

RELAZIONE TECNOLOGIE WEB

Chen Hao Lorenzo (146787)

La maggior parte dei servizi che ci vengono offerti sul Web, segue l'architettura Client-Server, nella quale possiamo individuare due entità fondamentali: il Client, che si interfaccia con l'utente che intende richiedere un servizio, il Server che offre questo servizio e restituisce le informazioni cercate al Client, in modo che l'utente possa visualizzare l'output finale. La comunicazione tra queste 2 parti può avvenire in diversi modi: abbiamo infatti il metodo Pull dove il Client si blocca attendendo la risposta dal Server (modalità sincrona) oppure controlla periodicamente se il Server gli ha trasmesso la risposta (modalità asincrona) e il metodo Push in cui il Client non effettua alcuna interruzione e il Server invia le informazioni richieste appena esse sono pronte. Alla base di questa interazione abbiamo dei protocolli tra cui il protocollo HTTP, HTTP/2 (che approfondirò in seguito) oppure i Web Socket.

Per comprendere meglio la tecnologia Client-Server abbiamo trattato nei suoi aspetti più importanti il protocollo HTTP, acronimo di HyperText Transfer Protocol, che regola lo scambio di messaggi (rispettivamente la richiesta e la risposta). Il protocollo HTTP fa uso degli URI, per ottenere le risorse di cui ha bisogno (si dividono in URL, che identificano in modo diretto una risorsa ma non sono necessariamente permanenti, e URN, che identificano definitivamente una risorsa ma devono essere trasformati da un apposito servizio in URL). Tra i principali difetti del protocollo HTTP abbiamo il fatto che esso è stateless, non esiste cioè nessuna dipendenza tra una richiesta e le precedenti: questo comporta la necessità di dover includere tutte le informazioni ogni volta; inoltre, per ogni connessione TCP può essere elaborata solo una richiesta alla volta.

Per quanto riguarda il codice tramite il quale vengono scritte le pagine web si fa utilizzo, nella maggior parte dei casi, del linguaggio HTML (linguaggio a marcatori) che rappresenta le informazioni all'utente. In particolare abbiamo pagine statiche, cioè non modificabili e che contengono solamente codice HTML, e pagine dinamiche che contengono anche istruzioni scritte in linguaggi Server-Side (il più celebre è PHP). La sintassi del linguaggio HTML è molto semplice: ogni sezione/elemento è contrassegnato da una coppia di tag (uno di apertura e uno di chiusura), ciascun tag ammette poi delle ulteriori specificazioni dati dai relativi attributi.

Possiamo inoltre migliorare l'aspetto grafico dei documenti HTML grazie a dei fogli di stile che possono essere inclusi nei nostri file (scritti in linguaggio CSS).

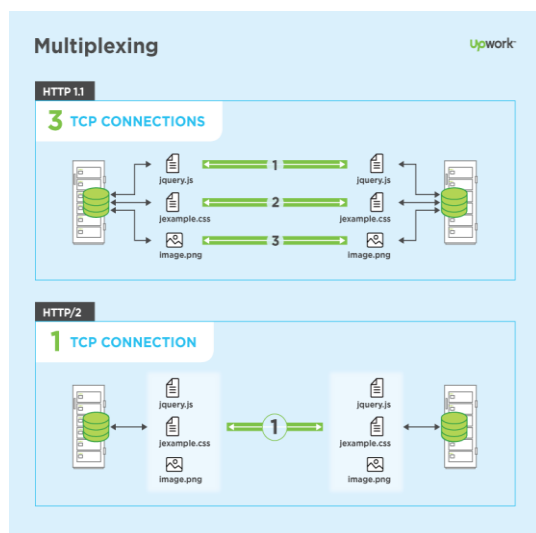
Come già accennato prima, per la creazione di pagine dinamiche il solo codice HTML non è sufficiente ma è necessario un linguaggio di scripting come PHP, che ha una sintassi simile al C. Dalla versione 3.0 è orientato agli oggetti e permette infatti la creazione di classi con la definizione di relativi metodi; sono inoltre possibili l'ereditarietà, il polimorfismo e l'incapsulamento, come, ad esempio, in Java. Il codice PHP può essere facilmente incluso in una pagina HTML tramite l'uso di semplici tag.

Infine, abbiamo trattato il Sistema di Controllo di Versione (VCS secondo l'acronimo inglese), che permette di tenere traccia delle modifiche che nel tempo abbiamo apportato ai nostri file. Il software più diffuso a tal scopo è git.

APPROFONDIMENTO HTTP/2

HTTP/2 rappresenta l'evoluzione del protocollo HTTP, senza però modificarne la semantica e i concetti fondamentali, agendo invece sulla formattazione e sul trasporto dei dati.

Si basa sul protocollo applicativo SPDY, ideato da Google nel 2009 per il trasporto di contenuti web con riduzione della latenza in un'ottica di ottimizzazione del trasferimento tra Server e Browser (si voleva, ad esempio, ridurre il tempo di caricamento di una pagina). Da quel momento, fino ad arrivare al 2015, SPDY e HTTP/2 si sono sviluppati in parallelo per garantire altri vantaggi quali la compressione dei dati degli header HTTP grazie a un nuovo livello di framing binario



(diminuisce quindi il consumo di banda) e l'utilizzo di tecnologie push: il Server infatti invia più dati di quelli inizialmente richiesti senza attendere poi ulteriori richieste aggiuntive da parte del Client ed è possibile il multiplexing (richieste e risposte multiple gestite in parallelo), tramite l'utilizzo di più flussi bidirezionali (denominati stream) nei

quali vengono inviati i pacchetti (si noti che ogni pacchetto è dotato di un identificatore di flusso nella sua intestazione per poter permettere il riassettaggio); gli stream possono essere inoltre dotati di meccanismi di priorità e dipendenze da altri stream.

Come fonti sono state utilizzate le slide del corso e alcune risorse del Web come Wikipedia, serverplan.