

Tesi di Laurea

Dispositivi e politiche di switching per cluster di Web server

Candidato:

Massimiliano Antonioli

Relatore:

Prof. Salvatore Tucci

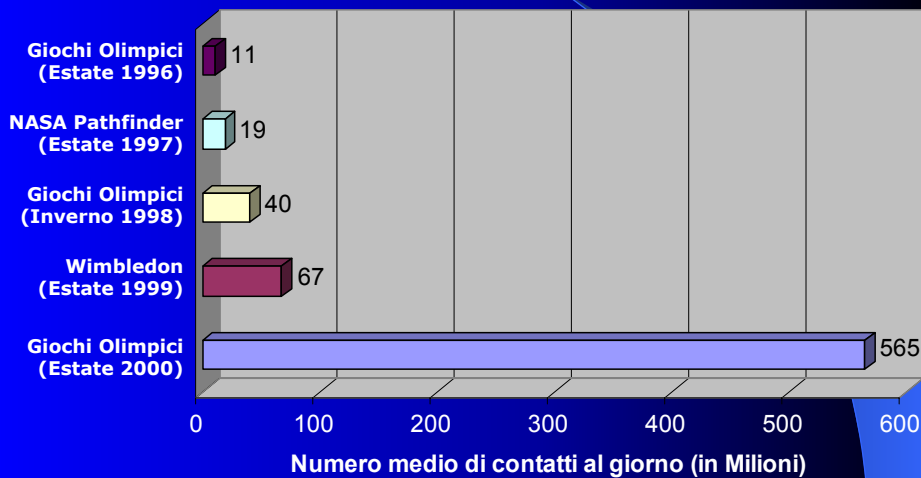
Correlatore:

Prof. Michele Colajanni

Sommario

- Siti Web popolari: traffico e soluzioni
- Web Server Localmente Distribuiti
- Algoritmi di selezione
- Politiche Dinamiche vs. Politiche Statiche
- Implementazione di Politiche Dinamiche
- Architettura del prototipo
- Risultati sperimentali
- Conclusioni e sviluppi futuri

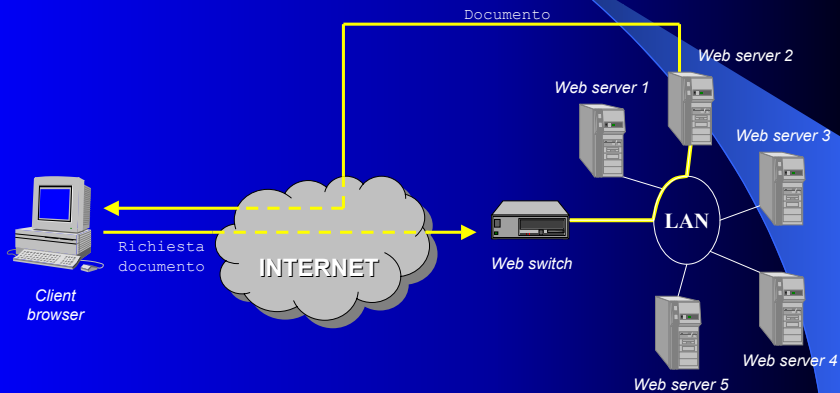
Siti Web popolari: Traffico



Siti Web popolari: Soluzioni

- Incrementare la capacità del Web server
- Mirroring
- **Distribuire le richieste**
 - Architetture locali o geografiche
 - Trasparenza
 - Scalabilità
 - Affidabilità
 - Tolleranza ai guasti

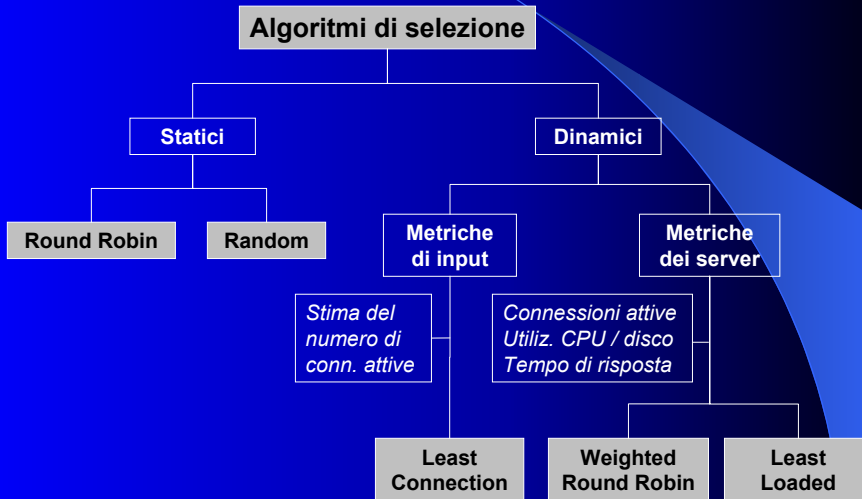
Web Server Localmente Distribuiti: Architettura



Web Server Localmente Distribuiti: Algoritmi

- Quarto Livello OSI
 - TCP/IP
 - Content blind
 - Statici e dinamici
- Settimo Livello OSI
 - Application (HTTP)
 - Content aware
 - Statici e dinamici

Algoritmi di Quarto Livello



Algoritmi Statici vs. Dinamici

Statici

- Facile implementazione
- Overhead trascurabile (sullo switch)
- Possibili situazioni di sbilanciamento del carico

Dinamici

- Implementazione più complessa
- Overhead di comunicazione (tra switch e server) e di computazione (sullo switch)
- Migliore bilanciamento del carico

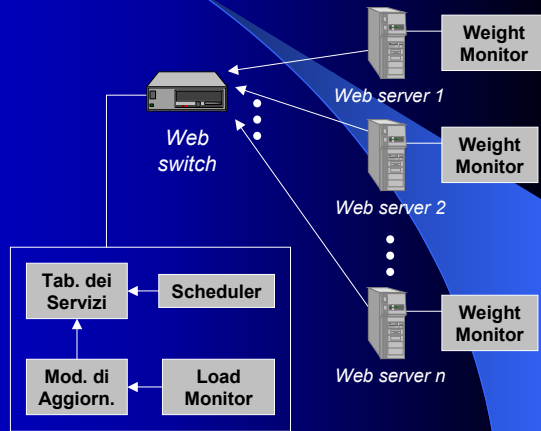
Componenti del sistema

Web switch

- Load Monitor
- Modulo di Aggiornamento
- Scheduler

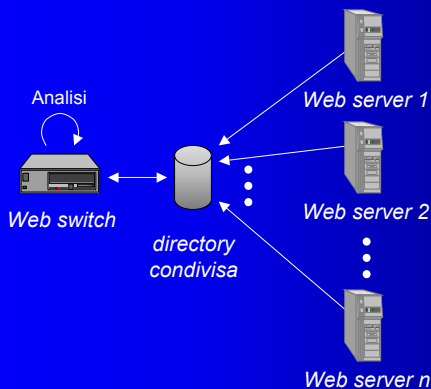
Web server

- Weight Monitor

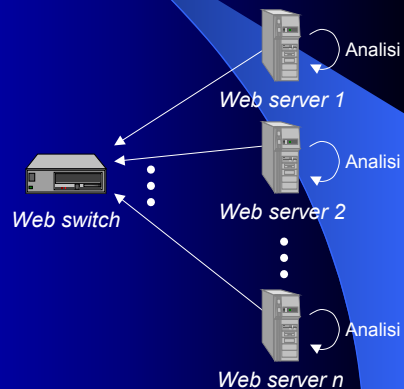


Meccanismi di comunicazione

Network File System



Socket



Meccanismi di comunicazione

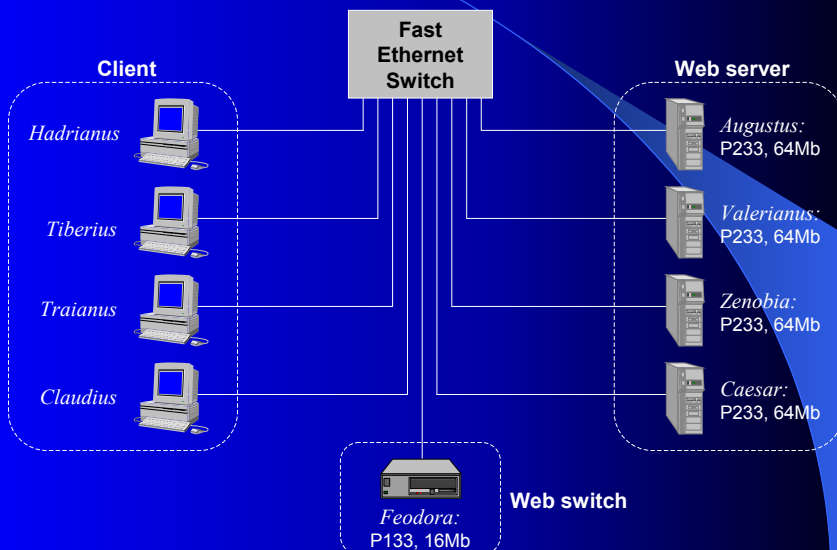
Network File System

- Facile implementazione
- Analisi delle connessioni attive sullo switch
- Viene introdotto un overhead di comunicazione non trascurabile

Socket

- Implementazione più complessa
- Analisi delle connessioni attive sui server
- L'overhead di comunicazione introdotto è trascurabile

Architettura del prototipo



Scenari di test

Webstone

- Aggiunto il Think Time
- Reso il codice più robusto
- Aggiunta la generazione casuale del numero di oggetti Web che compongono una pagina HTML

Workload

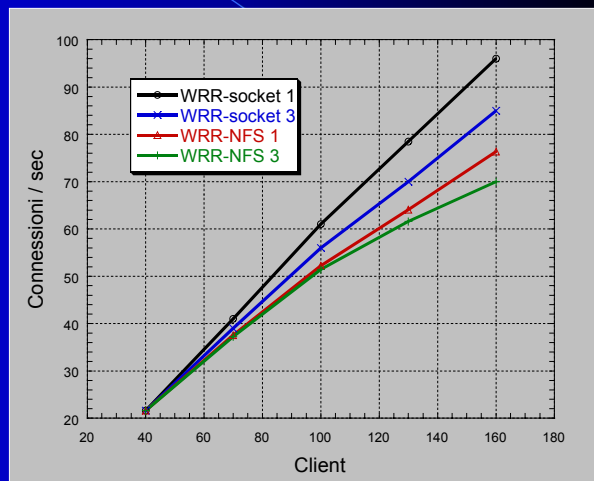
- Statico (solo pagine statiche)
- Dinamico light (20% di richieste dinamiche)
- Dinamico heavy (50% di richieste dinamiche)

Metriche misurate

- Connessioni / sec
- Mbit / sec
- Tempo di risposta (sec)

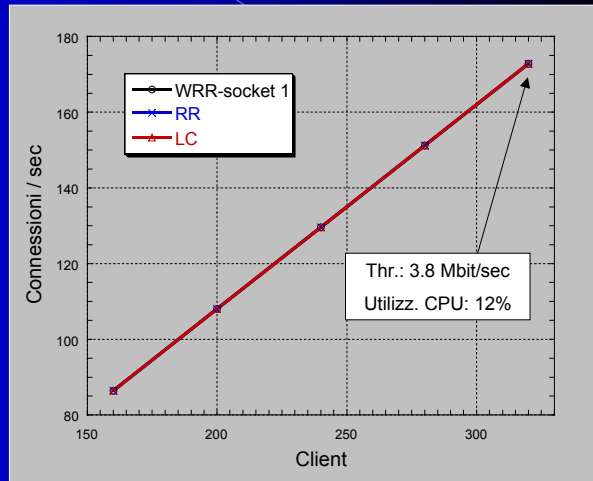
Confronto tra politiche dinamiche

- Le prestazioni peggiorano all'aumentare dell'intervallo di campionamento dei carichi
- L'implementazione via socket è più efficiente della implementazione via NFS



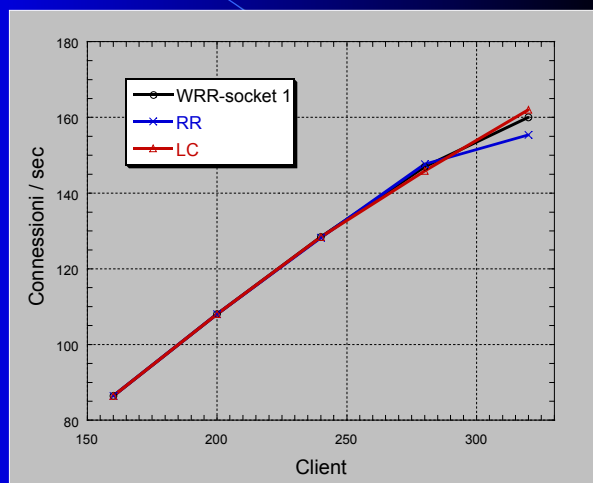
Risultati sperimentali: Workload statico

- Massimo throughput raggiungibile dalla rete LAN su cui sono stati effettuati gli esperimenti: 4.3 Mbit/sec
- A 4 Mbit/sec circa, i Web server sono ben lontani dalla saturazione
- La rete satura prima di poter effettuare un confronto tra politiche statiche e dinamiche



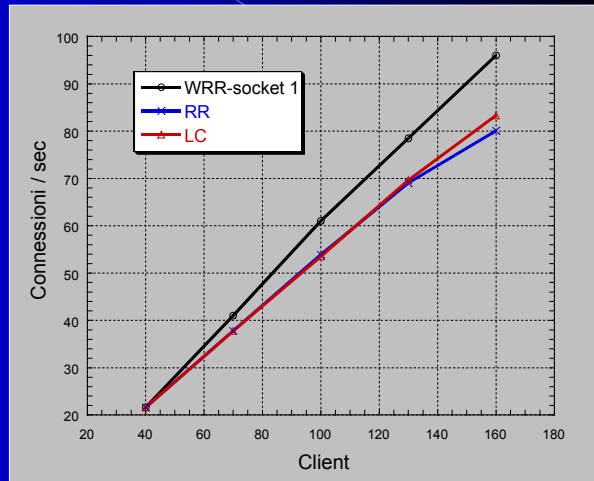
Risultati sperimentali: Workload dinamico light

- Le politiche statiche e dinamiche hanno un comportamento simile
- La migliore distribuzione e l'overhead di comunicazione introdotto si compensano



Risultati sperimentali: Workload dinamico heavy

- La politica dinamica risulta nettamente più efficiente delle politiche statiche
- La migliore distribuzione compensa ampiamente l'overhead di comunicazione introdotto



Conclusioni e sviluppi futuri

Conclusioni:

- Realizzazione di un sistema di Web server localmente distribuito
- Implementazione di algoritmi di distribuzione dinamica del carico
- Le politiche dinamiche si rivelano molto più efficienti per siti Web che gestiscono informazioni dinamiche (per esempio database o e-commerce)

Sviluppi futuri:

- Studio ed implementazione delle politiche dinamiche di settimo livello OSI