Web caching
e Content Delivery Networks

Corso di ***Applicazioni Telematiche***

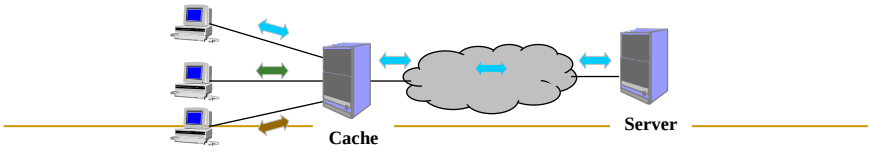
A.A. 2006-07 – Lezione n.4

Prof. Roberto Canonico

Università degli Studi di Napoli Federico II
Facoltà di Ingegneria

Web caching

- Si parla genericamente di Web caching quando le richieste di un determinato client non raggiungono il Web Server ma vengono intercettate da una proxy cache
- Tipicamente, un certo numero di client di una stessa rete condividono una stessa cache web, posta nelle loro prossimità (es. nella stessa LAN)
- Se l'oggetto richiesto non è presente nella cache, questa lo richiede *in vece* del client conservandone una copia per eventuali richieste successive
- Richieste successive alla prima sono servite più rapidamente
- Due tipi di interazione HTTP: client-cache e cache-server



Caching: vantaggi

- Possibilità di limitare il traffico entrante e/o uscente da una specifica rete servita da un opportuno sistema di caching
- Diminuzione, avvertibile dal client, della latenza relativa al recupero degli oggetti Web
- In alcune configurazioni può aiutare a realizzare una forma di *load balancing*

Problemi connessi al caching

- Violazione del modello *end-to-end*
  - sono più difficili , per il Content Server, le forme di autenticazione e la personalizzazione dei contenuti
- Gestione della coerenza
  - gli oggetti disponibili nelle cache devono essere allineati con gli originali
  - una possibile soluzione:
    il content server associa ad ogni oggetto un'informazione sulla validità temporale del documento; le cache utilizzano questa informazione per calcolare un "Time-To-Live"
- Non tutti gli oggetti sono "*cacheable*": alcune tipologie di contenuti, per la loro dinamicità non si prestano ad essere conservati localmente

Caching (1)

- Può essere client-side, server-side o intermedia (su un proxy)
- La cache server-side riduce i tempi di computazione di una risposta, ma non ha effetti sul carico di rete
- Le altre riducono il carico di rete
- HTTP 1.0 si basava su tre header:
  - **Expires**: il server specifica la data di scadenza di una risorsa
  - **If-Modified-Since**: il client richiede la risorsa solo se modificata dopo il giorno X. Richiede una gestione del tempo comune tra client e server
  - **Pragma: no-cache**: Fornita dal server, istruisce il client di non fare cache della risorsa in ogni caso
- HTTP 1.1 introduce due tipi di cache control:
  - Server-specified expiration
  - Heuristic expiration

Caching (2)

- Server-specified expiration
  - Il server stabilisce una data di scadenza della risorsa, con l'header Expires o con la direttiva max-age nell'header Cache-Control
  - Se la data di scadenza è già passata, la richiesta deve essere rivalidata. Se la richiesta accetta anche risposte scadute, o se l'origin server non può essere raggiunto, la cache può rispondere con la risorsa scaduta ma con il codice 110 (Response is stale)
  - Se Cache-Control specifica la direttiva must-revalidate, la risposta scaduta non può mai essere rispedita. In questo caso la cache deve riprendere la risorsa dall'origin server. Se questo non risponde, la cache manderà un codice 504 (Gateway time-out)
  - Se Cache-Control specifica la direttiva no-cache, la richiesta deve essere fatta sempre all'origin server

Caching (3)

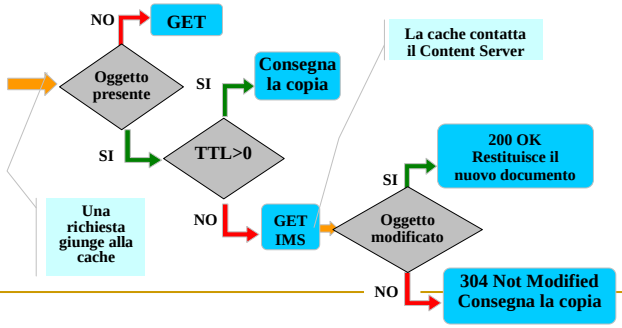
- Heuristic expiration
  - Poiché molte pagine non conterranno valori espliciti di scadenza, la cache stabilisce valori euristici di durata delle risorse, dopo le quali assume che sia scaduta
  - Queste assunzioni possono a volte essere ottimistiche, e risultare in risposte scorrette

Caching (4)

- Validazione della risorsa in cache
  - Anche dopo la scadenza, nella maggior parte dei casi, una risorsa sarà ancora non modificata, e quindi la risorsa in cache valida
  - Un modo semplice per fare validazione è usare HEAD: il client fa la richiesta, e verifica la data di ultima modifica. Ma questo richiede una richiesta in più sempre
  - Un modo più corretto è fare una richiesta condizionale: se la risorsa è stata modificata, viene fornita la nuova risorsa normalmente, altrimenti viene fornita la risposta 304 (not modified) senza body della risposta. Questo riduce il numero di richieste

Gestione della coerenza

- Problema: cosa succede se l'oggetto presente nel server è aggiornato ?
- La copia in cache deve essere aggiornata per mantenersi uguale all'originale
- HTTP fornisce due meccanismi per la gestione della coerenze:
  - TTL (Time To Live) : il server quando fornisce un oggetto dice anche quando quell'oggetto "scade" (header *Expires*)
    - Quando TTL diventa < 0, non è detto in realtà che l'oggetto sia stato realmente modificato
  - Il client può fare un ulteriore controllo mediante una GET condizionale (*If-Modified-Since*)



Parametri per valutare le prestazioni di una cache

- *Page Hit Rate*
  - percentuale di oggetti richiesti forniti senza interpellare il Server originario
- *Byte Hit Rate*
  - rapporto tra byte forniti localmente e byte richiesti al Content Server

Politiche per la gestione degli oggetti conservati nella cache

- Al riempirsi della cache, possono essere utilizzate tecniche diverse per il *replacement* dei documenti:
  - Least Recently Used (LRU)
  - Least Frequently Used (LFU)
  - Least Frequently Used with Dinamic Aging (LFUDA)
  - Greedy Dual Size e sue varianti

Organizzazione e comunicazione tra cache

- In presenza di più cache, queste possono essere organizzate in strutture gerarchiche o distribuite
- L'appartenenza di una cache ad una generica struttura può essere statica o dinamica
- Opportuni protocolli di gestione e comunicazione facilitano l'organizzazione e lo scambio di dati tra differenti cache
  - ICP (IETF)
  - WCCP (CISCO)
  - Cache-Digest (IETF)

Approcci per aumentare la scalabilità dei siti web

- Caching
- Web replication
- Content delivery networks: realizzano una forma dinamica di web replication

Web replication

- Soluzione server-side per permettere la scalabilitá Web:
- il sito Web è replicato su più server, eventualmente dislocati su aree geografiche differenti. Questo consente generalmente di migliorare la QoS percepita dagli utenti.
- Tre approcci possibili alla replicazione Web:
  - Mirroring esplicito
  - Mirroring statico trasparente
  - Replicazione dinamica trasparente
- Ciascuna tipologia fornisce un diverso compromesso tra flessibilità, complessità e costo.

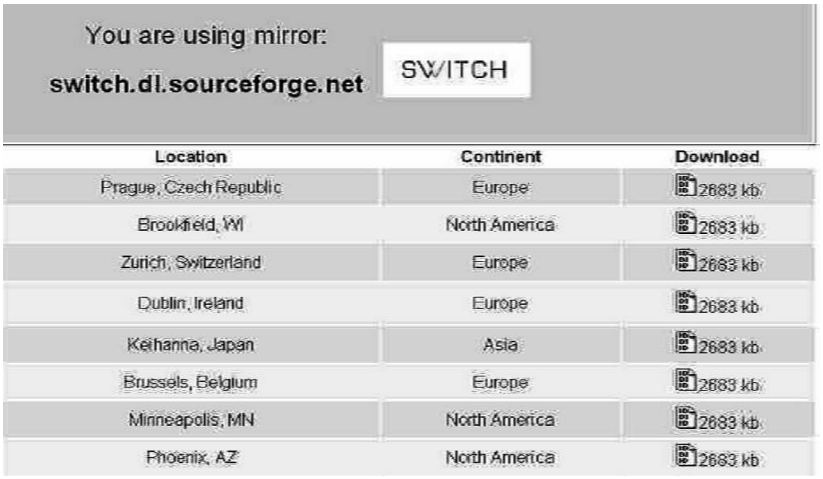
Problematiche legate alla replicazione

1. Come distribuire le richieste sui server?
   - Definizione delle architetture
   - Definizione delle tecniche di routing
2. Quale server scegliere per soddisfare una richiesta?
   - Definizione di politiche di selezione dei server
3. Su quanti server replicare lo stesso oggetto?
   - Definizione di politiche di replicazione
4. Come garantire la consistenza delle repliche?

Replicazione mediante mirroring esplicito

- La più semplice forma di replication,largamente usata, è il mirroring esplicito
  - Il sito è interamente duplicato su altri server
  - Esempi: Tucows, GNU, Sourceforge, Linux, …
- Tuttavia questa soluzione presenta diversi svantaggi:
  - Gli utenti devono scegliere esplicitamente quale mirror usare
  - Il bilanciamento del carico non è controllabile
  - La decisione di rimuovere un mirror è difficile perchè può creare notevoli disagi agli utenti (es. link salvati dagli utenti possono diventare invalidi)

Esempio di mirroring esplicito



You are using mirror:
switch.dl.sourceforge.net SWITCH

Location	Continent	Download
Prague, Czech Republic	Europe	2683 kb
Brookfield, WI	North America	2683 kb
Zurich, Switzerland	Europe	2683 kb
Dublin, Ireland	Europe	2683 kb
Keihanna, Japan	Asia	2683 kb
Brussels, Belgium	Europe	2683 kb
Minneapolis, MN	North America	2683 kb
Phoenix, AZ	North America	2683 kb

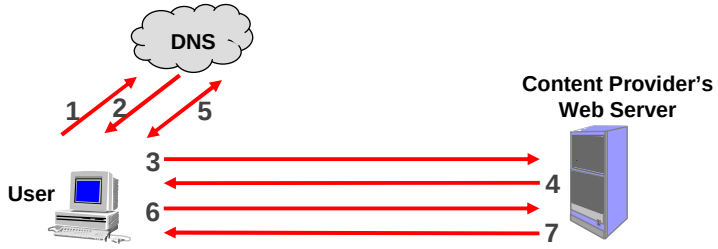
Cosa è una CDN

- Una Content Delivery Network è un'infrastruttura creata per distribuire efficacemente agli utenti di Internet i contenuti dei siti web più popolari
- Una CDN si basa sulla distribuzione di repliche dei contenuti dal server principale del "Content Provider" ad una molteplicità di server disposti sulla rete da un "Content Delivery Operator"
- Si presenta come un servizio a pagamento del quale usufruiscono i gestori dei siti web commerciali più popolari
- Esempi: Akamai, Speedera, Inktomi

Obiettivi di una CDN

1. Alleviare il server web “master” dal carico degli utenti, in particolare proteggerlo da picchi di traffico improvvisi (flash crowds)
2. Offrire i contenuti ai singoli utenti tramite server collocati in prossimità degli utenti (alla periferia della rete)
3. Rendere il sistema di distribuzione dei contenuti più affidabile e robusto ai guasti

Interazioni che determinano il tempo di accesso ad una pagina web



- **User enters www.xyz.com**
- **Browser requests IP address for www.xyz.com**
- **DNS returns IP address**
- **Browser requests HTML**
- **Content provider’s web server returns HTML**
- **Browser obtains IP addresses for hostnames listed in URLs of objects embedded on page**
- **Browser requests embedded objects**
- **Content provider’s web server returns embedded objects**

Il modello *end-to-end* del Web

- La rete è un'entità passiva
- I servizi vengono forniti solo quando richiesti: *"just in time delivery model"*

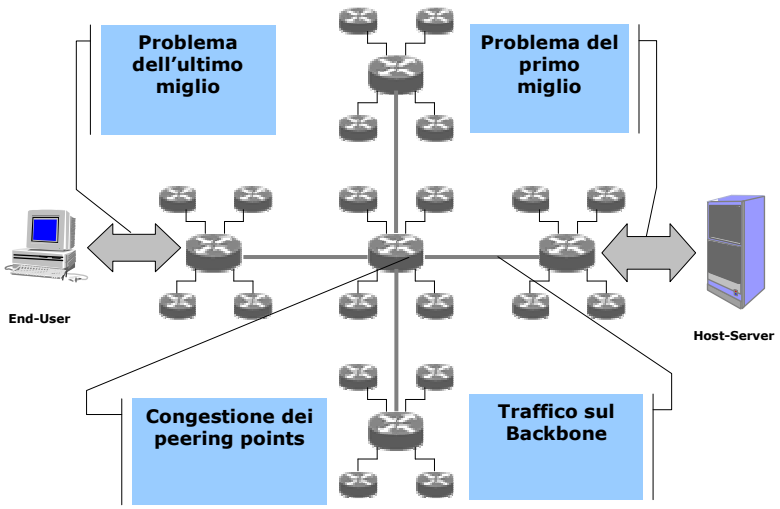
Vantaggi

- Le modifiche apportate ai dati dal Content Server si riflettono immediata-mente nei documenti consegnati
- Il Content Server può *tracciare* le richieste dei client e calcolare la popolarità dei servizi forniti
- Possibilità di differenziare l'accesso ai servizi

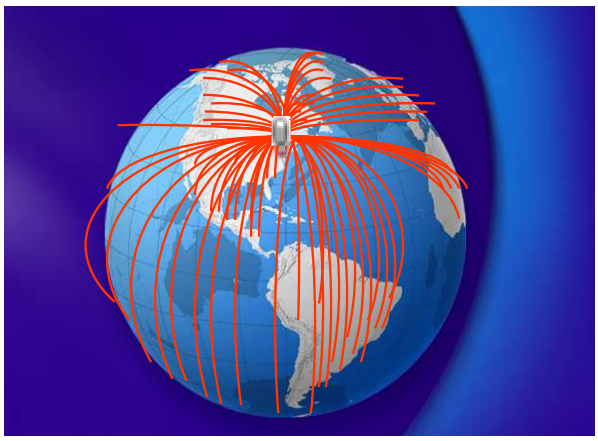
Svantaggi

- Elevato tasso di duplicazione del traffico Internet
- Carico elevato per i server che distribuiscono contenuti molto popolari

Limiti dell'approccio centralizzato



Limiti dell'approccio centralizzato (2)



L'approccio CDN

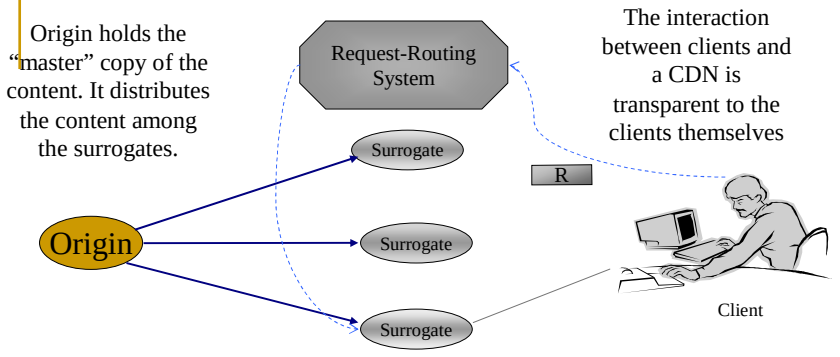
Content Delivery Networks

- Tramite una infrastruttura, spesso privata, distribuiscono, in maniera capillare i contenuti di uno specifico Content Server
- Utilizzano forme proprietarie di caching basate su una complessa gestione del DNS, caratterizzata, tra l'altro, dalla conoscenza dell'indirizzo IP del Client
- Gestione centralizzata dei contenuti

Servizi offerti

- Le Content Delivery Network offrono ai Content Server la possibilità di raggiungere, con una certa QoS, una vasta utenza
- Le CDN, d'altra parte, propongono a ISP di medie e grandi dimensioni, di collaborare, spesso gratuitamente, alla loro struttura

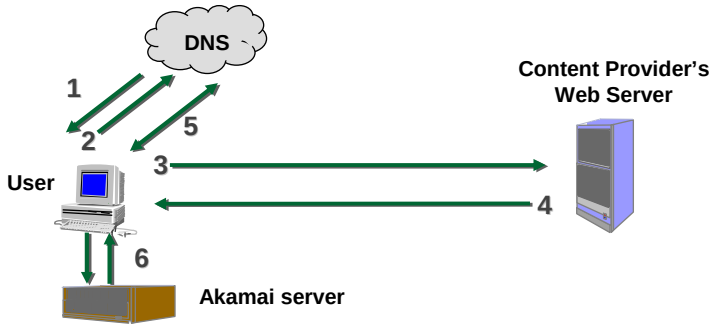
CDN infrastructure



Surrogates are geographically distributed servers that maintain copies of the content

26

Come funziona Akamai



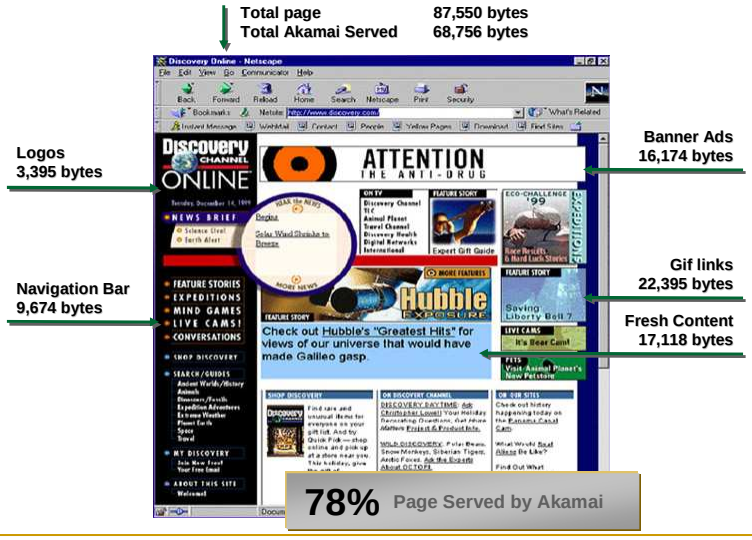
- User enters www.xyz.com and browser requests IP address for www.xyz.com
- DNS returns IP address
- Browser requests HTML
- Content provider's web server returns page with *Akamaized* URLs
- Browser obtains IP address of optimal Akamai server for embedded objects
- Browser obtains objects from optimal Akamai server

Cosa significa "Akamaizzare" i riferimenti

```
<html>
<head>
<title>Welcome to xyz.com!</title>
</head>
<body>
<img src="http://~~www~~.xyz.com/logos/logo.gif">
<img src="http://~~www~~.xyz.com/jpgs/navbar1.jpg">
<h1>Welcome to our Web site!</h1>
<a href="page2.html">Click here to enter</a>
</body>
</html>
```

ak

Efficacia: un esempio



Domande?