“Capítol 17: Arquitectures del sistema de base de dades”

Fitxers i bases de dades

November 14, 2014

Capítol 17: Arquitectures del sistema de base de dades

- Sistemes centralitzats i client-servidor
- Arquitectures del sistema del servidor
- Sistemes paral·lels
- Sistemes distribuïts
- Tipus de xarxes

Sistemes centralitzats

- S'executen en un sol sistema informàtic i no interactuen amb altres sistemes informàtics.
- Propòsit general del sistema informàtic: d'un a uns pocs CPUs i un nombre de controladors de dispositius que es connecten a través d'un bus comú que proporciona accés a la memòria compartida.
- Sistema single-user (per exemple, ordinador o estació de treball): unitat de sobretaula, un sol usuari, en general només té un CPU i un o dos discos durs; el sistema operatiu pot suporten només un usuari.
- Sistema multi-user: més discos, més memòria, múltiples CPU i un sistema operatiu multiusuari. Serveix a un gran nombre d'usuaris que estan connectats al sistema de terminals que competeixen. Sovint anomenat sistemes de servidors.

Un sistema centralitzat per ordinador

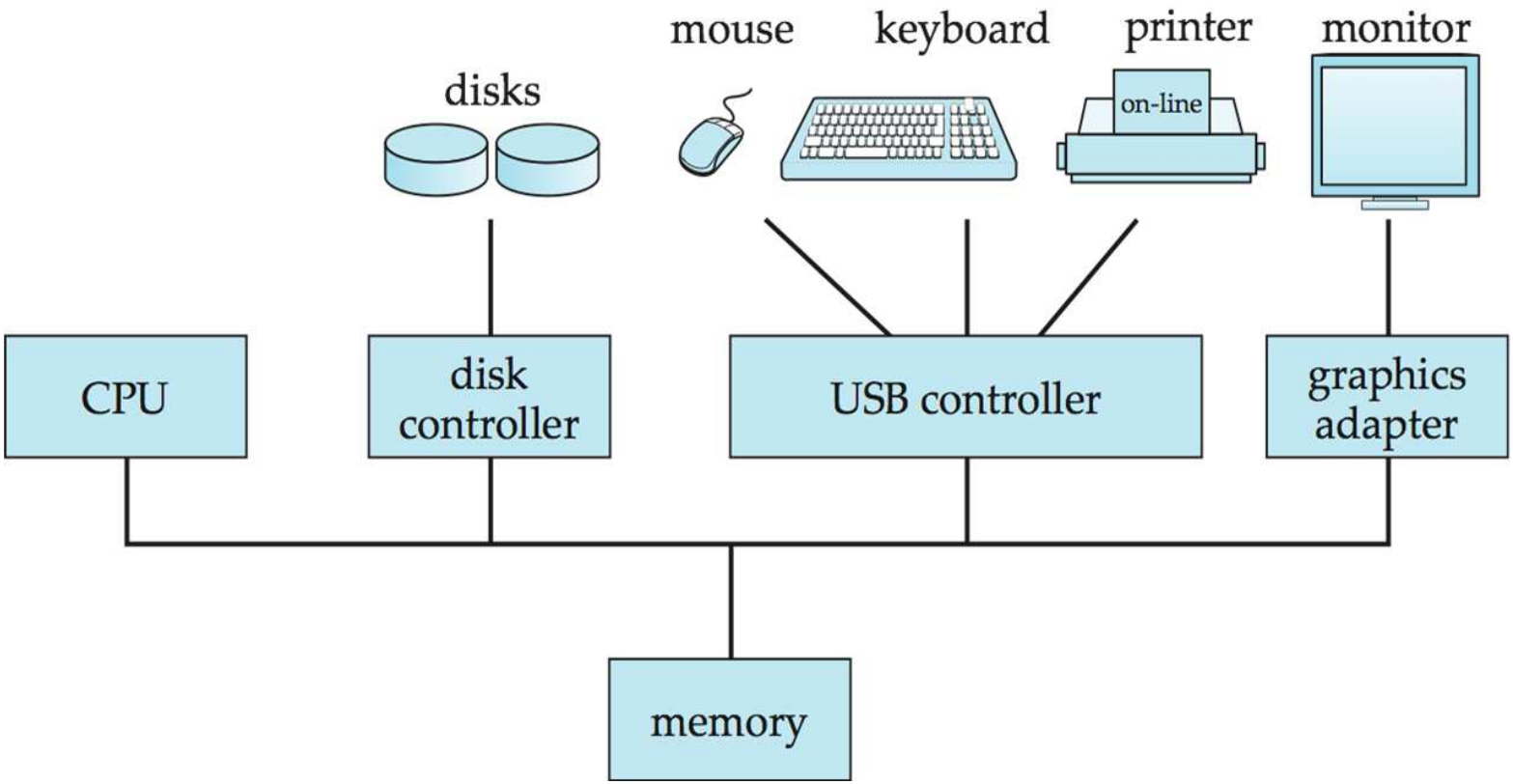

Figure: Exemple d'un sistema centralitzat per ordinador.

Sistemes Client-Servidor

- Els sistemes de servidor satisfan les sol · licituds generades en m sistemes client, l'estructura general es mostra a continuació:

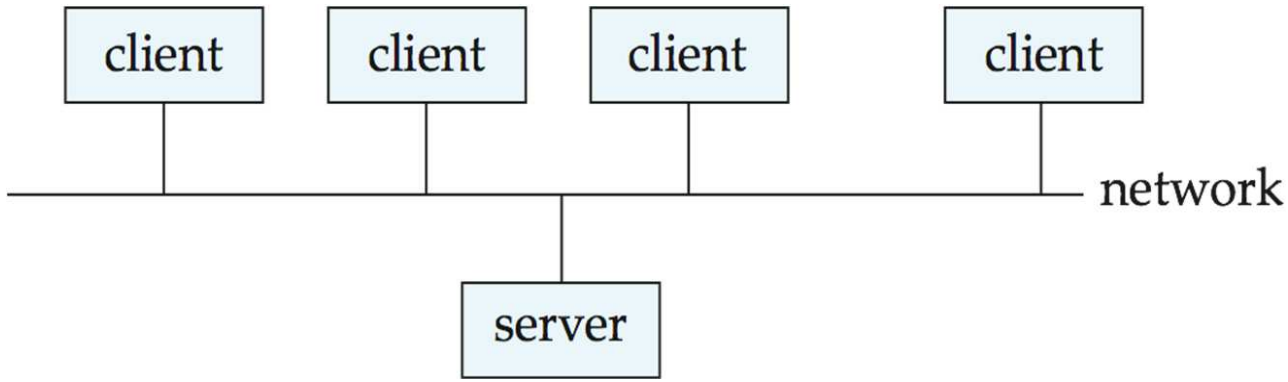

Figure: Exemple de sistemes client-servidor

Sistemes Client-Servidor (Cont.)

- La funcionalitat de la base de dades es pot dividir en:
  - Back-end:gestiona les estructures d'accés, avaluació i optimització de consultes, control de concurrència i recuperació.
  - Front-end: consisteix en eines com ara formularis, informes-escrits i les facilitats de la interfície gràfica d'usuari.
- La interfície entre el front-end i el back-end és a través de SQL o una interfície del programa de l'aplicació.

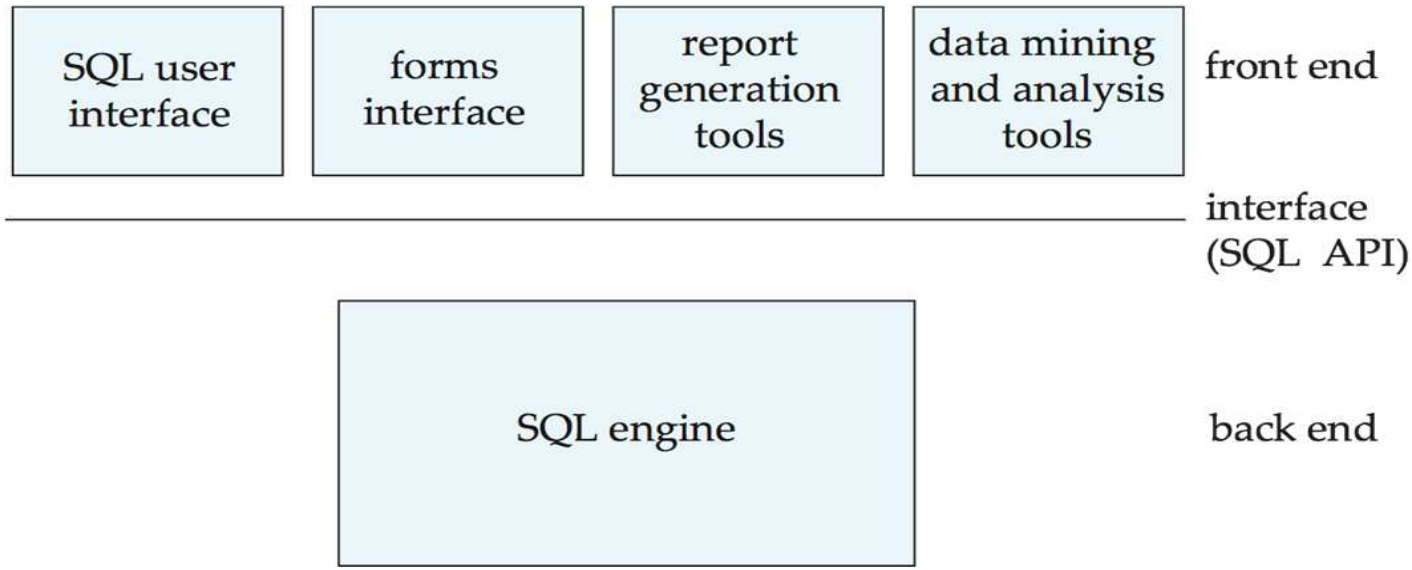

Figure: Exemple de la funcionalitat de la base de dades.

Sistemes Client-Servidor (Cont.)

- Els avantatges de la substitució dels mainframes amb xarxes d'estacions de treball o ordinadors personals connectats a màquines de servidor back-end són:
  - Millor funcionalitat per al cost
  - Flexibilitat en la localització de recursos i l'ampliació de les instal · lacions
  - Millors interfícies d'usuari
  - Més fàcil manteniment

Arquitectura del sistema del servidor

- Servidors de transaccions, que són àmpliament utilitzats en els sistemes de bases de dades relacionals.
- Servidors de dades, utilitzats en els sistemes de bases de dades orientades a objectes.

Servidors de transacció

- També anomenats sistemes de servidor de consulta o sistemes de servidor SQL.
  - Els clients envien peticions al servidor
  - Les transaccions s'executen en el servidor
  - Els resultats s'envien de tornada al client
- Les sol · licituds s'especifiquen en SQL, i es comuniquen amb el servidor a través d'un mecanisme de crida a procediment remot (RPC).
- Transaccional RPC permet que moltes trucades RPC per formar una transacció.
- Open Database Connectivity (ODBC) és un estàndard de la interfície del programa de l'aplicació en llenguatge C de Microsoft per connectar-se a un servidor, enviar automàticament peticions SQL, i rebre resultats.
- Estàndard JDBC és similar a ODBC, per a Java.

Estructura del procés del servidor de transacció

- Un servidor de transacció típica consisteix en múltiples processos que accedeixen a les dades en la memòria compartida.
- Els processos de servidor:
  - Aquests reben consultes d'usuaris (transaccions), els executen i retornen els resultats.
  - Els processos poden ser multithreaded, permetent un únic procés executar diverses consultes dels usuaris a la vegada.
  - Normalment múltiples processos de servidor multiprocessos.
- Procés de gestió de bloqueig
  - Més sobre això, després
- Procés d'escriptura de la base de dades
  - Sortida contínua de blocs de memòria modificats per a discos

Estructura del procés del servidor de transacció (Cont.)

- Procés log writer
  - Els processos de servidor simplement afegeixen log records per log record buffer.
  - Els processos log writer, produeixen log records per a un emmagatzematge estable.
- Processos de revisió (checkpoint)
  - Realitza revisions periòdiques.
- Processos de monitorització de processos
  - Monitoritza altres processos, i pren accions de recuperació si algun dels altres processos falla.

    Per exemple, avortar qualsevol transacció que estigui sent executada per un procés de servidor i reiniciar.

Procés del sistema de transacció

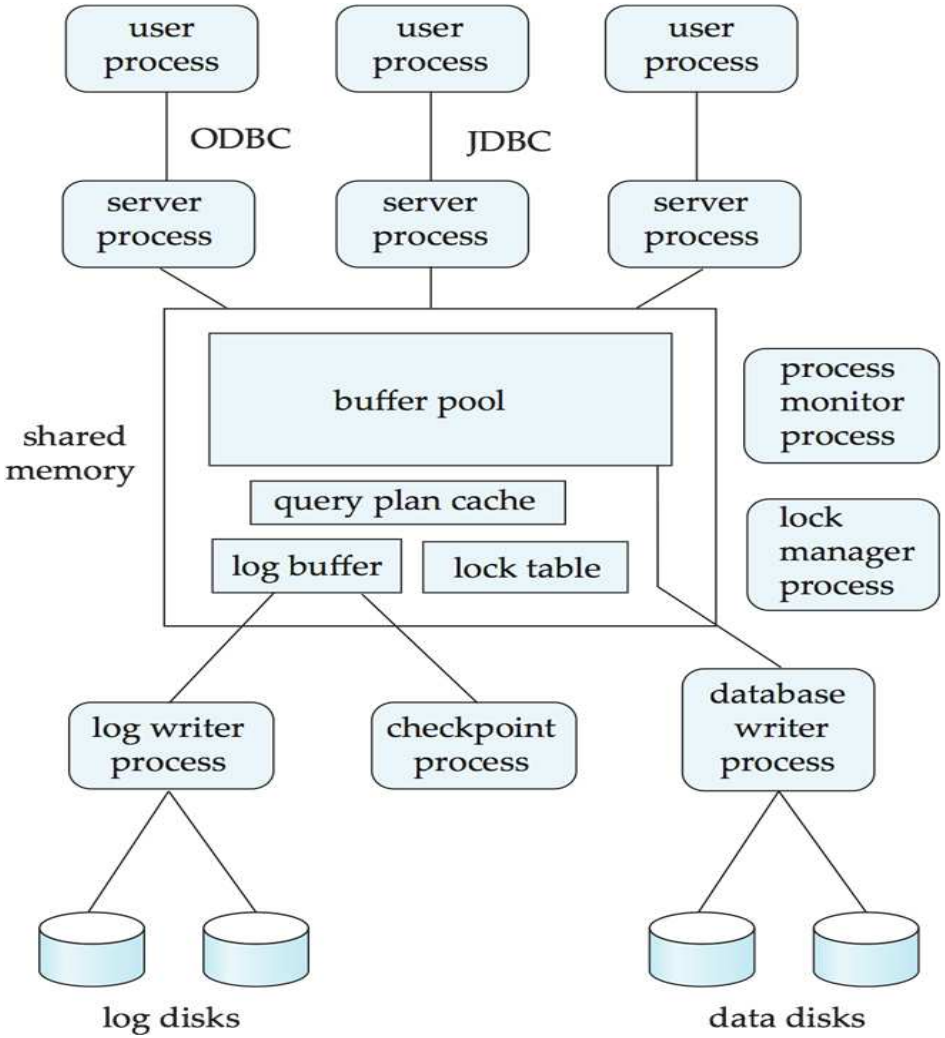

Figure: Processos del sistema de transacció.

Procés del sistema de transacció

- La memòria compartida conté dades compartides:
  - Buffer pool
  - Lock table
  - Log buffer
  - Cached query plans (reutilitzats si aquesta consulta és presentada novament)
- Tots els processos de base de dades poden accedir a la memòria compartida.
- Per assegurar que no hi ha dos processos que estan accedint a la mateixa estructura de dades, al mateix temps, els sistemes de bases de dades implementen exclusió mútua, utilitzant ja sigui.
  - Semàfors del sistema operatiu.
  - Instruccions atòmiques com a test-and-set.
- Per evitar la sobrecàrrega de comunicació entre processos per al bloqueig de sol · licitud / concessió, cada procés de base de dades opera directament en la lock table.
  - En lloc d'enviar les sol · licituds per lock manager process.
- Lock manager process encara s'utilitza per a la detecció d'estancament.

Servidors de dades

- S'utilitza en xarxes d'àrea local d'alta velocitat, en casos on:
  - Els clients són comparables pel que fa a la potència de processament al servidor.
  - Les tasques a executar són de còmput intensiu.
- Les dades s'envien als clients on es realitza el processament, i després s'envien els resultats de tornada al servidor.
- Aquesta arquitectura requereix la funcionalitat de back-end completa als clients.
- S'utilitza en molts sistemes de bases de dades orientades a objectes.
- Temes:
  - Page-Shipping versus Item-Shipping
  - Locking
  - Data Caching
  - Lock Caching

Servidors de dades (Cont.)

- Page-shipping versus item-shipping
  - Unitat d'enviament més petita $\Rightarrow$ més missatges.
  - Val la pena prefetching elements relacionats amb element sol · licitat.
  - Page shipping pot ser pensat com una forma de prefetching.
- Bloqueig
  - El sobrecàrrec de sol · licituds i l'obtenció de bloquejos del servidor és alta a causa dels retards de missatges.
  - Es poden concedir bloquejos dels elements sol · licitats i prefetched; amb page shipping, se li concedeix la transacció el bloqueig a tota la pàgina.
  - Bloquejos sobre un element prefetched poden ser Pcalled back pel servidor, i retornat per la transacció del client si l'element prefetched no ha estat utilitzat.
  - Bloquejos a la pàgina poden ser deescalated de bloquejos en els elements de la pàgina quan hi hagi conflictes de bloqueig. Bloquejos en els elements no utilitzats poden ser retornats al servidor.

Servidors de dades (Cont.)

- Data Caching
  - Les dades es poden emmagatzemar en la memòria cau del client, fins i tot enmig de les transaccions.
  - Però comprovar que les dades són de fins al dia abans del seu ús (coherència de caché).
  - El Check es pot fer quan es demana bloqueig sobre elements de dades.
- Lock Caching
  - Els bloquejos poden ser retinguts pel sistema client, fins i tot enmig de les transaccions.
  - Les transaccions poden adquirir bloquejos d'emmagatzematge en la memòria cau localment, sense contactar amb el servidor.
  - El servidor retorna les trucades (calls back) de bloquejos dels clients quan rep la sol · licitud de bloqueig en conflicte. Les devolucions del client es bloquegen una vegada que cap transacció local està utilitzant.
  - Similar a deescalation, però a través de transaccions.

Sistemes paral·lels

- Els sistemes de bases de dades paral · lels consisteixen en múltiples processadors i diversos discos connectats per una xarxa d'interconnexió ràpida.
- Una màquina coarse-grain parallel consisteix en un petit nombre de processadors potents.
- Massively parallel o fine grain parallel utilitza milers de processadors més petits.
- Dues mesures d'acompliment principals:
  - Rendiment el nombre de tasques que es poden completar en un interval de temps donat.
  - Temps de resposta la quantitat de temps que es necessita per completar una única tasca des del moment de la seva presentació.

Speed-Up i Scale-Up

- Speedup: un problema de grandària fixa executat en un sistema petit és donat a un sistema que és N vegades més gran.
  - Mesurat a través de:

    speedup$= \frac{\textit{temps transcorregut en el sistema petit}}{\textit{temps transcorregut en el sistema gran}}$
  - Speedup es **lineal** si la ecuación es igual a N.
- Scaleup: augmenta la mida de tots dos, el problema i el sistema.
  - Sistema N-vegades més gran utilitzat per realitzar treball N-vegades més gran.
  - Mesurat a través de:

    scaleup$= \frac{\textit{temps transcorregut problema petit sistema petit}}{\textit{temps transcorregut problema gran sistema gran}}$
  - Scaleup és **lineal** si l'equació és igual a 1.

Speed-Up

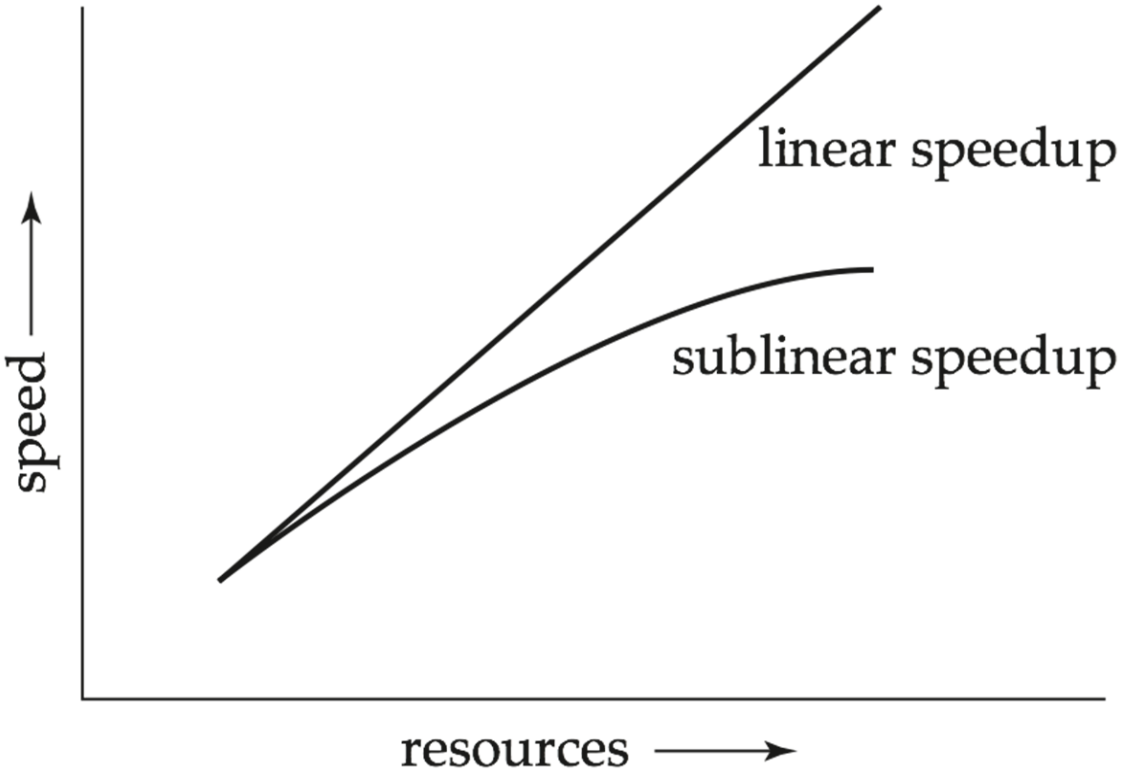

Figure: Speedup.

Scale-Up

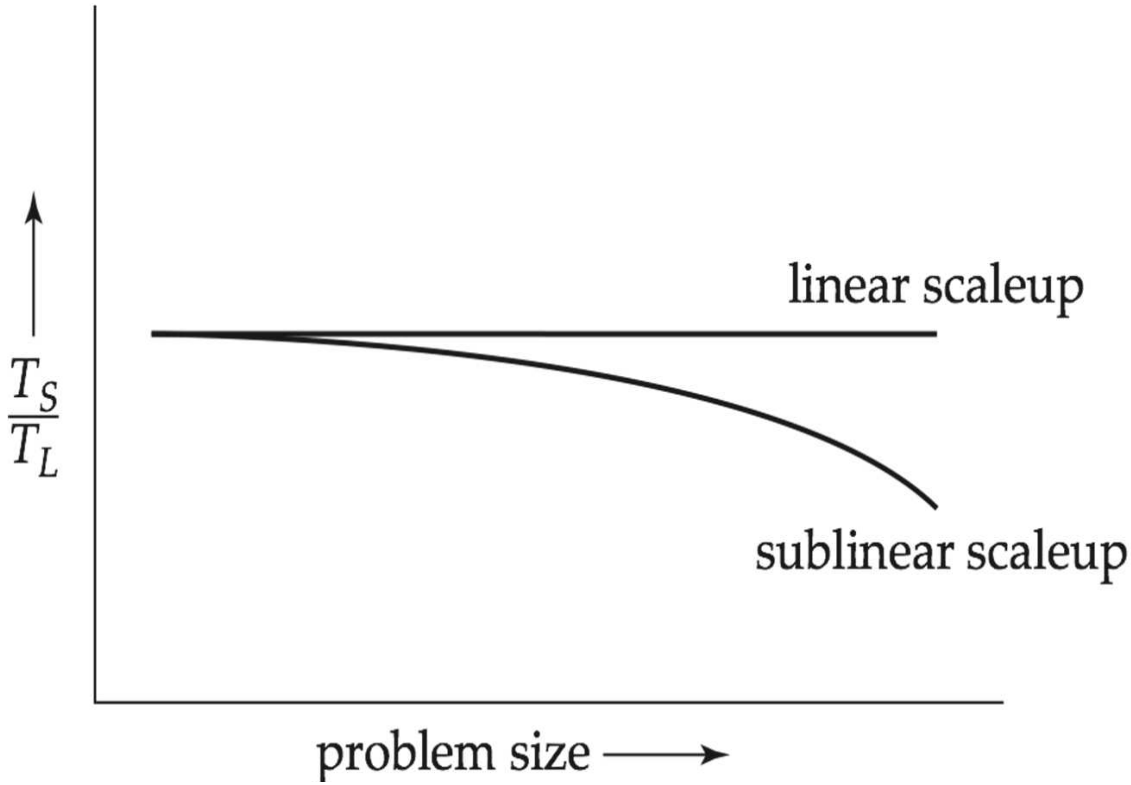

Figure: Scaleup.

Batch i Transaction Scaleup

- Batch scaleup:
  - Un sol treball gran; típic de la majoria de decisions sobre consultes de suport i simulació científica.
  - Utilitza un ordinador N-vegades més gran sobre un problema N-vegades més gran.
- Transaction scaleup:
  - Nombroses petites consultes presentades per usuaris independents a una base de dades compartida; sistemes típics de processament de transacció i temps compartit.
  - N-vegades més usuaris presenten sol · licituds (per tant, N-vegades més sol · licituds) a una base de dades N-vegades més gran, en un ordinador N-vegades més gran.
  - Molt adequat per a l'execució en paral · lel.

Factors limitants de speedup i scaleup

Speedup y scaleup són sovint sublineales (sublinear) a causa de:

- Costos inicials: El cost de posada en marxa de múltiples processos poden arribar a dominar el temps de càlcul, si el grau de paral · lelisme és alt.
- Interferències: els processos d'accés als recursos compartits (per exemple, bus de sistema, discs, o panys) competeixen entre si, gastant així temps esperant altres processos, en lloc de realitzar un treball útil.
- Biaix: ugmentar el grau de paral · lelisme augmenta la variància en els temps de servei de les tasques executades en paral · lel. Temps total d'execució determinat per **la més lenta** de les tasques executada en paral · lel.

Arquitectures d'interconnexió en xarxa

- Bus. Els components del sistema envien dades i reben dades d'un únic bus de comunicació:
  - No s'adapta bé en augmentar el paral · lelisme.
- Mesh. Els components estan disposats com a nodes en una xarxa, i cada component està connectat a tots els components adjacents.
  - Els enllaços de comunicació creixen amb el nombre creixent dels components, i així escala millor.
  - Però pot requerir $2\sqrt{n}$ salts per enviar el missatge a un node (o $\sqrt{n}$ amb connexions envoltants en la vora de la xarxa).
- Hypercube. Els components estan numerats en binari; els components estan connectats entre si, si les seves representacions binàries difereixen en exactament un bit.
  - n components estan connectats al $log(n)$ d'altres components i poden arribar ells a través de la majoria dels enllaços $log(n)$; redueix els retards de comunicació.

Arquitectures d'interconnexió

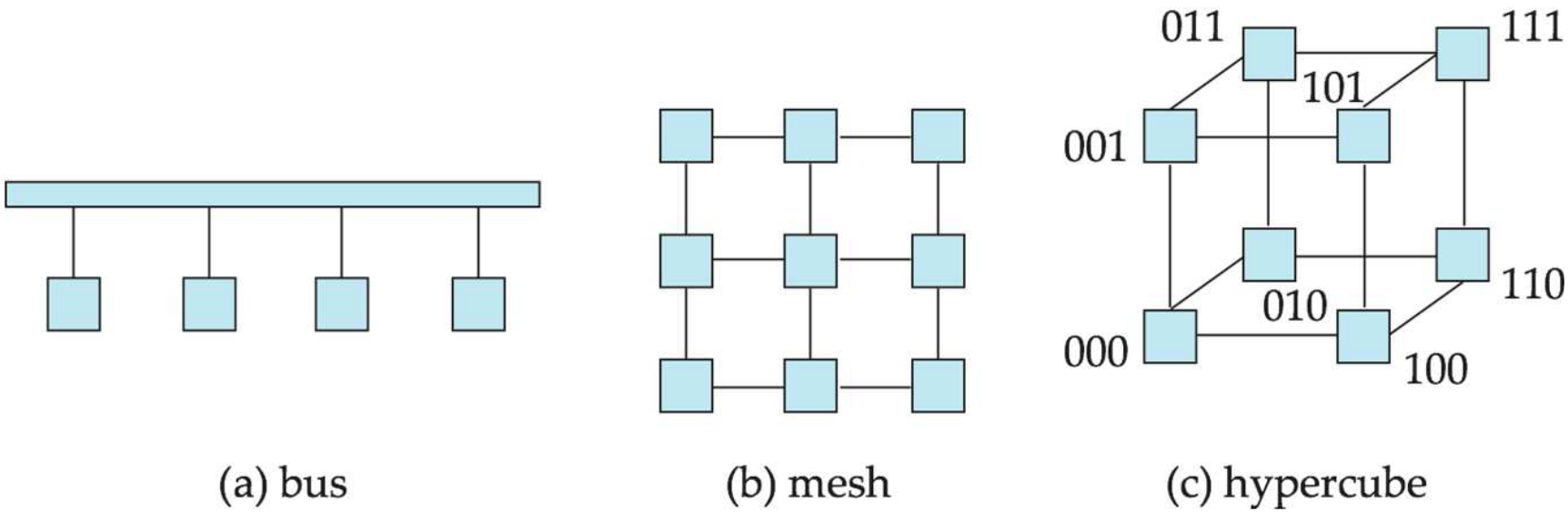

Figure: Exemples d'arquitectures d'interconnexió.

Arquitectures de bases de dades paral · leles

- Shared memory: Els processadors comparteixen una memòria comuna
- Disc compartit: Els processadors comparteixen un disc comú
- No-compartit: els processadors no comparteixen ni una memòria en comú ni disc en comú
- Jeràrquica: híbrid de les arquitectures anteriore

Arquitectures de bases de dades paral · leles (Cont.)

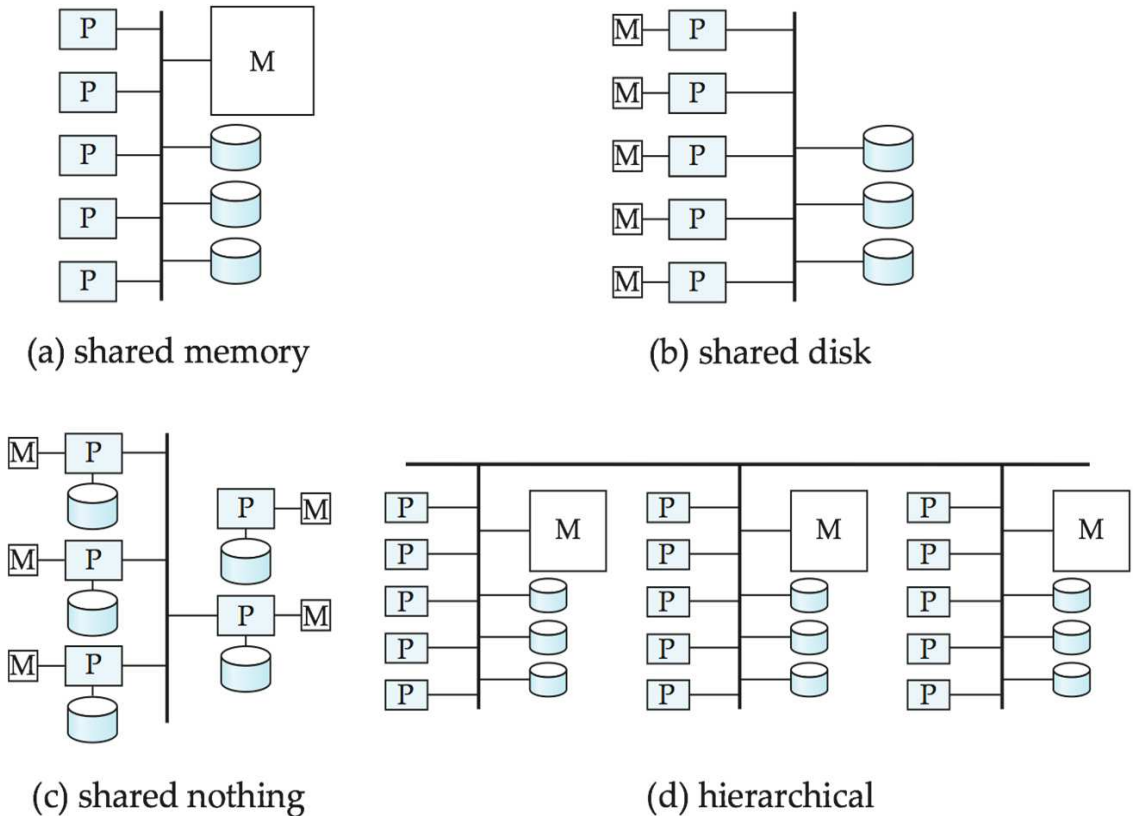

Figure: Exemples d'arquitectures de bases de dades paral · leles.

Memòria compartida

- Els processadors i discs tenen accés a una memòria comuna, normalment a través d'un bus o a través d'una xarxa d'interconnexió.
- Comunicació extremadament eficient entre els processadors - es pot accedir a les dades en la memòria compartida per qualsevol processador sense haver de moure utilitzant un programari.
- Desavantatge - l'arquitectura no és ampliable més enllà de 32 o 64 els processadors ja que el bus o la xarxa d'interconnexió es converteixen en un coll d'ampolla.
- Àmpliament utilitzat per a menors graus de paral · lelisme (4-8).

Disc compartit

- Tots els processadors poden accedir directament a tots els discos a través d'una xarxa d'interconnexió, però els els processadors tenen memòries privades.
  - El bus de memòria no és un coll d'ampolla.
  - L'Arquitectura ofereix un grau de tolerància a fallades - si un processador falla, els altres processadors poden fer-se càrrec de les seves tasques ja que la base de dades resideix en els discos que són accessibles des de tots els processadors
- **Exemples**: IBM Sysplex i DEC clústers (ara part de Compaq) eren els primers usuaris comercials corrent Rdb (ara Oracle Rdb).
- **Desavantatge**: el coll d'ampolla es produeix ara en la interconnexió amb el subsistema del disc.
- Sistemes de disc compartit pot escalar a un nombre una mica més gran de processadors, però la comunicació entre els processadors és més lenta.

No-compartit

- El node consta d'un processador, la memòria, i un o més discos. Els processadors en un node es comuniquen amb un altre processador en un altre node mitjançant una xarxa d'interconnexió. Un node posi funcions com el servidor per a les dades en el disc o discos del node.
- **Exemples**: Teradata, Tandem, Oracle-n CUBE.
- L'accés a les dades des de discos locals (i accés a la memòria local) no passen a través de la xarxa d'interconnexió, el que minimitza la interferència de la compartició de recursos.
- Multiprocessadors no-compartits poden ser ampliats fins a milers de processadors sense interferències.
- **Inconvenient principal**: el cost de comunicació i l'accés al disc no local; l'enviament de dades implica la interacció del programari en ambdós extrems.

Jeràrquic

- Combina característiques de memòria compartida, de disc compartit i arquitectures no-compartides.
- El nivell superior és una arquitectura no-compartida - nodes connectats per una xarxa d'interconnexió, i no comparteixen discos o memòria amb els altres.
- Cada node del sistema podria ser un sistema de memòria compartida amb alguns processadors.
- Alternativament, cada node podria ser un sistema de disc compartit, i cada un dels sistemes que comparteixen un conjunt de discos podria ser un sistema de memòria compartida.
- Redueix la complexitat de la programació d'aquests sistemes per les arquitectures de memòria virtual distribuïts.

  També anomenat arquitectura de memòria no uniforme (NUMA).

Sistemes distribuïts

- Les dades s'estenen sobre múltiples màquines (també conegudes com llocs o nodes).
- La xarxa interconnecta les màquines.
- Les dades compartits pels usuaris en múltiples màquines.

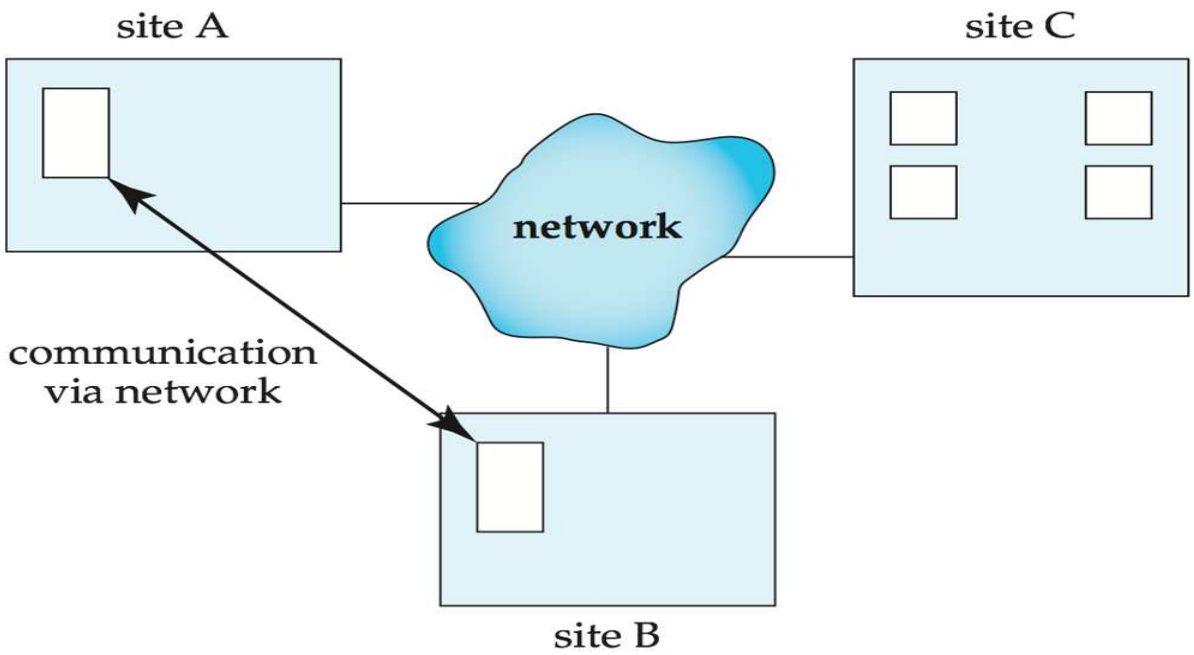

Figure: Exemples de sistemes distribuïts.

Bases de dades distribuïdes

- Bases de dades distribuïdes homogènies
  - El mateix programari / esquema en tots els llocs, les dades poden ser dividits entre els llocs.
  - **Objectiu**: proporcionar una vista d'una sola base de dades, ocultant els detalls de la distribució.
- Bases de dades distribuïdes heterogènia
  - Diferent programari / esquema en diferents llocs.
  - **Objectiu**: integrar les bases de dades existents per a proporcionar una funcionalitat útil.
- Diferenciar entre les operacions locals i globals
  - Una transacció local accedeix a les dades en l'únic lloc on es va iniciar la transacció.
  - Una transacció global o bé accedeix a les dades en un lloc diferent d'aquell en què es va iniciar la transacció o accedeix a les dades en diversos llocs diferents.

Compensacions (Trade-offs) en sistemes distribuïts

- Compartir dades - els usuaris són capaços d'accedir a les dades que resideixen en alguns altres llocs.
- Autonomia - cada lloc és capaç de retenir un grau de control sobre les dades emmagatzemades localment.
- Major disponibilitat del sistema des de redundància - les dades poden ser replicats en llocs remots, i el sistema pot funcionar fins i tot si un lloc falla.
- Desavantatge: complexitat addicional requerida per assegurar la coordinació adequada entre els llocs.
  - Costos de desenvolupament de programari.
  - Majors possibilitats de fallades.
  - Augment de la sobrecàrrega de processament.

Qüestions d'implementació per a bases de dades distribuïdes

- Atomicidad necessària fins i tot per a les transaccions que actualitzen dades en múltiples llocs.
- El protocol de confirmació en dues fases (2PC) s'utilitza per garantir la atomicitat.
  - Idea bàsica: cada lloc executa la transacció fins just abans de comprometre, i deixa la decisió final a un coordinador.
  - Cada lloc ha de seguir la decisió del coordinador, fins i tot si hi ha un error mentre s'espera per la decisió dels coordinadors.
- 2PC no sempre és apropiat: també s'utilitzen altres models de transacció basats en missatgeria persistent i fluxos de treball.
- Es requereix control de concurrència distribuït (i detecció de punt mort).
- Els elements de dades es poden replicar per millorar la disponibilitat de dades.
- Detalls de l'anterior en el capítol 22.

Tipus de xarxes

- **Xarxes d'àrea local (LAN)** - compost pels processadors que es distribueixen en petites àrees geogràfiques, com un sol edifici o uns quants edificis adjacents.
- **Xarxes d'àrea àmplia (WAN)** - compost per processadors distribuïts en una àmplia zona geogràfica.

Xarxa d'àrea local

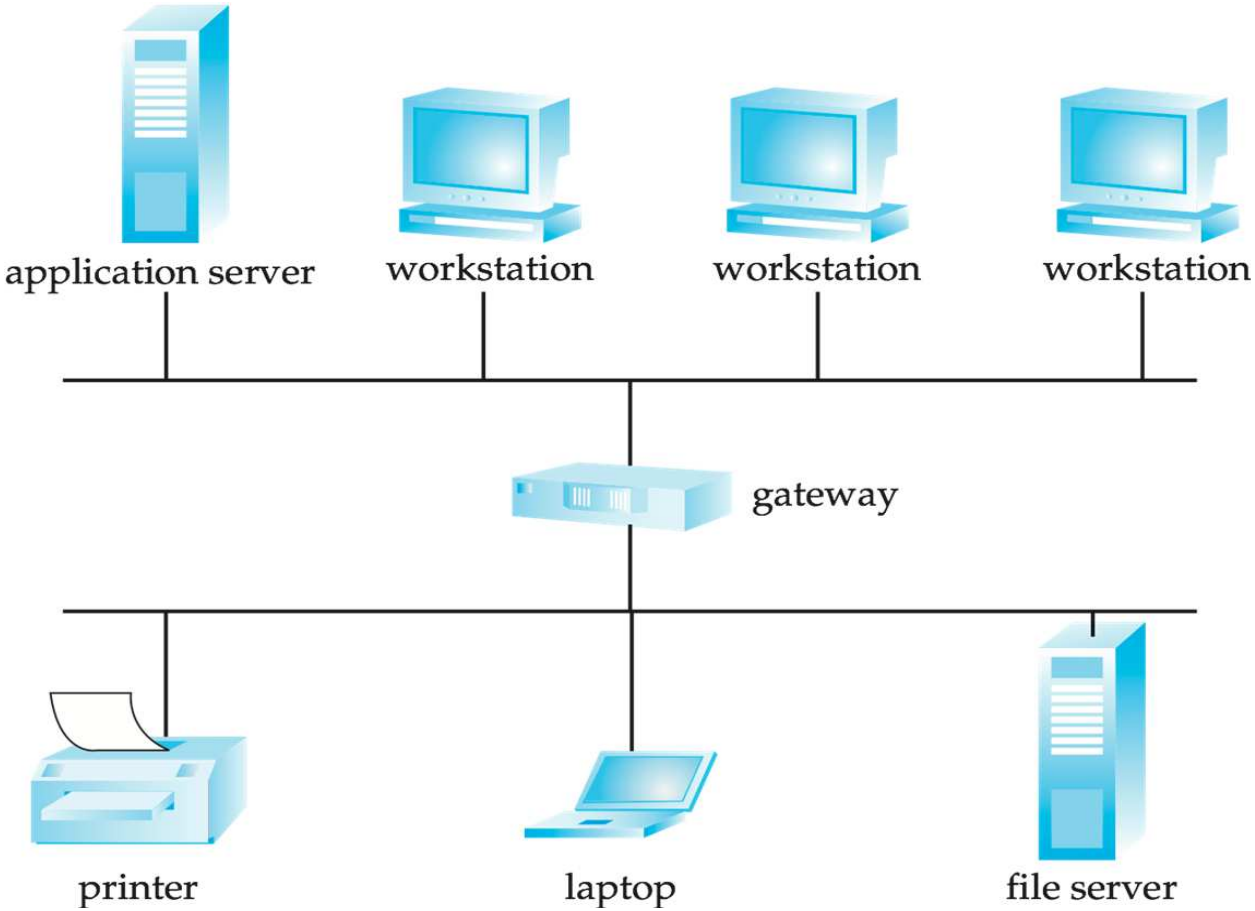

Figure: Exemples de xarxa d'àrea local.

Xarxa d'àrea local (Cont.)

- WAN amb connexió contínua (per exemple, Internet) són necessaris per a la implementació de sistemes de bases de dades distribuïdes.
- Aplicacions de groupware com Lotus Notes poden treballar en xarxes WAN amb connexió discontínua:
  - Les dades es repliquen.
  - Les actualitzacions es propaguen a les rèpliques periòdicament.
  - Les còpies de les dades es poden actualitzar de forma independent.
  - Execucions no serializables poden així ser. La resolució és depenent de l'aplicació.

Fina del Capítol 17

Figures

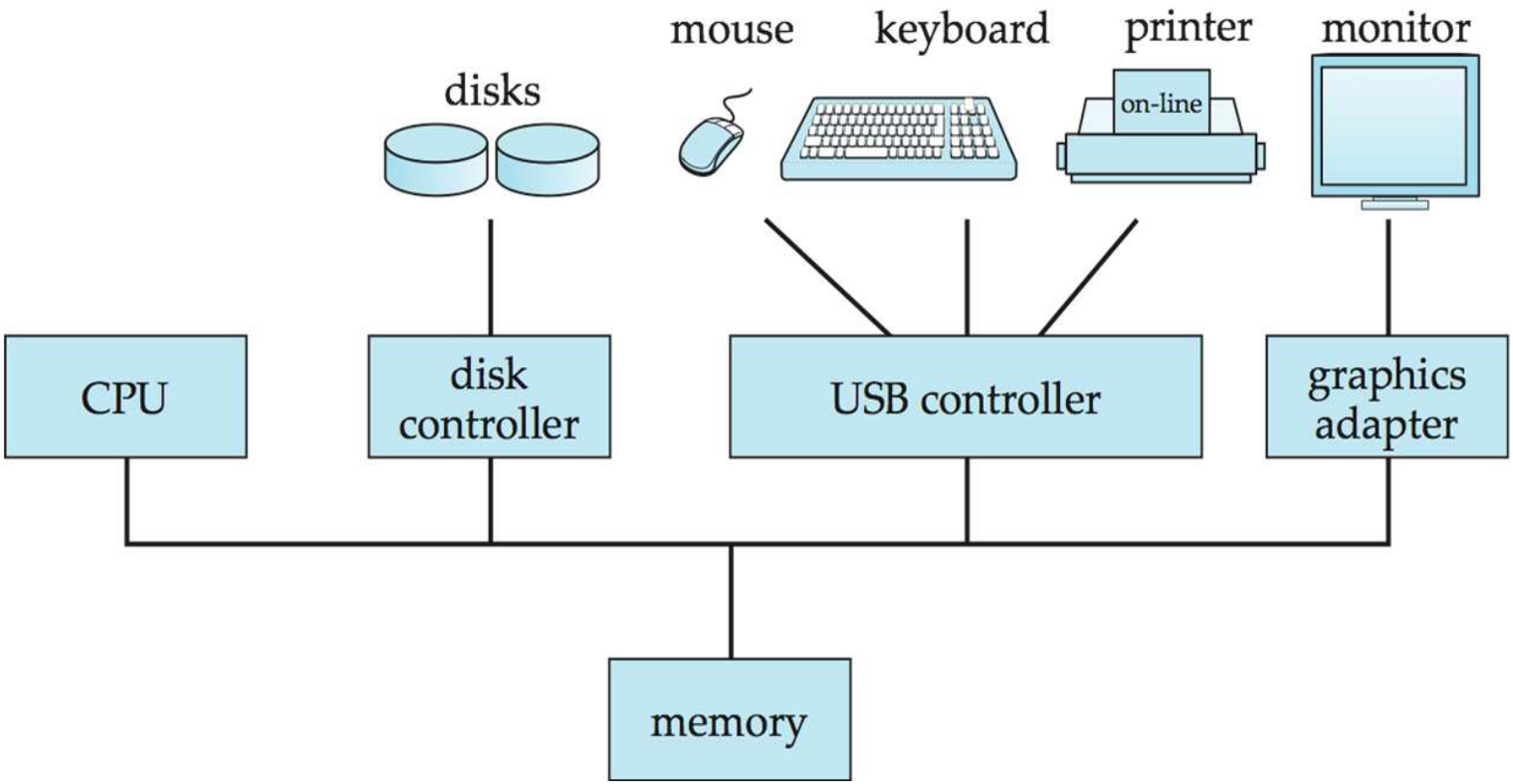

Figure: 17.01.

Figures

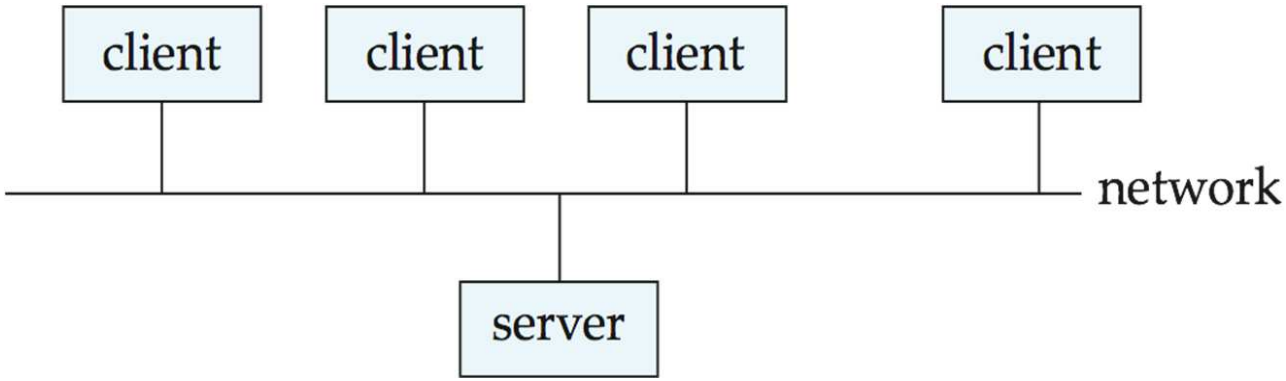

Figure: 17.02.

Figures

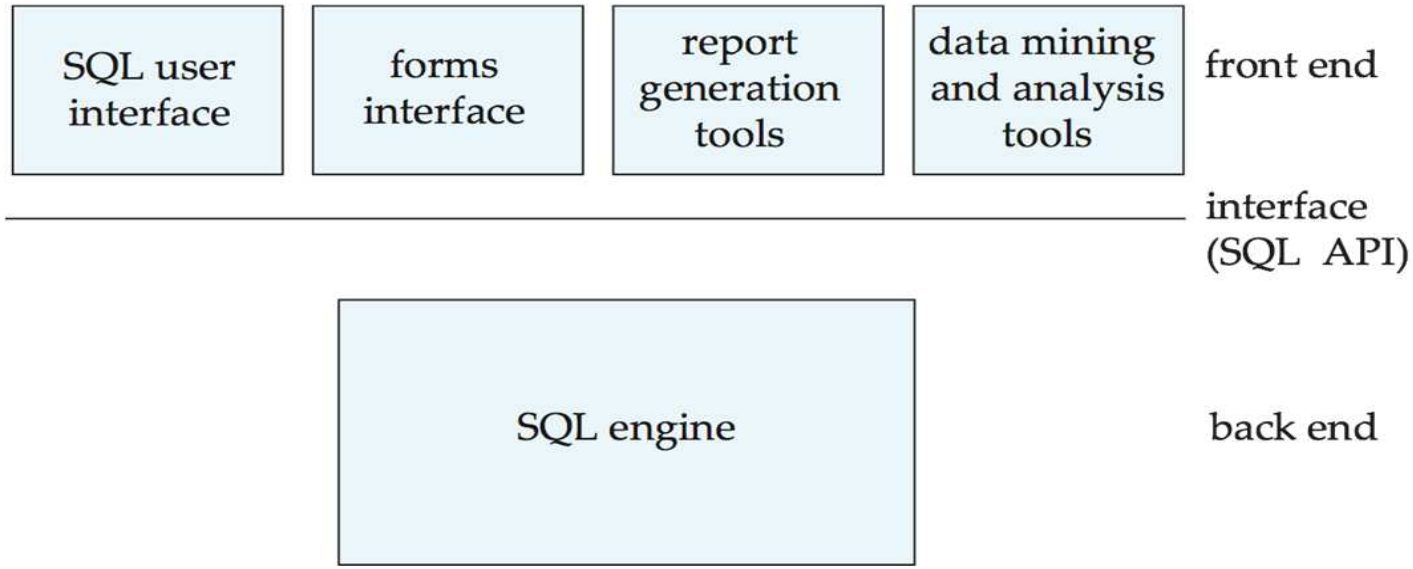

Figure: 17.03.

Figures

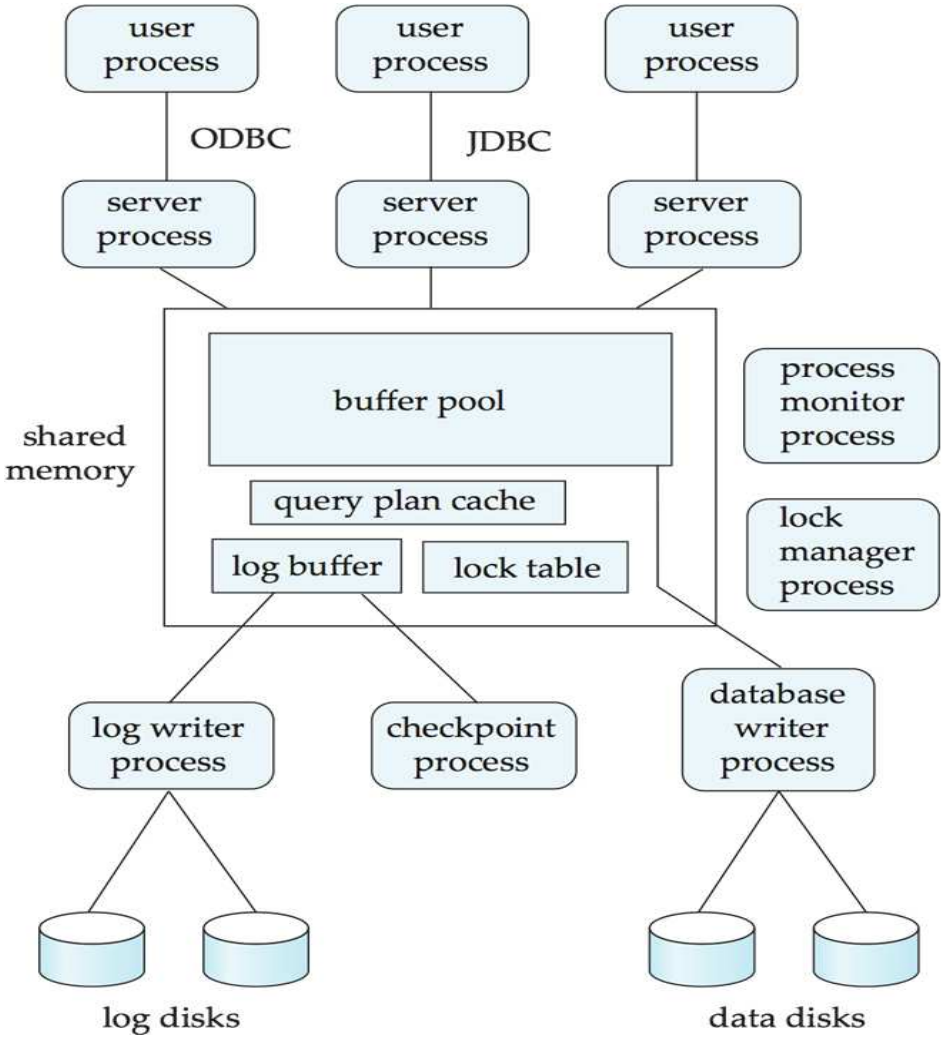

Figure: 17.04.

Figures

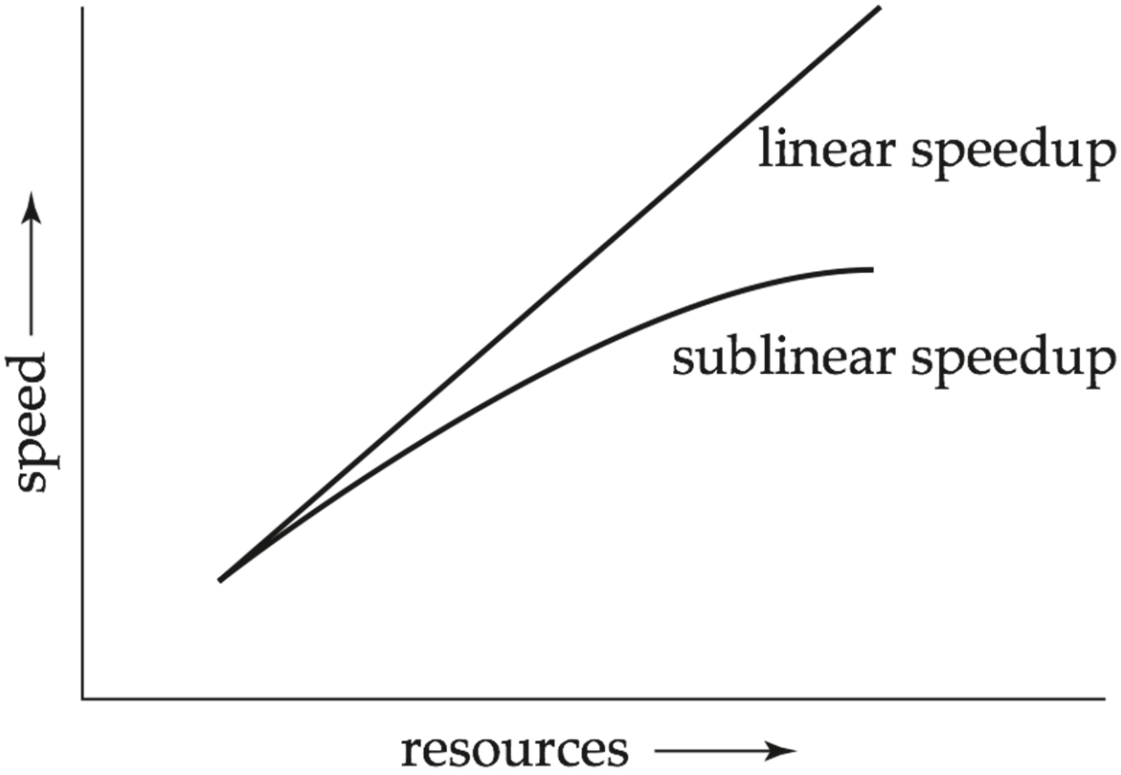

Figure: 17.05.

Figures

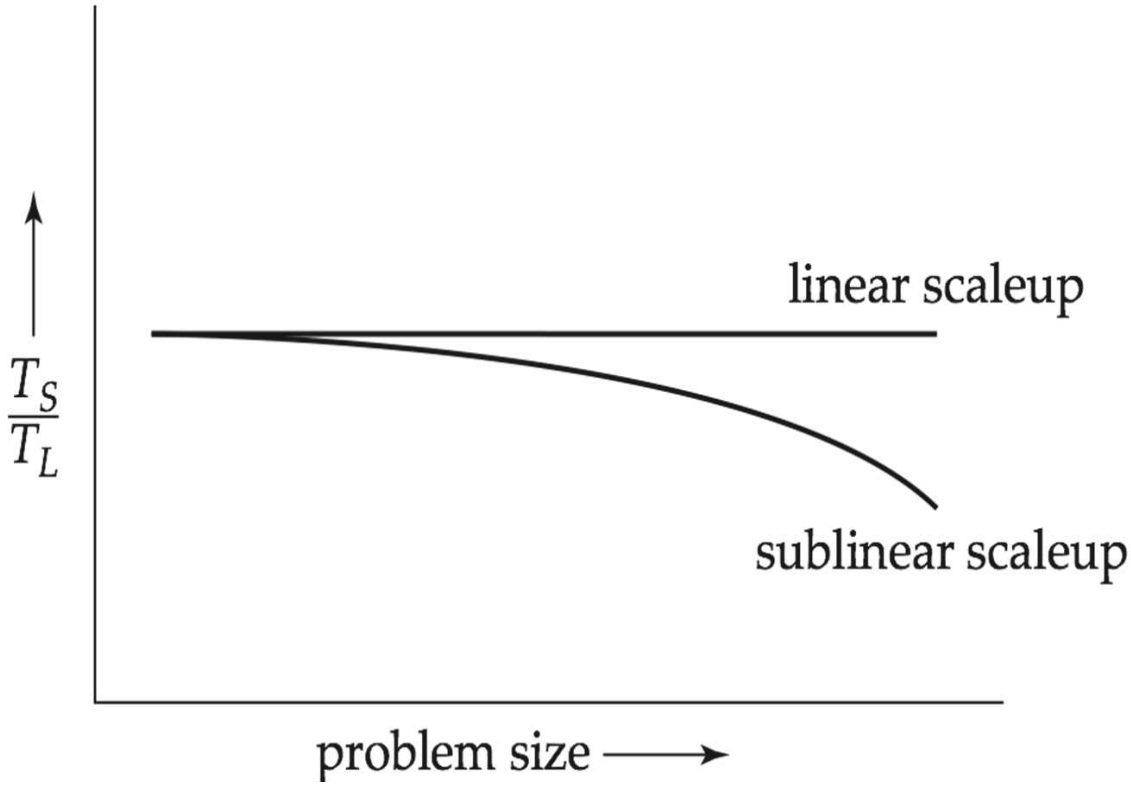

Figure: 17.06.

Figures

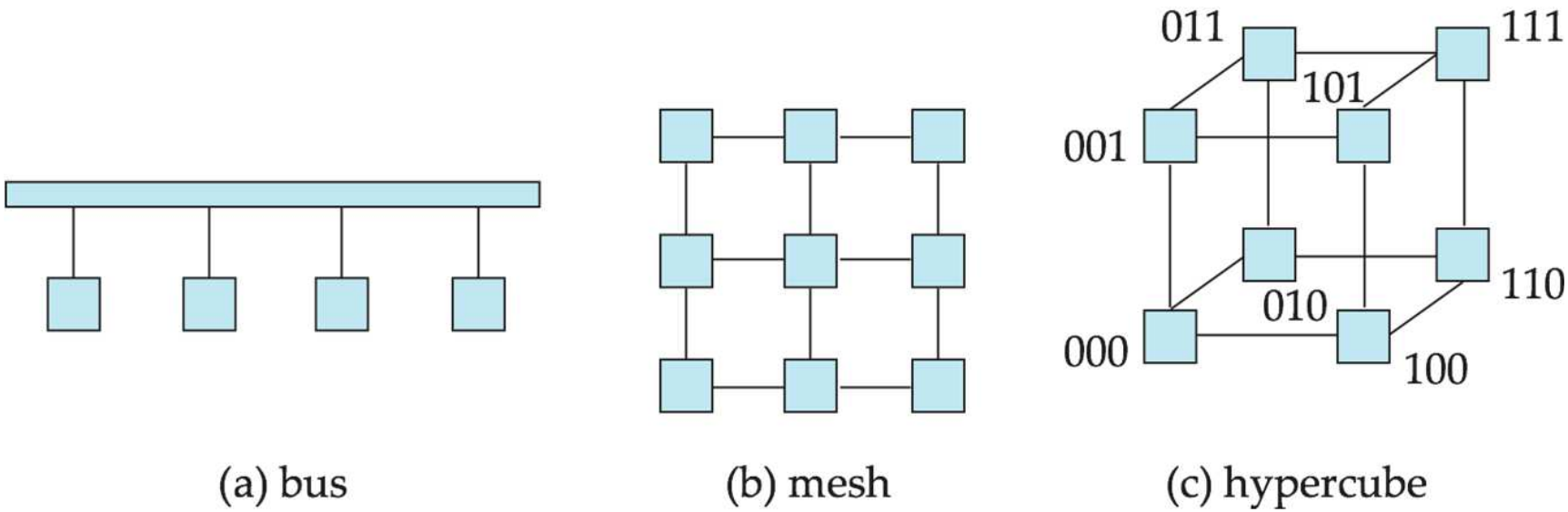

Figure: 17.07.

Figures

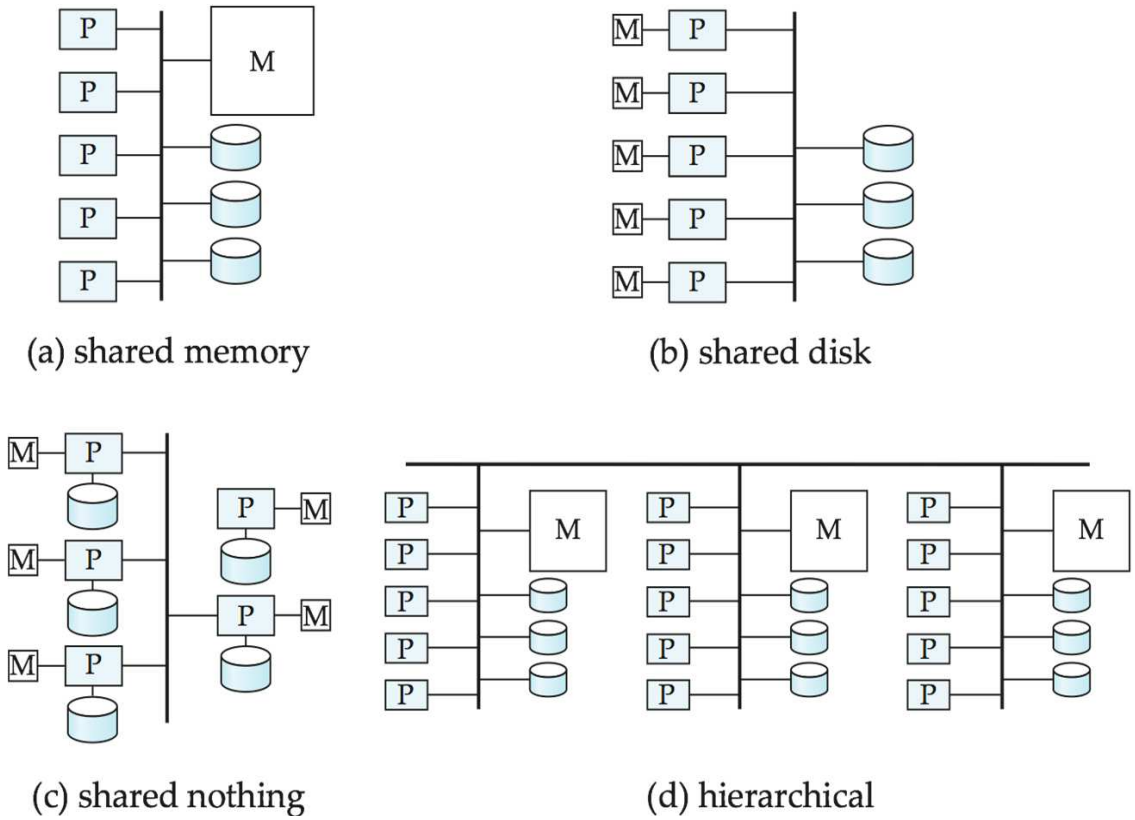

Figure: 17.08.

Figures

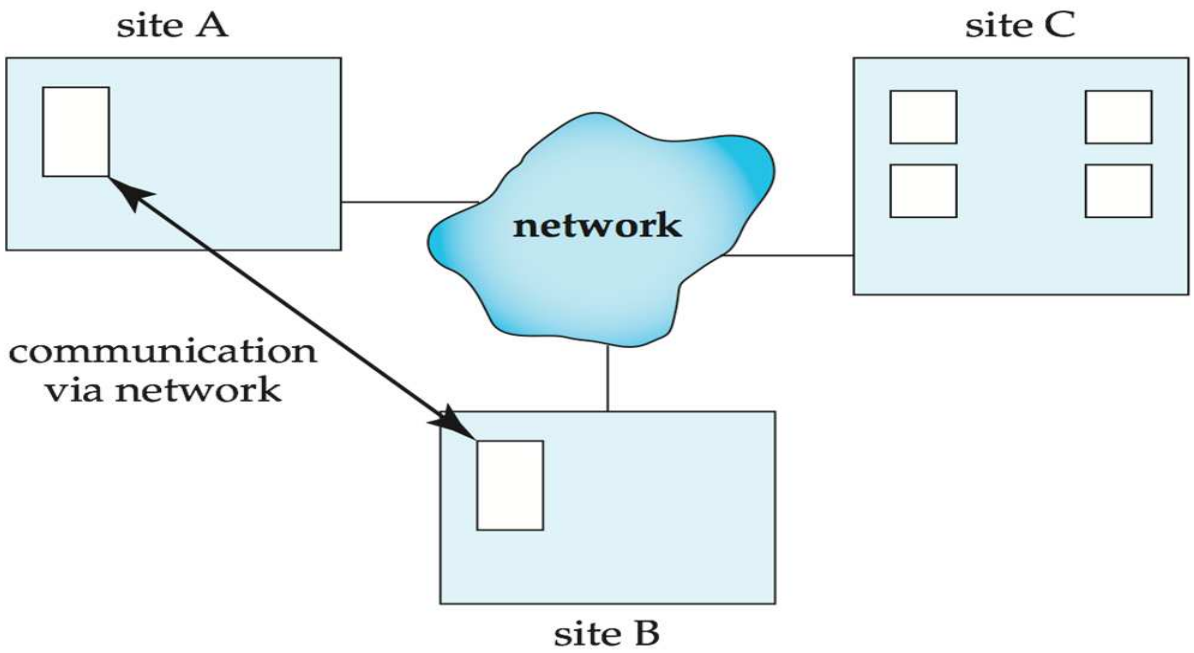

Figure: 17.09.

Figures

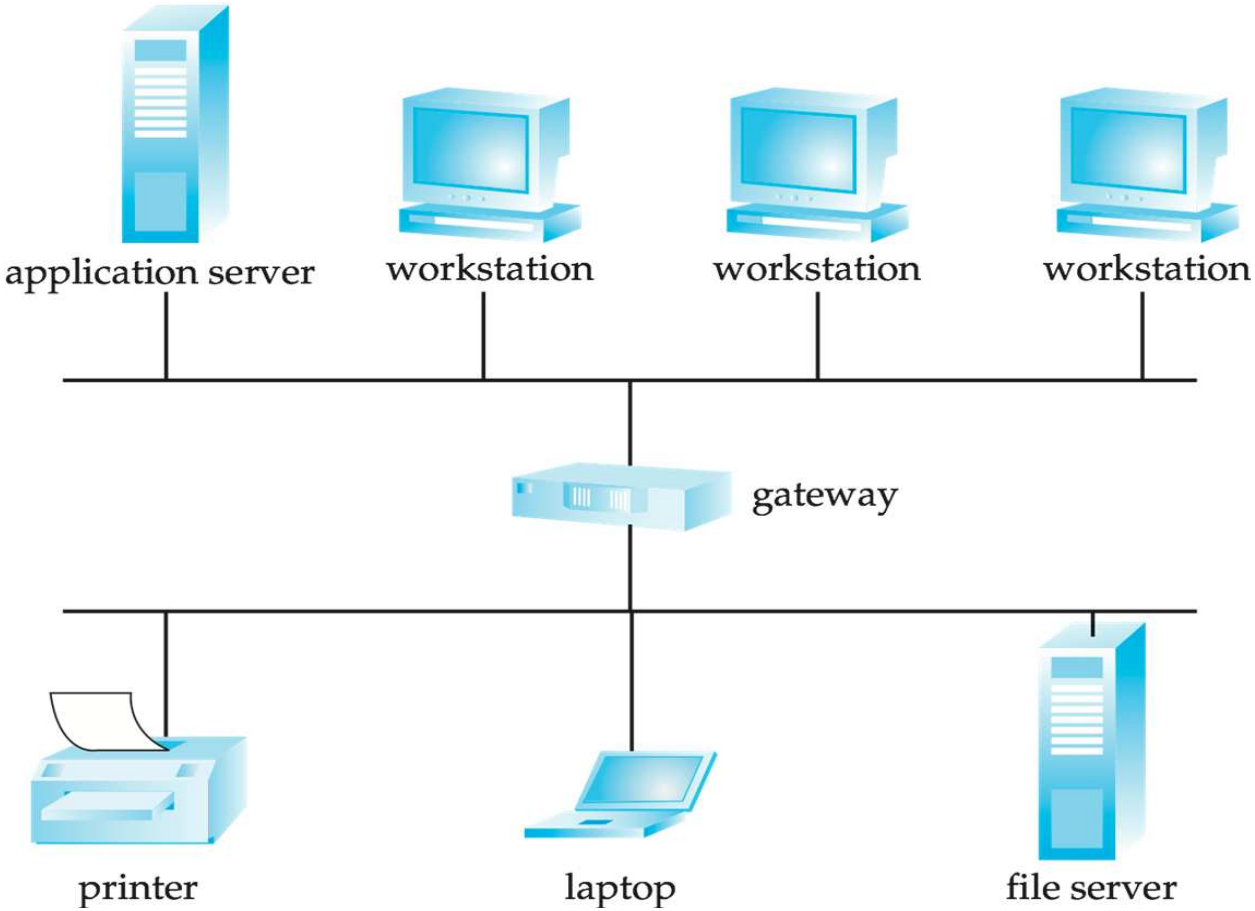

Figure: 17.10.

Figures

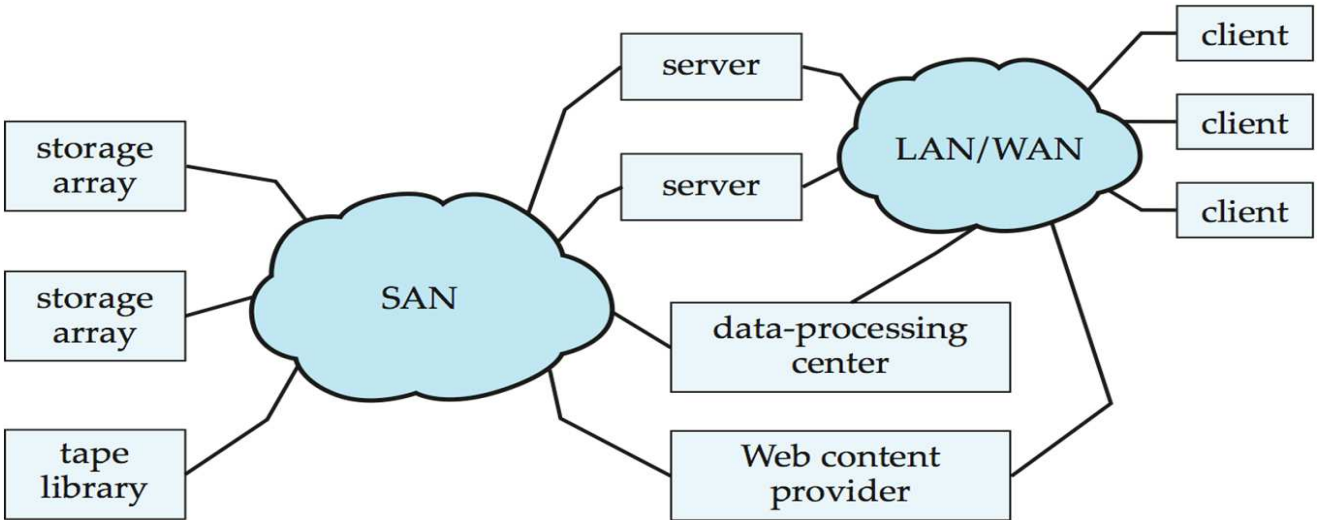

Figure: 17.11.