Tesi di Laurea

WebSim: un simulatore basato su tracce per sistemi Web distribuiti localmente

Candidato:

Mauro Ranchicchio

Relatore:

Prof. Salvatore Tucci

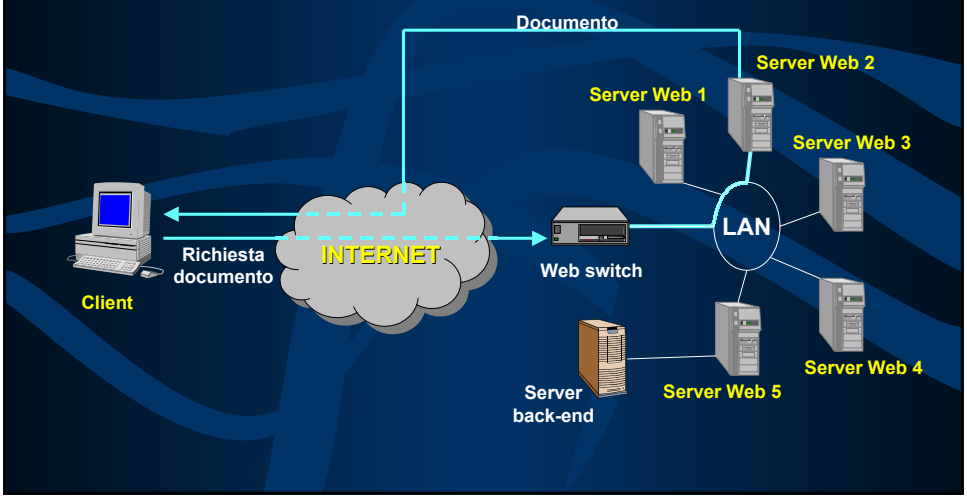
Correlatore:

Ing. Valeria Cardellini

Sommario

- Sistemi Web Cluster one-way
- Misurazione del carico nel Web
- Simulazione di un sistema Web Cluster
- Implementazione del simulatore a tracce
- Politiche di dispatching e QoWS
- Caratterizzazione del carico
- Risultati dell'analisi simulativa
- Conclusioni e sviluppi futuri

Sistemi Web Cluster one-way: architettura



Misurazione del carico nel Web: le tecniche

- ***Server Logging:*** registrazione di tutte le richieste HTTP trattate dal server Web
- ***Proxy Logging:*** utilizzato per valutare politiche di caching e popolarità delle risorse
- ***Client Logging:*** implica modifiche al codice del browser; utilizzato per conoscere gli schemi di navigazione degli utenti
- ***Packet Monitoring:*** "cattura" dei singoli pacchetti IP transitanti per un link di rete o attraverso un router
- ***Misurazioni attive:*** generazione di richieste in modo controllata ed osservazione prestazioni

Common Log Format

Common Log Format (CLF): è il formato di *server log* più diffuso; è adottato come default dal server Apache

Esempio di istanza:

Il formato del record consiste di **sette campi**:

- **host remoto**
- **identità remota**
- **utente autenticato**
- **time-stamp**
- **stringa della richiesta**
- **codice di stato della risposta**
- **numero di byte trasferiti**

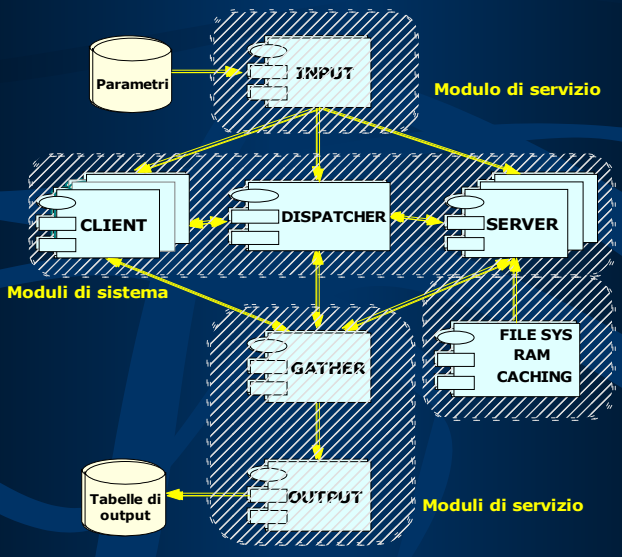
Simulazione di un sistema Web Cluster: il motore di simulazione CSIM

CSIM è una libreria di simulazione a eventi discreti, orientata ai processi

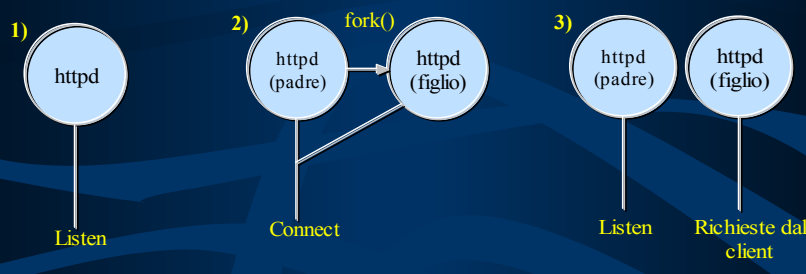
- **Processi:** Modellazione **entità attive** del sistema (Client, processi HTTP, processi di gestione del dispatching, etc)
- **Eventi:** un processo può attenderne l'occorrenza o settarlo
- **Risorse:**
  - ***Facility:*** Modellazione **risorse con uso seriализzato** (CPU, server back-end, Web switch, dischi)
  - ***Storage:*** Modellazione **memoria centrale** dei server Web
- **Mailbox:** strutture per scambio di informazioni sotto forma di messaggi

Interazione e **comunicazione** tra processi:

Simulazione di un sistema Web Cluster: lo schema funzionale di WebSim

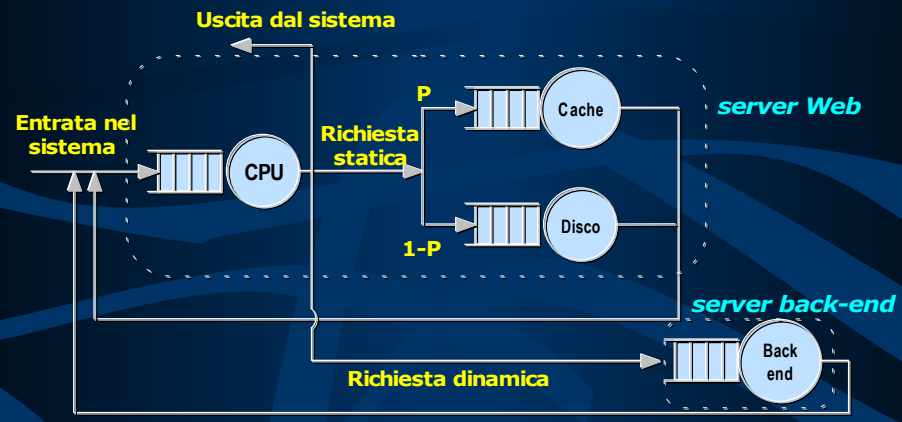


Simulazione di un sistema Web Cluster: processi server



Il **server** rimane in attesa di connessioni: per ciascuna nuova richiesta **client** assegnatagli dal **Web switch**, crea **un nuovo processo figlio** che si occuperà di soddisfarla

Implementazione del simulatore a tracce WebSim: modellazione nodi server



Richieste dinamiche: sono servite da appositi nodi *back-end* (*application* o *database server*)

Esempio di **richiesta dinamica**:

```
"GET /cgi-bin/script.cgi?param1=val1&param2=val2...
                                  ...param3=val3 HTTP/1.1"
```

Simulazione di un sistema Web Cluster: generazione del carico

Il **workload** da sottoporre ad un simulatore può essere generato:

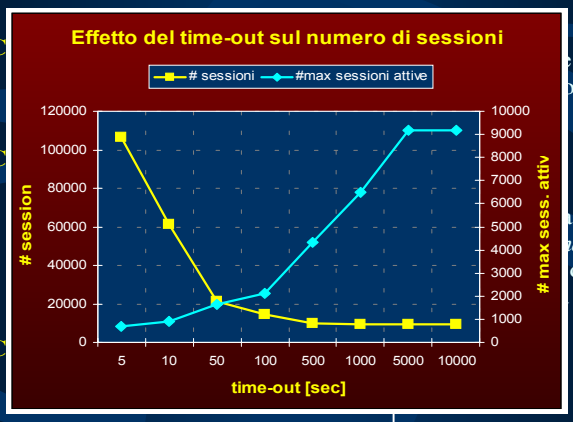
- **Analiticamente:** *distribuzioni probabilistiche, invarianti del traffico Web*
- **Da tracce:** *ricostruzione delle sessioni, dati memorizzati in file di log*

WebSim è un simulatore **basato su tracce** ricavate dall'analisi di server log in formato CLF

Riproduzione del **comportamento degli utenti** registrato nel periodo di osservazione

Implementazione del simulatore a tracce WebSim: ricostruzione sessioni utente

Determinazione di un time-out adeguato per suddividere in sessioni il traffico registrato proveniente dallo stesso client



Si adotta un time-out di 100 secondi

Caratterizzazione del carico

Le tracce sono state ottenute tramite elaborazione di file di log del sito *World Cup'98* rappresentanti **differenti tipologie** di carico

Carico	Arco temporale	#File distinti trasferiti	Dimensione media file	Frequenza hit
Low	1 giorno	3266	21517 byte	8.7 hit/sec
Mid	28' 18''	5189	14903 byte	396 hit/sec
High	5' 54''	3748	9540 byte	1912 hit/sec

Carico	% Richieste dinamiche
Heavydyn	~80%

Il quarto tipo di carico (*Heavydyn*) è rappresentativo del mix di richieste verso un sito di *e-commerce*

Qualità del Servizio Web: principi

- **Classificazione**
  - Identificazione classi di utenti e servizi
  - Classificazione utenti e servizi
- **Isolamento delle prestazioni**
  - Politiche di scheduling con priorità
  - Partizionamento delle risorse
- **Alta utilizzazione delle risorse**
  - Partizionamento dinamico delle risorse
- **Richiesta di ammissione**
  - DICHIARAZIONE: stima della richiesta di risorse
  - CONTROLLO DELL'ACCESSO: decisione sull'ammissione della richiesta di connessione

Analisi simulativa: organizzazione

Sono stati svolti gli esperimenti utilizzando quattro differenti **politiche di dispatching**:

- **Analisi di scalabilità:** sensibilità delle prestazioni del sistema al numero di nodi server nel cluster
- Politiche di **4° livello** dinamiche**:**
  - Totalità delle richieste
  - Richieste dinamiche
  - Risorse statiche *large file* (oggetti *non-embedded*)

- **Analisi della Qualità del Servizio Web:** verifica del rispetto dei Service Level Agreement e confronto tra tempi di servizio per classe High e classe *Low*

Weighted Round Robin

Least Loaded

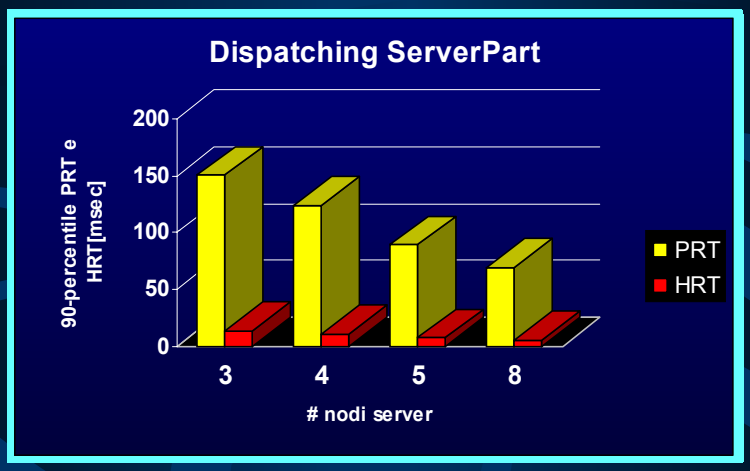
SwitchPriority

(*classificazione e controllo di ammissione*)

StaticServerPartitioning

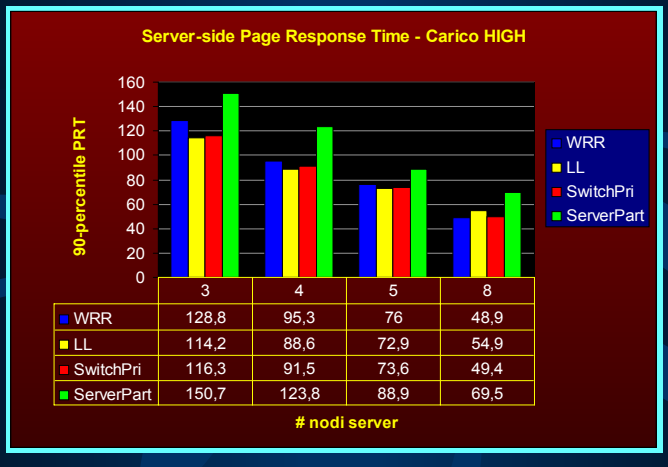
(*isolamento delle prestazioni*)

Risultati dell'analisi simulativa: analisi di scalabilità



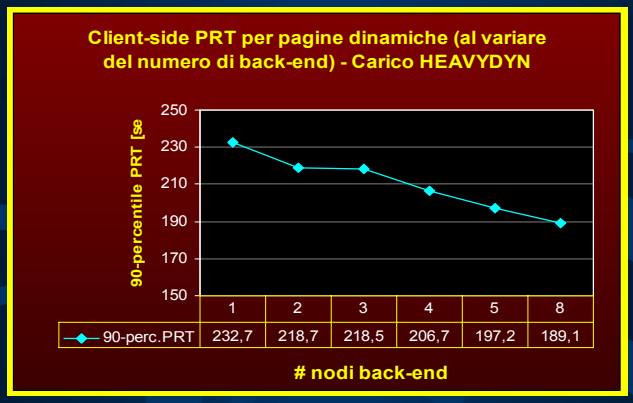
Poli

Risultati dell'analisi simulativa: analisi di scalabilità



Carico *High:* evidenzia meglio degli altri il vantaggio apportato dallo scale-out del Web cluster

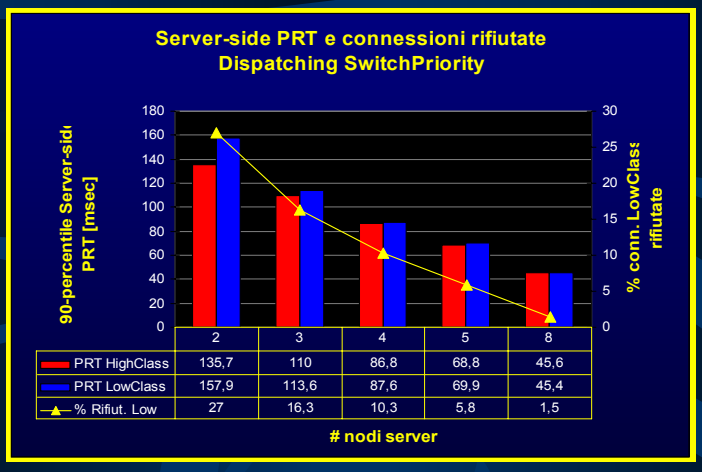
Risultati dell'analisi simulativa: richieste dinamiche



Generazione dinamica dei contenuti: conduce ad **incrementi sul PRT dal 71% al 163%** per i tre scenari considerati

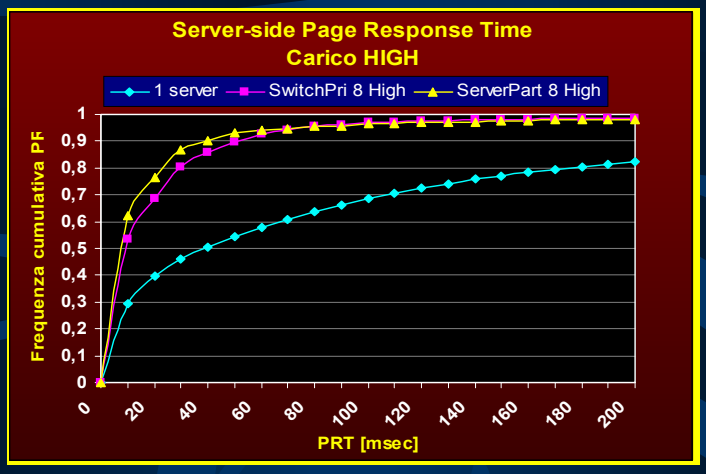
Il carico *Heavydyn* evidenzia i vantaggi dello scale-out dell'insieme di server back-end

Risultati dell'analisi simulativa: QoWS



La politica **SwitchPriority** riesce a garantire il SLA nel *Server-side PRT* solamente per sistemi ad alto numero di nodi

Risultati dell'analisi simulativa: QoWS



La politica **ServerPartitioning** permette di rispettare sempre il SLA, ma al costo di un maggior numero di richieste rifiutate di classe *Low*

Conclusioni

- E' stato implementato un **simulatore basato su tracce per sistemi Web Cluster**, adottando diverse politiche di switching
- L'**analisi di scalabilità** ha evidenziato la capacità delle politiche dinamiche di 4° livello di beneficiare dello scale-out in modo pressoché lineare
- Sono state confrontate le politiche **ServerPartitioning** e **SwitchPriority**, nell'ottica della QoWS

Sviluppi futuri

- Utilizzo tracce da siti di **e-commerce** o **information retrieval** (alta presenza di contenuti dinamici)
- Classificazione delle richieste dinamiche per l'**utilizzo di politiche Client-aware** (es., *CAP*)
- Utilizzo di formati di log con migliore granularità del time-stamp