

---

## Parte VII

# Analisi delle prestazioni

---

## Valutazione delle prestazioni

---

- Tuning dei sistemi
  - Monitoraggio dei sistemi durante il funzionamento
  - Analisi delle criticità
  - Tuning della configurazione
- Progetto di nuovi sistemi
  - Caratterizzazione del carico
  - Modelli di prestazione
  - Confronto e scelta delle soluzioni progettuali
- Capacity planning
  - Modelli di evoluzione del carico
  - Dimensionamento del sistema
  - Pianificazione degli upgrade
  - Analisi estesa a tutto il ciclo di vita del sistema

# Misura e modellistica

---

## Approcci complementari

### **Misura**

- Rilevazione durante il funzionamento del sistema
- Fornisce dati affidabili e precisi
- Occorre che il carico sia quello significativo
- Possibile *solo* su sistemi già operativi

### **Modellistica**

- Il modello costituisce un'astrazione del sistema
- Cattura i soli aspetti significativi
- Si ottengono risultati prima che il sistema sia realizzato
- Costi bassi e tempi brevi

# Misura

---

## ➤ Monitor

- Strumenti software per l'analisi delle prestazioni
- Monitor del sistema operativi
- Monitor di ambienti applicativi (es. DBMS)

## ➤ Generazione del carico

- Le prestazioni dipendono dal carico
- Occorrono adeguatezza dei livelli e stabilità
- Il carico reale raramente ha i requisiti

## ➤ Benchmark

- Carico sintetico rappresentante una situazione reale
- Specificità rispetto ai contesti applicativi
- Presenta livelli di parametricità
- Consente il confronto di sistemi diversi
- Es. benchmark TPC-C e TPC-W

# Modellistica

---

- Livello di astrazione
  - Definizione degli obiettivi dello studio
  - Scelta di un livello di astrazione adeguato agli obiettivi
  - La rappresentazione dei dettagli è inessenziale
- Analisi strutturale
  - Il modello evidenzia la struttura del sistema
  - Individuazione dei rapporti causa-effetto
  - Valutazione delle azioni correttive
- Approssimazione
  - Dipende dall'astrazione ma non è un problema
  - La scelta è fra un insieme discreto di alternative
- Analisi parametrica
  - Può essere effettuata in modo semplice ed estensivo
  - Esame di un ampio ventaglio di situazioni

# Modelli analitici

---

- Modelli la cui soluzione è calcolabile in modo algoritmico
- Modelli basati su reti di code trattabili matematicamente
- Impongono semplificazioni nella struttura del modello
  - Modalità di servizio (discipline di scheduling)
  - Distribuzione dei tempi di servizio
  - Modalità di arrivo delle richieste
- La struttura del sistema è molto schematizzata
  - Impossibile rappresentare i dettagli
  - Forniscono risultati approssimati

*Malgrado le semplificazioni, i modelli analitici presentano notevoli vantaggi*

- **Robustezza:** producono risultati attendibili
- **Semplicità:** bassi costi di sviluppo e di calcolo

# Modelli di simulazione

---

**Simulatore:** *programma configurabile che consente di emulare l'evolvere di una situazione reale riproducendone passo per passo lo sviluppo, e consentendo la rilevazione dei relativi indici di prestazioni*

- È possibile rappresentare qualsiasi livello di dettaglio
- I costi naturalmente crescono in proporzione
- Lunghe corse di simulazione necessarie per stimare gli indici di prestazione
- Necessario ricorrere a sofisticate tecniche statistiche
- Possibile stimare anche le distribuzioni dei valori degli indici (oltre ai valori attesi)
- I costi precludono l'effettuazione di analisi parametriche

## Indici di prestazione

---

- Grandezze misurabili che quantificano le prestazioni del sistema
- Dipendono dal contesto applicativo e dal *punto di vista*

### ***Punti di vista***

- *Amministratore del sistema*
  - Guarda allo sfruttamento delle risorse
  - Ottimizza il volume di lavoro svolto
  - Indici: utilizzazioni, throughput
- *Utente del sistema*
  - Guarda alla qualità del servizio
  - Minimizza i tempi di attesa
  - Indici: tempi di risposta

# Utilizzazione delle risorse

---

**Utilizzazione:** *frazione del tempo in cui una risorsa è utilizzata, cioè impegnata a svolgere lavoro a vantaggio del sistema*

## **Esempio: unità a disco**

### ➤ Caratteristiche

- Tempo medio di accesso: 5 ms
- Velocità di trasferimento: 128 MB/s
- Dimensione cluster: 32 KB

### ➤ Richieste di servizio

- Richieste al secondo: 50 rich/s
- Cluster per richiesta: 6 (valore medio)

### ➤ Utilizzazione

- $U_{\text{disk}} = 50 \cdot (0,005 + (6 \cdot 32) / 128000) = 0.325 = 32.5\%$

# Tempo di risposta

---

- Esistono varie definizioni per il tempo di risposta
- La differenza dipende dalla natura del sistema

### ➤ **Sistemi batch**

- *Tempo di turnaround*
- Tempo che intercorre fra la sottomissione di un lavoro e la consegna del relativo risultato

### ➤ **Sistemi interattivi**

- *Tempo di risposta*
- Tempo che intercorre fra la sottomissione di una richiesta e la comparsa della risposta

# Distribuzione del tempo di risposta

---

- Del tempo di risposta non basta conoscere il valore medio
- Interessa anche conoscere la variabilità
- La variabilità è legata strettamente alla qualità del servizio

## **Esempio: operazioni di sportello**

### ➤ Ufficio postale A

- Tempo medio per operazione: 440 s
- Tempo lungo: 1600 s (probabilità 20%)
- Tempo breve: 300 s (probabilità 80%)

### ➤ Ufficio postale B

- Tempo medio per operazione: 440 s
- Tempo lungo: 500 s (probabilità 20%)
- Tempo breve: 425 s (probabilità 80%)

Quale ufficio postale scegliereste?

# Media e varianza

---

- Si considera una variabile aleatoria  $V$  che assume valori diversi con probabilità diverse
  - $V_i, i = 1 \dots n$  : valori diversi assunti dalla variabile  $V$
  - $P_i, i = 1 \dots n$  : probabilità con cui  $V$  assume il valore  $V_i$
  - $\sum P_i = 1$

### ➤ **Media (valore atteso)**

- $E[V] = \sum P_i \cdot V_i$
- È la somma dei valori assunti pesata con la probabilità

### ➤ **Varianza**

- $\text{Var}[V] = \sum P_i \cdot (V_i - E[V])^2$
- Misura gli scarti quadratici rispetto alla media

# Media e varianza: esempio

## Distibuzione A

$i$	1	2	3	4	5	6	7	8	9	10
$P_i$	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
$V_i$	11	12	13	14	15	16	17	18	19	20

Media  $E[V] = 15.5$

Varianza  $\text{Var}[V] = 8.25$

## Distibuzione B

$i$	1	2	3	4	5	6	7	8	9	10
$P_i$	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
$V_i$	2	4	6	8	10	14	18	24	30	39

Media  $E[V] = 15.5$

Varianza  $\text{Var}[V] = 133.45$

*La media è la stessa ma le varianze sono molto diverse*

# Percentili

Il percentile  $p_r$  esprime il valore tale che la probabilità che la variabile aleatoria  $V$  assuma valori inferiori a  $p_r$  è  $r$

$$P[V \leq p_r] = r$$

- I percentili forniscono una caratterizzazione molto efficace delle distribuzioni
- Si tratta di una misura molto più intuitiva della varianza
- Fornisce *una garanzia rispetto alla qualità di servizio*
- Si utilizza spesso nelle specifiche dei sistemi

## Esempio

*Se  $p_{0.90}$  nella distribuzione A vale 19, questo mi garantisce che solo nel 10% dei casi rischio di pagare più di 19*

# Percentili: esempio

## Distribuzione A

$i$	1	2	3	4	5	6	7	8	9	10
$P_i$	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
$V_i$	11	12	13	14	15	16	17	18	19	20

$$p_{0.5} = 16, \quad p_{0.7} = 17, \quad p_{0.9} = 19$$

## Distribuzione B

$i$	1	2	3	4	5	6	7	8	9	10
$P_i$	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
$V_i$	2	4	6	8	10	14	18	24	30	39

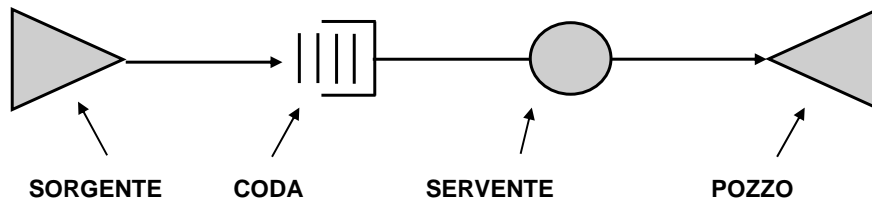
$$p_{0.5} = 10, \quad p_{0.7} = 18, \quad p_{0.9} = 30$$

# Sistemi di congestione

- I sistemi di elaborazione possono essere trattati come *sistemi di congestione*
- Essi sono costituiti da:
  - Risorse : CPU, dischi, linee, periferiche,...
  - Processi di elaborazione : che si contendono l'uso delle risorse che necessitano
- Alle risorse si creano code (cioè file di attesa), e questo rallenta i processi e incide sulle prestazioni

- Per le file d'attesa esistono modelli matematici molto interessanti con soluzioni in forma chiusa (formule)
- I modelli basati sulle code consentono di trarre conclusioni molto interessanti
- Forniscono anche una comprensione strutturale dei sistemi

# Coda isolata



Modella la situazione di congestione nell'accesso ad una risorsa

- **Sorgente**: da cui provengono le richieste di servizio
- **Clients**: rappresentano le singole richieste di servizio
- **Servente**: eroga il servizio ai clienti
- **Coda**: costituita dai clienti che attendono di essere serviti
- **Pozzo**: dove tornano i clienti una volta serviti

## Parametri del problema

$$\lambda = 1 / T_a , \quad \mu = 1 / T_s , \quad \rho = \lambda / \mu = T_s / T_a , \quad T_R = T_s + T_W$$

- $\lambda$  *Frequenza di arrivo*: frequenza con cui i clienti entrano nel sistema
- $T_a$  *Tempo di interarrivo*: tempo medio che intercorre tra l'arrivo di un cliente e quello del successivo
- $\mu$  *Frequenza di servizio*: frequenza con cui i clienti sono serviti
- $T_s$  *Tempo di servizio*: tempo medio necessario a servire un cliente
- $\rho$  *Utilizzazione*: frazione di tempo in cui il servente è impegnato
- $T_R$  *Tempo di risposta*: tempo medio che un cliente passa nel sistema
- $T_W$  *Tempo in coda*: tempo medio che un cliente passa in coda
- $L$  *Lunghezza della coda*: numero medio di clienti nel sistema

# Esempio: ufficio postale

## Esempio

- *Nell'ufficio postale i clienti arrivano in media al ritmo di un cliente ogni 120 secondi:*
  - $T_a = 120 \text{ s}$ ,  $\lambda = 1 / T_a = 0,0066 \text{ clienti/s}$
- *C'è un unico impiegato e il servizio di un cliente richiede in media 80 secondi:*
  - $T_s = 80 \text{ s}$ ,  $\mu = 1 / T_s = 0,0125 \text{ clienti/s}$
- *Utilizzazione della risorsa (l'impiegato):*
  - $\rho = \lambda / \mu = T_s / T_a = 0,533 = 53\%$

*Cosa possiamo dire sul tempo di risposta e sull'attesa in coda?*

# La legge di Little

- La legge di Little è un legame, che vale in condizioni molto generali, fra le tre grandezze:
  - $\lambda = 1 / T_a$  : *Frequenza di arrivo*
  - $T_R$  : *Tempo medio di risposta*
  - $L$  : *Lunghezza media della coda*

$$\lambda T_R = L$$

- Osservando due delle grandezze è possibile dedurre la terza
- Non richiede assunzioni sulle modalità degli arrivi ed ei servizi
- Si assume però che il sistema sia in una *condizione stabile*

# Esempio: ufficio postale

## Esempio

- *Arrivando all'ufficio postale noto che in media arriva un cliente ogni 100 secondi:*
  - $T_a = 100$  s,  $\lambda = 1 / T_a = 0,01$  clienti/s
- *Sbirciando all'interno vedo che ci sono 20 clienti:*
  - $L = 20$
- *Quanto tempo resterò nell'ufficio postale?*
- *In base alla legge di Little:*

$$T_R = L / \lambda = 20 / 0,01 = 2000 \text{ s}$$

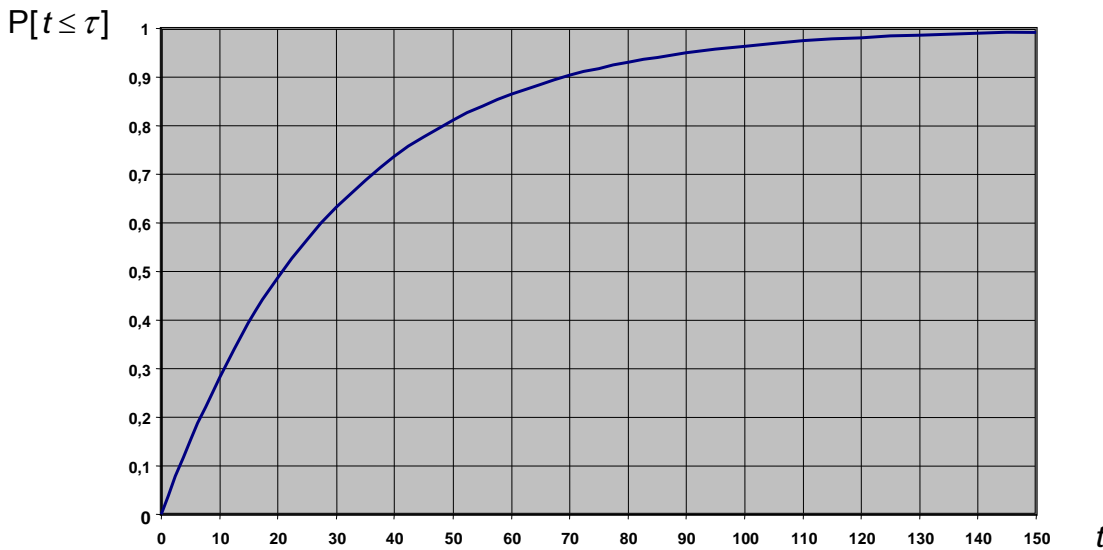
- *Devo prepararmi ad aspettare oltre mezz'ora .....*

## Modalità di arrivo e di servizio

- Per fare affermazioni più precise su cosa succede in una coda occorre precisare non solo i *valori medi*, ma *come sono distribuiti* i tempi di arrivo e di servizio
  - Distribuzione dei tempi di interarrivo (intorno alla media  $T_a$ )
  - Distribuzione dei tempi di servizio (intorno alla media  $T_s$ )
  - Discipline di servizio (scheduling dei clienti)
- Per alcune distribuzioni è possibile dare soluzioni in forma chiusa, cioè calcolare tramite formula le misure che interessano a partire dai parametri della coda
- Il caso più interessante è quello della *distribuzione esponenziale*

**N.B.** *Benché la distribuzione esponenziale non sempre rappresenti fedelmente la realtà, i relativi modelli si dimostrano notevolmente 'robusti', cioè forniscono risultati molto attendibili*

# Distribuzione esponenziale



$$E[t] = \tau = 30 \text{ s}$$

$$\text{Var}[t] = \tau^2$$

- È una tipica distribuzione per variabili continue (es. tempi)
- Si riscontra in moltissimi casi della vita reale (ad es. tempo tra il passaggio di due veicoli sull'autostrada, etc.)

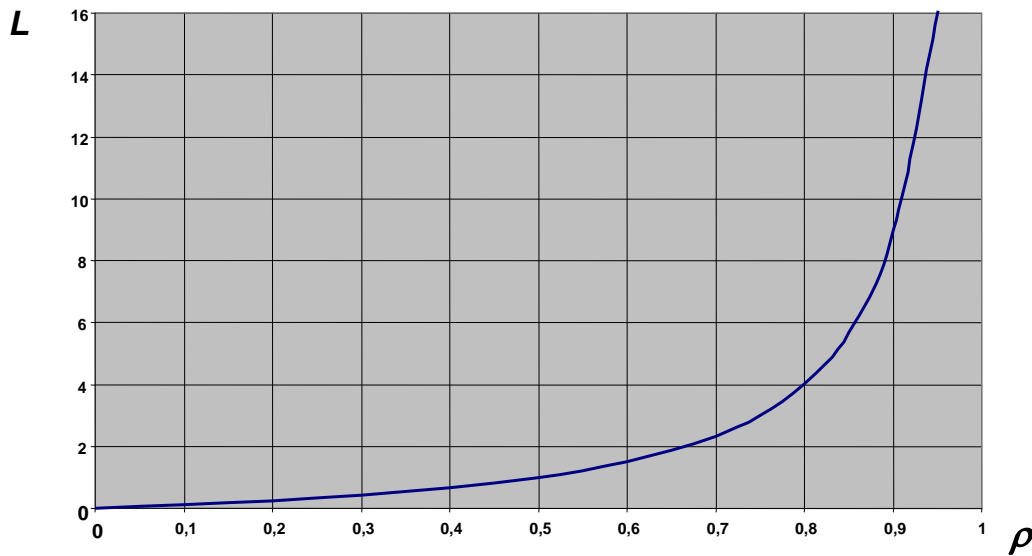
## La coda M/M/1

- In questa coda sia i tempi di arrivo che quelli di servizio hanno distribuzione esponenziale
- C'è un unico servente, che opera con disciplina FCFS
- Si hanno risultati in forma chiusa per la lunghezza media della coda  $L$  e per il tempo medio di risposta  $T_R$

$$T_R = T_s \frac{1}{1 - \rho} \quad L = \frac{\rho}{1 - \rho}$$

**N.B.** Sia la lunghezza media della coda  $L$  che il tempo medio di risposta dipendono  $T_R$  dal livello di congestione  $\rho$ , e per  $\rho = 1$  le formule divergono, perché il sistema non raggiunge l'equilibrio

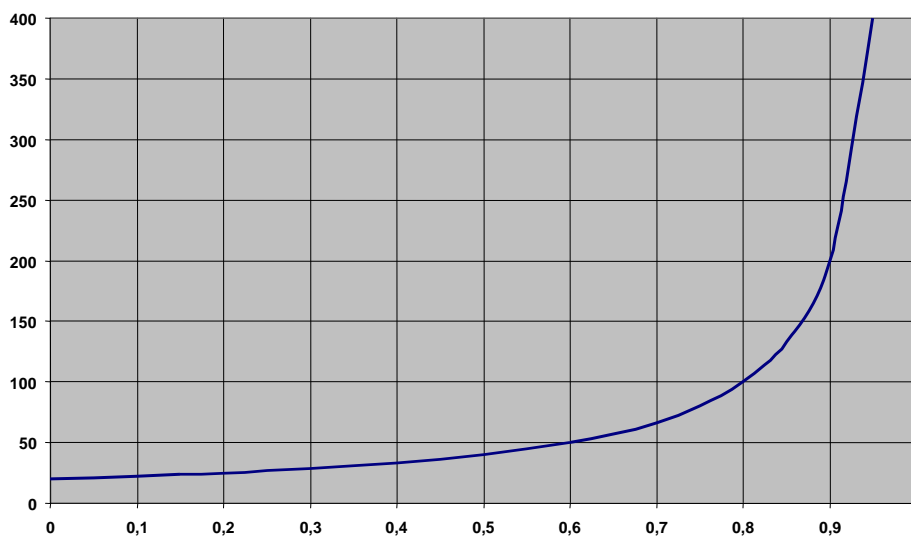
# M/M/1: lunghezza media della coda



$$L = \frac{\rho}{1 - \rho}$$

- La curva ha un asintoto per  $\rho = 1$
- La lunghezza della coda si mantiene moderata poi esplose per alti valori della congestione  $\rho$

# M/M/1: tempo medio di risposta



$$T_R = T_s \frac{1}{1 - \rho}$$

$$T_s = 20 \text{ s}$$

- La curva ha un asintoto per  $\rho = 1$
- Il tempo medio di risposta si mantiene accettabile fino a  $\rho = 0.6$  poi esplose per alti valori della congestione  $\rho$

# Effetto delle distribuzioni

*Un aumento della varianza (cioè del disordine) sia del tempo di arrivo che del tempo di servizio peggiora la situazione*

- Per caratterizzare in modo semplice una distribuzione si usa il *coefficiente di variazione* cioè il rapporto tra varianza e quadrato della media

$$C_v[t] = \text{Var}[t] / (E[t])^2$$

- Distribuzione deterministica ( $\text{Var}[t] = 0$ ):  $C_v = 0$
- Distribuzione esponenziale ( $\text{Var}[t] = \tau^2$ ):  $C_v = 1$

**N.B.** *La distribuzione esponenziale costituisce lo spartiacque tra le distribuzioni a bassa e ad alta varianza*

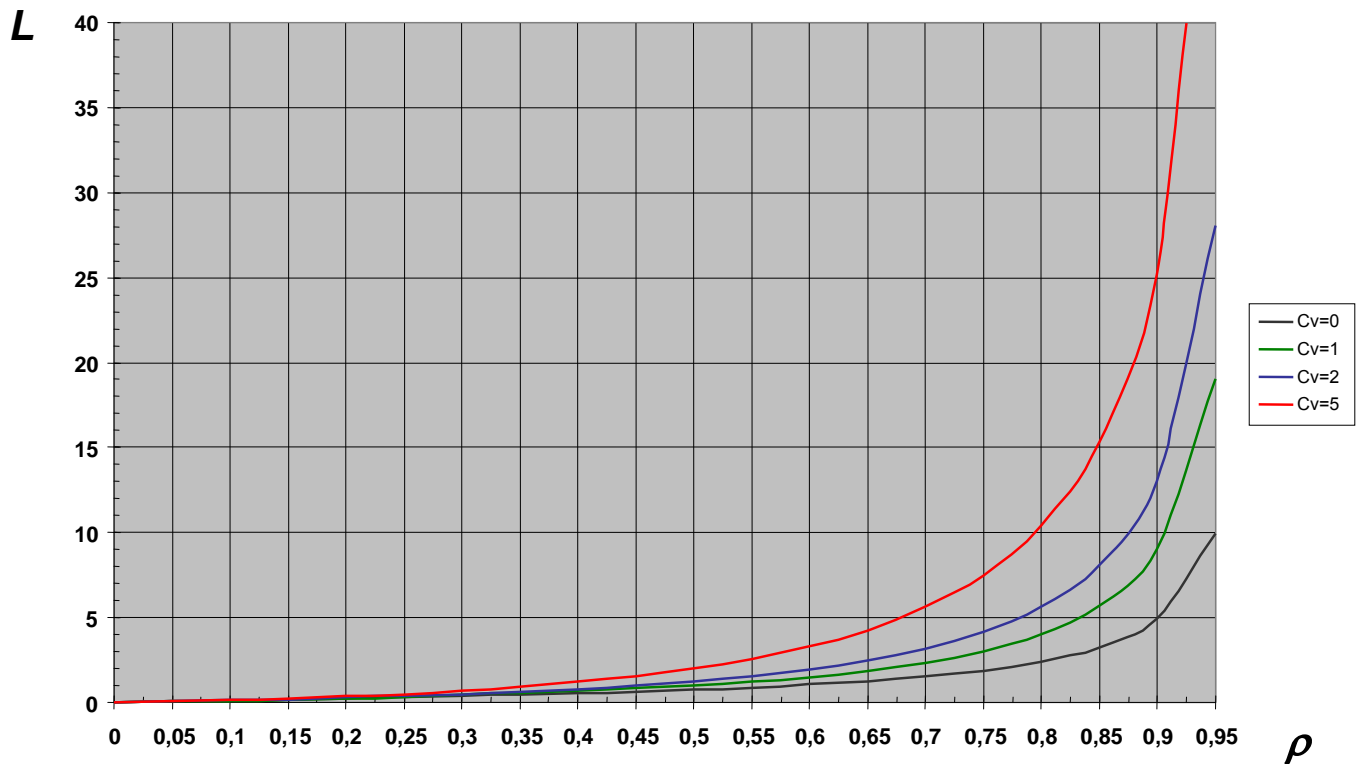
## La coda M/G/1

- La distribuzione degli arrivi è esponenziale, ma quella dei servizi è *generale*, con *coefficiente di variazione*  $C_v$
- Senza altre ipotesi si possono dare formule molto interessanti per la lunghezza media della coda  $L$  e per il tempo medio di risposta  $T_R$

$$T_R = T_s \frac{1}{1 - \rho} \left[ 1 - \frac{\rho}{2} (1 - C_v) \right]$$
$$L = \frac{\rho}{1 - \rho} \left[ 1 - \frac{\rho}{2} (1 - C_v) \right]$$

**N.B.** *Sono le stesse formule della coda M/M/1, con un fattore correttivo (evidenziato in rosso), che cresce al crescere di  $C_v$*

# M/G/1: lunghezza media della coda

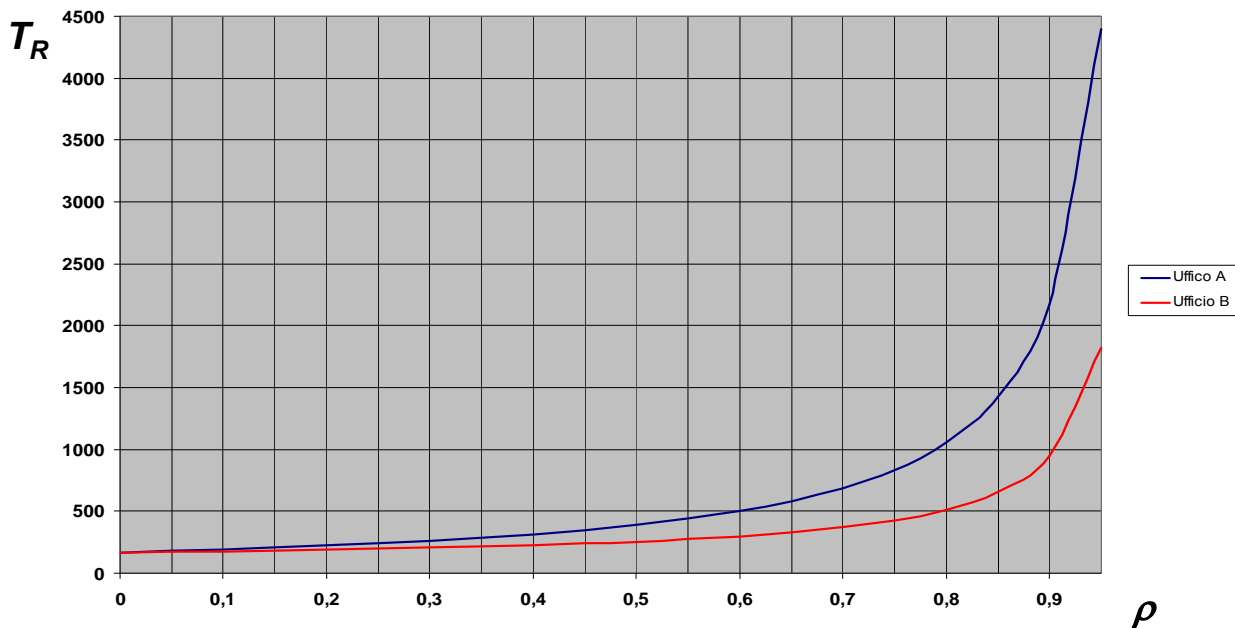


## Esempio: ufficio postale

### Esempio

- Negli uffici postali il servizio di un bollettino richiede 60 s
- Nell'ufficio **A** la distribuzione delle richieste di servizio è:
  - 1 bollettino con probabilità 80%
  - 10 bollettini con probabilità 20%
- Nell'ufficio **B** la distribuzione delle richieste di servizio è:
  - 1 bollettino con probabilità 5%
  - 2 bollettini con probabilità 10%
  - 3 bollettini con probabilità 85%
- In entrambi gli uffici il tempo di servizio medio è di 168 s
- Ma i coefficienti di variazione sono sensibilmente diversi:  
Ufficio **A**:  $C_v = 1.65$       Ufficio **B**:  $C_v = 0.033$

# Esempio: tempi di risposta



Con l'impiegato impegnato al 90% nell'ufficio **A** si attende in media oltre 36 minuti, mentre nell'ufficio **B** meno di 16

## Reti di code

- I risultati sulle code isolate possono si estendono alle reti di code
- Il sistema di elaborazione è rappresentato come un insieme di risorse (CPU, dischi, linee, etc.), a ciascuna delle quali nel modello corrisponde una coda
- I clienti sono i processi, oppure le unità di lavoro: ad esempio le transazioni in un sistema bancario
- Ciascun 'cliente' richiede una percentuale d'uso da ciascuna risorsa: ad es. X secondi di CPU e Y accessi a disco
- In base alla *profilo del carico* ed alla *configurazione del sistema* è possibile determinare:
  - L'utilizzazione delle risorse
  - La lunghezza delle code che si creano alle varie risorse
  - I tempi di attesa, ed il tempo di risposta complessivo

# Esempio: profilo del carico

## Esempio

- Un sistema bancario gestisce transazioni
- L'intensità del carico  $\lambda$  è espressa in transazioni/s
- Ci sono 4 tipi diversi di transazioni
- Per ciascun tipo si determina:
  - La richiesta complessiva di CPU (riferito ad una CPU campione)
  - Gli accessi a ciascuno dei dischi (dipendono dalla allocazione dei file)
- Si stima la frequenza relativa di ciascun tipo di transazione
- Si calcolano le richieste medie a ciascuna risorsa come somma pesata

# Esempio: caratterizzazione del carico

Tipo di transazione	Frequenza	CPU 500 MHz	Disk 1	Disk 2	Disk 3	Disk 4
T1	0,1	2	20	40	3	1
T2	0,3	1,5	0	10	5	0
T3	0,2	1,2	10	30	0	2
T4	0,4	5	20	10	5	2
Media		2,89	12	17	3,8	1,3

- Le richieste comprendono l'overhead del sistema operativo
- La richiesta di CPU è riferita ad una CPU campione
- Per i dischi si parla di accessi, per passare ai tempi occorre specificare le caratteristiche dei dispositivi
- Si assume un trasferimento di 8 KB per accesso a disco

# Esempio: configurazione del sistema

## Esempio

- Si prevede di adottare la seguente configurazione:
  - 4 CPU da 2 GHz
  - Disk 1:  $t_{acc} = 4.8 \text{ ms}$  ,  $t_{trasf} = 8 \text{ ms}$
  - Disk 2:  $t_{acc} = 4.8 \text{ ms}$  ,  $t_{trasf} = 8 \text{ ms}$
  - Disk 3:  $t_{acc} = 5,2 \text{ ms}$  ,  $t_{trasf} = 10 \text{ ms}$
  - Disk 4:  $t_{acc} = 5,2 \text{ ms}$  ,  $t_{trasf} = 10 \text{ ms}$
- Il tempo di accesso ai dischi fa riferimento a tempi di seek medi, sommati al latency, che dipende dal dispositivo
- I tempi di trasferimenti dipendono dalla velocità di rotazione e dalla densità di registrazione

## Analisi dei colli di bottiglia

- Dal profilo del carico e dalla configurazione si possono calcolare le *utilizzazioni relative* dei dispositivi
- Queste sono le utilizzazioni che essi hanno per ciascuna transazione tipo
- Le utilizzazioni relative mantengono i loro rapporti al crescere dell'intensità del carico

- La risorsa che ha la più alta utilizzazione relativa è detta *collo di bottiglia*
- Quando l'intensità del carico sale, essa satura per prima
- Il collo di bottiglia condiziona le prestazioni di tutto il sistema
- L'individuazione dei colli di bottiglia consente di apportare al sistema le necessarie azioni correttive

# Analisi dei colli di bottiglia: esempio

Intensità	UTILIZZAZIONI				
	4 CPU 2 GHz	Disk 1	Disk 2	Disk 3	Disk 4
10	0,03	0,03	0,04	0,01	0,00
50	0,15	0,13	0,18	0,05	0,02
100	0,30	0,26	0,36	0,10	0,03
200	0,60	0,51	0,73	0,19	0,07
300	0,90	0,77	1,09	0,29	0,10

- *Il collo di bottiglia è chiaramente Disk 2, seguito da Disk 1*
- *Molto scarichi sono Disk 3 e Disk 4*
- *Le 4 CPU a 2 GHz reggono bene fino a 300 transazioni/min*
- *Si dovrebbero riallocare i file*

## Calcolo del tempo di risposta

- Una volta calcolate le utilizzazioni per una data intensità di carico, è possibile stimare i tempi di risposta
- Il tempo di risposta di ciascun tipo di transazione non è altro che la somma dei tempi che la transazione spende nelle code corrispondenti alle varie risorse
- Se non ci sono distribuzioni strane dei tempi di servizio, le code possono essere trattate come code M/M/1, e risolte utilizzando le relative formule
- Per ogni 'visita' che il cliente (la transazione) fa alla coda, egli accumula il relativo tempo tempo di risposta
- Ad esempio la transazione  $T_3$  fa 30 visite al disco 2, e 'paga' altrettante volte il tempo di risposta della relativa coda

# Tempi di risposta: esempio

- Con 100 transazioni/min i tempi di risposta alle code sono:

Intensità	TEMPI DI RISPOSTA ALLE CODE (secondi)				
Transazioni/min	4 CPU 2 GHz	Disk 1	Disk 2	Disk 3	Disk 4
100,000	0,258	0,017	0,020	0,017	0,016

- Calcolando il numero di visite che  $T_3$  fa a ciascun disco:

TEMPI COMPLESSIVI SPESI DA T3 ALLE CODE					
CPU	Disk 1	Disk 2	Disk 3	Disk 4	Totale
0,258	0,172	0,603	0,000	0,031	1,064

## Tuning dei sistemi

- Sintomo evidente del problema sono in generale *tempi di risposta insoddisfacenti*
- Occorre però *individuare le cause*, prima di poter intervenire correttamente
- Un primo passo può essere l'*analisi delle utilizzazioni*
- Si individuano i *colli di bottiglia* e si progettano *le azioni correttive*

### Tipologia delle azioni correttive

- *Ridistribuire il carico per livellare le utilizzazioni*
- *Modificare le modalità di servizio*
- *Potenziare le risorse, con azioni dirette e indirette*

# Congestione della CPU

---

- Non è circostanza comune, anche se molti interventi ‘alla cieca’ consistono in un potenziamento della CPU
- Occorre quindi analizzare con cura se il problema è proprio questo, e individuare i processi ‘consumatori’

## *Possibili azioni:*

- Potenziamento della CPU: aumentare la frequenza può essere costoso, considerare anche eventuali opzioni sulle cache L2 e L3
- Piattaforma multi-CPU: la potenza di calcolo scala bene a causa dell’alto numero di processi e di thread
- Load balancing: i sistemi operativi ed alcuni ambienti applicativi gestiscono il load balancing

# Congestione dei dischi (1)

---

- Una prima linea di intervento non richiede alcun investimento in nuovi acquisti, ma una semplice riorganizzazione
- Le azioni interessano sia i singoli dischi che il loro insieme

## *Possibili azioni:*

- Deframmentazione: la frammentazione causa aumenti del numero di accessi e del tempo medio di seek
- Riallocazione dei file: conviene allocare i file più acceduti in cilindri contigui per minimizzare i tempi di seek
- Distribuzione del carico: discende dal raffronto delle utilizzazioni dei diversi dischi: si possono bilanciare spostando i file sui dischi con minore utilizzazione

## Congestione dei dischi (2)

---

- Se le azioni di riallocazione non sono sufficienti occorre potenziare il sottosistema di I/O
- Due tipi di azioni possibili:
  - Sostituire i dischi con altri di migliori caratteristiche: ha senso solo se sono veramente obsoleti
  - Aumentare il numero di dispositivi, ridistribuendo fra loro i file in modo bilanciato
- Considerare anche il possibile ricorso a dischi RAID

Attenzione a non creare altri colli di bottiglia

- I bus di I/O possono diventare colli di bottiglia
- Attenzione a posizionare i dischi sui bus IDE
- I dischi con connessione seriale punto-punto scalano in maniera molto efficace

## Ampliamento della memoria (3)

---

- Problemi sui dischi possono essere risolti anche con interventi sulla memoria
- Memoria più ampia significa anche buffer cache più ampia
- Aumenta la probabilità che un accesso a disco trovi i dati nella buffer cache
- Diminuiscono quindi, anche in misura molto rilevante, gli accessi ai dischi
- I DBMS gestiscono anche una loro cache interna: in genere conviene cedere memoria ai DBMS

*Quello in memoria centrale è quasi sempre il migliore investimento che possiamo fare per migliorare le prestazioni di un sistema di elaborazione*

# Benchmark

---

*Un benchmark è un campione di carico, costruito allo scopo di misurare le prestazioni di un sistema mentre lo elabora*

- Un *benchmark* è utile soprattutto perché consente di confrontare le prestazioni di sistemi diversi
- I benchmark, per la loro stessa natura sono rappresentativi di specifici contesti applicativi, dei quali simulano situazioni di carico tipiche

## **Esempio**

Sono molti diffusi benchmark per il confronto di computer desktop per applicazioni home e di automazione di ufficio: BapCo, Ziff Davies, Winstone....

# BAPCO SYSmark 2004 SE

---

- Per valutare piattaforme desktop di Personal Computing
- Realizzati da BAPCO, consorzio non-profit cui partecipano tutti i principali costruttori (Intel, AMD, Microsoft, HP, Toshiba etc.).  
<http://www.bapco.com>
- Previsti due diversi profili di utenza, con gruppi di operazioni:
  - **Internet Content Creation**: 3D creation, 2D creation, Web publication
  - **Office Productivity**: Communication, Document creation, Data analysis
- Simulato l'alternarsi dell'utente tra diverse *applicazioni reali*
- Indice di prestazione: *tempo complessivo per eseguire una serie predefinita di operazioni* (rispetto ad un sistema base predefinito)
- Prestazioni a livello di profilo e di gruppo

# BAPCO MobileMark 2005

---

- Per sistemi laptop
- Misura sia le prestazioni che la durata della batteria
- Previsti diversi profili di utenza:
  - **DVD playback**
  - **Wireless browsing**
  - **Office productivity**
  - **Reader**
- Simulato l'alternarsi dell'utente tra diverse *applicazioni reali*
- Indice di prestazione: *tempo complessivo a disposizione per eseguire un determinato mix di operazioni*
- Prestazioni a livello di profilo e indice pesato

# BAPCO WebMark 2004

---

- Misura le prestazioni di piattaforme client per applicazioni Internet
- Diverse tipologie di accessi per cui sono simulati siti campione
- **Information processing**
  - **Portal**: portale aziendale, entry point ai servizi per i dipendenti
  - **Reserch**: attività di navigazione alla ricerca di informazioni
  - **Training**: attività di e-learning e training online
- **Commercial transactions**
  - **Purchasing**: acquisto di prodotti da siti di e-commerce
  - **Finance**: servizi informativi e di intermediazione finanziaria
  - **Marketplace**: intermediazione commerciale
- Indice di prestazione: *tempo di risposta per transazioni elementari definite nei vari contesti*
- Prestazioni a livello di contesto e indice pesato

# Benchmark per server

---

- Misurano le prestazioni di sistemi server nei principali contesti applicativi (principalmente Web e DataBase)
- Il sistema comprende i vari strati di un'architettura a più livelli:
  - Piattaforma Hardware
  - Sistema operativo
  - Web e/o DataBase server
  - Applicazione
- Richiedono esperimenti complessi e molto costosi
  - Allestimento di sistemi complessi e costosi
  - Predisposizione di sistemi per la generazione del carico
- **TPC** (*Transaction Processing Performance Council*)
- **SPEC** (*Standard Performance Evaluation Corporation*)

## Principali benchmark TPC

---

- **TPC** (*Transaction Processing Performance Council*), organizzazione indipendente in cui sono rappresentati i principali costruttori  
<http://www.tpc.org>
- Benchmark per vari tipi di contesti applicativi
- **TPC-C**: simula un contesto applicativo dove i clienti eseguono transazioni predefinite su di un base di dati.
- **TPC-H**: simula un contesto applicativo di un sistema di supporto alle decisioni, secondo standard industriali consolidati
- **TPC-R**: come TPC-H ma con una consistente componente di transazioni predefinite che possono essere ottimizzate
- **TPC-W**: simula un contesto applicativo di un sistema Web per l'elaborazione di transazioni, è lo standard industriale per le applicazioni di e-business

# Il benchmark TPC-C

---

- Specifica in dettaglio la struttura di un tipico database aziendale (magazzini, ordini, clienti, etc.)
- Specifica i parametri estensionali del database (n. di clienti,...)
- Specifica le modalità di popolazione del database
- Definisce cinque tipi di transazioni:
  - New-Order
  - Payment
  - Order-Status
  - Delivery
  - Stock-Level
- Specifica le regole con cui devono essere svolti gli esperimenti
- Specifica le regole di calcolo del costo dei sistemi

La metrica è tpmC: numero di transazioni New-Order al minuto

# Benchmark SPEC

---

- **SPEC** (*Standard Performance Evaluation Corporation*), organizzazione indipendente in cui sono rappresentati i principali costruttori  
<http://www.tpc.org>
- Benchmark per vari tipi di contesti applicativi
- **CPU**: simula carichi di tipo CPU bound.
- **Graphics/Application**: per applicazioni grafiche pesanti
- **HPC/OMP**: per piattaforme parallele di High Performance Computing
- **Java Client/Server**: per contesti applicativi Java2EE
- **Mail Server**: per server di posta elettronica
- **Network File System**: per file server
- **Web Server**: per web server

# Affidabilità

---

- L'affidabilità costituisce un aspetto importante nella caratterizzazione della qualità dei sistemi
- L'affidabilità si misura quantitativamente mediante lo MTBF

*MTBF (Mean Time Between Failures) rappresenta il tempo medio che intercorre tra due guasti consecutivi del sistema*

- L'affidabilità delle componenti del sistema (es. dischi, CPU, linee etc.) è a sua volta caratterizzata tramite il MTBF
- È possibile calcolare l'affidabilità di sistemi complessi a partire da quella delle loro componenti

## MTBF (Mean Time Between Failures)

---

- Rappresenta l'intervallo di tempo medio che intercorre tra il verificarsi di due guasti consecutivi

**ES**

$$\text{MTBF} = 500 \text{ Khr} = 500.000 / (24 \cdot 365) \approx 57 \text{ anni}$$

Ciò non significa che, considerato un utilizzo medio di *soli* 5 anni, uno possa stare tranquillo!

- Per riportare quest'informazione in termini più facilmente percettibili occorre fare anche un'ipotesi sulla distribuzione temporale dei guasti
- L'ipotesi più verosimile è quella di una distribuzione esponenziale, cioè che i guasti siano generati da un processo di Poisson
- Del resto l'esperienza insegna che questa distribuzione è riscontrata molto spesso processi di questo tipo (arrivi dei clienti ad una coda etc.)

# Affidabilità dei dischi

---

- MTBF (*Mean Time Between Failures*)  $\tau$
- Distribuzione *esponenziale* della probabilità di guasto
- Probabilità che un guasto si verifichi in periodo lungo  $T$ :

$$P_1[0, T] = 1 - e^{-T/\tau}$$

- Se si hanno  $n$  dischi, e si considera che i loro guasti siano indipendenti lo MTBF diminuisce di un fattore  $n$ :  $\tau_n = \tau / n$
- Probabilità che si guasti uno qualsiasi degli  $n$  dischi in un periodo lungo  $T$ :

$$P_n[0, T] = 1 - e^{-nT/\tau}$$

## Affidabilità dei dischi: esempio

---

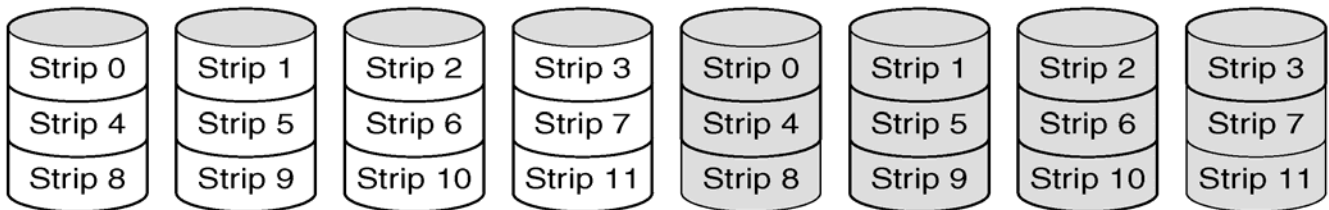
### ES

- *In un sistema ci sono 100 dischi, il loro ciclo di vita è di 5 anni*
- *Il tipo di dischi utilizzato ha un un MTBF di  $10^6$  ore, cioè circa 100 anni.*
- *La probabilità che un dato disco si guasti in cinque anni, cioè durante la sua vita operativa, ( $T/\tau = 0.05$ ) è circa  $5 \cdot 10^{-2}$ .*
- *La probabilità che un qualsiasi disco dei 100 si guasti durante i 5 anni ( $T/\tau = 5$ ) è oltre il 99%, cioè quasi una certezza.*

Occorre assolutamente costruire sottosistemi di dischi che migliorino l'affidabilità sfruttando la ridondanza

# RAID Level 1

---



- Ciascun disco è duplicato da un altro disco 'ombra': *shadowing*
- Ogni scrittura interessa sia un disco che al sua ombra
- Ottime prestazioni soprattutto in lettura: molte possibilità di bilanciare il carico
- Supportato anche da vari Sistemi Operativi (Es. Windows 2000 e Windows XP)

Pro: eccellente resistenza ai guasti

Contro: occorre aumentare lo spazio su disco del 100%

# Affidabilità di RAID 1

---

## ES

- *Un sottosistema RAID 1 ha 100+100 dischi*
- *Il tipo di dischi utilizzato ha un un MTBF di  $10^6$  ore, cioè circa 100 anni.*
- *Si suppone che un disco guasto sia sostituito entro 5 ore*
- *La perdita dei dati si ha solo se il suo disco ombra si guasta nelle 5 ore*
- *La probabilità che nella vita operativa (5 anni) ci siano guasti è altissima (oltre il 99%)*
- *La probabilità che durante un guasto il secondo disco si guasti nelle 5 ore necessarie a sostituire il primo è di  $5 \cdot 10^{-6}$*