SDDX - SISTEMES OPERATIUS DISTRIBUITS EN XARXA

TEMA 1- CONCEPTS OF DISTRIBUTED SYSTEMS

- Definition of distributed systems
- Challenges of distributed systems
- Architectures ofor distributed systems

DEFINITION OF A DISTRIBUTED SYSTEM

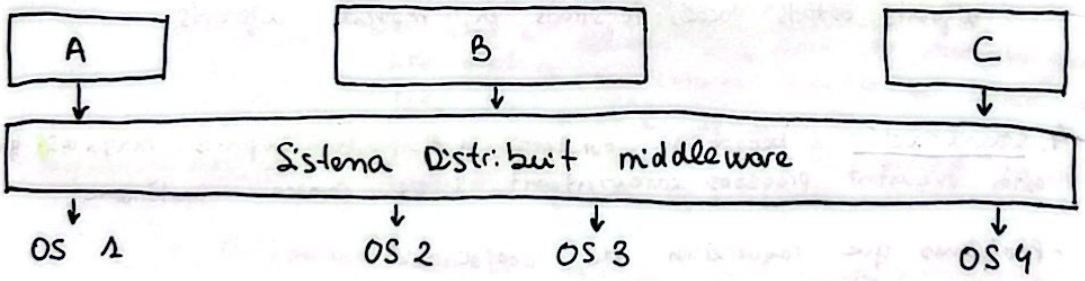
- Un sistema distribuit es quan la fallada d'un sistema no t'implica no poder fer alguna tasca.

Aspectes bàsics de un sistema distribuit

1. Transparència de distribució
2. Interdependència de components

- Conjunt de ordinadors independents que semblen un únic i coherent sistema.
- Sistema on els components estan connectats comunicats i coordinats, només mitjançant missatges.

Independent computers Single-system view : middleware



Conseqüències de la definició:

- Components autonoms i heterogènis
- Coneixement local
- Rellotges no syncronitzats
- Procesos concurrents

partial failure

- Els components poden fallar independentment
- Els components poden no sempre estar disponibles

Perque construim sistemes distribuits?

Perque els ordinadors poden tenir diferents funcionalitats, el treball pot estar dividit entre més ordinadors, hi ha redundancia per si falla algún sistema, per compartir recursos i perqué és més econòmic ja que diversos ordinadors petits es més barat que un de gran.

REPTES / DESAFIOS
OBJECTIUS DELS SISTEMES DISTRIBUITS

1. Heterogeneïtat → Necessitat de adaptar cada ordinador.
2. Seguretat → necessitat de integritat, privacitat i autentificació entre dominis
3. Falta de visió global → es impossible tenir una visió global de forma instantània i exacta.

a) Temps Global → "Tema 3"

. Baixa precisió de rellotge en cada node.

• Desviació de rellotge

• ↳ Sincronitzar les màquines amb un rellotge global
↳ Acordar l'ordre en el que s'executen els events a res de mirar el temps en que s'han fet.

b) Estat Global → "Fora d'aquest curs"

• Cada proces és independent

• La perfecta sincronització de rellotge és inviable.

↳ Hem de fer un estat global significatiu a partir dels diferents estats locals registrats en moments diferents.

4. Coordinació → Necessiten cordinar accions en diferents maquines que estàn executant processos concurrentment i de manera autònoma.

• Problemes que requereixen de coordinació

1 - Exclusió mútua → coordinar l'acces a un recurs compartit

2 - Elecció de líder → elecció de la màquina líder

3 - Atomic multicast → enviament de missatges

4 - Consens → posarse d'acord

5. Asincronia

• Els missatges triden un temps variable en ser enviats
Dos tipus de sistemes distribuits

• Síncrons
- Tenen timeouts i es temps d'execució i missatges tenen límits coneguts.

• Asíncrons
- No es pot fer suposicions sobre els intervals de temps i es temps

SOOX - SISTEMES OPERATIUS DISTRIBUITS EN XARXA

TEMA 1 - CONCEPTS OF DISTRIBUTED SYSTEMS

REPTES DELS SISTEMES DISTRIBUITS

5. Asincronia → dispositius síncrons o asíncrons

6. Accesibilitat / Apertura

1. **Interoperabilitat**: components de diferents proveedors poden treballar conjuntament apoyarse en els recursos de cadascú.

2. **Portabilitat**: un component fet per un sistema determinat pot correr sense ser modificat en un altre sistema.

3. **Extensibilitat**: poder afegir nous components o remplaçar-los sense afectar el sistema.

=> Oferir serveis d'acord amb interfícies estàndard que descriuen la seva semàntica i sintaxis.

7. Transparència

- Poder presentar un sistema distribuit com un sol sistema

Access	hide
Location	hide where a resource is located
Migration	hide that a resource may move to other location
Relocation	hide that a a resource could be moved while in use.
Replication	hide the presence of multiple copies of a source.
Concurrency	hide that a resource may be accessed concurrently by several users.
Failure	hide the failure of a source

La transparència és un objectiu però assolir-la pot ser una altra història.

A. Latències degut a una distància molt gran (geogràfica)

B. Ocultar totes les fallades és impossible
↳ En un sistema asíncron no pots distingir un node lent d'un node que ha fallat.

C. Compensar entre transparència i rendiment

D. Exposar la distribució podria ajudar

ex. utilitzar una impresora ocupada a més d'una impresora però propera (nearby)

lliure en un altre edifici.

distance is important

8. Tolerància a fallades

- Fallos en un únic sistema afecten a tot el sistema.
- Fallos en sistemes distribuits poden ser parcials. Només fallen alguns nodes.
- Un sistema amb tolerància a fallades és capaç de conèixer les seves especificacions tot i tenint fallades parcials.
- La probabilitat de fallada augmenta a mesura que el nombre de nodes també ho fa.

Tipus de fallades

Crash	Usar detectors de fallades per saber quins processos fallen.
Omission	2 implementations ⟶ Heartbeats I m alive!
Response	⟶ Pinging Are you alive? Yes!
Timing	
Arbitrary	

- Com de fiables són els detectors?

A fiables B. No fiables

A. Fiables: un procés pot ser

- Inesperat: evidència recent de que el procés no ha fallat. ← ex. ha rebut fa poc un missatge
- Esperat: pista de que pot ser que el procés hagi fallat. ← ex. fa molta estona que no rep un missatge

B. No Fiable: un procés pot ser inesperat o...

- Fallat: el detector ha dit que el procés ha fallat.

(⇒) Això només és possible en els sistemes síncrons.

- Amagar la fallada d'altres processos usant la redundància.

1. Informació redundant • Afegir bits per detectar errors o per recuperar
2. Redundància física ⟶ "Tema 5" • Replicar hardware o software
3. Temps redundant ⟶ "Tema 2" • Repetir una operació fins que la resposta es rebi

SODX - SISTEMES OPERATIUS DISTRIBUITS EN XARXA

TEMA 1 - CONCEPTS OF DISTRIBUTED SYSTEMS

REPTES DELS SISTEMES DISTRIBUITS

9. Escalabilitat :

- Poder suportar l'augment d'usuaris la localització geogràfica o el nombre de dominis administratius.

A Número d'usuaris → mida escalable

B Màxima distància entre nodes → escalabilitat geogràfica

C Numero de dominis administratius → escalabilitat administrativa

Problemes d'escalabilitat

- Serveis centralitzats
- Informació centralitzada
- Algoritmes centralitzats

Tècniques d'escalabilitat

1. Amagar latències de comunicació
2. Distribució
3. Replicació / caching
4. Utilitzar algoritmes descentralitzat

- **Tècniques d'escalabilitat**

1. Amagar latències de comunicació

+ Comunicació asíncrona
+ Ajuda a assolir escalabilitat geogràfica
↓ No totes les aplicacions poden utilitzar aquest model

2. Distribució

+ Dividir les dades i càlculs entre diferents ordinadors
ex. DNS, move computations to clients Java Applets

3. Replicació / Caching

+ Replicar components en un sistema distribuit
+ El client accedeix com utilitzar el recurs. → Caching
+ Tenir múltiples còpies pot desencadenar en inconsistències.
↳ Consistència requereix de fer una sincronització global.

4. Utilitzar algoritmes descentralitzats

+ Ningua màquina té accés a la informació total del estat del sistema.
+ Decisions basades en la màquina local.
+ La falla de una màquina no arruina l'algoritme.

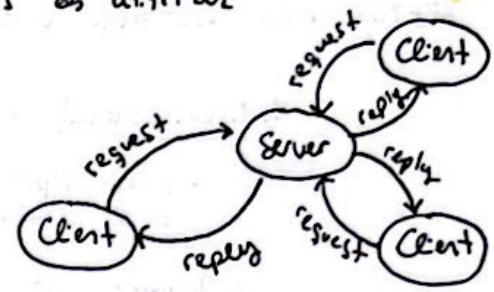
ARCHITECTURES FOR DISTRIBUTED SYSTEMS

A. Arquitectura Client - Servidor

Els servidors ofereixen serveis i els clients els utilitzen

Request - Reply

Problemes d'escalabilitat i robustesa



CLIENTS

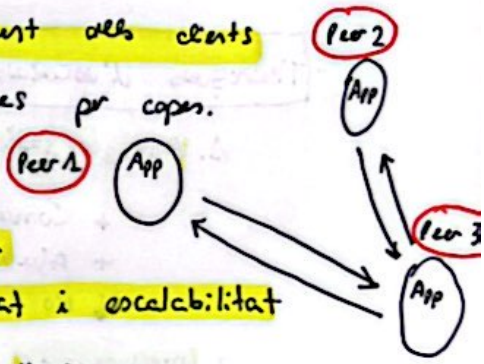
- El software combina interfícies i solucions per aconseguir una distribució transparent.

SERVIDORS

- Sequential one request at a time
- Concurrent crea un procés o thread per a cada request (petició)

- Stateless el servidor no guarda dades dels clients i pot envia l'estat sense informa-los
- Stateful el servidor guarda el comportament dels clients

Els servidors poden organitzar-se en arquitectures per capes.



B. Arquitectura punt a punt

- Elimina la diferencia entre client i servidor
- Xarxa superposada
- Gran disponibilitat i escalabilitat
- No control centralitzat
- Resiliencia a fallades.

1. Arquitectures estructurades punt a punt

- Topologia construida de manera determinista ← Hash tables
- Nodes organitzats segons una estructura de dades basats en IO

2. Arquitectures desestructurades punt a punt

- Topologia basada en algoritmes aleatoris

C. Arquitectura híbrida

- Combinació client-servidor amb punt a punt