

HTTP

Ing. Cesare Monti - 20 aprile 2005

organizzazione della conoscenza

- tramandare conoscenza è esigenza umana
 - forme orali
 - leggende
 - ... pensate a Omero e alla sua Odissea
 - forme scritte
 - manoscritti
 - biblioteche
 - l'uomo lo fa da secoli

organizzazione della conoscenza

- una forma per la rappresentazione come standard "de facto"
 - scrittura lineare
 - usata da millenni
 - deve esserci un filo logico che lega il discorso
 - legata al supporto fisico
 - rappresenta bene le forme organizzate del pensiero umano
 - ... che non sempre lo è!

organizzazione della conoscenza

- ... e qualora volessimo rappresentare a pieno il pensiero umano?
 - teoria dell'ipertesto
 - slegare la rappresentazione dal supporto fisico
 - catturare i nessi e non la sola informazione
 - focalizzarsi sull'informazione come oggetto

organizzazione della conoscenza

- tappe storiche
 - 1588 - "Le diverse et artificiosae macchine del Capitano Agostino Ramelli"
 - 1945 - Vannevar Bush - consigliere di Roosevelt
 - As we may think - Atlantic Monthly
 - Memex - sistema basato sui microfilm
 - 1965 - Theodor Holm Nelson
 - conio il termine ipertesto
 - Xanadu - dal luogo magico di "Kubla Khan" di S. Taylor

organizzazione della conoscenza

- 1990 - Tim Berners-Lee
 - World Wide Web
 - HTML
 - URLs
 - HTTP
- 1991/92
 - NCSA - Mosaic - primo browser

HTTP

- insieme di RPC (Remote Procedure Calls) basate su TCP/IP
- repository di risorse identificate da URLs
- ogni risorsa è un ipertesto
 - almeno inizialmente
- ad ogni richiesta corrisponde una nuova connessione

Il protocollo HTTP

- Si basa su due fasi:
 - Richiesta (HTTP:Request)
 - Risposta (HTTP:Response)
- Ognuna di queste fasi ha un suo possibile protocollo:
 - Request: {line, header, body}
 - Response: {header line, headers fields, body}

HTTP: Request

- Request line (nome del comando da invocare, es: GET)
- Request header field (informazioni aggiuntive, es: parametri di RPC)
- Entity body (riservato al passaggio di informazioni "bulk" al server)

HTTP:Request

```
- Sintassi:
<method><resource identifier><http version> <clrf>      Request line
[<Header> : <value>] <clrf>                               Request
: :                                                       Header
: :                                                       Fields
[<Header> : <value>] <clrf>
blank line <clrf>                                       Entity Body
[Entity body]
```

```
- Esempio:
GET /path/to/file.html HTTP/1.0      Request line
Accept: text/html                    Request
Accept: audio/x                      Header
Accept: image/gif                    Fields
User-agent: MacWeb
```

HTTP: Request

- di Method ne esistono diversi !
 - get
 - richiesta semplice
 - head
 - = get ma restituisce solo le intestazioni
 - post
 - chiede di accettare i dati in stream
 - put
 - chiede di memorizzare i dati sulla risorsa URI specificata
 - delete
 - chiede di cancellare la risorsa URI specificata
 - options
 - chiede quali methods siano supportati dal server web
 - trace
 - chiede di mostrare la request http e le sue intestazioni
 - connect
 - ...mai implementato per l'utilizzo di proxy per operazioni di tunneling

HTTP : Response

- Response Header Line (protocollo e numero di errore)
- Response Header Field (informazioni aggiuntive, contenuto, lunghezza ecc ...)
- Entity body (il corpo della pagina richiesta)

HTTP: Response

- Sintassi:

```
<HTTP Version> <result code> [<explanation>] <clrf> } Response line
[<Header> : <value>] <clrf> } Response
: : } Header
: : } Fields
[<Header> : <value>] <clrf>
blank line <clrf>
[Entity body] } Entity Body
```

- Esempio:

```
HTTP/1.0 200 OK } Response line
SERVER: NCSA/1.3 } Response
Mime_Version: 1.0 } Header
Content_Type: text/html } Header
Content_Length: 2000 } Fields
<HTML> } Entity Body
:
:
</HTML>
```

HTTP: limiti

- tra ogni richiesta non c'è interazione
- completa perdita di stato ad ogni connessione
 - stateless
 - questo deriva dalla sua progettazione
- ogni RPC è slegata dalle altre
 - non c'è modo di mantenere la storia dell'interazione
 - decade il legame spazio-tempo

HTTP Sistemi Distribuiti

- Per ovviare al problema negli anni sono stati sviluppati differenti modelli di interazione:
 - spostare capacità di calcolo sul server
 - sistemi a mainframe
 - spostare capacità di calcolo sul client
 - reti di PC
 - Modelli ibridi
 - architetture client-server

Sistemi Distribuiti

- Il modello Client - Server oggi è il più diffuso
 - c'è computazione sul client
 - ... e sul server
- nel mondo WEB si traduce in tecnologie abilitanti
 - client
 - javascript
 - ECMAScript
 - actionscript
 - server
 - CGI
 - Application Server