

Metodologia

Andrea Marin

Università Ca' Foscari Venezia
Laurea in Informatica
Corso di Simulazione e Perf. delle Reti

a.a. 2013/2014

Licenza

These slides are adapted from those of J.Y. Le Boudec designed for the course of Performance evaluation at the EPFL. You may use these slides under the Attribution-Sharealike license.



Di cosa si occupa la valutazione delle prestazioni?

Metriche, carico e obiettivi

Fattori nascosti

Il metodo scientifico

Patterns



Di cosa si occupa la valutazione delle prestazioni?

Andrea Marin

Corso di simulazione e performance delle reti



Cos'è la performance evaluation? /1

Esempio

In [1] i S.O. Windows e Linux vengono confrontati. Come metriche sono usate: numero di cicli di CPU, numero di istruzioni, numero di operazioni di lettura/scrittura richiesta da un'applicazione tipica. Il carico di lavoro è stato generato da diversi *benchmarks*: **syscall** genera operazioni elementari; **memory read** genera riferimenti ad un array; e un application benchmark esegue un'applicazione diffusa.



Cos'è la performance evaluation? /2

Esempio

L'energia elettrica che consuma un computer o un dispositivo di comunicazione dipende da quanto efficientemente il dispositivo riesce a trarre vantaggio dai periodi di scarsa attività per risparmiare energia. Un operatore propone la seguente metrica come misura del consumo energetico [3]:

$$P_{\text{Total}} = 0.35P_{\text{max}} + 0.4P_{50} + 0.25P_{\text{sleep}}$$

con P_{max} la potenza consumata a massimo carico, P_{50} quando il carico di lavoro è al 50% e P_{sleep} la potenza consumata nei periodi di inattività.

Metriche, carico e obiettivi

Andrea Marin

Corso di simulazione e performance delle reti

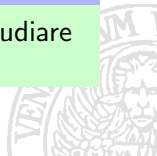


Carico di lavoro (load)

- ▶ Bisogna definire il carico di lavoro al quale è soggetto il sistema in studio
- ▶ Fare la differenza tra:
 - ▶ Intensità del carico (es. jobs al secondo)
 - ▶ Tipologia del carico
 - ▶ Ci sono alcuni dettagli statistici che possono essere importanti

Definizione

I **benchmark** sono generatori di traffico artificiali usati per studiare le reazioni di un sistema o un modello

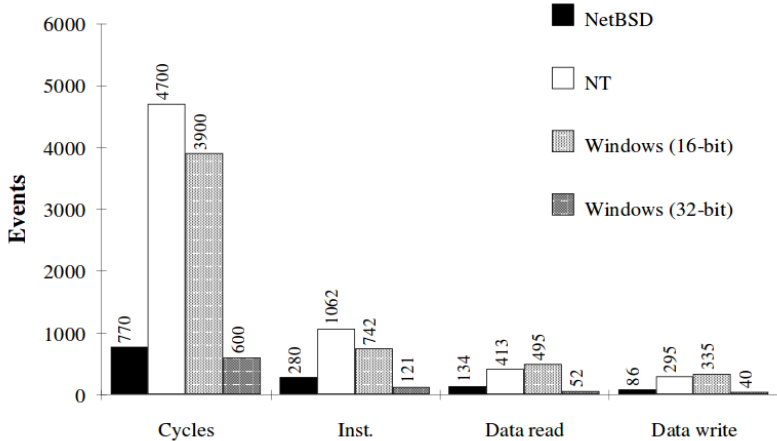


Metriche

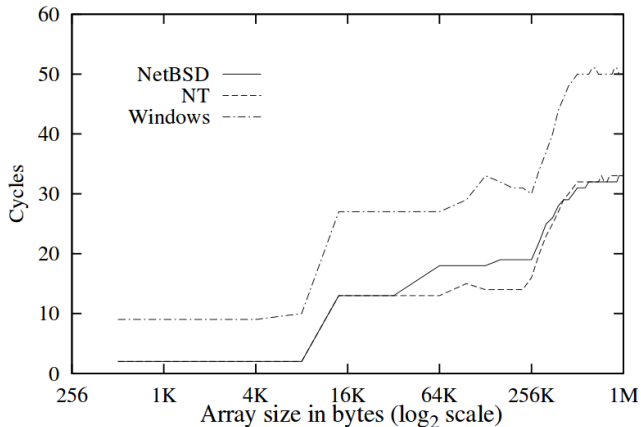
- ▶ È importante definire delle metriche
 - ▶ Tempi di risposta (medi/distribuzione)
 - ▶ Consumo energetico
 - ▶ Throughput
- ▶ Definire le condizioni di lavoro sotto le quali le metriche sono usate



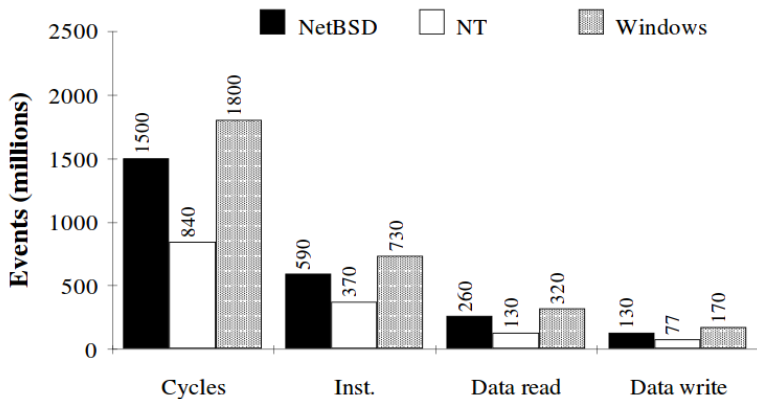
Syscall



Memory read



Ghostscript



Metriche multi-dimensionali

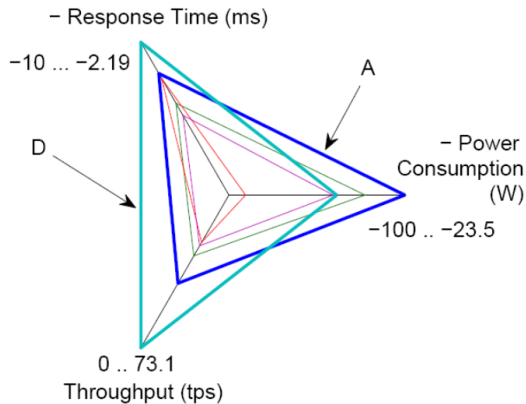
Esempio

Misuriamo le performance di un web server sottoposto al carico di un workbench standard. Confrontiamo 5 configurazioni ottenendo i seguenti risultati:

Config	Power (W)	Response (ms)	Throughput (tps)
A	23.5	3.78	42.2
B	40.8	5.30	29.1
C	92.7	4.03	22.6
D	53.1	2.19	73.1
E	54.7	5.92	24.3

Diagramma di Kiviat

- A e D non sono dominati



Conosci i tuoi obiettivi

Esempio

1. La configurazione del PC 1 è del 25% più veloce della configurazione 2 quando viene eseguito Excel
2. Per l'applicazione di video on demand, il numero di server richiesto è 35, e il numero di dischi è 68
3. Usando la nuova versione di `sendfile()` aumenta il throughput del server del 51%
4. Per l'applicazione di video on demand il numero di server richiesti è $N_1 = \lceil \frac{R}{59.3} + \frac{B}{3.6} \rceil$ e il numero di dischi è $N_2 = \lceil \frac{R}{19.0} + \frac{B}{2.4} \rceil$ dove R (B) è il numero di utenti residenziali (business).

Cosa ricavare?

A seconda delle esigenze dello studio che si vuole condurre si può ricavare. . .

- ▶ 1, 3 sono confronti
- ▶ 2 è uno statement assoluto
- ▶ 4 è una legge ingegneristica



Fattori nascosti

Andrea Marin

Corso di simulazione e performance delle reti



Fattori

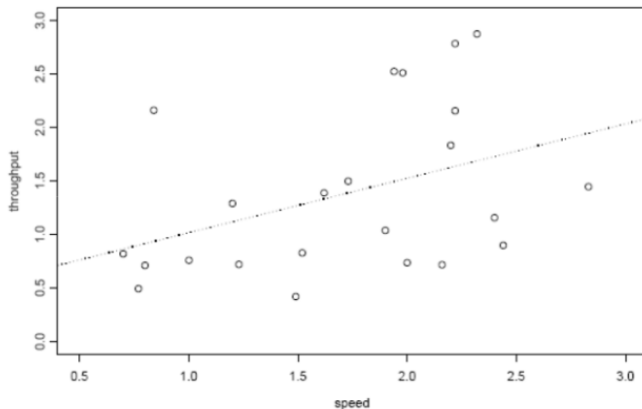
Definizione

Si chiama **fattore** un qualsiasi elemento che possa avere un impatto sulle performance. Si distinguono:

- ▶ Fattori desiderati: intensità del carico, numero di servers
 - ▶ Fattori di disturbo: ora del giorno, presenza di attacchi *denial of service*
-
- ▶ Bisogna porre estrema attenzione ai fattori da considerare come illustra il prossimo esempio



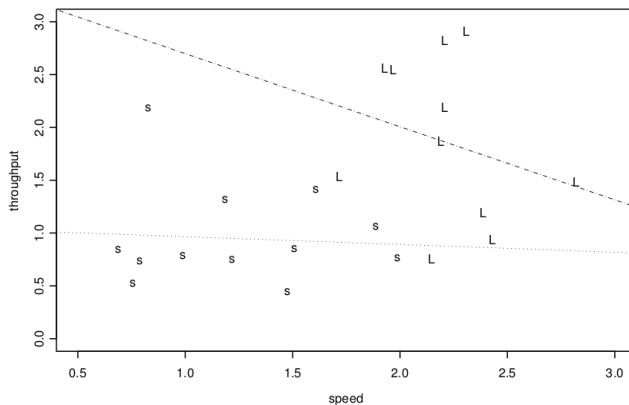
TCP su nodi mobili, throughput vs. velocità dei nodi



Il grafico suggerisce che il throughput aumenta con l'aumentare della velocità dei nodi!

Fattore nascosto

Stesso grafico di prima, *s* sono i nodi con un buffer di 4KB e *L* i nodi con un buffer di 16KB. Conclusioni invertite.



Cosa ci ha depistato?

- ▶ Il fattore nascosto aveva più importanza del fattore sul quale focalizzavamo la nostra attenzione
- ▶ Abbiamo interpretato la **correlazione** come **causalità**
- ▶ Bisogna essere consapevoli di tutti i fattori ed incorporarli nell'analisi
- ▶ In alternativa si deve randomizzare gli esperimenti per ridurre l'impatto dell'hidden factor



Il paradosso si Simpson

Siano i nodi lenti con velocità $\leq 2m/s$ e veloci $> 2m/s$. Un nodo è in condizione di successo se il suo throughput è $\geq 1.5Mb/s$

	failure	success		$\mathbb{P}(\text{success})$
slow	11	3	14	0.214
fast	5	4	9	0.444
	16	7	23	

Dalla tabella concludiamo che i nodi veloci hanno maggiore probabilità di successo.



Con il fattore nascosto

S mobile nodes	failure	success		$\mathbb{P}(\text{success})$
slow	10	1	11	0.091
fast	1	0	1	0.00
	11	1	12	

L mobile nodes	failure	success		$\mathbb{P}(\text{success})$
slow	1	2	3	0.667
fast	4	4	8	0.500
	5	6	11	



Scopri il fattore nascosto (Berkeley 73)/1

Nel 1973 l'Università di Berkeley fu accusata di avere un trattamento discriminatori nei confronti delle donne dati i seguenti dati di ammissione:

	Applicants	Admitted
Men	8442	44%
Women	4321	35%

Tuttavia i dati scorporati per corso di laurea non mostrano discrepanze significative tra le percentuali di accettazione maschile e femminile



Scopri il fattore nascosto (Berkeley 73)/2

Major	Men		Women	
A	825	62%	108	82%
B	560	63%	25	68%
C	325	37%	593	34%
D	417	33%	375	35%
E	191	28%	393	24%
F	373	6%	341	7%

Cosa si può concludere dalla ricerca?



Il metodo scientifico

Andrea Marin

Corso di simulazione e performance delle reti



Passi del metodo scientifico

1. Formula un'ipotesi sul sistema

- ▶ Es. il server è lento a causa del processore
- ▶ Ptogetta degli esperimenti che tentino di *invalidare* l'ipotesi

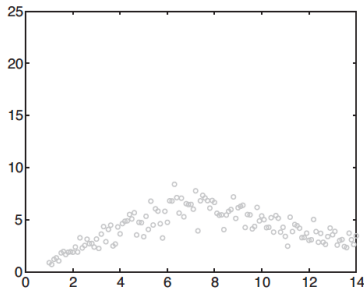
Porre attenzione ai seguenti aspetti:

- ▶ Specificare l'accuratezza dei risultati
 - ▶ Es. Intervalli di confidenza
- ▶ I risultati devono essere riproducibili
- ▶ Rimuovi tutto quello che può essere rimosso



Esempio

Joe è ha realizzanto un punto di servizio dove fornisce video online ai clienti con smartphone. Il sistema è formato da una stazione wi-fi 802.11 e un server. Prima dello sviluppo Joe conduce una simulazione per stimare il massimo throughput del sistema ottenendo il seguente grafico



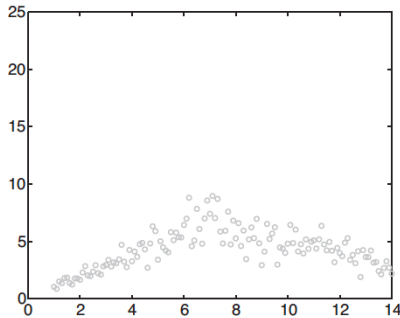
(a) Initially

Sulle ordinate il throughput sulle ascisse transazioni al secondo

Formulo un'ipotesi

- Hyp1: Il collo di bottiglia è l'access point

Quindi Joe progetta un esperimento aggiungendo altri due access points ottenendo il seguente grafico:

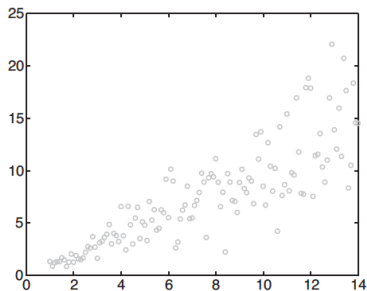


Ipotesi invalidata

- ▶ Il nuovo esperimento suggerisce che l'ipotesi Hyp1 era sbagliata. Infatti ulteriori misure mostrano che il numero di collisioni nella rete e l'utilizzo sono molto bassi
- ▶ Joe formula una nuova ipotesi, Hyp2: Il collo di bottiglia (bottleneck) sta nella memoria del server
- ▶ Quindi Joe prova ad invalidare Hyp2



Ipotesi confermata



(c) With additional server memory



Patterns

Andrea Marin

Corso di simulazione e performance delle reti



Pattern ricorrenti nella valutazione delle prestazioni

- ▶ Colli di bottiglia (bottlenecks)
- ▶ Collasso per congestione (congestion collapse)
- ▶ Effetti collaterali della competizione
- ▶ Collasso latente per congestion



Bottlenecks

Definizione

Il collo di bottiglia di un sistema è un componente o un insieme di componenti la cui scarsa performance causa il rallentamento complessivo del sistema

- ▶ L'identificazione dei bottlenecks dipende da tutti i parametri del sistema incluso il carico di lavoro
- ▶ La rimozione di un bottleneck può introdurre un altro



Esempio

Il monitoraggio di un sistema mostra che il costo medio per una transazione è di 1,238,400 istruzioni di CPU, 102.3 accessi al disco e 4 pacchetti inviati in rete. Il processore è in grado di gestire 10^9 istruzioni al secondo, il disco support 10^4 accessi al secondo e la rete può inviare 10^4 pacchetti al secondo. Quante transazioni al secondo il sistema può supportare?

Soluzione.

- ▶ Individuiamo l'utilizzo delle risorse per transazione:
 - ▶ CPU: 0.12%
 - ▶ Disco: 1.02%
 - ▶ Rete: 0.04%
- ▶ Quindi la capacità del sistema è determinata dal disco portando ad una stima di circa 99 transazioni al secondo



Dalla letteratura

In [2] si studia come sviluppare portali web ad alte prestazioni. L'autore osserva che l'ottimizzazione dei portali dal lato server è molto elevata e che i colli di bottiglia si presentano maggiormente sul lato client. Per esempio un collo di bottiglia è il DNS lookup. Eliminando il DNS lookup, emerge un altro collo di bottiglia che è l'interpretazione degli script sul browser. L'eliminazione di questo bottleneck usando script server side causa l'introduzione di un altro bottleneck e così via. 14 possibili cause di bottleneck sono individuate.



Collasso per congestione

Definizione

La congestione si manifesta quando l'intensità del carico eccede la capacità del sistema (determinata dall'analisi dei bottleneck).

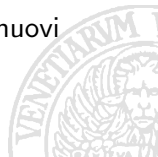
Il controllo della congestione è molto complicato, e l'unica possibilità sta nell'eliminare parte del carico di lavoro.



Cause frequenti della congestion collapse

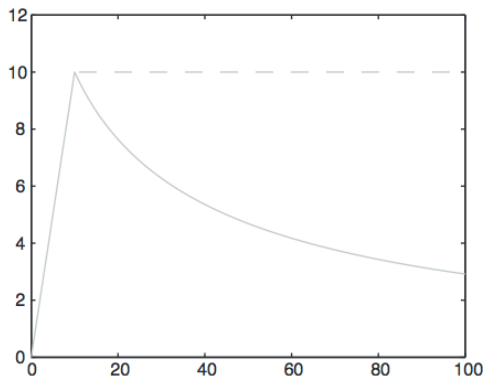
- ▶ Il sistema dedica tempo a svolgere del lavoro che non può essere completato. Ad esempio, questo è causato dall'impazienza degli utenti: quando il tempo di risposta diventa lungo, la richiesta è abbandonata ma il sistema continua a processarla
- ▶ Il tempo di servizio per ciascun job aumenta quando il carico di lavoro aumenta. Un esempio è la paginazione della memoria

L'**admission control** è una tecnica che consiste nel rifiutare nuovi job quando il sistema è il pericolo di congestione



Differenze tra congestion e congestion collapse

Il grafico mostra il throughput di una rete (Mb/s) vs il carico di lavoro (Mb/s). La linea tratteggiata mostra la congestione, la linea continua il collasso per congestione.

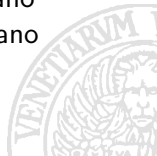


Effetti collaterali della competizione

Definizione

Si chiama effetto collaterale della competizione il peggioramento delle performance del sistema osservato da un certo utente a seguito del potenziamento del sistema

- ▶ Il pattern si verifica quando del potenziamento del sistema si avvantaggia principalmente una classe di utenti. In questo caso può accadere che gli utenti delle altre classi debbano sostenere una competizioni più aggressiva e quindi vedano peggiorare le loro performance



Collasso latente per congestione

Definizione

Il collasso latente per congestione si ha quando a seguito della rimozione di un bottleneck il sistema viene messo a rischio di collasso per congestione

- Essenzialmente il bottleneck che viene rimosso funzionava da admission control



To study

- ▶ Ch. 1 of textbook



References I



J. B. Chen, E. Yasuhiro, and C. Kee.

The mesured performance of personal computer operating systems.
In Proc. of 15th ACM SOSP, pages 169–173, Colorado, USA, 1995.



S. Souders.

High-performance web sites.
Comm. ACM, 51(12):36–41, 2008.



Verizon Corporation.

Verizon NEBS(TM) compliance: Energy efficiency requirements for telecommunications equipment.
Technical Report VZ.TPR9205, 2008.

