

MODELLO CLIENT/SERVER

Gianluca Daino

Dipartimento di Ingegneria dell'Informazione

Università degli Studi di Siena

daino@unisi.it

POSSIBILI STRUTTURE DEL SISTEMA INFORMATIVO

La struttura di un sistema informativo può essere basata su due filosofie diverse :

- **Mainframe** : in questa struttura una macchina (mainframe) effettua tutte le operazioni sui dati, memorizza i dati e li distribuisce all'utente. L'utente ha a disposizione un terminale *stupido* nel senso che non richiede capacità elaborative, ma viene usato soltanto come unità di visualizzazione e di inserimento dati.

- **Client/Server :**

  Il sistema è formato da due tipi di moduli : il client e il server, che generalmente sono eseguiti su macchine diverse collegate in rete.

  Il server svolge le operazioni necessarie per realizzare un servizio; ad esempio gestisce le banche dati, gestisce l'aggiornamento dei dati e la loro integrità,....

  Il client può effettuare alcune operazioni e quindi richiede un terminale con capacità elaborative (generalmente un PC). Tipicamente il client gestisce la porzione di interfaccia utente dell'applicazione, verifica i dati inseriti e provvede ad inviare al server le richieste formulate dall'utente. Inoltre gestisce le risorse locali, come la tastiera, il monitor, la CPU, e le periferiche. In pratica il client è quella parte dell'applicazione che l'utente vede e con la quale interagisce.

  L'affermazione di questo modello è legata alla disponibilità di reti locali a basso costo ed alla diffusione della rete Internet, in cui i servizi seguono tale struttura.

Architettura CLIENT/SERVER

La maggior parte dei servizi telematici offerti da Internet si basano su una particolare modalita' di interazione denominata tecnicamente architettura client-server. Tale nome indica un'architettura software che e' costituita da due moduli integrati ma distinti, residenti generalmente su calcolatori diversi.



Cos'e' un client/server computing?

In un ambiente client/server, sul computer client e' in esecuzione un software applicativo (**programma client**).

Il programma client :

- Abilita l'utente a spedire una richiesta di informazione al server.
- Formatta la richiesta in modo che il server possa capirla.
- Formatta la risposta del server in modo che l'utente possa leggerla



Cos'e' un client/server computing?

In un ambiente client/server, sul computer server viene eseguito un software applicativo chiamato programma server.

Il programma server :

- Riceve una richiesta da un client e processa la richiesta
- Risponde, spedendo l'informazione richiesta, al client

Client/server

- La logica sottesa al modello di comunicazione client-server è presentata nella figura che segue:





Client/server

- Affinché l'interazione tra client e server possa essere effettuata, è necessario che entrambi utilizzino un linguaggio comune, ovvero un protocollo applicativo.
- Su Internet vengono utilizzati numerosi protocolli specifici delle applicazioni, uno per ogni servizio di rete:
- **Esempi di protocolli:**
  - *Simple Mail Transfer Protocol* (*SMTP*) per la posta elettronica;
  - *File Transfer Protocol* (*FTP*) per il trasferimento di file tra host;
  - Il protocollo su cui si basa World Wide Web, denominato *Hyper-Text Transfer Protocol* (*HTTP*);
  - Ovviamente tutti questi protocolli applicativi debbono appoggiarsi sui protocolli di rete TCP/IP e sul DNS per poter effettivamente scambiare richieste e messaggi attraverso la rete.



Il modello client-server

- Il termine *server* è applicabile a qualsiasi programma che offra un servizio accessibile su una rete. Un server accetta una richiesta sulla rete, esegue il suo servizio e restituisce il risultato al mittente
- Per i servizi più semplici, ogni richiesta arriva in un unico datagram IP e il server restituisce una risposta in un altro datagram.
- Un programma in esecuzione diventa un *client* quando invia una richiesta a un server e aspetta una risposta



Un esempio semplice: un server di eco UDP

- Un server di eco UDP ottiene dal suo sistema operativo il permesso di usare la porta di eco UDP.
- Una volta ottenuto il permesso, il processo del server di eco entra in un ciclo infinito che consta di tre fasi:
  - (1) aspettare che arrivi un datagram alla porta dell'eco,
  - (2) invertire le posizioni degli indirizzi sorgente e di destinazione (compresi gli ID delle porte UDP)
  - (3) restituire il datagram al suo mittente originale.
- Nel sito interlocutore gira un programma detto *client di eco* UDP. Questo programma richiede una porta UDP inutilizzata, invia un messaggio UDP al server di eco e aspetta la risposta.
- Il client si aspetta di vedersi restituire esattamente gli stessi dati che ha inviato.



In pratica …

- Il server inizia l'esecuzione prima che cominci l'interazione e (di solito) continua ad accettare richieste e inviare risposte senza mai smettere.
- Il client è un qualsiasi programma che invia una richiesta e aspetta una risposta; tipicamente termina dopo avere usato un server un numero finito di volte.
- Il server aspetta le richieste dal client a una porta ben nota che è stata riservata per il servizio che offre.



Server side

- I programmi server di solito sono molto più difficili da scrivere dei client, perché devono gestire più richieste simultaneamente, soprattutto se serve molto tempo per elaborare una singola richiesta
- Tipicamente questi server sono costituiti da due parti: il programma principale o *primario*, che ha il compito di accettare nuove richieste e un insieme di moduli ausiliari o *secondari* che si occupano della gestione di singole richieste.



Server side

- Il primario esegue le seguenti cinque fasi:
  - **Apri porta**: Il modulo primario apre la porta ben nota a cui può essere raggiunto;
  - **Attendi richiesta:** Il modulo primario aspetta che un client gli invii una richiesta;
  - **Scegli porta**: Se necessario, il modulo primario ottiene una nuova porta locale per elaborare la richiesta e ne informa il client; questa fase è inutile con TCP;
  - **Attiva secondario**: Il modulo primario attiva un modulo secondario indipendente e simultaneo per gestire la richiesta;
  - **Continua:** Il modulo primario ritorna alla fase di attesa e continua ad accettare nuove richieste mentre il secondario appena creato gestisce la richiesta precedente.

Il modulo secondario gestisce una richiesta e poi termina senza aspettare richieste da altri client.



Alternative al modello client-server

Peer-to-peer o servent: tutti i programmi agiscono sia da client sia da server. Es: i programmi di scambi audio e video (WinMx, Kazaa, eMule, ecc.)

Raccolta anticipata o caching; il programma client raccoglie periodicamente tutte le informazioni potenzialmente necessarie e le memorizza localmente



MODELLO CLIENT/SERVER

Gianluca Daino

Dipartimento di Ingegneria dell'Informazione

Università degli Studi di Siena

daino@unisi.it