

**Tesi di Laurea**

**Meccanismi per la redirectione delle richieste in sistemi di server Web distribuiti geograficamente**

**Candidato:**

**Serena Ramovecchi**

**Relatore:**

**Prof. Salvatore Tucci**

**Correlatrice:**

**Ing. Valeria Cardellini**

**Sommario**

- Sviluppo di Internet, traffico e soluzioni
- Server Web distribuiti geograficamente
- Prototipo: progetto ed implementazione
- Testing
- Risultati sperimentali
- Conclusioni e sviluppi futuri

# Internet: crescita esponenziale

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
Luglio 2000	93.047.000
Gennaio 2001	109.574.000



## • Soluzioni:

- caching
- mirroring
- content distribution network
- architetture di server Web distribuiti



Fonte:  
Internet Software Consortium  
(<http://www.isc.org>)

# Server Web distribuiti geograficamente:

Event	Period	Peak day	Peak minute	Peak second
NCSA server (Oct. 1995)	-	2 Million	-	-
Olympic Summer Games (Aug. 1996)	192 Million (17 days)	8 Million	-	-
Nasa Pathfinder (July 1997)	942 Million (14 days)	40 Million	-	-
Olympic Winter Games (Feb. 1998)	634.7 Million (16 days)	55 Million	110,000	1,800
Wimbledon (July 1998)	-	-	145,000	2,400
FIFA World Cup (July 1998)	1,350 Million (84 days)	73 Million	209,000	3,500
Wimbledon (July 1999)	-	125 Million	430,000	7,200
Wimbledon (July 2000)	-	282 Million	964,000	16,000
Olympic Summer Games (Sept. 2000)	-	875 Million	1,200,000	20,000

# contatti

Throughput richiesto:  
20,000/sec \* 10KB  
=  
200MB/sec

## Server Web distribuiti geograficamente:

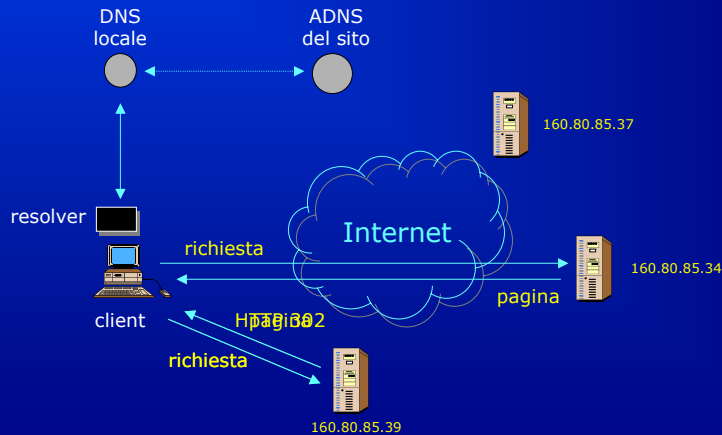
- più punti di accesso alle risorse
- distribuzione del carico
- fault tolerance

# Server Web distribuiti geograficamente:

## Distribuzione delle richieste

1° livello: DNS

2° livello: server Web



## Obiettivo del lavoro di tesi:

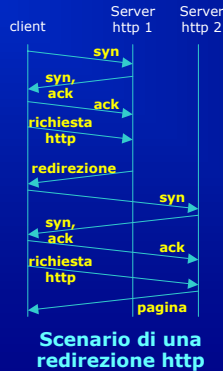
Realizzare un server Web  
in grado di effettuare la redirezione  
delle richieste  
in ambito geografico

## Progetto del prototipo:

### Processo di redirezione composto da 4 fasi:

- Attivazione
- Selezione
- Localizzazione
- Redirezione

**Scopo:** utilizzare la redirezione solo quando l'overhead introdotto può essere compensato da una diminuzione del  $T_{risposta}$



Overhead della redirezione

Attivazione basata su una soglia di carico

## Progetto del prototipo:

### Processo di redirezione in 4 fasi:

- Attivazione
- Selezione
- Localizzazione
- Redirezione

**Scopo:** redirigere solo le richieste t.c.  $T_{processamento}$  compensa l'overhead della redirezione

Politiche di selezione

statiche

dinamiche

# Progetto del prototipo:

## Processo di redirectione in 4 fasi:

- Attivazione

- Selezione

- Localizzazione

- Redirezione

Scopo: individuare il target server

Politiche di localizzazione

statiche

dinamiche

Scopo: effettuare la redirectione HTTP

## Progetto del prototipo: *attivazione*



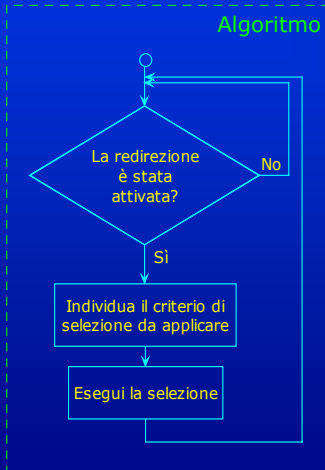
### Controlli

- la richiesta è già stata rediretta?
- l'utilizzazione del server supera la soglia?

### Funzionalità

- calcolo dell'utilizzazione del server
- attivazione della redirectione

## Progetto del prototipo: *selezione*



### Controlli

- la redirectione è stata attivata?

### Funzionalità

- effettuare la selezione:
  - random
  - in base alle proprietà del documento (dimensioni e numero di oggetti nascosti)
  - in base al tipo di documento

## Progetto del prototipo: *localizzazione*



### Controlli

- la redirectione è stata abilitata?

### Funzionalità

- effettuare la localizzazione:
  - random
  - round Robin

## Progetto del prototipo: *redirezione*



### Controlli

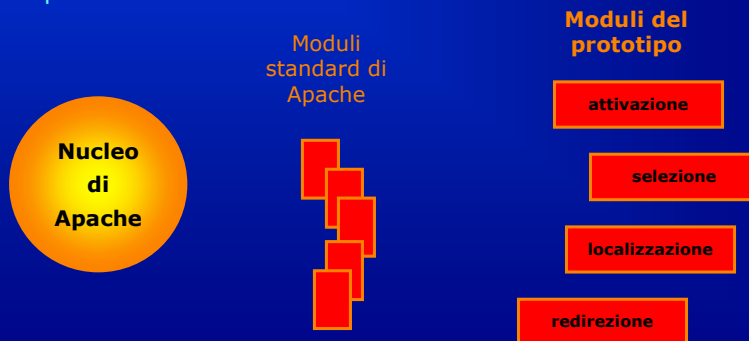
- la redirezione è stata abilitata?

### Funzionalità

- restituire al client un header HTTP contenente:
  - il codice HTTP 302
  - l'IP del target server

## Implementazione: *il server Web Apache*

- Diffusione: ad Agosto 2002 il 66% dei siti Web utilizzava Apache
- Disponibilità del codice sorgente
- Struttura modulare:
  - caricando solo i moduli necessari non si consumano risorse inutilmente
  - semplicità nell'estensione del codice







# Implementazione: *il prototipo*

## Moduli del prototipo

attivazione

selezione

localizzazione

redirezione

## Tool

monitor

parser

Parametri di configurazione

Politica di localizzazione

Modulo di localizzazione

Target server

Elenco server del sito

Redirezione abilitata dal Modulo di selezione

## Politiche stateless:

- Random
- Round Robin

## Target server:

`http://IPtargetServer/REDIRECTION_STRING/uri_risorsa`  
(es. `http://160.80.85.39/<redirected>/index.html`)

# Implementazione: *il prototipo*

## Moduli del prototipo

attivazione

selezione

localizzazione

redirezione

## Tool

monitor

parser

Target server dal Modulo di localizzazione

Modulo di redirezione

Header HTTP

## Header HTTP della risposta

### Status code:

**HTTP 302** (moved temporarily)

### Location:

stringa ottenuta dal modulo di localizzazione  
(es. `http://160.80.85.39/<redirected>/index.html`)

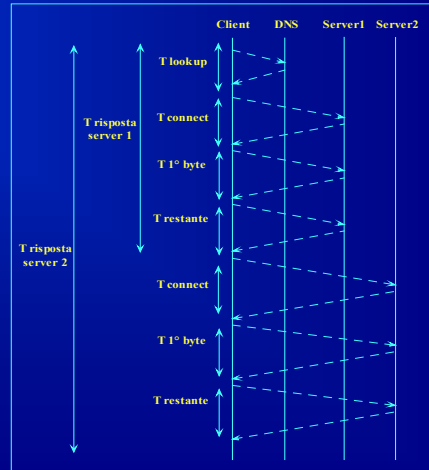
# Testing: *httpget\_test*

I benchmark esistenti  
(ad es. Webstone, httpperf e SPECweb)  
non supportano la redirectione

Sviluppo del tool  
**httpget\_test**

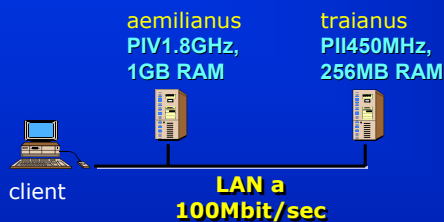
## httpget\_test:

- client HTTP
- interpreta il codice HTTP 302
- misura:
  - $T_{lookup}$
  - $T_{connessione}$
  - $T_{al\ 1^o\ byte}$
  - $T_{restante\ di\ trasferimento}$
  - $T_{risposta\ dal\ 1^o\ server}$
  - $T_{risposta\ dal\ 2^o\ server}$
- memorizza i risultati



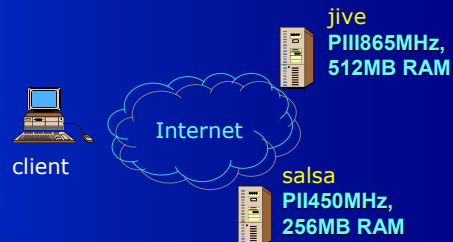
# Testing: *piattaforma*

## Sistema locale



**Scopo:** misurare l'efficacia della redirectione in funzione dell'utilizzazione del server

## Sistema remoto



**Scopo:** verificare i risultati in locale considerando l'impatto di rete, a parità di distanza client-server

# Testing: *scenari*

**Test:** diurni, notturni

**Utilizzazione del server:**

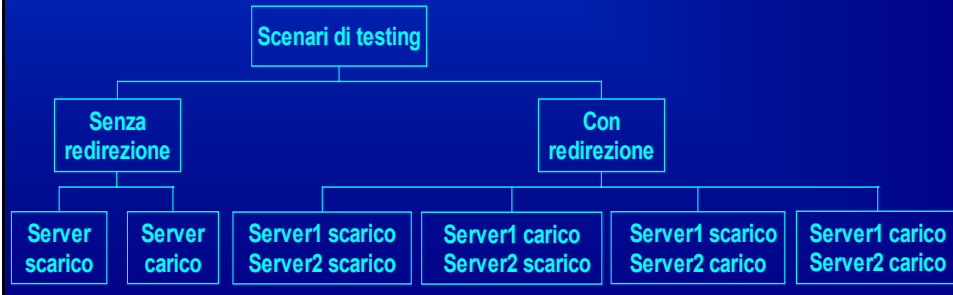
**File campione:**

- 1KB, 10KB, 100KB, 1MB

- scarico

- carico (50%, 70%, 90%)

## Classificazione degli scenari di testing



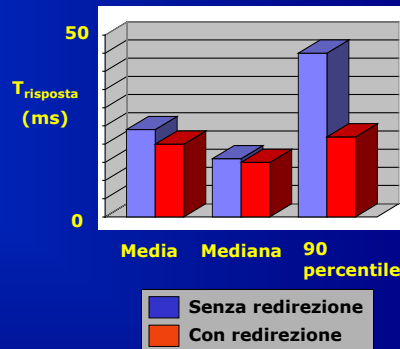
## Risultati sperimentali: *sistema locale*

**Aumento del  $T_{risposta}$ :**

- utilizzazione server1 scarico
- documenti 100KB-1MB
- inversamente proporzionale al carico del server2

Validità dell'attivazione della redirectione in base ad una soglia di carico  
Overhead della redirectione compensato dal  $T_{processamento}$  su server carico

Test diurno su file da 100KB  
Server1 carico al 90%  
Server2 scarico



## Risultati sperimentali: *sistema remoto*

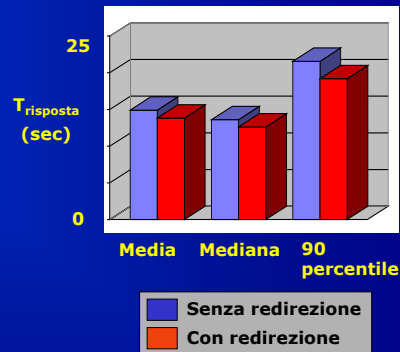
### Conferma dei risultati in locale:

l'impatto della rete non modifica i risultati ottenuti considerando la sola utilizzazione dei server



**Utilità della  
redirezione per  
l'incremento delle  
prestazioni**

Test notturno su file da 1MB  
Server1 carico al 90%  
Server2 scarico



## Conclusioni e sviluppi futuri

### Conclusioni

- Realizzazione di un prototipo di server Web in grado di supportare la redirezione delle richieste in ambito geografico
- Monitoraggio del livello di carico del sistema
- Parsing dei documenti del sito
- Sviluppo di un tool per il benchmarking del prototipo

### Sviluppi futuri

- Attivazione: soglia di carico dinamica
- Selezione: nuovi criteri
- Localizzazione: politiche basate sul carico e sulla prossimità
- Redirezione: Javascript e URL rewriting
- Utilizzo di piattaforme di testing più estese