

Università di Napoli
13 febbraio 2001

Sistemi ad Alte Prestazioni per Servizi Web

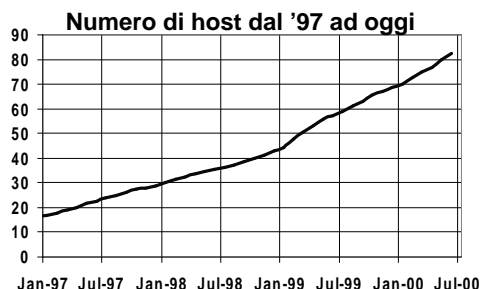
Prof. Michele Colajanni
Dipartimento di Ingegneria dell'Informazione
Università di Modena e Reggio Emilia
colajanni@unimo.it

Argomenti

0. Motivazioni e background
1. Sistemi Web ad alte prestazioni distribuiti localmente
2. Sistemi Web a Qualità del Servizio garantita (*cenni*)
3. Sistemi Web ad alte prestazioni distribuiti geograficamente

Motivazione 1: il successo di Internet

Host collegati	
Gennaio 1993	1.313.000
Luglio 1993	1.776.000
Gennaio 1994	2.217.000
Luglio 1994	3.212.000
Gennaio 1995	4.852.000
Luglio 1995	6.642.000
Gennaio 1996	9.472.000
Luglio 1996	12.881.000
Gennaio 1997	16.146.000
Luglio 1997	19.540.000
Gennaio 1998	29.670.000
Luglio 1998	36.739.000
Gennaio 1999	43.230.000
Luglio 1999	56.218.000
Gennaio 2000	72.340.000



Anni impiegati per 50M utenti	
Radio	38
Televisione	13
TV via cavo	10
Internet	5

Fonte: www.isc.org

© Michele Colajanni, 2001

2

Motivazione 2: il successo del Web

[Load misurato in hit]

Yahoo, Netscape, Microsoft, Pointcast, AltaVista, CNN, ... (>50 Milioni hits/day)

Evento	Periodo	Peak hits/day	Peak hits/minute
NCSA server (Oct. 1995)		2 Milioni	
Olympic Games 1996 (Atlanta, 1996)	180 Milioni	8 Milioni	
NASA Pathfinder (July 1997)	942 Milioni (14 days)	40 Milioni	
Olympic Winter Games (Japan, 1998)	634.7 Milioni (16 days)	55 Milioni	110.000
FIFA World Cup (France, 1998)	1.350 Milioni (90 days)	73 Milioni	209.000
Wimbledon (July, 1999)	942 Milioni (14 days)	125 Milioni	430.000
Wimbledon (July, 2000)		282 Milioni	964.000
Olympic Games 2000		875 Milioni	1.200.000

© Michele Colajanni, 2001

3

Motivazione 3: dalla 1^a generazione ...

- Un ulteriore canale per informazione non critica
- 95% dell'informazione costituita da testo più eventuali immagini
- Manutenzione e aggiornamenti occasionali
- Canale economico
- Prestazioni molto variabili
- Affidabilità non garantita
- Sicurezza non necessaria



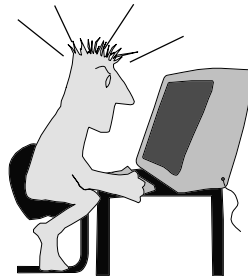
*"I don't know why.
I don't care too much"*

© Michele Colajanni, 2001

4

... alla 2^a generazione del Web

- Canale di informazione critica, che sta diventando privilegiato per molti utenti
- Sistema transazionale
- Contenuti dinamici ed attivi in continuo aumento
- Streaming audio e video
- Servizi a pagamento (diretto o indiretto)
- "Vetrina" importante per industrie e organizzazioni



"I paid for it!!!"

"What a primitive company!"

© Michele Colajanni, 2001

5

Ingredienti del Web

- **Meccanismi di comunicazione e naming di Internet**
 - TCP/IP
 - DNS
- **Tre nuovi standard**
 - Sistema di indirizzamento delle risorse URL
 - Linguaggio HTML
 - Protocolli HTTP
- **Sistema client-server**
 - Funzionamento Client (*browser*)
 - Componenti e funzionamento Server Web

© Michele Colajanni, 2001

6

Ingredienti di Internet

Componenti

- host/nodo
- router
- Internet Service Provider
- LAN, Ethernet
- indirizzo IP (*netid, hostid*)
- classi di indirizzi IP

Protocolli

- packet switching
- suite protocolli Internet
- protocollo IP
- protocollo TCP

Naming

- hostname
- sistema DNS
- name server
(*locali, root, top-level, autoritativi*)

© Michele Colajanni, 2001

7

Internet: Cosa non è ...

- Non è una singola rete, ma un insieme di reti esteso in tutto il mondo
- Non è governata da un gruppo né da un ente né da un'azienda (*anche se alcuni spot pubblicitari inducono a ritenere il contrario ...*)
- Non è gestita in modo centralizzato perché tutte le singole reti che compongono Internet hanno una gestione autonoma
- **E soprattutto: Non è sinonimo di World Wide Web**

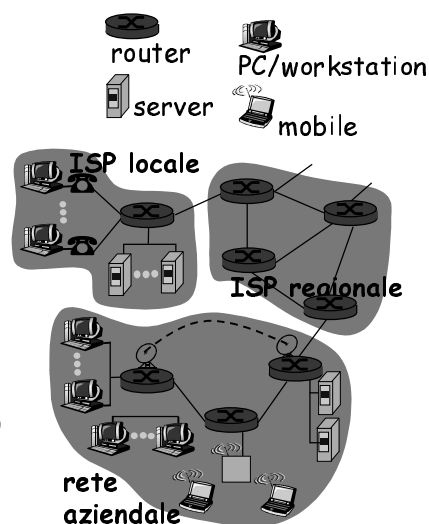
© Michele Colajanni, 2001

8

INTERNET: Componenti

Componenti principali:

- Milioni di dispositivi connessi detti **nodi** o **host**: PC, server, workstation, cellulari, PDA, TV, agende elettroniche, frigoriferi,...
- **Link di comunicazione**: cavi, fibra ottica, radio, satellitari,...
- **Router**: dispositivi che instradano i messaggi da un punto all'altro della Rete



© Michele Colajanni, 2001

9

Suite di protocolli Internet (TCP/IP)

- **Applicativo:** supporta le applicazioni di rete: **FTP, HTTP, SMTP, ...**
- **Trasporto:** supporta i trasferimenti da host a host: **TCP, UDP**
- **Rete:** trasferisce i pacchetti dal nodo mittente al destinatario: **IP**
- **Link:** effettua i trasferimenti dei dati tra componenti della rete confinanti: PPP, Ethernet, ...
- **Fisico:** trasferisce bit "sul cavo"

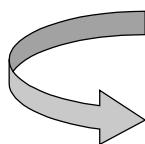


© Michele Colajanni, 2001

10

Principali applicazioni Internet

- **Domain Name System**
- **Posta elettronica (SMTP)**
- **Login remoto (Telnet)**
- **Trasferimento file (FTP)**
- **World Wide Web**



Tutte usano il modello *client/server*

© Michele Colajanni, 2001

11

Internet: Identificatori

I dispositivi collegati ad Internet (*host, router*) sono caratterizzati tipicamente da due identificatori:

- **Indirizzo IP** (*numero di 32 bit*): utilizzato per indirizzare ed instradare i pacchetti nella rete
- **Hostname** (*stringa alfanumerica*): nome logico utilizzato dalle persone

Indirizzi IP

4 classi principali di indirizzi IP:

classe A, classe B, classe C, classe D

Classe

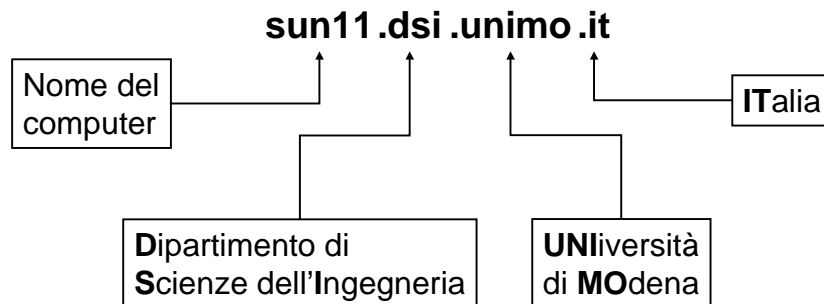
A	0	netid	hostid	da 0.1.0.0 a 126.0.0.0
B	10	netid	hostid	da 128.0.0.0 a 191.255.0.0
C	110	netid	hostid	da 192.0.0.0 a 239.255.255.0
D	1110	multicast address		da 240.0.0.0 a 247.255.255.255

← 32 bit →

Hostname

- Dato lo scopo rivolto verso l'utente, all'hostname si preferiscono attribuire valori mnemonici.

Es.: *nome del computer e posizione:*

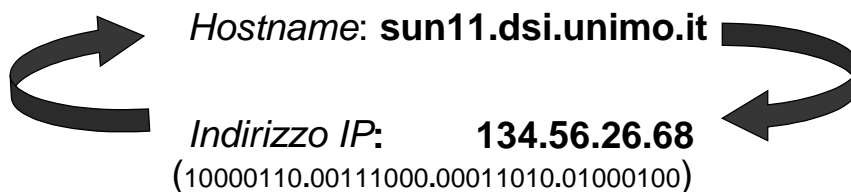


© Michele Colajanni, 2001

14

Domain Name System (DNS)

- Realizza uno spazio dei nomi gerarchico e permette la traduzione del nome mnemonico di un host in un indirizzo IP. Es.



- Implementa un meccanismo efficiente (mediante **name servers**), distribuito su scala geografica, per convertire un hostname in un indirizzo IP e viceversa.

© Michele Colajanni, 2001

15

Domini di massimo livello

Nome del Dominio	Significato
COM	Organizzazioni commerciali
EDU	Istituzioni USA per l'istruzione
GOV	Istituzioni governative USA
MIL	Istituzioni militari USA
NET	Maggiori centri di supporto per la rete
ORG	Organizzazioni senza scopo di lucro diverse dalle precedenti
ARPA	Dominio temporaneo della rete ARPANET (<i>obsoleto</i>)
INT	Organizzazioni internazionali (<i>schema geografico</i>)
Codice nazionale (it, ch, fr, jp, ...)	Nomi nazionali (<i>schema geografico</i>)

+ i nuovi nomi "top-level domain"

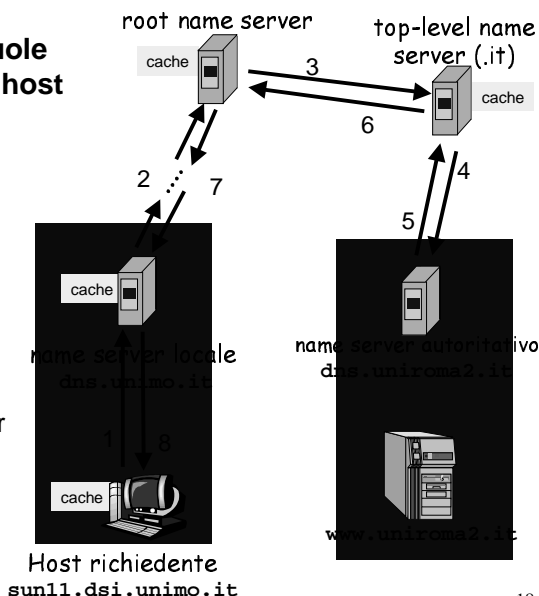
Sistema DNS: *Meccanismo di risoluzione*

- **Nessun name server ha tutte le corrispondenze tra *Hostname* e *Indirizzo IP***
- Esistono vari *name server*
 - **Locali**: ogni ISP e organizzazione gestisce uno o più name server locali
 - **Autoritativi**: relativamente ad un hostname
 - **Root**: contattati dai name server periferici quando non riescono a risolvere un indirizzo, contattano i name server autoritativi per quell'indirizzo o name server intermedi (di zona)
 - **Top-level**: relativi ai domini top-level

Sistema DNS: *address resolution*

L'host `sun11.dsi.unimo.it` vuole conoscere l'indirizzo IP dell'host `www.uniroma2.it`

- 1) Contatta il suo *DNS locale*
- 2) Se necessario, il *DNS locale* può contattare altri DNS intermedi, ed eventualmente uno dei *root DNS*
- 3) Se necessario, il *root DNS* contatta il *DNS autoritativo* per quell'indirizzo (o un *top-level DNS* nel caso in cui non conosca un *DNS autoritativo* per quell'indirizzo)



© Michele Colajanni, 2001

18

Uniform Resource Locator (URL)

`schema://host.domain/pathname`
`http://www.dsi.unimo.it/docenti/orari/esami.html`

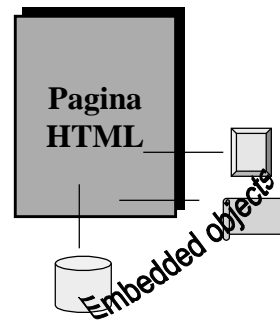
- **schema**: indica il modo con cui accedere alla risorsa, cioè quale protocollo bisogna usare per interagire con il server che controlla la risorsa. Il metodo di accesso più comune è **HTTP** (protocollo nativo del WWW per il recupero di risorse Web)
- **host.domain**: è l'hostname del nodo nel quale risiede la risorsa Web.
- **pathname**: identifica la risorsa presso il server Web.

© Michele Colajanni, 2001

19

Risorse Web

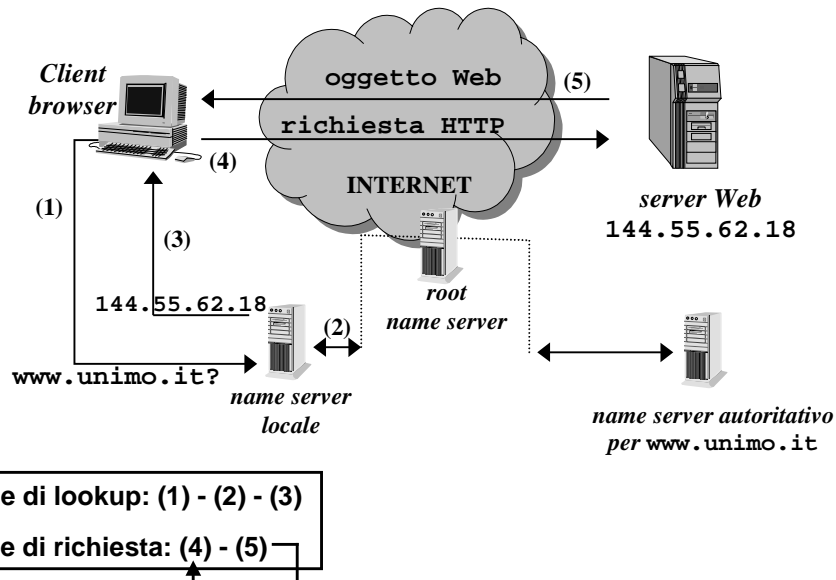
- **Documenti** (detti anche *pagine*)
 - testo
 - immagini
 - suoni
 - video
- **Risultati di esecuzioni**
- **Programmi eseguibili**



Alcune definizioni

- *Sessione utente*: serie di *richieste di risorse* effettuate dallo stesso utente al medesimo sito Web
- *Richiesta di risorsa (o pagina)*: una richiesta che tipicamente consiste di multipli *hit* inviati dal client dell'utente al sito Web
- *Hit*: una richiesta per un singolo oggetto effettuata dal client al server Web

Fasi di una richiesta HTTP



© Michele Colajanni, 2001

22

Protocollo HTTP

- **HyperText Trasmission Protocol (HTTP)** è il protocollo che permette il reperimento delle risorse Web.
- E' un **protocollo applicativo** (7° livello) di richiesta/risposta basato tipicamente sulla suite di protocolli TCP/IP.
- Tutti i client e server Web devono supportare il protocollo HTTP per poter scambiare richieste e risposte. Per questa ragione i **client** e i **server Web** sono chiamati anche **client HTTP** e **server HTTP**.

© Michele Colajanni, 2001

23

Richiesta HTTP

- Una *richiesta HTTP* comprende un **metodo**, un **URL**, l'identificativo della **versione del protocollo HTTP**, ed un insieme di **extension header**.
- Il **metodo** specifica il tipo di operazione che il client richiede al server. Il metodo più comune è **GET** che serve per acquisire pagine Web.
- Gli **header** contengono informazioni addizionali, quali la data e l'ora della comunicazione, il tipo di software utilizzato dal client, i tipi di dato che il browser è in grado di visualizzare, per un totale di circa 50 tipi di header differenti.

Risposta HTTP

- Una *risposta HTTP* comprende, oltre al contenuto della risorsa richiesta, un **header** contenente l'identificativo della versione del protocollo HTTP, il codice di stato, l'informazione di stato in forma testuale, ed un insieme di possibili altre informazioni di risposta.
- Se la pagina richiesta, oltre al testo HTML, contiene altri oggetti, ciascuno di essi sarà identificato da un URL differente, per cui è **necessario che il browser invii un esplicito messaggio di richiesta per ognuno degli elementi collegati alla pagina**.

Alcuni codici di risposta

Esito	Codice numerico (status code)	Specifica testuale (reason phrase)
OK, la risorsa è stata trovata e viene inviata	200	OK
Risorsa spostata	301	Moved permanently
Risorsa non modificata	304	Not Modified
Richiesta non valida	400	Bad request
Richiesta di autenticazione	401	Unauthorized
Richiesta di pagamento	402	Payment required
Accesso vietato	403	Forbidden
Risorsa non esistente	404	Not found
Errore nel server	500	Server error
Non implementato	501	Not Implemented
Gateway difettoso	502	Bad Gateway
Servizio non disponibile	503	Service Unavailable

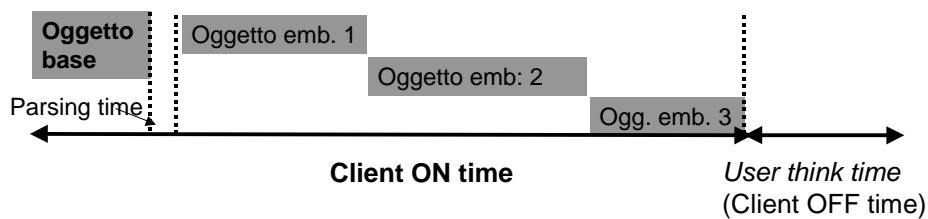
© Michele Colajanni, 2001

26

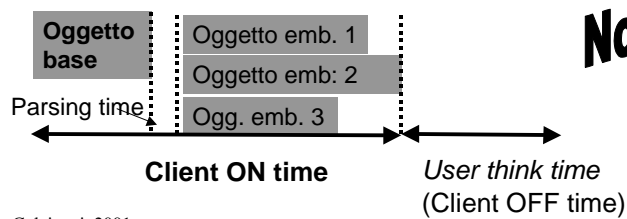
Protocolli HTTP/1.0 e HTTP/1.1

[Hei97, Bar98, Bar99b, Kri99]

HTTP/1.0



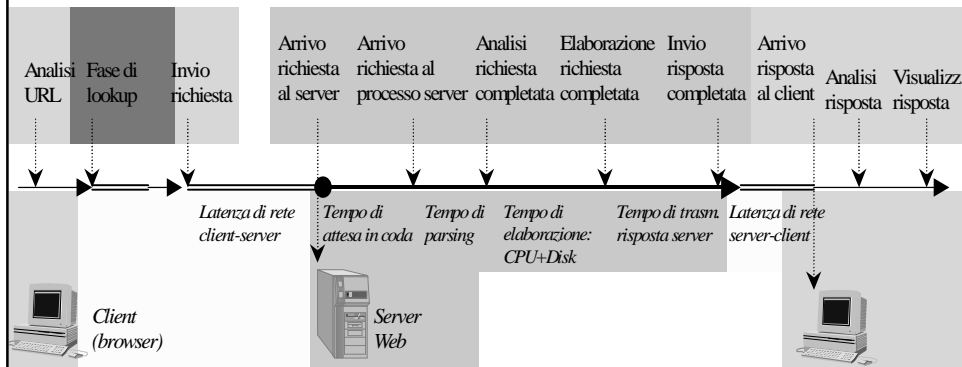
Pipelining HTTP/1.1



© Michele Colajanni, 2001

27

Interazioni client-server Web



Dov'è il collo di bottiglia?

- | | |
|------------------------|-------------|
| (1) Client/conessione? | (2) DNS? |
| (3) Rete? | (4) Server? |

© Michele Colajanni, 2001

28

Proiezioni a livello di client

- PC sempre più potenti
- Tecnologie DSL (per risolvere *last mile problem*)
- Connessioni via CableTV (per USA)

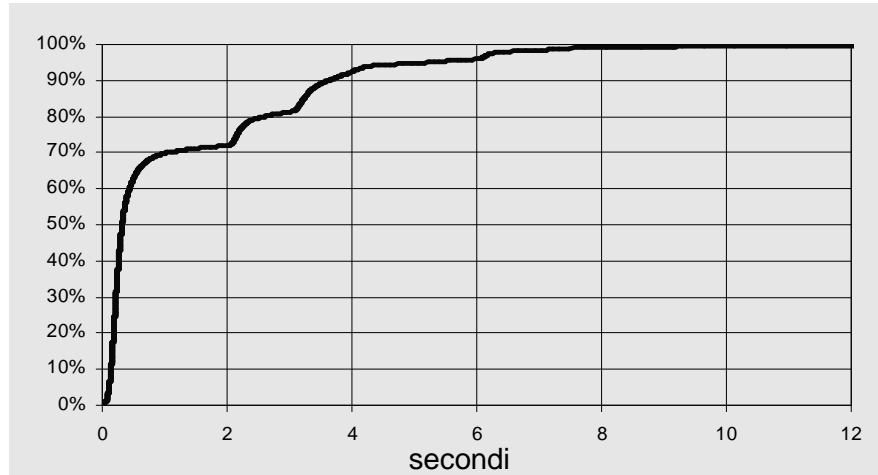
"...e, comunque, il nostro intervento è limitato..."

© Michele Colajanni, 2001

29

Tempi per la risoluzione mediante DNS

(domini random: 70% entro 2 sec., 90% entro 4 sec.)

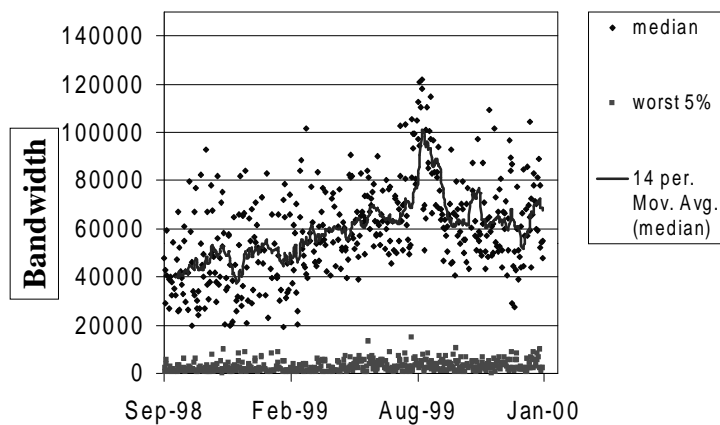


(i nomi di domini popolari hanno tempi inferiori)

© Michele Colajanni, 2001

30

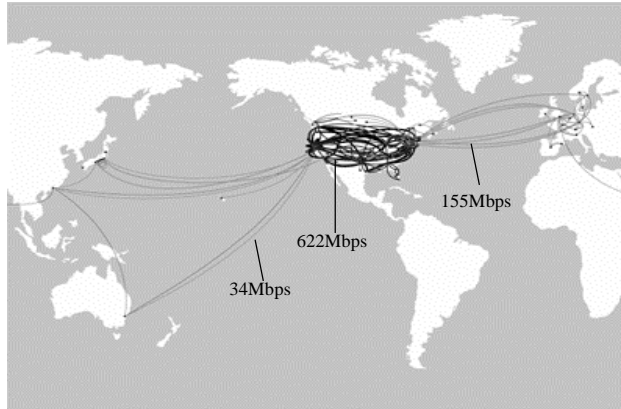
La bandwidth tende ad aumentare



© Michele Colajanni, 2001

31

UUNET Backbones (MCI)

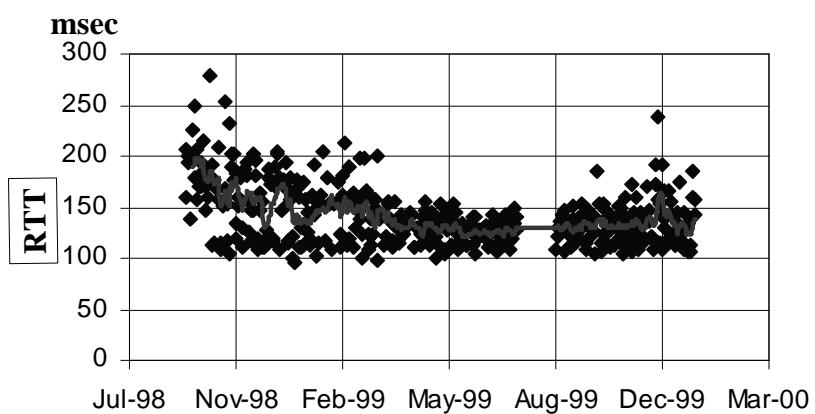


Courtesy of UUNET, 2000

© Michele Colajanni, 2001

32

II *Round-Trip Time* tende a diminuire



© Michele Colajanni, 2001

33

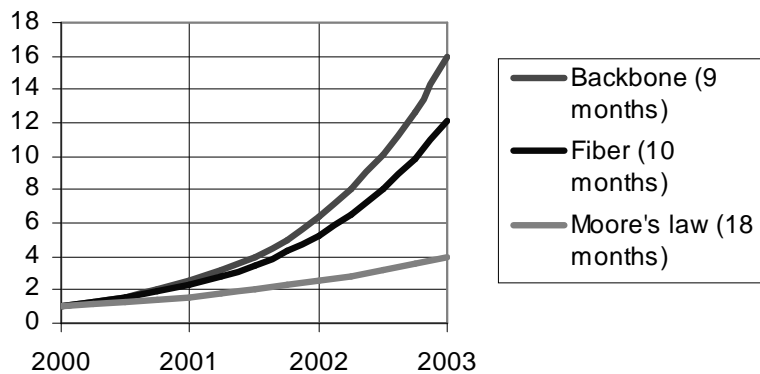
Round-trip times (marzo 2000)

- Italia - Nord Europa <100ms
- Italia - New York 150-200ms
- Italia - Los Angeles 250-300ms
- Italia - Tokio 300-400ms
- Italia - Sidney 300-400ms

Fonte: www.cs.stanford.edu (*The pinger project*)

**I server sono sempre più potenti,
ma tale crescita è sufficiente?**

Proiezione 1: La sfida della rete a 10 Gigabit *(ovvero i limiti della legge di Moore)*



© Michele Colajanni, 2001

36

Proiezione 2: Impatto computazionale delle richieste

**Dal Web Publishing all'
E-Commerce**

- **oggetti statici**
 - piccoli (es., pochi msec)
 - grandi (disk bound)
- **oggetti volatili**
- **oggetti dinamici**
(CPU e/o disk bound)
- **transazioni sicure**
(CPU bound)

Principali componenti		
CPU	Disk	Network



© Michele Colajanni, 2001

37

Potenziamenti del Web

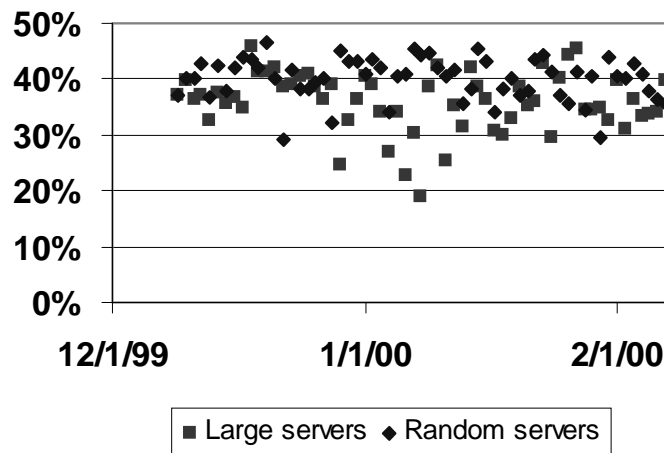
- **Azioni a livello di RETE**
"Network guys are doing an excellent job"
- **Azioni a livello di SERVER** 
- **Azioni a livello di INFRASTRUTTURE**
 - **DNS**
 - **Meccanismi di Caching***

* Vedi Approfondimenti

Componenti di un sito Web

- **Piattaforma hardware**
- **Software di base**
- **Parte informativa del sito Web**
 - Il sito deve mettere a disposizione un insieme di risorse Web che possono essere richieste dai client con cui vengono instaurate delle connessioni HTTP.
- **Server Web**
 - Il processo server ed il relativo software che viene eseguito sulla piattaforma stabilisce il collegamento tra la piattaforma (hardware - software di base) e la parte informativa del sito Web.

Ritardi (percentuali) dovuti al server



© Michele Colajanni, 2001

40

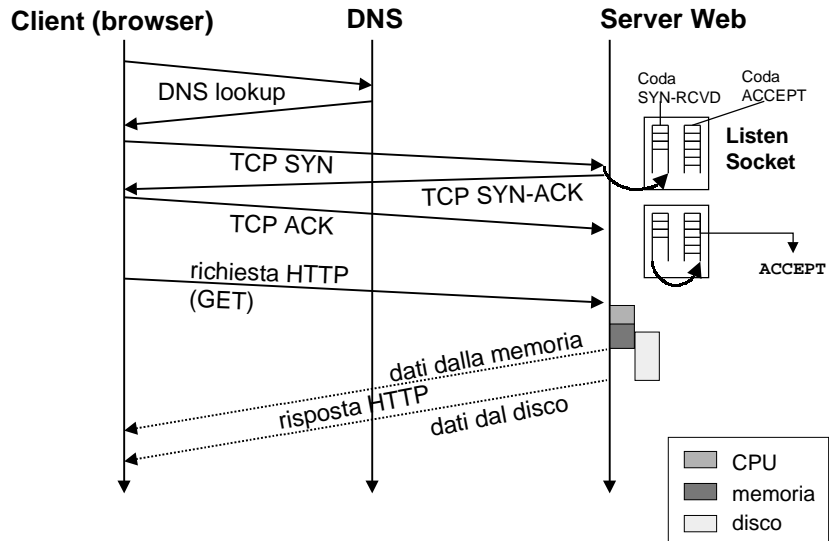
Tempi di gestione di una richiesta (es.)

Passo di elaborazione	CPU (ms)	Disco (ms)
Accettare una connessione TCP/IP	1,0	
Passare i dati dalla connessione TCP/IP al processo server	1,0	
Generare un processo (<i>metodo fork</i>) per l'esecuzione dello script CGI	2,0	
Eeguire uno script CGI che non opera su disco	138,0	8,0
Eeguire uno script CGI che opera su disco	130,0	305,0
Passare i risultati al processo server	4,0	
Passare i risultati dal processo server alla connessione TCP/IP	4,0	

© Michele Colajanni, 2001

41

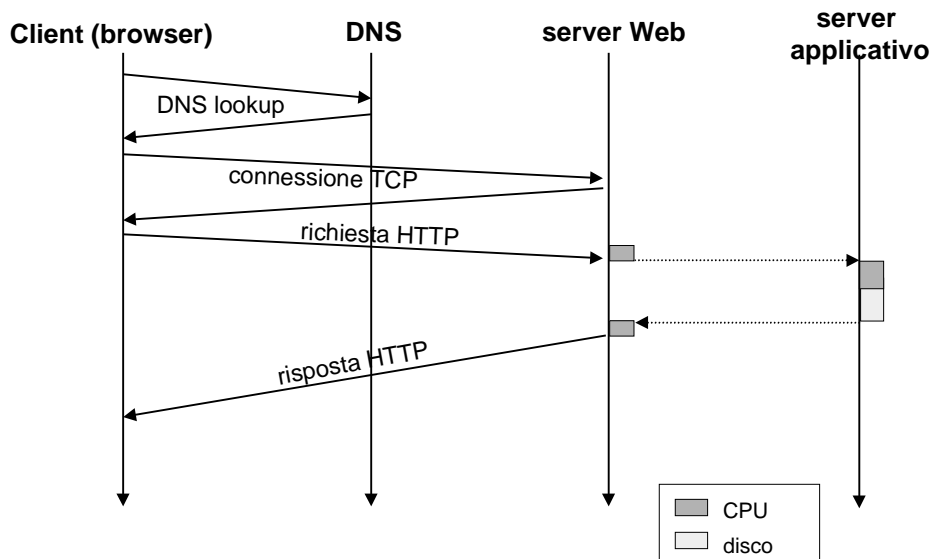
Richiesta di un oggetto statico



© Michele Colajanni, 2001

42

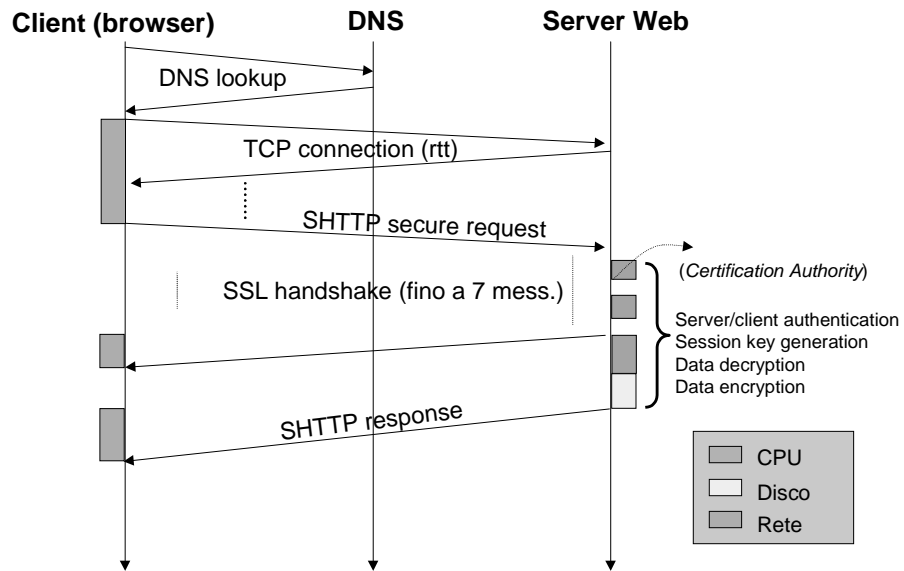
Richiesta di un oggetto dinamico



© Michele Colajanni, 2001

43

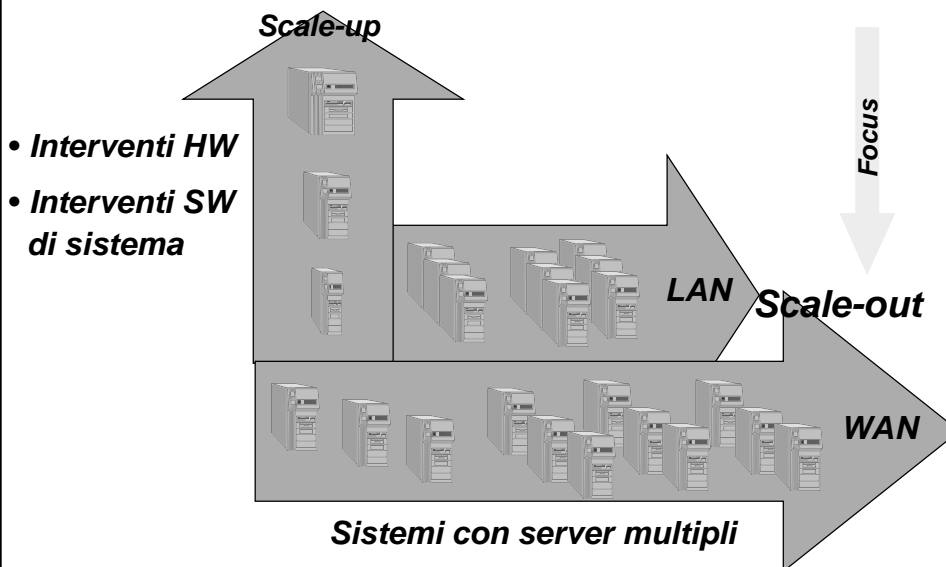
Richiesta "sicura" HTTP



© Michele Colajanni, 2001

44

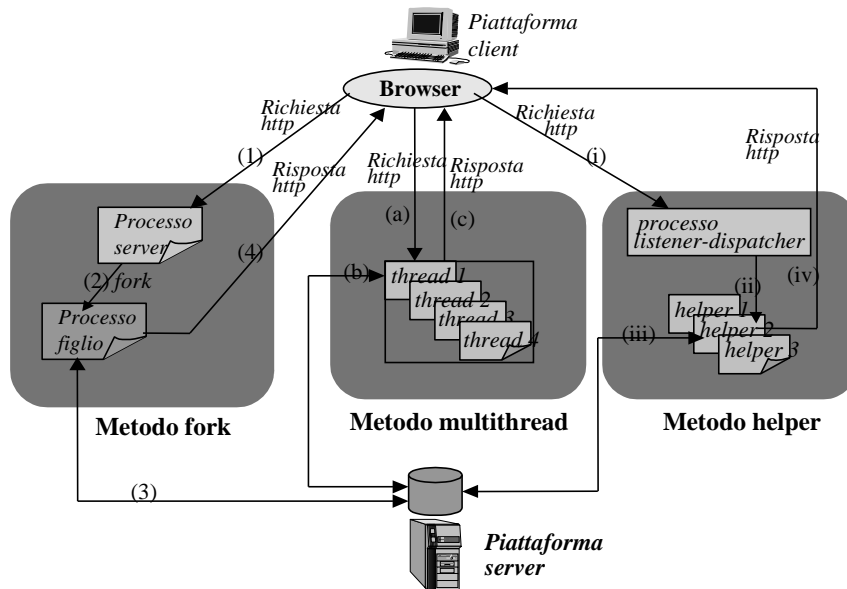
Ottimizzazioni dal lato server Web



© Michele Colajanni, 2001

45

Tre modalità di gestione *richieste HTTP*



© Michele Colajanni, 2001

46

Requisiti

Sistemi Web basati su piattaforme
con server multipli




- Un **meccanismo di distribuzione** (*dispatching*) per indirizzare le richieste client al “miglior” server Web
- Un **algoritmo di distribuzione** per individuare il “miglior” server Web
- Un componente **esecutore** per eseguire l’algoritmo di distribuzione utilizzando il relativo meccanismo

© Michele Colajanni, 2001

47

Meccanismi di distribuzione

Meccanismo	Esecutore	Oggetto	Basso
Risoluzione hostname - <i>Local dispatching</i> - Global dispatching (<i>Global cluster</i>)	DNS / DNS-like	Sessione utente	Livello di controllo 
Ridirezione HTTP - <i>Local dispatching</i> - Global dispatching (<i>Global cluster</i>)	Web server	Richiesta risorsa	
Ridirezione pacchetti - <i>Local dispatching</i> (<i>Web cluster</i>)	Web switch	Richiesta hit - risorsa (<i>HTTP/1.0 - HTTP/1.1</i>)	
			Elevato

© Michele Colajanni, 2001

48

Algoritmi di distribuzione

– **Statici** (*information-less*)

– **Dinamici**

- ◆ client info aware
- ◆ server state aware
- ◆ client info and server state aware

– **Adattativi** (non ancora investigati)

Nulla

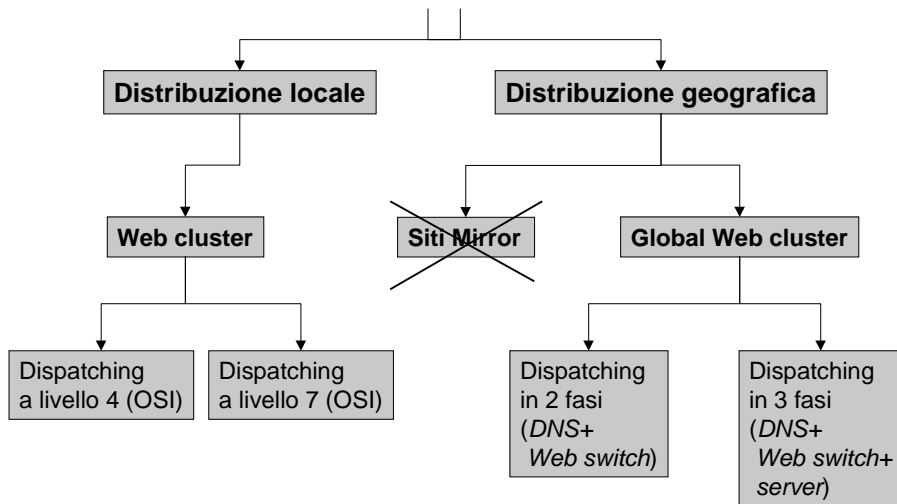


Elevato

© Michele Colajanni, 2001

49

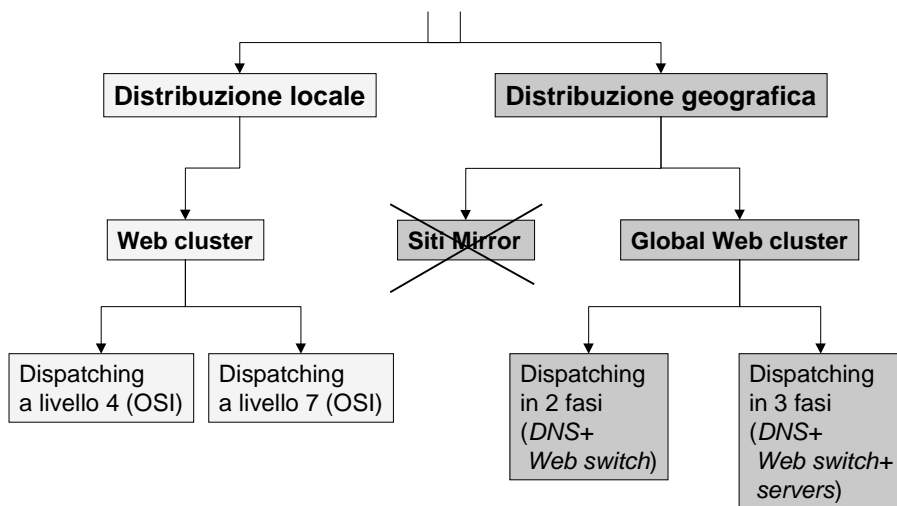
Sistemi con server Web multipli



© Michele Colajanni, 2001

50

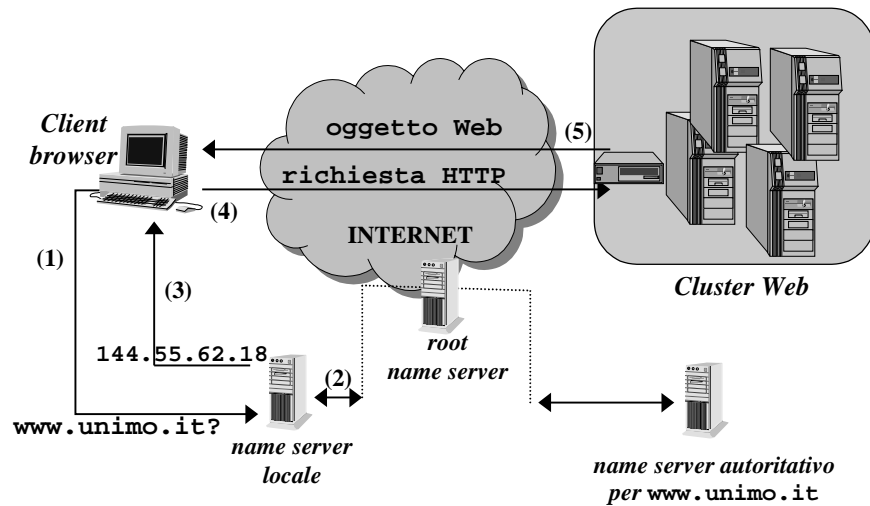
Sistemi con server Web multipli



© Michele Colajanni, 2001

51

Richiesta HTTP ad un cluster Web



© Michele Colajanni, 2001

52

Web cluster

- *Sito Web implementato su di un'architettura parallela o distribuita localmente.*
- Indirizzi sito Web
 - Un solo hostname (es., "www.unimo.it")
 - Un solo indirizzo IP (*virtual IP address*)
- Web server con indirizzi IP "mascherati"
- Un *Web switch (dispatcher)* il cui indirizzo IP è l'indirizzo IP del sito Web

© Michele Colajanni, 2001

53

Web switch del cluster

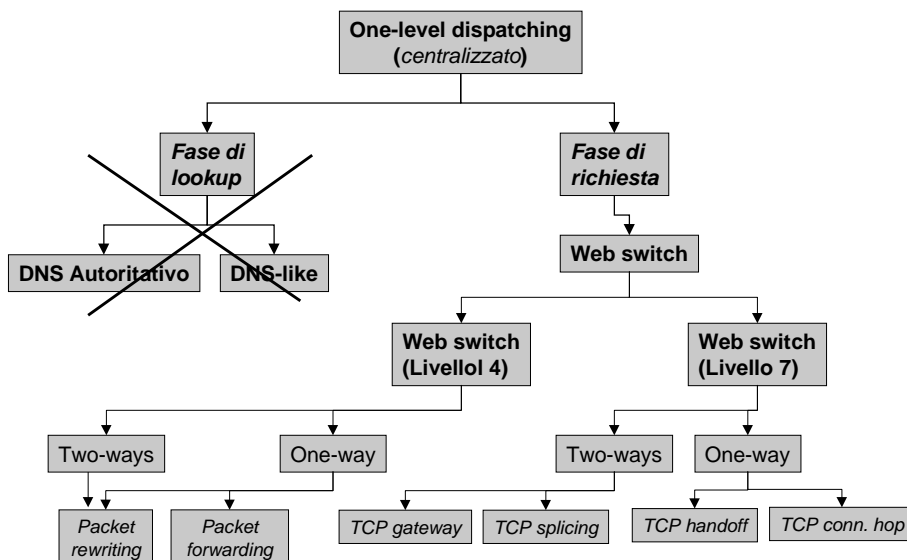
Componente di rete che agisce come dispatcher

- Mapping dal VIP agli indirizzi dei server
- Distribuzione delle richieste a granularità fine (i pacchetti entranti con indirizzo VIP sono indirizzati dal Web switch)
 - ◆ dispositivo hardware special-purpose
 - ◆ modulo software eseguito su SO general-purpose

Architetture alternative

- **Level-4 Web switch** (*Content information blind*)
 - ◆ sorgente e destinazione indirizzo IP, numeri di porta TCP, SYN/FIN bit nell'header TCP
- **Level-7 Web switch** (*Content information aware*)
 - ◆ contenuto URL, cookie, SSL id

Architetture per Web cluster

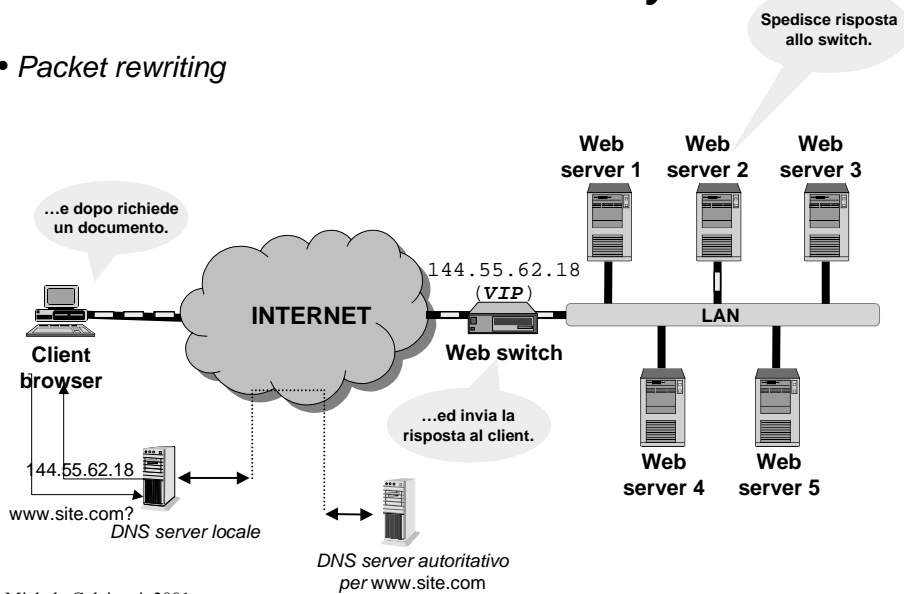


Web switch di livello-4

- Opera a livello TCP/IP
- TCP session management
 - Pacchetti appartenenti alla stessa connessione devono essere assegnati alla stessa macchina server
 - Il Web switch utilizza una **binding table** per la gestione delle connessioni attive
 - Il Web switch esamina l'header di ogni pacchetto:
 - ◆ nuova connessione (*SYN bit*) → *assegnamento server*
 - ◆ connessione esistente → *ricerca nella binding table*

Web Server Distribuiti Localmente : Architettura Two-Way

- *Packet rewriting*



Architetture Two-ways (L4)

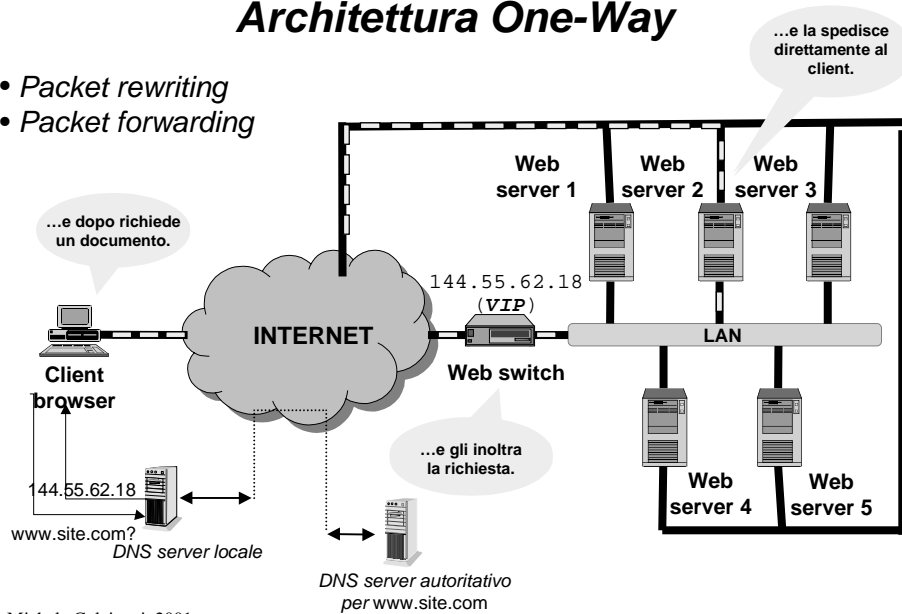
Ogni server ha il suo indirizzo IP privato

- I pacchetti in uscita devono riattraversare il Web switch
- Web switch modifica dinamicamente sia i pacchetti entranti sia quelli in uscita
 - ♦ Indirizzo IP *destinazione* dei pacchetti entranti (*VIP* → *IP server*)
 - ♦ Indirizzo IP *sorgente* dei pacchetti uscenti (*IP server* → *VIP*)
 - ♦ Ricalcolo dei checksum IP e TCP

Tecnica basata sul meccanismo di Network Address Translation (NAT) [Ege94]

Web Server Distribuiti Localmente : Architettura One-Way

- Packet rewriting
- Packet forwarding



Architetture One-way (L4)

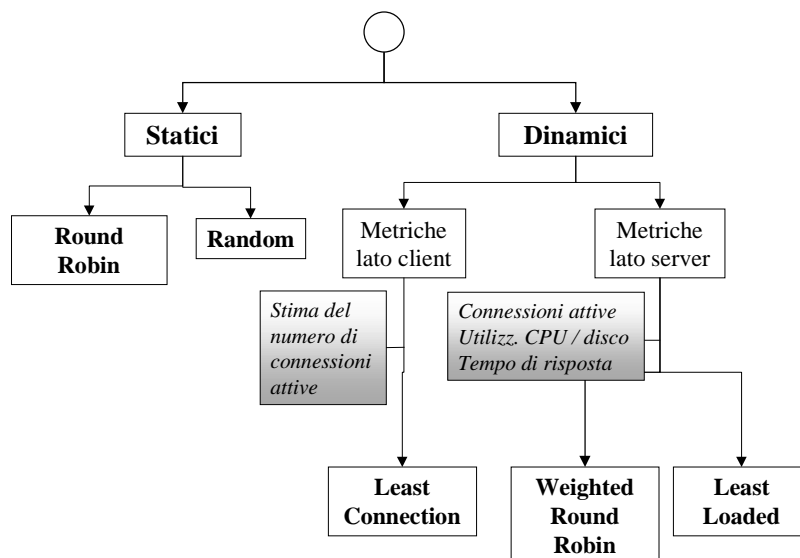
Packet rewriting

- Lo switch modifica solo i pacchetti IP entranti
- Il server modifica i pacchetti IP in uscita (IP server → VIP)

Packet forwarding

- Non c'è modifica dei pacchetti IP entranti ed uscenti. I pacchetti sono inoltrati a livello MAC (*ri-indirizzamento del frame MAC*)
- **PRO:** Minor overhead per pacchetto
- **CONTRO:** Web switch e server devono trovarsi sulla stessa sottorete fisica

Algoritmi di dispatching di Livello 4



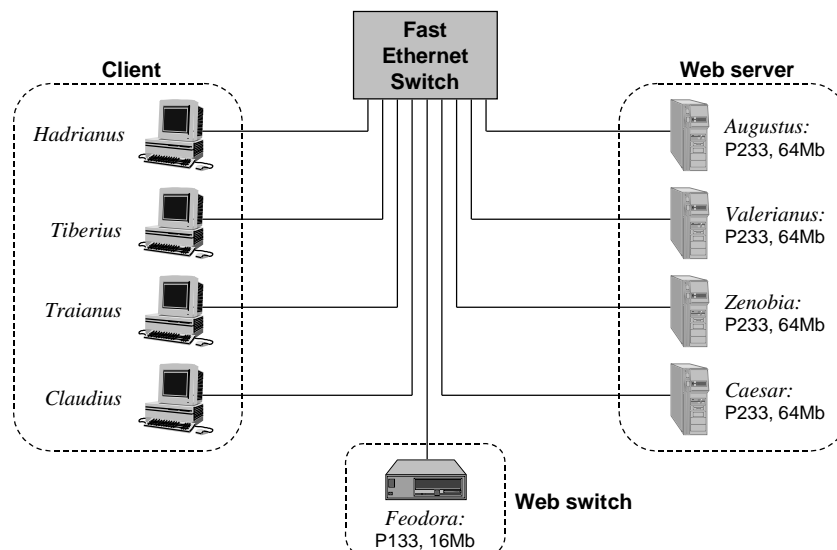
Dispatching server state aware

Weighted Round Robin

- distribuzione delle richieste in base allo stato di carico dei server
- possibili metriche per la stima dello stato di carico dei server
 - ◆ **Input metric:** informazioni di stato raccolte dal Web switch senza il coinvolgimento dei server
 - ◆ **Server metric:** informazioni di stato raccolte dai server (*CPU/Disk Utilization* o *Active Connections*)
 - ◆ **Forward metric:** informazioni di stato raccolte dal Web switch emulando delle richieste ai server

→ migliore

Architettura del prototipo “Roma4”



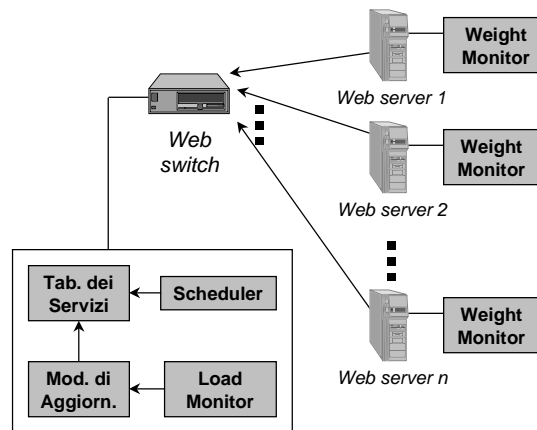
Componenti del sistema

Web switch

- Load Monitor
- Modulo di Aggiornamento
- Scheduler

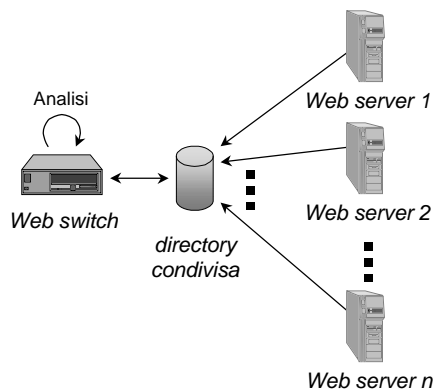
Web server

- Weight Monitor

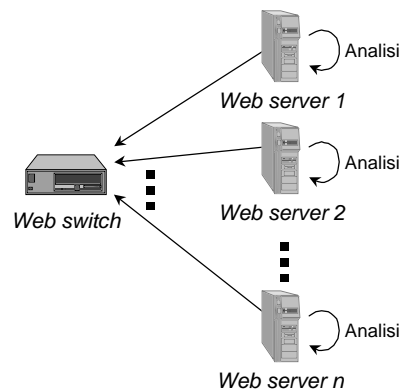


Meccanismi di comunicazione interni

Network File System

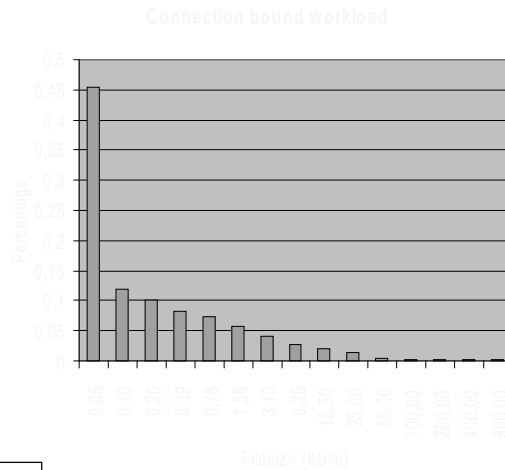


Socket



Workload: distribuzione dei file*

- Benchmark: **Webstone**
- Workload: **statico**
- Filesize distribution: **Pareto**
- Scenari:
 - *Connection bound*
 - *Network bound*



* Vedi Approfondimenti

© Michele Colajanni, 2001

66

Metriche di prestazione

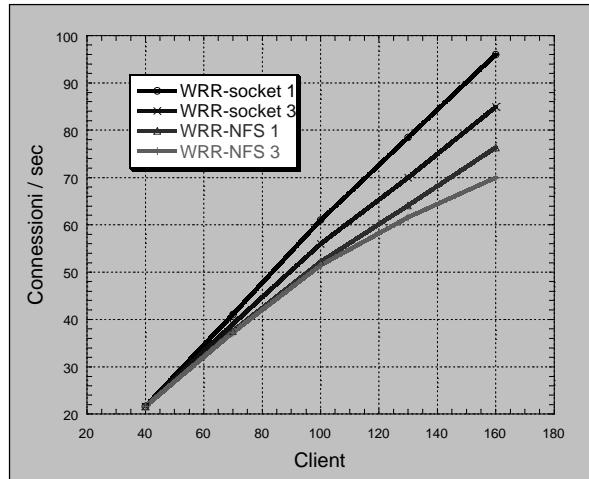
- **Tempo di risposta: tempo per completare tutta la richiesta o per ottenere il primo pacchetto di risposta**
Vista client (considerando ritardi Internet)
- **Tempo di latenza: tempo per completare la richiesta di un hit o di una risorsa**
Vista sistema Web (trascurando ritardi Internet)
- **Throughput: - quantità elaborate per unità di tempo**
 - ◆ numero di hit completati per unità di tempo (es., secondo)
 - ◆ numero di file serviti per secondo
 - ◆ numero di (K)bytes serviti per secondo
 - ◆ numero di connessioni al secondo (eventualmente, con rifiuti)
- **Utilizzazione**
 - ◆ sistema
 - ◆ componenti (CPU, dischi, memoria, interfaccia di rete)

© Michele Colajanni, 2001

67

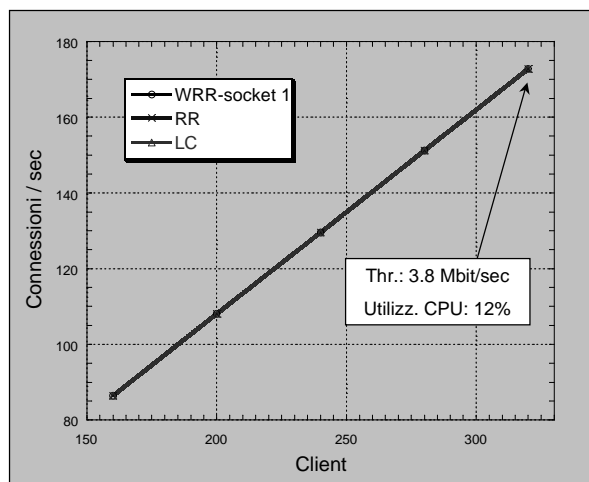
Confronto tra politiche dinamiche

- Le prestazioni peggiorano all'aumentare dell'intervallo di campionamento del carico sui server
- L'implementazione via socket è più efficiente dell'implementazione via NFS



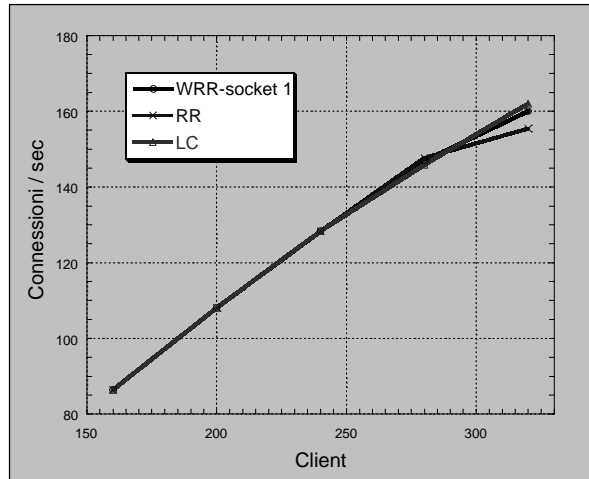
Risultati sperimentali: Workload statico

- 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 in condizioni di stress



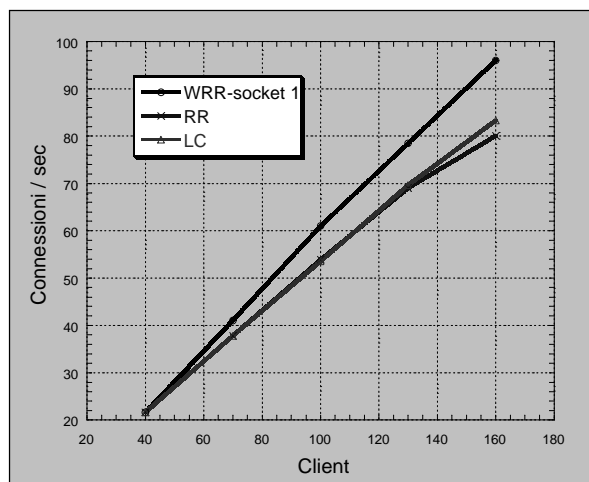
Risultati sperimentali: Workload dinamico “light”

- Le politiche statiche e dinamiche hanno un comportamento simile
- La migliore distribuzione del carico e l’overhead di comunicazione “server-switch” introdotto si compensano



Risultati sperimentali: Workload dinamico “heavy”

- La politica dinamica risulta nettamente più efficiente delle politiche statiche
- La migliore distribuzione del carico compensa l’overhead di comunicazione introdotto



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)
- Miglior bilanciamento del carico a parità di politica adottata

Dispatching per architetture Livello 4

- **Il Web switch controlla il 100% del traffico in arrivo al sito Web**
- **Non servono algoritmi particolarmente sofisticati**
- Gli algoritmi statici forniscono prestazioni confrontabili a quelle di algoritmi dinamici nel caso di richieste/servizi Web che rientrano in intervalli temporali di 2 ordini di grandezza
- Oltre i 2 ordini di grandezza, è opportuno utilizzare algoritmi state aware (server o client aware)
- Difficoltà nella scelta dei parametri ottimi per algoritmi server aware, per cui i client aware risultano migliori

Prototipi/prodotti Web cluster L4

Two-ways

Packet rewriting

- **Cisco's** LocalDirector [CisLD]
- **Magicrouter** [And96]
- **Foundry Networks'** ServerIron [Fou]
- **Alteon WebSystems** [Alt]
- **LSNAT** [Sri98]
- **Linux Virtual Server** [Lin]
- **F5 Networks** BIG/ip [F5]
- **HydraWeb Techs** [Hyd]
- **Coyote Point Systems'** Equalizer [Coy]
- **Radware's** WSD [Rad]

One-way

Packet rewriting

- **IBM** TCP router [Dia96]

Packet forwarding

- **IBM** Network Dispatcher [Hun98, IBMND]
- **ONE-IP** [Dam97]
- **LSMAC** [Gan00]
- **Foundry Networks'** ServerIron SwitchBack [Fou]