

# Impianti informatici AA 2004 / 2005 – Seconda Parte

## 1 INTRANET

### 1.1 Tecnologie e componenti

Ci sono varie disposizioni

- Rete locale a bus condiviso
- Switched Ethernet
- Connessione internet con modem

**Router:** sistemi che collegano le parti della intranet e instradano i pacchetti. Connessioni pubbliche oppure private. I router inoltrano grazie alle tabelle di routing

**Firewall:** svolgono funzioni relative alla sicurezza digitale. Generalmente incorporati nei router, includono tecniche di packet filtering e proxy

**Server Proxy:** permettono efficienza nell'uso della rete (riduce uso di banda e ottimizza tempi di accesso alle pagine). Caching, sicurezza, controllo accessi e condivisione risorse.

Una **intranet** è un sistema di *comunicazione e scambio di informazioni* tra componenti aziendali (persone, funzioni, dipartimenti), con accessi controllati e con ambito geografico predefinito. E' un *sistema informativo aziendale web-based*, basato sul modello *client server*.

Essa è un sistema costituito da una rete aziendale protetta da firewall, e dalle applicazioni eseguite, realizzato utilizzando le tecnologie tipiche dell'*Internetworking*. La rete privata può essere formata da più LAN interconnesse, il cui scopo principale è la *condivisione di informazioni e risorse*, facilitando la *collaborazione* anche tra persone fisicamente disanti, rendendo accessibile la *Knowledge Base* e strumenti per la *collaborazione*.

#### 1.1.1 SERVIZI OFFERTI

- **Cooperative work – groupware – collaborative computing:** condivisione di documenti, gestione di messaggi, forum di discussione, controllo sicurezza, versioning, ricerche, integrazione con il database.
- **Education – e-learning:** seguire corsi in modo asincrono anche da casa; controlli remoti del rendimento, velocità di fruizione personalizzata, disponibilità per azioni di rinforzo sugli argomenti.
- **Knowledge Management:** massimizzare capacità di apprendimento; rendere la conoscenza accessibile, espandibile e riciclabile; ognuno può aggiungere conoscenza al sistema, che deve essere sempre disponibile, accessibile anche dai non esperti, aggiornato in modo automatico.

I vantaggi di una intranet:

- *Integrazione con internet*
- *Interoperabilità*
- *Bassi costi di produzione di sw applicativo*

- *architettura scalabile*
- *interfaccia semplice*

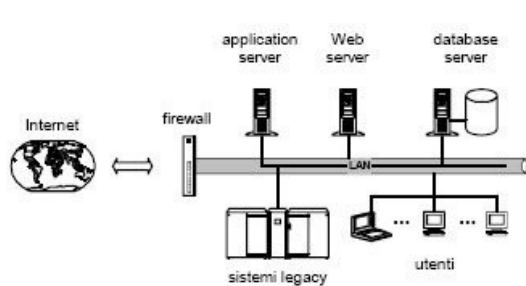
Gli svantaggi

- *Colli di bottiglia distribuiti*
- *Problemi di sicurezza*
- *Obsolescenza delle tecnologie*

Si distingue, in base alle funzioni, tra intranet istituzionali, operative e di Knowledge management.

- **Istituzionali:** generalmente unidirezionali, bassa interattività. Offrono ai dipendenti informazioni e servizi relativi ad attività di supporto ai dipendenti. **Obiettivi:** riduzione dei costi tramite procedure efficienti, aumento della fidelizzazione dei dipendenti e dell'immagine aziendale.
- **Operative:** supportano l'attività dell'azienda (logistica, produzione, marketing, vendite...), elevata integrazione con tutti i sistemi aziendali e dei fornitori. **Obiettivi:** aumento QoS con efficiente gestione clienti (CRM) e vendite.
- **Knowledge-Management:** supporto di condivisione e reperimento delle informazioni e della conoscenza. Archiviazione e reperimento intelligente di documenti, gestione modelli di analisi statistiche, individuazione di gruppi di competenza.

### 1.1.2 TIPI DI ARCHITETTURE



Generalmente le architetture sono strutturate su più livelli

- **3 livelli:** *browser, web server & applications, database servers*. Eliminano la necessità di aggiornamento pagine, consentono di operare online sul db, permettono realizzazione di pagine dinamiche.
- **Zona privata e zona pubblica:** maggiore sicurezza e posizionamento strategico dei servizi e degli apparati mission-critical.
- **Extranet:** collegamento di più intranet distanti (o di utenti remoti) tramite Internet.
- **Architettura a 4 livelli:** con utilizzo di EJB. *Client Side presentation, Server side presentation, Business logic, Enterprise information system*.

Sono possibili variazioni e personalizzazioni di questi schemi, e anche accoppiamenti tra essi.

## 1.2 Esempi di Architetture

Vedere i *case studies* sulle slide. Nelle architetture avanzate vengono usati sistemi più sofisticati:

**Load Balancer:** permette di ottenere prestazioni elevate, ripartendo il carico su più server, e maggior robustezza in caso di guasti. Esso adotta un meccanismo di bilanciamento di tipo *round-robin*. **Problema:** gestire le sessioni **Soluzione:** Stateful LB inviare le richieste di un client specifico allo stesso server. In alternativa si serializzano gli oggetti *session* e gli *EJB* sul DB (meccanismo di sincronizzazione)

**SSL Accelerator:** i protocolli crittografati utilizzati per le applicazioni critiche (SSL/https) sono

onerosi come risorse di calcolo sul webserver. Per questo si adottano componenti HW specifici che operano in congiunzione con il load balancer (prendendo in carico la gestione della crittografia e comunicando ai web-server flussi http non crittografati).

**Web Server:** generalmente replicati per garantire efficienza e tempi di risposta accettabili in caso di traffico elevato (Blade Server IBM).

**Application Server:** fornisce la logica applicativa delle applicazioni, inoltre può supportare le funzionalità di *middleware* (message queing, RMI/CORBA...) e la gestione del pooling delle risorse (DB connections, threads...)

**Database Server:** cuore di tutte le applicazioni, richiede considerevoli risorse in termini di CPU e RAM, nonché di sistemi di storage veloci e capienti.

**Gestione dei domini:** si può partizionare un server enterprise in più “domini”. Ogni dominio rappresenta un server virtuale con il proprio sistema operativo, le proprie applicazioni e le sue interfacce di rete. I domini possono essere modificati online, aggiungendo o rimuovendo risorse.

**Gestione dei cluster:** i cluster sono insiemi di server in parallelo. Obiettivo: l'aumento dell'affidabilità mediante la ridondanza, secondariamente il load balancing. Possono essere *Attivo/Attivo* oppure *Attivo/Passivo*. Il nodo attivo riceve le richieste, in caso di failure si ha lo “switch del servizio” (nodo Passivo diventa Attivo, le richieste vengono ridirezionate).

**Mainframe:** rappresentano i “sistemi legacy” che sono “mission critical”. E' necessario interfacciarsi con essi per garantire i servizi dell'azienda

## 2 Web Caching

### 2.1 Introduzione

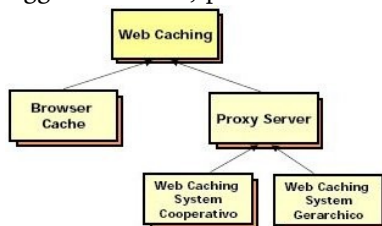
Esiste il problema della scalabilità, peggiorato dalla diffusione della banda larga, dall'aumento di esigenze degli utenti, e dalle nuove applicazioni emergenti a maggior richiesta di banda e tempo di processing.

Si può utilizzare un approccio *brute force* andando ad aggiungere risorse hardware sfruttandone il basso costo, tuttavia questa soluzione è accettabile solo nel breve periodo e contribuisce a far crescere i costi e la difficoltà di manutenzione dell'impianto. Inoltre è probabile che a breve il sistema sia nuovamente congestionato!

### 2.2 Web caching

Si sfrutta la **ripetitività** nel *comportamento degli utenti* per velocizzare il download degli oggetti più popolari (salvati su un proxy condiviso). Questo permette di ottenere maggior velocità di accesso ai documenti e minor traffico in uscita verso internet.

Tramite la **web replication** invece si replicano in parallelo i server da cui è possibile ottenere gli oggetti richiesti, permettendo di ottenere bilanciamento del carico, eliminazione della congestione dovuta ai picchi e miglioramento dei tempi di risposta percepiti.



La classificazione distingue posizionamento ed architettura del sistema di web-caching.

**Browser Cache:** per ogni richiesta, il browser controlla se l'oggetto è stato caricato recentemente. Se ho un *hit*, se rilevo che la copia non è aggiornata la riscarico, altrimenti la carico dalla

cache e non la riscarico. Se non è disponibile (*miss*) lo scarico e una copia è salvata nella cache.

**Proxy Server:** server di rete che svolgono funzioni di controllo accessi ed implementano web caching. E' un sistema di caching *condiviso*, che sfrutta la ripetitività del comportamento di una comunità di utenti. Per evitare colli di bottiglia, il proxy spesso è una macchina dedicata (per reti di medie dimensioni: Squid con 128MB di RAM e disco da 512MB a 24GB).

Nella **configurazione semplice**, il proxy intercetta tutte le richieste uscenti (soluzione poco realistica per reti con più link verso l'esterno e inefficiente per reti a traffico elevato: tuttavia vi è una notevole semplicità di installazione e manutenzione).

Nei sistemi complessi, esiste una **gerarchia di proxy**. Le richieste salgono la gerarchia finché vengono servite oppure vengono inoltrate in internet. Essa coinvolge anche i proxy esterni all'azienda che si trovano nel cammino della richiesta.

### 2.2.1 ESEMPIO: RETE DI UN ISP

La rete di un ISP è composta da:

- **ISP Backbone Network:** dorsale di rete che smista il traffico verso internet
- **Access Router:** consente l'instradamento del traffico aziendale verso ISP Backbone
- **Dial-up concentrator:** spartisce la banda di accesso alla ISP backbone tra utenti privati con connessioni lente
- **Backbone Router:** router di servizio per l'instradamento all'interno della ISP backbone.
- **Gateway Router:** instradamento della backbone verso reti gestite da altre società.

ISP stipula contratti di peering che consentono il mutuo instradamento di traffico ad altre reti. Il contratto può anche prevedere flussi di traffico unidirezionali.

**Multihoming:** molteplicità dei link che collegano l'azienda e l'ISP. Permettono di mantenere la connettività in caso di guasto e di bilanciare il carico. Essi rendono difficile la collocazione del proxy aziendale, e anche di quello dell'ISP.

- Proxy non-trasparente: richiede che i browser degli utenti vengano configurati esplicitamente. Soluzione semplice ma richiede collaborazione ed è complessa da realizzare per un ISP. **Collocazione:** *Configurazione esplicita, browser autoconfigurante, proxy autodiscovery.*
- Proxy trasparente: maschera l'esistenza dei proxy, ma richiede hardware dedicato oppure una configurazione molto complessa. **Collocazione:** *Interception Proxy, switch L4, switch L7, Interception Link.*

Configurazione Esplicita: permette *flessibilità* nella collocazione del proxy, ma è possibile che l'utente eviti di configurare il proxy. Scarsa tolleranza ai guasti

Browser Autoconfiguration: il browser scarica un file *Proxy AutoConfiguration* che descrive la politica di instradamento delle richieste. Permette *flessibilità nelle politiche di gestione e buona tolleranza ai guasti*. Tuttavia il server da cui si scarica il file di configurazione è il nuovo *single point of failure* della rete.

Proxy Autodiscovery: permette convergenza di collocazione trasparente e non. La posizione del file PAC è indicata dal server DHCP o DNS. E' *trasparente all'utente* e più *tollerante ai guasti* dell'autoconfiguration. Per contro richiede l'adozione di un protocollo non standard (WPAD – *Web Proxy Auto-discovery*)

**Interception Proxy:** i router ridirezionano il traffico uscente verso il proxy (IP-in-IP encapsulation). Alla ricezione il proxy estrare il pacchetto IP interno e si finge il server di destinazione (*Connection Hijacking*). E'vantaggioso perchè in caso di caduta del proxy, essa è rilevata dai router che escludono automaticamente il meccanismo. Tuttavia il *connection hijacking* viola il principio end-to-end alla base di internet "Un'applicazione può essere implementata completamente e correttamente soltanto con la conoscenza e l'aiuto delle applicazioni che stanno ai nodi del sistema di comunicazione". Possono inoltre sorgere problemi connessi al routing delle richieste di reti multipath.

Il problema del multipath può essere risolto dallo **Switch L4:** è un dispositivo di rete che smista i pacchetti in transito anche in base al contenuto dell'header TCP. Si può imporre che i pacchetti di una stessa connessione siano instradati sempre attraverso lo stesso ISP. Può svolgere anche funzioni di load balancing, ed evita l'IP-in-IP encapsulation.

Un'ulteriore estensione è rappresentata dallo **Switch L7,** che può interpretare anche richieste a livello HTTP (non ridirezione di richieste a oggetti non cacheabili come le pagine dinamiche). Permette anche la separazione dei proxy in base al contenuto cacheato. Può ridurre il throughput della rete se non è usato con criterio.

**Intercepting Link:** estensione dello switch L7 (macchina che è sia un proxy, sia uno switch L7). E'un dispositivo integrato ed autoconfigurante.

## 2.2.2 TECNICHE DI ANALISI PRESTAZIONALI

E'necessario effettuare analisi costi-benefici per prevedere l'effettivo incremento prestazionale. Bisogna valutare la quantità di **shared hits** (cache hits dovuti all'attività degli altri utenti): generalmente sono compresi tra 45% e 50% . Il **byte hit rate** è attorno al 50% (50% dei KB scaricati proviene dal proxy). Assumeremo sh pari a 45%.

**Tempo di download: external latency** (77-88% totale) + **internal latency**

Il guadagno di banda massimo è pari a  $45\% \times 88\% = 40\%$  circa. Il calcolo però è scorretto, non tiene conto delle dimensioni dei file. Nel caso ottimo si arriva a guadagni del 22 – 26% . Bisogna tenere conto che generalmente si guadagna sui trasferimenti (già brevi) di files piccoli.

Inoltre:

- Necessaria una fase di connessione tra proxy e server web *in caso di cache miss*.
- Il proxy introduce overhead di gestione TCP, ovvero un *ritardo* dovuto all'analisi del pacchetto in transito.

Il guadagno nel caso pessimo risulta essere del 3% ! Convieni? La stima di questi studi è severa perchè non tiene conto delle *congestioni causate dai download* che potrebbero essere resi locali. Inoltre si permette anche il **TCP connection caching:** vengono mantenute attive le connessioni già aperte per i cache miss (si risparmiano i tempi di connessione, ma non si permette l'uso simultaneo del canale da parte di più clients – *no pipelining*, perchè andrebbe servita in ordine FIFO e gli utenti dovrebbero attendere il loro turno)

**RTT (Round Trip Time):** tempo di viaggio di un pacchetto da i a j. Se è troppo alta, si limita notevolmente il throughput TCP. La presenza di proxy spezza il viaggio in pacchetti su due RTT minori (TCP Connection splitting). Vantaggi maggiori per oggetti di grandi dimensioni, (teoricamente 3 x throughput base , 2 x nella realtà)

**Riduzione del traffico:** dovuta al proxy, è stimata al 25% Tuttavia bisogna adottare certe cautele (interrompere immediatamente il download di oggetti in presenza di abort del client)

	Aggregato	HTML	Immagini	ZIP/RAR/...
<b>Numero oggetti</b>	<b>20728</b>	<b>1141</b>	<b>7025</b>	<b>67</b>
<b>Media (kB)</b>	<b>15.1</b>	<b>7.1</b>	<b>8.7</b>	<b>1501.8</b>
<b>Mediana (kB)</b>	<b>4.6</b>	<b>4.5</b>	<b>4.4</b>	<b>38.1</b>

**Dimensione oggetti:** la distribuzione è fortemente asimmetrica, prevalgono le piccole dimensioni (oltre il 75%). Tra i rimanenti, una parte consistente è di grandi dimensioni.

Questa asimmetria implica requisiti contrastanti per il web-caching:

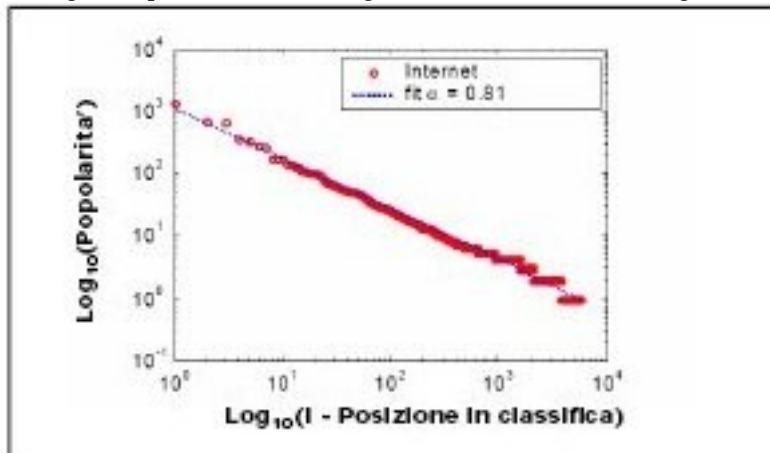
1. Cacheare oggetti piccoli: miglioramento dell'*hit rate*
2. Cacheare oggetti grandi: *riduzione tempi di download* più lunghi

L'obiettivo è fare in modo che il tempo medio di download percepito dagli utenti sia minimo. Questo dipende dalle caratteristiche del traffico in esame e dalla popolarità dei diversi tipi di oggetti web (es. 90% download diretti a file HTML e immagini, ogni pagina contiene mediamente 10 immagini). Si deduce che esiste un **grado di dipendenza** tra gli accessi ai singoli oggetti.

La **Legge di Zipf** (1949) descrive bene questa dipendenza: "La frequenza d'uso della *k*-esima parola più popolare è inversamente proporzionale a *k*" (applicata alla linguistica, occorrenze delle parole negli articoli).

In ambito web: se A e B sono il primo ed il secondo oggetto per popolarità, allora la frequenza di accesso a B è la metà di quella di A.

Forma generale: l'*i*-esimo oggetto più popolare è proporzionale a  $1/i^\alpha$ , con  $1 \geq \alpha > 0$ . Se  $\alpha=1$  allora si ritorna nel caso precedente. Il parametro  $\alpha$  si ricava in modo empirico dall'analisi dei log di traffico. Nell'immagine si può vedere il fitting di  $\alpha$  come coefficiente angolare.



$$\text{Popolarità} \sim 1/i^\alpha \rightarrow \text{Log}_{10}(\text{Popolarità}) \sim -\alpha \text{Log}_{10}(i)$$

**Performance a regime:** esistono formule analitiche in grado di stimare da  $\alpha$  le performance a regime (per  $t \rightarrow$  infinito) di un sistema di web-caching. Nel caso semplice  $\alpha=1$  la probabilità che un oggetto sia referenziato entro *k* richieste dall'ultimo download è pari a:

$$d(k) \approx \frac{1}{k \ln N} \left( \left(1 - \frac{1}{N \ln N}\right)^k - \left(1 - \frac{1}{\ln N}\right)^k \right)$$

Dove N rappresenta il numero di oggetti web di cui è composto il sito.

La metrica della **Stack Distance (SD)** consente di stimare l'hit rate previsto per un proxy. Partendo dai dati nei log, si controlla la frequenza di accesso alle pagine. Consente inoltre il dimensionamento in funzione dell'hit-rate scelta.

Come si calcola? Si parte da Stack vuoto

1. Leggi prossima richiesta dal file di log
2. Sia Y la destinazione di tale richiesta, se lo stack **non** contiene Y allora inserisco Y in cima allo stack e print SD=0
3. Se Y è presente alla posizione k, sposto Y in cima e print SD=k
4. Se esistono altre richieste torno a 1

I numeri stampati dalla print costituiscono i valori osservati per la SD. Si osserva che con un buffer di 50 oggetti si riesce a gestire il 90% delle richieste.

**Località spaziale:**  $LS = (\text{N}^\circ \text{di path diversi presenti nel log}) / (\text{N}^\circ \text{di path possibili})$

**Change Ratio:** il 15% delle richieste (dato empirico) è diretto a pagine aggiornate di recente.

**Richieste Abortite:** si tiene conto, nel calcolo prestazionale, di un 5 - 10% di richieste abortite.

**Piggybacking:** c'è la possibilità che un oggetto non sia aggiornato -> sviluppo di meccanismi di validazione consistenza.

1. Richiesta genera *cache miss*
2. Proxy inoltra la richiesta
3. Un thread del proxy aggancia in coda alla richiesta un treno di richieste di validazione per altri oggetti presenti in cache e scaricati in precedenza da quel server
4. Tra questi, se un oggetto è stato modificato, allora è scaricato soltanto se è abbastanza popolare. Richiesta di validazione eseguita con metodo HEAD di HTTP che non obbliga al download dell'oggetto

### 3 Web replication

E'una soluzione server-side che permette la scalabilità: il sito web è replicato su più server, eventualmente dislocati in aree geografiche differenti. Questo migliora QoS percepita dagli utenti. Gli approcci possibili sono:

- **mirroring esplicito**
- **mirroring statico trasparente**
- **replicazione dinamica trasparente**

Ciascuna tipologia fornisce diversi compromessi. Le problematiche principali sono la definizione delle architetture e delle tecniche di routing, delle politiche di selezione dei server, delle politiche di replicazione e della garanzia di consistenza. Inoltre vi sono problematiche di sicurezza e di tolleranza ai guasti (non trattate in questa sede).

**Mirroring esplicito:** è la totale duplicazione del sito su differenti server. Presenta diversi svantaggi: gli utenti devono scegliere quale mirror usare, il bilanciamento del carico non è controllabile, rimuovere un mirror è una decisione critica (es. Link utenti diventano invalidi)

Le politiche di mirroring e di consistenza sono variabili e definite dall'amministratore.

**Mirroring static trasparente:** le richieste sono automaticamente redirezionate ai mirror senza l'intervento dell'utente. E'una soluzione semplice da implementare, la limitazione principale consiste nell'impossibilità di rilocare in tempo reale le risorse per fronteggiare picchi di richieste. Utilizzo di tecniche *content-blind* oppure *content-aware* per la redirezione.

**Replicazione dinamica trasparente:** repliche create, cancellate e rilocate in funzione dello stato di carico del sistema. La politica di allocazione è spesso gestita in automatico a livello software. Si pongono proxy di fronte ai webserver in funzione di *reverse proxy*. Un insieme di reverse proxy condiviso da più siti è detto Content Delivery Network (CDN).

## 3.1 Tecniche di redirezione

Il routing delle richieste deve affidarsi a tecniche di redirezione sulle varie repliche.

**Redirezione Content Blind:** è indipendente dal tipo di oggetto e ogni server contiene una replica completa del sito

- **DNS client:** il DNS restituisce una lista di IP dei server replica. Al client viene *delegata la scelta* a quale server collegarsi (in genere basata su qualche metrica). In alternativa, il DNS può restituire a rotazione uno degli IP della lista. Limitazione: amministratore non ha controllo sulle politiche di bilanciamento del carico (che sono delegate)
- **DNS autoritativo:** per ogni nome di dominio, si aggiorna la richiesta DNS contattando un DNS autoritativo controllato dall'amministratore del sito. L'indirizzo IP che fornisce è fatto variare a seconda delle esigenze di bilanciamento (es. geografico). Non c'è garanzia sul fatto che l'aggiornamento venga con la frequenza desiderata da chi effettua il balancing.
- **Load Balancer:** utilizzo di switch dedicato (*Load Balancer*) davanti a una server farm con l'obiettivo di bilanciarne il carico. Il load balancer può essere uno switch L4 o L7. Questa soluzione evita di riconfigurare il DNS ma richiede l'acquisto di hw dedicato. Inoltre può essere inadeguato per effettuare scalabilità geografica. Se poi se ne usa uno singolo, si introduce un single point of failure nel sistema. Come funzionano: il DNS restituisce l'IP del LB, che riceve la richiesta e fa la scelta di instradamento. Inoltre salva l'IP del client in un *session database* (permette ai successivi pacchetti utente di essere gestiti dallo stesso server). Il server risponde, LB intercetta la risposta e sostituisce il suo IP a quello del client (per evitare il contatto diretto tra i due)

**Content-blind con replicazione parziale:** si cerca di ottimizzare la replicazione (pochi oggetti sono responsabili della maggior parte del traffico). Si replicano i file più popolari, si crea un Filesystem distribuito per collegare i vari server.

**Content-aware replication:** Appled Java capaci di selezionare lato client il server da contattare. HTTP redirection (con performance scadenti per la latenza di riconnessione). L7 switching.

### 3.1.1 SELEZIONE DEL SERVER

Essa avviene in base a metriche di performance.

**Metriche di prossimità:** selezione del server "più vicino" al client. Distanza *geografica*, *hopcount*, *AS hopcount*. Esiste scarsa correlazione tra tempi di download e posizione geografica, e anche rispetto al numero di hop. AS hop è la metrica più efficace (conta il numero di router BGP attraversati)

**Metriche di carico:** si sceglie il server cui inviare le richieste in base a numero di connessioni aperte, tasso di arrivo delle richieste, tempo di processi in esecuzione, tempi di risposta

**Metriche aggregate:** si cerca di rilevare lo stato di congestione della rete, basata su diverse misure (ping, banda, latenze medie, RTT, packet loss rate). Esse sono le più diffuse.

## 4 Parallel Computing

Si elaborano sistemi in grado di lavorare in parallelo per ovviare alle richieste sempre maggiori di potenza di calcolo, e per ridurre i costi ed il time to market.

I problemi principali sono dati dall'efficienza ottenibile, dall'accesso concorrente alle risorse, cache coherence, capacità di parallelizzare un problema e suddivisione del dominio del problema.

**Speedup:**  $S_p = T_1 / T_p$  (l'incremento di prestazioni ottenuto al crescere del numero di processori usati rispetto al caso di monoprocesso)

**Efficienza:**  $E_p = S_p / p$  (indica la capacità di sfruttare in modo effettivo la potenza di calcolo dei processori paralleli)

I modelli hardware sono classificabili secondo la struttura di memoria (**distribuita** oppure **condivisa**)

La **legge di Amdahl** lega l'incremento prestazionale alla frazione di calcolo eseguita in modalità seriale  $S(n) = n / [n + F_s + (1 - F_s) \cdot n]$ . Lo speed up reale tende ad assestarsi sul valore della frazione seriale, al crescere del numero di processori. In realtà  $F_p$  non è uguale a  $T_p/n$  perchè ci sono anche le comunicazioni