

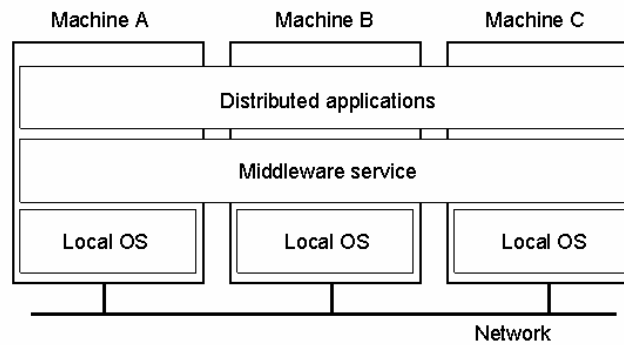
# Introduzione ai Sistemi Distribuiti

## Definizione di Sistema Distribuito (1)

Un **sistema distribuito** è:

Una collezione di computer  
indipendenti che appaiono  
agli utenti come un  
sistema singolo coerente.

## Definizione di Sistema Distribuito (2)



Un sistema distribuito organizzato come middleware.  
(Si noti che il middleware layer si estende su tutte le macchine.)

## Esempi di Sistemi Distribuiti

- Una rete di workstations.
- Una rete Peer-to-Peer.
- Un sistema di prenotazione di voli.
- Il World Wide Web.
- Una Griglia computazionale

## Obiettivi dei Sistemi Distribuiti

- Connettere utenti e risorse.
- Offrire Trasparenza.
- Apertura al mondo esterno.
- Realizzare sistemi scalabili.
- Maggiore tolleranza ai guasti.

## Trasparenza in un Sistema Distribuito

| Trasparenza         | Descrizione   |
|---------------------|---|
| <b>Accesso</b>      | Nasconde le differenze nella rappresentazione dei dati e su come una risorsa è acceduta   |
| <b>Locazione</b>    | Nasconde dove una risorsa si trovi  |
| <b>Migrazione</b>   | Nasconde che una risorsa si possa spostare in una differente locazione                    |
| <b>Rilocazione</b>  | Nasconde che una risorsa si possa spostare in una differente locazione mentre viene usata |
| <b>Replicazione</b> | Nasconde che una risorsa possa essere replicata   |
| <b>Concorrenza</b>  | Nasconde che una risorsa possa essere condivisa da più utenti concorrenti                 |
| <b>Fallimenti</b>   | Nasconde il guasto e il recovery di una risorsa (prossima slide)                          |
| <b>Persistenza</b>  | Nasconde se una risorsa (software) è in memoria o nel disco.                              |

Forme differenti di trasparenza in un sistema distribuito.

## Sistema Distribuito

Definizione di Lamport :

*“tu sai di averne uno quando il guasto di un computer di cui non hai mai sentito parlare non ti permette di fare il tuo lavoro.”*

## Problemi di Scalabilità

| Concetto                              | Esempio   |
|---------------------------------------|---|
| <b><i>Servizi Centralizzati</i></b>   | Un solo server per tutti gli utenti                   |
| <b><i>Dati Centralizzati</i></b>      | Un singolo elenco telefonico on line                  |
| <b><i>Algoritmi Centralizzati</i></b> | Effettuare il routing basato su informazioni complete |

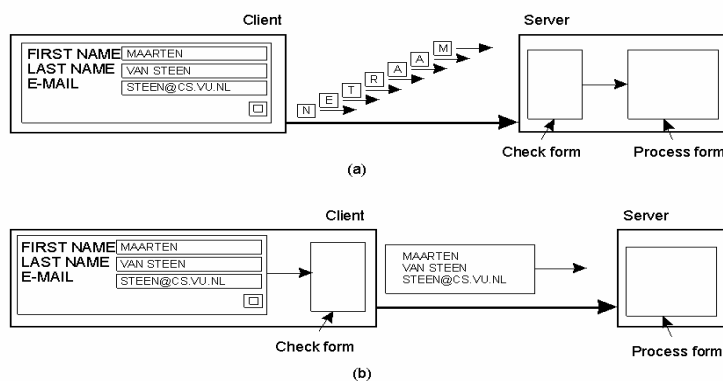
Esempi di limitazione di scalabilità

## Algoritmi Decentralizzati

- **Nessuna macchina ha l'informazione completa sullo stato del sistema**
- Le macchine prendono decisioni sulla base di informazioni locali.
- Il fallimento di una macchina non fa fallire l'algoritmo.
- Non vi è una assunzione implicita di un clock globale.

**Più grande è il sistema, maggiore è l'incertezza.**

## Tecniche di Scalabilità (1)



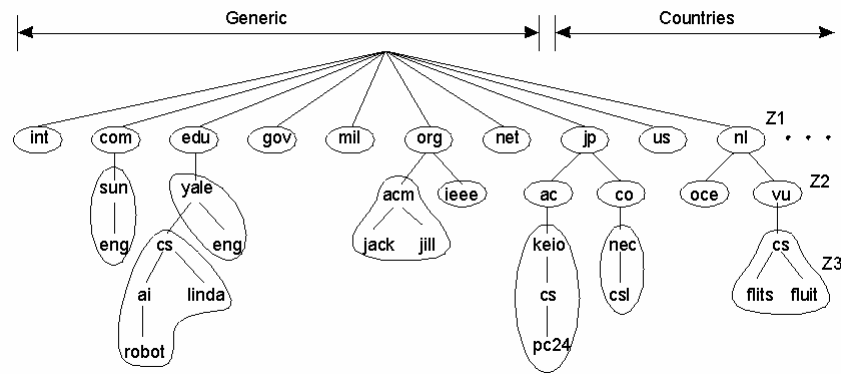
La differenza tra chi tra:

a) un server o

b) un client

controlla le forme che sono state riempite.

## Tecniche di Scalabilità (2)



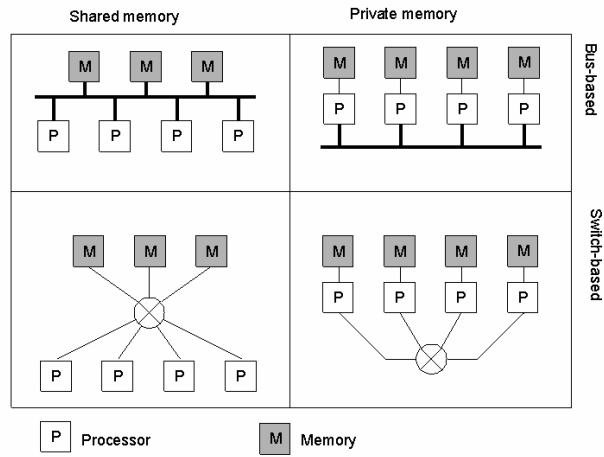
Un esempio di divisione dello spazio dei nomi del DNS in zone

## Tipi di Sistemi Distribuiti

- Dual/Multi processor (?)
- Cluster Computers
- Sistemi Informativi Distribuiti
- Reti Peer-to-Peer
- Sistemi di Grid Computing
- Sistemi di Mobile Computing
- Sistemi Pervasivi

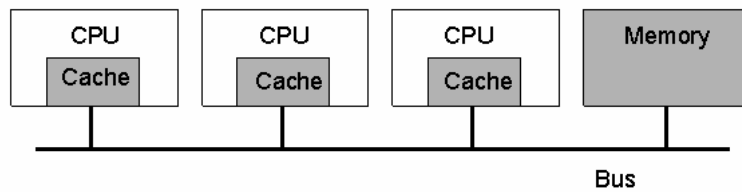


# Concetti Hardware



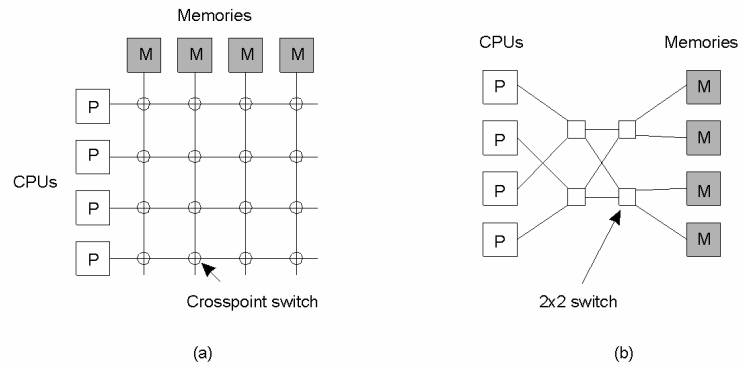
Differenti architetture e memorie in sistemi di elab. distribuiti

# Multiprocessori (1)



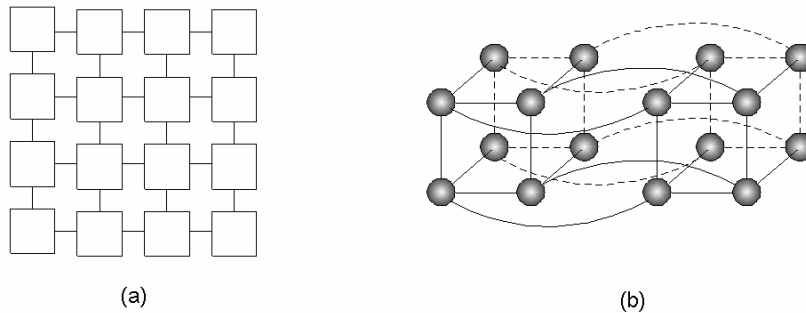
Un multiprocessore basato su architettura a bus.

## Multiprocessors (2)



- a) Un crossbar switch
- b) Una rete omega

## Multicomputer Systems Omogenei



- a) Mesh
- b) Hypercube



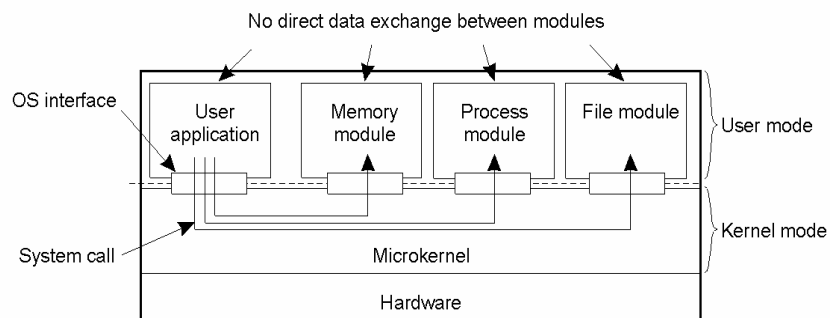
## Concetti Software

| Sistema           | Descrizione   | Obiettivo                                       |
|-------------------|---|---|
| <b>DOS</b>        | Sistemi operativi integrati per multi-processori e multi-computer omogenei        | <i>Nascondere e gestire le risorse hardware</i> |
| <b>NOS</b>        | Sistemi operativi lascamente accoppiati per multi-computer eterogenei (LAN e WAN) | <i>Offrire servizi locali a clienti remoti</i>  |
| <b>Middleware</b> | Livello aggiuntivo sul NOS per implementare servizi general-purpose               | <i>Fornire trasparenza di distribuzione</i>     |

Tre approcci differenti:

- DOS (Distributed Operating Systems)
- NOS (Network Operating Systems)
- Middleware

## Sistemi Operativi per Mono-processore

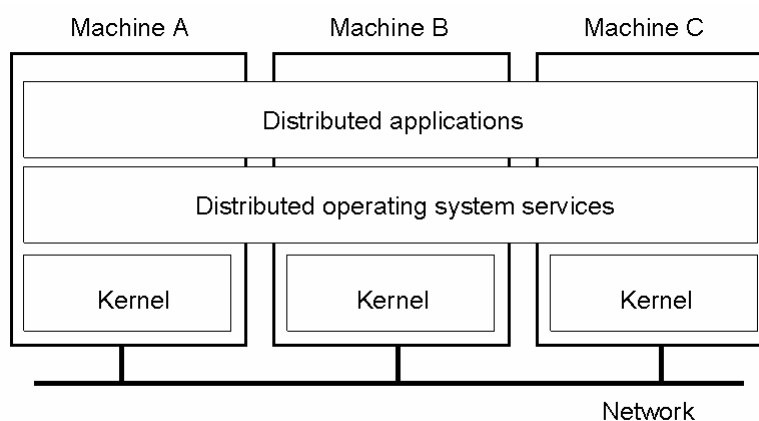


Separazione delle applicazioni dal codice del sistema operativo tramite un microkernel.

## Sistemi Operativi per Multi-processori

- Uso di più di una CPU e di una memoria condivisa tra le CPU.
- Gestione di accessi concorrenti alla memoria condivisa.
- Meccanismi: Semafori e Monitor per proteggere gli accessi concorrenti
- Semplice estendere i sistemi operativi per mono-processori sviluppati come un insieme di processi concorrenti.

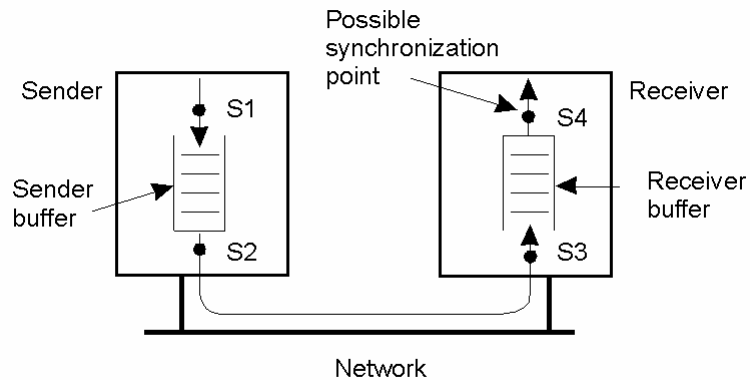
## Sistemi Operativi per Multi-computer (1)



Struttura generale di un sistema operativo per multi-computer

## Sistemi Operativi per Multi-computer (2)

Per la comunicazione è usato il Message Passing



Alternative per blocking e buffering in message passing.

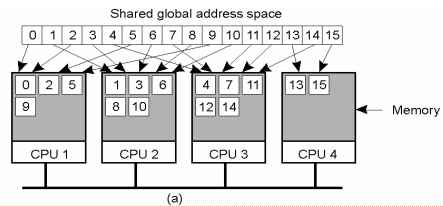
## Sistemi Operativi per Multi-computer (3)

| Punto di Sincronizzazione                                  | Send Buffer? | Comunicazione Affidabile garantita? |
|--|--------------|-------------------------------------|
| Il mittente si blocca quando il buffer è pieno             | Si           | Non necessaria                      |
| Il mittente si blocca finchè il messaggio non è inviato    | No           | Non necessaria                      |
| Il mittente si blocca finchè il messaggio non è ricevuto   | No           | Necessaria                          |
| Il mittente si blocca finchè il messaggio non è consegnato | No           | Necessaria                          |

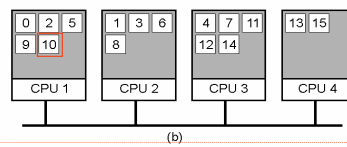
Relazione tra blocking, buffering, e comunicazioni affidabili.

## Distributed Shared Memory (1)

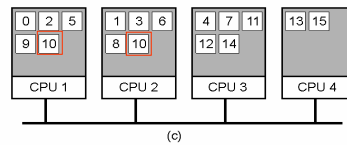
- a) Pagine dello spazio degli indirizzi distribuite tra 4 macchine



- b) Situazione dopo che la CPU 1 riferisce la pagina 10

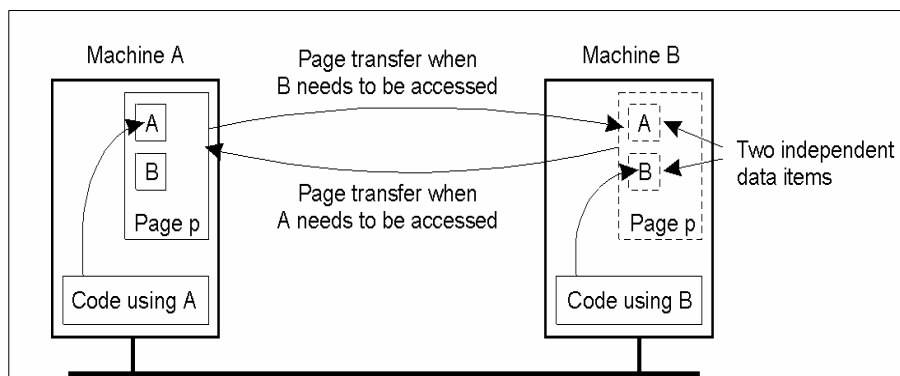


- c) Situazione se la pagina 10 è *read only* ed è usata la replicazione



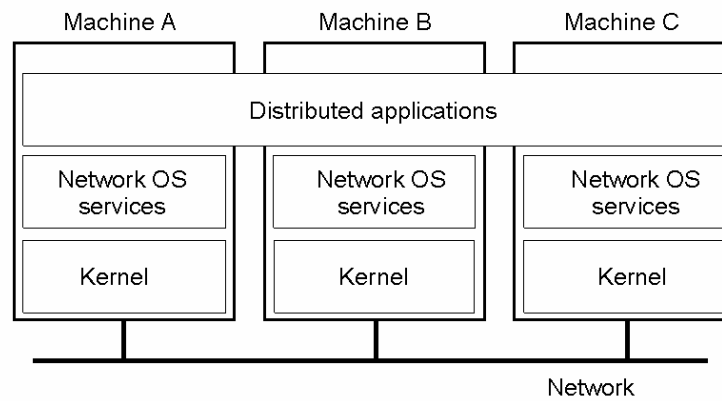
## Distributed Shared Memory (2)

Quale dimensione per le pagine ?



**False sharing** di una pagina tra due processi indipendenti

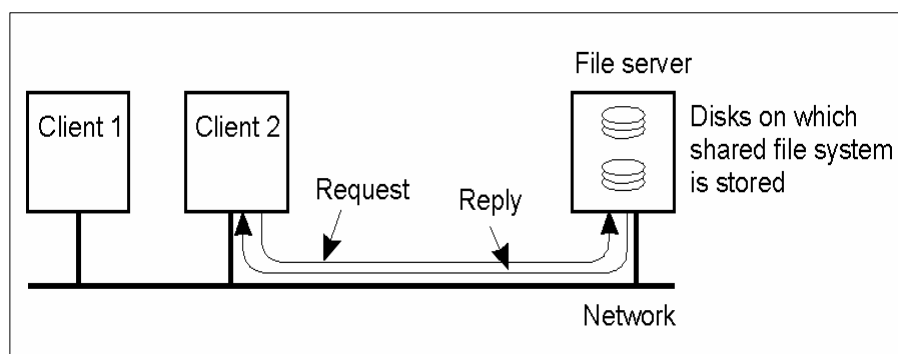
## Network Operating System (1)



Struttura generale di un NOS

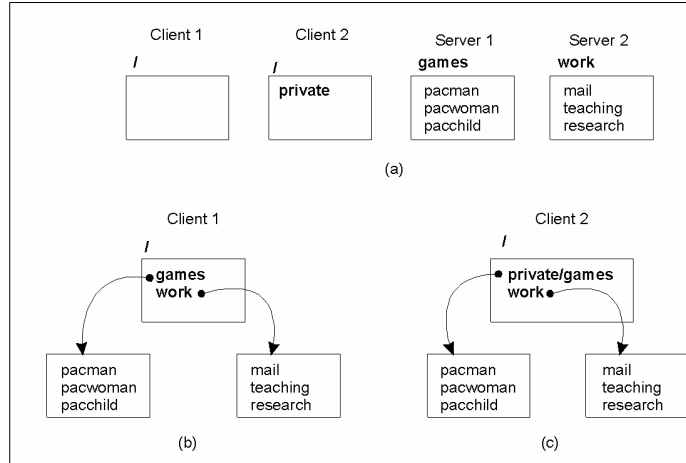
Esempi di comandi: *rlogin*, *rcp*, *rsh*

## Network Operating System (2)



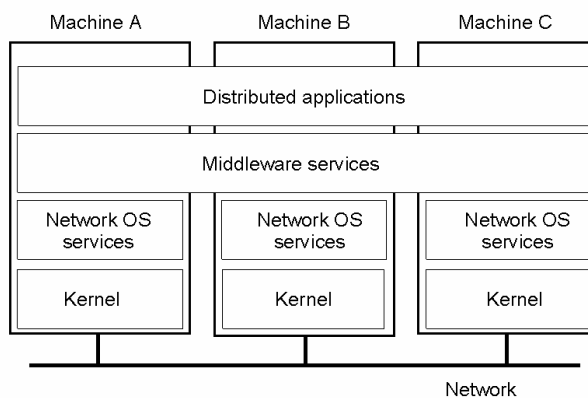
Due clienti e un server in un Network Operating system

## Network Operating System (3)



Differenti clienti possono montare i server in posti differenti, avendo così una vista differente del file system.

## Posizionamento del Middleware

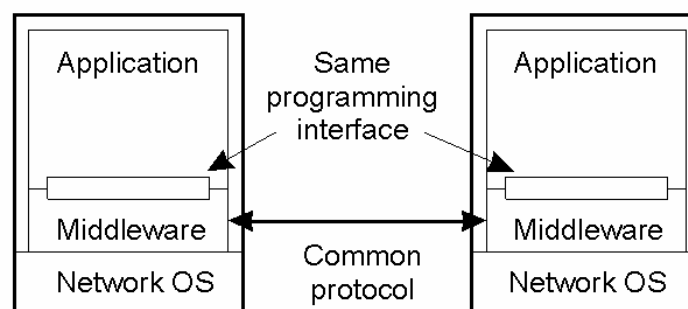


Struttura generale di un Sistema Distribuito come Middleware

## Esempi di Middleware Services

- Naming Distribuito
- File System Distribuito
- Inter-process communication (MPI)
- Remote Procedure Calls (RMI)
- Distributed Objects
- Documenti Distribuiti
- Single sign-on

## Middleware e Openness

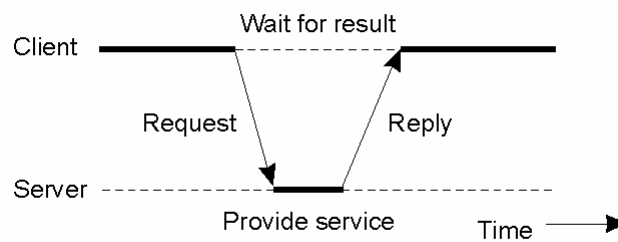


In un sistema distribuito aperto middleware-based i protocolli usati da ogni middleware devono essere uguali come le interfacce che offrono alle applicazioni.

## Confronto tra Sistemi DOS, NOS e Middleware

| Elemento                    | Distributed OS      |                     | Network OS | Middleware-based OS |
|-----------------------------|---------------------|---------------------|------------|---------------------|
|                             | Multiproc.          | Multicomp.          |            |                     |
| Grado di trasparenza        | Molto Elevato       | Elevato             | Basso      | Elevato             |
| Stesso S.O. su tutti i nodi | Si                  | Si                  | No         | No                  |
| Num. di copie del S.O.      | 1                   | N                   | N          | N                   |
| Base per la comunicazione   | Shared memory       | Messaggi            | File       | Model specific      |
| Resource management         | Global, centralizz. | Global, distribuito | Per nodo   | Per nodo            |
| Scalabilità                 | No                  | Moderately          | Si         | Variabile           |
| Openness                    | Chiuso              | Chiuso              | Aperto     | Aperto              |

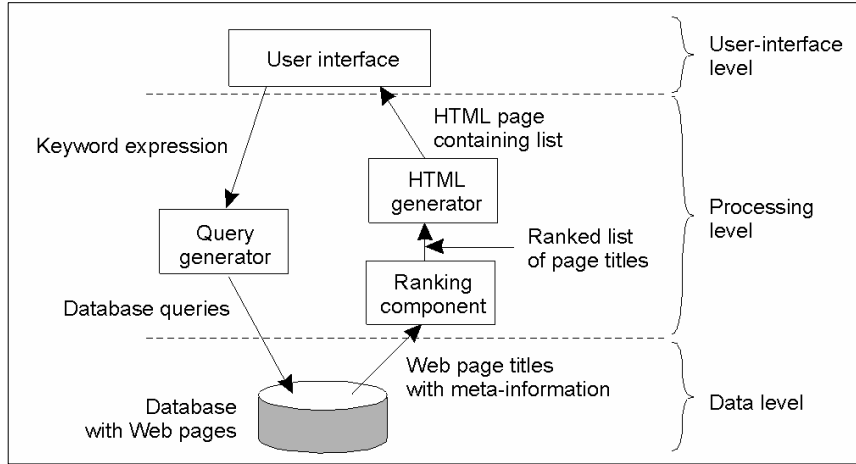
## Modello Client-Server



Interazione generale tra un cliente e un server

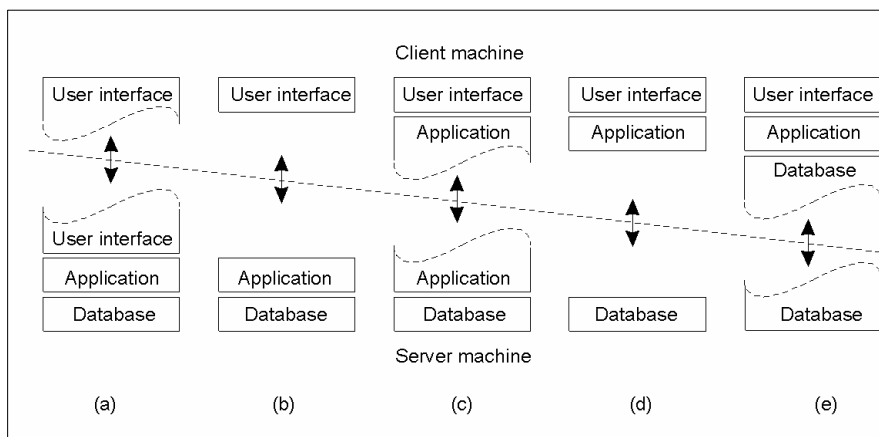


## Livelli di Elaborazione



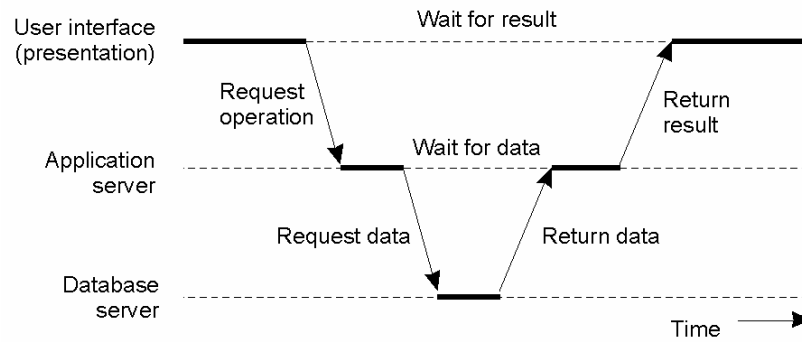
Organizzazione generale di una Internet search engine in tre livelli differenti

## Architetture Multi-tiered (1)



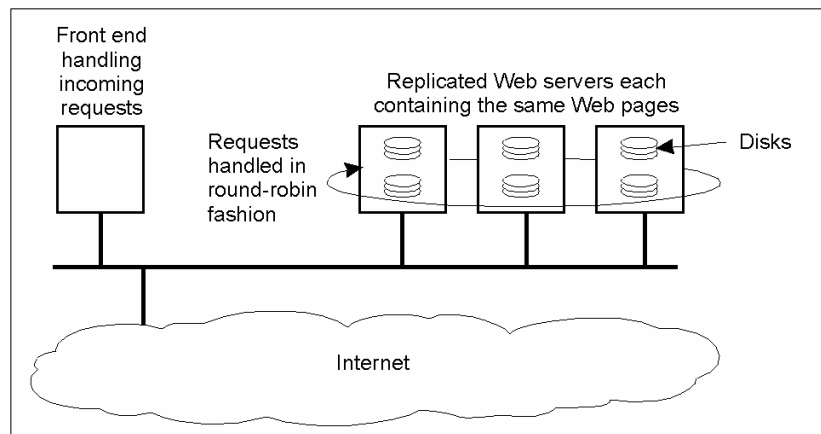
Organizzazioni client-server alternative (a) – (e).

## Architetture Multi-tiered (2)



Un esempio di un server che opera come un cliente.

## Architetture Moderne



Un esempio di distribuzione orizzontale di un servizio Web