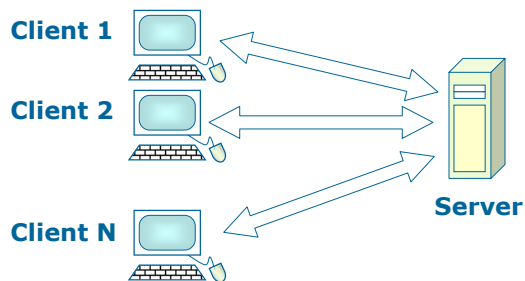


Componenti Web: client-side e server-side

A. Bechini 2004

Attività di applicazioni web

- Applicazioni web: un insieme di componenti che interagiscono attraverso una rete (geografica)
- Sono *applicazioni distribuite* logicamente caratterizzate da un'unità che fornisce un servizio, e da molteplici clienti per tale servizio



A. Bechini 2004



Scenario particolare

- Il server in realtà è un *punto di accesso logico* a funzionalità fornite da un insieme di moduli.
- La comunicazione tra il server e i client è standardizzata tramite il protocollo HTTP.
- Un generico modulo client è basato su un'applicazione, il browser, che deve supportare un insieme minimo di funzionalità (visualizzazione HTML)

A. Bechini 2004



Chi fa cosa

- Come in ogni applicazione distribuita, occorre progettare come distribuire l'attività computazionale sui vari moduli del sistema.
- In particolare, occorre capire quali compiti assegnare al server e ai client.
- Fattori di cui tener conto:
 - latenza di comunicazione
 - Possibile limitate capacità del client
 - Eterogeneità dei client
 - Scalabilità (congestione sul canale di comunicazione e sull'applicazione server)
 - ...

A. Bechini 2004



Latenza di comunicazione

- La comunicazione tra client e server ha sempre un costo in termini temporali
- La comunicazione deve avvenire soltanto quando è strettamente necessario
- La comunicazione ha luogo quando:
- L'utente richiede informazioni che non possono essere conosciute sul momento nell'ambito del modulo client.

A. Bechini 2004



Limitate capacità del client

- Due possibili limitazioni:
 - Piattaforma con basse (insufficienti) prestazioni
 - Browser incapace di supportare funzionalità aggiuntive alla visualizzazione HTML
- Il browser costituisce l'ambiente di esecuzione di potenziale "codice", che implementa le attività del modulo client
- Il browser deve perciò supportare tale esecuzione, al fine di poter sfruttare le capacità computazionali del client

A. Bechini 2004



Scalabilità e congestione

- Il server gestisce un numero max di richieste nell'unità di tempo: diminuendo la comunicazione, si possono servire contemporaneamente più client
- La diminuzione delle richieste HTTP abbassa il traffico di rete
- A ogni richiesta HTTP, il server deve impegnare la CPU: meno richieste significa minor carico (e quindi maggiore efficienza)

A. Bechini 2004



La cache del browser

- Il browser generalmente mantiene sul file system una copia dei dati scaricati più recentemente. Nel caso tali dati vengano richiesti di nuovo, essi sono già presenti localmente e non occorre reperirli tramite la rete.
- Questo meccanismo è noto come caching; tipicamente si possono configurare alcuni parametri relativi alla cache del browser (p.es. la dimensione).

A. Bechini 2004

Utilizzo di proxy

- Il meccanismo del caching può essere utilizzato anche ricorrendo a calcolatori detti *Proxy*
- Un Proxy è un server, "geograficamente" vicino al client, a cui il browser richiede una risorsa web, prima di contattare il server che effettivamente la detiene.
- Se la risorsa è presente sul proxy, si scarica da lì, altrimenti si ricorre al server appropriato.

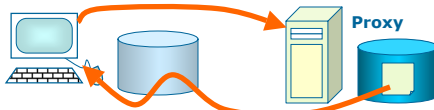
A. Bechini 2004

Uso di caching: scenari

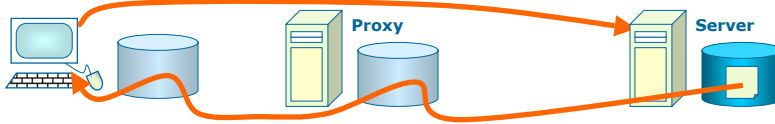
- Reperibilità nella cache del browser



- Reperibilità sul proxy



- Accesso al server (no caching)



A. Bechini 2004



Problemi con cache e proxy

- La cache e i proxy tendono a limitare il traffico di rete e a diminuire le latenze, ma:
 - Non si ha la sicurezza che la risorsa reperita sia al momento *coerente* con quella effettivamente richiesta
 - Quando la maggior parte delle richieste di riferisce a contenuti dinamici, la probabilità di reperire in cache/proxy la risorsa diventa molto bassa, e il meccanismo diventa inefficiente
- Sono state sviluppate altre soluzioni per questo problema, che richiedono però l'intervento di applicazioni server-side (Akamai...)

A. Bechini 2004



Far lavorare il client!

- In definitiva, un modo per migliorare le prestazioni generali dell'applicazione è cercare di trasferire, quando possibile, una parte dell'attività computazionale sul client
- Questo attività dovrà essere supportata/controllata dal browser
- Occorre sviluppare specifiche tecnologie per questo scopo

A. Bechini 2004



Fat client vs. Thin client

- Due alternative "classiche":
 - FAT CLIENT, ovvero buona parte dell'attività è svolta sul client
 - THIN CLIENT, non si fanno ipotesi sulle capacità del client, e quanto più possibile è svolto sul server
- Tendenza attuale: inizialmente il client viene implementato "leggero"; eventualmente poi lo si carica (per migliorare la grafica)
- Se il client è troppo pesante, diventa lento e goffo.



Plug-in per il Browser

- Il Browser viene generalmente progettato in modo da poter ospitare moduli addizionali (plug-in) che ne arricchiscono le funzionalità
- Nei browser moderni, i plug-in possono essere caricati automaticamente nel momento in cui si reperisce una risorsa che li richiede per svolgere la sua specifica attività client-side.



Programmabilità client-side

- Perché infine ha senso fornire la programmabilità client-side?
 - Evitare comunicazioni
 - P.es. controllare gli argomenti di un form HTML prima di inviarli
 - Migliorare i tempi di risposta
 - Strutturare meglio l'applicazione web
 - Riferendoci al design pattern MVC:
 - Il *controller* viene fornito dai moduli Server-side
 - Il *model* viene fornito da Database server
 - Il modulo Client-side si presta per implementare alcuni comportamenti dinamici della *view*.

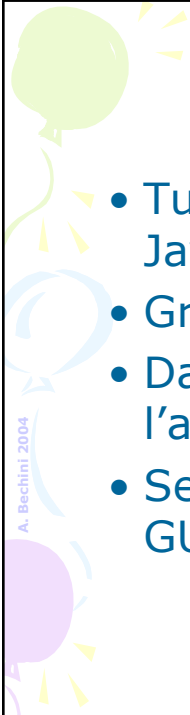
A. Bechini 2004



Client-side: Applet

- Una tecnologia per poter sfruttare appieno le capacità del client è quello delle Applet Java.
- Il browser deve contenere un interprete Java (JVM), e accedere alle API del linguaggio.
- Una Applet è un programma vero e proprio, con sofisticata interfaccia grafica, che viene collocato all'interno di una pagina HTML.

A. Bechini 2004



Applet: vantaggi

- Tutta la potenzialità del linguaggio Java
- Grafica spesso molto curata
- Dati possibilmente contenuti entro l'applicazione che viene scaricata
- Semplice riuso della Applet come GUI in applicazione standalone.



Applet: svantaggi

- Codice di dimensioni relativamente contenute (programmi piccoli)
- Tempo di download elevato
- ... ma soprattutto: insufficiente integrazione con l'ambiente HTML (l'interazione tra la Applet e documento è difficoltosa e limitata, se non addirittura impossibile)



Client side: scripting

- Eterogeneità delle piattaforme client
- Il modo più naturale per poter eseguire codice client-side è attraverso un interprete all'interno del browser
- Il "codice" prende la forma di scripting, inserito direttamente all'interno di documenti HTML.
- Speciali tag HTML devono supportare la presenza di script.
- Integrazione e interazione script/HTML

A. Bechini 2004



Browser e scripting

- Un browser può supportare diversi linguaggi di scripting, implementando i rispettivi interpreti.
- Linguaggi di scripting client-side più diffusi:
 - client-side JavaScript (Netscape)
 - VBScript (subset MS VisualBasic)

A. Bechini 2004



JavaScript

- JavaScript è il linguaggio di scripting client-side più utilizzato
- Permette di interagire con l'ambiente del Browser e con i documenti
- NON E' UN SUBSET DI JAVA
- Il nome è stato scelto per motivi di marketing: inizialmente si chiamava LiveScript
- Nelle versioni più recenti può operare sulla rappresentazione interna del documento (DOM di W3C)

A. Bechini 2004



Tecnologie server-side

- Se sul client non si possono fare ipotesi, il server può essere equipaggiato con HW e SW specifico
- I contenuti HTML statici possono essere trattati da un semplice HTTP server
- Le tecnologie server-side sono mirate alla generazione di contenuti HTML dinamici, e si affiancano alle funzionalità di base del web server

A. Bechini 2004



Server-side: CGI

- Storicamente, la prima tecnologia server-side è stata CGI (Common Gateway Interface).
- La generazione di contenuto dinamico è *esterna* al web server
- Il web server passa i parametri della richiesta a un programma esterno, che fornisce la pagina di risposta come suo output.
- Il programma esterno può essere sviluppato con qualsiasi linguaggio.



Server-side: scripting

- CGI, nella sua forma basilare, ha problemi di prestazioni e di manutenibilità.
- 1. La prima soluzione è stata l'introduzione dello scripting sul lato server
- 2. Una prima forma consiste in scripting esterno al web server
- 3. L'integrazione dello scripting nel web server permette ulteriori miglioramenti



Scripting: perché?

- Le pagine che compongono un'applicazione web sono in genere soggetti a una velocità di aggiornamento molto elevata
- La possibilità di evitare un lungo ciclo "edit-compile-link" *migliora la manutenibilità*
- Un interprete in background che esegue script CGI a fronte di request HTTP, *migliora le prestazioni* (rispetto ai semplici processi CGI)



L'esperienza di Perl

- Un linguaggio particolarmente usato per i programmi CGI è Perl
- Caratteristiche:
 - Linguaggio ad alto livello
 - Possibilità di strutturazione O-O
 - Interpretato (tipicamente)
 - Adatto per l'elaborazione di testo
 - Supporto per le espressioni regolari
 - Prodotto open-source



Servlets

- La necessità di migliorare ulteriormente le prestazioni ha spinto a una *integrazione* dell'elaborazione di script all'interno del server HTTP
- Un'esperienza decisiva è stata quella delle "Servlet": il web server è basato su una JVM, che esegue porzioni di codice Java a fronte di request HTTP
- Le attività del web server (p.es. Tomcat) vengono svolte utilizzando pesantemente il multithreading

A. Bechini 2004



ASP

- Per migliorare anche la manutenibilità, Microsoft ha proposto la tecnologia ASP (Active Server Pages)
- L'idea: affidare al contenuto HTML definizione della struttura della pagina, su cui poi intervenire con scripting (direttamente all'interno dell'HTML)
- ASP usa VBScript per questo scopo
- Gli strumenti tecnologici che seguono questo approccio sono noti come "Template systems"

A. Bechini 2004



JSP

- Analogamente ad ASP, JSP (Java ServerPages) è un template system.
- Il linguaggio di scripting utilizzato è Java (portabilità)
- Traggono vantaggio da tutta la ricchezza di Java e delle relative API
- Utilizzo del codice Java:
 - Diretto, all'interno delle pagine
 - Indiretto, accedendo componenti detti "JavaBeans"