

**PRESENTAZIONE PROGETTO**  
**PMCSN 2018/19**  
**TRACCIA: B2**

Di Somma Alessia  
Nedia Salvatore

# MODELLO CONCETTUALE — ALGORITMO 1

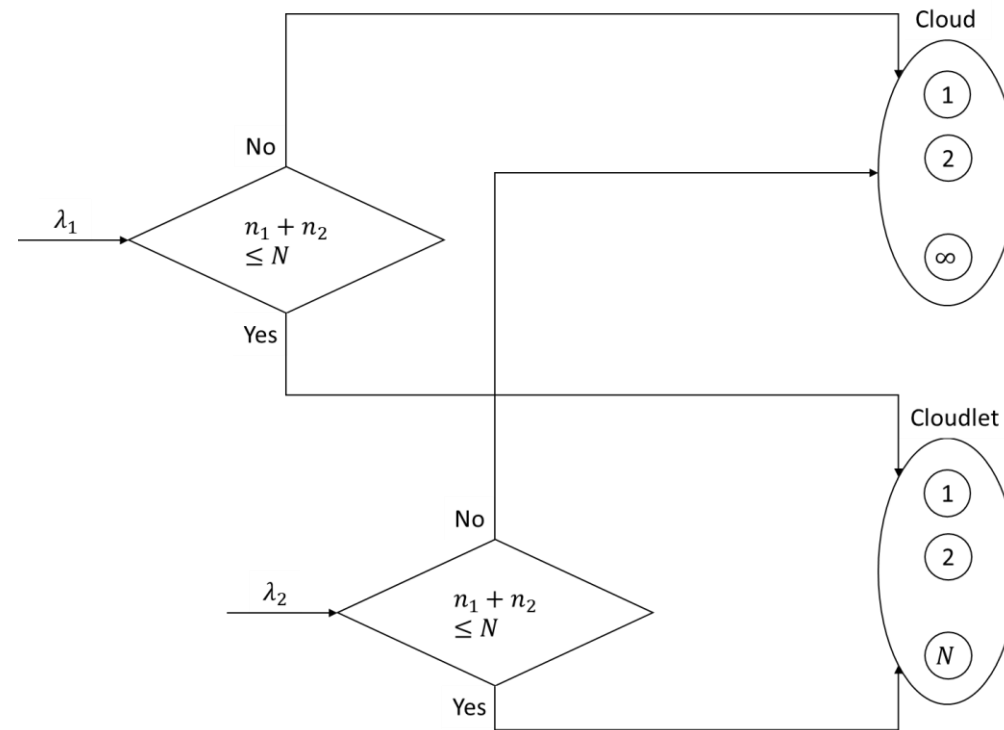
Controllore

Tipo di server e code

- Cloudlet:  $M/H_2/N/N$
- Cloud:  $G/M/\infty$

Stato del sistema

- $N$  = numero job nel cloudlet
  - Dove  $N = n_1 + n_2$



# SCELTA ALGORITMO 2

```
Algoritmo 2;  
Arrivi di classe 1;  
if  $n1=N$  then  
    | send on cloud;  
end  
if  $n1+n2<S$  then  
    | accept task on cloudlet;  
    if  $n2 > 0$  then  
        | preemption;  
    else  
        | accept task on cloudlet;  
    end  
end  
Arrivi di classe 1;  
if  $n1+n2\geq S$  then  
    | send on cloud;  
else  
    | accept task on cloudlet;  
end
```

Obiettivo: migliorare il tempo di risposta globale

Esecuzione task classe 1 nel cloudlet più conveniente di quella dei task di classe 2

Scelta utilizzare la prelazione

Migrazione task classe 2 dal cloudlet al cloud per favorire l'ingresso di uno di classe 1

Utilizzando valore di soglia  $S$  ( $S=20$ )

# MODELLO CONCETTUALE — ALGORITMO 2

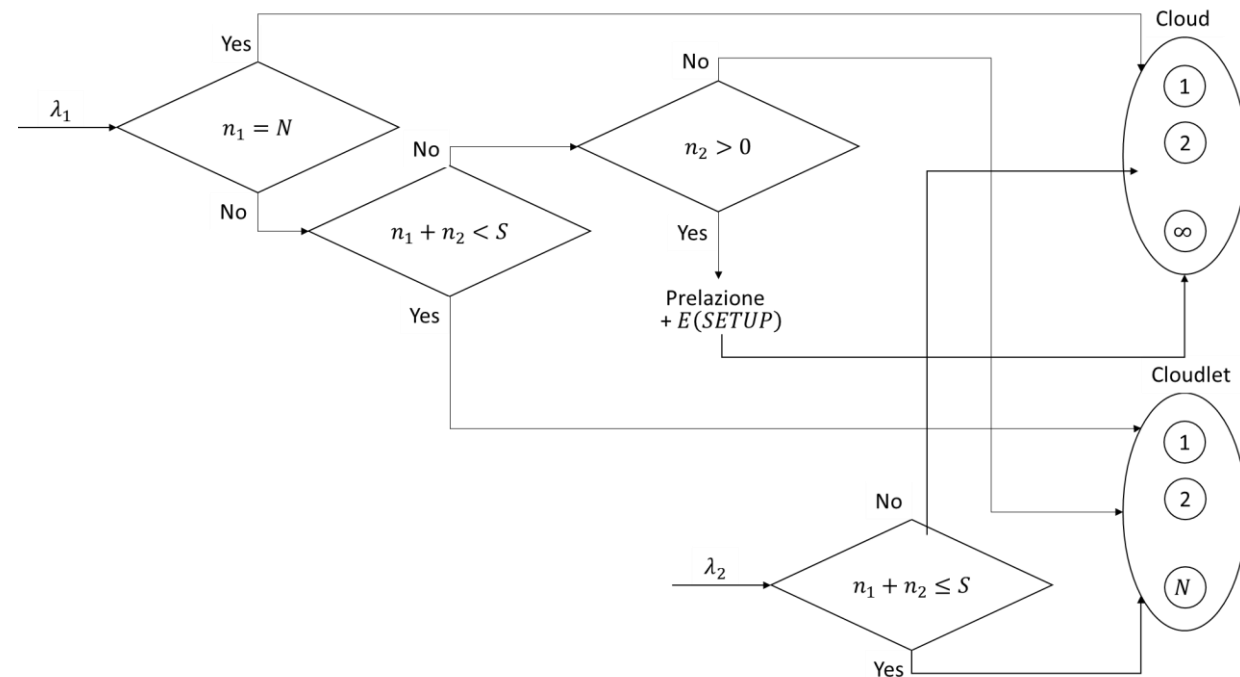
Controllore

Tipo di serveri e code

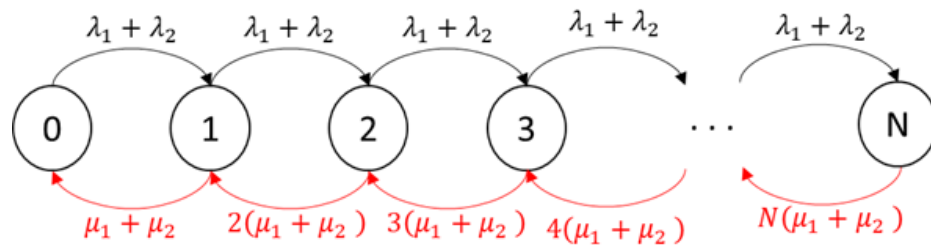
- Cloudlet:  $M/H_2/N/N$
- Cloud:  $G/M/\infty$

Stato del sistema

- $n_1, n_2, N$



# MODELLO ANALITICO – ALGORITMO 1

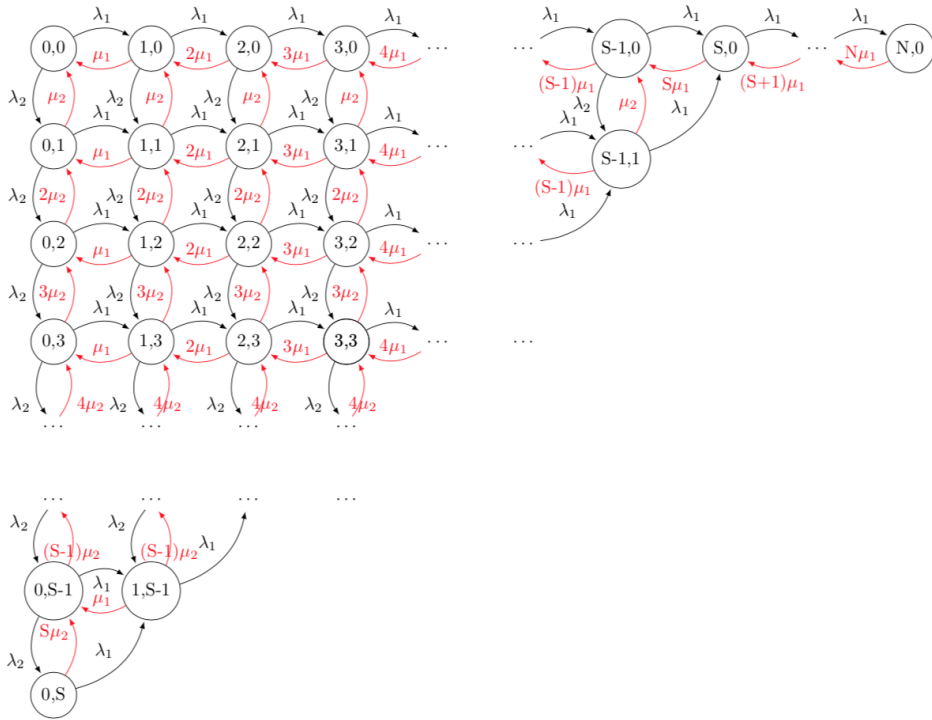


$$\begin{cases} \Pi_0(\lambda_1 + \lambda_2) = \Pi_1(\mu_1 + \mu_2) \\ \dots \\ \Pi_i(\lambda_1 + \lambda_2) = \Pi_{i+1}(i+1)(\mu_1 + \mu_2) \\ \dots \\ \Pi_{N-1}(\lambda_1 + \lambda_2) = \Pi_N N(\mu_1 + \mu_2) \\ \sum_{i=0}^N \Pi_i = 1 \end{cases} \Rightarrow \begin{cases} \Pi_0 = \frac{1}{\sum_{i=0}^N \left(\frac{\lambda}{\mu}\right)^i \left(\frac{1}{i!}\right)} \\ \dots \\ \Pi_i = \frac{\Pi_0(\lambda_1 + \lambda_2)^i}{i!(\mu_1 + \mu_2)^i} \\ \dots \\ \Pi_N = \frac{\Pi_0(\lambda_1 + \lambda_2)^N}{N!(\mu_1 + \mu_2)^N} \end{cases}$$

$$p_{cloudlet} = 1 - \Pi_{20}$$

$$p_{cloud} = \Pi_{20}$$

# MODELLO ANALITICO – ALGORITMO 2



Equazione generica dello stato:

$$\begin{aligned} & \Pi_{x,y}(a\lambda_1 + b\lambda_2 + y\mu_2 + x\mu_1) \\ &= \Pi_{x-1,y}(c\lambda_1) + \Pi_{x,y-1}(d\lambda_2) \\ &+ \Pi_{x+1,y}(e(x+1)\mu_1) + \Pi_{x,y+1}(f(x+1)\mu_2) \\ &+ \Pi_{x-1,y+1}(g\lambda_1) \end{aligned}$$

Numero degli stati :

$$numero_{stati} = N + 1 + \sum_{k=1}^S k = \frac{S(S+1)}{2} + N + 1$$

# MODELLO COMPUTAZIONALE

## Algoritmo 1

### Next-Event simulator

- Arrivo task di classe 1
- Arrivo task di classe 2
- Partenza da cloudlet
- Partenza da cloud

### Stato del sistema

- $N$

### Input

- $N$ , seed, tempo di simulazione, dimensione di batch, tassi di arrivo e completamento

## Algoritmo 2

### Next-Event simulator

- Arrivo task di classe 1
- Arrivo task di classe 2
- Partenza da cloudlet
- Partenza da cloud

### Stato del sistema

- $n1, n2, N$

### Input

- $N, S$ , seed, tempo di simulazione, dimensione di batch, tassi di arrivo e completamento

# METODOLOGIE UTILIZZATE

## Replicazione per verificare la stazionarietà:

- Run consecutivi
- Indipendenza dei seed garantita dalla consecutività delle simulazioni
- Il valore iniziale di ogni seed è l'ultimo numero pseudo-random generato nell'ultima simulazione

## Batch means:

- Usato per il calcolo delle statistiche a regime
- Con l'**algoritmo di Welford**
- Azzeramento delle statistiche parziali alla fine di ogni batch
- Evita il bias iniziale
- Parametri di simulazione
- Lunghezza del run: 86400 s
- Numero di batch: 24
  - In base alle linee guida
- Dimensione del batch: 3600 s



# VERIFICA E VALIDAZIONE

Simulatore Next-Event: Test per valori aspettati

## Algoritmo 1

	Valori teorici	Valori simulati
Tempo risposta globale (s)	3.625938	3.622958
Tempo risposta classe 1 (s)	2.959198	2.970106
Tempo risposta classe 2 (s)	4.05265	4.041308
Throughput globale (task/s)	10.25	10.249711
Throughput classe 1 (task/s)	4	4.003705
Throughput classe 2 (task/s)	6.25	6.246006

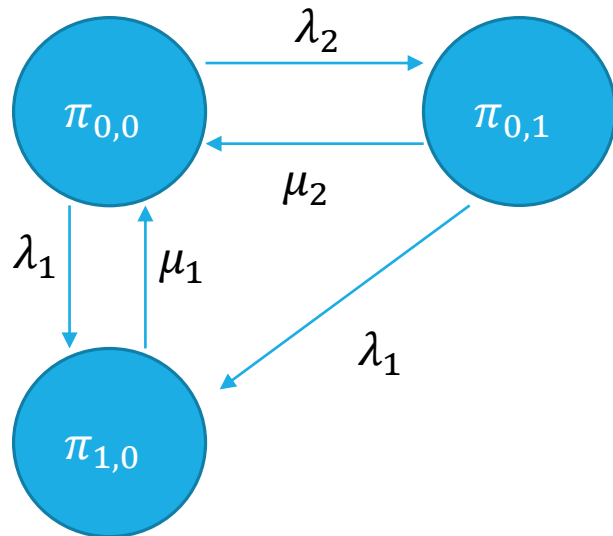
## Algoritmo 2

	Valori teorici	Valori simulati
Tempo risposta globale (s)	3.496811	3.519627
Tempo risposta classe 1 (s)	2.231252	2.222095
Tempo risposta classe (s)	4.509257	4.196501
Throughput globale (task/s)	10.25	10.231508
Throughput classe 1 (task/s)	4	3.989881
Throughput classe 2 (task/s)	6.25	6.241627

# VERIFICA E VALIDAZIONE – ALGORITMO 2

## Algoritmo 2:

- Generatore di matrice di transizione di stato
  - Verifica per  $N=1, S=1$  ( test manuale)



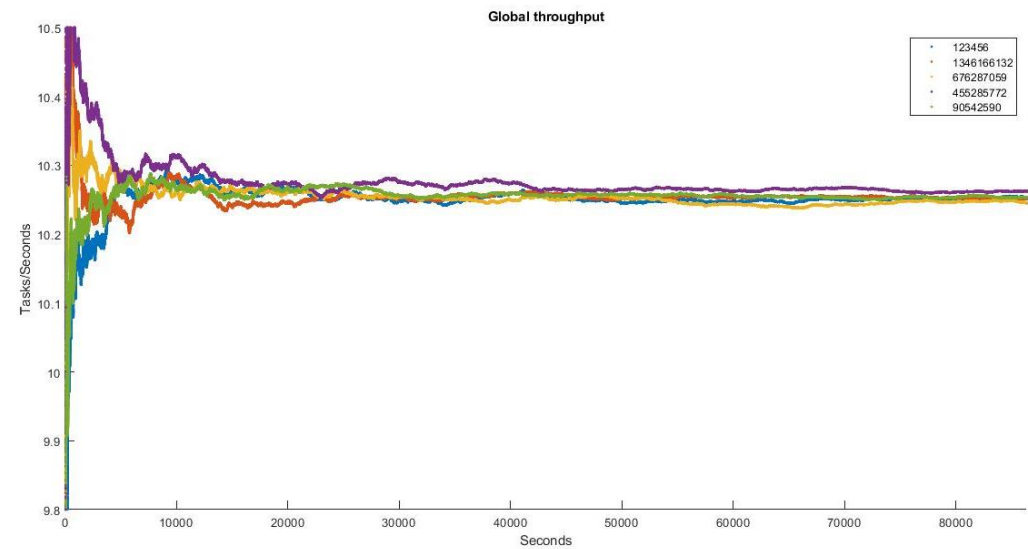
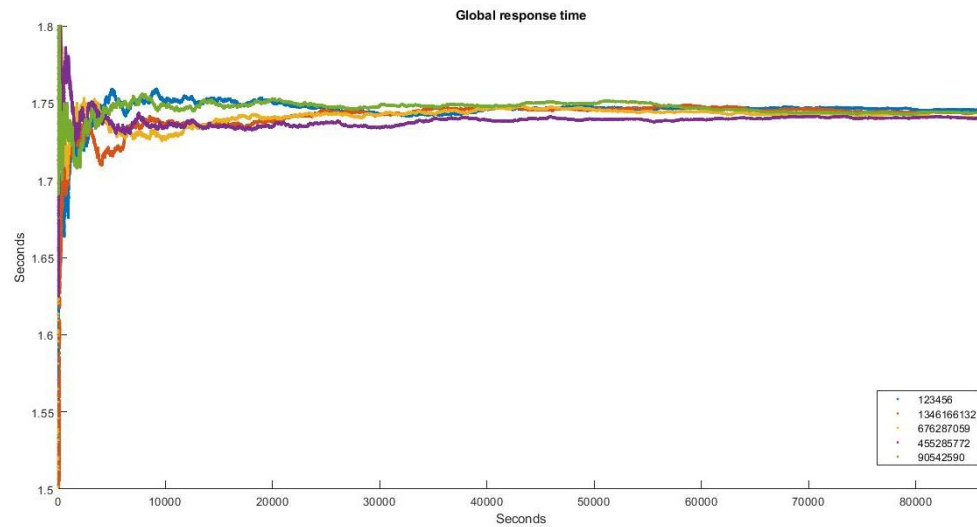
	$\pi_{00}$	$\pi_{10}$	$\pi_{01}$
$\pi_{00}$	0	$\lambda_1$	$\lambda_2$
$\pi_{10}$	$\mu_1$	0	0
$\pi_{01}$	$\mu_2$	$\lambda_1$	0

$$\text{Numero stati : } N + 1 + \sum_{k=1}^S k = \frac{S(S+1)}{2} + N + 1 = 3$$

$$\text{Equazioni : } \begin{cases} \Pi_{0,0}(\lambda_1 + \lambda_2) = \Pi_{1,0}(\mu_1) + \Pi_{0,1}(\mu_2) \\ \Pi_{1,0}\mu_1 = \Pi_{0,1}(\lambda_1) + \Pi_{0,0}(\lambda_1) \\ \Pi_{0,0} + \Pi_{1,0} + \Pi_{0,1} = 1 \end{cases}$$

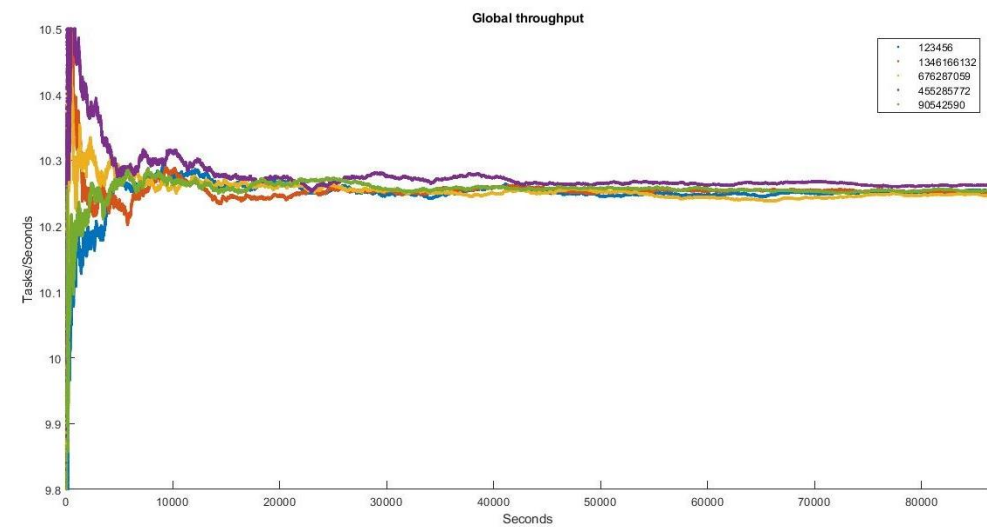
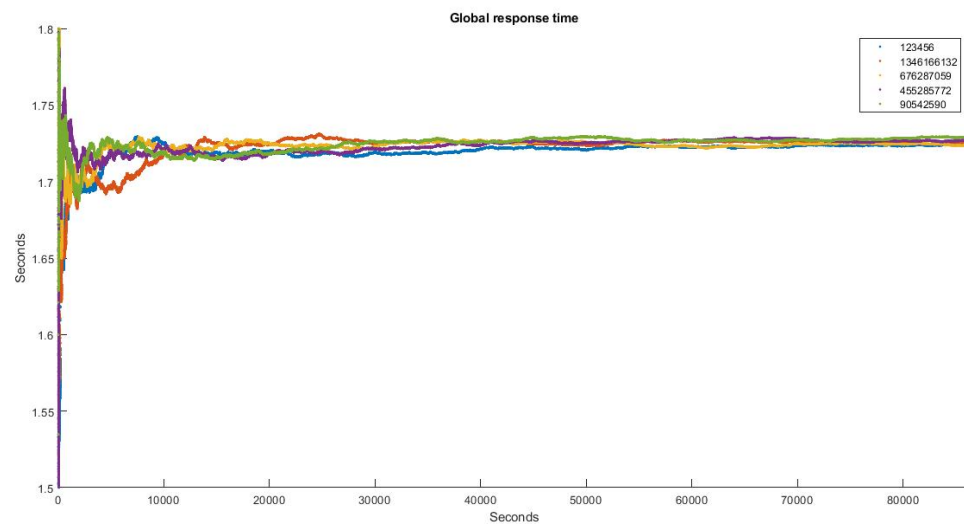
# ANALISI SUI DATI-STAZIONARIETÀ

## Algoritmo 1



# ANALISI SUI DATI-STAZIONARIETÀ

## Algoritmo 2



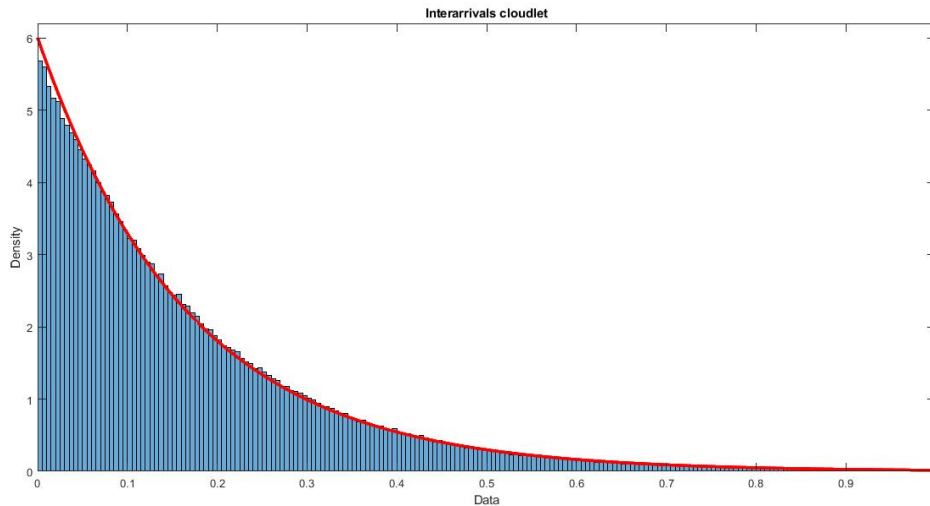
# DISTRIBUZIONE DEGLI INTERARRIVI-ALGORITMO1

## Cloudlet

Distribuzione data dal minimo dei due flussi esponenziali

$$\begin{aligned} P(\min(T_1, T_2) > t) &= P(T_1 > t, T_2 > t) \\ &= P(T_1 > t)P(T_2 > t) = e^{-\lambda_1 t} e^{-\lambda_2 t} = e^{-(\lambda_1 + \lambda_2)t} \end{aligned}$$

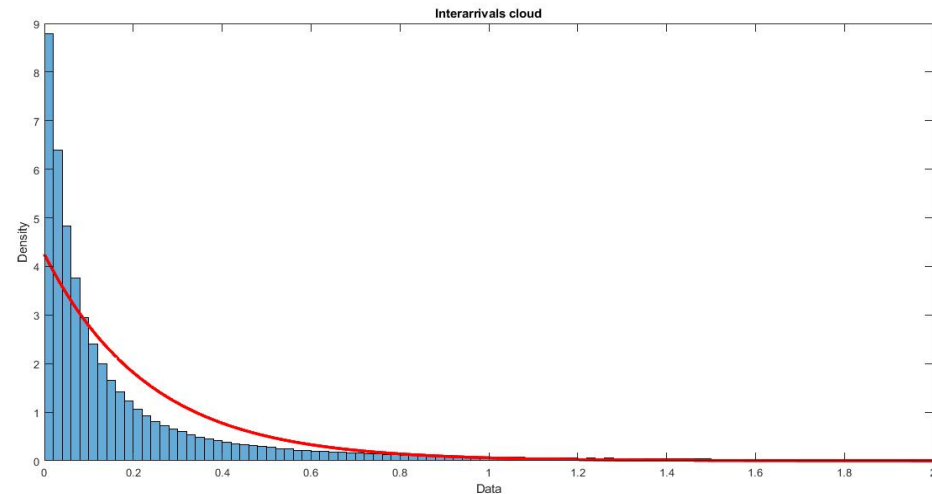
Sistema :  $M/H_2/N/N$



## Cloud

Distribuzione data dall'unione di tre flussi di arrivi -tassi di arrivo diretti al cloud (task1 & task2) solo se cloudlet pieno

Sistema :  $G/M/\infty$



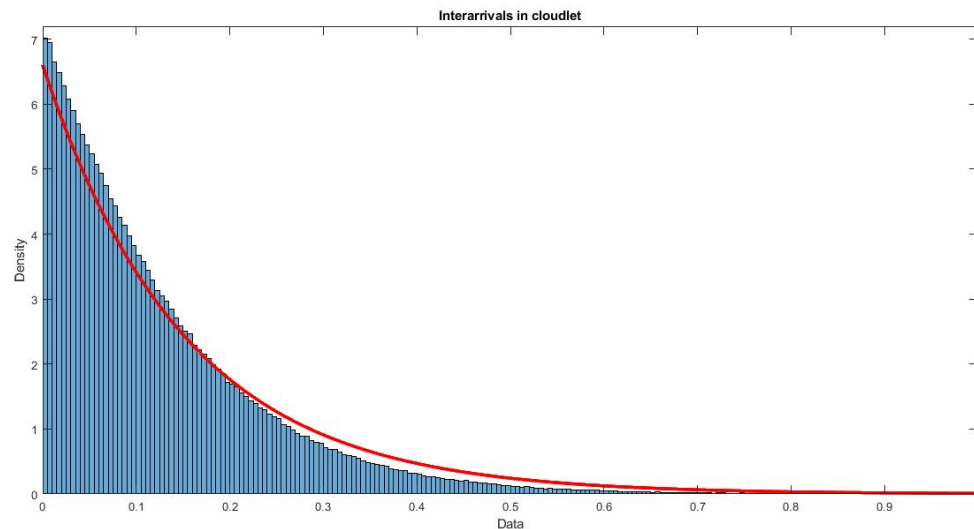
# DISTRIBUZIONE DEGLI INTERARRIVI-ALGORITMO2

## Cloudlet

Distribuzione data dal minimo dei due flussi esponenziali

$$\begin{aligned} P(\min(T_1, T_2) > t) &= P(T_1 > t, T_2 > t) \\ &= P(T_1 > t)P(T_2 > t) = e^{-\lambda_1 t} e^{-\lambda_2 t} = e^{-(\lambda_1 + \lambda_2)t} \end{aligned}$$

Sistema :  $M/H_2/N/N$

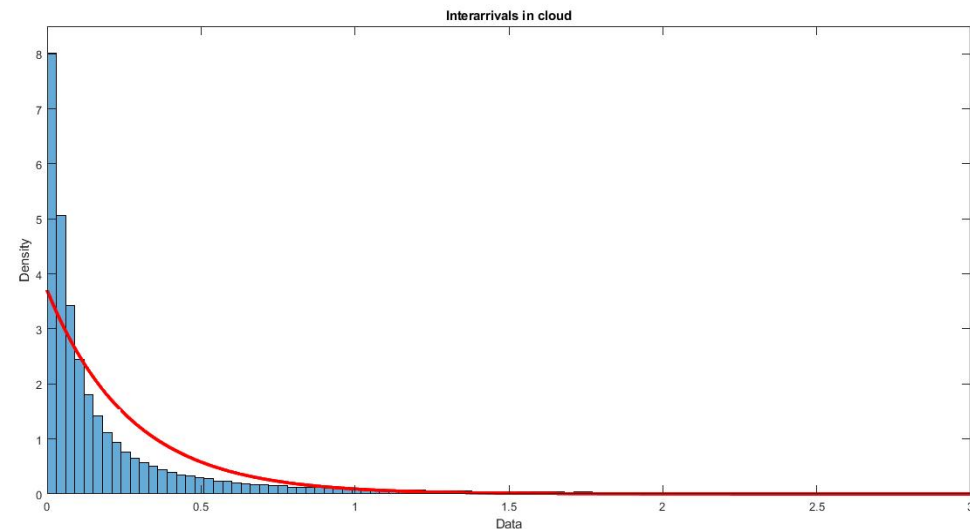


## Cloud

Distribuzione data dall'unione di tre flussi di arrivi

- tassi di arrivo diretti al cloud (task1 & task2)
- tasso di arrivo dei job interrotti nel cloudlet

Sistema:  $G/M/\infty$

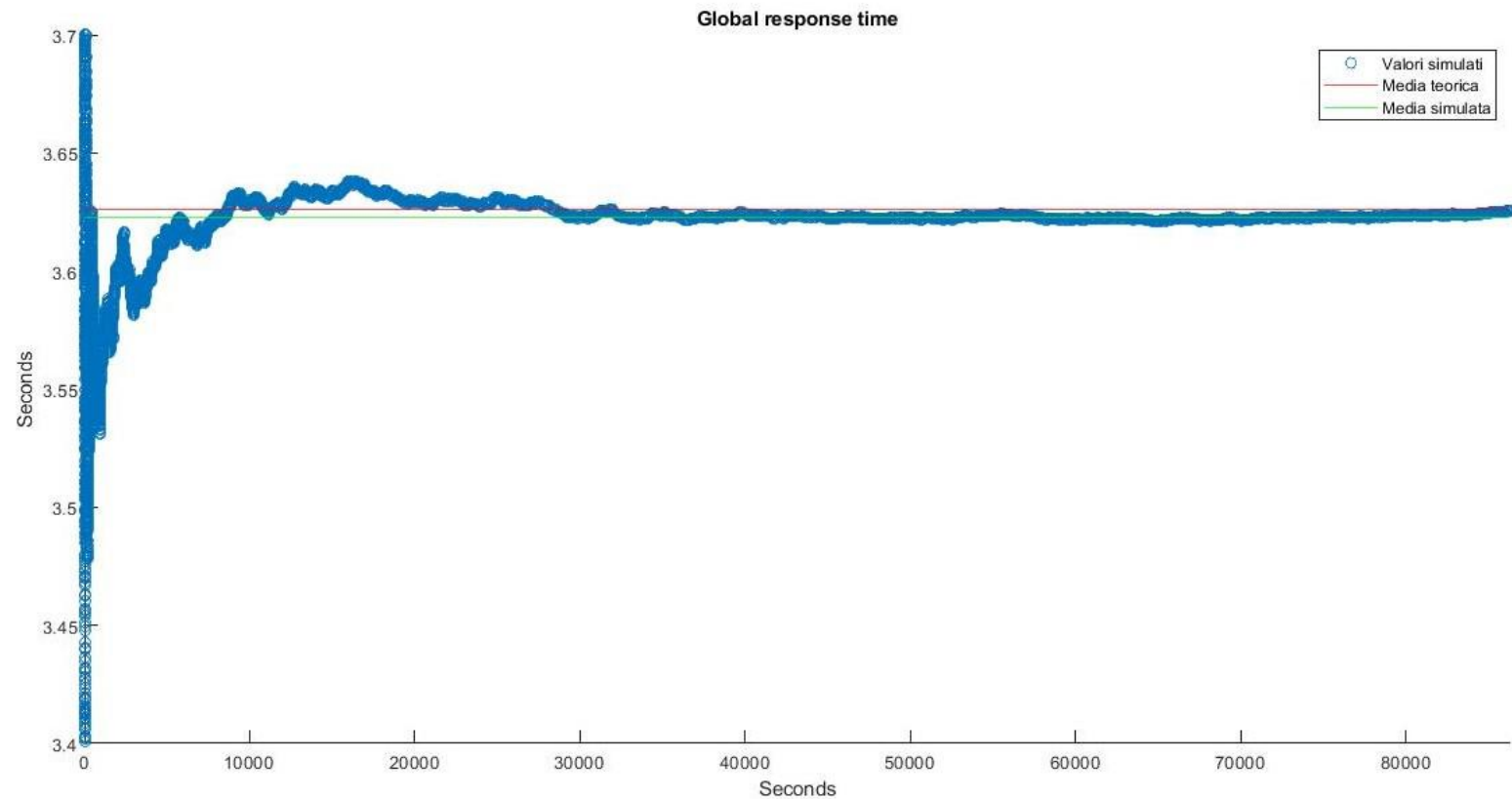


# CONFRONTI TRA METRICHE A REGIME E VALORI TEORICI- ALGORITMO 1

Parametri di simulazione

1. Tempo totale: 86400 secondi
2. Seed iniziale: 123456
3. Dimensione del batch: 3600 secondi
4. Scheduling: Random
5. Tempo servizio Cloudlet: Esponenziale

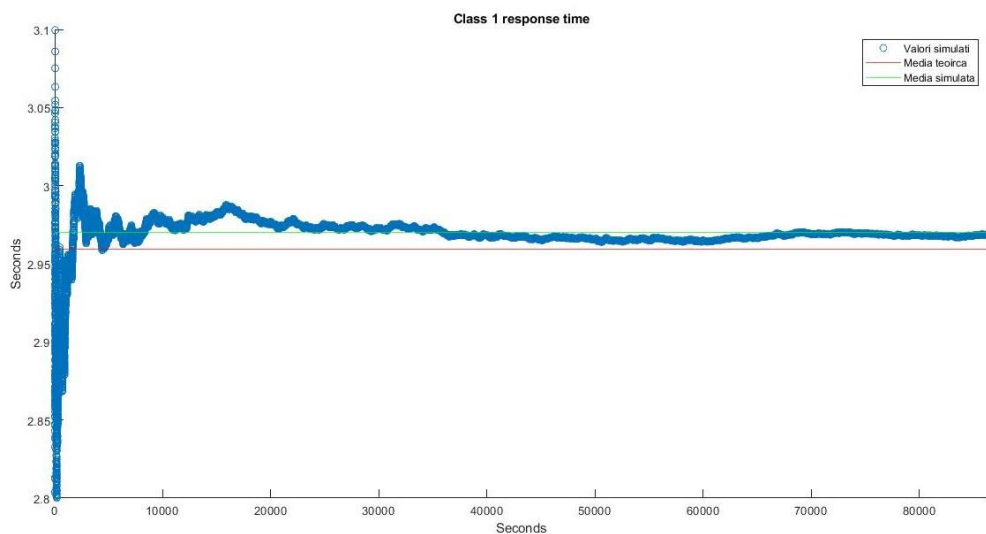
# TEMPO DI RISPOSTA GLOBALE



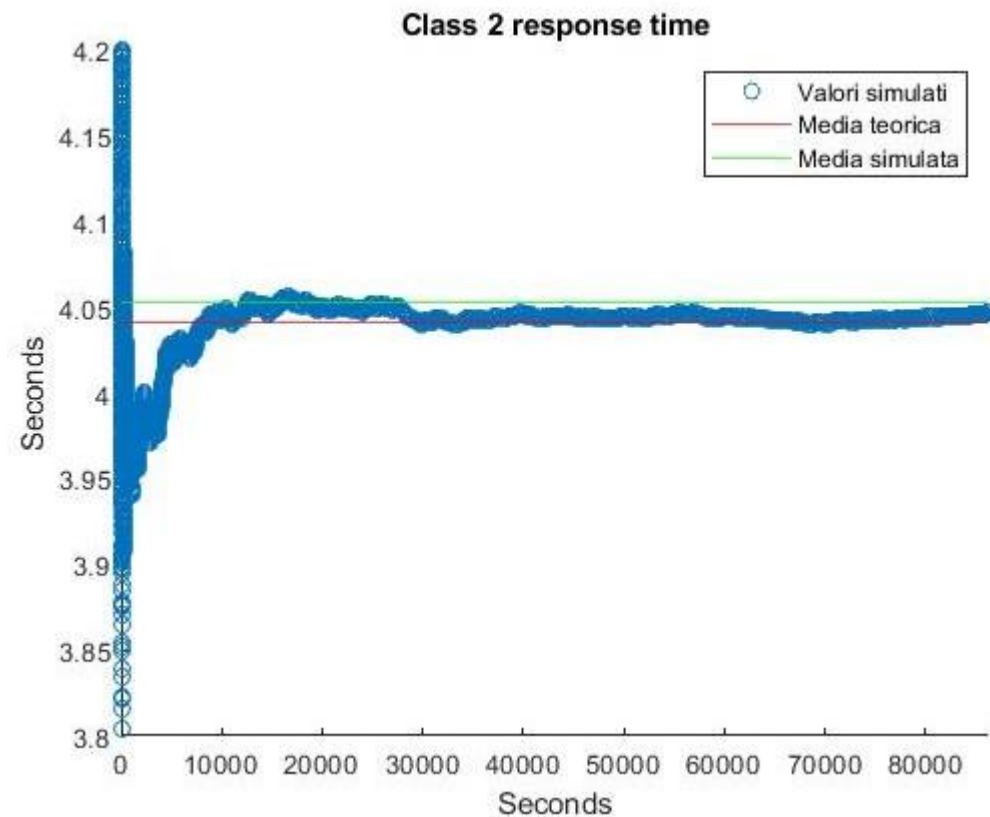
$\alpha$ : 3.625938 s  
s: 3.622958 (+/-0.00347)



# TEMPO DI RISPOSTA PER CLASSE

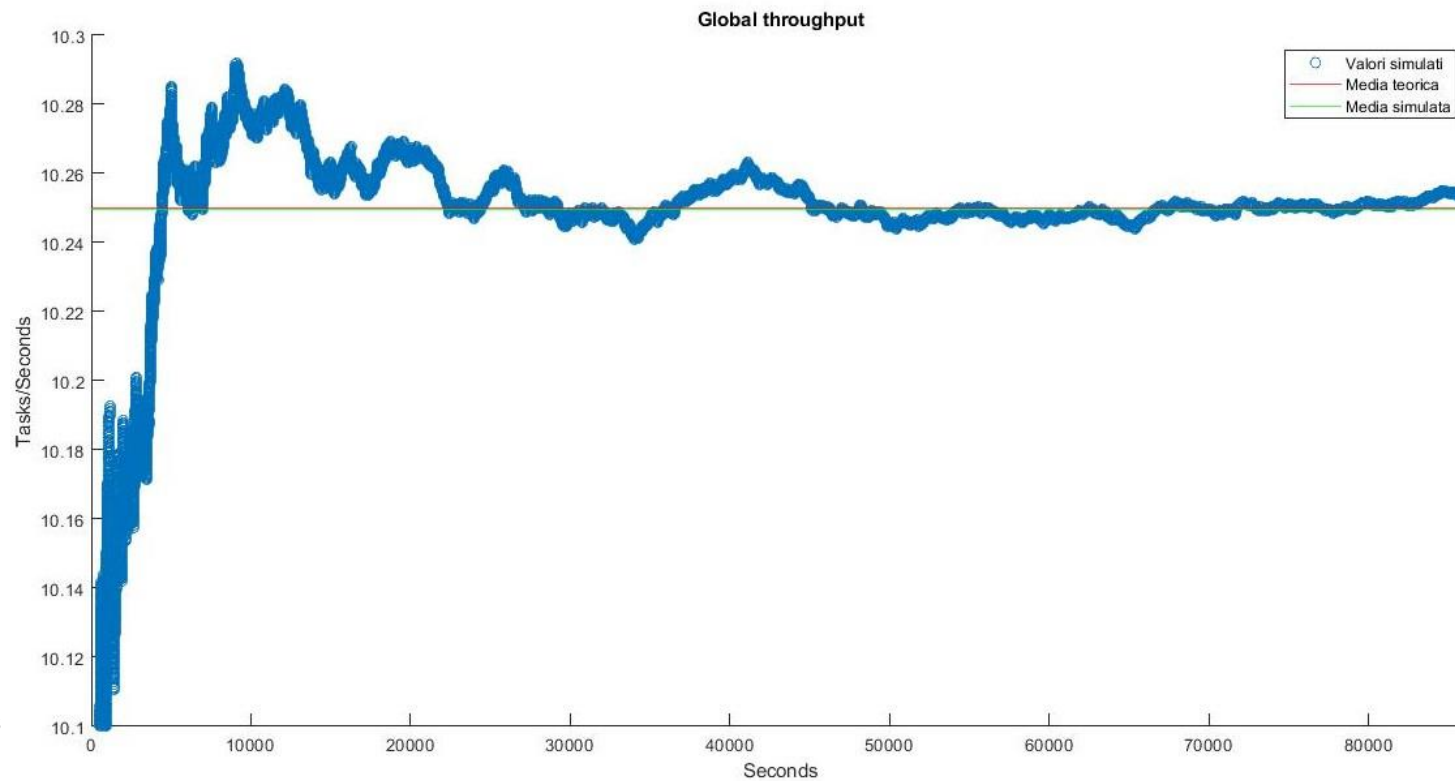


$\alpha$ : 2.959198 s  
 $s$ : 2.970106 (+/-0.00184) s



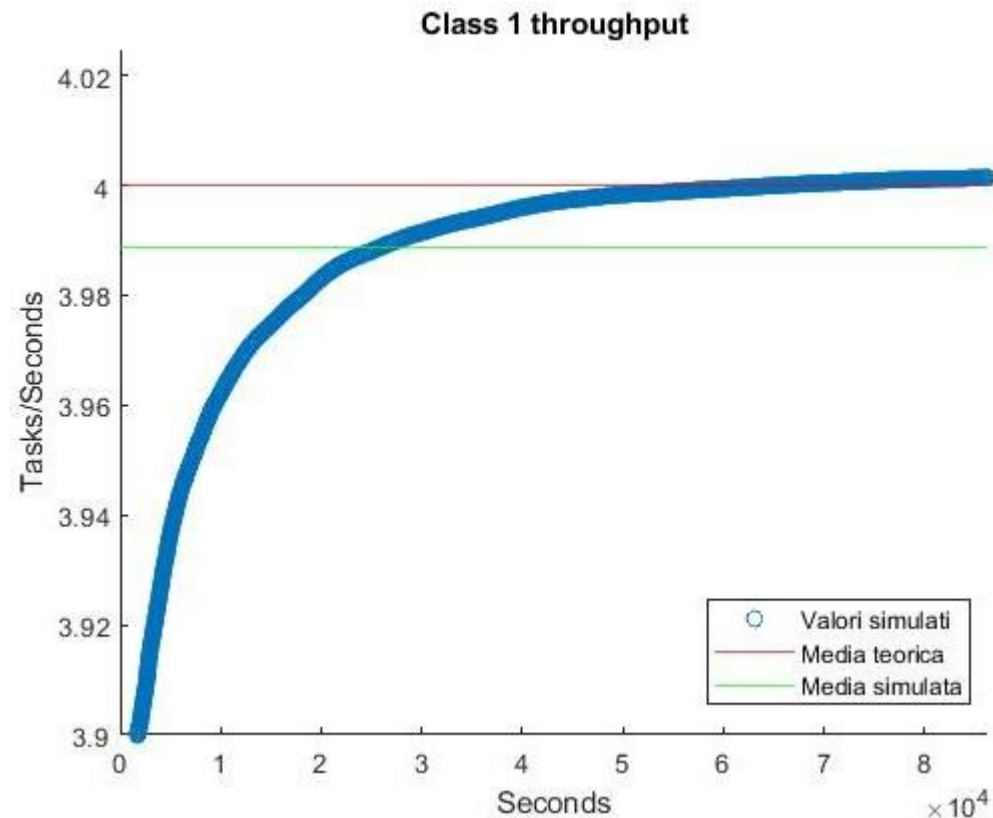
$\alpha$ : 4.05265 s  
 $s$ : 4.041308 (+/-0.00582) s

# THROUGHPUT GLOBALE

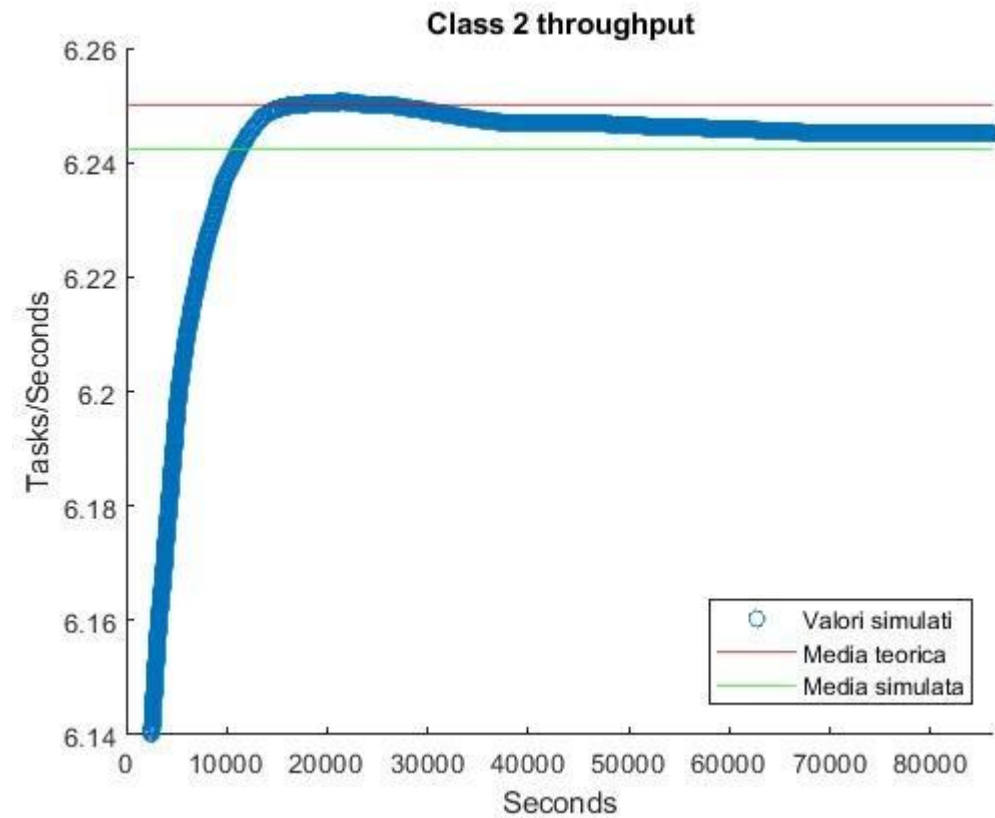


$\alpha$ : 10.25 task/s,  
 $s$ : 10.231102 (+/-0.01476) task/s

# THROUGHPUT PER CLASSE

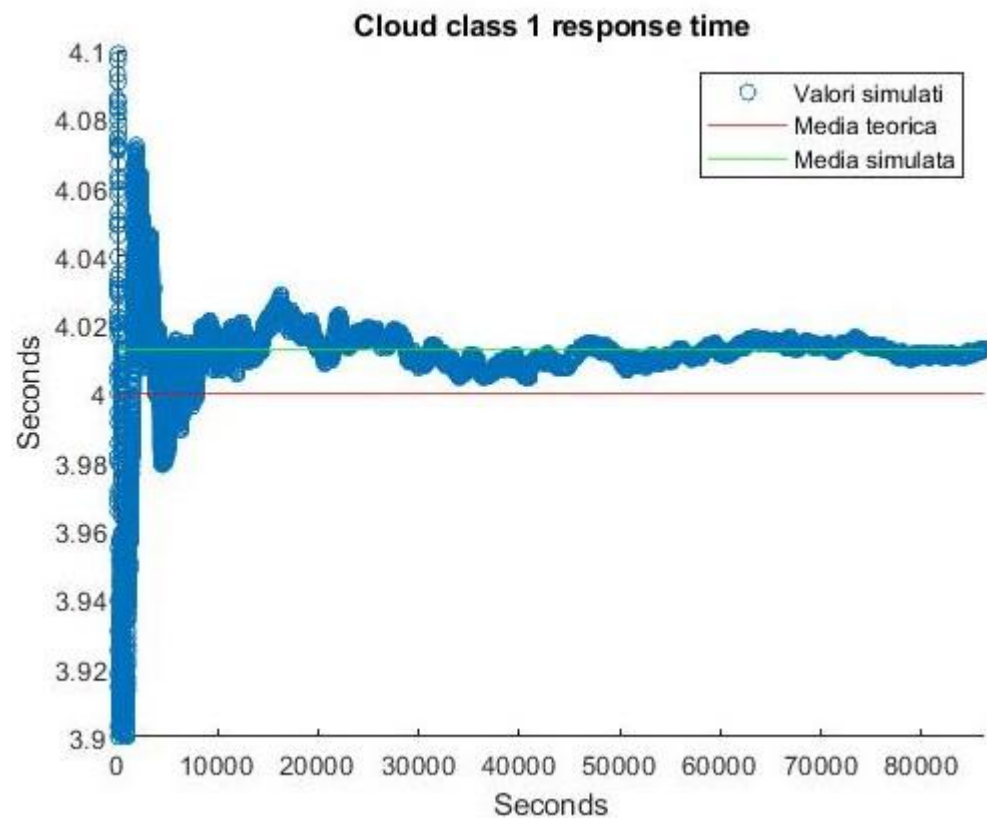


$\alpha$ : 4 task/s  
 $s$ : 3.988907 (+/-0.00812) task/s

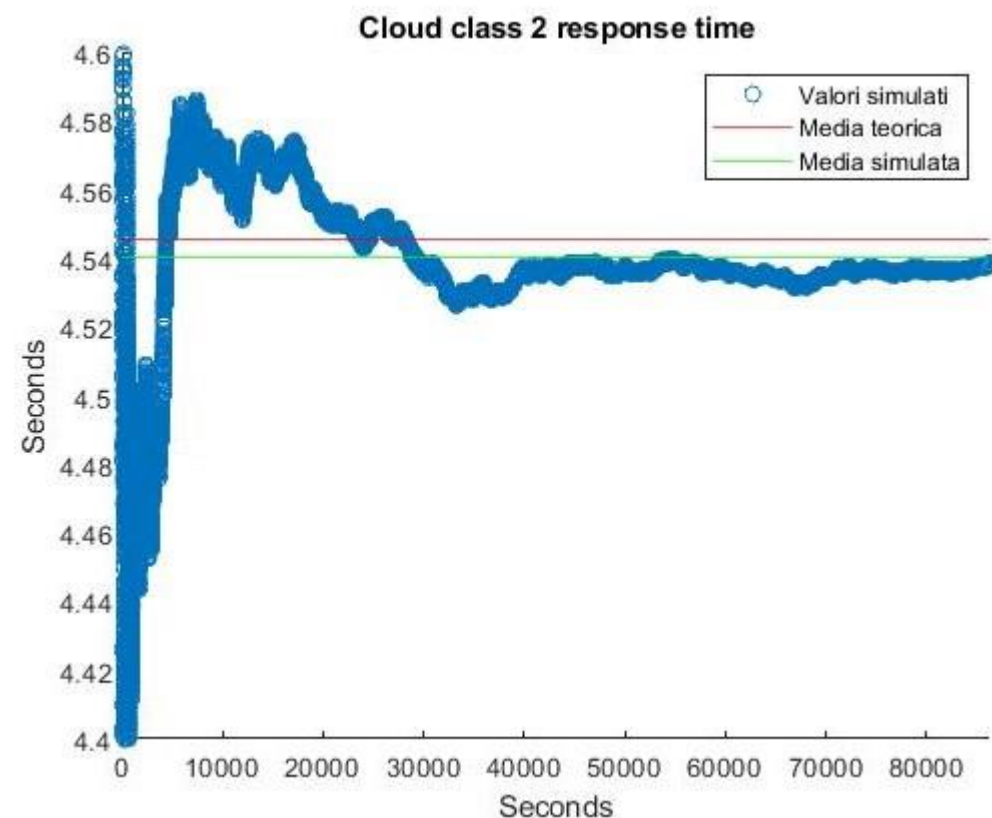


$\alpha$ : 6.25 task/s  
 $s$ : 6.242194 (+/-) 0.00729 task/s

# TEMPI DI RISPOSTA PER CLOUD

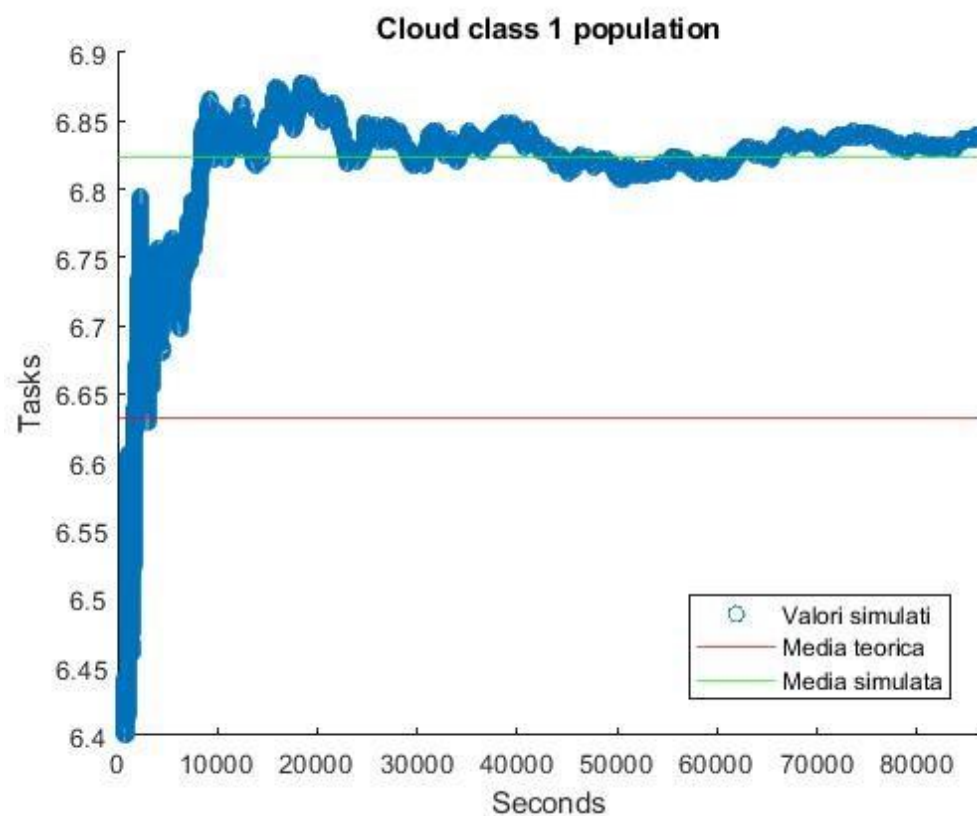


$\alpha$ : 4 s  
 $s$ : 4.012712 (+/-0.00148) s



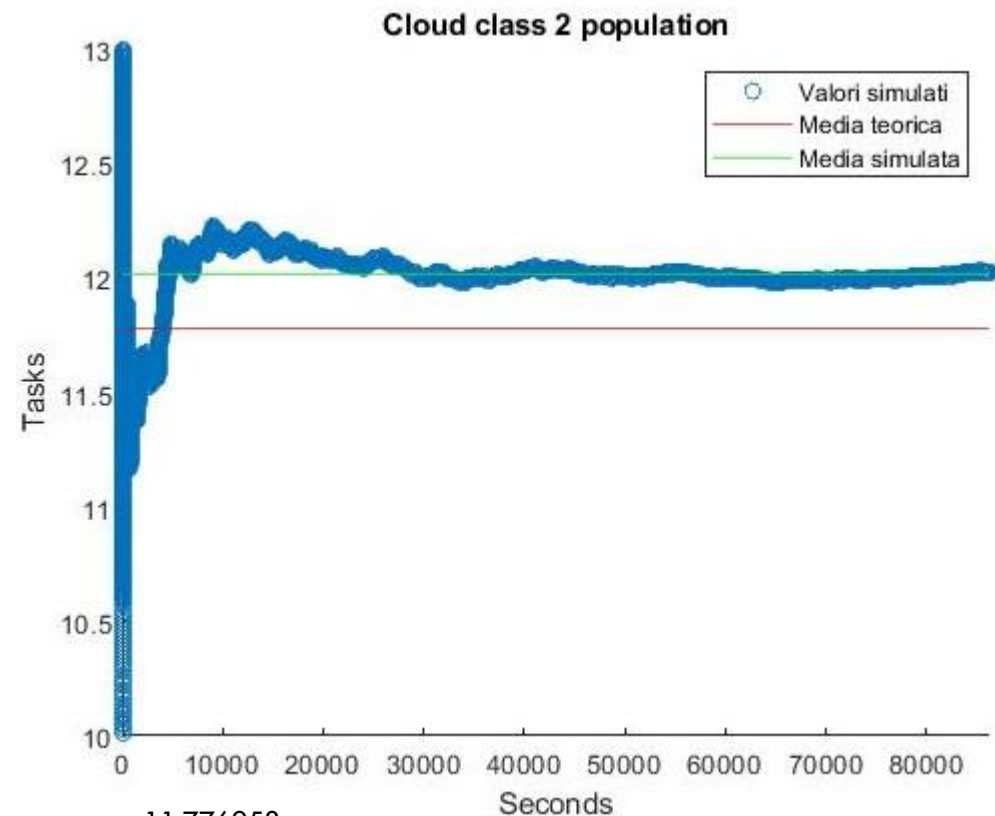
$\alpha$ : 4.54s  
 $s$ : 4.540414 (+/-0.00741) s

# POPOLAZIONE MEDIA PER CLOUD



$\alpha$ : 6.632784

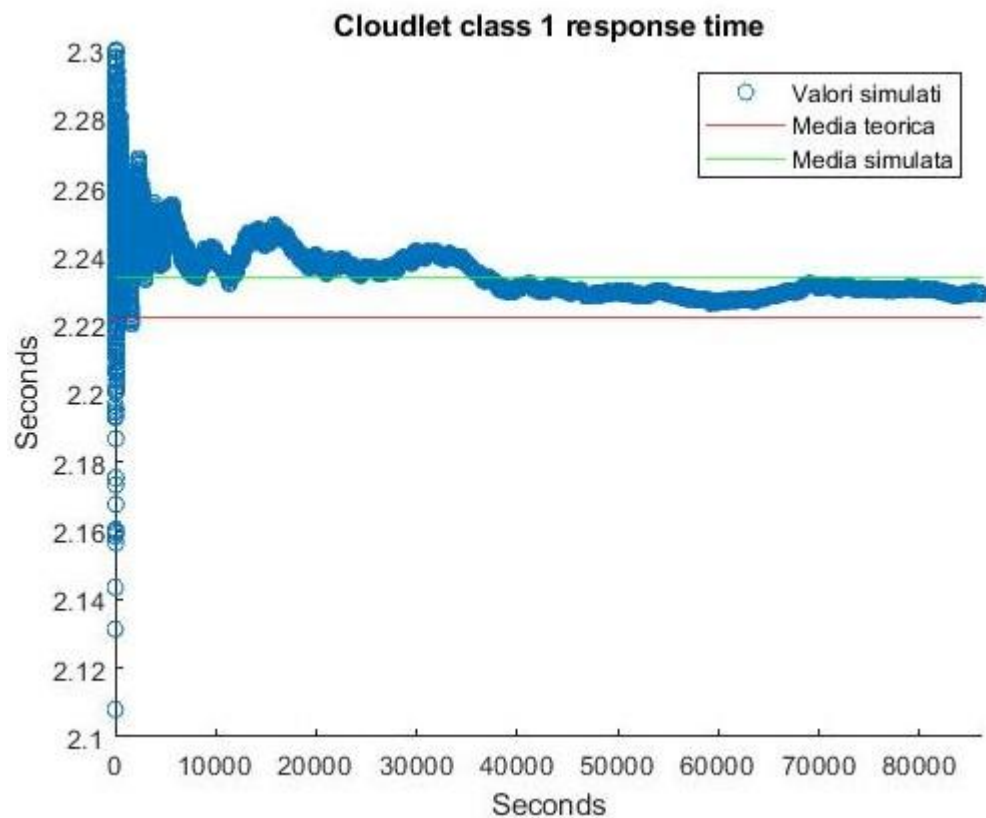
$s$ : 6.823794 (+/-0.01430)



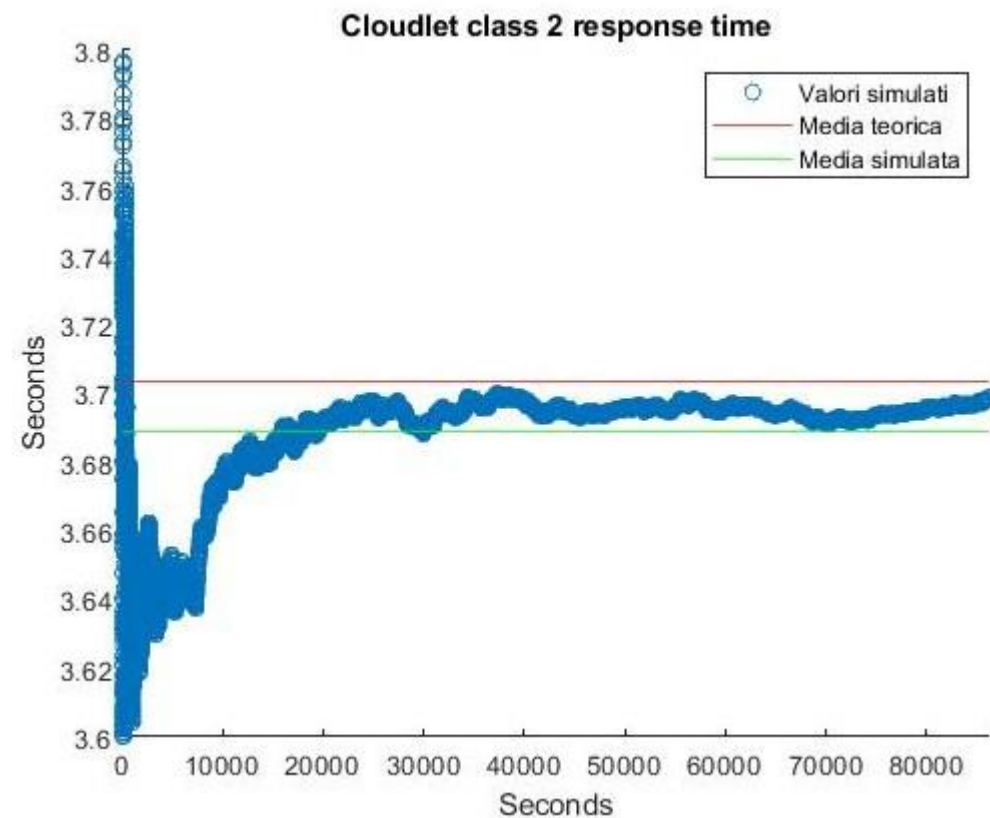
$\alpha$ : 11.776958

$s$ : 12.016666 (+/-0.04330)

# TEMPI DI RISPOSTA PER CLOUDLET

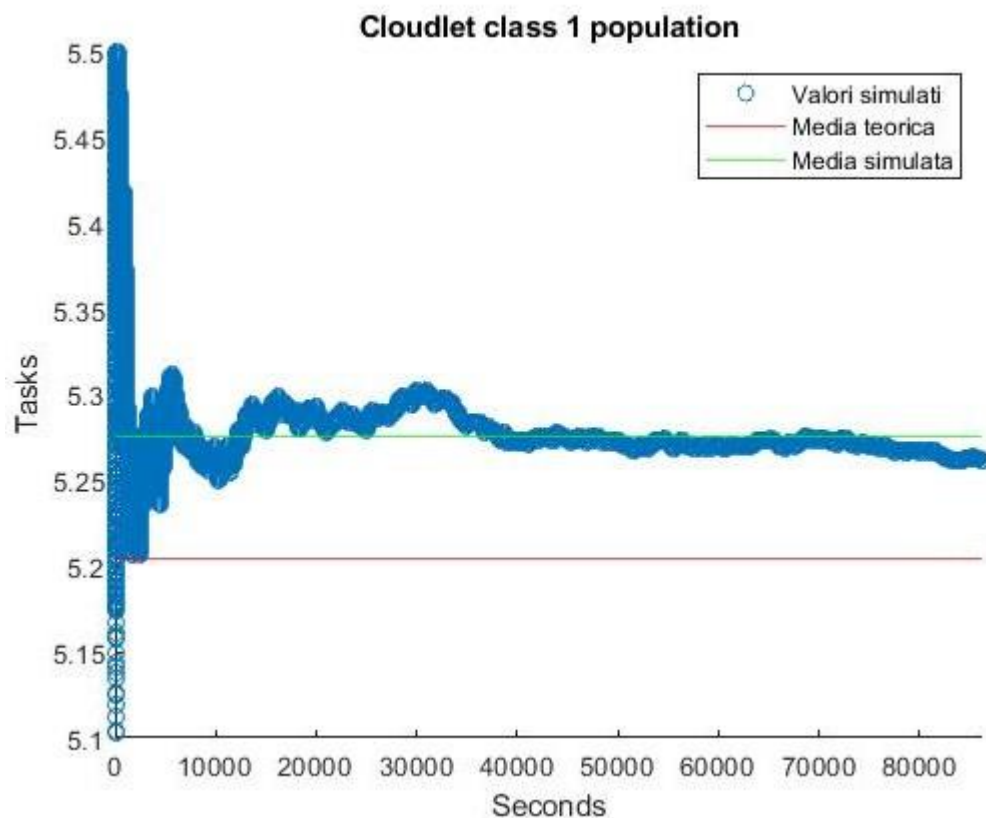


$\alpha$ : 2.2s  
s: 2.233544(+/-0.00231)s

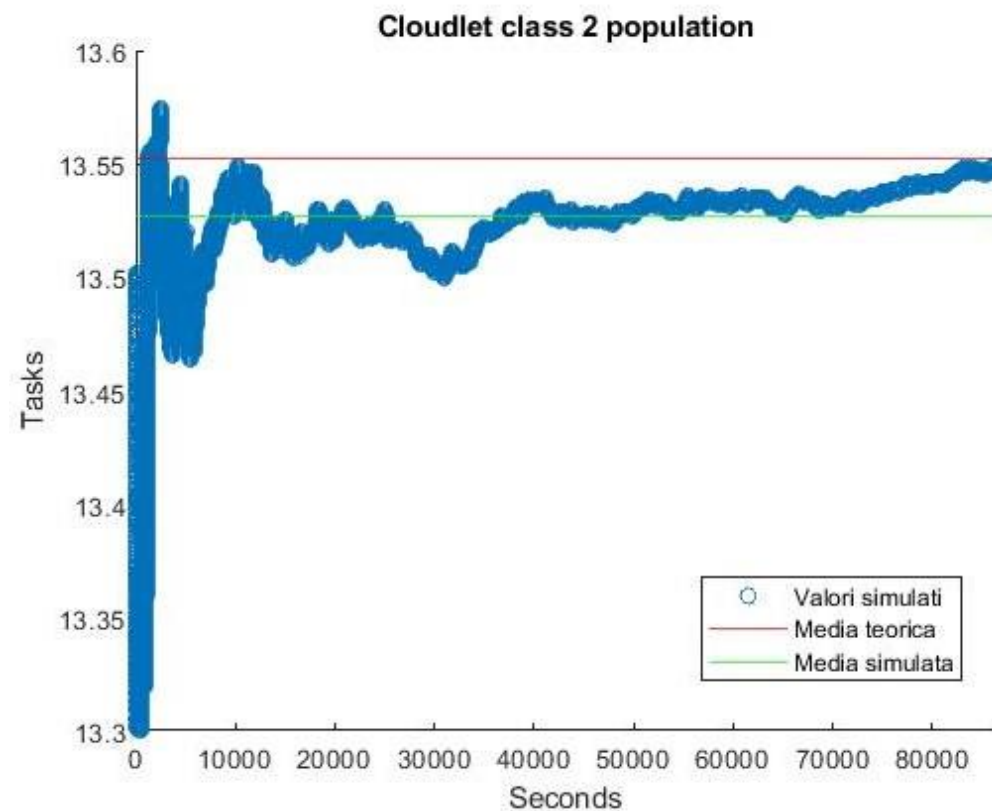


$\alpha$ : 3.703s  
s: 3.666056(+/-0.00689)s

# POPOLAZIONE MEDIA PER CLOUDLET



$\alpha$ : 5.204008  
 $s$ : 5.275428 (+/-0.00411)



$\alpha$ : 13.552107  
 $s$ : 13.526597 (+/-0.00556)

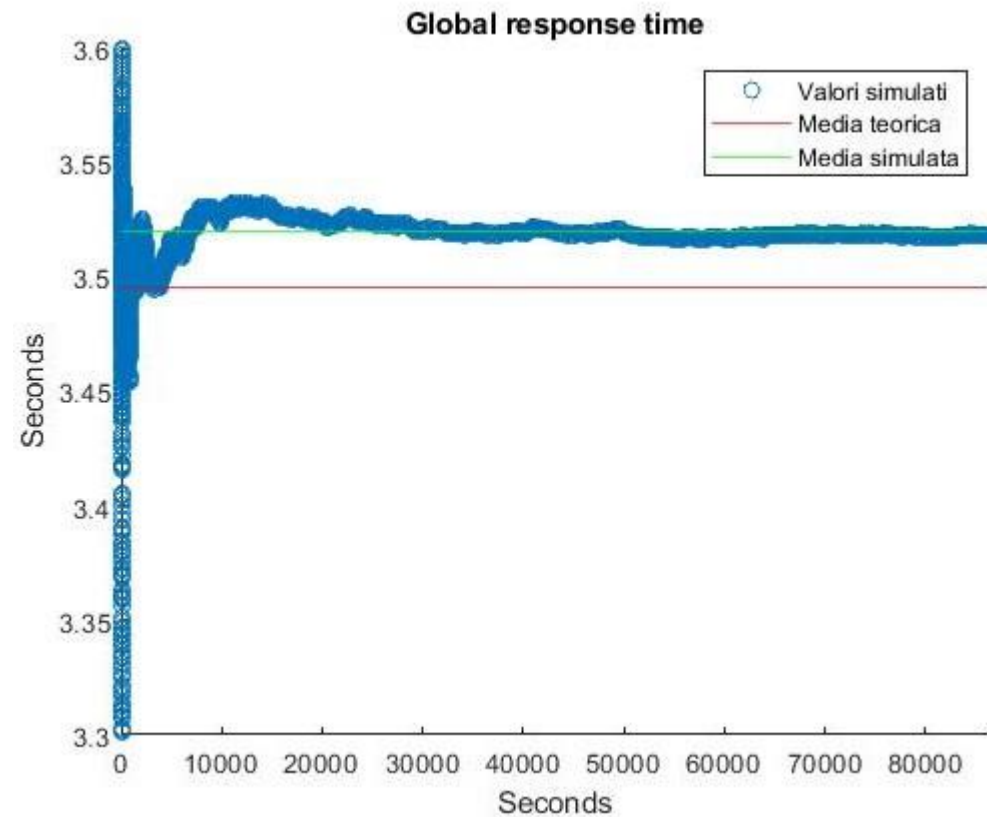
# CONFRONTI TRA METRICHE A REGIME E VALORI TEORICI- ALGORITMO 2

Parametri di simulazione

1. Tempo totale: 86400 secondi
2. Seed iniziale: 123456
3. Dimensione del batch: 3600 secondi
4. Scheduling: Random
5.  $S=20$
6. Tempo servizio Cloudlet: Esponenziale



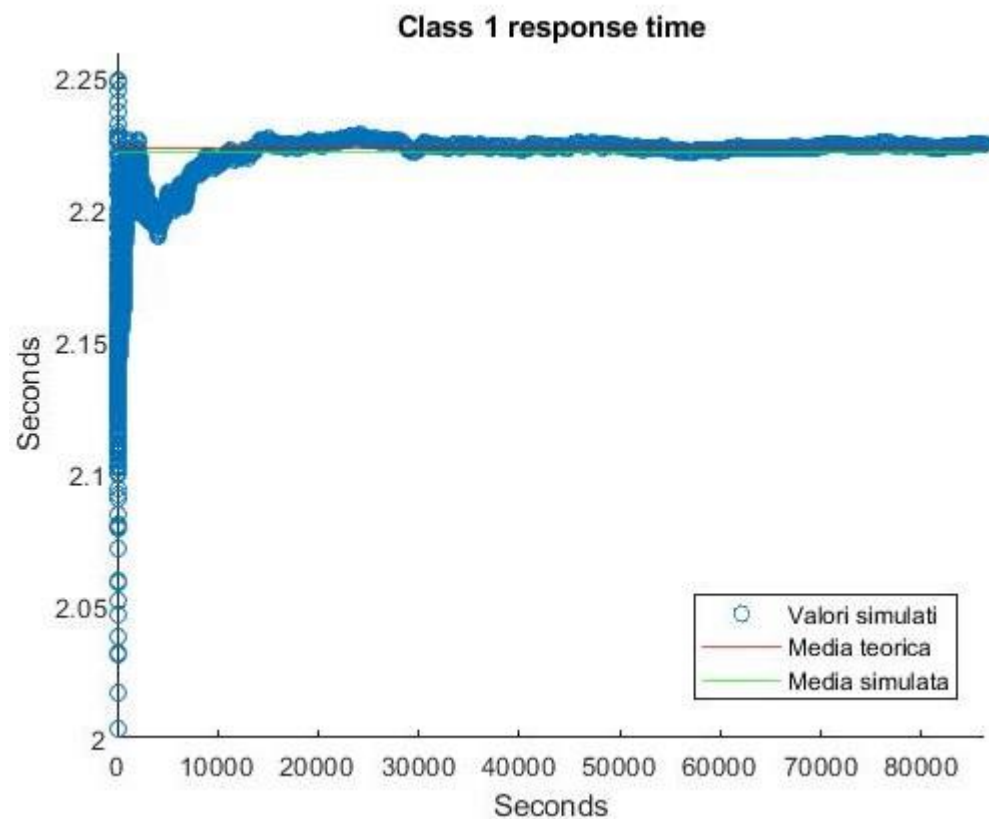
# TEMPO DI RISPOSTA GLOBALE



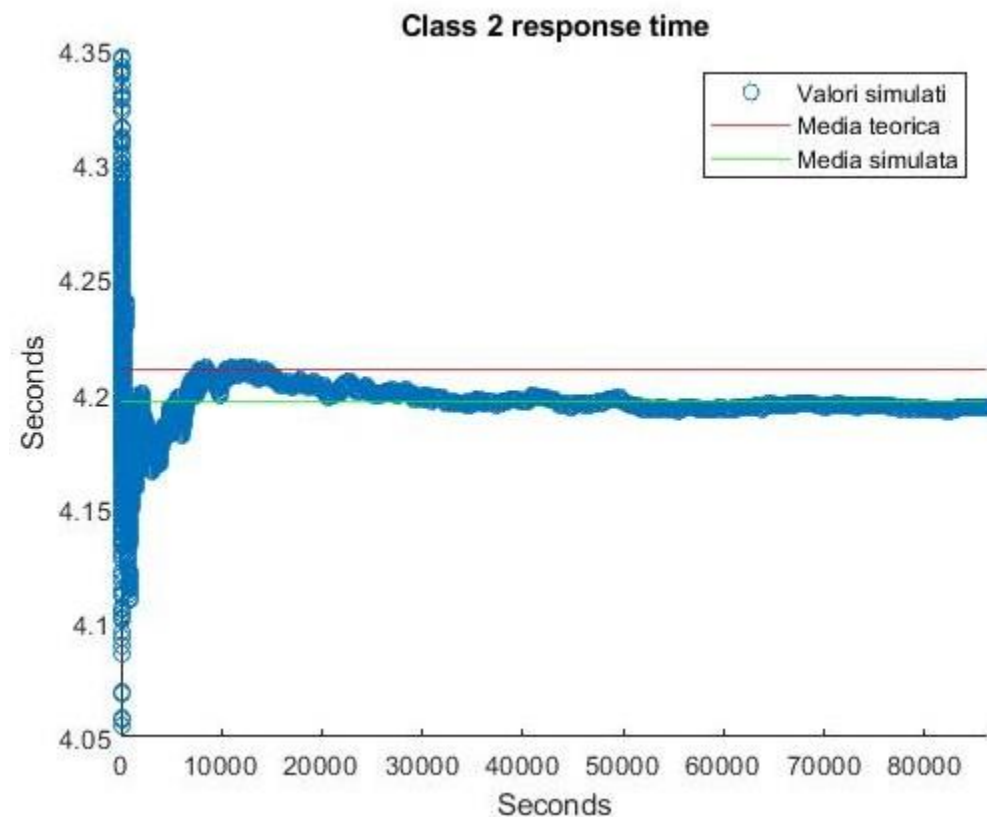
$\alpha$ : 3.495674 s

s: 3.458544 (+/-0.00237) s

# TEMPO DI RISPOSTA PER CLASSE

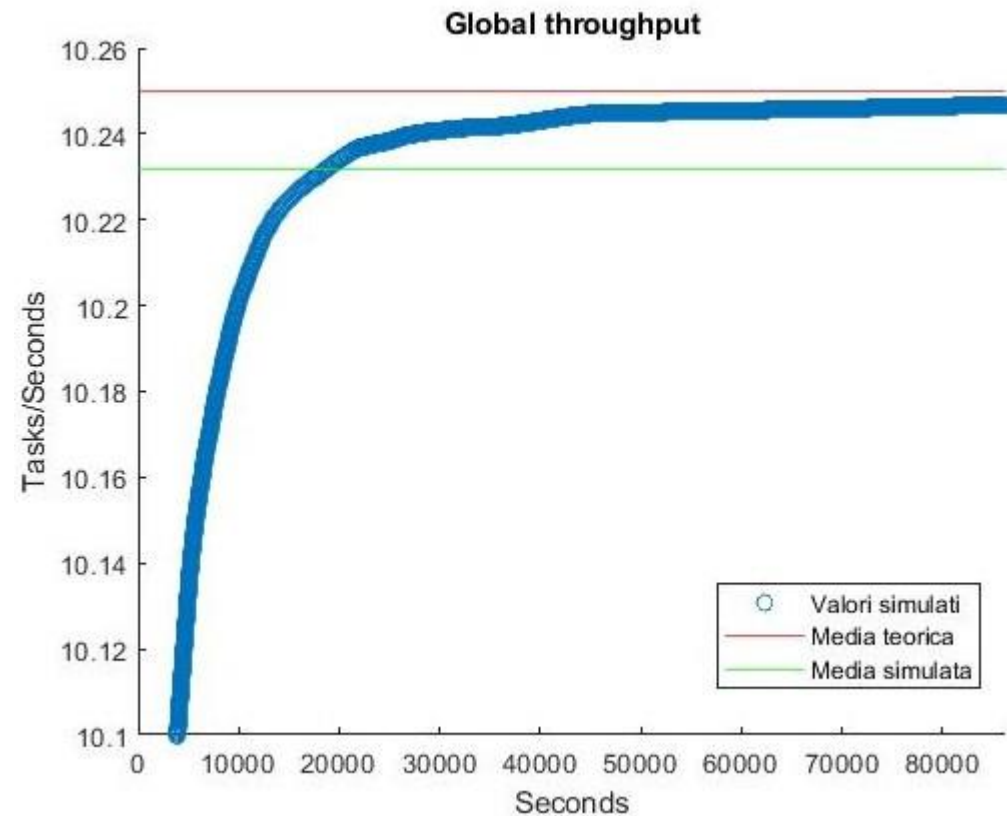


$\alpha$ : 2.2s  
 $s$ : 2.222094(+/-0.00274)s



$\alpha$ : 4.210071  
 $s$ : 4.103553 (+/-0.00254) s

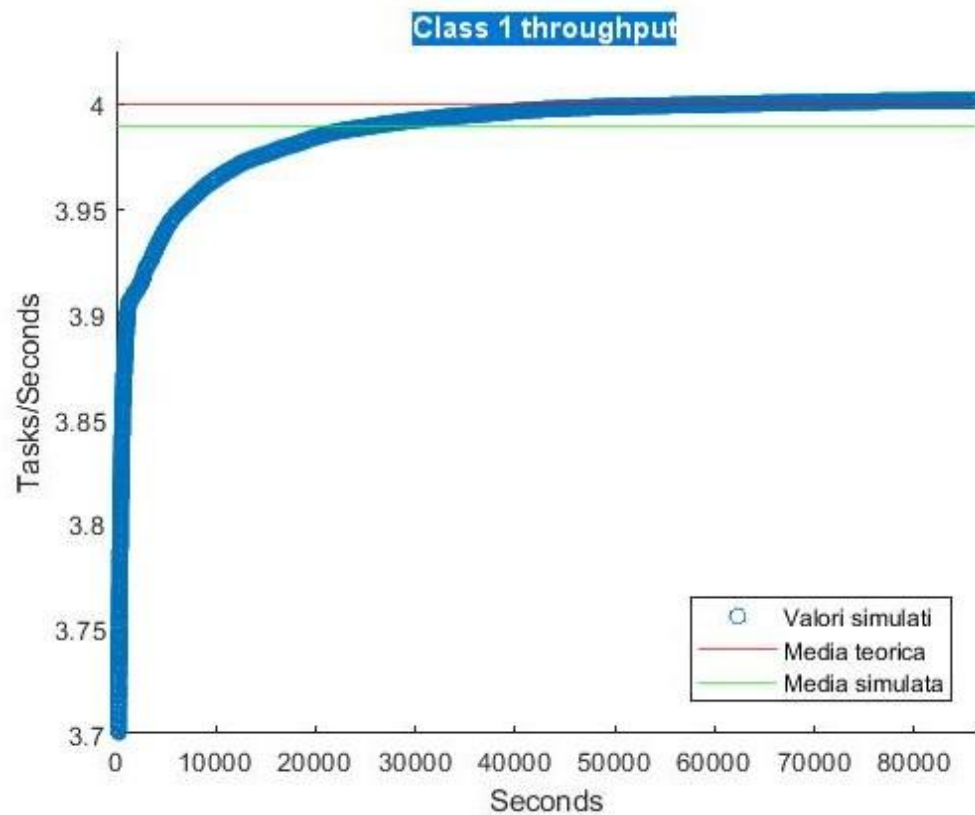
# THROUGHPUT GLOBALE



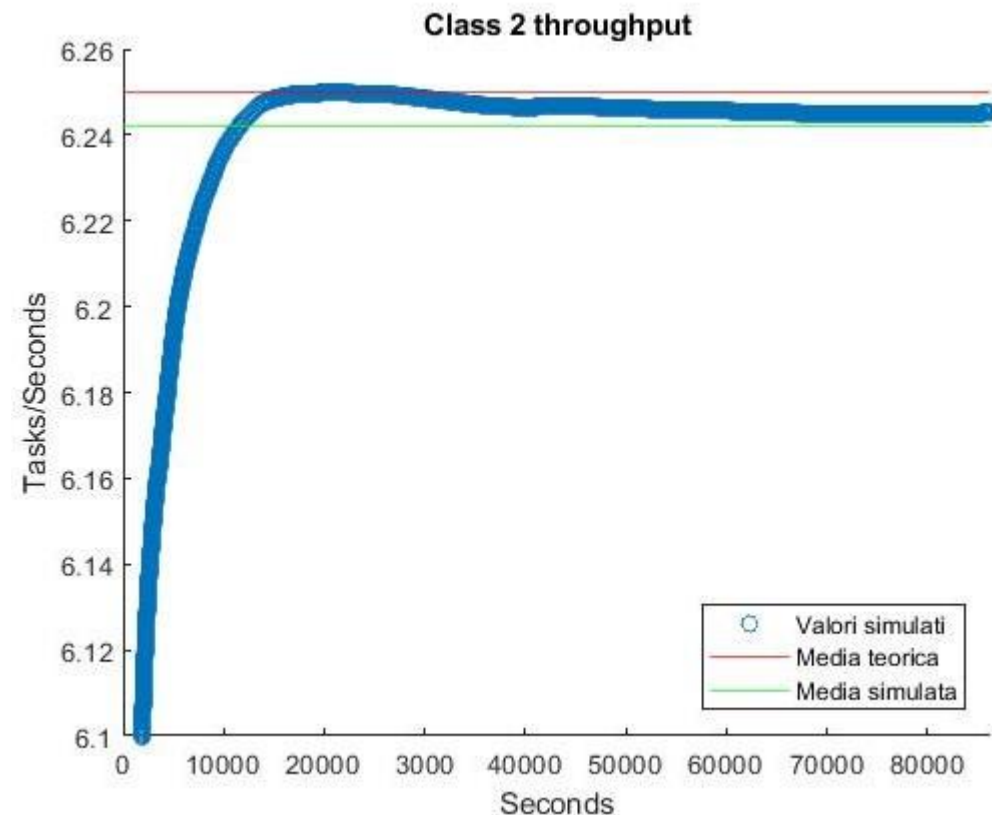
$\alpha$ : 10.25 task/s

$s$ : 10.231724 ( $\pm 0.01448$ ) task/s

# THROUGHPUT PER CLASSE

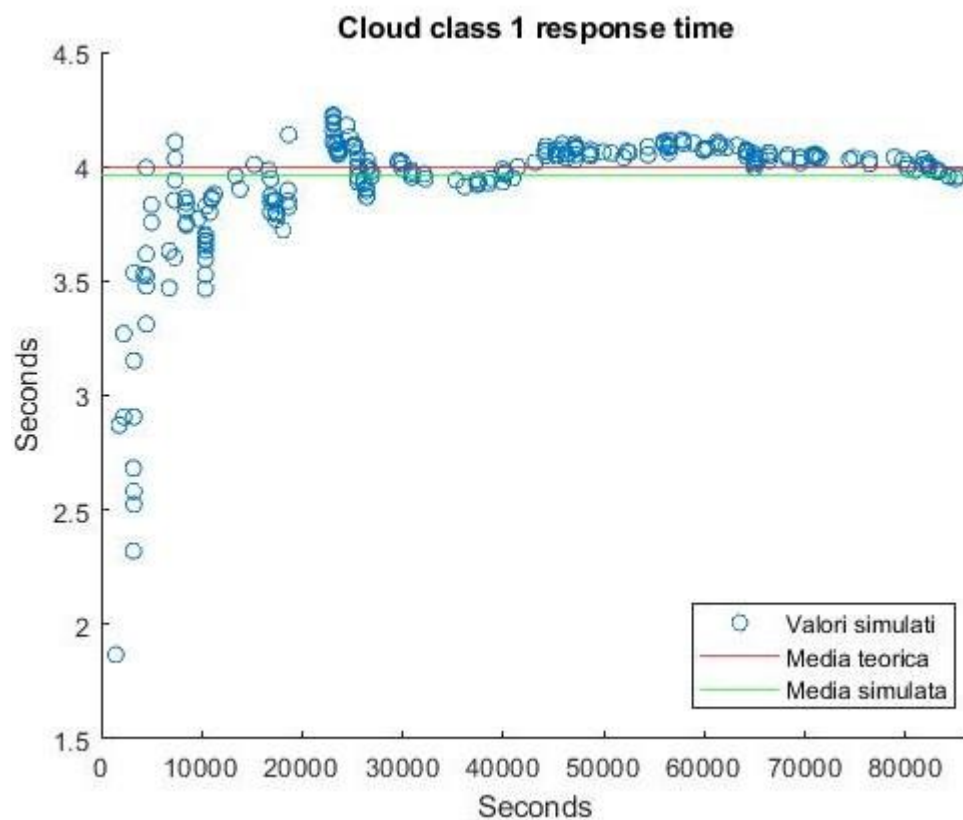


$\alpha$ : 4 task/s  
 $s$ : 3.989873 ( $\pm 0.00768$ ) task/s

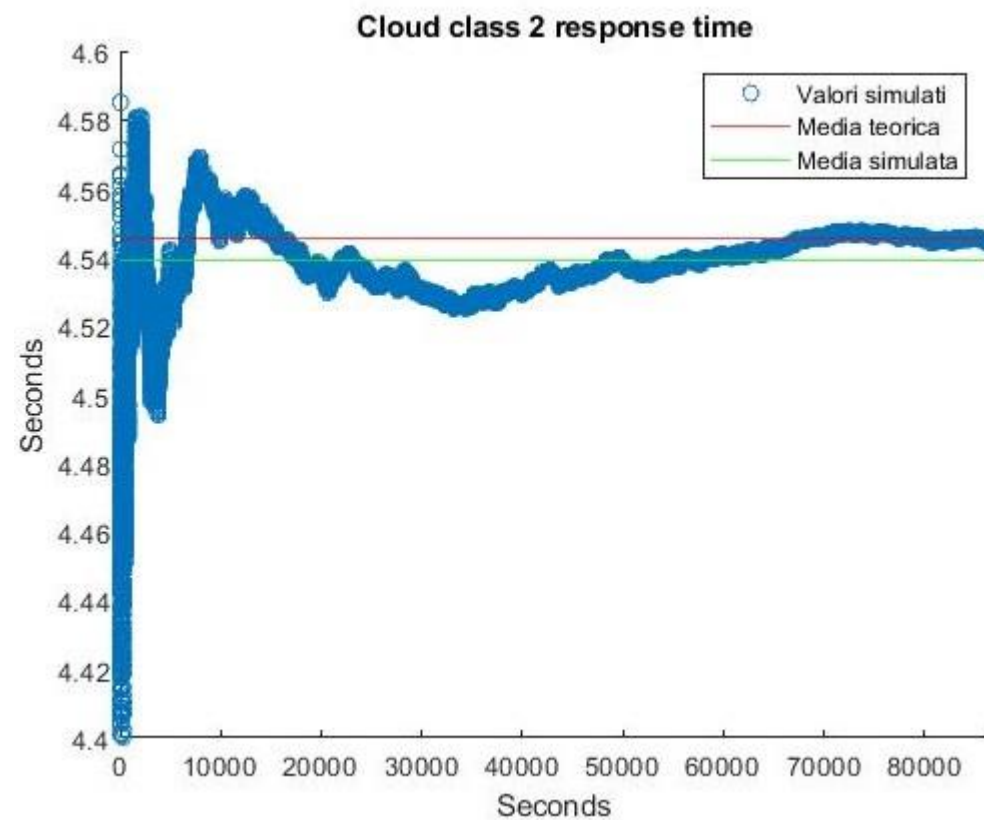


$\alpha$ : 6.25 task/s  
 $s$ : 6.241851 ( $\pm 0.00744$ ) task/s

# TEMPI DI RISPOSTA PER CLOUD

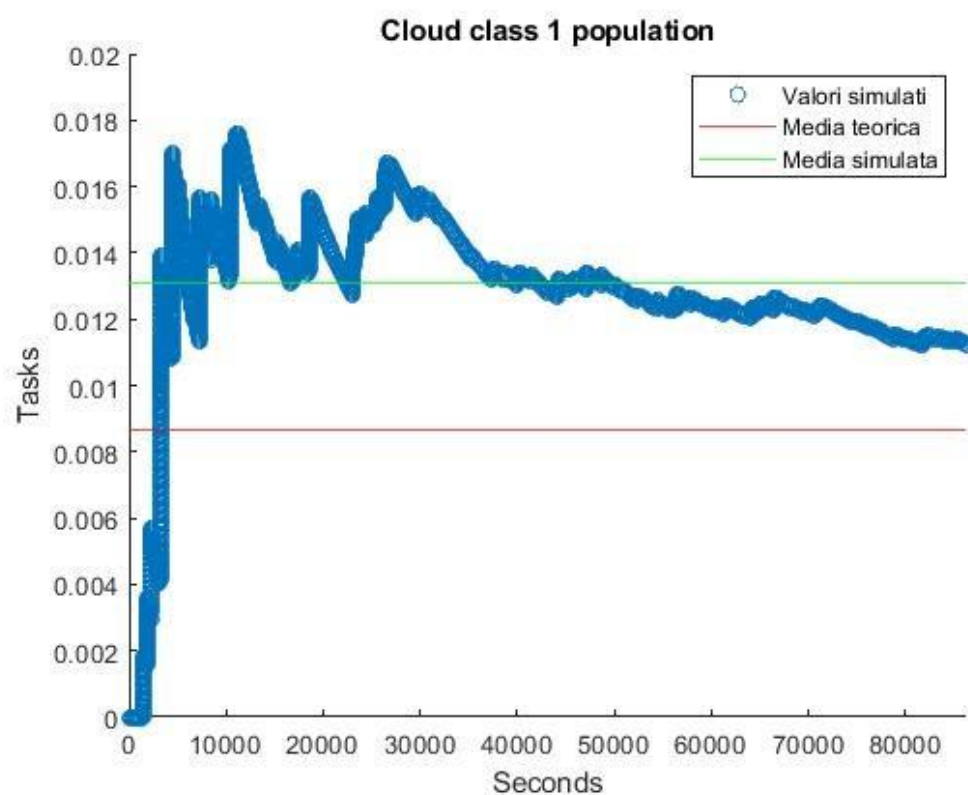


$\alpha$ : 4 s  
 $s$ : 4.130490 (+/-0.171825) s

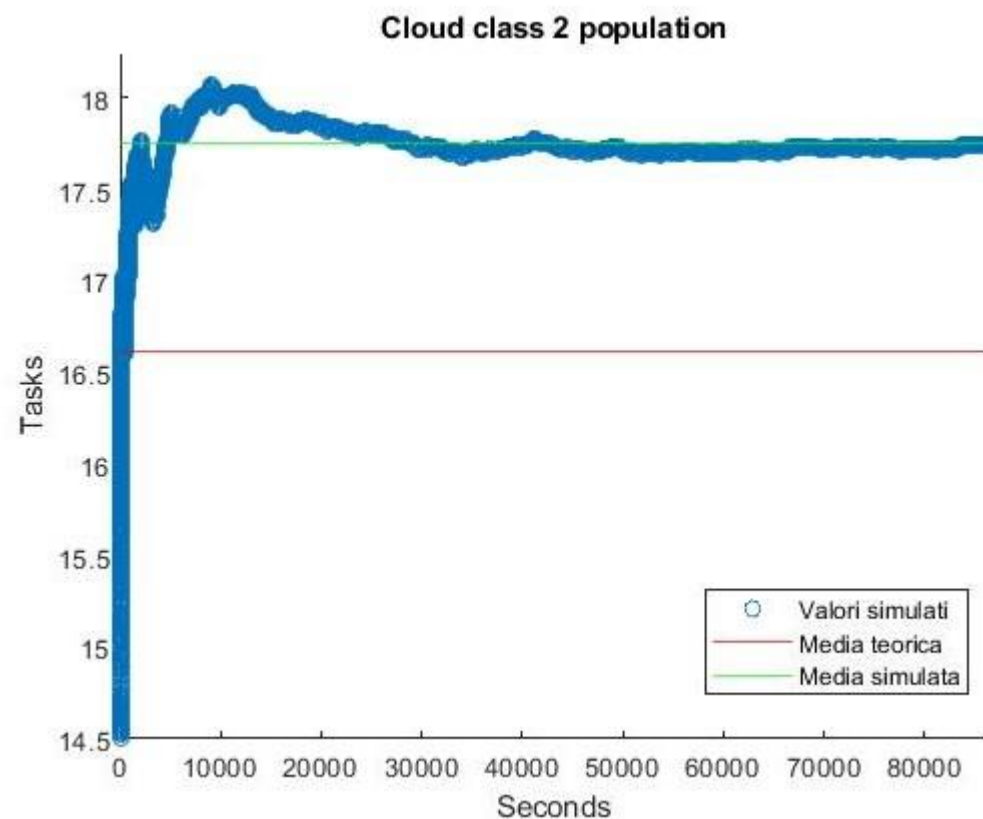


$\alpha$ : 4.5454 s  
 $s$ : 4.538889 (+/-0.00422) s

# POPOLAZIONE MEDIA PER CLOUD

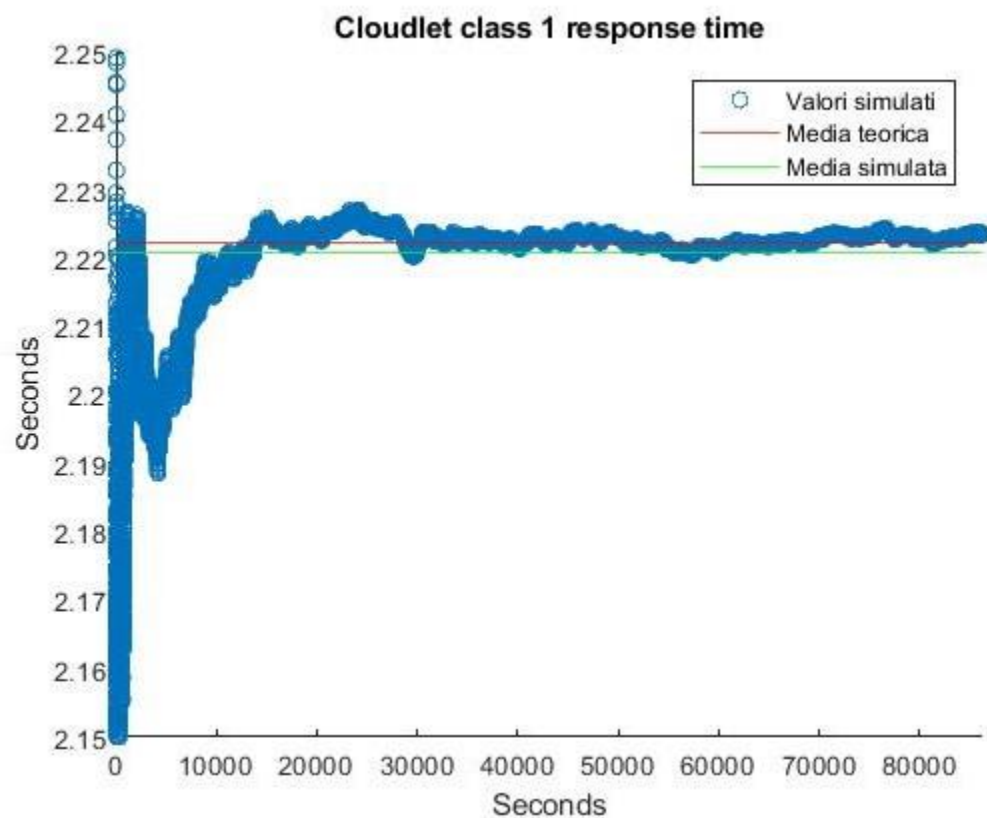


$\alpha$ : 0.00864224 task  
 $s$ : 0.013062 (+/-0.00059) task



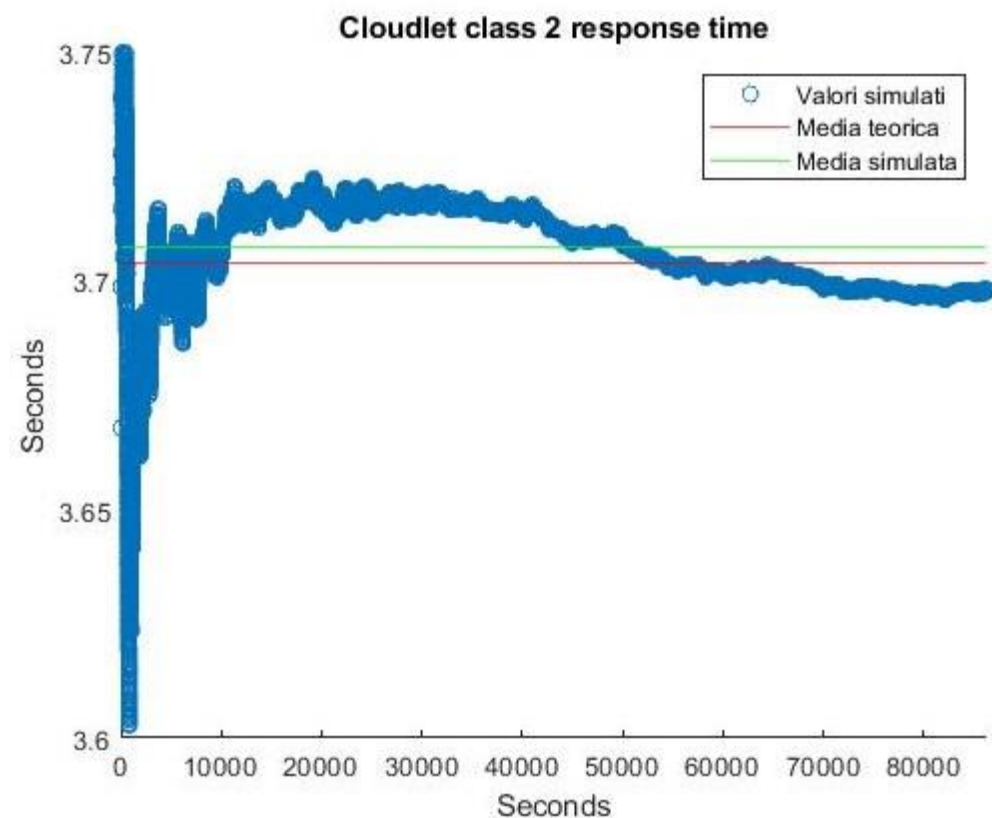
$\alpha$ : 16.61663 task  
 $s$ : 17.029314 (+/-0.04705) task

# TEMPI DI RISPOSTA PER CLOUDLET



$\alpha: 2.2s$

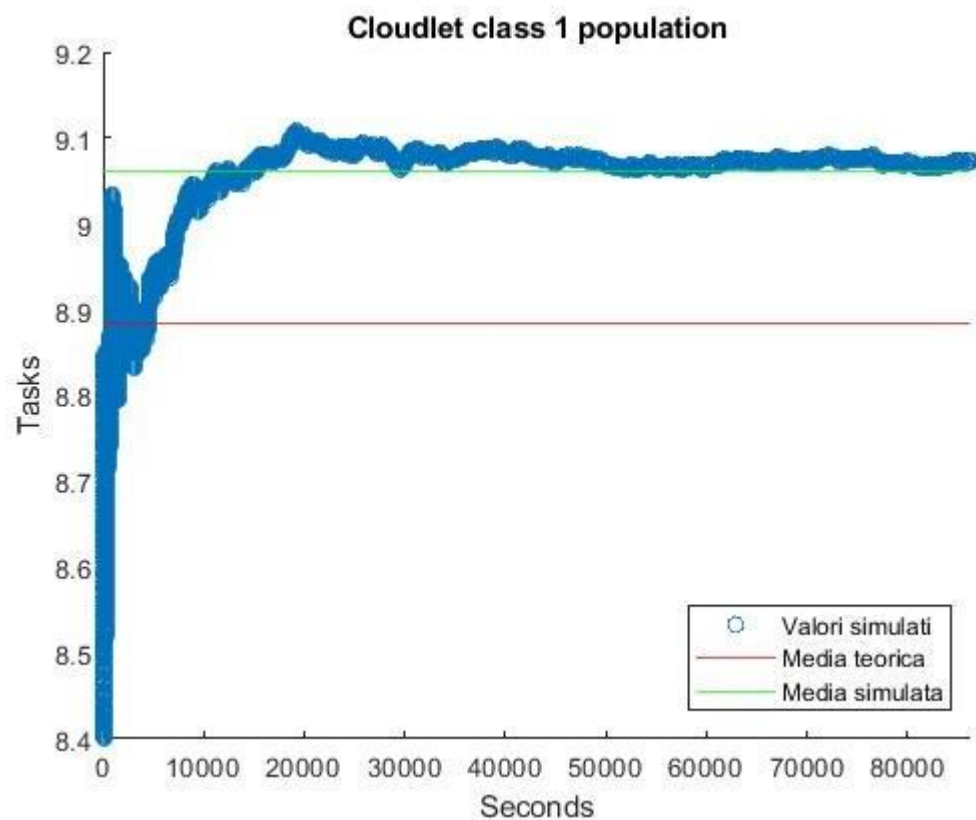
$s: 2.218467 (+/-0.007387)s$



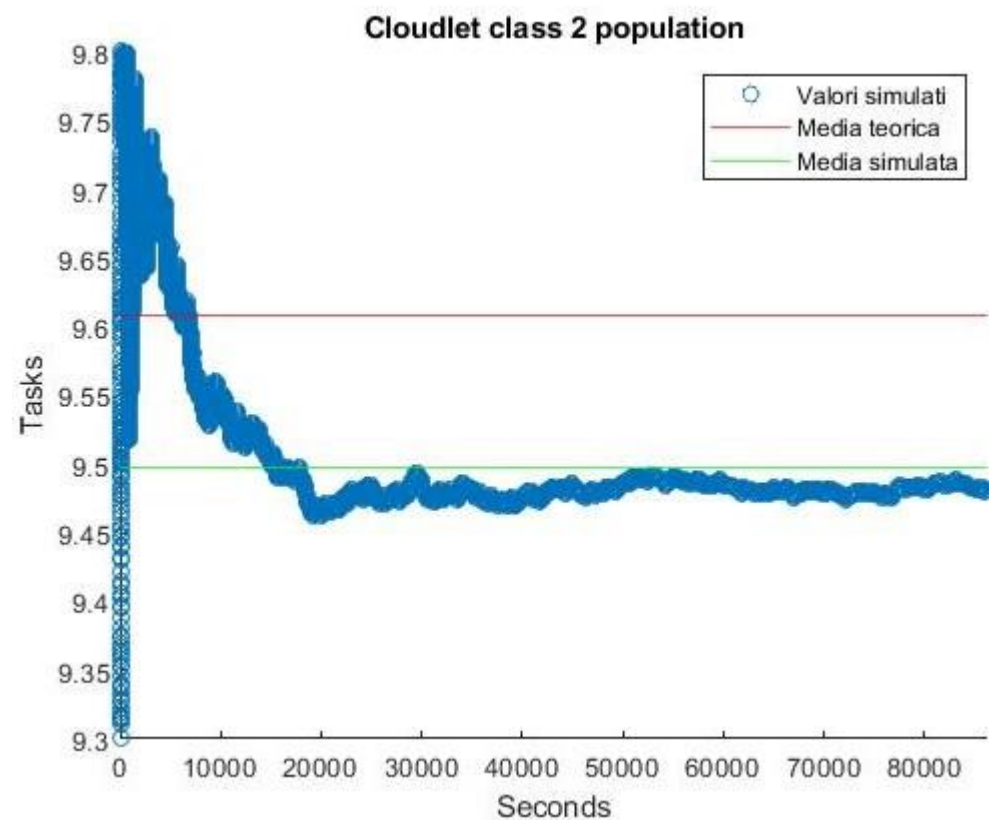
$\alpha: 3.7037 s$

$s: 2.426817 (+/-0.016152) s$

# POPOLAZIONE MEDIA PER CLOUDLET



$\alpha$ : 8.884109 task,  
 $s$ : 9.061018 (+/-0.02000) task



$\alpha$ : 9.608671 task  
 $s$ : 9.497830 (+/-0.02069) task



# VARIANTE ALGORITMO 2

Job di classe 2 prelazionati rieseguono il task partendo dal punto di partenza

Esecuzione di un tempo di set-up

Per permettere lo spostamento senza alcuna perdita di tempo si è utilizzata una euristica che permettesse di approssimare il tempo effettivo rimanente da eseguire sul cloud.

$$E(S_{remaining})_{cloud} = \frac{\mu_1}{\mu_2} \cdot E(S_{remaining})_{clet}$$

	Algoritmo 2	Variante algoritmo 2
Tempo di risposta globale (s)	3.458544 (+/-0.00237)	3.330771 (+/-0.00150)
Tempo di risposta prima classe (s)	2.222094 (+/-0.00274)	2.218547 (+/-0.00207)
Tempo di risposta seconda classe (s)	4.103553 (+/-0.00254)	3.911301 (+/-0.00242)
Tempo di risposta prima classe al cloud (s)	4.130490 (+/-0.00020)	4.213663 (+/-0.11359)
Tempo di risposta seconda classe al cloud (s)	4.538889 (+/-0.01014)	4.146376 (+/-0.00244)
Tempo di risposta prima classe al cloudlet (s)	2.218467 (+/-0.00750)	2.217407 (+/-0.00202)
Tempo di risposta seconda classe al cloudlet (s)	2.426817 (+/-0.016152)	3.698034 (+/-0.00290)