienti dal **BE Server** possono fare ritorno a **Client 1**.


Modello
Modello di Supporto
Soluzione del Modello
E[T_r1]
S1
S2

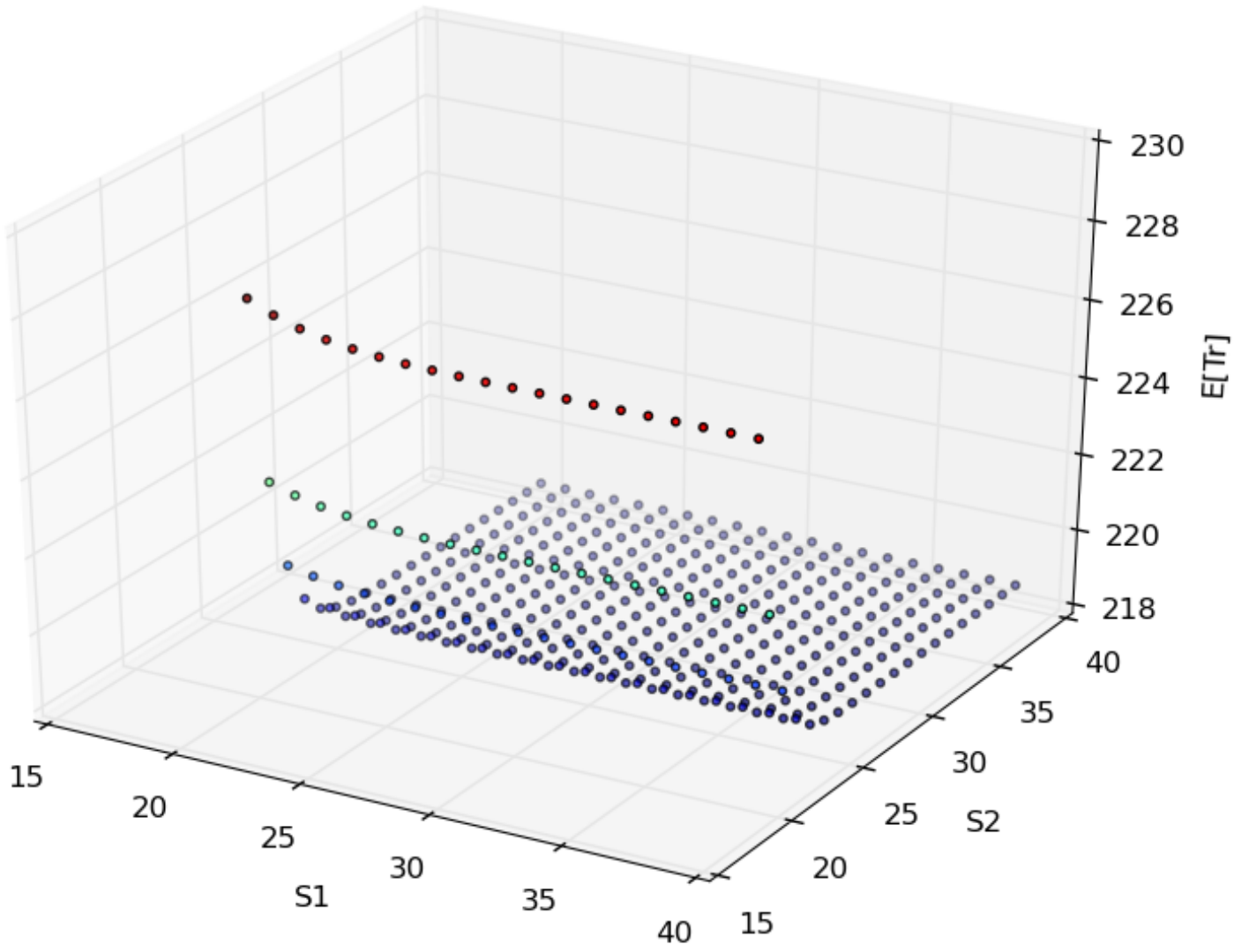
Tempi di Risposta del Sistema al variare di N



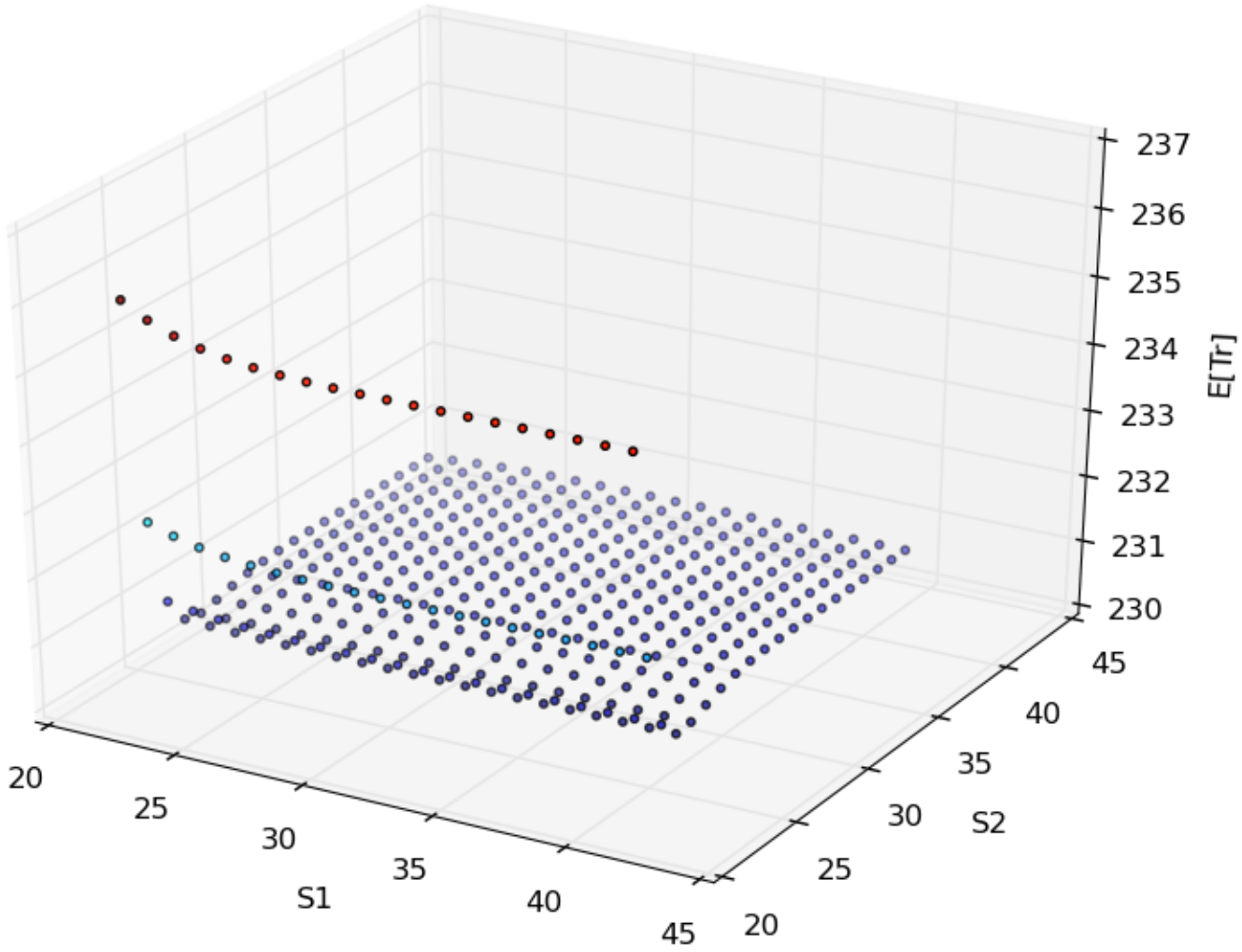
E[Tr] al variare di N



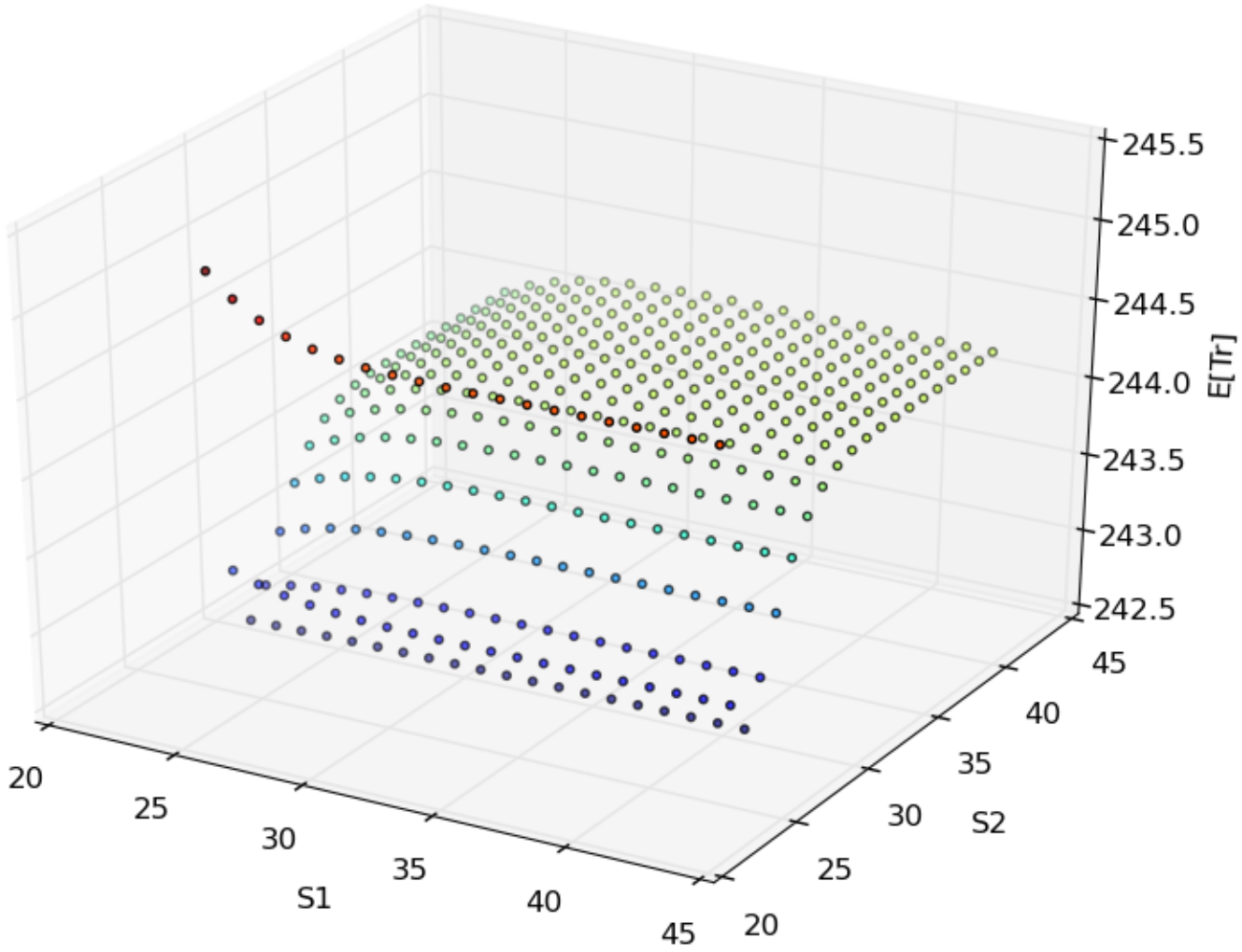
E[Tr] - N=39


230
228
226
224
222
220
218
E[Tr]
15
20
25
30
35
40
S1
15
20
25
30
35
40
S2

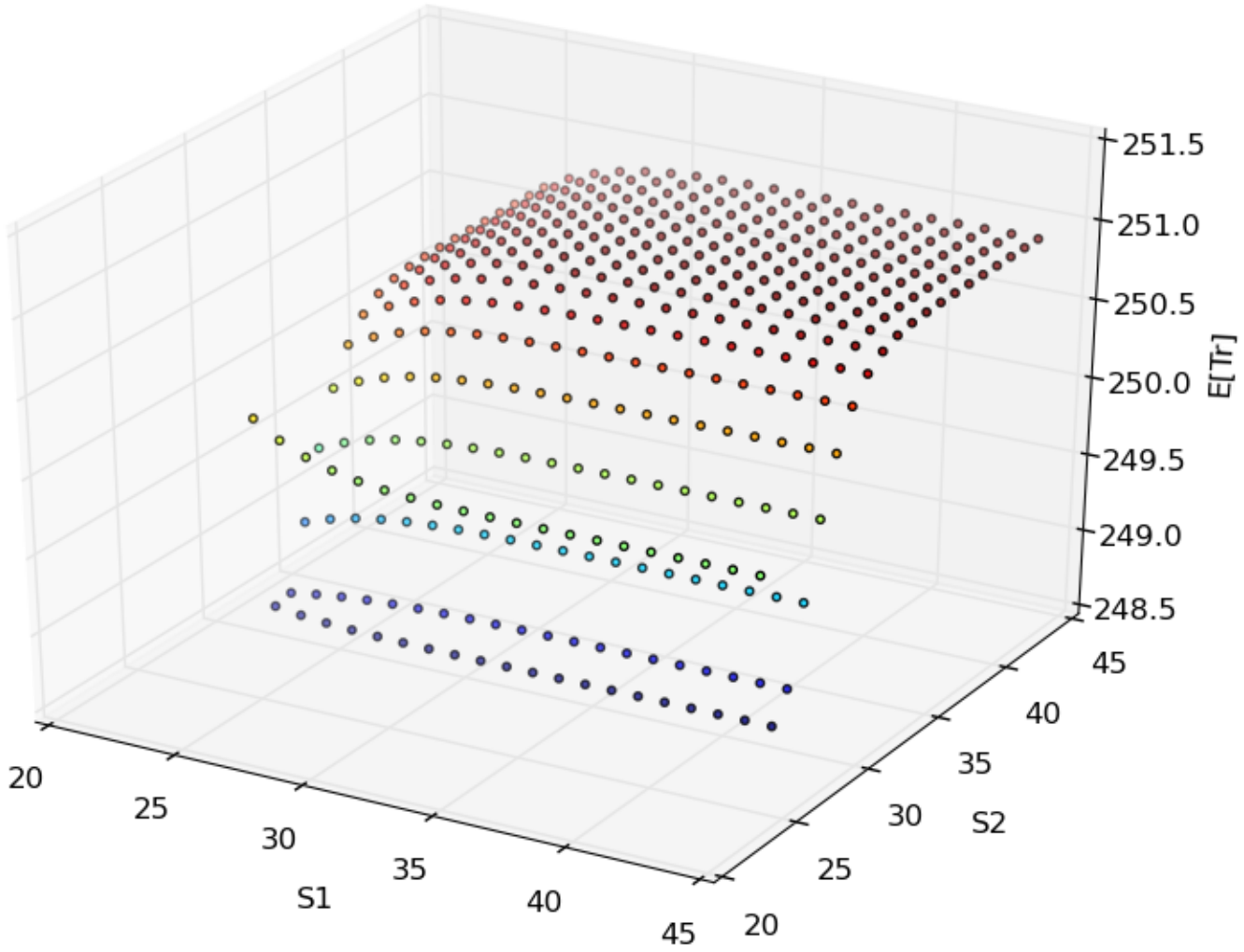
E[Tr] - N=41


237
236
235
234
233
232
231
230
E[Tr]
20
25
30
35
40
45
S1
20
25
30
35
40
45
S2

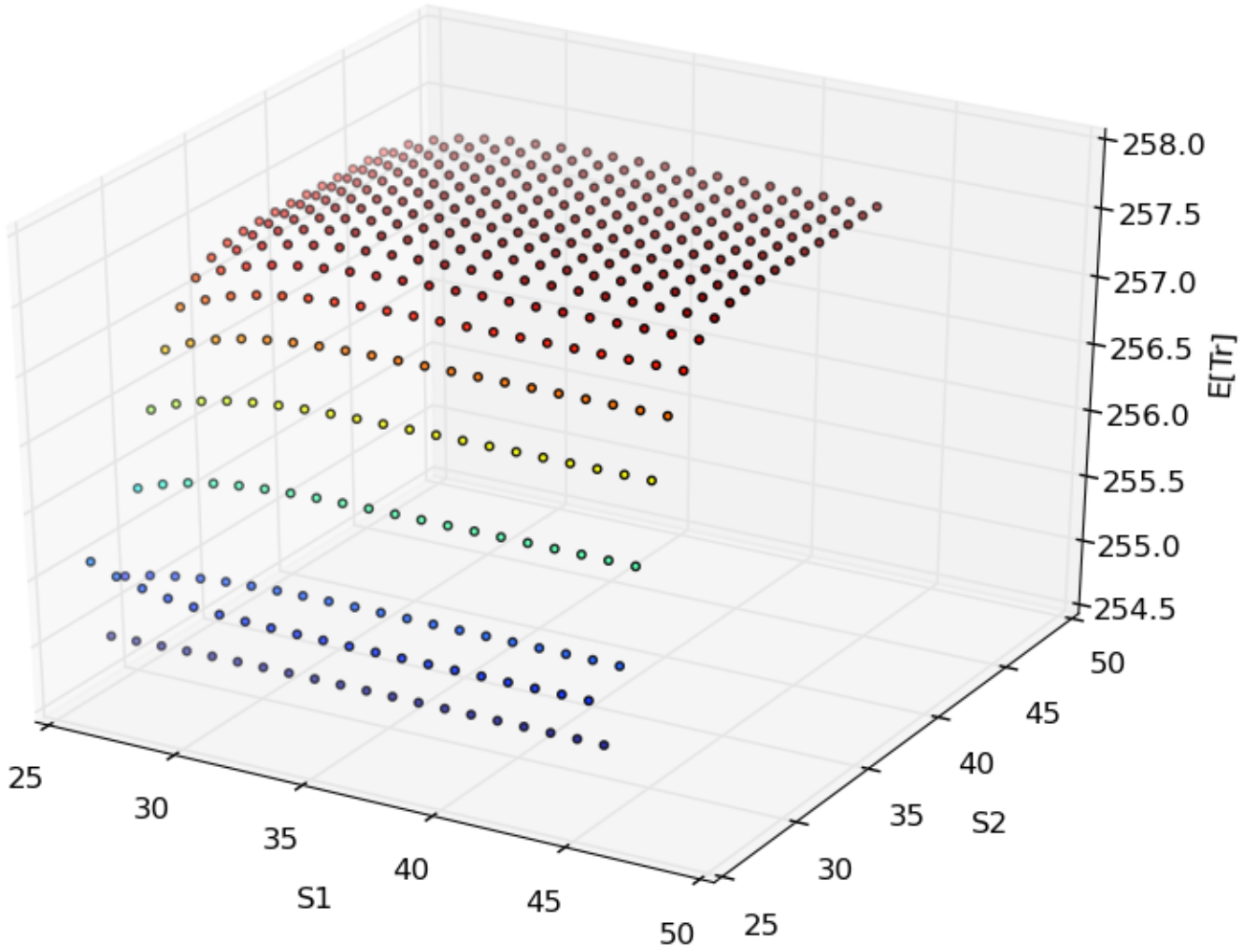
E[Tr] - N=43


245.5
245.0
244.5
244.0
243.5
243.0
242.5
E[Tr]
20
25
30
35
40
45
S1
20
25
30
35
40
45
S2

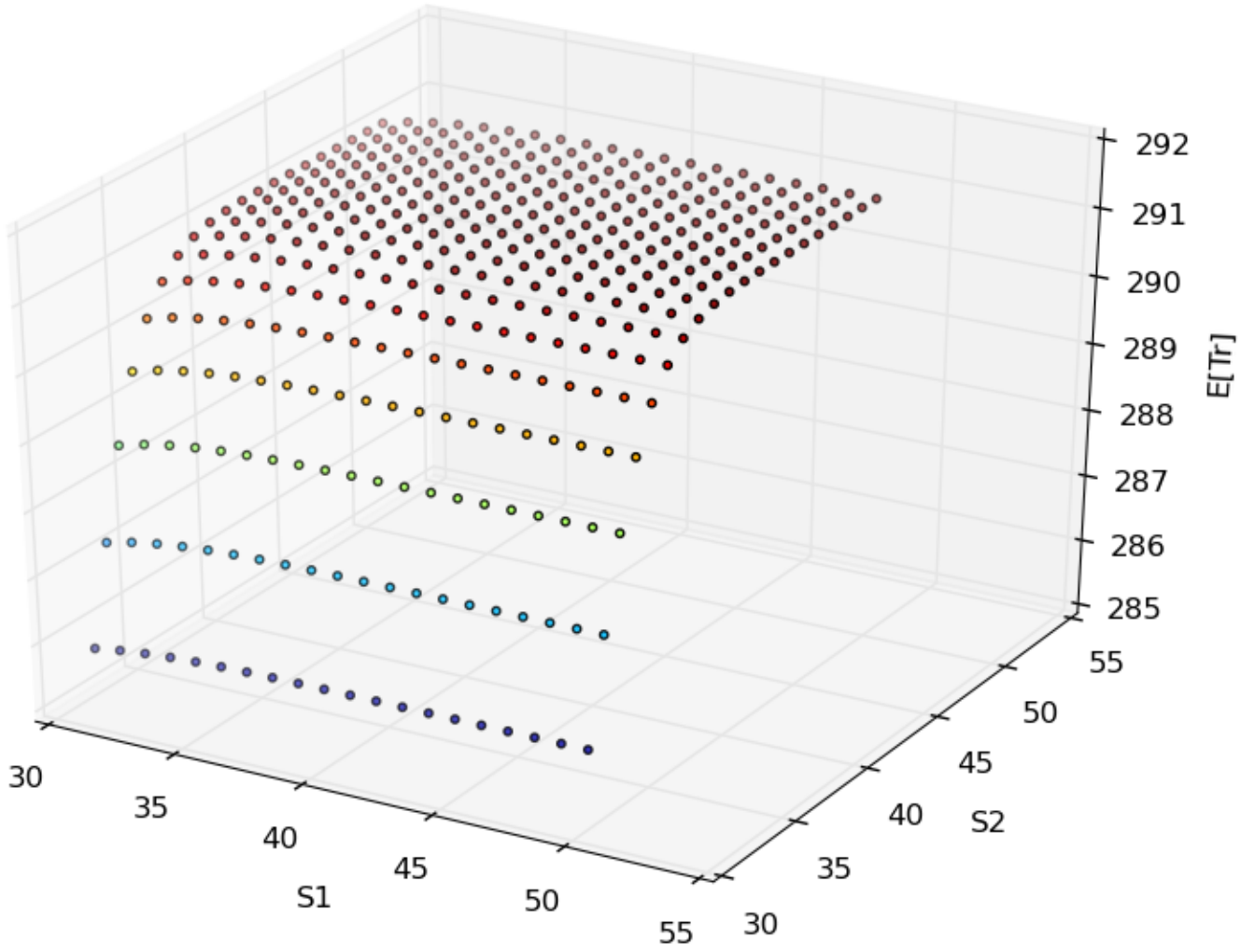
E[Tr] - N=44


251.5
251.0
250.5
250.0
249.5
249.0
248.5
E[Tr]
20
25
30
35
40
45
S1
20
25
30
35
40
45
S2

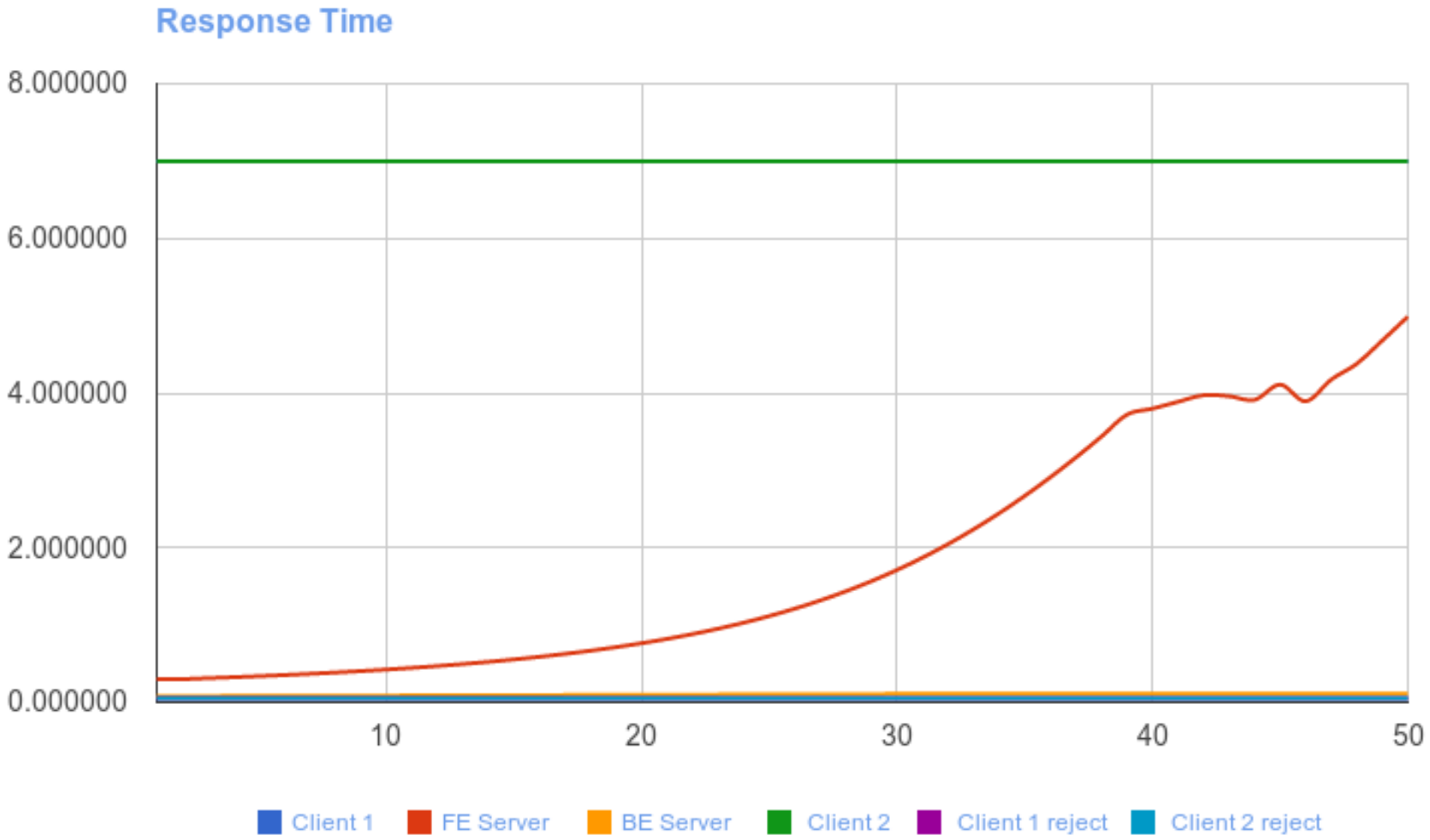
E[Tr] - N=45


258.0
257.5
257.0
256.5
256.0
255.5
255.0
254.5
E[Tr]
25
30
35
40
45
50
S1
25
30
35
40
45
50
S2

E[Tr] - N=50

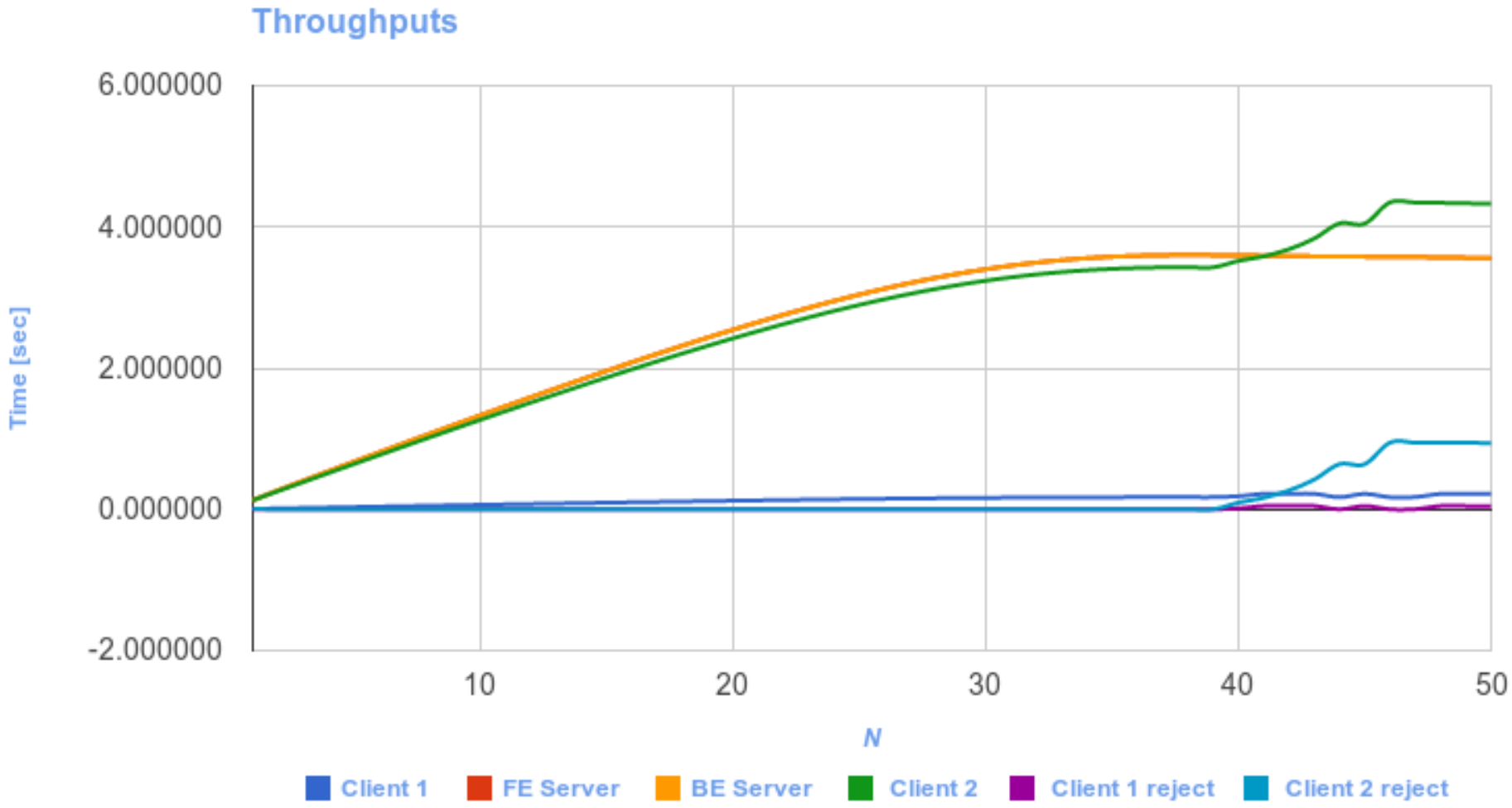

292
291
290
289
288
287
286
285
E[Tr]
30
35
40
45
50
55
S1
30
35
40
45
50
55
S2

Tempi di Risposta Locali



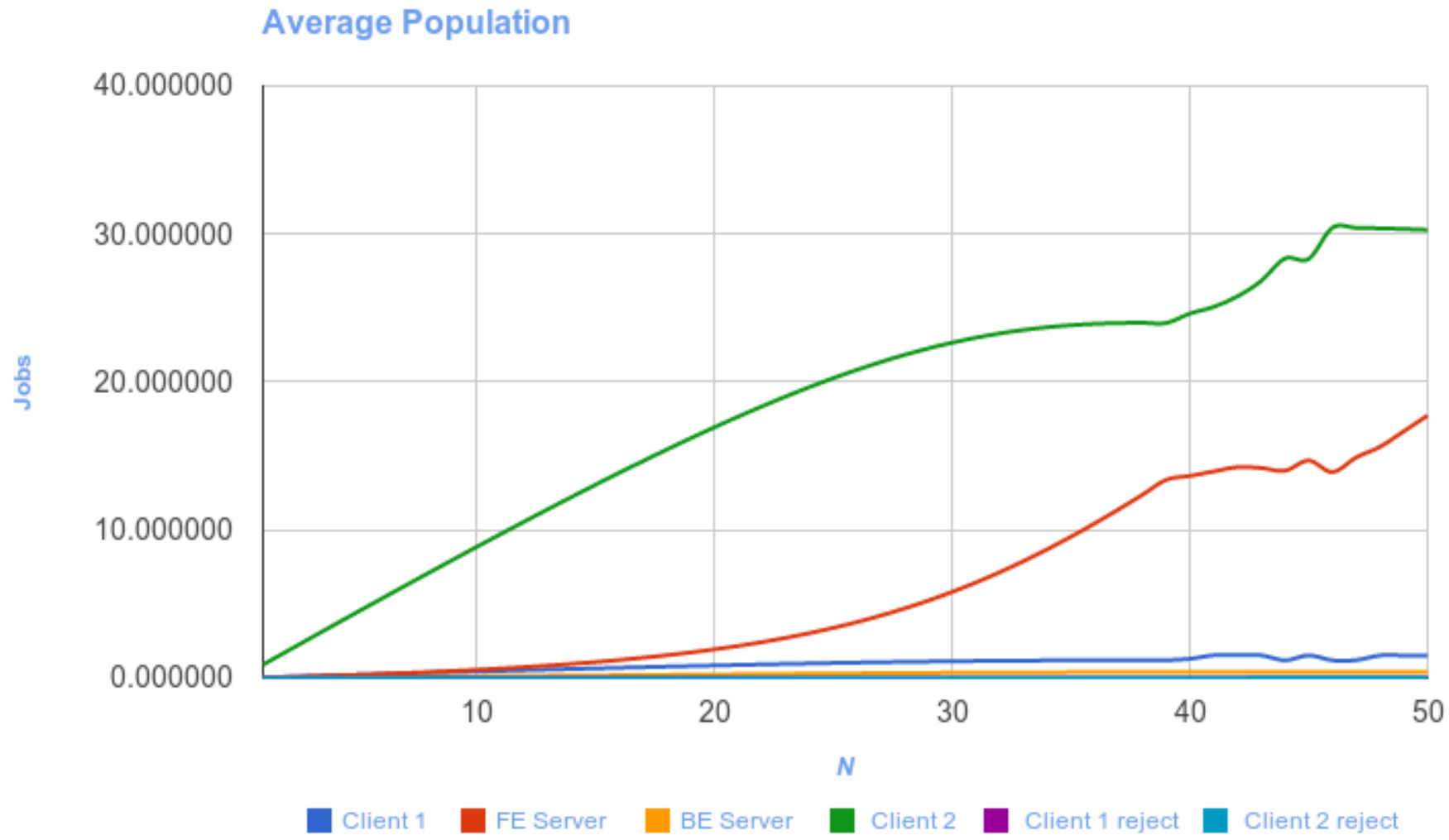
Si nota come il FE Server sia il collo di bottiglia del sistema. Il tempo di risposta del FE Server incrementa a partire da N=30, la derivata smette di aumentare soltanto quando si iniziano a manipolare i valori delle soglie.

Throughputs



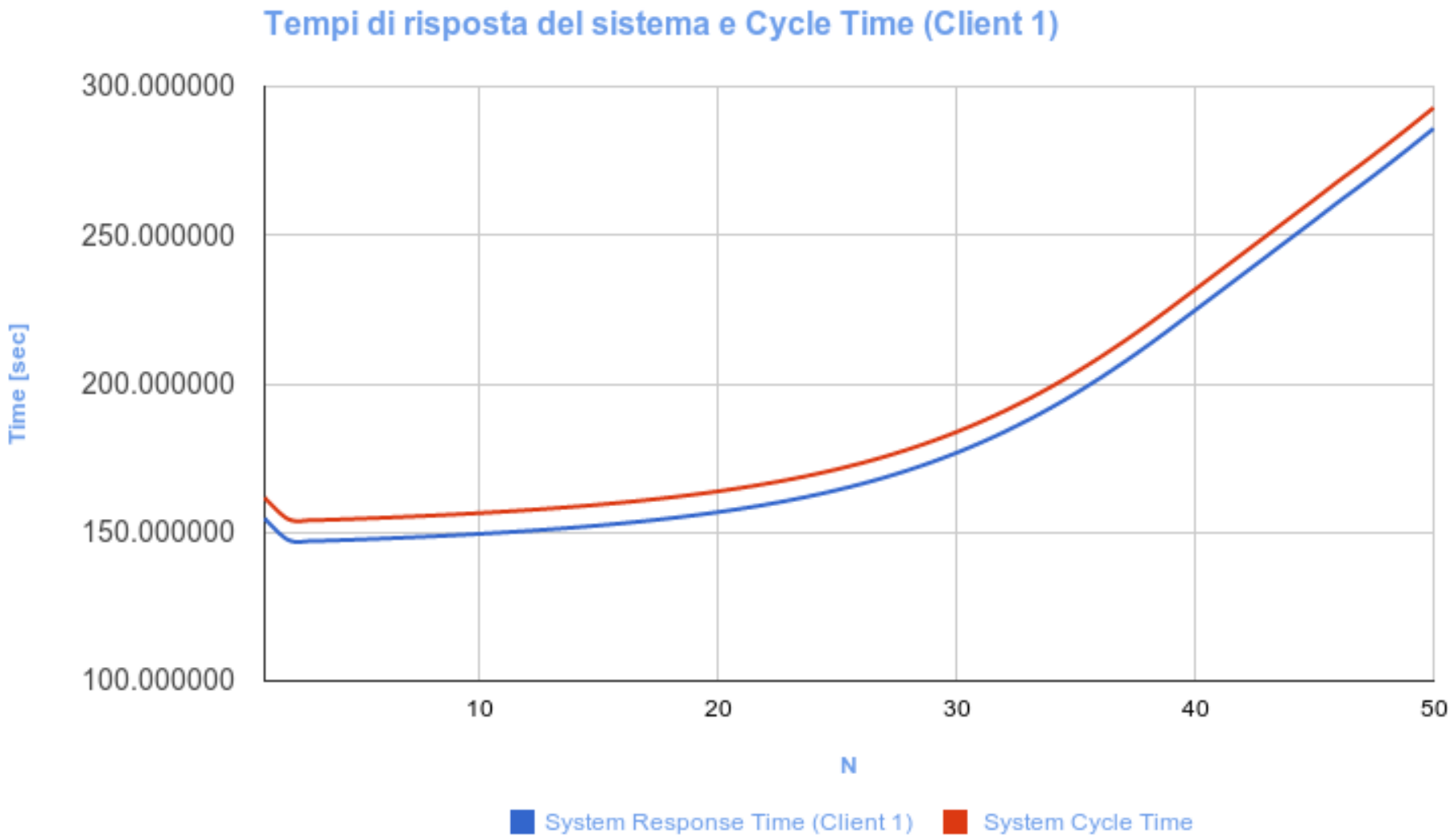
I Throughput dei **Client 1 e 2** e dei centri per la **rejection** subiscono visibili fluttuazioni in corrispondenza dell'attivazione del meccanismo **CAC**, ovvero a partire da quando nel sistema sono presenti almento 40 jobs.

Popolazioni Medie presso i Centri



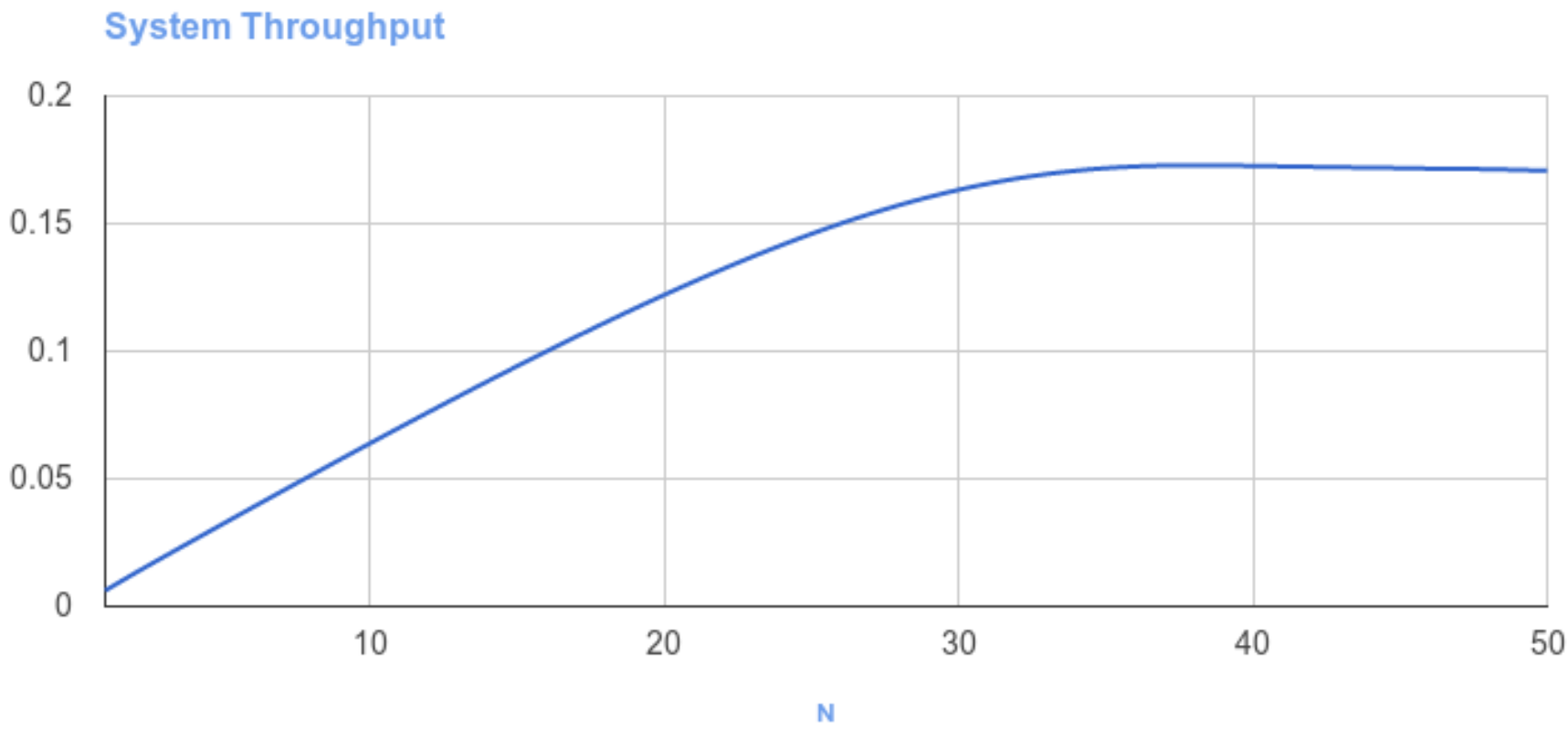
i jobs dell'impianto tendono ad accumularsi in coda al **FE Server** oppure in attesa del tempo di think presso **Client 2.** Si notano ancora le fluttuazioni dovuti all'attivazione del meccanismo **CAC.**

Tempo Medio di Risposta del Sistema



Il carico subito dal sistema, quando i jobs al suo interno sono più di 30, è tale da imporre una crescita sempre più sostenuta del tempo di risposta.

Throughput rispetto a Client 1



A seguito dell'attivazione delle soglie, il flusso di jobs all'interno del sistema viene in parte "assorbito" all'interno del meccanismo di **rejection**. Di fatto, vengono introdotti nuovi cicli che sottraggono jobs dal circuito principale.