

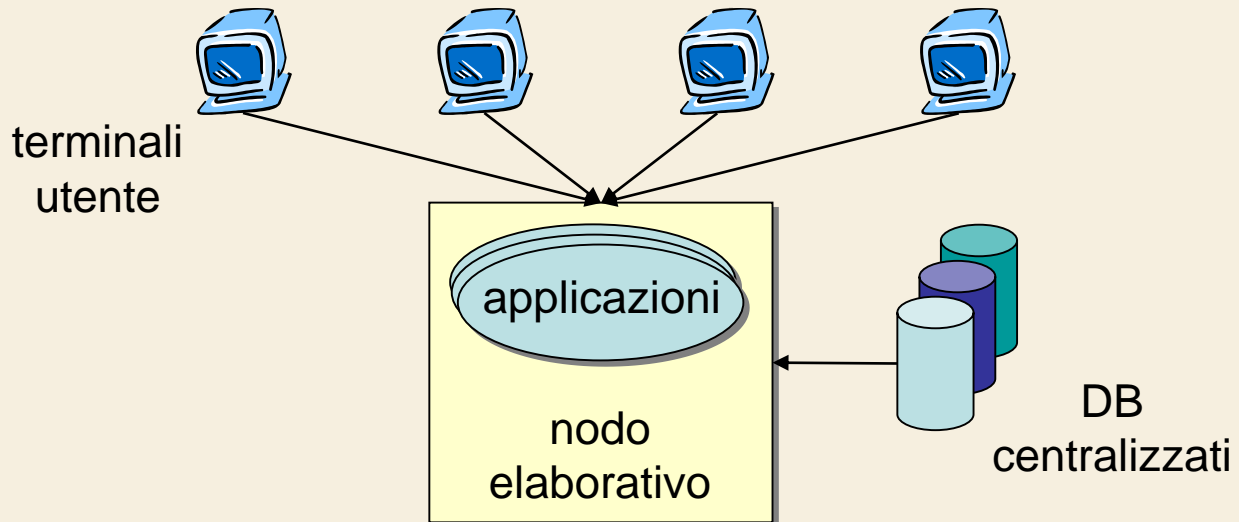


Basi di Dati

Architetture Client/Server

Architettura centralizzata

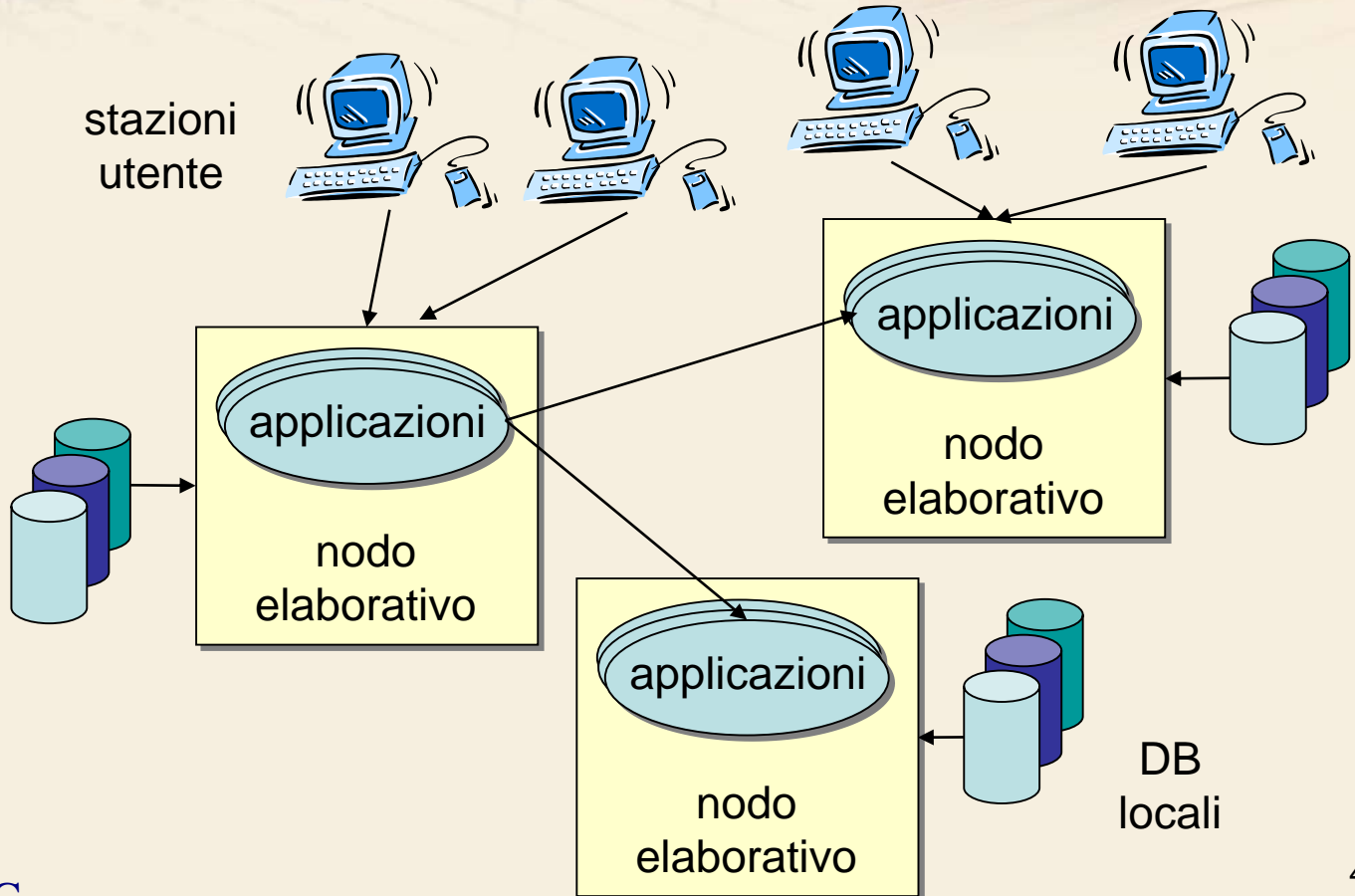
- Un'architettura è centralizzata quando i dati e le applicazioni (programmi) risiedono in un unico nodo elaborativo
- Tutta l'intelligenza è nell'elaboratore centrale (host o mainframe)



Architettura distribuita

- Un'architettura è distribuita quando si verifica almeno una delle due condizioni
- Le applicazioni, fra loro cooperanti, risiedono su più nodi elaborativi: elaborazione distribuita
 - Il patrimonio informativo (dati), unitario, è ospitato su più nodi elaborativi: base di dati distribuita

Architettura distribuita



Architettura distribuita

- È un insieme di nodi elaborativi connessi in qualche modo
 - LAN - Local Area Network, WAN - Wide Area Network, ...
- Ciascun nodo è in grado di comunicare con alcuni degli altri nodi
- I programmi in esecuzione su ciascun nodo sono in grado di condividere le proprie informazioni e di richiedere l'esecuzione di altri programmi da parte di altri nodi

Architettura distribuita

- Diversi livelli di complessità
 - Dipende dal livello di indipendenza dei nodi
- Vantaggi
 - Miglioramento delle prestazioni
 - Maggiore disponibilità
 - Migliore affidabilità

Proprietà importanti

➤ Portabilità

- Capacità di spostare un'applicazione da un sistema ad un altro sistema
 - Garantita dallo standard SQL per la componente "dati"
 - Garantita dal tipo di linguaggio di programmazione usare per la componente "applicativa"

➤ Interoperabilità

- Capacità di DBMS diversi di cooperare nell'esecuzione di un determinato compito
- Sono necessari dei protocolli di interazione
 - ODBC
 - X-Open-DTP

Architettura distribuita

- Le applicazioni sono caratterizzate dal ruolo che svolgono nel sistema
- Cliente (Client): quando l'applicazione utilizza dei servizi messi a disposizione da altre applicazioni
 - Servente (Server): quando l'applicazione fornisce servizi usati da altre applicazioni
 - Attore (Actor): quando l'applicazione in contesti diversi assume sia il ruolo di client sia il ruolo di server

Il paradigma client-server

- Client: utente del servizio
- Server: fornitore del servizio
- Client e server non hanno senso senza un protocollo di comunicazione che
 - definisca le possibili interazioni fra client e server
 - specifichi i dettagli di ogni interazione
 - definisca le condizioni di errore e le azioni da svolgere in conseguenza

Esempi di protocollo

Servizio	Client	Protocollo
World Wide Web	Mozilla Firefox, Internet Explorer, Google Chrome, Opera, ...	Hypertext Transfer Protocol (HTTP)
Trasferimento di file	UNIX FTP, MS FTP, browser Web	File Transfer Protocol (FTP)
Posta elettronica	Eudora, Outlook, browser Web	Simple Mail Transfer Protocol (SMTP)

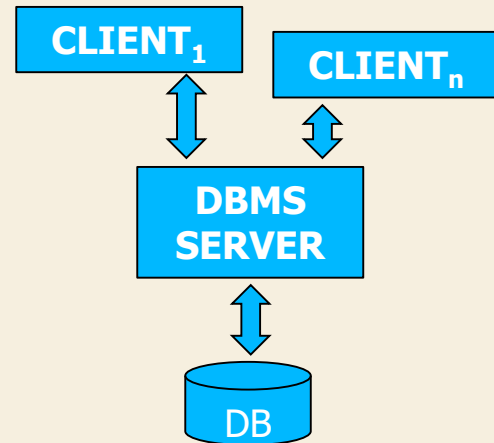
Architettura client-server

- Caso particolare di sistema distribuito
 - L'architettura più semplice e più diffusa
- Un client invia una richiesta ad un server per l'esecuzione di un compito (task)
 - Un task può consistere semplicemente nel richiedere delle informazioni, o nell'eseguire elaborazioni complesse
- Il server può essere a sua volta il client di un altro servizio

Architettura client-server

➤ Architettura a 2 livelli (2-Tier)

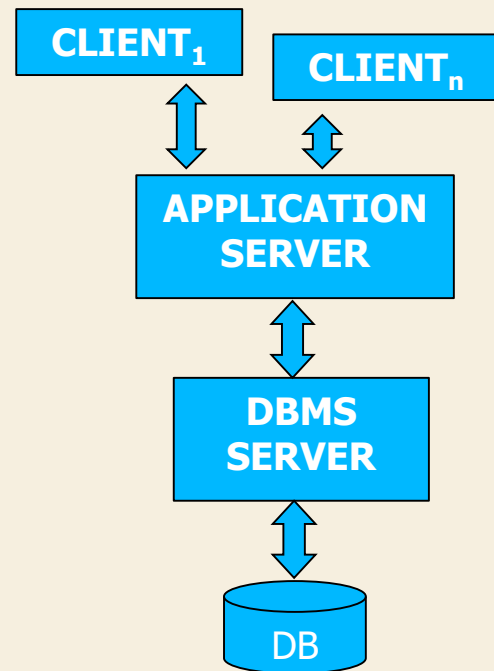
- *Thick* client
 - contiene della logica applicativa
- DBMS server
 - consente l'accesso ai dati



Architettura client-server

➤ Architettura a 3 livelli (3-Tier)

- *Thin* client
 - il browser
- Application server
 - implementa la logica applicativa
 - solitamente è anche un web server
- DBMS Server
 - consente l'accesso ai dati



➤ Compile & Go

- La query viene inviata al server
- La query viene compilata
 - generazione del piano di esecuzione
- La query viene eseguita
- Il risultato viene restituito

➤ Efficace per l'esecuzione di query non ripetitive

- Flessibile, adatta all'esecuzione di SQL dinamico

➤ Compile & Store

- La query viene inviata al server
- La query viene compilata
 - generazione del piano di esecuzione
 - il piano di esecuzione viene memorizzato per un uso futuro
- La query viene eseguita
- Il risultato viene restituito

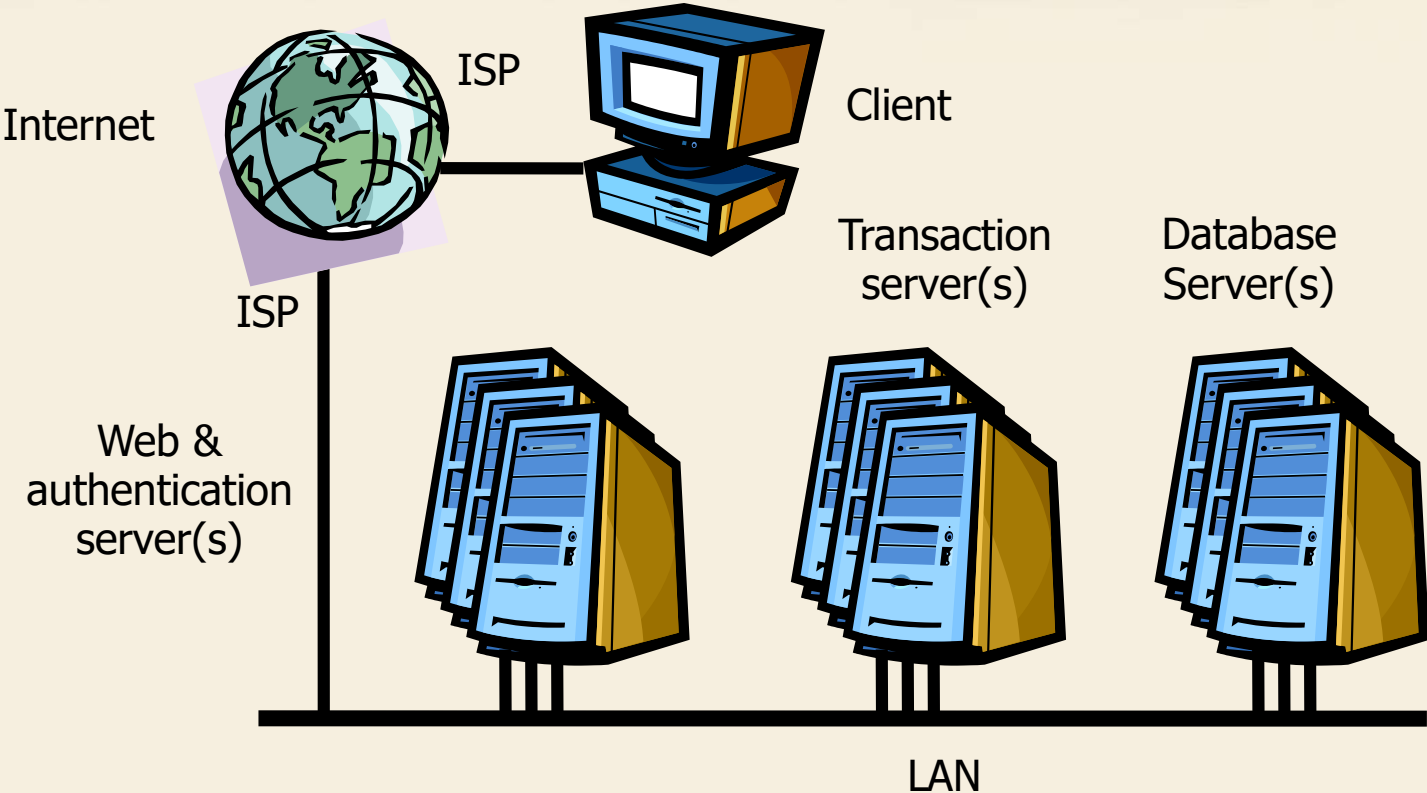
➤ Efficiente per query ripetute

- Esecuzioni parametriche della stessa query

Architettura C/S: il World Wide Web

- Il Web è una ragnatela (grafo) di documenti
- Il server mantiene localmente un insieme di documenti e su richiesta invia al client (browser Web) un documento
 - testo, immagine, suono, filmato, ...
- Alcuni tipi di pagine chiedono al server di eseguire un programma (es. script PHP)

II World Wide Web





Basi di Dati

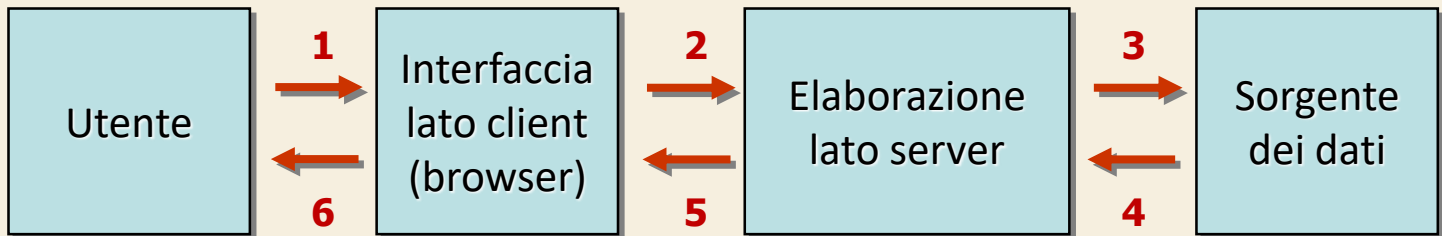
Le applicazioni Web

Workflow delle applicazioni Web

L'utente definisce una query tramite la query interface

La query è inviata all'agente sul lato server

L'agente risponde alla query utilizzando la sorgente dei dati



Il risultato della query è mostrato all'utente

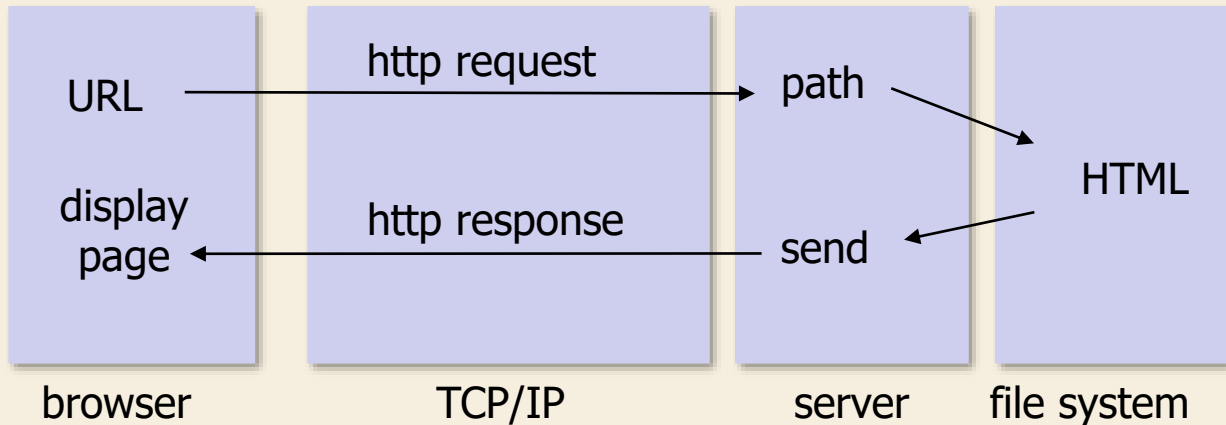
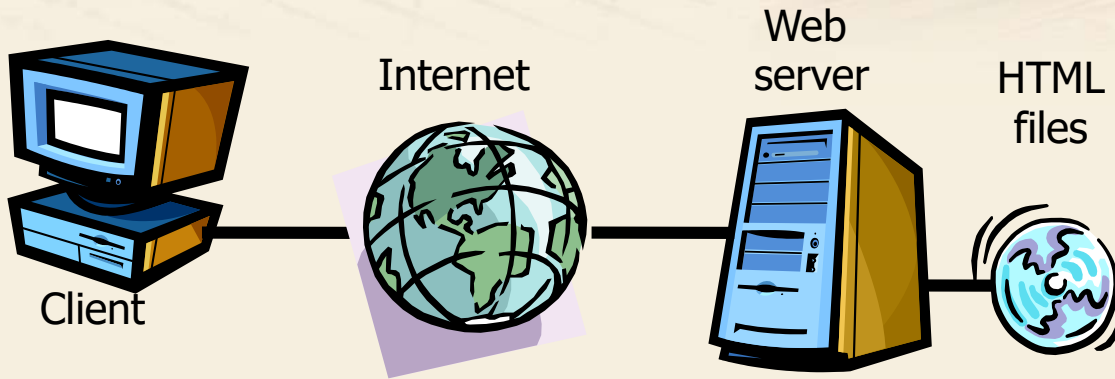
L'agente sul lato server restituisce il risultato della query

La sorgente dei dati restituisce il risultato della query

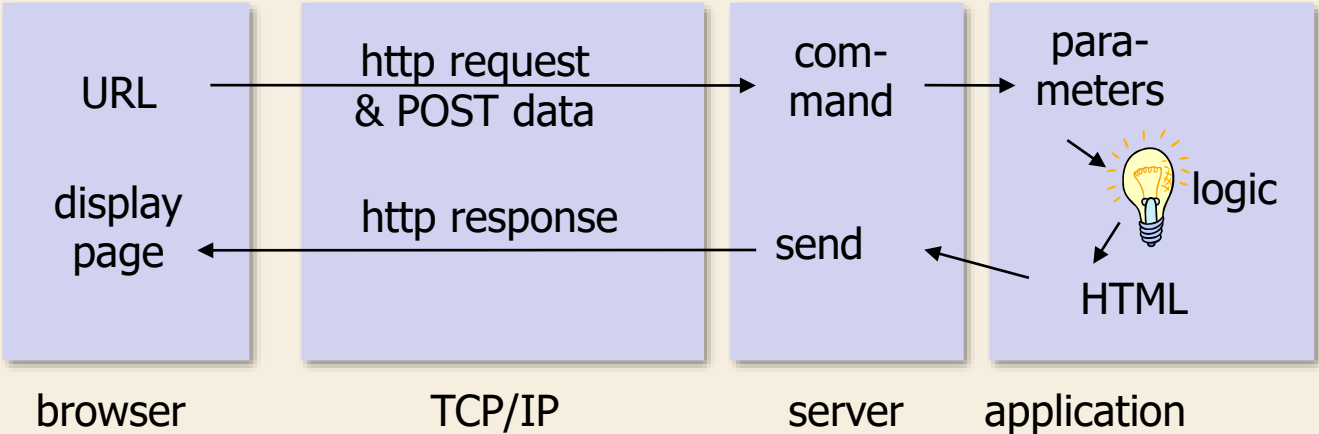
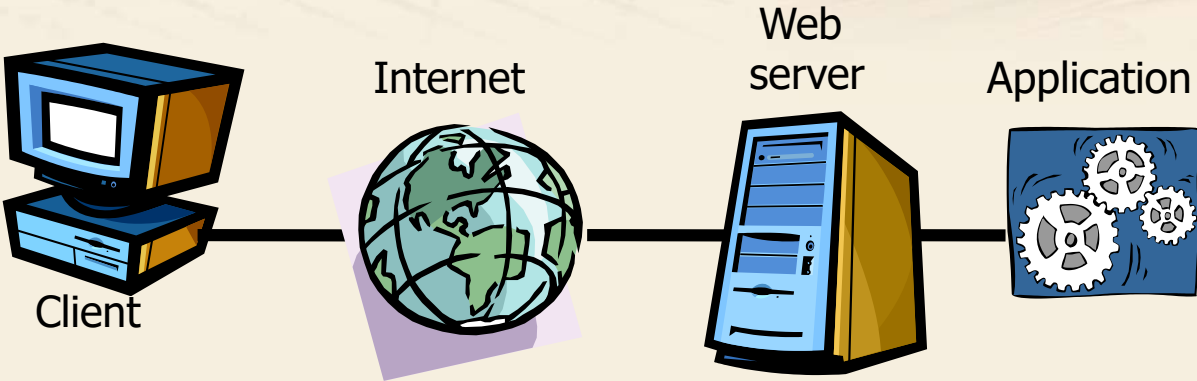
Workflow delle applicazioni Web

- Tutte le applicazioni Web fanno all'incirca le stesse cose:
- Forniscono un'interfaccia per la richiesta di informazioni (query interface); es: ricerca in un database, richiesta di un file, acquisto di un libro, prenotazione di un volo aereo, ...
 - Trasmettono al Web server i dati dell'utente
 - Eseguono l'elaborazione dei dati sul lato server (server-side processing) eventualmente accedendo ad un database
 - Trasmettono al client i risultati delle query

Transazione statica



Transazione dinamica



Transazione appoggiata su database

