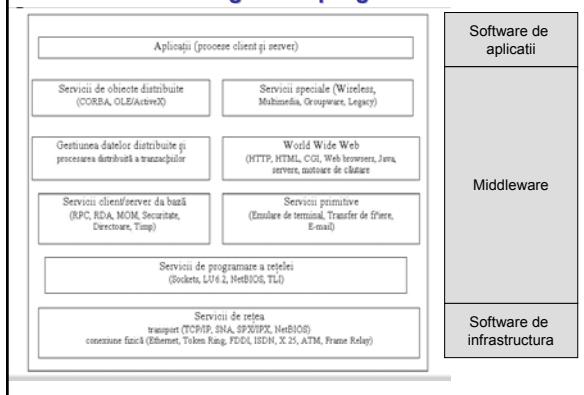


Sisteme de programe pentru Rețele de calculatoare

Categoriile de programe



Servicii middleware (1)

Servicii de programare a rețelei

Invoca serviciile rețelei prin APIs

Prima generație de middleware (un fel de limbaj de asamblare în ierarhia de limbaje).

Exemple: TCP/IP sockets, SNA Logical Unit 6.2, NetBIOS al IBM pentru rețele locale, TLI (Transport Layer Interface) de la AT&T.

Servicii primitive

emularea de terminale
transfer de fișiere
e-mail.

Servicii de bază client-server

Remote procedure call, RPC
Remote data access, RDA
Message oriented middleware, MOM
Alte facilități: servicii de securitate și de directoare.

Servicii middleware (2)

Servicii Web

Slujesc aplicațiile ce folosesc interfața grafică Web pentru acces la "resurse" Internet.

Gestiuinea datelor distribuite

Permite accesul "transparent" (localizare) la date distribuite.

Două niveluri de transparență:

Transparența la citirea din mai multe locuri (site-uri): utilizatorul poate citi și reuni date din mai multe site-uri fără să cunoască în ce locuri sunt păstrate datele

Transparența față de producător: utilizatorul poate citi și reuni (join) informații din baze de date de proveniență diferite (Informix, Oracle, . . .)

Servicii middleware (3)

Prelucrarea tranzacțiilor distribuite

(DTP – Distributed Transaction Processing)

- Tranzacție = acțiune atomică.
- DTP trebuie să furnizeze mai multe niveluri de transparență:
 - Transparență la actualizarea tuturor replicilor cu menținerea consistenței
 - Transparență la execuție distribuită
 - Transparență defectărilor
- Realizarea DTP folosește
 - algoritmi de sincronizare, detecția blocării, tolerarea defectărilor, etc.

Servicii middleware (4)

Obiecte distribuite

- Permit ca obiectele client să facă apel la serviciile (metodele) unor obiecte server aflate la distanță.
- Se bazează pe:
 - Definirea serviciilor folosind IDL
 - Folosirea unui suport pentru transmiterea invocarilor la distanță; ex. Object Request Broker (ORB)
 - Folosirea unor depozite pentru stocarea și evidența obiectelor.
- Exemple:
 - Java RMI
 - CORBA (Common Object Request Broker Architecture) definit de OMG (Object Management Group)
 - DCOM, definit de Microsoft

Servicii middleware (5)

Servicii middleware speciale

- **Groupware** (ex Lotus Notes) permite activitatea unor grupuri prin intermediul rețelei
- **Mobile computing** suportă utilizatori mobili (wireless)
- **Multimedia** middleware permite dezvoltarea aplicațiilor distribuite multimedia
- **Legacy-access/integration** permite integrarea unor aplicații tradiționale cu aplicații noi.

Internet = cel mai mare sistem distribuit

Definiție

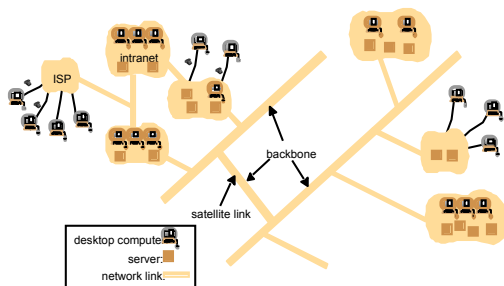
Un **sistem distribuit** este unul în care componente hardware și software localizate în calculatoare conectate în rețea comunică și își coordonează acțiunile doar prin transfer de mesaje.

[Coulouris 2001]

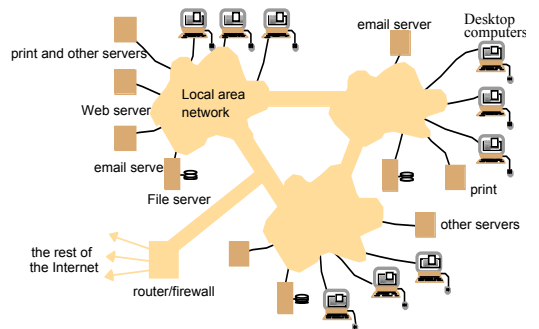
Exemple

- Internet
- Intranet
- Rețele mobile și ubiquie
- Web

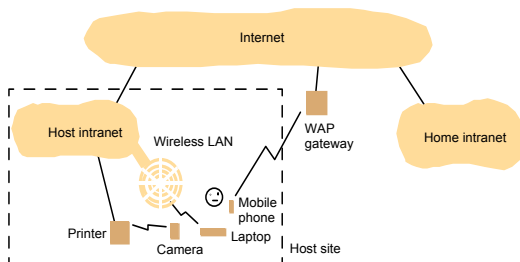
O porțiune tipică de Internet



Un Intranet tipic



Rețea cu echipamente mobile



Un alt punct de vedere

- Un **sistem distribuit** este o colecție de calculatoare independente care apar utilizatorilor ca un singur sistem coerent.

[Tanenbaum 2002]

- Un sistem distribuit este organizat în jurul unei paradigme (de ex. "orice este un fișier")

Exemple

- sisteme de fișiere distribuite (NFS)
- sisteme bazate pe obiecte distribuite (CORBA, DCOM)
- sisteme bazate pe documente distribuite (Web)
- sisteme bazate pe coordonare distribuită (Jini)

Probleme de proiectare

Heterogeneitatea (Heterogeneity) – diversitatea componentelor Se aplica la:

rețele – tratata prin utilizarea protocoalelor Internet pentru comunicare
hardware – reprezentarea diferita a datelor (de ex. big endian, little endian)

sisteme de operare – implementari diferite ale protocoalelor Internet; apelurile ptr schimb de mesaje in Unix sunt diferite de Windows NT
limbaje de programare – folosesc reprezentari diferite pentru caractere si structuri de date (tablouri, inregistrari)

implementari de la dezvoltatori diferiti – nefolosirea sau nerespectarea standardelor

Solutii pentru mascarea heterogeneitatii
middleware
virtual machine (pentru cod mobil)

• **Deschiderea (Openess)** – determina daca sistemul poate fi extins si re-implementat in moduri diferite.

Solutii:

- interfețele principale sunt publice (ex. disponibile in IDL)
- este asigurata conformitatea componentelor cu standardele publice

– Notiune legata de **interoperabilitate** si **portabilitate**

• **Securitatea** – se refera la:

- confidentialitate
- integritate
- autentificare
- non-repudiere

Solutii

- algoritmi, mecasime, protocoale si servicii criptografice

• **Scalabilitatea (Scalability)** – sistemul ramane performant cu cresterea semnificativa a:
– numarului de utilizatori (dimensiunea)
– distantei maxime intre noduri (scalabilitatea geografica)
– numarului de domenii administrative (scalab admin.)

• **Solutii**

- comunicatia asincrona
 - evita blocajele lungi din comunicatia sincrona
- distributia task-urilor
 - ex. DNS
- replicarea componentelor (in particular caching)
 - ridica probleme de consistenta

• **Tratarea defectelor (Failure handling)** – detectia, mascarea, prevenirea, sau recuperarea din defectari.

– defectariile sunt pariale

Solutii

- detectarea (ex. sume de control)
- mascarea (ex. repetarea operatiei)
- toleranta la defectari
 - proiectare pentru recuperare din eroare
 - redundanta

• **Concurenta (Concurrency)** – executia simultana a task-urilor

– probleme la accesul resurselor comune

Solutii

– sincronizarea

• **Transparenta (Transparency)** – sistemul se prezinta unitar fata de utilizatori si aplicatii.

- nivel inalt - **ascunderea distributiei fata de utilizatori**
- nivel scazut – **ascunderea distributiei fata de programe.**

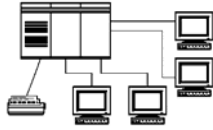
• Opt **forme de transparenta:**

- **acces** (resursele locale si la distanta sunt accesate cu aceleasi operatii)
- **locatie** (nu se stie unde este localizata resursa)
- **concurenta** (o resursa poate fi partajata de mai multe procese)
- **replicare** (pot fi folosite mai multe instante de resurse fara a se sti ce replica este folosita)
- **defectare** (task-urile pot fi duse la capat chiar daca se defecteaza unele componente)
- **migrare** (resursele si procesele pot fi mutate fara a afecta utilizatorii si programele)
- **relocare** (resursele pot fi mutate in alte locatii in timp ce sunt utilizate)

Modele de servicii

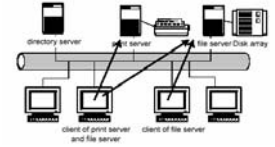
• Centralizat

- Aplicatiile sunt gazduite de un singur sistem de calcul accesibil prin terminale simple (dumb).
- Sistemul face toate prelucrarile si transmite rezultatele la terminale.
- Terminalele fac doar operatii de intrare/iesire.



Modelul Client - Server

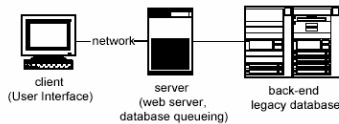
- Prelucrarea este distribuita intre masini client si server dedicate



- **Server** = proces care implementeaza un serviciu specific (ex. server de fisiere, de nume, de tiparire)
- **Client** = proces care cere un serviciu de la un server trimittand o cerere si asteptand raspunsul.
- Tipuri de clienti
 - subtire (thin) – implementeaza interfața cu utilizatorul
 - gras (fat) – implementeaza si o parte a aplicatiei
- **Arhitectura** cu doua niveluri (two-tiered)

Modelul multi-tiered

- **Aplicatie Web** – o aplicatie client / server in care clientul este un browser
- **Arhitectura multi-tiered** – un server poate fi client pentru un alt server; fiecare server-client adauga un nou tier;
- Ex. 3-tier
 - client
 - server de aplicatie
 - server de baze de date



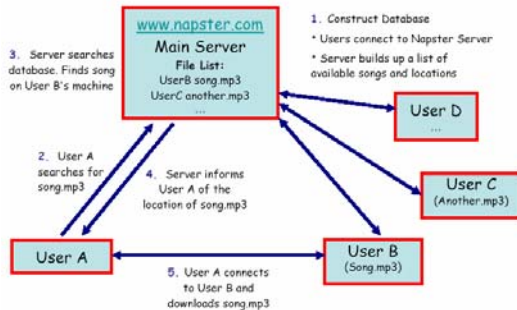
Modelul peer-to-peer

- Bazat pe accesul direct intre calculatoare "egale" (peer computers) si nu printr-un server
- **peer** -> ofera si consuma (in contrast cu client - server)

- Exemple:
 - Gnutella, Napster – partajare/stocare fisiere
 - SETI@HOME – partajare procesoare
 - ICQ – Instant messaging
 - netmeeting – conferinte

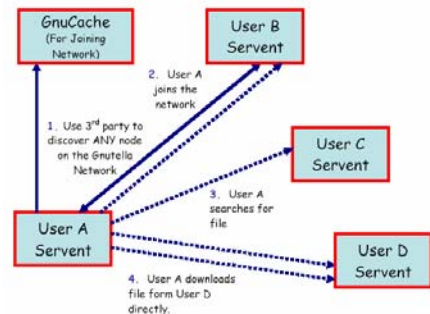
- Se clasifica dupa gradul de distributie

Napster – brokered p2p



[Distributed Systems Course – Omer Rana and Ian Taylor]

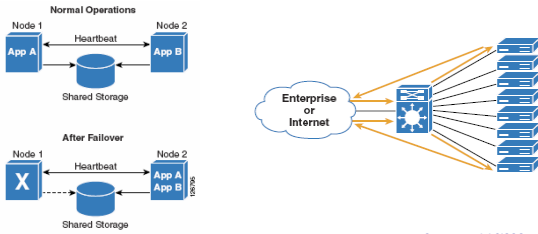
Gnutella – p2p descentralizat



[Distributed Systems Course – Omer Rana and Ian Taylor]

Clustere

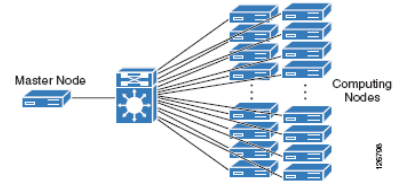
- ansamblu de statii integrate prin retea si software pentru a se comporta ca un singur calculator
- folosite traditional pentru calcul de inalta performanta
- noi utilizari: cresterea disponibilitatii si echilibrarea incarcarii serverelor



Sursa: tutorial CISCO

Tipuri de noduri in cluster

- master
 - ruleaza sistemul de fisiere si middleware-ul de cluster
 - aloca nodurile pentru aplicatii
 - interfața cu utilizatorul
 - pastreaza coada de joburi
- calcul
 - procesare si memorare date



Grid

- grupeaza resurse din diferite organizatii pentru a sustine dezvoltarea colaborative a unor proiecte
- suporta organizatii virtuale
- arhitectura orientata pe servicii

