

Web caching e sistemi per Web content delivery

Corso di Sistemi Distribuiti

Valeria Cardellini

Anno accademico 2008/09

Tipi di caching

- Caching lato client (*content consumer*)
 - I browser memorizzano su disco le risorse per accessi futuri
- Caching lato proxy (*terze parti*)
 - I proxy server memorizzano su disco le risorse più popolari
 - Gestiti da istituzioni o ISP
- Caching lato server (*content provider*)
 - Per ridurre il carico sulla piattaforma server
- Distribuzione dei contenuti e servizi (*terze parti*)
 - Gli edge server rappresentano dei nodi in una rete di livello applicativo per la distribuzione e consegna dei contenuti e servizi Web
 - Gestiti da società specializzate
- Per approfondimenti su caching e content delivery
<http://www.web-caching.com/>

Politiche di rimpiazzamento della cache

- Quando una risorsa viene richiesta dal proxy ad un server (origin server o altro proxy server) viene memorizzata in cache (se possibile) per sfruttare il principio di località
 - Cache del proxy su disco
- Se la cache è piena, occorre sostituire una o più risorse presenti in cache per fare spazio alla nuova risorsa
 - Occorre definire una politica di rimpiazzamento della cache (**cache replacement**)

Politiche di rimpiazzamento della cache (2)

- Differenza fondamentale rispetto alle politiche di rimpiazzamento studiate per la memoria cache o per la memoria virtuale in un calcolatore
- *Nel calcolatore*: caching di *parti* di risorse, parti aventi dimensioni fisse (blocchi o pagine)
- *Nel Web*: caching di *interi* risorse, caratterizzate da dimensioni largamente variabili (distribuzioni **heavy-tailed**)
 - Distribuzione heavy-tailed: la coda della distribuzione è più pesante della coda della distribuzione esponenziale
 - Grandissimi valori sono possibili con probabilità non nulla
 - Ad es., dimensioni delle risorse Web da 10^3 a 10^7 byte
 - Una variabile casuale X con $F(x)=P[X\leq x]$ è distribuita heavy-tailed (con indice di tail α) se:
$$1-F(x) = \text{Prob}[X>x] \sim c x^{-\alpha}, 0 < \alpha < 2$$

Politiche di rimpiazzamento della cache (3)

- Principali classi di politiche di rimpiazzamento della cache in ambito Web
 - 1) Derivate dal caching tradizionale
 - **LRU** e **LFU**
 - 2) Basate su una chiave primaria (tipicamente dimensione della risorsa)
 - Rimpiazzamento della risorsa più grande all'interno della cache
 - Varianti di LRU applicate ad un sottoinsieme di risorse
 - 3) Basate su più chiavi
 - 4) Basate sul costo
 - **Greedy-Dual-Size** e sue varianti
- Numerose proposte, ma in generale semplici strategie di sostituzione (come LRU) funzionano sufficientemente bene

Greedy-Dual-Size e Greedy-Dual*

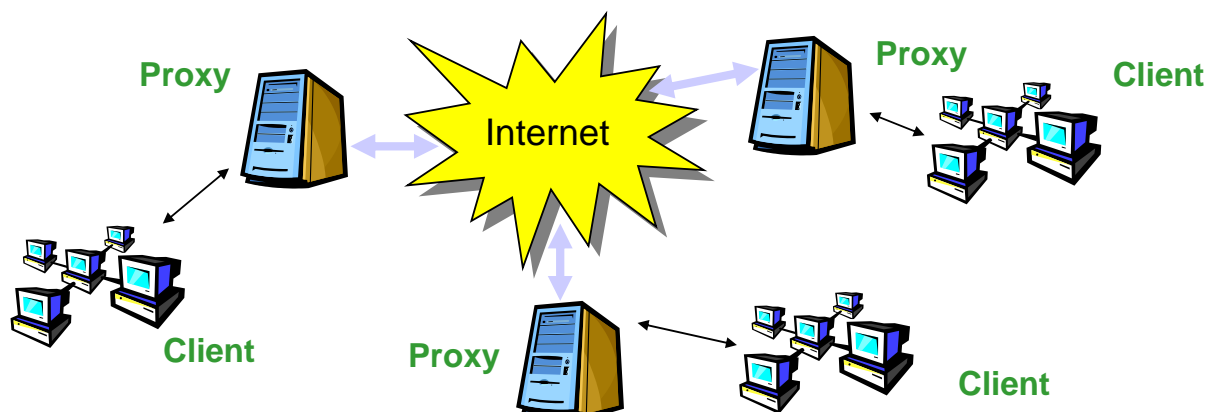
- Associa un costo ad ogni risorsa
- Rimpiazza la risorsa con il rapporto minimo tra costo e dimensione
 - Riferimento: P. Cao, S. Irani, "Cost-aware WWW proxy caching algorithms", Proc. of USENIX Symp. on Internet Technologies and Systems, 1997.
- Variante Greedy-Dual*
 - Sfrutta il fatto che risorse appartenenti alla stessa pagina hanno maggiore probabilità di essere richieste insieme
 - Se una qualunque risorsa nella pagina genera un cache hit, tutte le risorse della pagina sono mantenute in cache
 - Riferimento: S. Jin, A. Bestavros, "GreedyDual* Web caching algorithm: Exploiting the two sources of temporal locality in Web request", Computer Communications, 2001.

Metriche di prestazione per caching

- Per misurare l'efficacia del caching (sia delle politiche di rimpiazzamento della cache sia del caching cooperativo) si usano le seguenti metriche:
 - **Tasso di hit (cache hit rate)**
 - Percentuale di richieste servite dalla cache
 - **Tasso di byte hit**
 - Percentuale di byte serviti dalla cache
 - **Banda risparmiata**
 - Quantifica la diminuzione nel numero di byte richiesti agli origin server
 - **Riduzione della latenza**

Web caching cooperativo

- Diverse cache possono cooperare per ...
 - Migliorare le metriche di prestazione del caching
 - Distribuire meglio il carico tra più server
 - Aumentare la scalabilità
 - Aumentare l'affidabilità
- Ma la cooperazione introduce un overhead

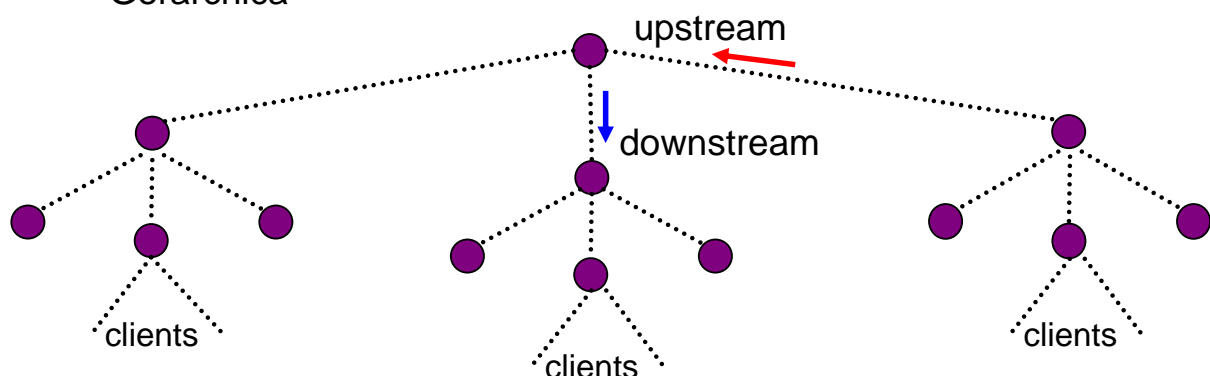


Modalità di cooperazione

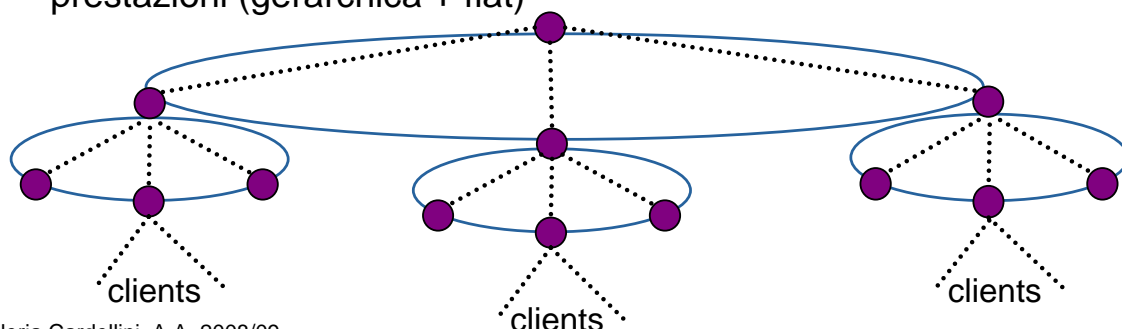
- A chi chiedere informazioni?
 - Modalità **gerarchica**: relazione **genitori-figli** verticale
 - Se un cache server figlio ha un cache miss chiede al cache server genitore
 - Modalità **flat**: relazione **paritetica** orizzontale
 - Se un cache server ha un cache miss chiede a tutti o a un sotto-insieme di cache di pari livello (*peer*)
- Quando chiedere informazioni?
 - Schema **query-based**: query al momento del cache miss
 - Schema **informed-based**: scambio periodico di informazioni tra i cache server indipendentemente dalla richiesta

Modalità gerarchica e flat

- Gerarchica

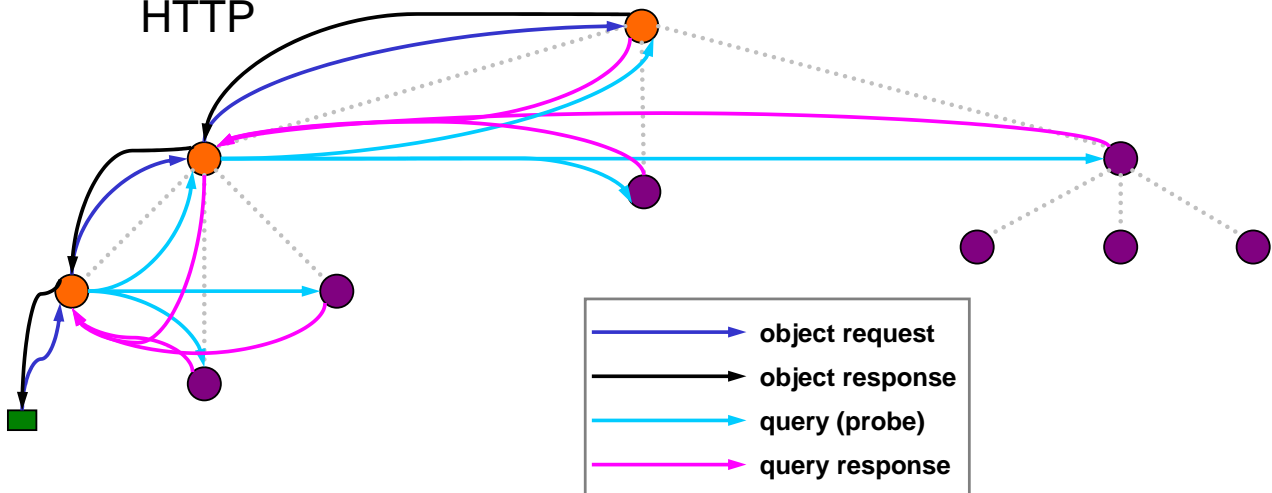


- Anche cooperazione tra cache di pari livello per aumentare le prestazioni (gerarchica + flat)



Internet Cache Protocol (ICP)

- Schema di tipo **query-based** basato su probe di tipo multicast tra sibling
- Si contatta il proxy che risponde per primo con un hit per ottenere la risorsa oppure si attende di ricevere tutte le risposte di miss (o la scadenza di un timeout)
- Messaggi ICP su UDP, recupero della risorsa su HTTP

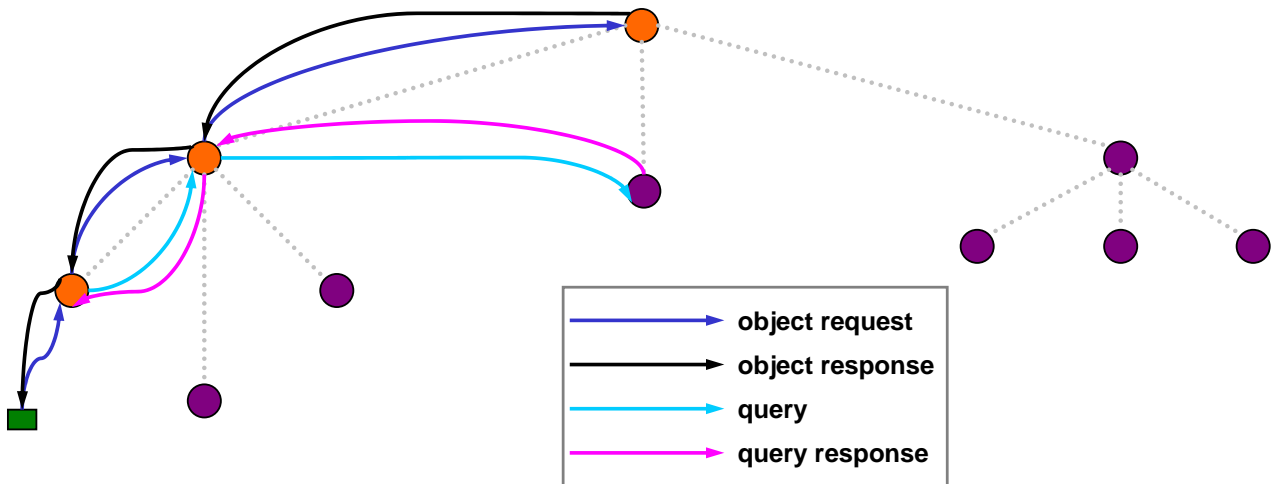


Cache Digests

- Schema di tipo **informed-based** basato sullo scambio tra i cache server di un **digest** del contenuto delle cache
- Ogni cache server replica la directory (**summary**) di ciascuno dei suoi peer
- Un peer viene contattato solo se il summary locale indica un hit per la risorsa cercata
- Summary basato su **filtri di Bloom** per ridurre l'overhead di memorizzazione
 - Ogni summary è di circa tre ordini di grandezza più piccola della cache corrispondente
 - **Filtro di Bloom**: struttura dati probabilistica molto compatta, usata per determinare se un elemento appartiene ad un insieme prestabilito di elementi
 - Usando un filtro di Bloom possono esserci falsi hit (circa 1%) ma non falsi miss (ovvero possibilità di falso positivo ma non di falso negativo)

Gerarchia con Summary e ICP

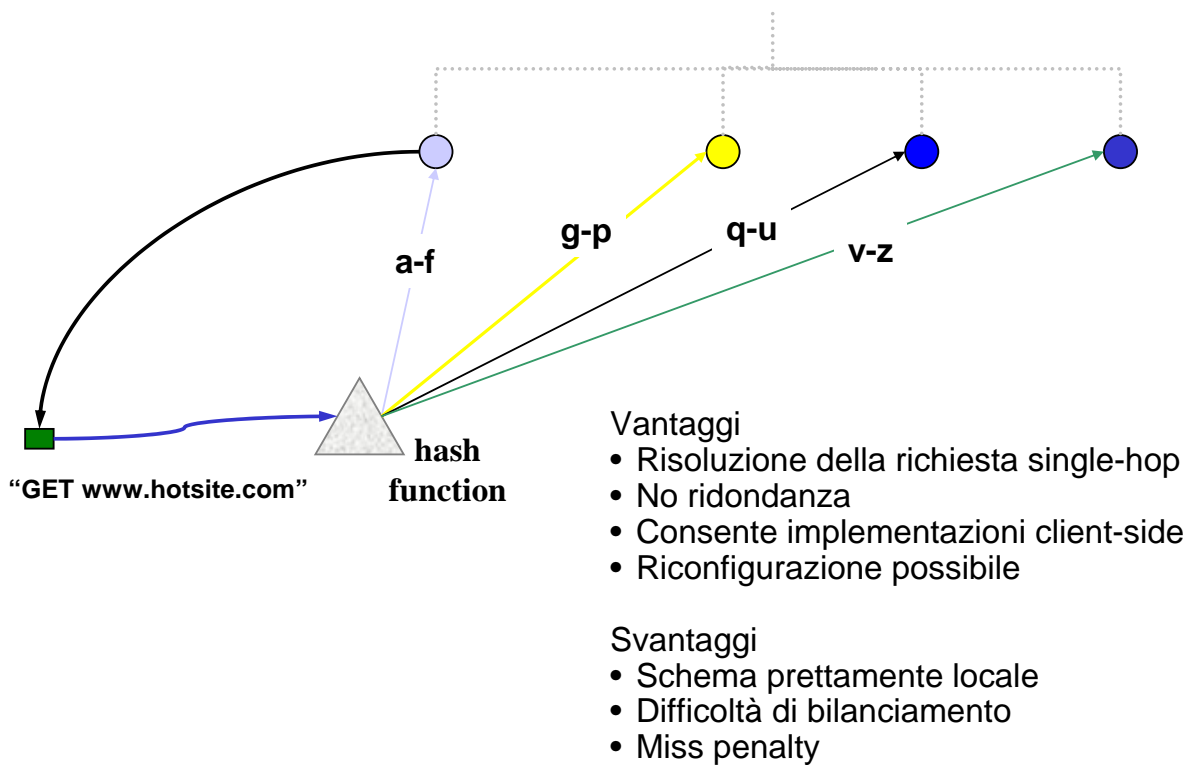
- L'uso di summary ad ogni livello della gerarchia permette di ridurre il numero di query tra sibling risultanti in miss fino al 95%



Scambio di informazioni

- **Schema query-based**
 - Minor rischio di informazioni incongruenti (*falsi hit* e *falsi miss*)
 - Maggiore latenza: occorre aspettare la risposta
 - Minore overhead di comunicazione (si scambiano informazioni solo quando serve)
- **Schema informed-based**
 - Maggior rischio di informazioni incongruenti
 - Minima latenza (si sa già dove andare)
 - Maggiore overhead di comunicazione (si scambiano informazioni periodicamente)
- Quindi: tradeoff tra overhead e incongruenza

Cache Array Routing Protocol (CARP)



SD - Valeria Cardellini, A.A. 2008/09

15

Squid: caratteristiche

- Squid è il server proxy più usato
<http://www.squid-cache.org/>
 - Progettato per sistemi Unix, disponibile anche per Windows
 - Free e open-source
 - Efficiente, stabile ed affidabile (processo di sviluppo open)
- Squid supporta
 - Protocolli applicativi HTTP, FTP, ...
 - Tunneling tra client e server dei protocolli SSL, HTTPS, TLS
 - Architetture distribuite e gerarchiche di proxy cooperativi
 - Protocolli di cooperazione: ICP, CARP, Cache Digests
 - Controllo di accesso
 - Autenticazione degli utenti
 - Caching trasparente
 - Funzionalità di reverse proxy
 - Caching delle risoluzioni DNS



SD - Valeria Cardellini, A.A. 2008/09

16

Squid: controllo di accesso

- Per decidere quali client sono abilitati ad usarlo come proxy, Squid utilizza una **Access Control List (ACL)**
 - Lista di indirizzi IP (a livello di dominio) per filtrare le richieste in ingresso, ad esempio:

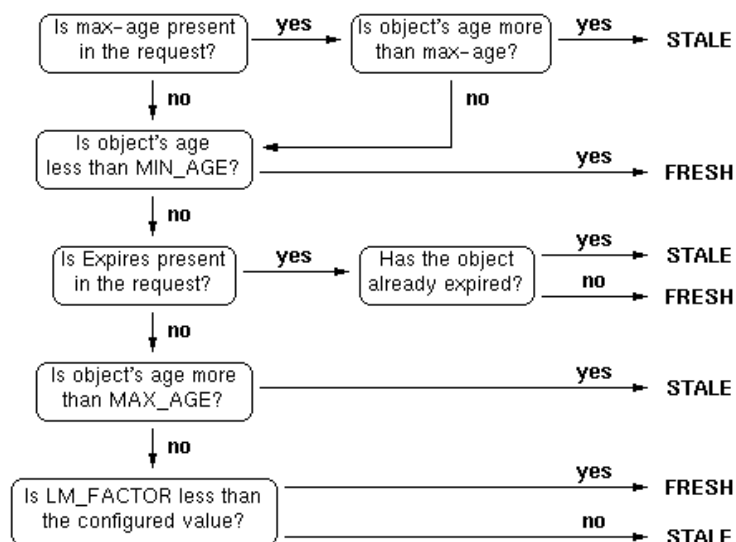
```
acl myclients src 172.16.5.0/24
http_access allow myclients
```

- Squid può anche essere configurato per disabilitare l'accesso a determinati siti Web

```
acl dumb dstdomain notbright.com nitwit.com
http_access deny dumb
```

Squid: gestione della cache

- Refreshment della cache
- LRU come politica di rimpiazzamento della cache



age = now - object date

lm_age = object date - last modified time

lm_factor = age / lm_age

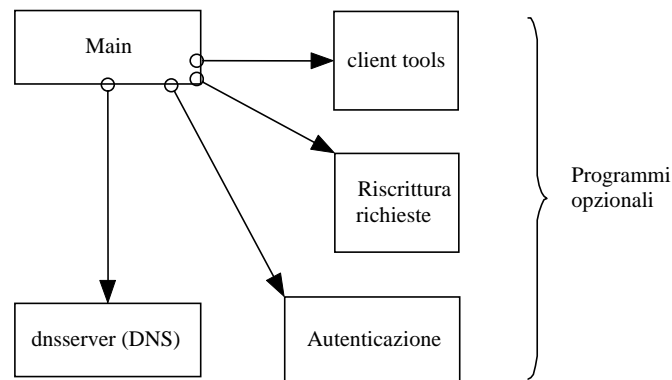
MIN_AGE, MAX_AGE, LM_FACTOR are configuration variables and depend on the object's URL as well

Last-Modified: optional header containing the last modification date of an object

Expires: optional expiry time specified by HTTP servers

max-age: optional directive which specifies the maximum allowed age of an object

Squid: architettura

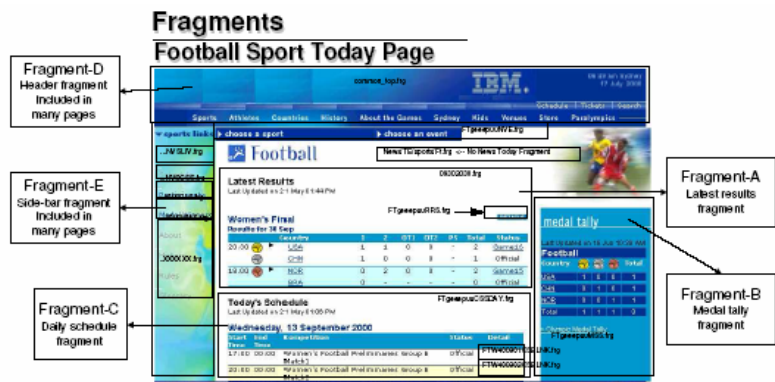


- Proxy a singolo processo
- I/O non bloccante
 - Uso di *callback function* (in C con puntatori a funzioni)
 - Per la risoluzione dei nomi in indirizzi IP non usa `gethostbyname()` ma un processo esterno (`dnsserver`)

Caching di contenuti dinamici

- Numerose proposte, tra cui
 - Active Cache: applet Java eseguiti sui cache server
 - Soluzioni integrate con tecnologie di database, es. Oracle 10g
 - **Frammentazione della pagina**
- Come mantenere la consistenza delle copie?
- Soluzione basata su frammentazione della pagina

I frammenti di cui è composta la pagina possono essere statici o dinamici, avere diverse caratteristiche di personalizzazione e di durata

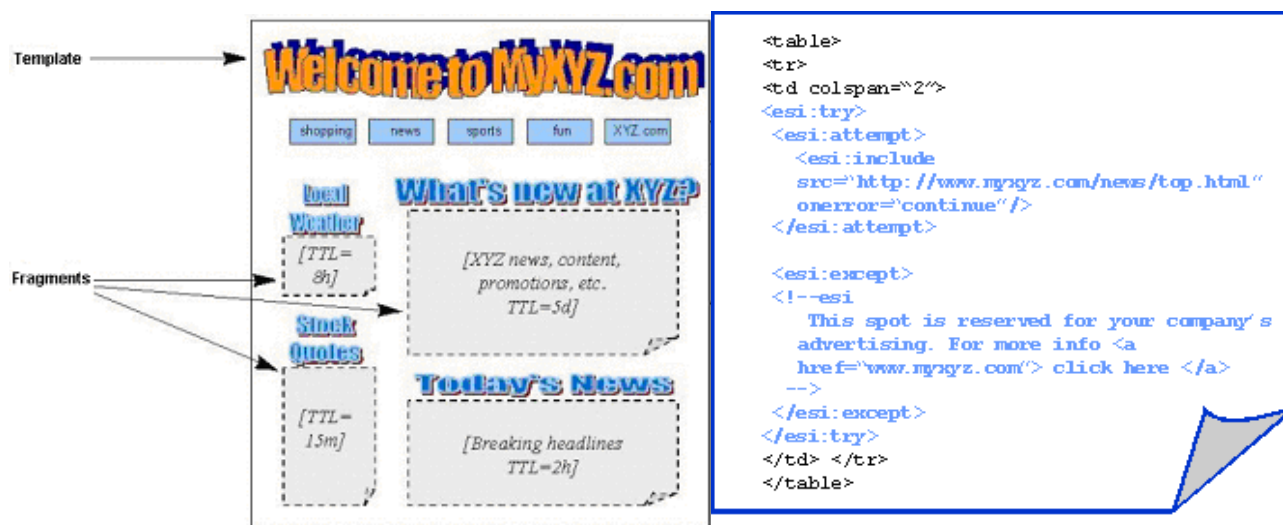


Fonte: A. Iyengar, L. Ramaswamy, B. Schroeder, "Techniques for Efficiently Serving and Caching Dynamic Web Content", Book chapter in *Web Content Delivery*, Springer, 2005.

Tecnologia ESI

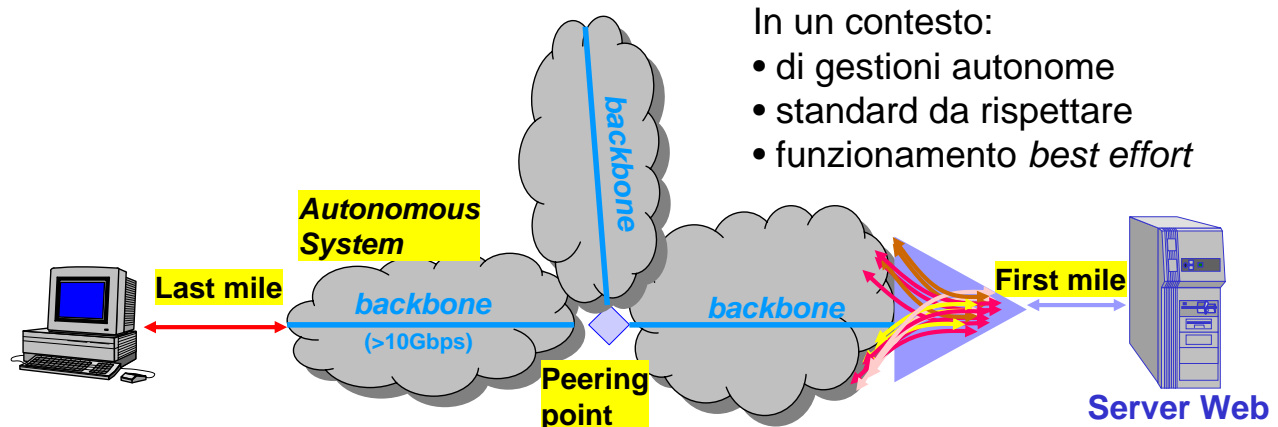
- Tecnologia standard **Edge Side Includes** (ESI) per la frammentazione della pagina
 - Riferimento: <http://www.esi.org/>
 - Obiettivo: separare le funzionalità di consegna del contenuto e generazione del contenuto per incrementare la scalabilità e ridurre i costi
- Linguaggio di markup basato su XML per descrivere componenti di una pagina Web in termini di frammenti ESI che sono cacheable e non-cacheable
 - Tag XML per specificare frammenti ESI in una pagina Web
- Per ciascun frammento ESI si può specificare un diverso requisito per il caching

Tecnologia ESI (2)



Il TTL permette di specificare fino a quando è valida la copia del frammento in cache

Colli di bottiglia del delivery



- First mile:
 - linea di collegamento tra il server Web ed il resto di Internet
- Peering point
 - punto di connessione tra diversi sistemi autonomi
- Backbone
- Last mile
 - collegamento del client al proprio ISP

Content delivery

- **Content delivery**: strategia per distribuire il contenuto mediante Internet su una base di **pay-per-delivery**
- Content Delivery Network o Content Distribution Network (CDN)
- Il content delivery genera profitti
- Il problema del network delivery
 - Più di 20000 Autonomous System (AS)
 - Nessuna rete gestisce più del 5% del traffico
 - La grande maggioranza degli AS gestisce meno dell'1% del traffico
- Soluzione al problema del network delivery
 - Distribuire i server in molti (quasi tutti) gli AS
 - L'infrastruttura della CDN può avere dimensioni impressionanti (ad es. Akamai)

Content Delivery Network

- Content Delivery Network (o Content Distribution Network)
 - E' formata da un'infrastruttura di server (detti *content server*, *delivery server* o *edge server*) che lavorano insieme
 - Gli edge server sono distribuiti su una vasta area geografica per permettere la fornitura dei contenuti e servizi Web da locazioni più vicine all'utente (edge = margini della rete)
 - Il content provider delega il servizio del contenuto del proprio sito Web (fornito dall'*origin server*) ad una compagnia che gestisce una CDN (*content outsourcing*)
 - Gli edge server forniscono soltanto il contenuto dei siti Web gestiti dalla CDN
 - Una CDN commerciale stabilisce con i suoi clienti degli accordi sul livello di servizio da soddisfare e si impegna a rispettare tali accordi (Service Level Agreement o SLA)

Web caching e CDN

- Web caching e CDN hanno in comune:
 - architettura costituita da un'infrastruttura di server che operano in modo cooperativo
 - server distribuiti su una vasta area geografica per permettere la distribuzione da locazioni vicine all'utente
- Per il resto differiscono sostanzialmente

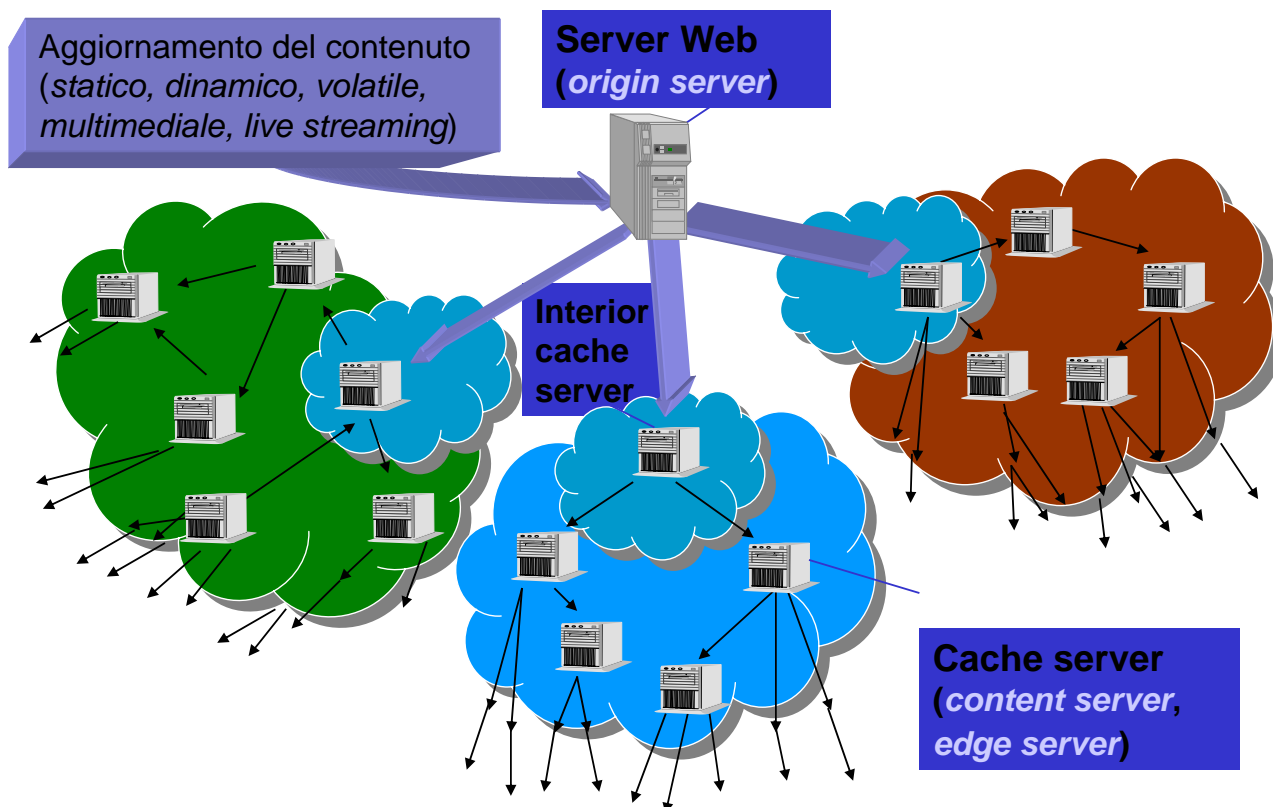
Web caching

- Opera in modo indipendente dai siti Web
- Costo per ISP (risparmio di banda), non per i content provider
- Caching generalizzato: memorizzato il contenuto di tutti i siti Web

CDN

- Il content provider delega la distribuzione alla CDN
- Costo per i content provider, non per ISP
- Caching selettivo: memorizzato solo il contenuto dei siti Web gestiti dalla CDN

Architettura di una CDN



Perché usare una CDN?

- Scopo simile al Web caching:
 - Avvicinare il contenuto agli utenti per ridurre il tempo di latenza ed il carico sull'origin server
- ... ma con garanzia sulle prestazioni ed altri vantaggi

Web caching

- cache riempite in seguito alle richieste degli utenti (*pull*)
- overhead sul sito Web per l'offloading ripetuto da parte di ogni proxy
- mancanza di controllo del contenuto nelle cache da parte del content provider
- perdita di molte informazioni sulla distribuzione del contenuto da parte del content provider

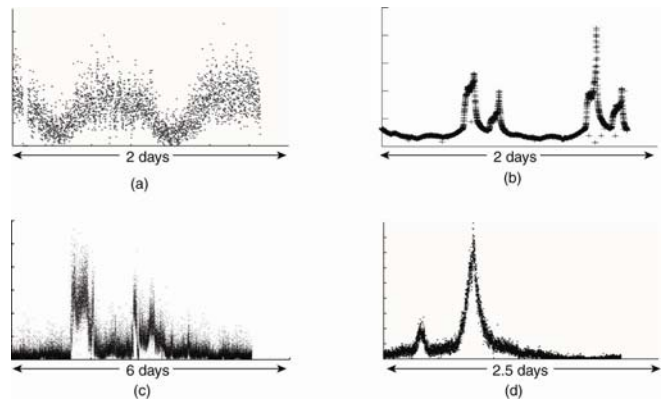
CDN

- contenuto memorizzato in anticipo (*prefetching*)
- overhead sul sito Web derivante dall'unico offloading
- controllo diretto del contenuto degli edge server da parte del content provider
- statistiche accurate sui pattern di accesso al sito Web per il content provider

Perché usare una CDN? (2)

- Perché una CDN e non un sistema Web distribuito localmente o geograficamente?
 - I Web cluster possono essere lontani dagli utenti e possono fallire a causa di congestioni di rete
 - Il content provider deve possedere l'infrastruttura di servizio
 - Le CDN possono avere una migliore reattività a sovraffollamenti improvviso (*flash crowd*) rispetto ai Web multi-cluster
- La scelta dipende anche dal tipo di servizio offerto dal sito

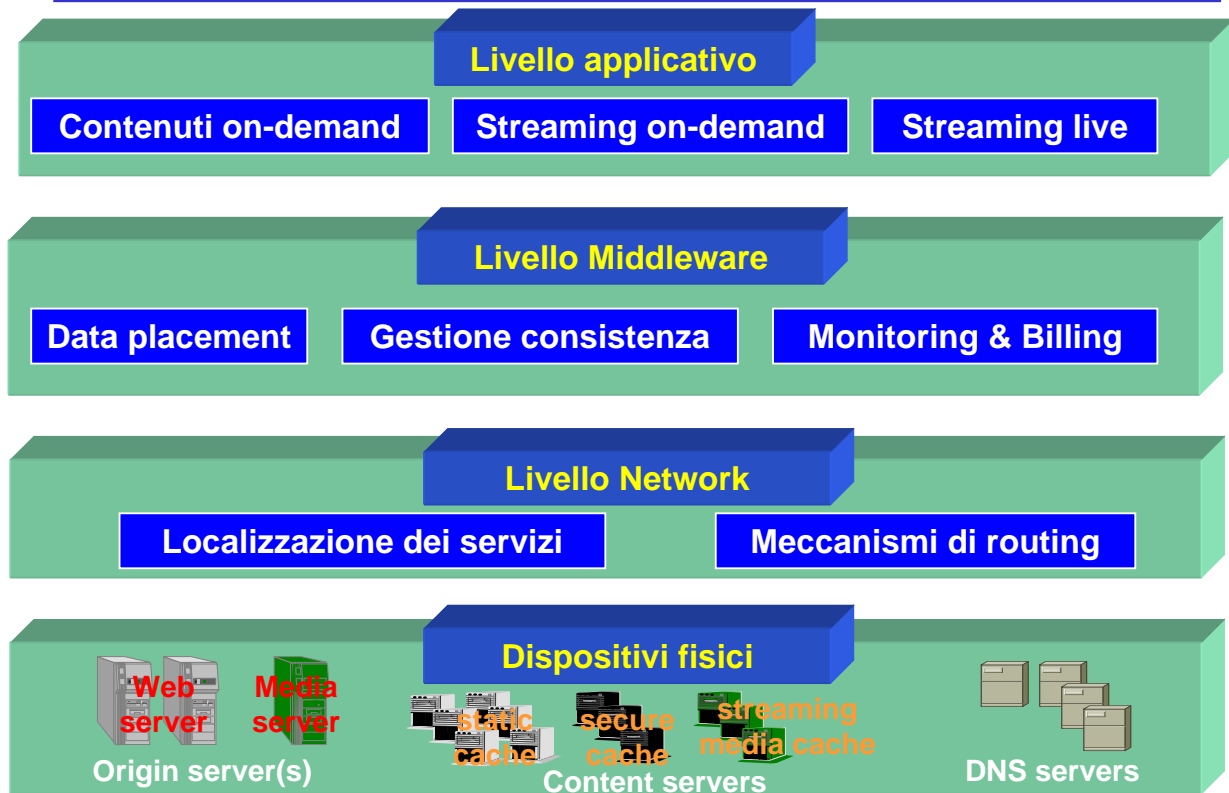
Un pattern di accesso normale e tre corrispondenti a flash crowd



Contenuti serviti da CDN

- Una CDN è generalmente utilizzata per fornire i seguenti servizi di delivery:
 - Distribuzione **contenuti non streaming**
 - Contenuto statico delle pagine Web, in particolare immagini
 - Contenuto dinamico
 - Per lo più contenuto “volatile” (contenuto statico che cambia frequentemente)
 - Contenuto con autenticazione
 - Autenticazione prevista dal protocollo HTTP
 - Contenuto sicuri
 - Protocollo HTTPS
 - Distribuzione **contenuti streaming on-demand**
 - Il contenuto è digitalizzato e memorizzato come media file su media server. Esempio: video-on-demand, clip musicali
 - Distribuzione **contenuti streaming live**
 - Il contenuto è distribuito quasi istantaneamente come media file. Esempio: eventi sportivi e musicali

Componenti dell'architettura di una CDN



Gestione della consistenza

- Come garantire che le copie delle risorse sugli edge server siano consistenti con la risorsa sull'origin server?
- Approcci possibili per gestire la consistenza guidati dall'origin server (di tipo *push*):
 - Aggiornamento periodico
 - Propagazione dell'aggiornamento
 - Aggiornamento on-demand
 - Invalidazione

Meccanismi di routing per CDN

- Quale meccanismo di routing con cui ridirigere verso un edge server una richiesta di un client per una risorsa servita dalla CDN?
- Principali tecniche adottate:
 - uso del Domain Name System: **redirezione DNS**
 - riscrittura URL con hostname simbolici: **URL rewriting**
 - riscrittura URL con indirizzo IP: **URL-IP rewriting**
- Altre possibilità
 - Redirezione mediante protocollo HTTP/RTSP
 - Anycast (IP o applicativo), IP tunneling
- Le società CDN usano nomi altisonanti (es. *Global Traffic Management system*), ma in realtà i principali dispositivi per il routing sono costituiti da **DNS autoritativi**, arricchiti con funzionalità di controllo sullo stato della rete e degli edge server

Redirezione DNS

- Il **DNS autoritativo** del sito Web delega la risoluzione dell'hostname nell'URL in un indirizzo IP ad un **DNS autoritativo controllato dalla CDN**
 - Nell'effettuare il mapping da hostname a indirizzo IP, il DNS della CDN sceglie un determinato edge server
 - Come nei sistemi Web distribuiti geograficamente
- Il valore del TTL assegnato dal DNS della CDN è prossimo a 0
 - Limitando il caching nei name server locali e intermedi, il DNS della CDN aumenta il controllo sul mapping tra hostname e indirizzo IP, facilitando così la condivisione del carico tra gli edge server
 - Per evitare che il DNS della CDN sia il collo di bottiglia, la CDN deve possedere un'infrastruttura DNS scalabile (es. Akamai)
 - Ci sono comunque DNS server **non cooperativi** (circa 17%), che violano il TTL stabilito dal DNS della CDN

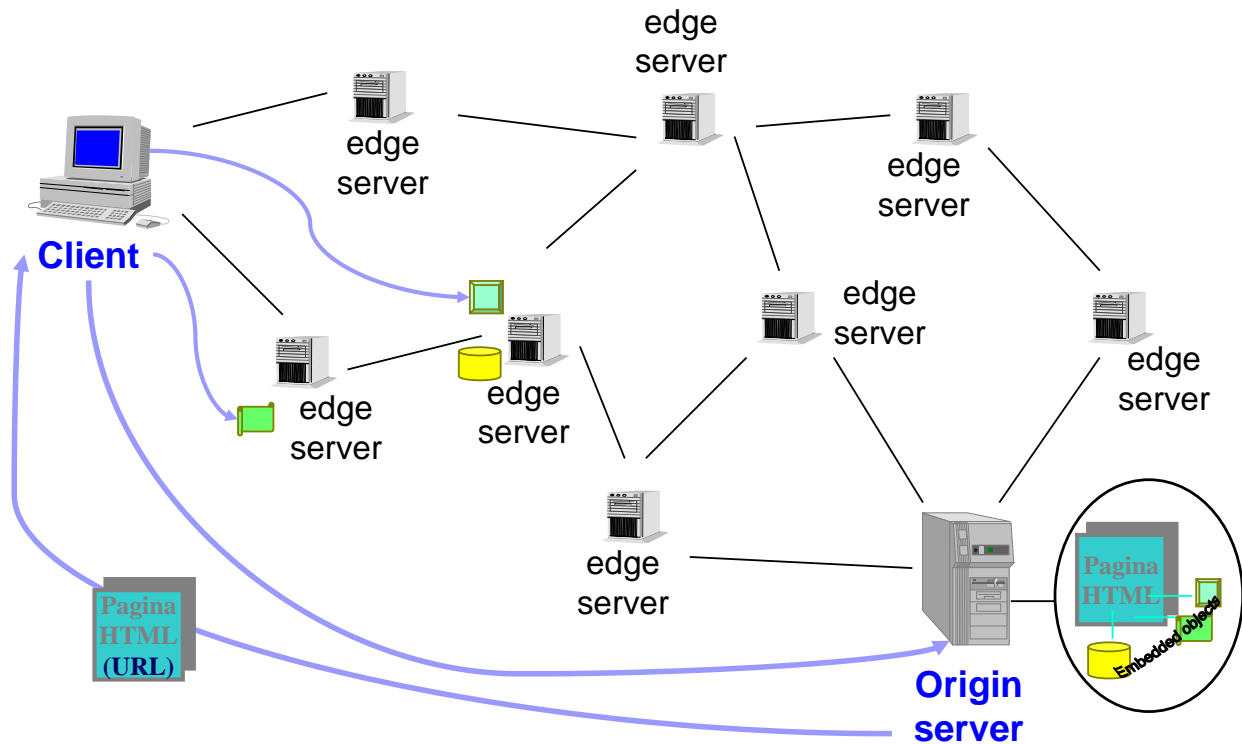
Redirezione DNS (2)

- Due tipologie di CDN che usano la redirezione DNS:
 - **full-content delivery** (o *first hit at CDN*)
 - L'origin server è nascosto a tutti, eccetto che alla CDN
 - Il DNS autoritativo dell'origin server è modificato in modo da delegare le richieste di risoluzione al DNS server autoritativo della CDN (**DNS outsourcing**)
 - Tutte le richieste indirizzate all'origin server arrivano agli edge server
 - **partial-content delivery** (o *first hit at origin*)
 - L'origin server modifica nella pagina (X)HTML l'URL delle risorse incluse nella pagina (immagini, video, audio, ...) applicando la tecnica dell'**URL rewriting**
 - Esempio: <http://www.foo.com/bar.gif> viene modificato in <http://cdn-foo.net/www.foo.com/bar.gif>
 - L'hostname nell'URL modificato è risolto in un indirizzo IP dal DNS server autoritativo della CDN

URL rewriting

- L'origin server riscrive dinamicamente gli URL nella pagina (X)HTML in modo da ridirigere le richieste dei client per le risorse incluse (embedded) verso un edge server della CDN
 - L'edge server più vicino, il meno carico, una combinazione di entrambe le metriche, ...
- Due alternative di **rewriting**
 - URL riscritta **con indirizzo IP**: indirizzamento **diretto** dell'edge server
 - Soluzione meno usata
 - URL riscritta **con hostname**: indirizzamento **indiretto** dell'edge server
 - Nella successiva fase di risoluzione del nuovo hostname, il DNS server della CDN seleziona l'edge server "migliore"

Routing mediante URL rewriting



Diffusione delle CDN

- Alcune CDN commerciali:
 - Akamai <http://www.akamai.com>
 - Mirror Image <http://www.mirror-image.com>
 - EdgeStream <http://www.edgestream.com>
 - Limelight Networks <http://www.limelightnetworks.com/>
- Alcune CDN accademiche:
 - Caratterizzate dall'uso di tecnologie P2P
 - Generalmente limitate al delivery di contenuti statici
 - CoDeeN, Coral, Globule
- Akamai è la società leader
 - Delivery del 10-20% del traffico Web (fino a 650 Gbps)
 - Akamai EdgePlatform composta da oltre 34000 edge server in 70 paesi

Partial content delivery in Akamai

- L'origin server che vuole delegare il servizio di parte delle proprie risorse all'infrastruttura di edge server gestita da Akamai deve rinominare gli URL ad esse relative usando un prefisso specifico
 - Il prefisso include un hostname, ad esempio a799.g.akamai.net
- Risoluzione dell'hostname nell'indirizzo IP di un edge server di Akamai gestita dai server DNS di Akamai
- L'edge server scelto è “vicino” al DNS server locale del client e non sovraccarico
 - L'edge server non è un singolo server fisico ma un Web cluster
 - L'algoritmo di selezione (proprietario) usa una combinazione di informazioni su: topologia di rete (BGP, protocollo di routing interdomain), prossimità dinamica (traceroute) e stato di carico degli edge server
 - Problema per la prossimità: non sempre il DNS server locale è “vicino” al client

Partial content delivery in Akamai (2)

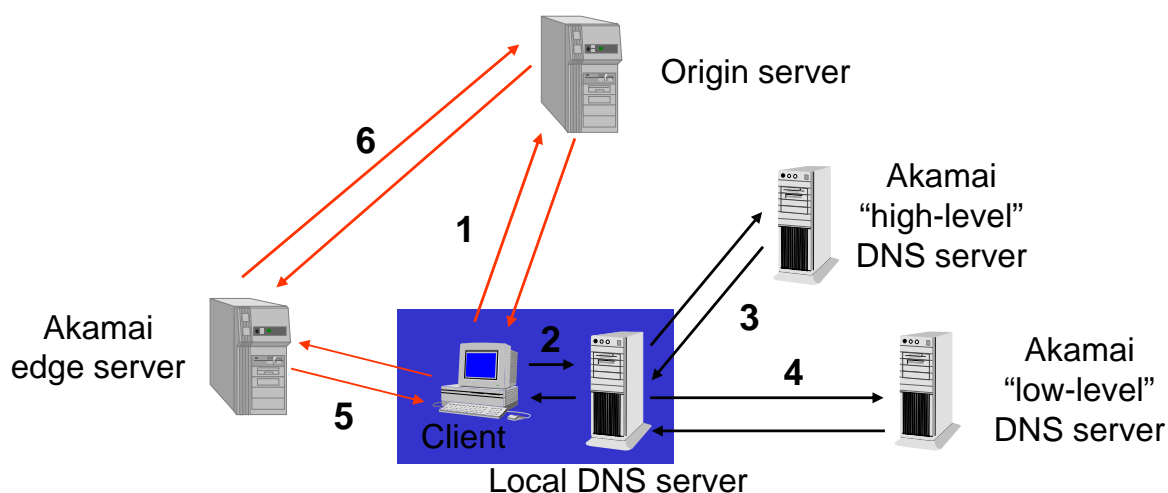
- Esempio:
 - Il client richiede all'origin server la pagina Web <http://www.xyz.com/> che contiene [image.gif](http://www.xyz.com/image.gif) come risorsa inclusa
 - L'origin server fornisce in risposta la risorsa (X)HTML, in cui l'URL della risorsa <http://www.xyz.com/image.gif> è riscritta come <http://a799.g.akamai.net/3/799/388/9fd0a26b9d5686/www.xyz.com/image.gif>
 - Il prefisso a799.g.akamai.net individua un edge server gestito da Akamai
 - Tramite DNS, il client risolve a799.g.akamai.net in un indirizzo IP, ad esempio 193.206.138.7 (distante solo 5 hop dal client, anziché più di 17 come l'origin server)
 - Il client invia la richiesta HTTP per [image.gif](http://www.xyz.com/image.gif) all'edge server

Partial content delivery in Akamai (3)

- Esempio (*continua*):
 - Se l'edge server 193.206.138.7 possiede una copia consistente della risorsa image.gif, la invia al client
 - Nell'URL riscritta vi è un **identificatore univoco** che viene modificato dall'origin server ogni volta che la risorsa cambia
 - Altrimenti, richiede la risorsa tramite un protocollo interno ad un altro edge server di Akamai oppure all'origin server www.xyz.com e la memorizza nella propria cache
 - Un altro client appartenente ad un'altra rete che richiede la stessa pagina avrà un URL uguale per l'immagine
<http://a799.g.akamai.net/3/799/388/9fd0a26b9d5686/www.xyz.com/image.gif> ma l'indirizzo IP dell'edge server scelto potrebbe essere diverso

Infrastruttura DNS di Akamai

- Akamai utilizza una gerarchia di name server (**doppio livello**) per risolvere l'hostname in un indirizzo IP degli edge server
 - Esempio (dopo la prima risoluzione del DNS)



Esempio: immagini di PCWorld

- Immagini dell'home page di `pcworld.com` servite da Akamai
- URL delle immagini con hostname `images.pcworld.com`, ad es.
`http://images.pcworld.com/images/common/feed-icon-14x14.png`
- Interrogando il DNS per risolvere l'hostname `images.pcworld.com` con il comando `dig`

```
[valeria@tiberius ~]# dig images.pcworld.com
```

```
; <<>> DiG 9.2.5 <<>> images.pcworld.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64887
;; flags:qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 9, ADDITIONAL: 9

;; QUESTION SECTION:
;images.pcworld.com.                IN      A
```

Esempio: immagini di PCWorld (2)

```
;; ANSWER SECTION:
images.pcworld.com. 853 IN CNAME
images.pcworld.com.edgesuite.net.
images.pcworld.com.edgesuite.net. 20580 IN CNAME
a1694.g.akamai.net.
a1694.g.akamai.net. 20 IN A 213.254.17.8
a1694.g.akamai.net. 20 IN A 213.254.17.25

;; AUTHORITY SECTION:
g.akamai.net. 1441 IN NS n5g.akamai.net.
g.akamai.net. 1441 IN NS n6g.akamai.net.
g.akamai.net. 1441 IN NS n7g.akamai.net.
g.akamai.net. 1441 IN NS n8g.akamai.net.
g.akamai.net. 1441 IN NS n0g.akamai.net.
g.akamai.net. 1441 IN NS n1g.akamai.net.
g.akamai.net. 1441 IN NS n2g.akamai.net.
g.akamai.net. 1441 IN NS n3g.akamai.net.
g.akamai.net. 1441 IN NS n4g.akamai.net.
```

TTL=20 sec

Akamai edge server

Akamai low-level DNS server

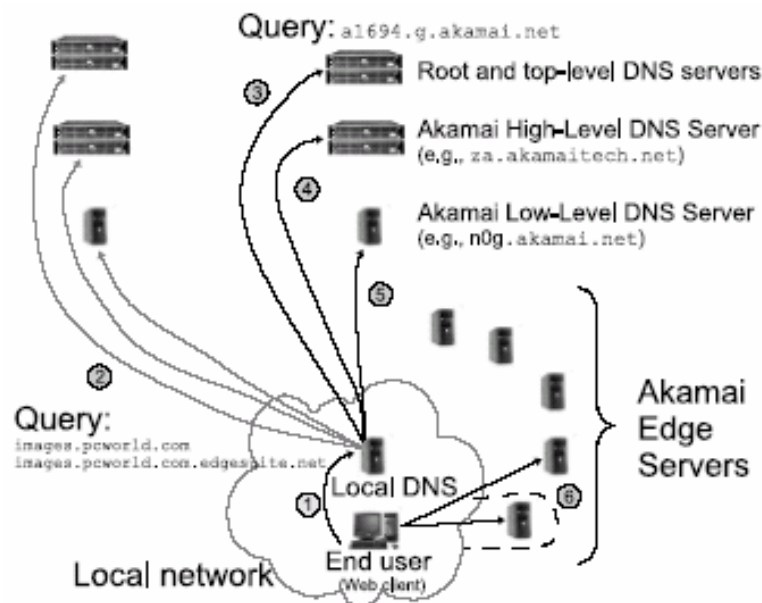
Esempio: immagini di PCWorld (3)

```
;; ADDITIONAL SECTION:
n0g.akamai.net.      1213    IN      A       195.176.255.147
n1g.akamai.net.      308     IN      A       213.254.17.29
n2g.akamai.net.      1252    IN      A       80.15.236.163
n3g.akamai.net.      1213    IN      A       213.254.17.29
n4g.akamai.net.      308     IN      A       195.176.255.147
n5g.akamai.net.      1252    IN      A       213.254.17.26
n6g.akamai.net.      1213    IN      A       213.254.17.20
n7g.akamai.net.      308     IN      A       213.254.17.20
n8g.akamai.net.      1213    IN      A       195.176.255.147

;; Query time: 33 msec
;; SERVER: 160.80.1.8#53(160.80.1.8)
;; WHEN: Fri Nov 7 15:17:46 2008
;; MSG SIZE rcvd: 449
```

- Con il comando `dig a1694.g.akamai.net +trace` si individuano i name server di Akamai di alto e basso livello

Esempio: immagini di PCWorld (4)



Tratto da: A. Su *et al.*, "Drafting behind Akamai (travelocity-based detouring)" *SIGCOMM Comput. Commun. Rev.* 36(4), Aug. 2006.

Un esempio: immagini di PCWorld (5)

- Dove sono localizzati gli edge server con indirizzo IP 213.254.17.8 e 213.254.17.25?
- Usando un servizio di traceroute, si scopre che sono distanti 8 hop da tiberius.ce.uniroma2.it e che entrambi gli edge server sono in Italia
- Usando un client in un altro ISP, oppure risolvendo il nome in un altro momento, è possibile ottenere un risultato diverso per il mapping tra l'hostname images.pcworld.com e l'indirizzo IP di un edge server

Content peering

- Cooperazione ed interoperabilità tra più CDN allo scopo di migliorare la scalabilità, la tolleranza ai guasti e le prestazioni
- Le soluzioni proposte per il content peering adottano il routing tra CDN effettuato a livello di DNS
 - Il server DNS (*potenziato*) deve essere in grado di fornire l'indirizzo di una CDN che serve una regione geografica
 - L'intera CDN agisce come un edge server, recuperando il contenuto e tariffando il costo dell'operazione alla CDN richiedente