

Performance Modeling of Computer Systems and Networks

Progetto A.A. 2020/2021

Alessandro Pontis
Gianmarco Bencivenni



Introduzione

In questo progetto si è lavorato su due differenti sistemi, uno l’evoluzione dell’altro, rappresentanti la medesima situazione, ovvero la coda in cassa al supermercato.

Modello Classico

- ▶ Si modella il *sistema classico* sperimentato alle casse di un supermercato
- ▶ Si simula il comportamento del sistema
- ▶ Si analizzano i risultati per determinare i limiti del modello, per poterli superare grazie ad un modello più avanzato

Modello Avanzato

- ▶ Il sistema avanzato viene ideato e modellato facendo riferimento ai *punti deboli* del sistema classico
- ▶ Si simula il comportamento e si effettuano i dovuti confronti per verificare l’effettivo miglioramento

Le fasi dello studio

Per entrambi i modelli presi in considerazione, sono stati seguiti i seguenti passi principali, come segue:

- Obiettivi dello studio
- Modello Concettuale
- Modello delle Specifiche
- Modello Computazionale
- Verifica
- Validazione
- Design degli esperimenti di simulazione
- Risultati
- Osservazioni

Modello Classico: Obiettivi dello studio

- ▶ Determinare il numero di serventi/cassieri operativi necessari a mantenere un'utilizzazione media che sia, per ogni centro, compresa tra 0.6 e 0.8. Questo range di valori è stato scelto in vista del fatto che i cassieri sono operatori non assimilabili a macchine, perciò l'utilizzazione non può mantenersi troppo alta in modo continuativo.
- ▶ Se esistono, valutare i valori degli indici di prestazione del sistema allo stato stazionario, in corrispondenza del carico in ingresso relativo ad ogni fascia oraria, al fine di stabilire i limiti del sistema, da superare con uno studio più avanzato.
- ▶ Una volta stabilito se il sistema modellato raggiunge lo stato stazionario, dire se i valori degli indici di prestazione del sistema raggiungono, entro la finestra temporale di ogni fascia oraria, gli eventuali valori studiati per lo stato stazionario con la configurazione relativa alla fascia oraria.

La simulazione dello stato stazionario acquista significato nel momento in cui i risultati dell'analisi transiente mostrano una convergenza ad uno stato stazionario.

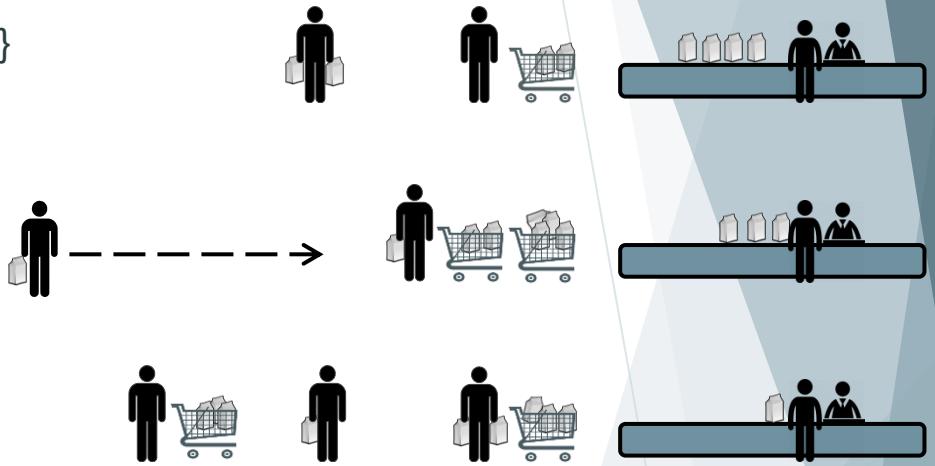
Modello Classico: Modello Concettuale (1 di 2)

► Variabili di Stato (al tempo t) :

- $E = \{ li(t) : \text{numero di utenti nel nodo } i\text{-esimo al tempo } t, 1 < i < r(t)\}$
- $r(t) : \text{numero di serventi (casse) attivi}$

► Lista degli Eventi :

- Arrivo al sistema
 - ❖ Mantenere una sola tipologia di arrivo (arrivo al sistema)
 - ❖ Arrivo al centro i -esimo : al processamento dell'arrivo il cliente sceglierà la coda presso la quale effettuare il servizio → Algoritmo di selezione (Routing dipendente dallo stato del sistema)
- Completamento al centro i -esimo:
 - ❖ È necessario definire un evento di completamento a sé per ognuno dei centri attivi nel sistema.



Modello Classico: Modello Concettuale (2 di 2)

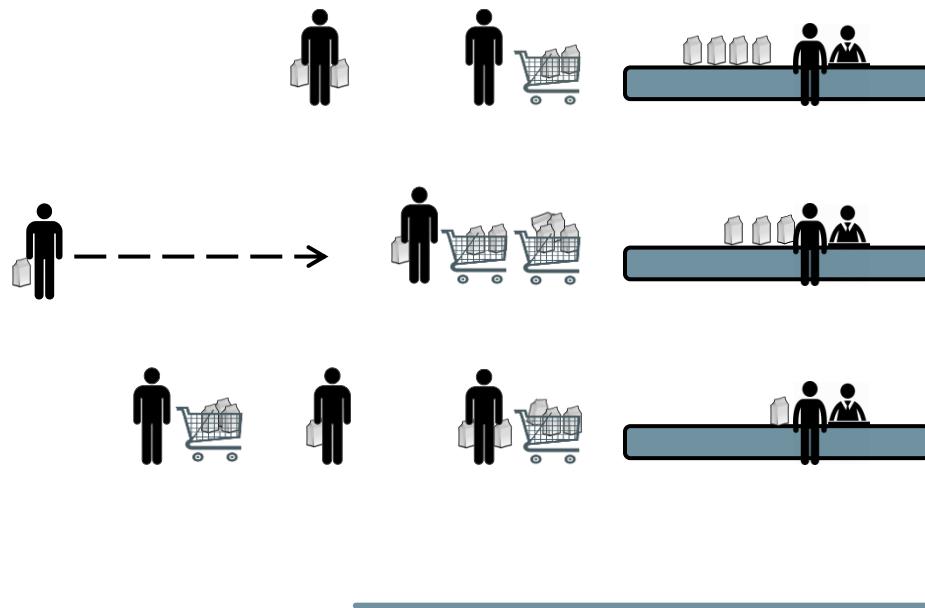
- ▶ Algoritmo di selezione (Routing dipendente dallo stato del sistema):
 - ▶ All'istante di processamento dell'arrivo vengono esaminate le lunghezze di tutte le code presenti nel sistema (dipendenza dallo stato del sistema).
 - ▶ Vengono selezionati gli indici di tutti i centri aventi il minor numero di clienti in attesa (set delle code candidate)
 - ▶ Tra queste, la scelta sul dove inoltrare la richiesta viene effettuata in modo randomico (distribuzione discreta uniforme).

Algorithm 1 Criterio di selezione della Coda (Greedy)

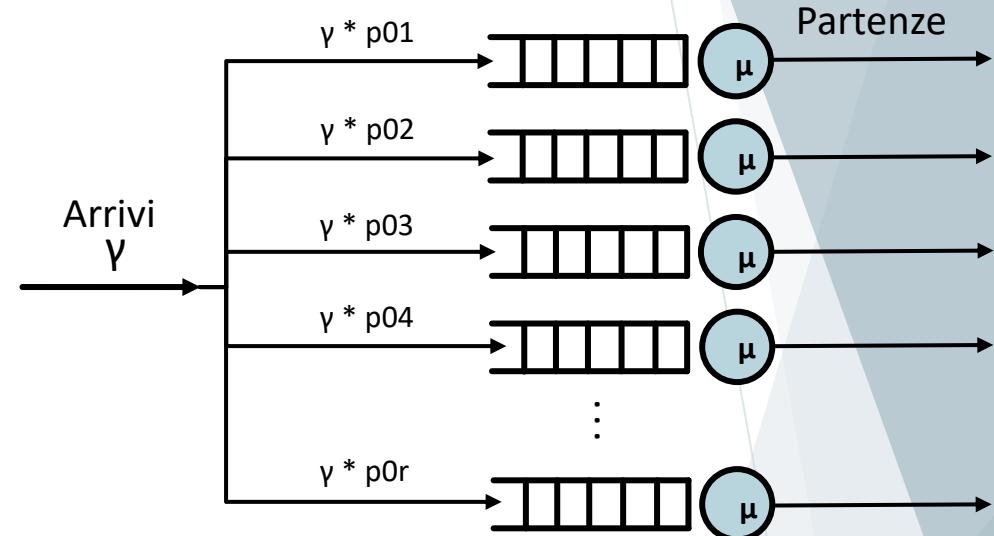
```
1: procedure SELECTNODE
2:   i  $\leftarrow$  0
3:   queueIndexes  $\leftarrow$  []
4:   while i  $<$  NODES-1 do
5:     i  $\leftarrow$  i + 1
6:     if nodes(choices[0]).number  $>$  nodes(i).number then
7:       while len(choices)  $\neq$  0 do
8:         choices.pop()
9:       end while
10:      choices.append(i)
11:    end if
12:    if nodes(choices[0]).number == nodes(i).number then
13:      choices.append(i)
14:    end if
15:  end while
16:  candidatesNumber  $\leftarrow$  len(choices)
17:  winner  $\leftarrow$  UniformSelection(0, candidatesNumber - 1)
18:  return choices[winner]
19: end procedure
```

(Dal modello delle specifiche...)

Modello Classico: Modello delle Specifiche (1 di 4)



❖ Schema concettuale



❖ Rete aperta di centri $M/G/1$
❖ r centri indipendenti e paralleli

Modello Classico: Modello delle Specifiche (2 di 4)

- ▶ Modellare le distribuzioni di arrivi e servizi :
 - ▶ Arrivi Esponenziali :
 - ▶ Tempo medio di inter-arrivo variabile nella giornata lavorativa di 14 ore (840 minuti)

FASCIA ORARIA	INTERARRIVO [min]	GRADO AFFOLLAMENTO
08.00 – 10.00	2.0	BASSO
10.00 – 13.00	1.5	INTERMEDIO
13.00 – 16.00	2.0	BASSO
16.00 – 20.00	1.0	ELEVATO
20.00 – 22.00	3.0	MOLTO BASSO

Modello Classico: Modello delle Specifiche (3 di 4)

- ▶ Modellare le distribuzioni di arrivi e servizi :
 - ▶ Distribuzione dei Servizi
 - ▶ Il tempo di servizio è determinato dalle dimensioni del carrello:
 - ▶ Clienti senza carrello (solo oggetti alla mano) → richiesta di servizio bassa
 - ▶ Clienti con cestino della spesa → richiesta di servizio media
 - ▶ Clienti con uno o più carrelli pieni → richiesta di servizio alta
 - ▶ In assenza di dati reali, una stima a grandi linee è di sperimentare:
 - ▶ il 20% delle richieste di servizio basse
 - ▶ La restante percentuale suddivisa tra richieste medie ed alte
 - ▶ Si cerca quindi una distribuzione con alta variabilità, ma che non sia perfettamente uniforme.
 - ▶ **Pareto troncata** tra due valori di confine ($k = 1 \text{ min}$, $p = 15 \text{ min}$)

Modello Classico: Modello delle Specifiche (4 di 4)

► Modellare le distribuzioni di arrivi e servizi :

► Servizi **Bounded-Pareto**

- ❖ Parametro **k** (valore minimo) : 1 minuto
- ❖ Parametro **p** (valore massimo) : 15 minuti
- ❖ Parametro α , derivazione:

$$F(1.5) = \frac{1 - 1.5^{-\alpha}}{1 - \left(\frac{1}{15}\right)^{\alpha}}$$

$$F(1.0) = 0$$

$$\text{Condizione per ricavare alpha : } F(1.5) = \frac{1 - 1.5^{-\alpha}}{1 - \left(\frac{1}{15}\right)^{\alpha}} = \frac{1.5^{\alpha} - 1}{1.5^{\alpha} - (0.1)^{\alpha}} = 0.2 \rightarrow 20\% \text{ richieste taglia piccola}$$

$$\alpha = 0.270343$$

Calcolato attraverso Wolfram Mathematica,
Poiché impossibile l'esplicitazione algebrica di α

$$\text{PDF : } f(x) = \frac{\alpha \cdot k^{\alpha} \cdot x^{-\alpha-1}}{1 - \left(\frac{k}{p}\right)^{\alpha}}$$

$$\text{CDF: } F(x) = \frac{1 - k^{\alpha} \cdot x^{-\alpha}}{1 - \left(\frac{k}{p}\right)^{\alpha}} = \frac{x^{\alpha} - k^{\alpha}}{x^{\alpha} - \left(\frac{kx}{p}\right)^{\alpha}}$$

Modello Classico: Verifica

- ▶ Sono stati applicati alcuni passaggi di controllo per verificare qualora il modello computazionale sia coerente e corretto rispetto a quanto illustrato nel modello delle specifiche.
- ▶ L'effettiva correttezza del modello computazionale è stata verificata attraverso una fase di testing, controllando la veridicità delle seguenti uguaglianze:
 - ▶ $\bar{w} = \bar{d} + \bar{s}$
 - ▶ $\bar{w}_i = \bar{d}_i + \bar{s}_i$
 - ▶ $\bar{l} = \bar{w} \cdot \bar{\gamma}$ (Little)
 - ▶ $\bar{q} = \bar{d} \cdot \bar{\gamma}$ (Little)
 - ▶ $\bar{q}_i = \bar{d}_i \cdot \bar{\gamma}_i$ (Little)
 - ▶ $\bar{r} = \frac{a_n}{n}$

Modello Classico: Validazione

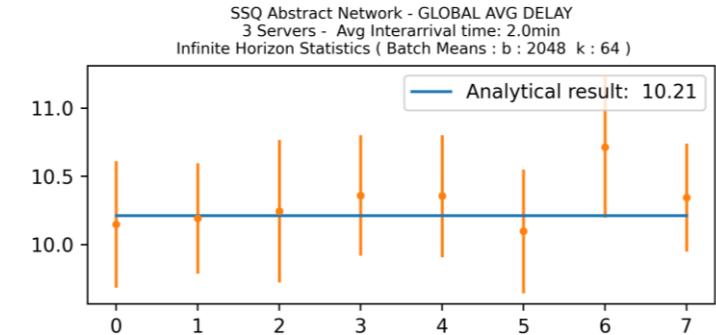
- ▶ Idea: confrontare alcune istanze di simulazione a risultati ottenuti con un sistema reale.
 - ▶ Confronto realistico, indistinguibilità agli occhi di un esperto.
- ▶ Non disponendo di tracce (dataset) reali, si opera una validazione rispetto al *modello analitico*.
Configurazione del sistema scelta per la validazione:
 - ▶ Distribuzione dei tempi di inter-arrivo :
 - ▶ Esponenziale di media $1/\gamma = 2 \text{ min}$
 - ▶ Tasso d'arrivo al sistema :
 - ▶ $\gamma = 0.5 \text{ req/min}$
 - ▶ Distribuzione dei tempi di servizio :
 - ▶ Bounded-Pareto di parametri $\alpha = 0.2703$, $k = 1 \text{ min}$, $p = 15 \text{ min}$ (4)
 - ▶ Numero di centri : 3

Modello Classico: Validazione

Risultati analitici:

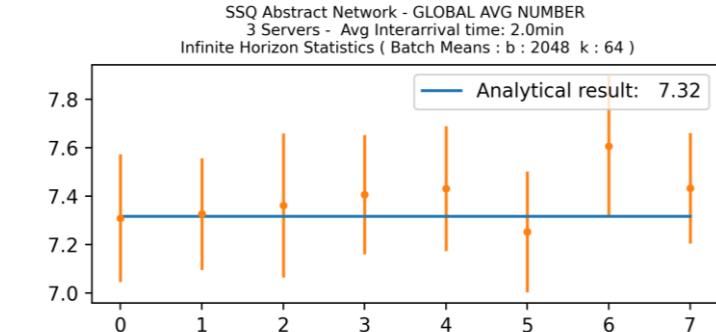
- ❖ $\lambda = \gamma / 3 = 0.166667 \frac{req}{min}$
- ❖ $E(S) = 4.42 min$
- ❖ $E(S^2) = 32.2791123$
- ❖ $\rho = \lambda \cdot E(S) = 0.736667$
- ❖ $E(TQ) = \frac{\lambda E(S^2)}{1-\rho} = 10.213139 min$
- ❖ $E(NQ) = \lambda \cdot E(TQ) = 1.702190174$
- ❖ $E(TS) = E(TQ) + E(S) = 14.633139 min$
- ❖ $E(NS) = 3 \cdot (E(NQ) + \rho) = 7.316571521$

- ❖ Il sistema risulta validato una volta effettuate le prove di simulazione *infinite horizon* con la tecnica delle *batch-means*



SSQ Abstract Network - GLOBAL AVG NUMBER
3 Servers - Avg Interarrival time: 2.0min
Infinite Horizon Statistics (Batch Means : b : 2048 k : 64)

Replica	SEED	MEAN VALUE	ERROR	CONFIDENCE LEVEL
0	123456789	7.310198596439312	±0.2631732848992967	95%
1	982312514	7.327212755458038	±0.23029985702827446	95%
2	1859433496	7.362103898510023	±0.297367427179700177	95%
3	950087593	7.406421717907727	±0.24625801905882788	95%
4	192157662	7.431162294423393	±0.2577915140412271	95%
5	993523983	7.725838068704847	±0.24945546840218502	95%
6	283500777	7.6075233693971684	±0.290944257223265	95%
7	1557799130	7.433112123910361	±0.2282999489856343	95%



Modello Avanzato: Obiettivi dello studio

- ▶ Mostrare prestazioni globali migliori rispetto al modello classico, con particolare interesse rivolto ai tempi d'attesa e di risposta.
- ▶ Per ogni fascia oraria presa in considerazione, limitare l'attesa media della maggior parte dei clienti entro il valore medio globale osservato per il caso di studio classico e relativo alla stessa fascia oraria.
- ▶ Adottare una policy di scheduling fair, che garantisca un attesa proporzionata alla richiesta di servizio.

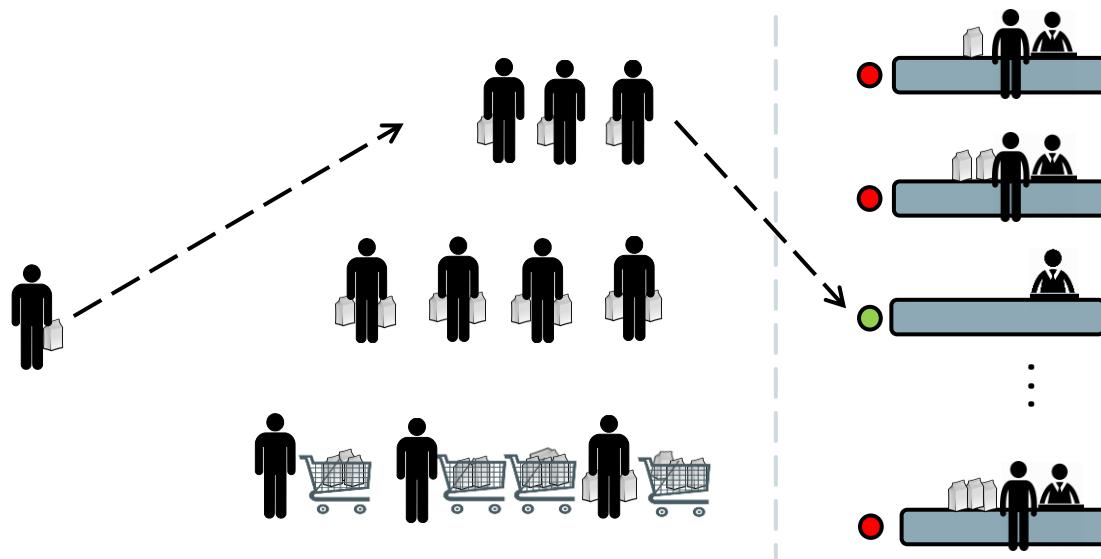
Modello Avanzato: Modello Concettuale (1 di 2)

- ▶ Il Modello Avanzato viene ideato sulla base degli obiettivi che ci si è posti:
 - ❑ prestazioni globali migliori rispetto al modello classico (scheduling astratto)
 - ❑ limitare l'attesa media della maggior parte dei clienti entro il valore medio globale osservato per il caso di studio classico
 - ❑ Fairness
 - ⇒ scheduling basato sulla dimensione della richiesta
 - ⇒ introduzione di classi di priorità
- ▶ L'evoluzione dal modello classico è a costo zero (la topologia rimane la stessa)
- ▶ La dimensione della richiesta di servizio è nota nel momento dell'arrivo
- ▶ Lista degli Eventi :
 - ❑ Arrivo al sistema
 - ❑ Completamento al centro i-esimo

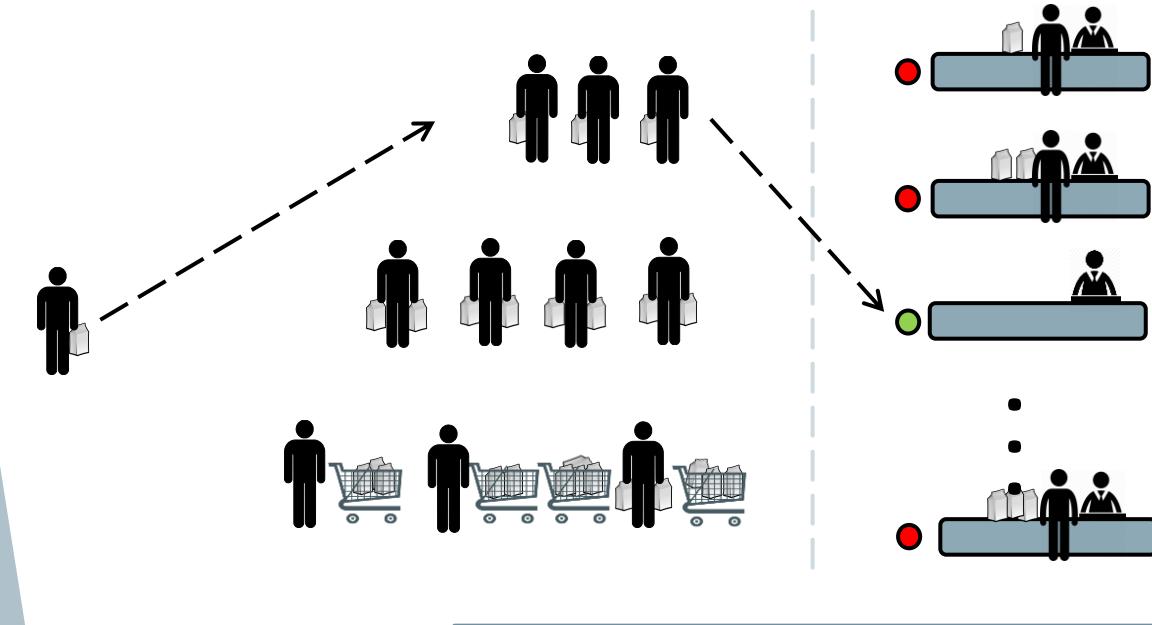
Modello Avanzato: Modello Concettuale (2 di 2)

► Variabili di Stato (al tempo t) :

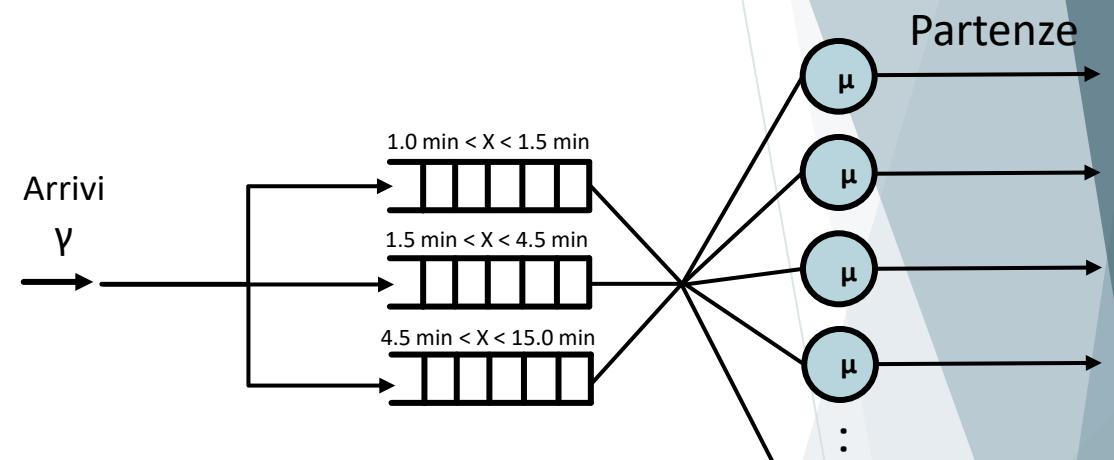
- $E(t) = \begin{cases} N_{Qi}(t) & \text{numero di utenti nella coda } i\text{-esima al tempo } t, 1 \leq i \leq 3 \\ x_i(t) & \text{stato (0/1) del server } i\text{-esimo al tempo } t, 1 \leq i \leq m \end{cases}$
- $l(t) = \sum_{i=1}^3 N_{Qi}(t) + \sum_{i=1}^m x_i(t)$ è il numero totale di utenti nel sistema al tempo t.



Modello Avanzato: Modello delle Specifiche (1 di 2)



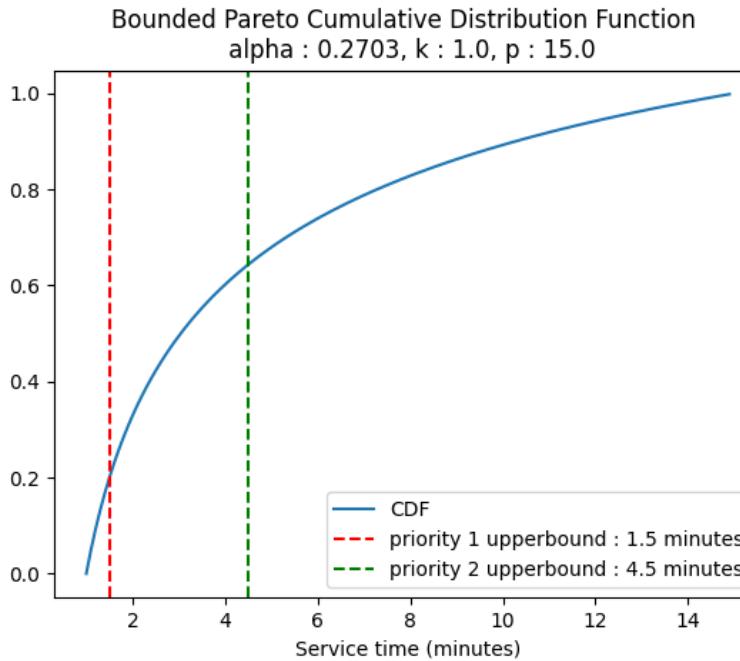
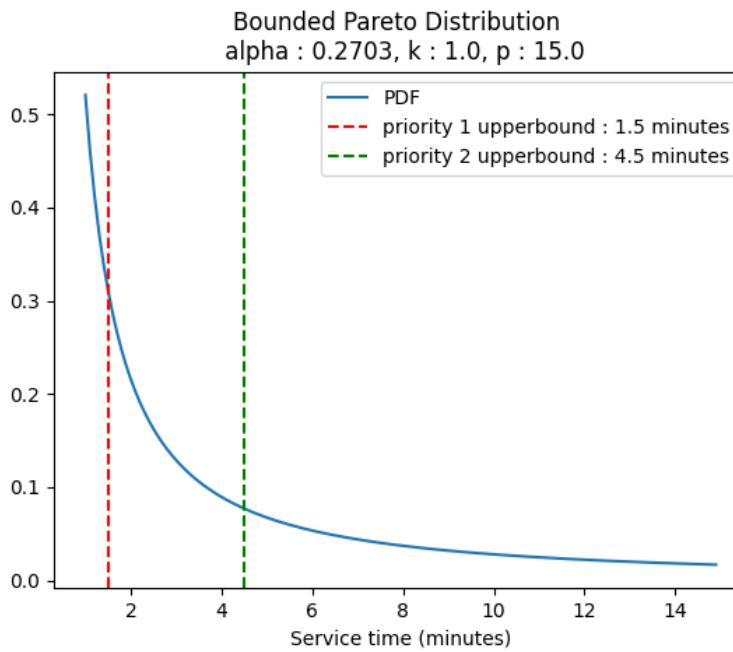
❖ Schema concettuale



❖ Multi-server (m server)
❖ 3 code a priorità
❖ Policy di scheduling Size Based

Modello Avanzato: Modello delle Specifiche (2 di 2)

- ▶ P1 : $Pr\{1.0 \leq X < 1.5\} = F(1.5) - F(1.0) = 0.199991$
- ▶ P2 : $Pr\{1.5 \leq X < 4.5\} = F(4.5) - F(1.5) = 0.443606$
- ▶ P3 : $Pr\{4.5 \leq X \leq 15\} = F(15) - F(4.5) = 0.356402$



Modello Avanzato: Verifica

- ▶ Sono stati applicati alcuni passaggi di controllo per verificare qualora il modello computazionale sia coerente e corretto rispetto a quanto illustrato nel modello delle specifiche.
- ▶ L'effettiva correttezza del modello computazionale è stata verificata attraverso una fase di testing, controllando la veridicità delle seguenti uguaglianze:
 - ▶ $\bar{w} = \bar{d} + \bar{s}$
 - ▶ $\bar{w}_i = \bar{d}_i + \bar{s}_i$
 - ▶ $\bar{l} = \bar{w} \cdot \bar{\gamma}$ (Little)
 - ▶ $\bar{q}_i = \bar{d}_i \cdot \bar{\gamma}_i = \bar{d}_i \cdot \bar{\gamma} p_i$ (Little)
 - ▶ $\bar{r} = \frac{a_n}{n}$

Modello Avanzato: Validazione

- ▶ Idea: confrontare alcune istanze di simulazione a risultati ottenuti con un sistema reale.
 - ▶ Confronto realistico, indistinguibilità agli occhi di un esperto.
- ▶ Non disponendo di tracce (dataset) reali, si opera una validazione rispetto al *modello analitico*.
Configurazione del sistema scelta per la validazione:
 - ▶ Distribuzione dei tempi di inter-arrivo :
 - ▶ Esponenziale di media $1/\gamma = 6 \text{ min}$
 - ▶ Tasso d'arrivo al sistema :
 - ▶ $\gamma = 0.1666667 \text{ req/min}$
 - ▶ Distribuzione dei tempi di servizio :
 - ▶ Bounded-Pareto di parametri $\alpha = 0.2703$, $k = 1 \text{ min}$, $p = 15 \text{ min}$ (4)
 - ▶ Numero di server: 1 → Necessario per effettuare i calcoli per i risultati analitici.

Modello Avanzato: Validazione

► Risultati analitici:

$$\rho = \lambda \cdot E(S) = 0.736667$$

$$E(T_{Qk}) = \frac{\frac{\lambda}{2} \cdot E(S^2)}{(1 - \lambda \cdot \int_{x_0}^{x_{k-1}} t \cdot f(t) dt) \cdot (1 - \lambda \cdot \int_{x_0}^{x_k} t \cdot f(t) dt)}$$

$$E(S_k) = \frac{1}{p_k} \cdot \int_{x_{k-1}}^{x_k} t \cdot f(t) dt$$

$$E(T_{Q1}) = 2.804787 \text{ min}$$

$$E(N_{Q1}) = \lambda \cdot p_1 \cdot E(T_{Q1}) = 0.09348871152$$

$$E(S_1) = 1.22865 \text{ min}$$

$$E(T_{Q2}) = 3.678428 \text{ min}$$

$$E(N_{Q2}) = \lambda \cdot p_2 \cdot E(T_{Q2}) = 0.2719621763$$

$$E(S_2) = 2.65837 \text{ min}$$

$$E(T_{Q3}) = 13.524543 \text{ min}$$

$$E(N_{Q3}) = \lambda \cdot p_3 \cdot E(T_{Q3}) = 0.8033623784$$

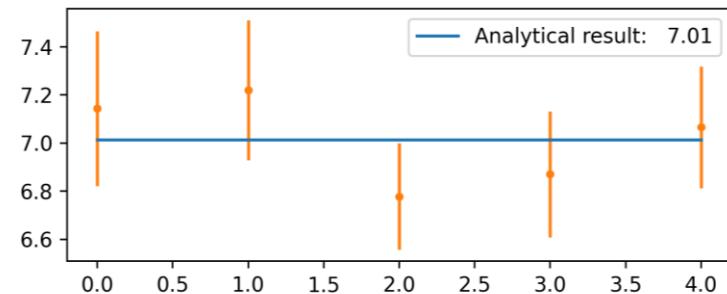
$$E(S_3) = 8.445309 \text{ min}$$

$$E(N_S) = E(N_{Q1}) + E(N_{Q2}) + E(N_{Q3}) + \rho = 1.905480266$$

$$E(T_Q) = p_1 \cdot E(T_{Q1}) + p_2 \cdot E(T_{Q2}) + p_3 \cdot E(T_{Q3}) = 7.012879063 \text{ min}$$

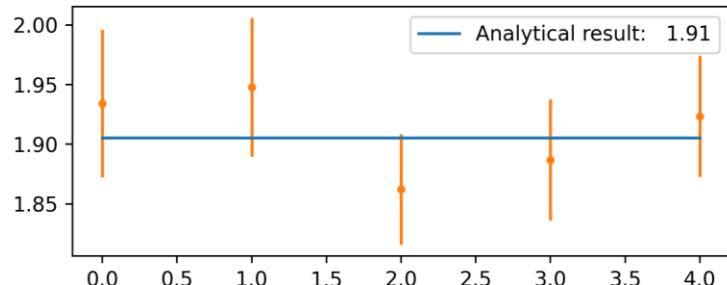
$$E(T_S) = E(T_Q) + E(S) = 11.43287906 \text{ min}$$

Size Based MSMQ - GLOBAL AVG DELAY
1 Servers - Avg Interarrival time: 6.0min
Infinite Horizon Statistics (Batch Means : b : 2048 k : 64)



	SEED	MEAN VALUE	ERROR	CONFIDENCE LEVEL
Replica 0	123456789	7.1431565723652986	±0.32112739681749664	95%
Replica 1	1118987882	7.219573436966225	±0.2913933520571782	95%
Replica 2	901760786	6.778185468317601	±0.22115596591348172	95%
Replica 3	1910652730	6.870467732385949	±0.2619309761162776	95%
Replica 4	773502992	7.0659972644530455	±0.2529499892949309	95%

Size Based MSMQ - GLOBAL AVG NUMBER
1 Servers - Avg Interarrival time: 6.0min
Infinite Horizon Statistics (Batch Means : b : 2048 k : 64)



	SEED	MEAN VALUE	ERROR	CONFIDENCE LEVEL
Replica 0	123456789	1.934108426458302	±0.06186670497592347	95%
Replica 1	1118987882	1.947854220490884	±0.05813968717108796	95%
Replica 2	901760786	1.8621721695480542	±0.04644398601889188	95%
Replica 3	1910652730	1.8868162407762716	±0.0507572580146677	95%
Replica 4	773502992	1.9233997719064284	±0.05075225635435334	95%

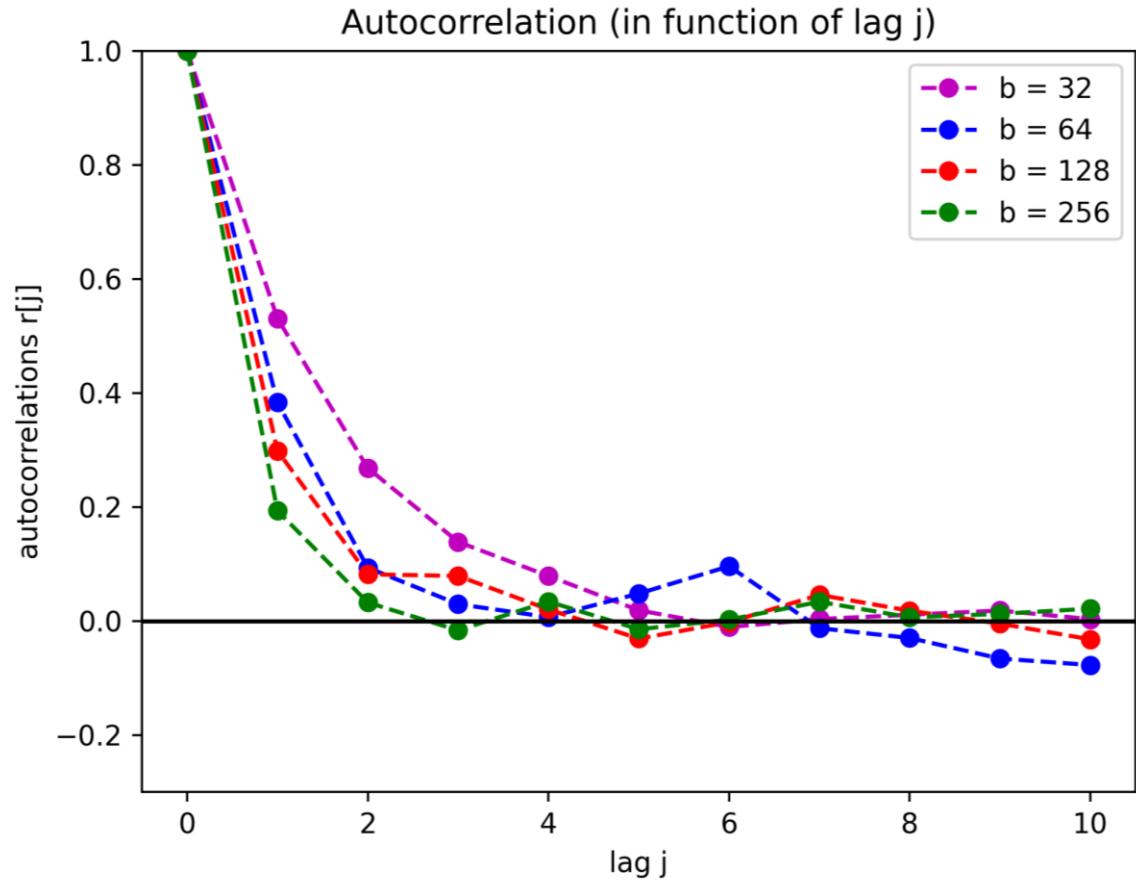
Metodo delle Batch-Means

- ▶ Si effettua una lunga run di simulazione, producendo un numero $n = b \cdot k$ di richieste processate.
- ▶ Il singolo batch è una sequenza di b valori consecutivi relativi alle statistiche di output d'interesse, acquisiti durante la simulazione.
- ▶ Procedendo nella simulazione fino al suo termine, verrà costruita una sequenza di valori di lunghezza $b \cdot k$, ovvero la sequenza dei k batches:
 - ▶ $x_1, \dots, x_b, x_{b+1}, \dots, x_{2b}, \dots, x_{(k-1)b+1}, x_{kb}$
- ▶ Si calcola dunque la media \bar{x}^i di ogni batch, ottenendo la sequenza delle batch-means:
 - ▶ $\bar{x}^1, \bar{x}^2, \bar{x}^3, \dots, \bar{x}_{k-1}$
- ▶ Si calcolano media e deviazione standard del set di batch-means (la media rappresenta la stima puntuale a regime della grandezza in questione).
- ▶ Per ricavare la stima intervallare, si sceglie il livello di confidenza $1 - \alpha$, e si ricava il valore critico $t *$ invertendo la funzione di distribuzione Student($k - 1, 1 - \alpha/2$).

Si è deciso di scartare il primo batch, poiché è quello relativo alle prime acquisizioni della run di simulazione, pertanto è il batch più influenzato dalle condizioni iniziali.

Metodo delle Batch-Means: *trade-off tra b e k*

- ▶ La politica scelta per effettuare il trade-off tra b e k è la seguente:
 - ▶ k viene scelto pari a **64**, essendo il *valore consigliato*.
 - ▶ L'ampiezza di **b** è determinata in riferimento al valore dell'autocorrelazione tra le successive acquisizioni della stessa statistica (si è utilizzato il tempo di risposta medio globale), secondo il seguente criterio:
 - ▶ Si analizza la funzione di autocorrelazione per batch-size crescenti, a partire da $b = 32$.
 - ▶ Si raddoppia il valore di b finché non viene soddisfatta la condizione:
 - Banks, Carson, Nelson, and Nicol:
l'autocorrelazione tra le batch means con lag $j = 1$ sia al più 0.2.

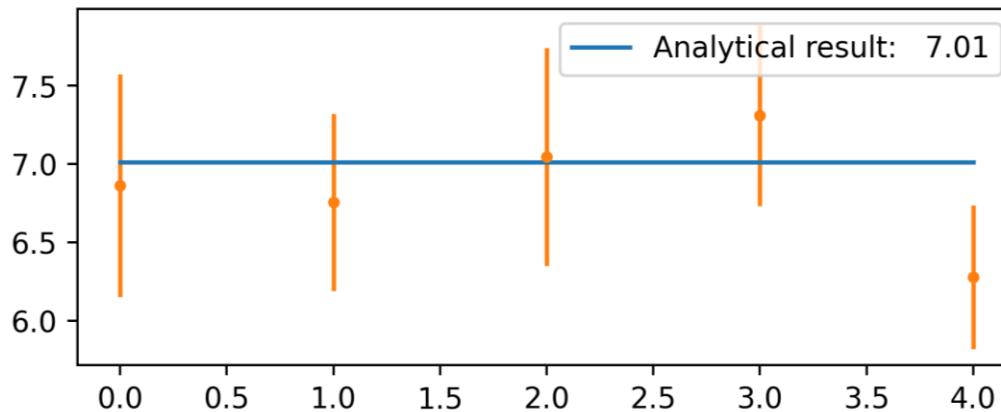


Metodo delle Batch-Means: *trade-off* tra b e k

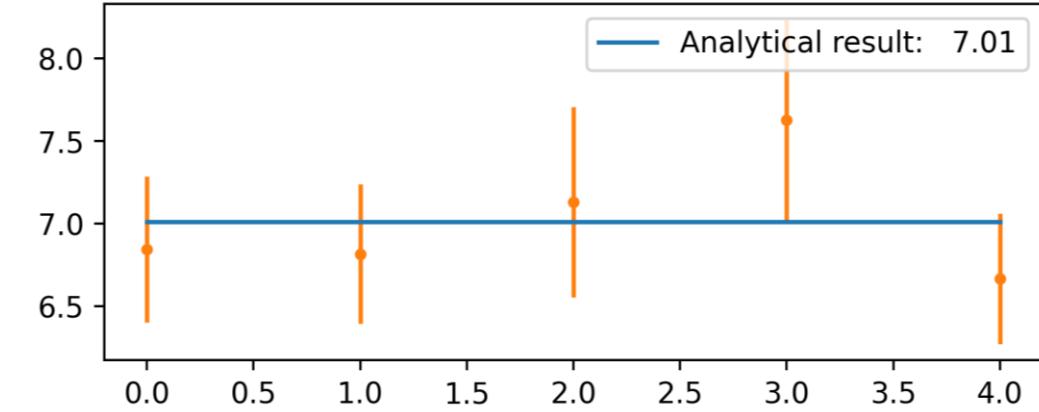
- ▶ Dal precedente studio si evince che utilizzando la configurazione $b = 256$ e $k = 64$ si raggiunge *all'incirca* l'indipendenza tra le batch-means consecutive.
- ▶ Tuttavia si è deciso di incrementare l'ampiezza di **b** seguendo un **processo iterativo (metodo sperimentale)**:
 - ▶ Partendo da $b = 256$
 - ▶ Si raddoppia il valore di b finché non viene raggiunto un buon compromesso tra la dimensione degli intervalli di confidenza e il livello di sovrapposizione questi e la media reale.
- ▶ Seguono alcuni esempi grafici del procedimento iterativo adottato

Metodo delle Batch-Means: *trade-off tra b e k*

Size Based MSMQ - GLOBAL AVG DELAY
1 Servers - Avg Interarrival time: 6.0min
Infinite Horizon Statistics (Batch Means : b : 256 k : 64)



Size Based MSMQ - GLOBAL AVG DELAY
1 Servers - Avg Interarrival time: 6.0min
Infinite Horizon Statistics (Batch Means : b : 512 k : 64)

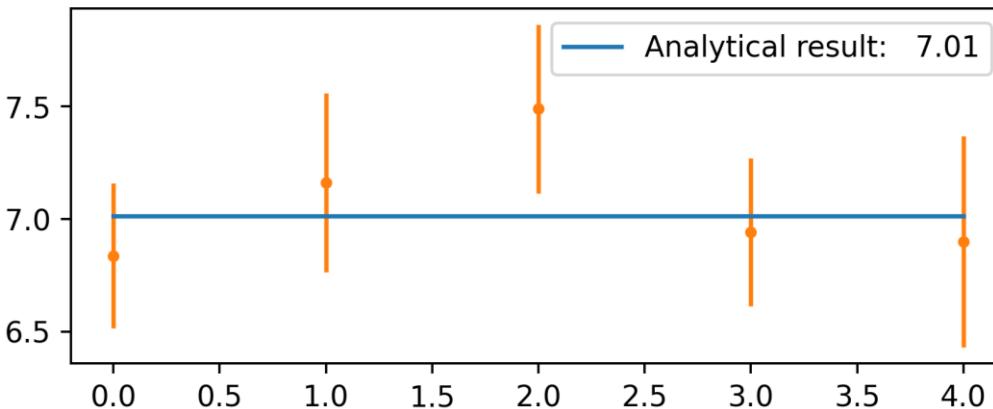


	SEED	MEAN VALUE	ERROR	CONFIDENCE LEVEL
Replica 0	123456789	6.864545302279541	±0.7106094948472409	95%
Replica 1	1234823845	6.756032774540483	±0.5656740914652859	95%
Replica 2	747393892	7.046633070258679	±0.6956029259669766	95%
Replica 3	1019426767	7.3115275634570445	±0.5769766185260496	95%
Replica 4	361517148	6.279322007430931	±0.4590207205490382	95%

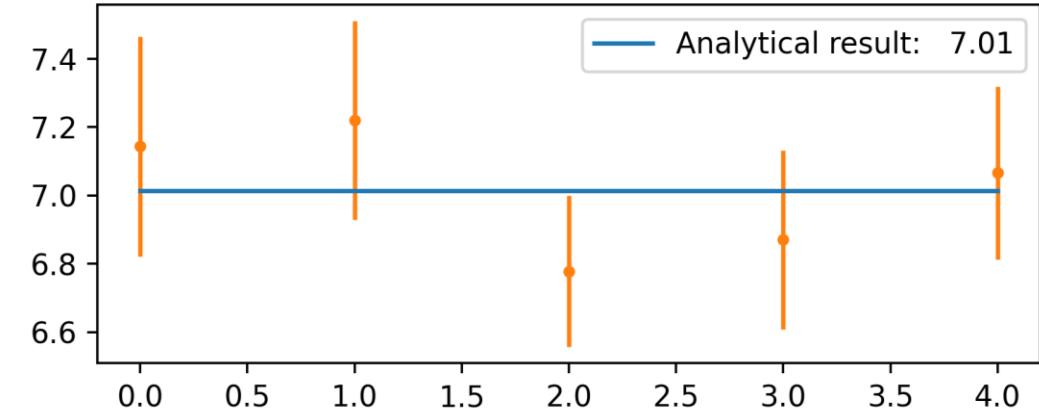
	SEED	MEAN VALUE	ERROR	CONFIDENCE LEVEL
Replica 0	123456789	6.8439586653315025	±0.441410207724007	95%
Replica 1	1234823845	6.816893905819939	±0.42260969761273975	95%
Replica 2	747393892	7.130584112528023	±0.5761883827545116	95%
Replica 3	1019426767	7.628275413355518	±0.605831265794567	95%
Replica 4	361517148	6.665670871286582	±0.39460797715622875	95%

Metodo delle Batch-Means: *trade-off tra b e k*

Size Based MSMQ - GLOBAL AVG DELAY
 1 Servers - Avg Interarrival time: 6.0min
 Infinite Horizon Statistics (Batch Means : b : 1024 k : 64)



Size Based MSMQ - GLOBAL AVG DELAY
 1 Servers - Avg Interarrival time: 6.0min
 Infinite Horizon Statistics (Batch Means : b : 2048 k : 64)



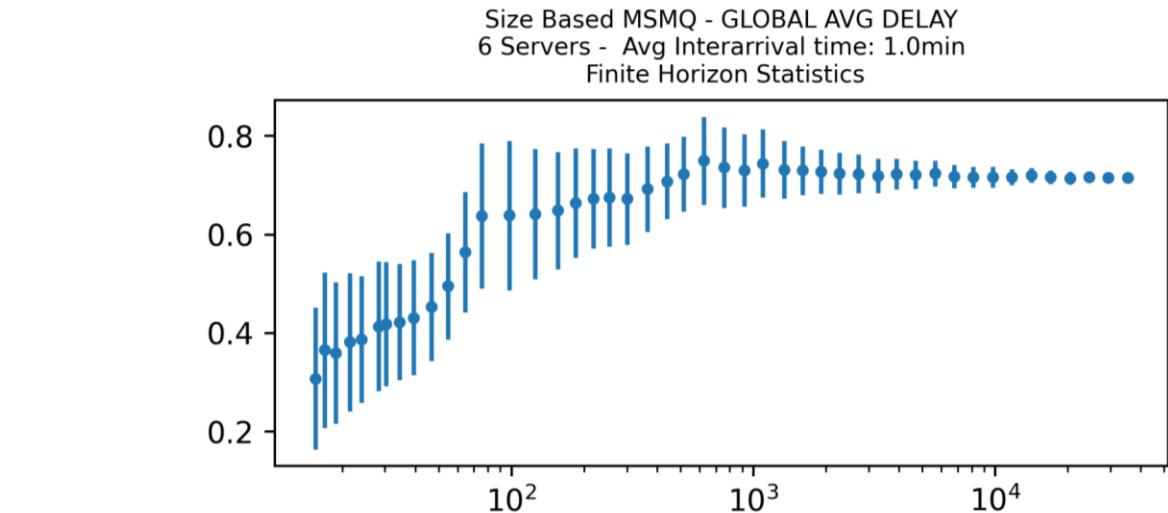
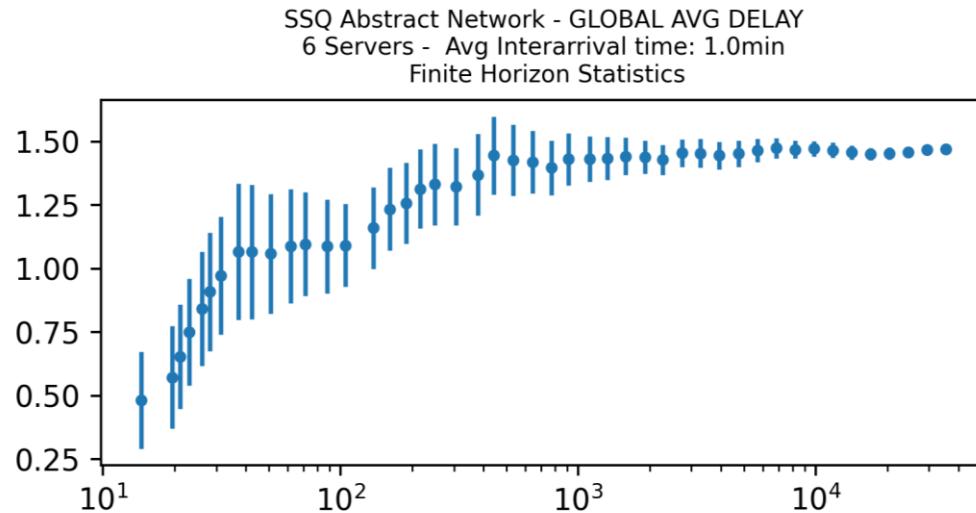
	SEED	MEAN VALUE	ERROR	CONFIDENCE LEVEL
Replica 0	123456789	6.836696051827544	± 0.32117083141245006	95%
Replica 1	491113262	7.162587748725365	± 0.3977409317414649	95%
Replica 2	1954995094	7.48962428788413	± 0.37542767300730034	95%
Replica 3	996641734	6.941519532545958	± 0.3278825336453585	95%
Replica 4	768437534	6.899672878626164	± 0.4698409556344268	95%

	SEED	MEAN VALUE	ERROR	CONFIDENCE LEVEL
Replica 0	123456789	7.1431565723652986	± 0.32112739681749664	95%
Replica 1	1118987882	7.219573436966225	± 0.2913933520571782	95%
Replica 2	901760786	6.778185468317601	± 0.22115596591348172	95%
Replica 3	1910652730	6.870467732385949	± 0.2619309761162776	95%
Replica 4	773502992	7.0659972644530455	± 0.2529499892949309	95%

Metodo delle Repliche

- ▶ La tecnica della replicazione è utilizzata per generare stime indipendenti della stessa statistica transiente. Questa tecnica è stata applicata per l'analisi del comportamento transiente dei due sistemi (*finite horizon*).
 - Secondo la procedura standard, si effettuano run di simulazione consecutive, utilizzando ogni volta come *seed* iniziale lo stato del sistema raggiunto alla fine della replica precedente.
 - Vengono calcolate stime puntuali e intervallari ad ogni acquisizione, ognuna di queste corrispondente ad un certo numero di richieste processate.
- ▶ **Si riportano grafici e tavelle di esempio** utilizzati nella fase di simulazione del comportamento transiente, in cui si è utilizzato un numero di repliche pari a 64 per calcolare le stime volute.

Metodo delle Repliche



	MEAN VALUE	ERROR	CONFIDENCE LEVEL
n° JOB: 10	0.4826796313309042	± 0.19132495762587795	95%
n° JOB: 29	0.9728808302931556	± 0.23164891873015692	95%
n° JOB: 86	1.087708174914137	± 0.18528917102843248	95%
n° JOB: 256	1.3314901768654868	± 0.1610357949400461	95%
n° JOB: 763	1.396681155095154	± 0.10810549875405334	95%
n° JOB: 2279	1.4279555258149743	± 0.058490865701562575	95%
n° JOB: 6804	1.474474491401095	± 0.039886409695528764	95%
n° JOB: 20318	1.4533951478816076	± 0.02441813999244722	95%

	MEAN VALUE	ERROR	CONFIDENCE LEVEL
n° JOB: 10	0.307798698803628	± 0.14391648792572212	95%
n° JOB: 29	0.41866712209574164	± 0.12612588157885243	95%
n° JOB: 86	0.6380520371218337	± 0.14734861103404565	95%
n° JOB: 256	0.6757484383468371	± 0.10001331888313937	95%
n° JOB: 763	0.7363935967504323	± 0.08156231843937194	95%
n° JOB: 2279	0.7238304673821097	± 0.04227572370684032	95%
n° JOB: 6804	0.7179277828020683	± 0.023715265634804486	95%
n° JOB: 20318	0.7144879825786478	± 0.013178767938955	95%

Esperimenti di Simulazione:

Analisi del comportamento transiente

- ▶ Seguono i grafici relativi al *comportamento transiente* dei due sistemi in esame.
- ▶ Attraverso questa analisi:
 - ❑ Si verifica l'**eventuale convergenza** ad uno stato stazionario, per i due sistemi.
 - ❑ Si **analizza e si confronta il comportamento transitorio dei due sistemi**, all'aumentare del numero di clienti serviti.
 - ❑ Per l'analisi del comportamento transitorio si è scelto di riportare:
 - ❑ Le osservazioni effettuate sui **tempi medi di attesa globali** dei due sistemi.
 - ❑ Le osservazioni effettuate sui **tempi medi d'attesa locali** al sistema avanzato

Esperimenti di Simulazione:

Ricerca delle configurazioni

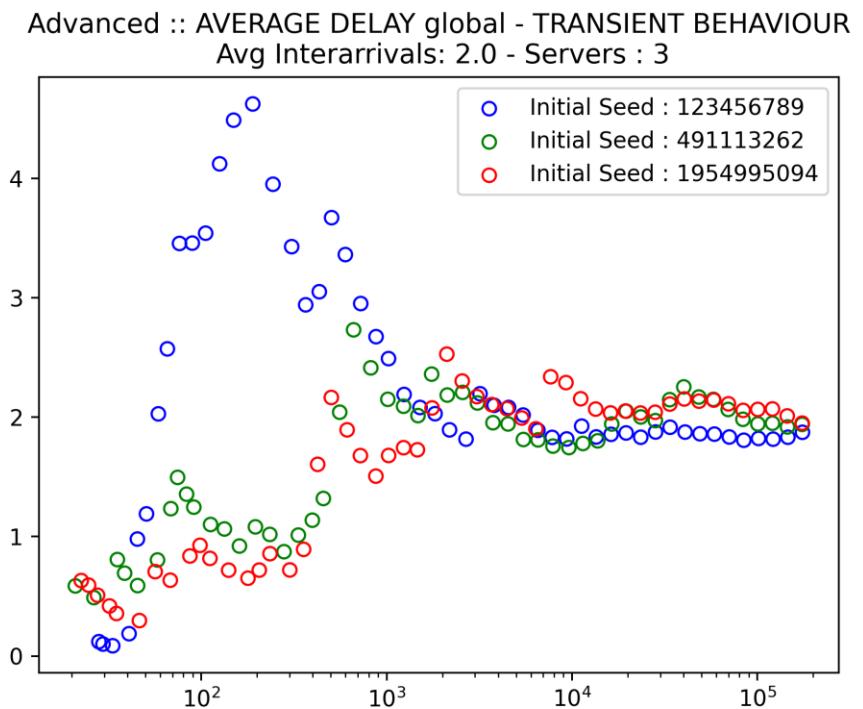
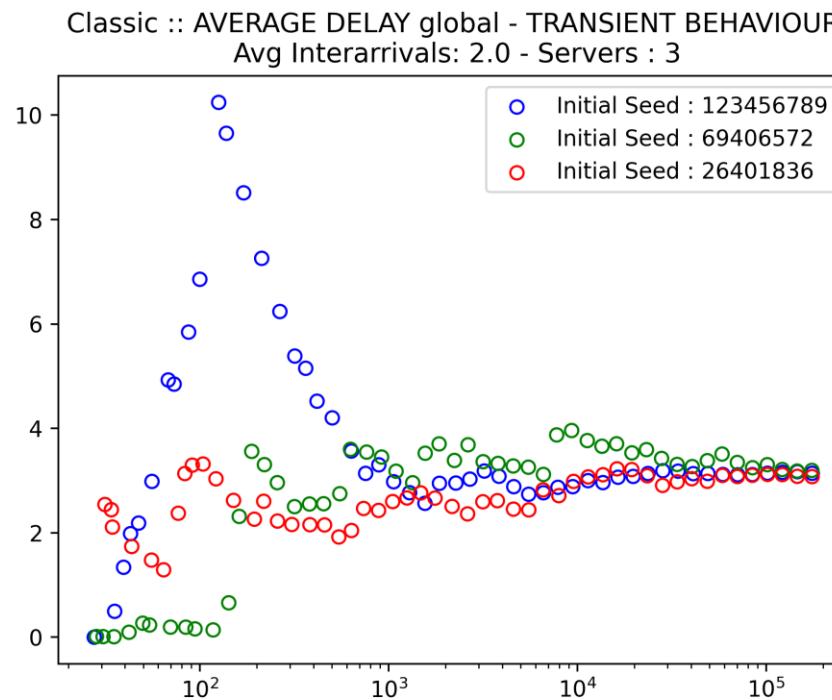
- ▶ Un esperimento di simulazione ha previsto la ricerca, per ogni fascia oraria e quindi al variare del parametro *tempo medio di inter-arrivo*, del numero esatto di serventi che permettesse di mantenere l'utilizzazione dei vari centri compresa tra il valore 0.6 e 0.8.
- ▶ Una “*Configurazione*” è definita dalla coppia
 - ▶ Tempo medio di inter-arrivo
 - ▶ Numero di Serventi/casse operativi/e
- ▶ Configurazioni ideali ($p = 0.74$)



FASCIA ORARIA	INTERARRIVO [min]	#SERVER
08.00 – 10.00	2.0	3
10.00 – 13.00	1.5	4
13.00 – 16.00	2.0	3
16.00 – 20.00	1.0	6
20.00 – 22.00	3.0	2

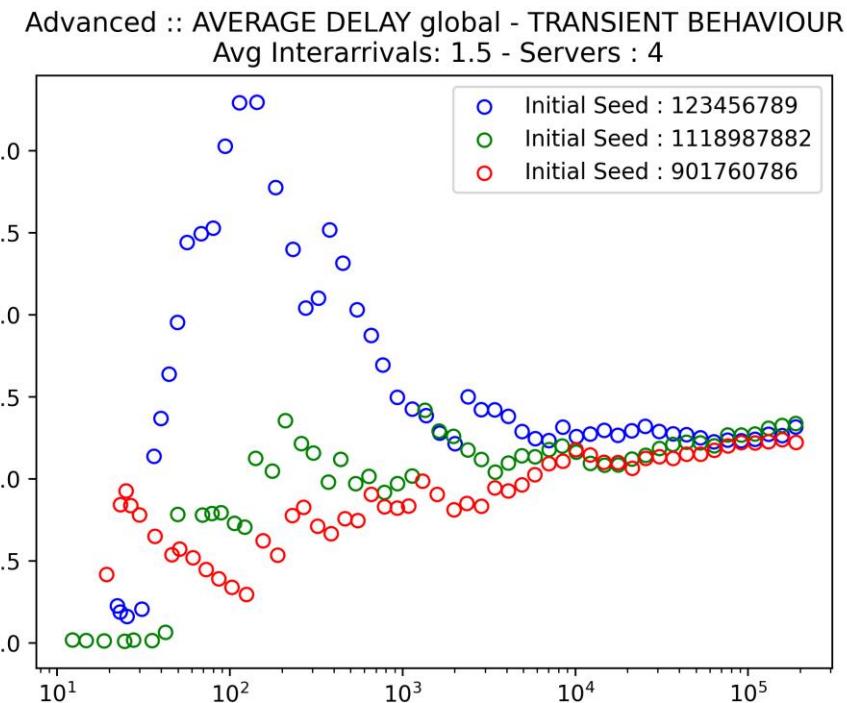
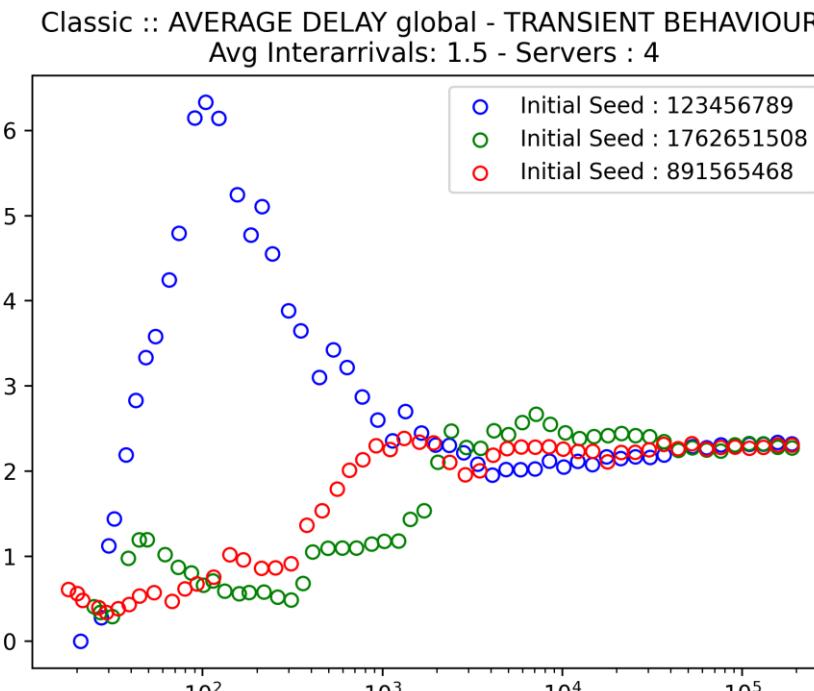
Analisi del comportamento transiente: Configurazione 1 e 3, Tempo medio di attesa globale.

FASCIA ORARIA	INTERARRIVO [min]	#SERVER
08.00 – 10.00	2.0	3
10.00 – 13.00	1.5	4
13.00 – 16.00	2.0	3
16.00 – 20.00	1.0	6
20.00 – 22.00	3.0	2



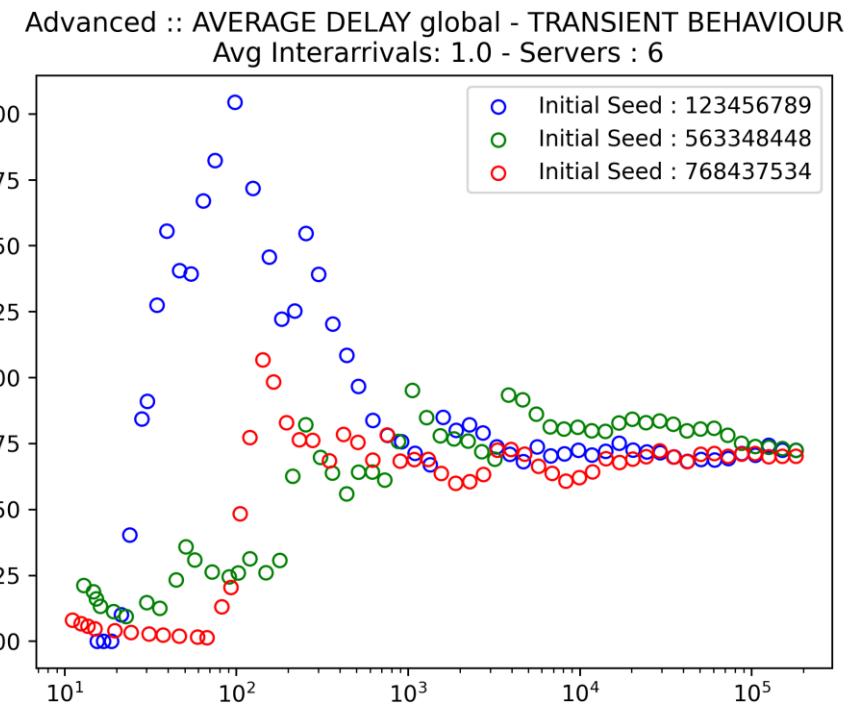
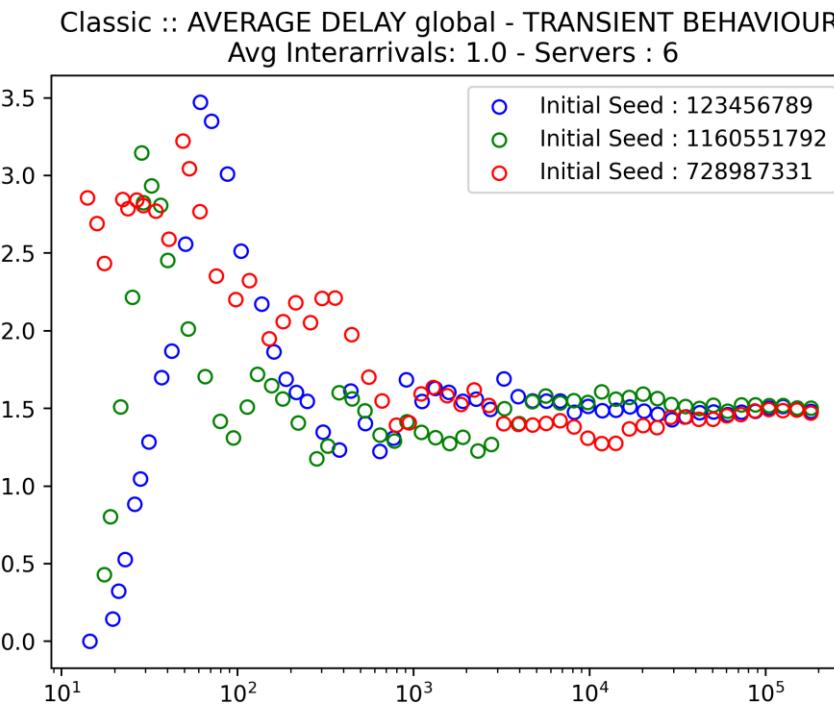
Analisi del comportamento transiente: Configurazione 2, Tempo medio di attesa globale.

FASCIA ORARIA	INTERARRIVO [min]	#SERVER
08.00 – 10.00	2.0	3
10.00 – 13.00	1.5	4
13.00 – 16.00	2.0	3
16.00 – 20.00	1.0	6
20.00 – 22.00	3.0	2



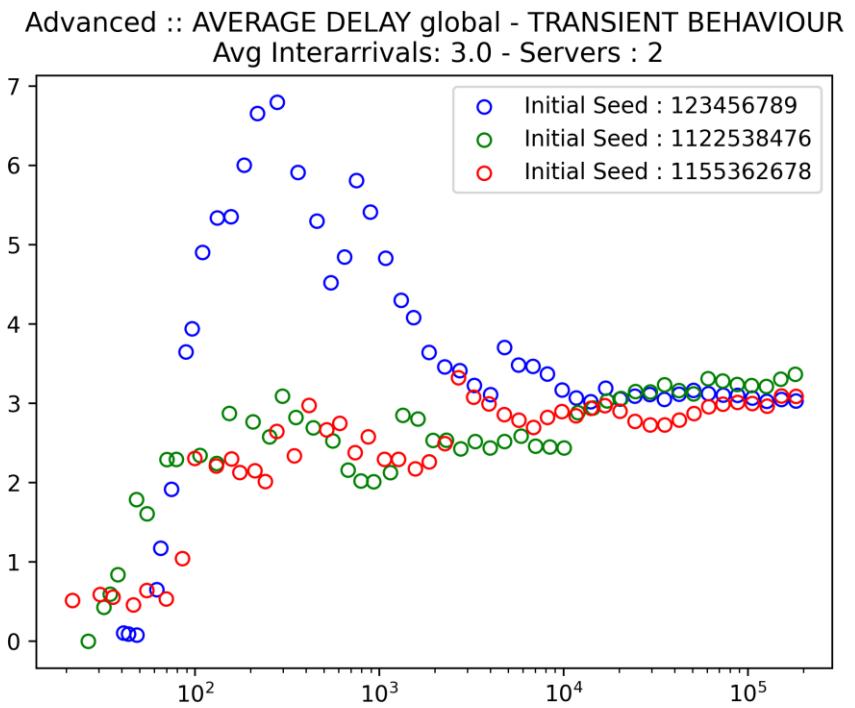
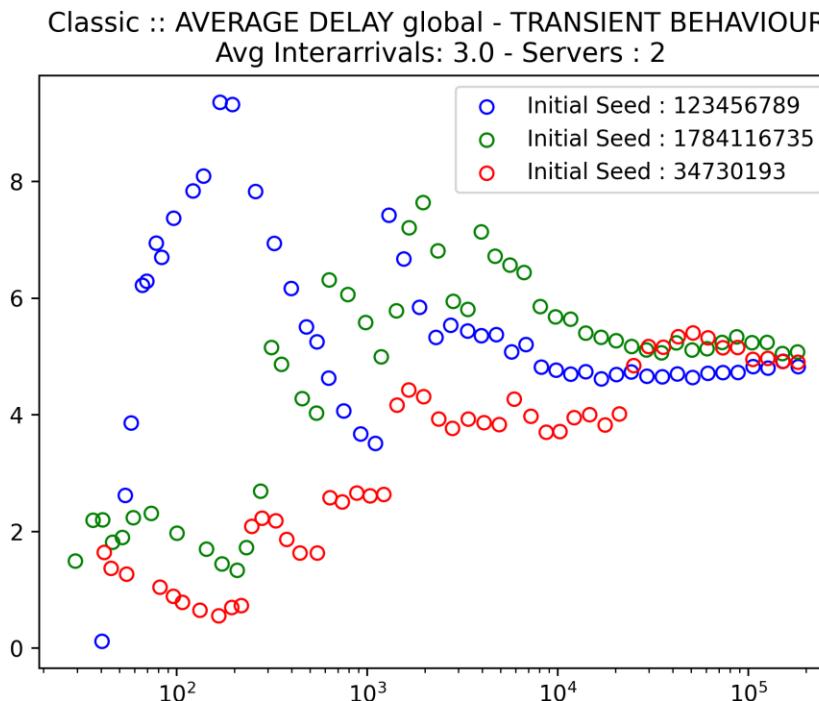
Analisi del comportamento transiente: Configurazione 4, Tempo medio di attesa globale.

FASCIA ORARIA	INTERARRIVO [min]	#SERVER
08.00 – 10.00	2.0	3
10.00 – 13.00	1.5	4
13.00 – 16.00	2.0	3
16.00 – 20.00	1.0	6
20.00 – 22.00	3.0	2



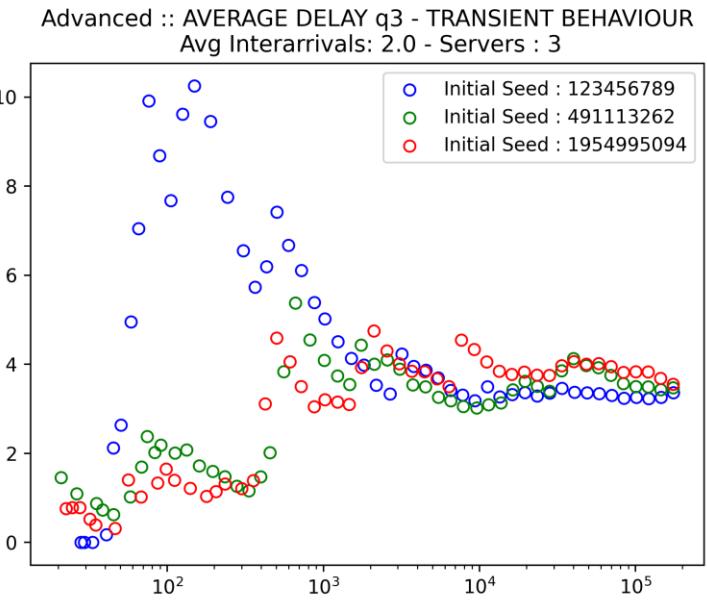
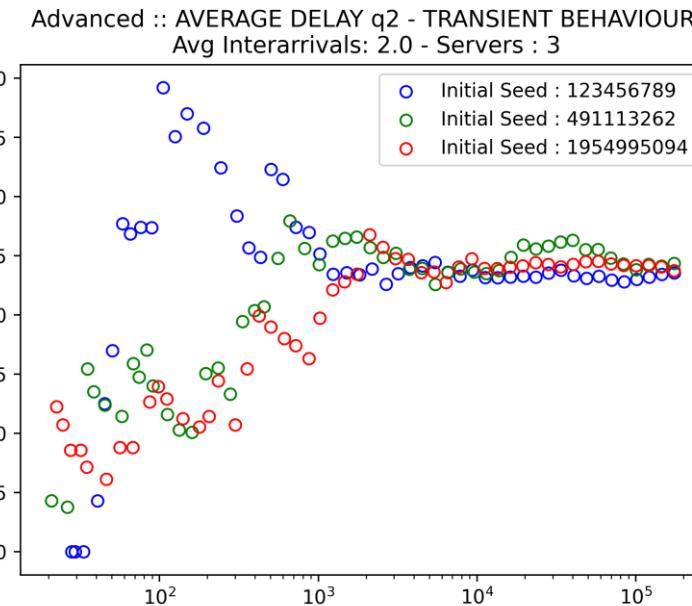
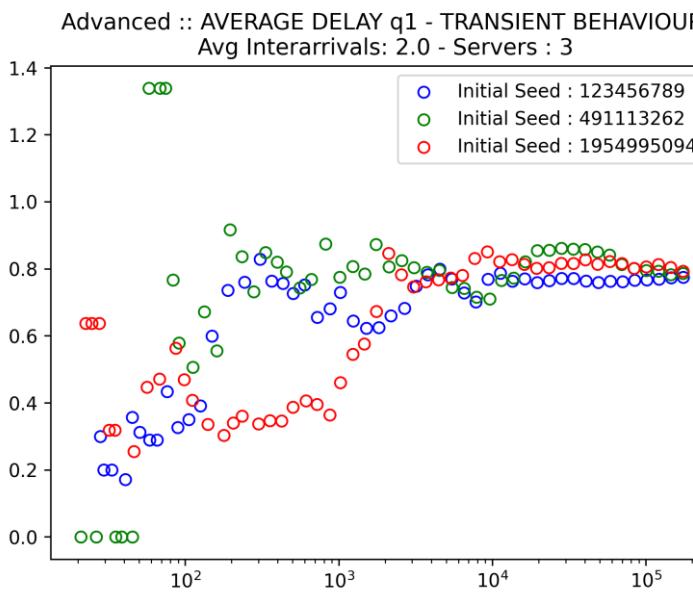
Analisi del comportamento transiente: Configurazione 5, Tempo medio di attesa globale.

FASCIA ORARIA	INTERARRIVO [min]	#SERVER
08.00 – 10.00	2.0	3
10.00 – 13.00	1.5	4
13.00 – 16.00	2.0	3
16.00 – 20.00	1.0	6
20.00 – 22.00	3.0	2



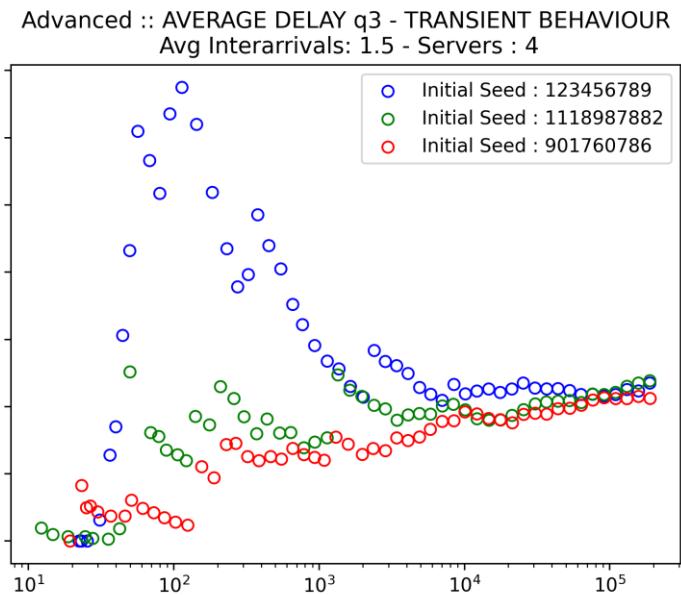
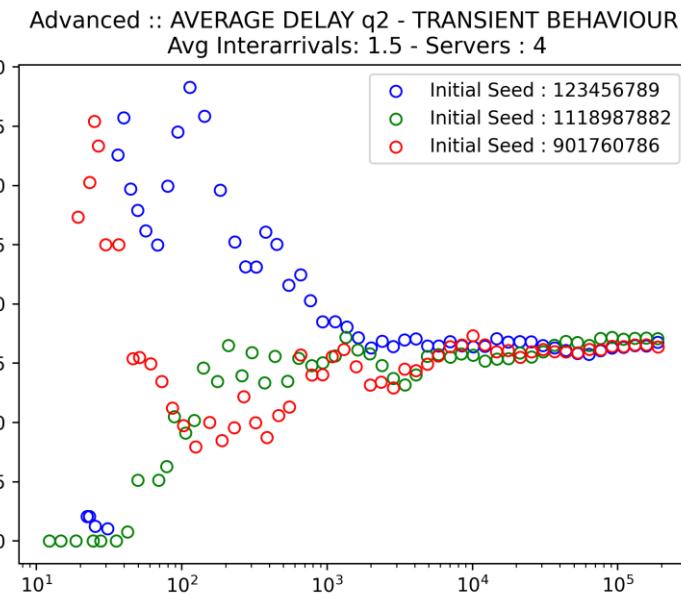
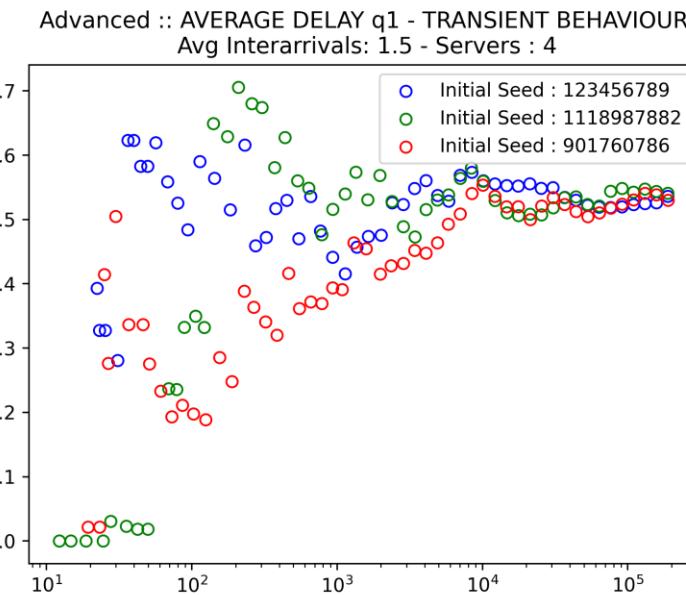
Analisi del comportamento transiente: Configurazione 1 e 3, Tempi medi di attesa locali.

FASCIA ORARIA	INTERARRIVO [min]	#SERVER
08.00 – 10.00	2.0	3
10.00 – 13.00	1.5	4
13.00 – 16.00	2.0	3
16.00 – 20.00	1.0	6
20.00 – 22.00	3.0	2



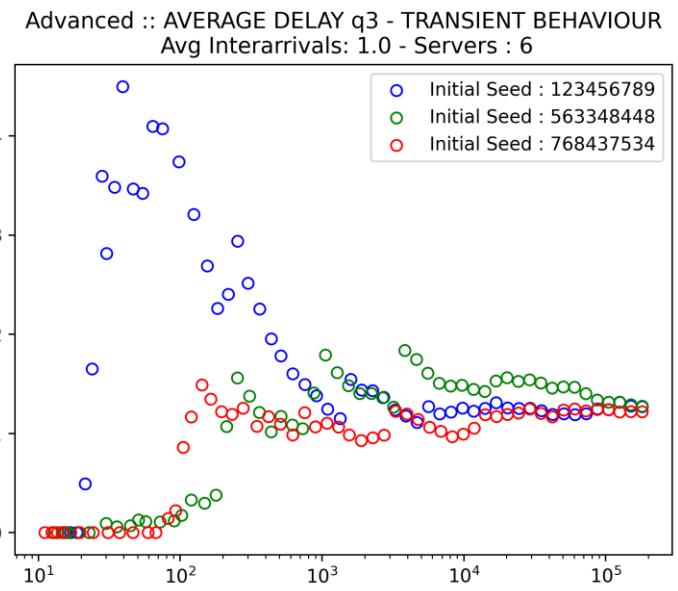
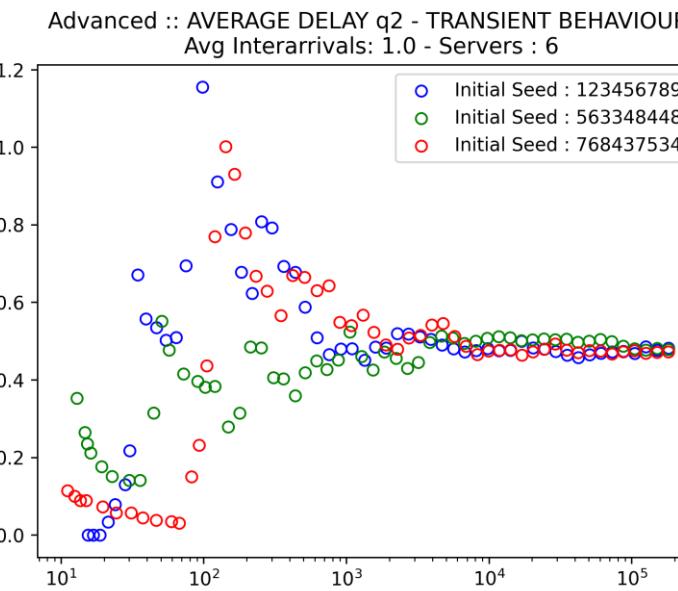
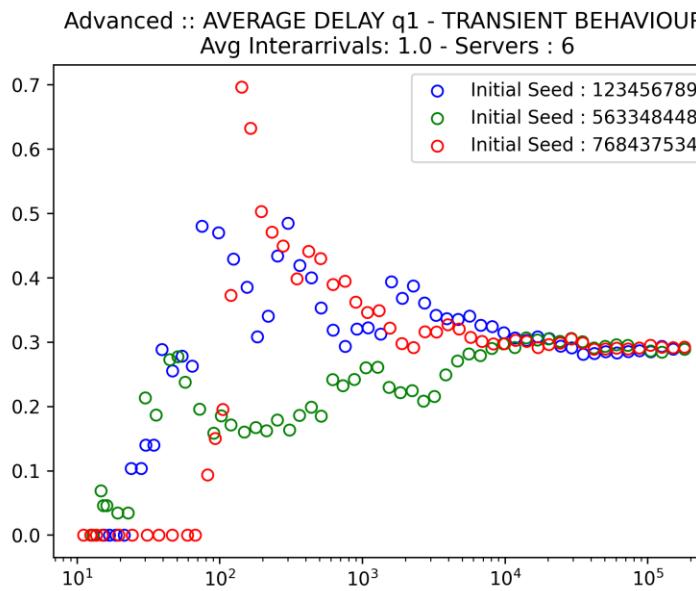
Analisi del comportamento transiente: Configurazione 2, Tempi medi di attesa locali.

FASCIA ORARIA	INTERARRIVO [min]	#SERVER
08.00 – 10.00	2.0	3
10.00 – 13.00	1.5	4
13.00 – 16.00	2.0	3
16.00 – 20.00	1.0	6
20.00 – 22.00	3.0	2



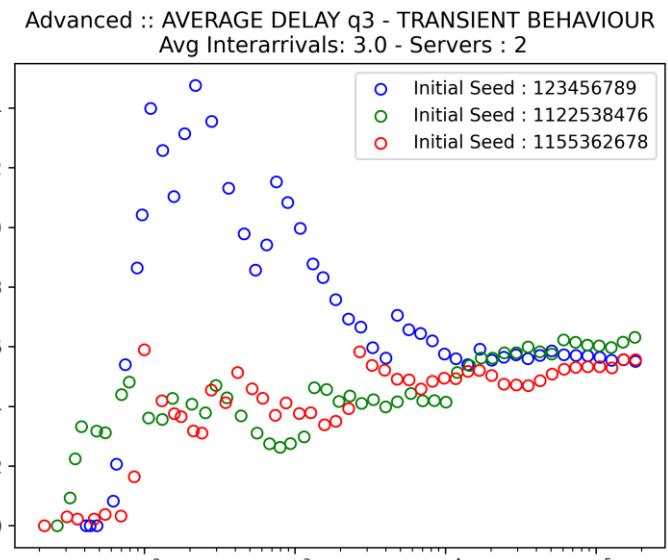
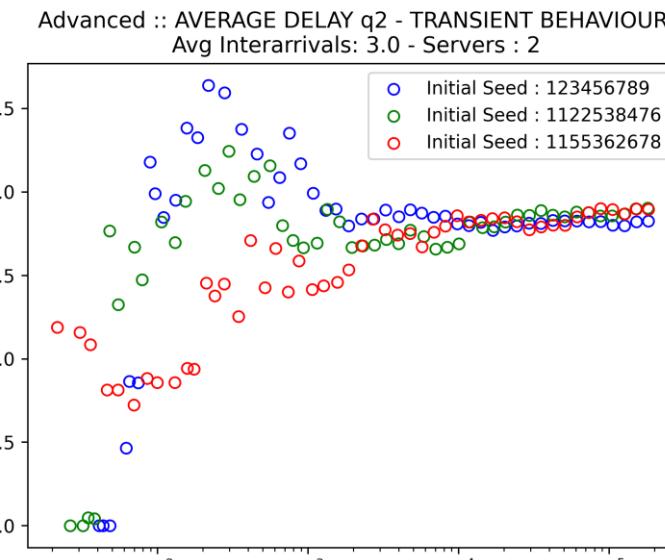
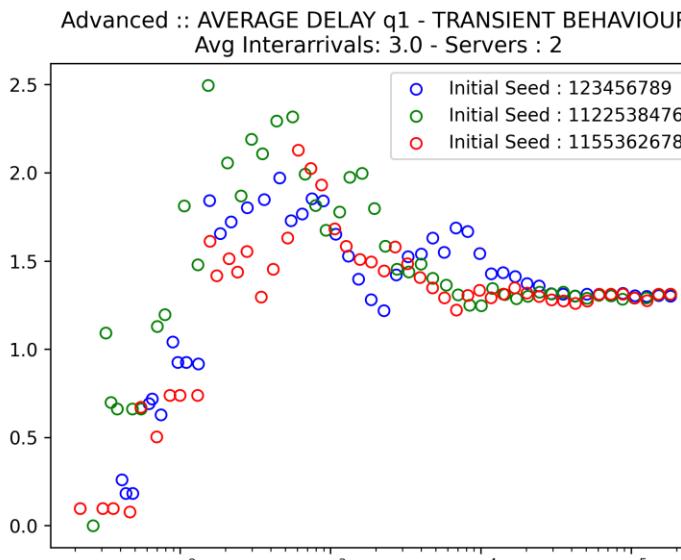
Analisi del comportamento transiente: Configurazione 4, Tempi medi di attesa locali.

FASCIA ORARIA	INTERARRIVO [min]	#SERVER
08.00 – 10.00	2.0	3
10.00 – 13.00	1.5	4
13.00 – 16.00	2.0	3
16.00 – 20.00	1.0	6
20.00 – 22.00	3.0	2



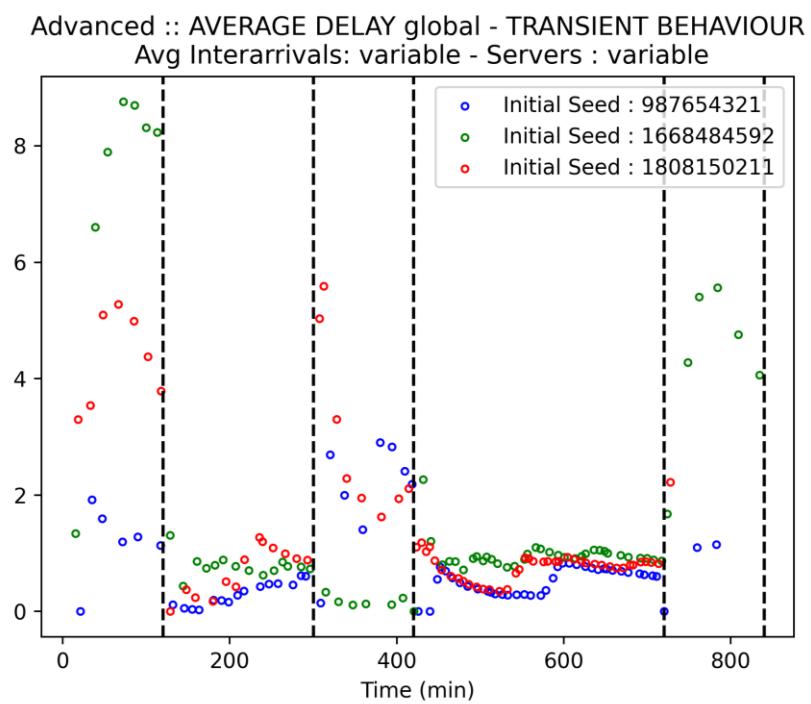
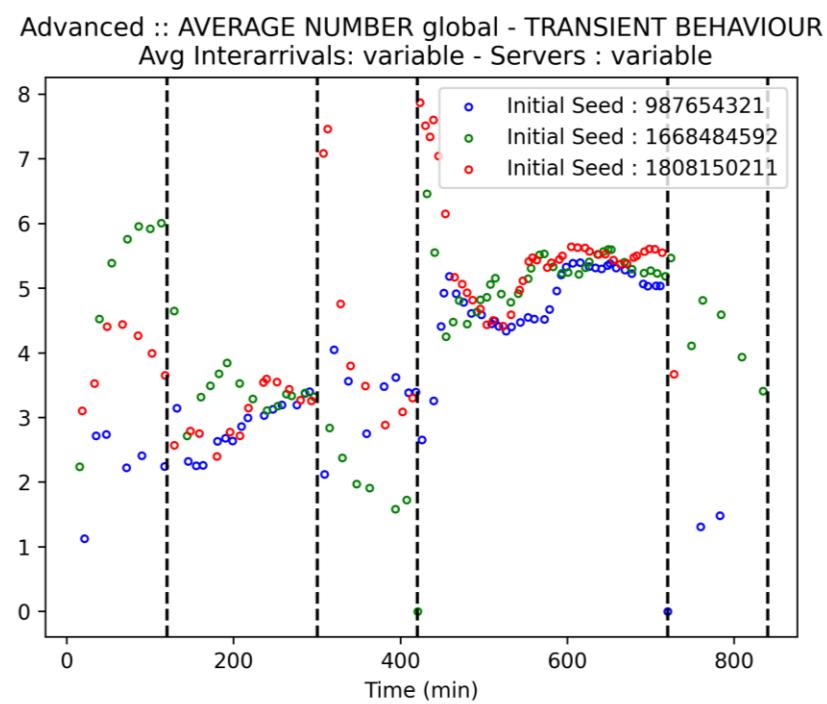
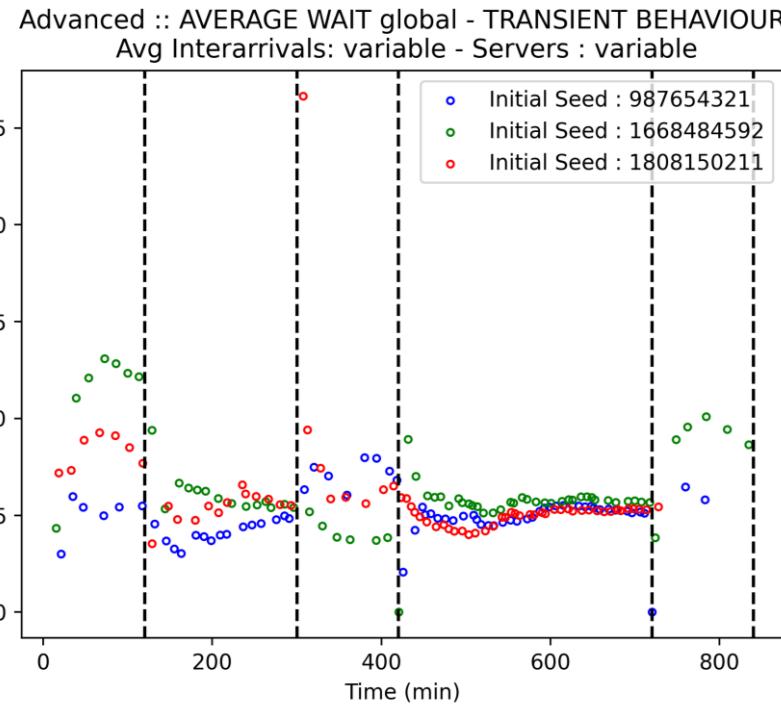
Analisi del comportamento transiente: Configurazione 5, Tempi medi di attesa locali.

FASCIA ORARIA	INTERARRIVO [min]	#SERVER
08.00 – 10.00	2.0	3
10.00 – 13.00	1.5	4
13.00 – 16.00	2.0	3
16.00 – 20.00	1.0	6
20.00 – 22.00	3.0	2



Analisi del comportamento transiente: Andamento delle prestazioni globali nelle varie fasce orarie.

FASCIA ORARIA	INTERARRIVO [min]	#SERVER
08.00 – 10.00	2.0	3
10.00 – 13.00	1.5	4
13.00 – 16.00	2.0	3
16.00 – 20.00	1.0	6
20.00 – 22.00	3.0	2

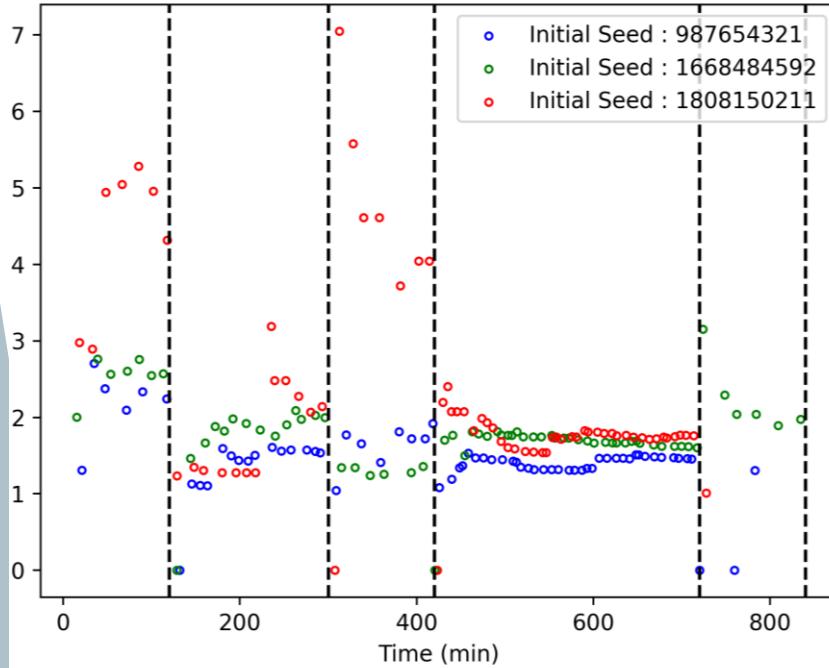


Analisi del comportamento transiente:

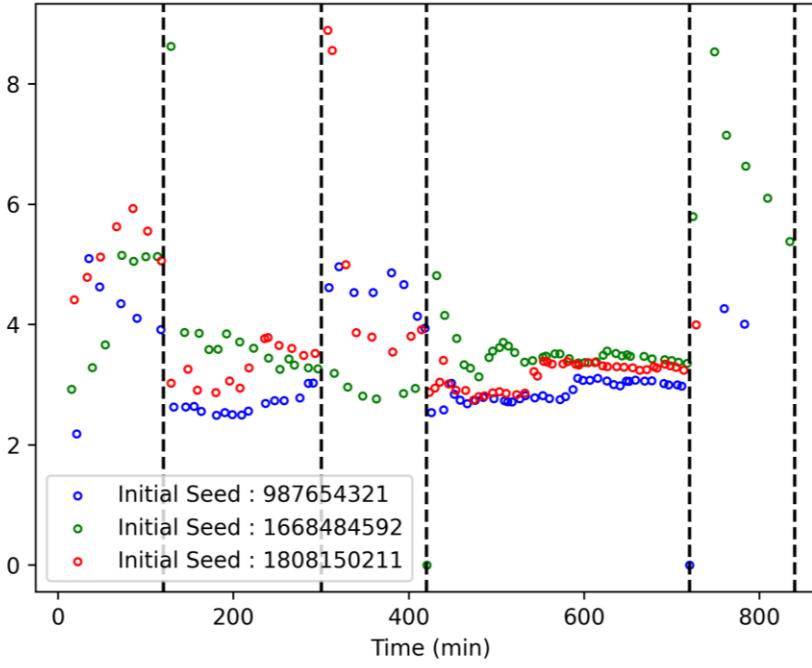
Andamento delle prestazioni locali nelle varie fasce orarie.

FASCIA ORARIA	INTERARRIVO [min]	#SERVER
08.00 – 10.00	2.0	3
10.00 – 13.00	1.5	4
13.00 – 16.00	2.0	3
16.00 – 20.00	1.0	6
20.00 – 22.00	3.0	2

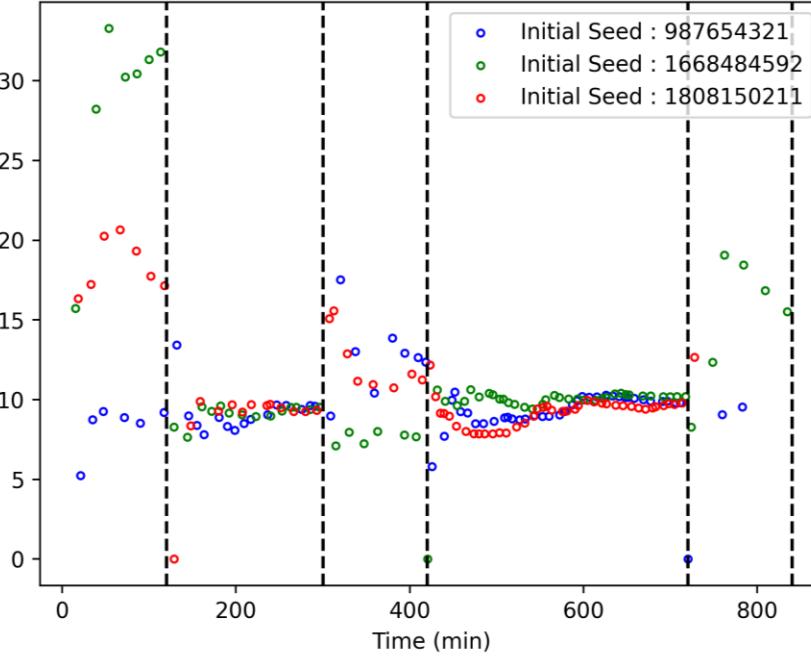
Advanced :: AVERAGE WAIT q1 - TRANSIENT BEHAVIOUR
Avg Interarrivals: variable - Servers : variable



Advanced :: AVERAGE WAIT q2 - TRANSIENT BEHAVIOUR
Avg Interarrivals: variable - Servers : variable



Advanced :: AVERAGE WAIT q3 - TRANSIENT BEHAVIOUR
Avg Interarrivals: variable - Servers : variable

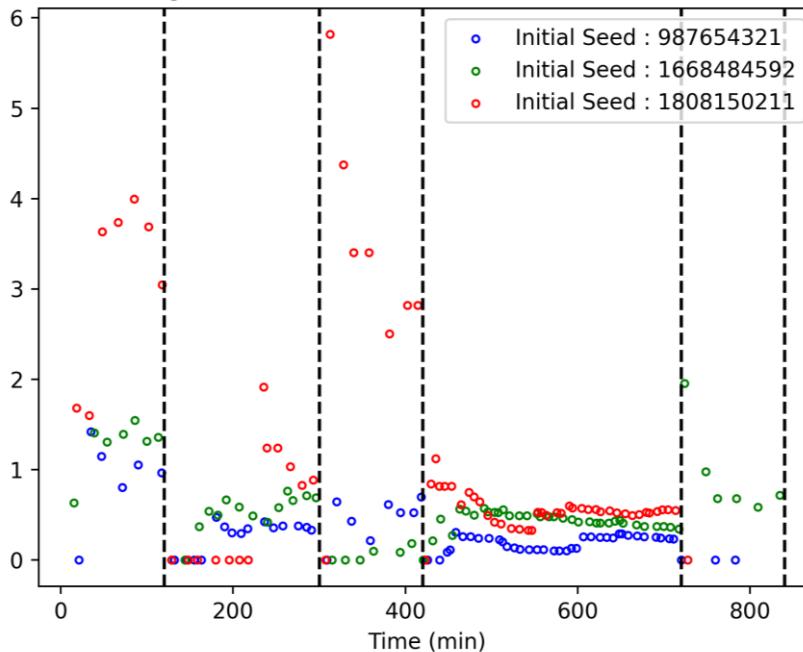


Analisi del comportamento transiente:

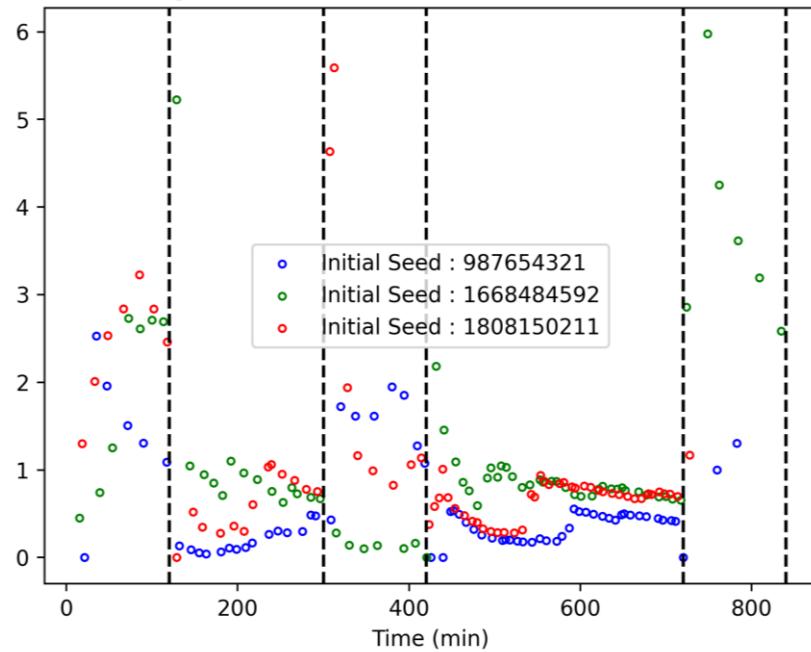
Andamento delle prestazioni locali nelle varie fasce orarie.

FASCIA ORARIA	INTERARRIVO [min]	#SERVER
08.00 – 10.00	2.0	3
10.00 – 13.00	1.5	4
13.00 – 16.00	2.0	3
16.00 – 20.00	1.0	6
20.00 – 22.00	3.0	2

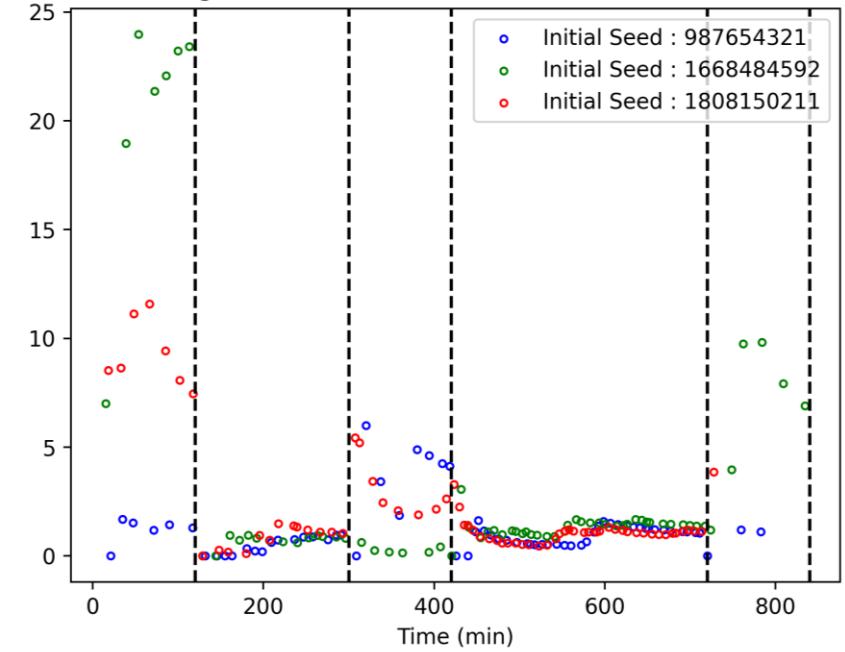
Advanced :: AVERAGE DELAY q1 - TRANSIENT BEHAVIOUR
Avg Interarrivals: variable - Servers : variable



Advanced :: AVERAGE DELAY q2 - TRANSIENT BEHAVIOUR
Avg Interarrivals: variable - Servers : variable



Advanced :: AVERAGE DELAY q3 - TRANSIENT BEHAVIOUR
Avg Interarrivals: variable - Servers : variable



Conclusioni: Prestazioni Globali

CONFIGURAZIONE CLASSIC	INTERARRIVO	NODES	GLOBAL DELAY	GLOBAL NUMBER	GLOBAL WAIT
1/3	2.0 min	3	3.20 +- 0.17	3.83 +- 0.10	7.62 +- 0.17
2	1,5 min	4	2.46 +- 0.13	4.63 +- 0.11	6.81 +- 0.12
4	1,0 min	6	1.51 +- 0.08	6.08 +- 0.12	5.97 +- 0.10
5	3,0 min	2	4.93 +- 0.20	3.11 +- 0.09	9.36 +- 0.20

CONFIGURAZIONE ADVANCED	INTERARRIVO	SERVERS	GLOBAL DELAY	GLOBAL NUMBER	GLOBAL WAIT
1/3	2 min	3	1,90 +- 0,11	3,15 +- 0,07	6,35 +- 0,13
2	1,5 min	4	1,31 +- 0,09	3,85 +- 0,08	5,75 +- 0,10
4	1,0 min	6	0,72 +- 0,05	5,15 +- 0,86	5,15 +- 0,07
5	3,0 min	2	3,10 +- 0,18	2,55 +- 0,07	7,65 +- 0,18

- ❑ Il sistema avanzato migliora il sistema classico dal punto di vista delle statistiche globali.
- ❑ La differenza più accentuata, allo stato stazionario, si apprezza sul tempo di attesa medio globale.
 - La policy di scheduling size based incide perlopiù sul tempo di attesa
- ❑ Per quanto questo possa sembrare un miglioramento marginale, c'è da dire che si tratta di un risultato ottenibile **"a costo zero"**, con una **semplice riorganizzazione del sistema di casse** del supermercato

Conclusioni: Prestazioni Locali

CONFIGURAZIONE	INTERARRIVO	SERVERS	DELAY 1	WAIT 1
1/3	2 min	3	0,79 +- 0,253	2,02 +- 0,026
2	1,5 min	4	0,54 +- 0,02	1,76 +- 0,196
4	1,0 min	6	0,29 +- 0,141	1,52 +- 0,1446
5	3,0 min	2	1,325 +- 0,038	2,550 +- 0,037
CONFIGURAZIONE	INTERARRIVO	SERVERS	DELAY 2	WAIT 2
1/3	2 min	3	1,175 +- 0,4	3,850 +- 0,43
2	1,5 min	4	0,82 +- 0,03	3,48 +- 0,0325
4	1,0 min	6	0,47 +- 0,215	3,13 +- 0,233
5	3,0 min	2	1,87 +- 0,05	4,52 +- 0,052
CONFIGURAZIONE	INTERARRIVO	SERVERS	DELAY 3	WAIT 3
1/3	2 min	3	3,4 +- 0,25	11,9 +- 0,258
2	1,5 min	4	2,3 +- 0,197	10,8 +- 0,206
4	1,0 min	6	1,25 +- 0,12	9,7 +- 0,124
5	3,0 min	2	5,75 +- 0,41	14,25 +- 0,42

Tempo di risposta medio

- Il 65% della clientela guadagna in tempo di risposta di almeno un fattore 2 rispetto al modello classico. Il restante 35% va invece a perdere, ma resta «comparabile»
- Il 20% della clientela guadagna in tempo di risposta di almeno un fattore 3.

Tempo di attesa medio

- Il 65% della clientela guadagna in tempo di attesa di almeno un fattore 2.5 rispetto al modello classico. Il restante 35% rimane comparabile.
- Il 20% della clientela guadagna in tempo di attesa di un fattore 3.5

CONFIGURAZIONE CLASSIC	INTERARRIVO	NODES	GLOBAL DELAY	GLOBAL WAIT
1/3	2.0 min	3	3.20 +- 0.17	7.62 +- 0.17
2	1,5 min	4	2.46 +- 0.13	6.81 +- 0.12
4	1,0 min	6	1.51 +- 0.08	5.97 +- 0.10
5	3,0 min	2	4.93 +- 0.20	9.36 +- 0.20

Conclusioni: Fairness

SIZE X [min] CLASSIC	MEAN CONDITIONAL SLOWDOWN	SERVERS	INTERARRIVAL
1.24	8.60 +-0.20	2	3
2.65	4.50+-0.09	2	3
4.42	4.10+-0.06	2	3
8.26	2.13+-0.03	2	3

SIZE X [min] ADVANCED	MEAN CONDITIONAL SLOWDOWN	SERVERS	INTERARRIVAL
1.24	2.06+-0.03	2	3
2.65	1.70+-0.02	2	3
4.42	1.42+-0.01	2	3
8.26	1.70+-0.05	2	3

Sistema Classico

- Il rallentamento sperimentato dai clienti con richieste piccole è di gran lunga superiore rispetto a quello sperimentato dai clienti con richieste più grandi.
- Questo comportamento è dovuto alla politica di scheduling astratta che è tipicamente adottata alle casse dei supermercati.

Sistema Avanzato

- Bassa variabilità dei rallentamenti tra le varie richieste di servizio nel modello avanzato.
- Grazie alla politica di scheduling size-based, riesce ad implementare la fairness che si cercava, adeguando il rallentamento di ogni cliente in base all'entità della sua richiesta.

Grazie per
l'attenzione!!!

