

# Queueing System with Potential for Recruiting Secondary Servers

Luca Lombardo

Seminario per Metodi Numerici per Catene di Markov

# Struttura del seminario

- 1 Introduzione
- 2 Modello Matematico
- 3 Studio del modello di coda in stato stazionario
  - QBD
    - Condizione di ergodicità del processo QBD
    - Calcolo della distribuzione stazionaria del processo QBD
  - Approccio G1/M/1
- 4 Risultati Numerici
- 5 Conclusioni

# Queueing Theory

- I modelli di coda sono strumenti matematici utilizzati per rappresentare sistemi di risorse che devono essere utilizzati da diversi utenti, chiamati "clienti".
- La terminologia deriva da applicazioni come gli sportelli bancari o i caselli autostradali, dove i clienti effettivamente si mettono in coda finché non vengono serviti da un dipendente.

# Queueing Theory

## Code semplici

- Le code semplici consistono di un solo server che attende un solo cliente alla volta, in ordine di arrivo
- Si assume che il tempo sia discretizzato in intervalli di lunghezza fissa
- Un numero casuale di clienti si unisce al sistema durante ogni intervallo
- Il server rimuove un cliente dalla coda alla fine di ogni intervallo, se presente

# Queueing Theory

Sia  $\alpha_n$  il numero di nuovi arrivi durante l'intervallo  $[n-1, n)$  e  $X_n$  il numero di clienti nel sistema al tempo  $n$ , abbiamo:

$$X_{n+1} = \begin{cases} X_n + \alpha_{n+1} - 1 & \text{se } X_n + \alpha_{n+1} \geq 1 \\ 0 & \text{se } X_n + \alpha_{n+1} = 0 \end{cases}$$

Se  $\alpha_n$  è una collezione di variabili casuali indipendenti, allora  $X_{n+1}$  è condizionalmente indipendente da  $X_0, \dots, X_{n-1}$  se  $X_n$  è noto. Se, inoltre, le  $\alpha_n$  sono identicamente distribuite, allora  $X_n$  è omogenea.

# Queueing Theory

Lo spazio degli stati è  $\mathbb{N}$  e la matrice di transizione è

$$P = \begin{pmatrix} q_0 + q_1 & q_2 & q_3 & q_4 & \dots \\ q_0 & q_1 & q_2 & q_3 & \ddots \\ \vdots & q_0 & q_1 & q_2 & \ddots \\ 0 & & \ddots & \ddots & \ddots \end{pmatrix}$$

$q_i$  è probabilità  $P[\alpha = i]$  che  $i$  nuovi clienti che entrino in coda durante un intervallo di un'unità di tempo

$\alpha$  denota ognuna delle possibili distribuzioni di  $\alpha_n$  identicamente distribuite.

# Queueing Theory

Le catene di Markov aventi matrice di transizione della forma

$$P = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 & \dots \\ A_0 & A_1 & A_2 & A_3 & \ddots \\ & A_0 & A_1 & A_2 & \ddots \\ 0 & & \ddots & \ddots & \ddots \end{pmatrix}$$

dove  $A_i, B_{i+1}, i \geq 0$  sono matrici non negative di dimensione  $k \times k$ , sono dette M/G/1-type Markov Chains

# Obiettivi del paper

Nuovo approccio per migliorare i modelli di coda utilizzando server secondari temporanei reclutati tra

- Server secondari disponibili solo temporaneamente e servono gruppi di diversa dimensione
- Dopo aver servito un gruppo, i server secondari lasciano il sistema



# Obiettivi del paper

Il paper analizza anche altri approcci di modelli di coda con server secondari presenti in letteratura, ma si concentra su due caratteristiche che sono intrinseche in alcuni sistemi del mondo reale e non sono state studiate in passato:

- Server secondari assegnati ad un gruppo e offrono i servizi uno alla volta
- Con probabilità, un cliente servito da un server secondario diventa insoddisfatto e torna al sistema principale per ottenere un nuovo servizio

# Markovian arrival process (*MAP*)

- Un *MAP* è un processo stocastico che descrive il comportamento degli arrivi in un sistema di coda.
- È caratterizzato dalla sua distribuzione di probabilità di interarrivo, che descrive il tempo tra due arrivi consecutivi, e dalla sua distribuzione di probabilità di dimensione, che descrive il numero di entità che arrivano contemporaneamente.
- Può essere definito come un processo di Markov a tempi continui, dove la probabilità di transizione dipende solo dallo stato corrente del sistema e non dalla sua storia passata.

## Caratterizzazione del *MAP*

- Il generatore irriducibile del processo di arrivo markoviano (*MAP*) è dato dalla somma delle matrici di parametro  $D_0$  e  $D_1$  di ordine  $m$ .

L'invariante di probabilità  $\delta$  soddisfa l'equazione

$$\delta(D_0 + D_1) = 0 \quad \delta e = 1$$

- La matrice  $D_0$  governa le transizioni del generatore sottostante che non producono arrivi, mentre la matrice  $D_1$  governa quelle transizioni corrispondenti agli arrivi nel sistema.

# Proprietà del *MAP*

Rate medio di arrivi ( $\lambda$ )

$$\lambda = \delta D_1 e$$

Varianza degli tempi interni di arrivo ( $\sigma^2$ )

$$\sigma^2 = \frac{2}{\lambda} \delta (-D_0)^{-1} e - \frac{1}{\lambda^2}$$

Correlazione ( $\rho_c$ ) tra due successivi tempi interni di arrivo

$$\rho_c = \frac{\lambda \delta (-D_0)^{-1} D_1 (-D_0)^{-1} e - 1}{2 \lambda \delta (-D_0)^{-1} e - 1}$$

# Modello di coda con server principale e secondario

Il sistema ha un singolo server che offre servizi in modo FCFS.

- Il server principale offre servizi esponenziali con parametro  $\mu_1$ .
- Con probabilità  $p$ , un cliente servito può essere reclutato per diventare un server secondario, che offre servizi ai clienti in attesa del sistema.
- I tempi di servizio del server secondario sono esponenziali con parametro  $\mu_2$ .

## Achtung!

Un cliente insoddisfatto dal servizio ricevuto dal server secondario potrebbe richiedere di essere servito di nuovo con probabilità  $v$ .

# Modello di coda con server principale e secondario

Il sistema può avere al massimo due server in qualsiasi momento.

- Il server secondario sarà assegnato a un gruppo di  $i$  clienti dove  $i = \min\{\text{numero nella coda}, L\}$ , ed  $L$  è un pre-determinato positivo finito intero.
- I clienti insoddisfatti sono reinseriti nel sistema. Quando il server secondario ha finito di servire tutti i clienti assegnati, viene rilasciato dal sistema.

## Edge case

Il caso in cui  $\nu = 1$  non è interessante poiché ogni cliente servito da un server secondario viene reinserito nel sistema e l'assunzione di server secondari rallenta solo il sistema nell'offrire servizi.

# Struttura del sistema

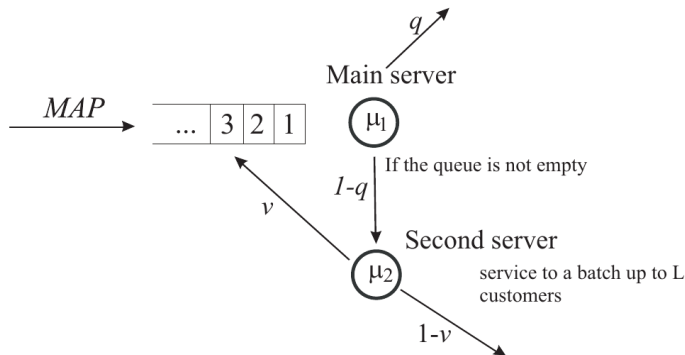


Figure: Immagine da [1]

## Due approcci possibili

### QBD

Primo processo che analizzeremo in questa sezione: un caso particolare della catena di Markov a tempo continuo (CTMC)

### G1/M/1

Una G1/M/1-type Markov chain assume che il tempo tra gli arrivi e il tempo di servizio dei clienti seguano una distribuzione generica, mentre è presente un solo server.



# Introduzione al QBD

Un *quasi-death-birth process* (QBD) è un caso particolare di una catena di Markov a tempo continuo (CTMC) che può essere utilizzato per modellare certi tipi di sistemi di coda. Ci sono due tipi di eventi che possono verificarsi: eventi di morte e eventi di nascita.

- Un evento di morte avviene quando un cliente lascia il sistema (i.e finisce di essere servito e se ne va)
- Un evento di nascita avviene quando un nuovo cliente entra nel sistema

# Introduzione al QBD

Imponendo le restrizioni di entrambi i tipi di code  $M/G/1$  che delle  $G/M/1$ , si vietano transizioni di più di livello alla volta, ottenendo così un processo QBD.

La matrice di transizione di tale processo è definita come segue:

$$P = \begin{pmatrix} B_0 & A_1 & & 0 \\ A_{-1} & A_0 & A_1 & \\ & A_{-1} & A_0 & A_1 \\ & & A_{-1} & A_0 & \ddots \\ 0 & & & \ddots & \ddots \end{pmatrix}, \quad A_{-1}, A_0, A_1 \in \mathbb{R}^{m \times m}, \quad B_0 \in \mathbb{R}^{m \times m}$$

# Introduzione al QBD

## Generalizzazione del QBD

Si potrebbe pensare al QBD come un semplice lista lineare in evoluzione: ogni livello è un nodo nella lista ed il processo è autorizzato a muoversi da un nodo ad uno dei suoi due vicini.

# Generatore infinitesimale del processo QBD

Il generatore infinitesimale di un processo QBD è una matrice tridiagonale a blocchi infinita che descrive la probabilità di transizione del sistema da uno stato  $i$  ad uno stato  $j$ , in un dato istante di tempo  $t$ , attraverso un evento infinitesimo

# Generatore infinitesimale del processo QBD

Al tempo  $t \geq 0$ , indichiamo:

- $i_t \geq 0$  il numero di clienti nel sistema
- $n_t \in \{0, \dots, \min(i_t, L)\}$  il numero di clienti in servizio al server secondario
- $\xi_t = 1, \dots, m$  lo stato del processo sottostante del *MAP* che descrive gli arrivi dei clienti

Allora, il processo stocastico  $\{\zeta_t = (i_t, n_t, \xi_t), t \geq 0\}$  che descrive il comportamento del modello in esame è un CTMC regolare e irriducibile.

# Generatore infinitesimale del processo QBD

Enumerando gli stati della CTMC,  $\{\zeta_t, t \geq 0\}$ , in ordine lessicografico e indicando con  $i$  il livello, per  $i \geq 0$ , definiamo l'insieme di stati come

$$\{(i, n, k) : 0 \leq n \leq \min(i, L), 1 \leq k \leq m\}$$

# Generatore infinitesimale del processo QBD

## Theorem

*Il generatore infinitesimale  $Q$  del processo stocastico CTMC  $\{\zeta_t, t \geq 0\}$  ha una struttura a blocchi tridiagonale come segue:*

$$Q = \begin{pmatrix} Q_{0,0} & Q_{0,1} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots \\ Q_{1,0} & Q_{1,1} & Q_{1,2} & 0 & \dots & 0 & 0 & 0 & 0 & \dots \\ 0 & Q_{2,1} & Q_{2,2} & Q_{2,3} & \dots & 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\ 0 & 0 & 0 & 0 & \dots & Q_{L,L-1} & Q_{L,L} & Q^+ & 0 & \ddots \\ 0 & 0 & 0 & 0 & \dots & 0 & Q^- & Q^0 & Q^+ & \dots \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & Q^- & Q^0 & Q^+ \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

# Generatore infinitesimale del processo QBD

Dove i blocchi  $Q_{i,i}$  sono definiti come segue:

$$Q_{0,0} = D_0$$

$$Q_{i,i} = I_{i+1} \otimes v\mu_2 E_i^- \otimes I_m - (\mu_1 \hat{l}_i + \mu_2 (I_{i+1} - \bar{l}_i)) \otimes I_m \quad 1 \leq i \leq L$$



# Generatore infinitesimale del processo QBD

$$Q_{i,i} = I_{i+1} \otimes \nu \mu_2 E_i^- \otimes I_m - (\mu_1 \hat{I}_i + \mu_2 (I_{i+1} - \bar{I}_i)) \otimes I_m \quad 1 \leq i \leq L$$

Dove:

$\otimes$  indica il prodotto di Kronecker per matrici

$E_i^-$  è una matrice quadrata di dimensioni  $i+1$  con  $(E_i^-)_{k,k-1} = 1$  per  $1 \leq k \leq i$  e tutte le altre componenti nulle.

$\hat{I}_i$  è una matrice quadrata di dimensioni  $i+1$  con  $(\hat{I}_i)_{k,k} = 1$  per  $0 \leq k \leq i-1$  e tutte le altre componenti nulle.

$\bar{I}_i$  è una matrice quadrata di dimensioni  $i+1$  con  $(\bar{I}_i)_{0,0} = 1$  e tutte le altre componenti nulle.

# Generatore infinitesimale del processo QBD

Mentre abbiamo

$$Q_{i,i+1} = E_i^+ \otimes D_1 \quad 0 \leq i \leq L-1$$

$$Q_{1,0} = (1-v)\mu_2 \tilde{E}_1^- \otimes I_m + \mu_1 I_1^- \otimes I_m \quad 1 \leq i \leq L$$

$$Q_{i,i-1} = (1-v)\mu_2 \tilde{E}_i^- \otimes I_m + q\mu_1 I_i^- \otimes I_m + (1-q)\mu_1 I_i^+ \otimes I_m \quad 1 \leq i \leq L$$

# Generatore infinitesimale del processo QBD

Dove

$E_l^+$  è una matrice di dimensioni  $(l+1) \times (l+2)$  con  $(E_l^+)_{k,k} = 0$  per  $0 \leq k \leq l$  e tutte le altre componenti nulle.

$\tilde{E}_l^-$  è una matrice di dimensioni  $(l+1) \times l$  con  $(\tilde{E}_l^-)_{k,k-1} = 1$  per  $1 \leq k \leq l$  e tutte le altre componenti nulle.

$I_l^-$  è una matrice di dimensioni  $(l+1) \times l$  con  $(I_l^-)_{k,k} = 1$  per  $0 \leq k \leq l-1$  e tutte le altre componenti nulle.

$I_l^+$  è una matrice di dimensioni  $(l+1) \times l$  con  $(I_l^+)_{0,l-1} = 1, (I_l^+)_{k,k} = 1$  per  $1 \leq k \leq l-1$  e tutte le altre componenti nulle.

# Introduzione alla proprietà di ergodicità

## Definizione

Una catena di Markov si dice *ergodica* se il limite

$$\pi(j) = \lim_{n \rightarrow \infty} \mathbb{P}_i\{X_n = j\}$$

esiste per ogni stato  $j$  e non dipende dallo stato iniziale  $i$ . Questo può anche essere scritto come

$$\pi(j) = \lim_{n \rightarrow \infty} (P^n)_{i,j}$$

## Condizione di ergodicità del processo QBD

Il seguente risultato stabilisce la condizione di ergodicità del processo QBD che governa il sistema in esame.

### Theorem

*Il processo stocastico CTMC  $\{\zeta_t, t \geq 0\}$  è ergodico se e solo se vale la seguente disuguaglianza:*

$$\lambda < \mu_1 + \mu_2(1 - \nu) \frac{L(1 - q)\mu_1}{L(1 - q)\mu_1 + \mu_2}$$

# Dimostrazione del teorema

## Dimostrazione

Il criterio per l'ergodicità del QBD con il generatore di forma data come nel teorema precedente soddisfa l'ineguaglianza:

$$yQ^-e > yQ^+e$$

dove il vettore  $y$  è l'unica soluzione del sistema

$$y(Q^- + Q^0 + Q^+) = 0, \quad ye = 1$$

con

$$Q^+ = I_{L+1} \otimes D_1, \quad i \geq L$$

$$Q^- = (1-v)\mu_2 E_L^- \otimes I_m + q\mu_1 I_{(L+1)m} + (1-q)\mu_1 I^+ \otimes I_m \quad i > L$$

$$Q^0 = I_{L+1} \otimes D_0 + v\mu_2 E_L^- \otimes I_m - (\mu_1 I_{L+1} + \mu_2 (I_{L+1} - \bar{I}_L)) \otimes I_m \quad i > L$$

## Dimostrazione

Si può inoltre verificare che

$$Q^- + Q^0 + Q^+ = I_{L+1} \otimes (D_0 + D_1) + S \otimes I_m$$

dove

$$S = \begin{pmatrix} -\mu_1(1-q) & 0 & 0 & \dots & 0\mu_1(1-q) & \\ \mu_2 & -\mu_2 & 0 & \dots & 0 & 0 \\ 0 & \mu_2 & -\mu_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \mu_2 & -\mu_2 \end{pmatrix}$$

## Dimostrazione

dove usando le regole del mixed product per il prodotto di Kronecker, e ricordando che

$$\delta(D_0 + D_1) = 0, \quad \delta e = 1$$

si verifica che

$$y = x \otimes \delta$$

dove  $x$  è soluzione del sistema

$$xS = 0, \quad xe = 1$$



## Dimostrazione

per sostituzione diretta, verifichiamo che le componenti del vettore  $x = (x_0, x_1, \dots, x_L)$ , corrispondenti alle uniche soluzioni del sistema visto prima, sono date da

$$x_0 = \frac{\mu_2}{L(1-q)\mu_1 + \mu_2}, \quad x_i = \frac{\mu_1(1-q)}{L(1-q)\mu_1 + \mu_2}, \quad i = 1, \dots, L$$

La tesi segue dalle equazioni viste in precedenza assieme alla definizione di  $\lambda$ . □

# Osservazioni sulla dimostrazione

## Osservazione 1

- La condizione di ergodicità richiede che il tasso di arrivo dei clienti per unità di tempo debba essere inferiore al tasso di servizio che i clienti ricevono per unità di tempo quando il sistema è sovraccarico.
- Il tasso di servizio medio totale nel modello di coda è dato dalla somma del tasso di servizio fornito dal server principale e del tasso di servizio fornito dal server secondario.

Possiamo esprimere il tasso di servizio medio totale come segue:

$$\mu = \mu_1 + \mu_2(1 - v) \frac{L(1 - q)\mu_1}{L(1 - q)\mu_1 + \mu_2}$$

# Osservazioni sulla dimostrazione

## Osservazione 2

Calcoliamo la probabilità  $x_0$  che il secondo server non sia presente nel sistema in un qualsiasi momento in cui il sistema è sovraccarico.

- Quando il sistema attiva un server secondario la durata media del server secondario continuamente presente nel sistema è data da  $\frac{L}{\mu_2}$ . Pertanto, abbiamo:

$$x_0 = \frac{\frac{1}{\mu_1(1-q)}}{\frac{1}{\mu_1(1-q)} + \frac{L}{\mu_2}} = \frac{\mu_2}{L(1-q)\mu_1 + \mu_2}$$

# Probabilità stazionarie

La distribuzione stazionaria di una CTMC irriducibile ricorrente è la distribuzione di probabilità a cui il processo converge per valori grandi di  $t$ .

In particolare, per un processo QBD con  $n$  stati, la distribuzione stazionaria è un vettore di probabilità

$$\pi = (\pi_1, \pi_2, \dots, \pi_n)$$

dove ogni  $\pi_i$  rappresenta la probabilità di trovare il sistema nello stato  $i$ .

# Probabilità stazionarie

Sotto l'assunzione che la condizione di ergodicità sia valida, esistono le seguenti probabilità stazionarie degli stati del CTMC  $\{\zeta_t, t \geq 0\}$ :

$$\pi(i, n, \zeta) = \lim_{t \rightarrow \infty} P\{i_t = i, n_t = n, \zeta_t = \zeta\}, \quad i \geq 0$$

Consideriamo i vettori riga delle probabilità di stato stazionario  $\pi_i$  come segue

$$\pi(i, n) = (\pi(i, n, 1), \dots, \pi(i, n, m))$$

$$\pi_i = (\pi(i, 0), \dots, \pi(i, \min\{i, L\})), \quad i \geq 0$$

# Probabilità stazionarie

Sappiamo che i vettori di probabilità stazionari  $\pi_i, i \geq 0$ , soddisfano il sistema di equazioni algebriche lineari (equazioni di equilibrio):

$$(\pi_0, \pi_1, \pi, \dots)Q = 0 \quad (\pi_0, \pi_1, \pi, \dots)\mathbf{e} = 1$$

dove  $Q$  è la matrice di transizione del CTMC  $\{\zeta_t, t \geq 0\}$  ed  $\mathbf{e}$  è il vettore colonna di tutti gli elementi 1

# Algoritmo per risolvere il sistema di equazioni di equilibrio

Vediamo un algoritmo che sfrutta la struttura tridiagonale a blocchi, ma dipendente dal livello, del generatore per i livelli inferiori a  $L + 1$ , in modo da risolvere più efficientemente il sistema di equazioni lineari algebriche quando il numero di livelli di confine è elevato.

# Algoritmo per risolvere il sistema di equazioni di equilibrio

## Theorem

*I vettori  $\pi_i, i \geq 0$ , sono trovati come soluzione del sistema di equazioni algebriche lineari:*

$$\pi_i = \alpha_i \left( \sum_{l=0}^{\infty} \alpha_l e \right)^{-1}, \quad i \geq 0$$

*dove il vettore  $\alpha_0$  è calcolato come l'unica soluzione del sistema di equazioni*

$$\alpha_0 (Q_{0,0} + Q_{0,1} G_0) = 0, \quad \alpha_0 e = 1$$

*ed i vettori  $\alpha_i, i \geq 1$ , sono definiti come*

$$\alpha_i = \alpha_0 \prod_{l=1}^i R_l, \quad i \geq 1$$



# Algoritmo per risolvere il sistema di equazioni di equilibrio

## Theorem

*Altrimenti tramite la formula ricorsiva*

$$\alpha_i = \alpha_{i-1} R_i, \quad i \geq 1$$

dove

$$R = \begin{cases} -Q_{i-1,i}(Q_{i,i} + Q_{i,i+1}G_i)^{-1}Q & 1 \leq i \leq L-1 \\ -Q_{L-1,L}(Q_{L,L} + Q^+G)^{-1} & i = L \\ -Q^+(Q^0 + Q^+G)^{-1} = R & i > L \end{cases}$$

# Algoritmo per risolvere il sistema di equazioni di equilibrio

## Theorem

*Le matrici stocastiche  $G_i$  sono calcolate utilizzando la seguente formula ricorsiva all'indietro:*

$$G_L = G$$

$$G_{L-1} = -(Q_{L,L} + Q^+ G_L)^{-1} Q_{L,L-1}$$

$$G_i = -(Q_{i+1,i+1} + Q_{i+1,i+2} G_{i+1})^{-1} Q_{i+1,i}, \quad i = L-2, L-3, \dots, 0$$

*dove la matrice  $G$  è la minima soluzione non negativa dell'equazione quadratica matriciale*

$$Q^+ G^2 + Q^0 G + Q^- = 0$$

# Algoritmo per risolvere il sistema di equazioni di equilibrio

- L'algoritmo proposto è una modifica dell'algoritmo per il calcolo della distribuzione stazionaria di una CTMC asintotica quasi-Toeplitz.
- Utilizzando la ricorsione di vettori anziché quella di matrici si ha una significativa riduzione della memoria del computer e del tempo di esecuzione.
- Le inverse delle matrici utilizzate nell'algoritmo sono sub-generatori irriducibili e semi-stabili, il che rende stabile l'implementazione numerica dell'algoritmo.

# Introduzione alle code di tipo GI/M/1

Una coda di tipo GI/M/1 è un processo stocastico che modella il comportamento di un sistema di code con un singolo server

- GI *General inter-arrival time distribution* distribuzione del tempo tra gli arrivi dei clienti alla coda.
- M *Markovian service time distribution*: si riferisce alla distribuzione dei tempi di servizio per ciascun cliente, che viene assunta essere un processo di Markov.
- 1 *One server*: un solo server nel sistema, e che solo un cliente alla volta può essere servito.

# Code di tipo GI/M/1

Definiamo come prima cosa lo spazio degli stati  $\Omega$  del CTMC come:

$$\Omega = \{(i, j, k) : i \geq 0, 0 \leq j \leq K, 1 \leq k \leq m\}$$

Definiamo il livello  $i$  come:

$$i = \{(i, j, k) : 0 \leq j \leq L, 1 \leq k \leq m\} = \{(i, 0), \dots, (i, L)\}, \quad i \geq 0$$

# Code di tipo GI/M/1

## Osservazione

- il livello  $(i,j)$  indica che il server principale è occupato, ci sono  $i - 1$  clienti in attesa nella coda principale; il server secondario è occupato e il processo di arrivo si trova in varie fasi
- Il livello  $(0,0)$  corrisponde al sistema inattivo con il processo *MAP* in una delle  $m$  fasi.

# Il generatore del CTMC

Il generatore  $\tilde{Q}$  della CTMC che governa il sistema in studio è:

$$\tilde{Q} = \begin{pmatrix} B_0 & A_0 & & & & & & & & \\ B_1 & A_1 & A_0 & & & & & & & \\ B_2 & A_2 & A_1 & A_0 & & & & & & \\ \vdots & & \ddots & \ddots & \ddots & & & & & \\ B_L & & & & & A_2 & A_1 & A_0 & & \\ B_{L+1} & & & & & A_2 & A_1 & A_0 & & \\ & A_{L+2} & & & & & & A_2 & A_1 & A_0 \\ & & A_{L+2} & & & & & A_2 & A_1 & A_0 \\ & & & \ddots & & & & \ddots & \ddots & \ddots \end{pmatrix}$$

# Il generatore del CTMC

Dove abbiamo:

$$B_0 = \begin{pmatrix} D_0 & & & & \\ \tilde{v}\mu_2 I & D_0 - \mu_2 I & & & \\ & \tilde{v}\mu_2 I & D_0 - \mu_2 I & & \\ & & \ddots & \ddots & \\ & & & \tilde{v}\mu_2 I & D_0 - \mu_2 I \end{pmatrix}$$



# Il generatore del CTMC

Dove abbiamo:

$$A_0 = \begin{pmatrix} D_1 & & & & \\ v\mu_2 I & D_1 & & & \\ & v\mu_2 I & D_1 & & \\ & & \ddots & \ddots & \\ & & & v\mu_2 I & D_1 \end{pmatrix}$$

$$A_1 = B_0 - \mu_1 I$$

$$A_2 = \mu_1 \Delta(q, 1, \dots, 1)$$

$$B_1 = \mu_1 I$$

$$B_r = \rho \mu_1 (e_r^T \otimes e(L+1)) \quad 2 \leq r \leq L+1$$

$$A_{L+2} = B_{L+1}$$

# Proprietà delle queue di tipo G1/M/1

Utilizzando i risultati per le code di tipo G1/M/1 in tempo continuo, si verificano le seguenti proprietà:

## Proprietà 1

Sia

$$\tilde{y} = (\tilde{y}_0, \dots, \tilde{y}_L)$$

il vettore invariante di  $A = \sum_{i=0}^{L+2} A_i$ . Allora:

$$\tilde{y}_0 = \delta(\mu_2 I - D_0 - D_1)[\mu_2 U + L\rho\mu_1 I - D_0 - D_1]^{-1}$$

$$\tilde{y}_r = \rho\mu_1\pi_0(\mu_2 I - D_0 - D_1)^{-1}, \quad 1 \leq r \leq L$$

# Proprietà delle queue di tipo GI/M/1

## Proprietà 2

La condizione di stabilità

$$\tilde{y}A_0e < \tilde{y} \sum_{i=1}^{L+2} (i-1)A_i e$$

si riduce alla disuguaglianza vista prima:

$$\lambda < \mu_1 + \mu_2(1-\nu) \frac{L(1-q)\mu_1}{L(1-q)\mu_1 + \mu_2}$$

# Proprietà delle queue di tipo GI/M/1

## Proprietà 3

Data  $R$  la matrice di rate, soddisfa l'equazione matriciale non lineare data da:

$$R^{L+2}A_{L+2} + R^2A_2 + RA_1 + A_0 = 0$$

# Proprietà delle queue di tipo GI/M/1

## Proprietà 4

Indicando con  $\tilde{\pi}$  il vettore di probabilità stazionario del generatore  $\tilde{Q}$  come visto prima, otteniamo qui la soluzione matriciale geometrica classica:

$$\tilde{\pi}_i = \tilde{\pi}_0 R^i, \quad i \geq 1$$

dove  $\tilde{\pi}_0$  è ottenuto risolvendo il seguente sistema di equazioni lineari:

$$\tilde{\pi}_0 \left[ \sum_{i=0}^{L+1} R^i B_i \right] = 0, \quad \tilde{\pi}_0 e = 1$$

# Introduzione ai risultati numerici

Vedremo 3 esempi illustrativi utilizzando 5 processi di arrivo. In particolare i 5 *MAP* considerati sono:

## 1. ERL

Erlang di ordine 5 con parametro 2.5 in ciascuno dei 5 stati. Prendiamo poi  $\lambda = 0.5$ ,  $\sigma = 0.899427$  e  $\rho_c = 0$ .

## 2. EXP

Un esponenziale con una frequenza di 0.5. Prendiamo poi  $\lambda = 0.5$ ,  $\sigma = 2$  e  $\rho_c = 0$ .

### 3. HEX

Distribuzione iper-esponenziale con una probabilità di mixing data da  $(0.5, 0.3, 0.15, 0.04, 0.01)$  con i corrispondenti tassi della distribuzione esponenziale pari a  $(1.09, 0.545, 0.2725, 0.13625, 0.068125)$ . Qui abbiamo  $\lambda = 0.5, \sigma = 3.3942$  e  $\rho_c = 0$ .

## 4. NCR

*MAP* negativamente correlato, con matrici di rappresentazione:

$$D_0 = \begin{pmatrix} -1.125 & 0.125 & 0 & 0 & 0 \\ 0 & -1.125 & 0.125 & 0 & 0 \\ 0 & 0 & -1.125 & 0.125 & 0 \\ 0 & 0 & 0 & -0.125 & 0 \\ 0 & 0 & 0 & 0 & -2.25 \end{pmatrix}$$

$$D_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.01125 & 0 & 0 & 0 & 1.11375 \\ 2.2275 & 0 & 0 & 0 & 0.0225 \end{pmatrix}$$

dove abbiamo  $\lambda = 0.5$ ,  $\sigma = 2.02454$  e  $\rho_c = -0.57855$



## 5. PCR

*MAP* positivamente correlato, con matrici di rappresentazione:

$$D_0 = \begin{pmatrix} -1.125 & 0.125 & 0 & 0 & 0 \\ 0 & -1.125 & 0.125 & 0 & 0 \\ 0 & 0 & -1.125 & 0.125 & 0 \\ 0 & 0 & 0 & -0.125 & 0 \\ 0 & 0 & 0 & 0 & -2.25 \end{pmatrix}$$

$$D_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1.11375 & 0 & 0 & 0 & 0.01125 \\ 0.0225 & 0 & 0 & 0 & 2.2275 \end{pmatrix}$$

dove abbiamo  $\lambda = 0.5$ ,  $\sigma = 2.02454$  e  $\rho_c = -0.57855$

# Introduzione ai risultati numerici

## Osservazioni

- Le cinque *MAP* sopra riportate sono qualitativamente diverse.
- Il processo di arrivo **PCR** è ideale per situazioni di arrivi altamente irregolari con periodi di alta e bassa attività.

# Primo esempio illustrativo

## Obiettivo

Discutiamo l'impatto del parametro  $L$  su alcune misure di performance del sistema per tutti e 5 i *MAPs*

Fissiamo  $\mu_1 = 1$ ,  $\mu_2 = 0.5$ ,  $q = 0.5$ , e  $\nu = 0.4$ , e variamo  $L$  da 1 a 30.

## Primo esempio illustrativo

$L_{\text{sec}}$

Definiamo  $L_{\text{sec}}$  come il numero medio di clienti nel sistema con server secondari ad un momento arbitrario come:

$$L_{\text{sec}} = \sum_{i=1}^{\infty} \sum_{n=1}^{\min\{i, L\}} n \pi(i, n) e$$

$L_{\text{system}}$

Definiamo  $L_{\text{system}}$  come il numero medio di clienti nell'intero sistema come:

$$L_{\text{system}} = \sum_{i=1}^{\infty} i \pi_i e$$

## Primo esempio illustrativo

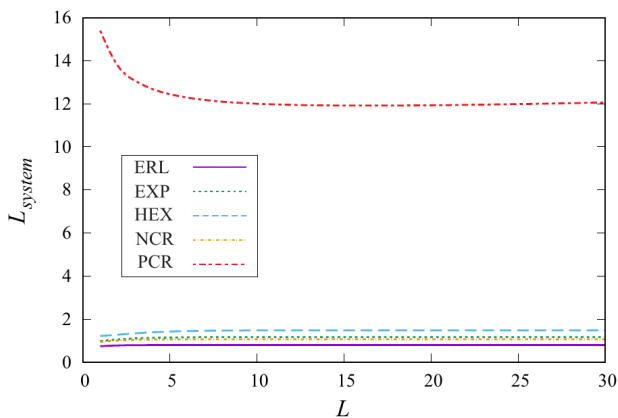


Figure: Impatto di  $L$  sul numero medio di clienti nel sistema  $L_{system}$  per diversi MAPs

## Primo esempio illustrativo

- Il **PCR** ha un alto numero medio di clienti nel sistema rispetto ad altri processi di arrivo.
- $L$  aumenta il numero medio di clienti nel sistema per i primi quattro  $MAP$ , ma per il **PCR** il trend non è crescente a causa della correlazione positiva.
- Alti valori di  $L$  aumentano la probabilità di avere più clienti nel sistema, soprattutto per i primi quattro  $MAP$ .
- Per gli arrivi **PCR**,  $L$  diminuisce il numero medio di clienti nel sistema perché i server secondari aiutano a ripulire la coda.

# Primo esempio illustrativo

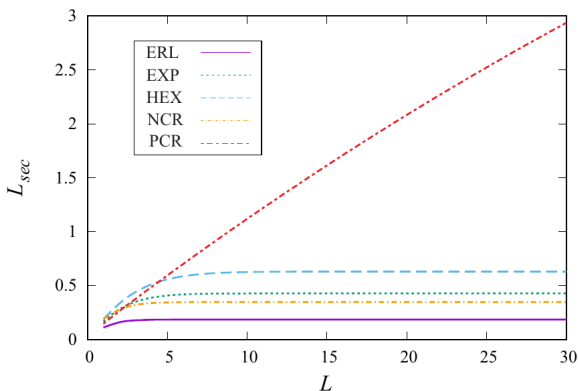


Figure: Dipendenza del numero medio di clienti con il server secondario  $L_{sec}$  al variare di  $L$  per diversi MAPs

## Primo esempio illustrativo

- $L_{\text{sec}}$  aumenta all'aumentare di  $L$ , come previsto.
- Il valore di  $L_{\text{sec}}$  è elevato per **PCR** e solo per valori piccoli di  $L$  è inferiore per **ERL-NCR**.
- L'alta irregolarità degli arrivi nel processo **PCR** causa la "fame" del sistema, durante la quale solo il server primario è occupato offrendo servizi per la maggior parte dei clienti.



# Primo esempio illustrativo

$P_{\text{idle-system}}$

Definiamo la probabilità che il sistema sia in equilibrio ad un momento arbitrario come:

$$P_{\text{idle-system}} = \pi_0 e$$

## Primo esempio illustrativo

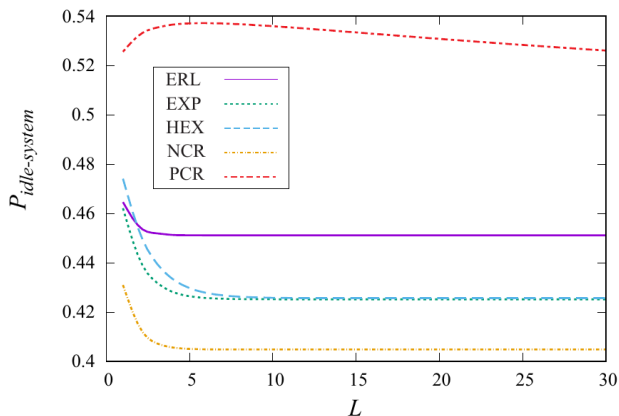


Figure: Dipendenza della probabilità  $P_{idle-system}$  rispetto ad  $L$  che il sistema sia in idle ad un momento arbitrario, per diversi MAPs

## Primo esempio illustrativo

- Esiste una grande differenza nella misura a seconda dei diversi *MAPs* utilizzati.
- Il valore ottimale di  $L$  dipende dall'obiettivo: ad esempio, per il processo di arrivo **PCR**, il valore ottimale di  $L$  è 16 se si cerca di minimizzare  $L_{\text{system}}$ , ma è 6 se si massimizza  $P_{\text{idle-system}}$ .

## Primo esempio illustrativo

### $P_{\text{idle-busy}}$

Definiamo la probabilità che il main server sia in idle quando il server secondario è occupato come:

$$P_{\text{idle-busy}} = \sum_{n=1}^L \pi(n, n) e$$

### $P_{\text{busy-idle}}$

Definiamo la probabilità che il main server sia occupato quando il server secondario è in idle come:

$$P_{\text{busy-idle}} = \sum_{i=0}^{\infty} \pi(i, 0) e$$

# Primo esempio illustrativo

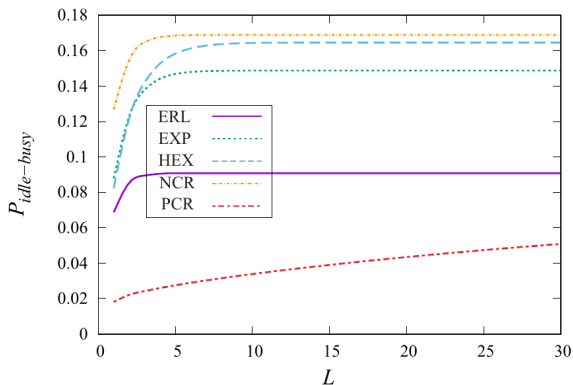
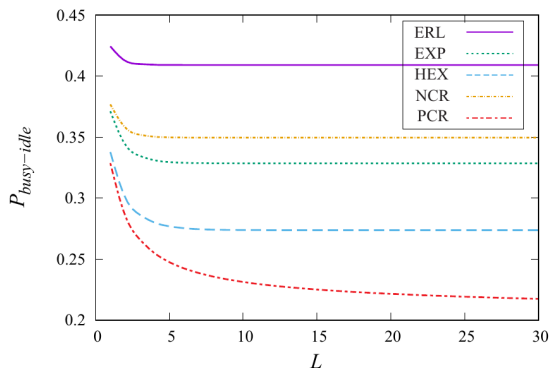


Figure: Dipendenza della probabilità  $P_{idle-busy}$  rispetto ad  $L$  che il main server sia in idle quando il server secondario è in occupato, per diversi MAPs

# Primo esempio illustrativo



**Figure:** Dipendenza della probabilità  $P_{\text{busy-idle}}$  rispetto ad  $L$  che il main server sia occupato quando il server secondario è in idle, per diversi MAPs

## Secondo esempio illustrativo

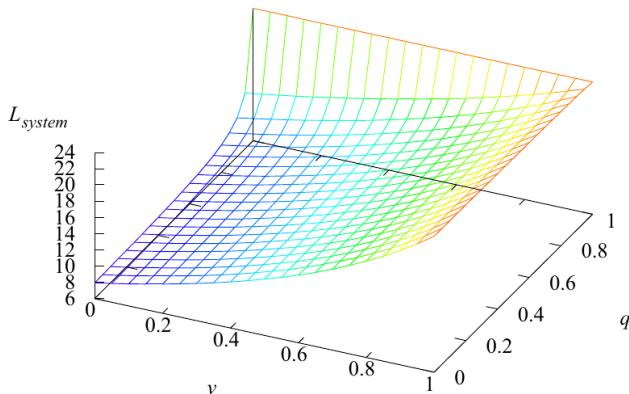
### Obiettivi

L'obiettivo è valutare l'impatto dei parametri  $q$  e  $\nu$  sulla prestazione del sistema. Dove

- $q$  è la probabilità che un cliente servito si rifiuti di agire come server secondario
- $\nu$  è la probabilità che un cliente servito da un server secondario non sia soddisfatto e venga mandato indietro al server primario

Fissiamo il valore di  $L$  a 10 e i tassi di servizio  $\mu_1$  e  $\mu_2$  a 1 e 0.5. Si variano i valori di  $q$  e  $\nu$  da 0 a 1 con passo 0.05 e si analizza l'impatto sulle misure di prestazione del sistema.

## Secondo esempio illustrativo



**Figure:** Dipendenza del numero medio di clienti nel sistema  $L_{system}$  rispetto a  $q$  e  $v$



## Secondo esempio illustrativo

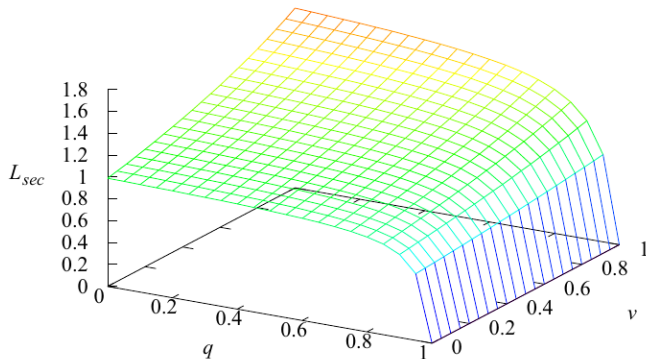
- L'analisi mostra che il valore di  $L_{\text{system}}$  è minimo a 7.9328 quando  $q$  e  $\nu$  sono entrambi uguali a 0.
- Aumentando  $q$  o  $\nu$ , il valore di  $L_{\text{system}}$  aumenta, con un aumento più veloce quando uno o entrambi si avvicinano a 1.
- Quando  $q = 1$ , il sistema diventa un modello *MAP/M/1* classico e il valore di  $L_{\text{system}}$  diventa 22.30425 per tutti i valori di  $\nu$ .
- L'uso di un server secondario riduce il numero medio di clienti nel sistema di oltre il 40%, e il punto di interruzione per il modello classico è  $\nu^* \sim 0.985$ .

## Secondo esempio illustrativo

### Modifichiamo i parametri

- Si aumenta  $\lambda$  del 50% a 0.75 per testare l'importo della riduzione del numero medio di clienti nel sistema.
- Mantenendo gli altri parametri costanti, si ottiene una riduzione superiore al 52,8%.
- Ciò suggerisce che l'aggiunta di un server secondario beneficia notevolmente l'aumento del carico del sistema anche con un tasso di insoddisfazione del cliente del 50%.

## Secondo esempio illustrativo

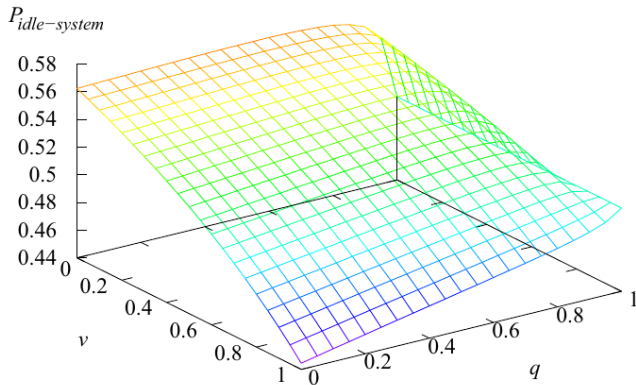


**Figure:** Dipendenza del numero medio di clienti nel sistema  $L_{sec}$  rispetto a  $q$  e  $v$  con  $\lambda = 0.75$

## Secondo esempio illustrativo

- $L_{\text{sec}}$  diminuisce significativamente quando  $q$  si avvicina a 1 e quando i clienti sono raramente reclutati per diventare server secondari.
- $L_{\text{sec}}$  ha il valore massimo quando  $q$  è uguale a zero e  $\nu$  è vicino a 1, ma questo può creare ulteriore lavoro per il sistema e riflettersi negativamente sulla fornitura di servizi.

## Secondo esempio illustrativo



**Figure:** Dipendenza della probabilità  $P_{idle-system}$  che il sistema sia in idle ad un momento arbitrario rispetto a  $q$  e  $v$ .

## Secondo esempio illustrativo

- $P_{\text{idle-system}}$  ha il valore minimo di 0.4445 quando  $\nu = 1$  e  $q = 0$ , il che è intuitivo poiché servire nuovamente i clienti dopo aver passato attraverso un server secondario mette un carico sul sistema.
- $P_{\text{idle-system}}$  aumenta quando  $q$  aumenta e/o  $\nu$  diminuisce, con un valore massimo di 0.5652 ottenuto quando  $q = 0.65$  e  $\nu = 0$ .
- Nel sistema  $MAP/M/1$  classico corrispondente, questa misura è  $P_{\text{idle-system}} = 0.5$ .

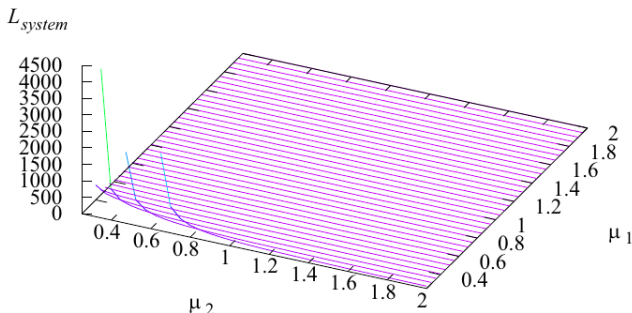
## Terzo esempio illustrativo

### Obiettivo

Analizzare l'impatto della variazione dei tassi di servizio  $\mu_1$  e  $\mu_2$  quando tutti gli altri parametri sono fissati.

- I parametri fissati sono  $L = 10$ ,  $q = 0.5$ ,  $\nu = 0.4$ , e  $\lambda = 0.5$ .
- I tassi  $\mu_1$  e  $\mu_2$  vengono variati da 0.25 a 2.0 con incrementi di 0.05, ma per soddisfare la condizione di ergodicità, il valore di  $\mu_2$  viene limitato quando  $\mu_1$  è piccolo.
- Solo per  $\mu_1 \geq 0.4$ , il valore di  $\mu_2$  può essere variato da 0.25, come originariamente indicato

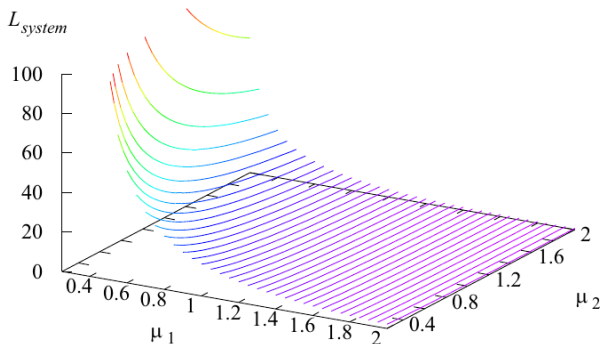
## Terzo esempio illustrativo



**Figure:** Dipendenza del numero medio di clienti nel sistema  $L_{system}$  rispetto a  $\mu_1$  e  $\mu_2$



## Terzo esempio illustrativo

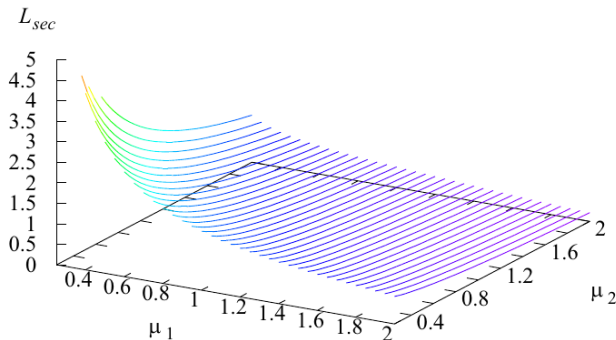


**Figure:** Dipendenza del numero medio di clienti nel sistema  $L_{system}$  rispetto a  $\mu_1$  e  $\mu_2$  (zoomed-in)

## Terzo esempio illustrativo

- La condizione di ergodicità limita il valore di  $\mu_2$  per valori piccoli di  $\mu_1$ .
- Gran parte della superficie mostrata nella prima figura è piatta a causa della violazione della condizione di ergodicità.
- La tendenza generale è che  $L_{\text{system}}$  diminuisce quando aumenta  $\mu_1$  o  $\mu_2$  (per valori non piccoli di  $\mu_1$ ).

## Terzo esempio illustrativo



**Figure:** Dipendenza del numero medio di clienti con server secondario  $L_{sec}$  rispetto a  $\mu_1$  e  $\mu_2$

## Terzo esempio illustrativo

- Il valore massimo di  $L_{sec}$  è di circa 5 quando  $\mu_1$  e  $\mu_2$  sono piccoli.
- Con un aumento di  $\mu_1$  e  $\mu_2$ , il valore di  $L_{sec}$  diminuisce come ci si aspetterebbe.
- Per valori piccoli di  $\mu_1$ , la diminuzione è significativa all'aumentare di  $\mu_2$ ; per valori più grandi di  $\mu_1$ , notiamo un tasso insignificante di diminuzione in  $L_{sec}$  con un aumento di  $\mu_2$ .

# Conclusioni

- Il sistema di coda analizzato prevede la possibilità di reclutare un cliente già servito come server secondario per aiutare il server principale.
- Il processo di arrivo dei clienti è stato modellizzato utilizzando un processo di punto Markoviano versatile, *MAP*.
- È stata considerata la possibilità che i clienti insoddisfatti con il servizio fornito dal server secondario possano ritornare nel sistema.
- L'analisi dello stato stazionario della catena di Markov multidimensionale ha permesso di ottenere risultati numerici utili per prendere decisioni manageriali.

# Generalizzazione del modello

- Si può effettuare il servizio fornito dal server secondario in gruppi.
- Si può rilassare l'ipotesi di avere solo un server secondario e vedere l'impatto dell'aumento a 2.
- Si possono incorporare diverse politiche di controllo, come la possibilità di reclutare molti server secondari con due tipi di clienti