

Tesi di Laurea

QoWSbench: un benchmark per sistemi Web a qualità del servizio garantita

Candidato:

Andrea Proietti

Relatore:

Prof. Salvatore Tucci

Correlatore:

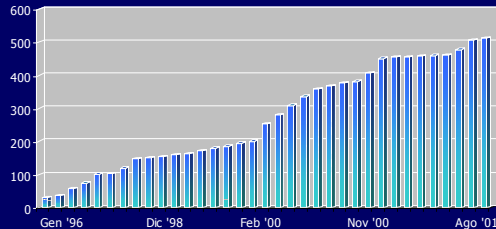
Prof. Michele Colajanni

Sommario

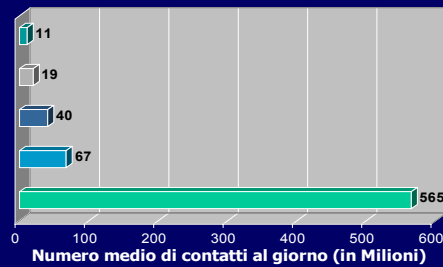
- Crescita di richieste per siti popolari
- Problemi conseguenti
- Soluzioni: QoS e Web server distribuiti
- Mancanza di un benchmark QoWS
- QoWSbench vs WebStone
- Risultati sperimentali
- Conclusioni
- Sviluppi futuri

Crescita del Web / Siti popolari

Numero degli utenti Web negli ultimi 5 anni (in Milioni)



Giochi Olimpici (1996)
NASA Pathfinder (1997)
Giochi Olimpici (1998)
Wimbledon (1999)
Giochi Olimpici (2000)



Problemi conseguenti

- Indipendentemente dal sistema a disposizione, è impossibile soddisfare tutte le richieste in modo ottimale
- ↓
- Necessità di garantire buone prestazioni ad un insieme ristretto di utenti (classificati in base a diversi possibili criteri)
- ↓
- Necessità di valutare le prestazioni sperimentate dalle varie classi di utenti in modo indipendente

- Web server distribuiti
- Qualità del servizio
- Strumento di valutazione

Soluzioni relative

- Utilizzo di Web server distribuiti
- Introduzione della qualità del servizio
- Sviluppo di uno strumento di valutazione delle prestazioni orientato alla qualità del servizio (QoWSbench)

- ⇒ ■ Web server distribuiti
- Qualità del servizio
- Strumento di valutazione

Web server distribuito

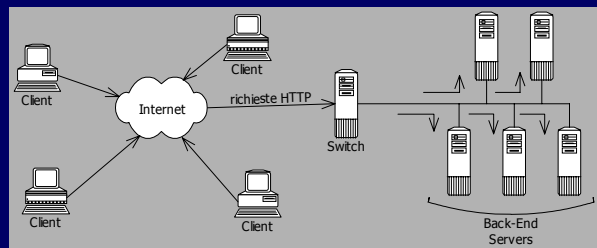
È un server HTTP composto da più macchine collegate fra loro che sia:

- Trasparente rispetto all'utente
- Scalabile

- ➔ ■ Web server distribuiti
- Qualità del servizio
- Strumento di valutazione

Un esempio: Web cluster

- È un Web server distribuito localmente
- Lo switch gestisce lo scheduling delle risorse
- Classificabile rispetto a livello OSI dello switch e architettura 1-Way o 2-Ways



- ➔ ■ Web server distribuiti
- Qualità del servizio
- Strumento di valutazione

Qualità del servizio (QoS)

È l'insieme di tutti i requisiti non funzionali relativi alla condotta di un certo servizio

Nella *QoS differenziata* i requisiti possono cambiare a seconda di alcuni fattori (ad esempio l'ora del giorno o la classe a cui appartiene l'utente che invoca il servizio)

- Web server distribuiti
- ⇒ ■ Qualità del servizio
- Strumento di valutazione

Service Level Agreement (SLA)

È l'accordo, su basi contrattuali, stipulato fra cliente e fornitore di un servizio, inerente requisiti non funzionali dello stesso (ad es. la performance)

- *Guaranteed*: violazioni delle specifiche non tollerate
- *Predictive*: si specifica una soglia massima di violazioni ammissibili (95-percentile)
- *Best effort*: assenza di qualsiasi vincolo contrattuale

- ⇒ ■ Web server distribuiti
- ⇒ ■ Qualità del servizio
- Strumento di valutazione

Politiche QoS in un cluster Web

- Switch Admission
- Static Partition
- Dynamic Partition

- Web server distribuiti
- Qualità del servizio
- ⇒ ■ Strumento di valutazione

Perché misurare il traffico Web?

- Conoscere le abitudini dei clienti e definire strategie di marketing
- Disporre di fonti certe su cui basare un eventuale accordo contrattuale con il fornitore di un servizio
- Valutare le prestazioni (*benchmarking*) di diverse soluzioni proposte dal mercato

- Web server distribuiti
- Qualità del servizio
- ⇒ ■ Strumento di valutazione

Cos'è il benchmarking?

- Generazione, verso un sistema informativo, di un carico di richieste (*workload*) sintetiche
- Raccolta di informazioni sul comportamento del sistema in risposta al workload sintetico
- Elaborazione, sulla base delle informazioni raccolte, di statistiche che siano indicative delle prestazioni del sistema

- Web server distribuiti
- Qualità del servizio
- ■ Strumento di valutazione

Workload characterization

È la creazione di un *workload model*, cioè una serie di parametri significativi che rappresentano un carico in modo analitico

Esempi di parametri: tipo di richieste, tipo di risposte, caratteristiche delle risorse a disposizione, comportamento dell'utente

QoSbench introduce:

- Caratterizzazione del carico
- Qualità del servizio

Cos'è QoSbench?

Un benchmark, ricavato da una serie di modifiche al preesistente software *WebStone*, orientato a:

- **Caratterizzazione del carico:** introduce diversi parametri di modellizzazione del comportamento dell'utente definendo una *sessione utente*
- **Qualità del servizio:** introduce il concetto di classi di utenti, produce in output informazioni suddivise in base alla classe e calcola distribuzione cumulativa e 95-percentile del tempo di risposta

QoWSbench introduce:

- Caratterizzazione del carico
- Qualità del servizio

Richiesta pagine (WebStone)

Una sola iterazione:

- Determinazione stocastica della pagina da richiedere fra quelle nella *filelist*
- Richiesta della pagina al server e memorizzazione dei risultati



QoWSbench introduce:

- Caratterizzazione del carico
- Qualità del servizio

Richiesta pagine (QoWSbench)



- 3 iterazioni: sessioni, pagine, hit
- Parametri determinati stocasticamente secondo distribuzioni

QoWSbench introduce:

- Caratterizzazione del carico
- ⇒ ■ Qualità del servizio

Statistiche (WebStone)

Un report per ogni URL di *filelist* e uno complessivo:

- Valori medi, minimi, massimi e deviazioni standard di tempi di risposta e tempi di connessione
- Quantità di dati trasmessi, con e senza header HTTP

- Connection rate (connessioni / sec)
- Server throughput (Mbit / sec)
- Error level (percentuale di errori sul totale richieste)

QoWSbench introduce:

- Caratterizzazione del carico
- ⇒ ■ Qualità del servizio

Statistiche (QoWSbench)

In più:

- Classificazione degli utenti
- Un report per ogni classe di utenti

Informazioni aggiuntive nei report:

- Drop level (percentuali degli scarti)
- Cumulativa e 95-percentile del tempo di risposta

Risultati sperimentali:

- Tempi di risposta
- Distribuzioni cumulative
- Media vs percentile

Ambiente di testing

hadrianus (webmaster) claudius zenobia valerianus Ethernet Switch
tiberius traianus saetta (dispatcher) augustus caesar

- Workload: 80% statico, 20% dinamico
- 2 classi: high (40%) e low (60%)
- Numero di processi client utilizzati: 100 ÷ 250
- Politiche utilizzate: Dynamic Partition, Switch Admission e Client Aware Policy (CAP) statica

Risultati sperimentali:

- Tempi di risposta
- Distribuzioni cumulative
- Media vs percentile

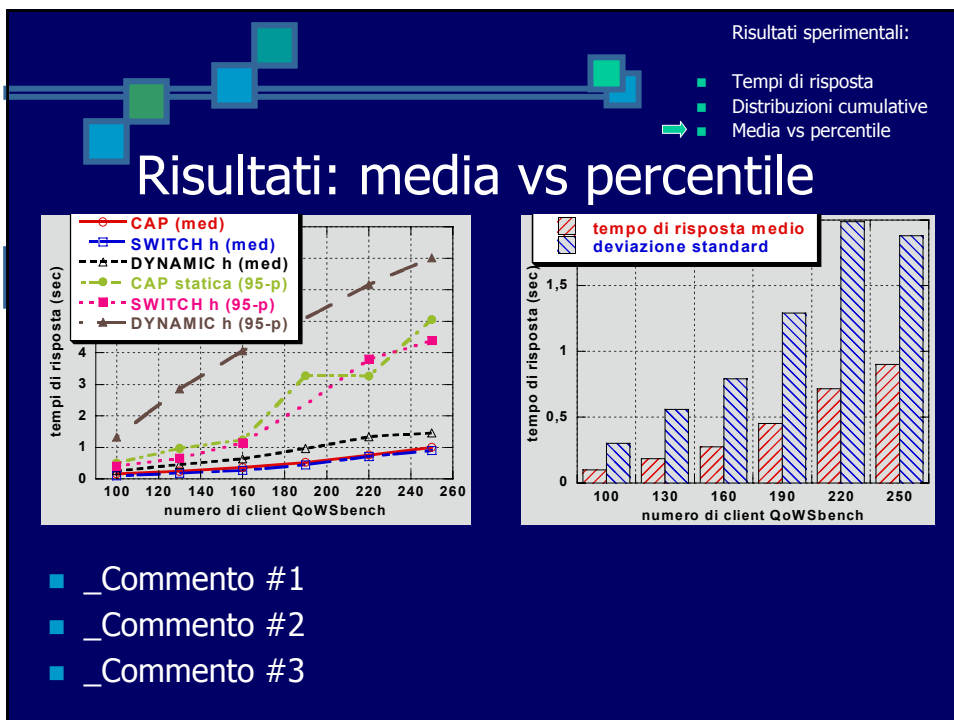
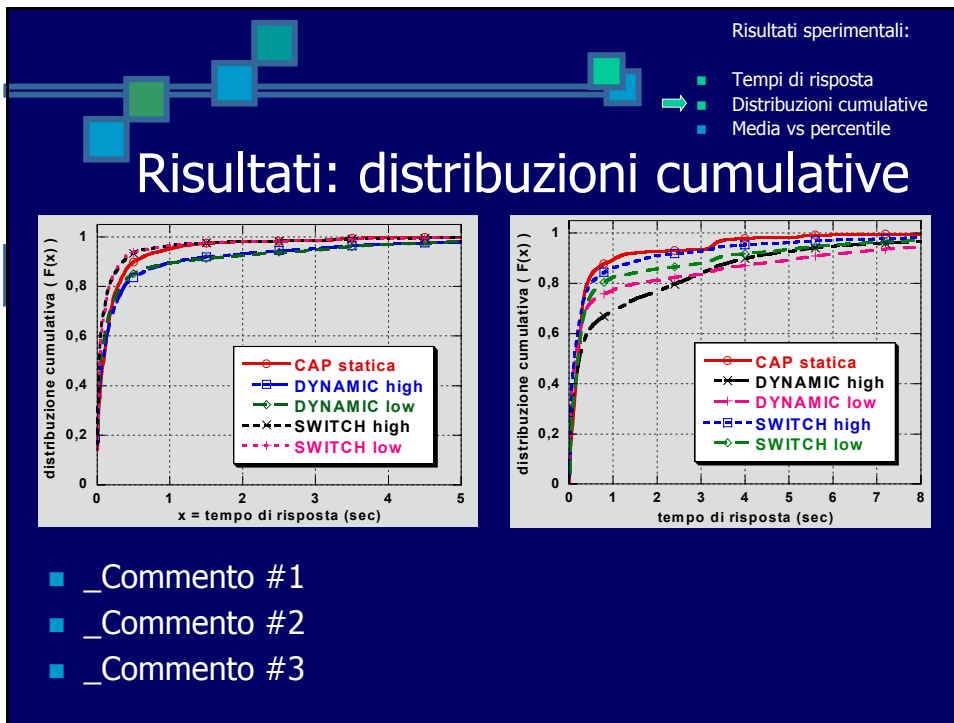
Risultati: tempi di risposta

tempo di risposta medio (sec)

95-percentile del tempo di risposta (sec)

numero di client QoWSbench

- _Commento #1
- _Commento #2
- _Commento #3



Conclusioni

- Statistiche più significative
- Il tuning di un prototipo "stabile" a QoS garantita risulta molto semplificato
- Alta variabilità dei valori medi nel Web

Sviluppi futuri

Aggiunta di funzionalità (HTTP/1.1):

- Connessioni persistenti
- Metodi diversi da GET (ad es. POST)
- Transazioni cifrate tramite SSL
- Supporto dei cookie