Progetto di simulazione di sistemi:

A queuing system with decomposed service and inventoried preliminary service

Argomenti

La struttura della seguente presentazione seguirà i seguenti macro-punti:

- **Introduzione e Terminologia**
  Parte introduttiva sul problema e descrittiva di alcune terminologie usate
- **Modello**
  Descrizione del modello e di come è stato implementato il sistema da simulare
- **Statistica**
  Descrizione dell'analisi statistica suii risultati ottenuti

Terminologia

Decomposed Service

Fase di servizio del sistema multi-stage suddivisa in due parti:

- Preliminary Service
- Complementary Service

Preliminary Service (PS)

- Presenza di un inventario dove salvare PS pre-calcolati
- Se PS è inattivo produce per l'inventario

Complementary Service (CS)

- Service fisso per ogni Job, senza possibilità di ottimizzazione

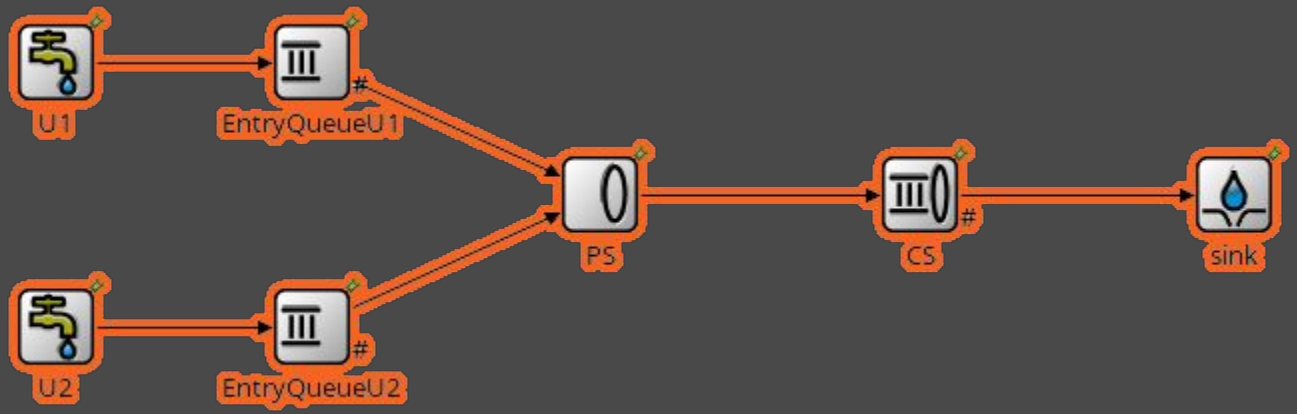
Come funziona il Preliminary Service (PS)

Nel PS è presente un inventario per ogni tipologia di Utente.

Quando entra un Job:

- Se nell'inventario è presente un PS pre-calcolato, il Job viene servito istantaneamente e passa al CS
- Se nell'inventario non è presente un PS pre-calcolato, il Job resta nel PS per il tempo richiesto e poi passa a CS

Descrizione Modello



Legenda

- U1/U2: Sorgenti dei Job
- EntryQueueU1/U2: code all'ingresso
- PS: Preliminary Service
- CS: Complementary Service
- sink: distruttore dei job

Parametri Modello

n: 2, 3, 4, 5
m(U1): 2s, 3s, 4s
z(U1): 0.8s, 1.0s, 1.4s
w(U1): 1.2s, 1.4s, 1.8s
y(U1): 0.8s, 1.4s, 2,3s
p: 0.5, 0.7, 0.8

m(U2): 4s, 5s, 6s
z(U2): 1.0s, 1.2s, 1,4s
w(U2): 1.4s, 1.6s, 1.8s
y(U2): 1.0s, 1.8s, 2,3s

Max PS nell'inventario
Interrarrivo
Produzione PS
Servizio PS
Servizio CS

Implementazione Modello

Source è il modulo che si occupa di generare i job in base a tipo e tempo di interarrivo

PassiveQueue è il modulo che si occupa di mantenere una coda dei job all'ingresso del sistema

Implementazione Modello

Server è il modulo che si occupa di simulare il **Preliminary Service**.

Implementa:

- Inventario dei PS pre-calcolati
- Senza Job al suo interno, pre-calcolo di PS per l'inventario
- Con Job, servizio normale con serviceTime da parametron di configurazione

Implementazione Modello

Queue è il modulo che si occupa di simulare il **Complementary Service**.

Implementa il servizio con tempo serviceTime da parametro di configurazione

Sink è il modulo che si occupa di raccogliere i Job all'uscita del sistema e farne le statistiche

Statistica

- **Transiente Iniziale**
  Visualizzazione dei risultati e individuazione della fase di transiente
- **Analisi dei Risultati**
  Analisi dei risultati scalari prodotti da OMNeT++

Analisi Statistica

In fase di Analisi Statistica sono stati studiati:

- **Tempo di permanenza** nel sistema (Medio, massimo e minimo)
- **Lunghezza della coda** all'ingresso

Con relativi intervalli di confidenza

Permutando **3 tipologie** di configurazione col parametro **n** sono state ottenute 12 configurazioni possibili.

Omnetpp.ini (Estrato del file di config)

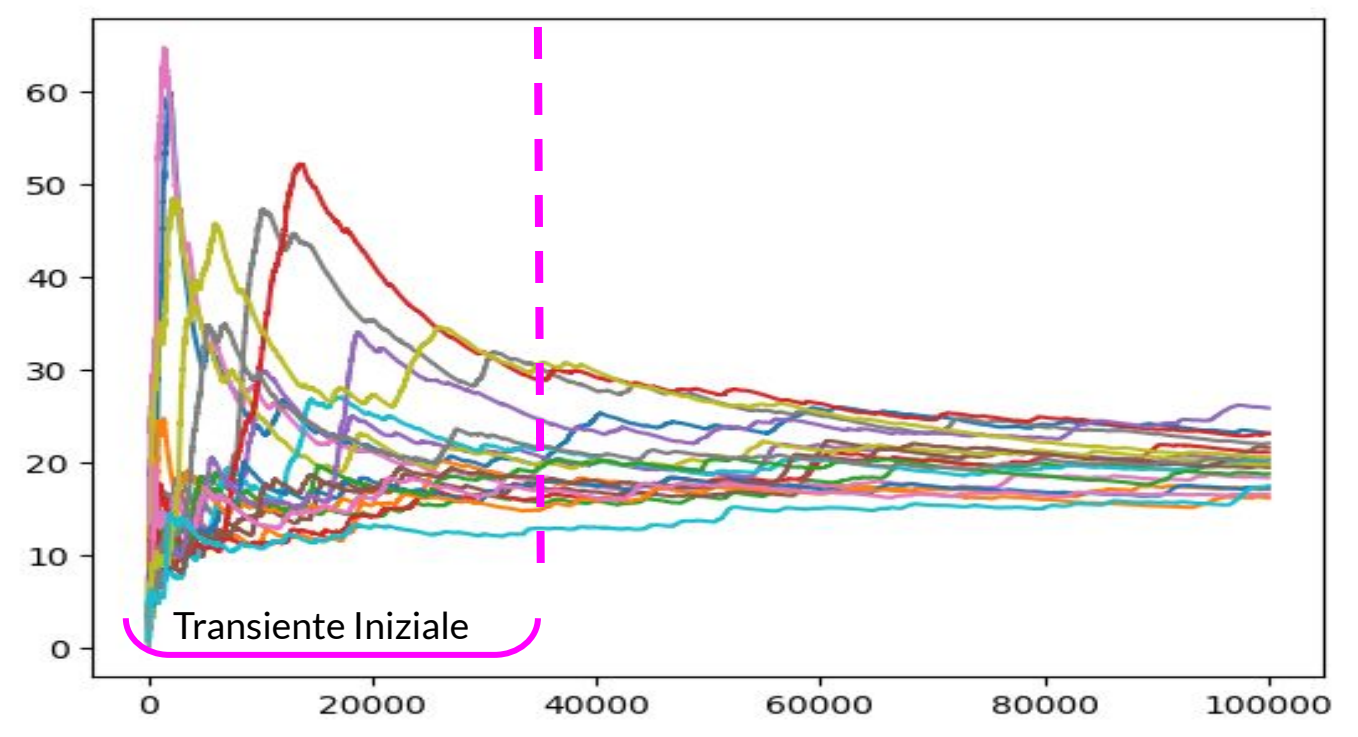
```
[General]
network = Pizza
repeat = 20
replication-label = #${repetition}
sim-time-limit = 100000s
simtime-resolution = ps
warmup-period = 30000s

[Config First]
**.U1.interArrivalTime = exponential(2s)
**.U2.interArrivalTime = exponential(4s)
**.PS.serviceTimeForInventoryU1 = exponential(0.8s)
**.PS.serviceTimeForInventoryU2 = exponential(1.0s)
**.PS.serviceTimeDirectU1 = exponential(1.2s)
**.PS.serviceTimeDirectU2 = exponential(1.4s)
**.PS.probabilityProducing = 0.5
**.CS.serviceTimeU1 = exponential(0.8s)
**.CS.serviceTimeU2 = exponential(1.0s)


[Config First-n2]
extends = First
**.PS.maxInventoriedPS = 2
```

Configurazioni generali:
- num ripetizioni
- warm-up per il transiente
- limite di tempo

Parametri fissi per ogni modulo per ogni configurazione

Combinazione tra configurazione generica e valore di n

Fase di Transiente Iniziale



L'analisi statistica finale è stata eseguita scartando la fase transiente iniziale

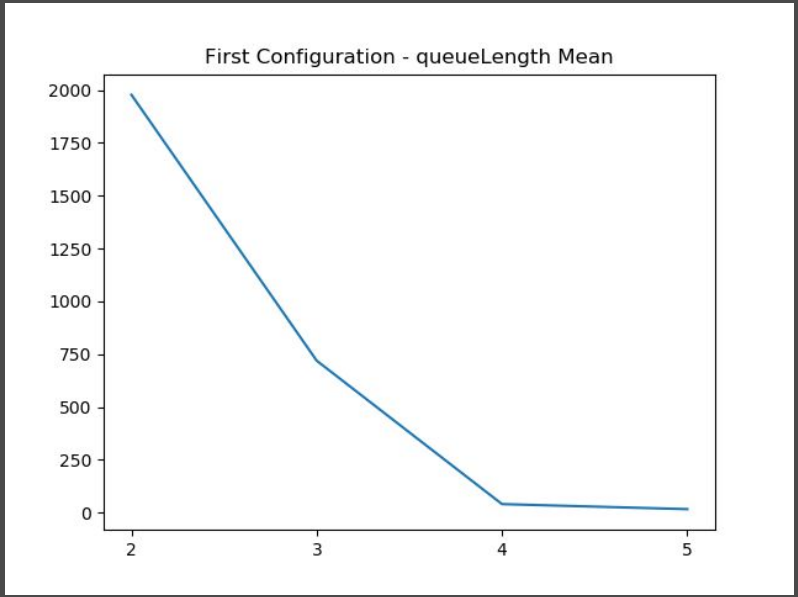
Osservazioni sui Risultati

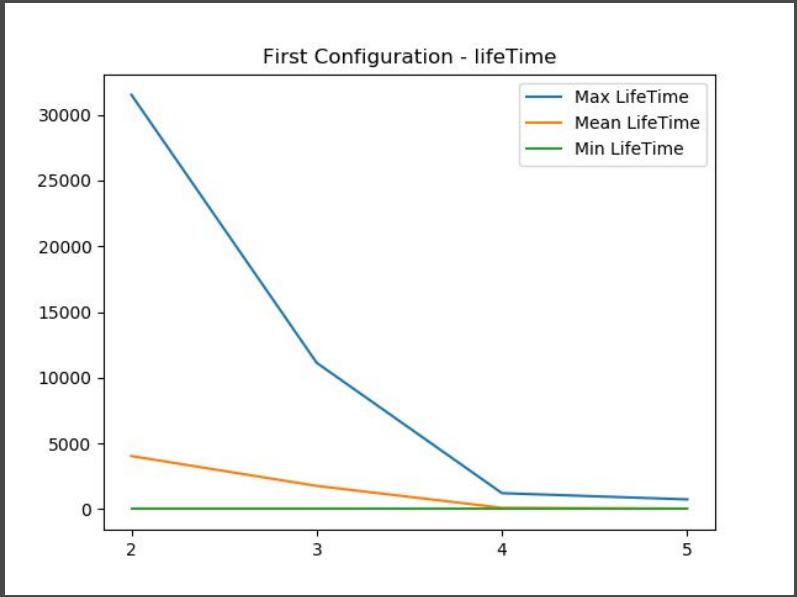
Dai risultati ottenuti, risulta evidente le migliorie sul sistema che comporta l'incremento dei PS pre-calcolati.

Miglioramenti misurabili soprattutto su:

- Massimo tempo di permanenza
- Tempo di permanenza medio
- Lunghezza della coda all'ingresso

Osservazioni sui Risultati





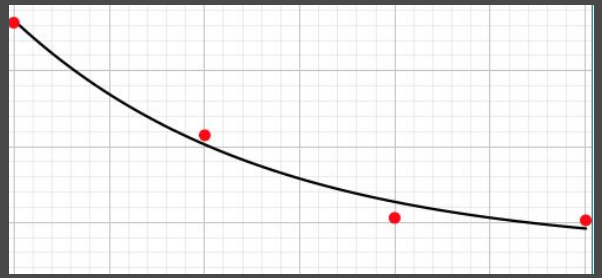
Osservazioni sui Risultati

n	First	Second	Third
2	5251.443	19.666	7269.2
3	2283.206	8.649	5818.03
4	101.426	6.305	4840.417
5	39.611	5.063	4398.531

La tabella a lato contiene la media delle medie dei tempi di permanenza nel sistema di tutte le 20 run.

L'andamento dei valori di tutte e tre le configurazione segue una curva esponenziale negativa (a, c < 0)

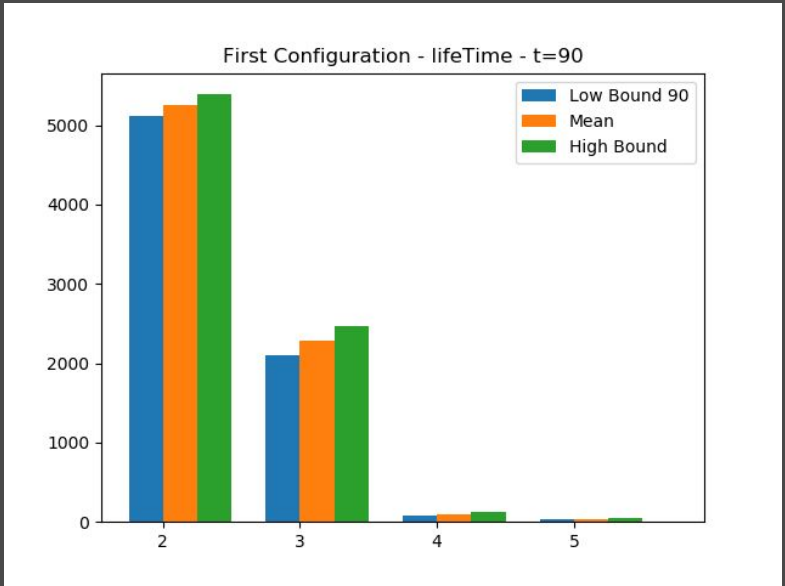
$$y = a + b\,e^{cx}$$



Intervalli di Confidenza

n	Mean	Min95	Max95
2	5251.443	5115.522	5387.364
3	2283.206	2097.268	2469.145
4	101.426	81.656	121.195
5	39.611	36.269	42.952

- min = media - (t95 * std_dev * √var / √n)
- max = media + (t95 * std_dev * √var / √n)



Riferimenti

- **A queuing system with decomposed service and inventoried preliminary services**
  Gabi Hanukova, Tal Avinadava, Tatyana Chernonoga , Uriel Spiegel, Uri Yechiali
- **OMNeT++**
  https://omnetpp.org/intro
- **Python 3.7**
  https://www.python.org/