



Tema I - Introducció



Tecnologies dels Sistemes d'Informació en la Xarxa



Objectius

- ▶ Entendre per què tot sistema que utilitze una xarxa d'intercomunicació és un sistema distribuït.
- ▶ Identificar què és un sistema distribuït, per què són rellevants i quines són les seues aplicacions principals.
- ▶ Conèixer alguns exemples de sistemes distribuïts.
- ▶ Estudiar l'evolució dels sistemes distribuïts escalables i identificar la computació en el núvol (“*cloud computing*”) com l'etapa actual d'aquesta evolució.



Índex

1. Concepte de sistema distribuït
2. Rellevància
3. Àrees d'aplicació
4. Computació en el núvol
5. Paradigmes de programació
6. Conclusions



I. Concepte de sistema distribuït

- ▶ Conjunt d'agents autònoms
 - ▶ Cada agent és un procés seqüencial que avança al seu propi ritme.
- ▶ Els agents interactuen. Opcions:
 - ▶ Intercanvi de missatges
 - ▶ Memòria compartida
- ▶ Els agents tenen el seu propi estat independent
- ▶ Hi ha algun objectiu comú en aquesta cooperació
 - ▶ Mitjançant el qual es podrà avaluar el comportament global del “sistema”.
- ▶ En la pràctica, un sistema distribuït és un sistema en xarxa.



Índex

1. Concepte de sistema distribuït
2. Rellevància
3. Àrees d'aplicació
4. Computació en el núvol
5. Paradigmes de programació
6. Conclusions



2. Rellevància

- ▶ **Sistemes distribuïts**
 - ▶ Àrea en evolució des dels seus orígens
 - ▶ Rama dels sistemes concurrents
 - ▶ Àmpliament estudiada per la seua utilitat en el disseny de sistemes de temps compartit.
 - ▶ CSD va proporcionar la base per a familiaritzar-nos amb múltiples aspectes dels sistemes concurrents.
 - ▶ Reforçada amb l'evolució de les xarxes d'ordinadors.
 - ▶ Com aconseguir que tots aquests ordinadors facen alguna cosa globalment útil?



2. Rellevància

- ▶ Aspectes rellevants (presentats en els 80)
 - 1. Millora del rendiment
 - ▶ Seleccionar una activitat (problema) complexa, dividir-la en tasques (subproblemes), assignar cada tasca a un ordinador diferent.
 - 2. Major disponibilitat. Idea bàsica:
 - ▶ Si un ordinador s'avaria, encara hi haurà altres ordinadors capaços d'executar les tasques del que ha fallat.
 - 3. Compartició de recursos
 - ▶ Un ordinador pot tenir recursos (p. ex., impressores, discos...) que altres ordinadors no tinguen (i que no necessiten tenir).
 - ▶ Ha de ser possible l'accés a recursos des de qualsevol ordinador.



2. Rellevància

- ▶ Totes aquestes raons són encara vàlides perquè l'entorn de computació actual **ESTÀ** distribuït i interconnectat
 - ▶ Infinitat d'“ordinadors” connectats
 - ▶ Infinitat de serveis remots
 - ▶ Accedits com a recursos compartits
 - ▶ Tots coneixem i utilitzem la web
- ▶ Desafiaments
 - ▶ Aprofitar la connectivitat per a obtenir resultats útils
 - ▶ Crear subsistemes capaços de proporcionar serveis robusts
 - Com se les apanya Google per a implantar el seu servei de cerca?
 - Com gestiona Dropbox l'ús compartit de fitxers per part de milions d'usuaris?
 - Com distribuir entre milions de voluntaris la simulació de nous fàrmacs contra el càncer?



Índex

1. Concepte de sistema distribuït
2. Rellevància
3. Àrees d'aplicació
4. Computació en el núvol
5. Paradigmes de programació
6. Conclusions



3. Àrees d'aplicació

- ▶ Les més destacables són:
 1. *World Wide Web*
 2. *Xarxes de sensors*
 3. *Internet of Things*
 4. *Computació cooperativa*
 5. *Clusters altament disponibles*
- ▶ Les tractem a continuació...



3.1. Aplicació a WWW

- ▶ Basada en el model client/servidor.
- ▶ El servidor espera peticions de documents.
 - ▶ Les peticions impliquen la lectura o modificació d'un document.
- ▶ Els clients són els navegadors web, que envien i reben documents
 - ▶ Els navegadors analitzen el document cercant metadades.
 - ▶ Els enllaços són un cas particular de metadades que apunta a altres documents.
 - ▶ Els documents poden estar en un altre servidor.
- ▶ Paradigma simple i potent
 - ▶ Dissenyat inicialment per a compartir documents.
 - ▶ Estès per a permetre que les peticions sobre documents es convertiren en peticions de servei
 - ▶ Els “documents” retornats inclouen el resultat de la petició efectuada.



3.2. Aplicació a xarxes de sensors

- ▶ Han sorgit gràcies al cost descendent dels equips.
- ▶ Mini-ordinadors de propòsit específic
 - ▶ “Motes”
- ▶ Encastats en dispositius d'ús quotidià
 - ▶ P. ex., en alguns electrodomèstics
- ▶ Contenen sensors
 - ▶ Humitat, temperatura, consum elèctric...
- ▶ Ampli rang d'aplicacions potencials
 - ▶ Vigilància
 - ▶ Detecció de desastres (químics, biològics...)
 - ▶ Monitoratge del consum elèctric
 - ▶ ...



3.3. Aplicació a la “Internet of Things”

- ▶ Motivació: facilitar la connectivitat i interoperabilitat de tots els dispositius
 - ▶ Generalització de les xarxes de sensors
 - ▶ Tots els dispositius poden interactuar entre si
 - ▶ Els dispositius poden alterar el seu entorn físic
 - ▶ S'obrin nous escenaris
 - ▶ Ciutats intel·ligents
 - ▶ Automatització de múltiples processos (construcció, fabricació...)
 - ▶ Cura mèdica informatitzada
 - ▶ ...

3.3. Aplicació a la “Internet of Things”





3.4. Aplicació a la computació cooperativa

- ▶ La major part dels recursos computacionals s'infrutilitzen
 - ▶ Els ordinadors personals passen moltes hores diàries sense fer res
- ▶ Molts problemes científics i d'enginyeria poden dividir-se en peces menors (tasques)
 - ▶ Cada tasca pot resoldre's en un interval breu.
 - ▶ Els resultats de cada tasca poden compondre's per a construir la solució del problema complet.
- ▶ Els servidors poden obtenir una instància d'aquests problemes
 - ▶ El servidor crea un conjunt de tasques
- ▶ Els ordinadors amb accés a Internet poden subscriure's per a rebre tasques que resoldre
 - ▶ Instal·len un client especial: el “*runtime*” per a executar tasques
 - ▶ El client es registra en el servidor
- ▶ El servidor distribueix tasques entre els clients registrats i arreplega els seus resultats



3.5. Aplicació als *clusters* altament disponibles

- ▶ Fins ara hem presentat àrees d'aplicació dedicades a la cooperació i la compartició de recursos.
- ▶ Fet:
 - ▶ Els dispositius fallen. Els ordinadors són dispositius. Ells fallen en algun moment amb probabilitat 100%.
- ▶ Fet:
 - ▶ No tots els dispositius d'un sistema fallen alhora.
 - ▶ Per què podria passar això?
- ▶ Alguns entorns necessiten un alt nivell de disponibilitat
 - ▶ Bancari
 - ▶ Empresarial
 - ▶ Assistència mèdica
 - ▶ ...
- ▶ Convé tenir més d'un dispositiu per a suportar les situacions de fallada.



3.5. Aplicació als *clusters* altament disponibles

- ▶ **Cluster altament disponible:**
 - ▶ Conjunt d'ordinadors amb programes servidors dels quals els clients depenen en tot moment.
 - ▶ Típicament mantenen un conjunt de dades crític.
 - ▶ Dissenyats amb protocols específics per a suportar fallades en els ordinadors.
 - ▶ Dos aspectes principals:
 - ▶ Mantenir la integritat de la informació gestionada
 - ▶ Mantenir la disponibilitat dels servidors



3.5. Aplicació als *clusters* altament disponibles

- ▶ Principal tendència actual per a construir i facilitar serveis
- ▶ Fets acceptats:
 - ▶ S'infraprofita la potència de còmput amb les architectures tradicionals
 - ▶ Ja s'ha discutit prèviament
 - ▶ Resulta car establir centres de còmput per a empreses, amb totes les aplicacions que aquestes requereixen:
 - ▶ Adquirir programes i equips
 - ▶ Sota dels enginyers que administren aquestes aplicacions i equips.
 - ▶ Cost de l'energia elèctrica
 - Encara més car si considerem la infraprofita

... això condueix a la computació en el núvol (“*cloud computing*” → CC)...



Índex

1. Concepte de sistema distribuït
2. Rellevància
3. Àrees d'aplicació
4. Computació en el núvol
5. Paradigmes de programació
6. Conclusions



4. Computació en el núvol (CC: *cloud computing*)

► Parlarem de:

1. Programes i serveis
2. Rols en el cicle de vida d'un servei
3. Evolució dels serveis de programari
 - a) *Mainframes*
 - b) Ordenadors personals
 - c) Centres de còmput empresarials
 - d) SaaS
 - e) IaaS
 - f) SaaS sobre IaaS
 - g) PaaS
4. Resum



4.1. Programes i serveis

- ▶ Objectiu general del CC:
 - ▶ Convertir la creació i explotació dels serveis de “programari” en un procés més senzill i més eficient.
- ▶ Un fet acceptat i obvi:
 - ▶ Els programes sempre s'han desenvolupat per a oferir algun tipus de servei.
 - ▶ Amb l'ajuda dels ordinadors, per descomptat.
- ▶ L'evolució de la indústria informàtica ha ocultat parcialment aquest fet:
 - ▶ La indústria dels ordinadors personals ha imposat una manera particular d'interacció dels usuaris amb els seus ordinadors.



4. 2. Rols en el cicle de vida d'un servei

- ▶ Considerem aquests quatre rols:
 - ▶ **Desenvolupador**
 - ▶ Implanta els components de les aplicacions
 - ▶ **Proveïdor de serveis**
 - ▶ Decideix les característiques del servei, els components que el constitueixen i com ha de ser configurat i administrat
 - ▶ **Administrador del sistema**
 - ▶ S'encarrega que cada peça de *programari* i *maquinari* estiga en el seu lloc apropiat i adequadament configurat.
 - ▶ **Usuari**
 - ▶ Accedeix al servei



4.3. Evolució dels serveis: a) *Mainframes*

- ▶ L'administració *del sistema* està realitzada per especialistes
- ▶ Molt pocs focus de contenció
 - ▶ Sistemes amb una reduïda base d'usuaris
- ▶ Ús eficient dels equips
 - ▶ Compartits per múltiples usuaris
 - ▶ Cost baix per a cada usuari
 - ▶ Cost d'adquisició suportat pel propietari de l'equip: institució
- ▶ Els usuaris segueixen rols mixts
 - ▶ Molts van ser desenvolupadors
 - ▶ Molts van ser també els seus propis proveïdors de serveis
 - ▶ Amb els programes que ells van desenvolupar
 - ▶ Amb els programes desenvolupats per uns altres
- ▶ Els usuaris estaven implicats en massa detalls de la gestió dels serveis que ells mateixos havien d'utilitzar



4.3. Evolució dels serveis:

b) Ordinadors personals

- ▶ Els ordinadors personals van ser el resultat de la tendència a una major potència de còmput en cada equip
 - ▶ Els usuaris ja no necessitaven accedir a un “*mainframe*” en un centre de càlcul.
- ▶ S'elimina la contenció
 - ▶ Un dels principals arguments de venda d'aquest paradigma
- ▶ Ús deficient dels recursos: l'ordinador s'infrutilitza
- ▶ Inversió directa a realitzar: cost de la compra
- ▶ Es racionalitza el rol de desenvolupador
 - ▶ Empreses especialitzades construeixen i comercialitzen els programes
- ▶ Però encara s'exigeix que l'usuari exercisca diversos rols:
 - ▶ Proveïdor de serveis
 - ▶ Ha de seleccionar quins programes necessitarà per a realitzar les seues tasques
 - ▶ Administrador del seu ordinador personal
- ▶ Entorn massa complex per a la majoria dels usuaris



4.3. Evolució dels serveis:

c) Centres de còmput empresarials

- ▶ Implantats mitjançant *clusters* altament disponibles
- ▶ Característiques similars a l'entorn d'ordinadors personals
 - ▶ L'usuari és ara l'empresa
 - ▶ Personal especialitzat que segueix compartint els rols d'administrador de sistema i proveïdor de serveis: Cost molt alt
 - ▶ En ocasions s'afeg el rol de desenvolupador de programes interns
- ▶ Variant basada a mantenir aquests programes en centres de dades externs
 - ▶ Evita el cost d'adquisició dels equips
 - ▶ Redueix i externalitza el cost d'administració i manteniment dels equips
 - ▶ Evita el cost fix de consum elèctric
 - ▶ La gestió dels costos informàtics resulta més senzilla



4.3. Evolució dels serveis:

d) *Software as a Service (SaaS)*

- ▶ S'accedeix als serveis a través de la xarxa
 - ▶ Mitjançant un navegador web
- ▶ Separació clara del rol d'usuari
 - ▶ El servei està definit per una tercera part: el proveïdor de serveis
- ▶ No queda tan clara la separació dels altres rols
 - ▶ Els programes solen ser desenvolupats inicialment pel proveïdor
 - ▶ Totes les tasques d'administració solen recaure en el proveïdor
 - ▶ Incloent l'administració d'equips en els centres de dades
 - ▶ Incloent l'administració dels programes instal·lats en aquests equips
- ▶ Inicialment, sorgeixen algunes ineficiències:
 - ▶ Falta de flexibilitat en la distribució dels equips
 - ▶ Comporta que el proveïdor es limite a cert ús dels recursos
 - ▶ Limita la compartició de recursos
 - ▶ Contenció limitada: es reserven recursos per a la demanda esperada

4.3. Evolució dels serveis: d) SaaS

- ▶ Factors que van conduir cap als sistemes SaaS:
 - ▶ Millora de les tecnologies de xarxa
 - ▶ Major ample de banda
 - ▶ Menor retard
 - ▶ Capacitat dels centres de dades existents
 - ▶ Va possibilitar l'oferta de serveis a usuaris externs
 - ▶ Milliores en la tecnologia dels navegadors web
 - ▶ Tipificades en el terme “Web 2.0”
 - ▶ Navegadors capaços d'executar localment interaccions complexes
 - Permeten interfícies d'usuari més atractives
 - Escalabilitat millorada: Menor càrrega en el servidor



4.3. Evolució dels serveis:

e) *Infrastructure as a Service (IaaS)*

- ▶ Facilita la capacitat per a assignar o redistribuir els recursos de còmput i de xarxa sota petició.
 - ▶ Peticions via API a un servei (el servei IaaS)
 - ▶ Possibilitat de carregar imatges de sistema operatiu sobre aquests ordinadors
 - ▶ Possibilitat de sol·licitar capacitats concretes per als ordinadors i els recursos de xarxa
- ▶ Possible gràcies a la tecnologia de virtualització del maquinari
 - ▶ L'assignació de recursos de còmput (virtuals) és fàcil i ràpida
 - ▶ Fàcil configuració de la capacitat dels recursos de còmput
 - ▶ Resulta fàcil la instal·lació d'una imatge de sistema sobre una màquina virtual



4.3. Evolució dels serveis: f) SaaS sobre IaaS

- ▶ IaaS introdueix un model de “pagament per ús”
 - ▶ Una característica central de la computació en el núvol
- ▶ Facilita la creació de SaaS que s'adapten a la càrrega generada pels seus usuaris
 - ▶ Com més gran siga aquesta càrrega, se sol licitarà un major nombre de recursos a la infraestructura
 - ▶ *Elasticitat*: una altra característica central dels sistemes Cloud
 - ▶ Trasllada el model de “pagament per ús” als sistemes SaaS
 - ▶ Els usuaris d'un sistema SaaS també paguen segons la seua utilització del servei
- ▶ Obliga a un ús eficient dels recursos per part del proveïdor SaaS
 - ▶ La majoria dels costos són variables
 - ▶ No hi ha costos directes per reservar certa capacitat (compra o compromís de pagament)
 - ▶ El que s'estalvie beneficiarà a l'usuari SaaS: mercat competitiu de serveis



4.3. Evolució dels serveis: f) SaaS sobre IaaS

- ▶ Els proveïdors IaaS prenen els riscos de la inversió directa (compra dels recursos físics)
 - ▶ Esperen una gran població de proveïdors SaaS
 - ▶ Al seu torn, facilitant serveis a un alt nombre d'usuaris SaaS
 - ▶ Gran demanda de recursos virtualitzats
- ▶ El proveïdor SaaS encara exerceix diversos rols
 - ▶ Proveïdor de serveis de *programari* (el seu rol natural)
 - ▶ Ha de gestionar l'assignació de recursos del *maquinari*
 - ▶ Ha de gestionar les imatges de sistema a instal·lar, les seues actualitzacions i la base de programes a utilitzar sobre aqueixos sistemes
 - ▶ Ha d'implantar la seua pròpia estratègia de gestió de serveis
 - ▶ Mecanismes de monitoratge
 - ▶ Mecanismes d'actualització

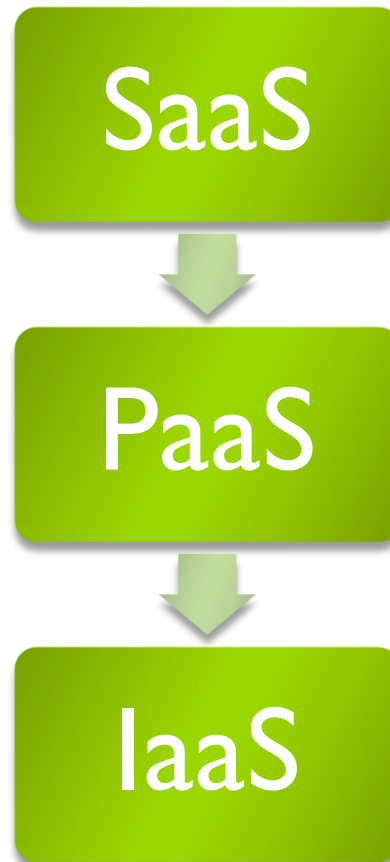


4.3. Evolució dels serveis:

g) *Platform as a Service (PaaS)*

- ▶ Hauria d'eliminar qualsevol tasca estranya per als proveïdors SaaS
 - ▶ Encara en els seus inicis, desafortunadament.
- ▶ Serà equivalent a un sistema operatiu
 - ▶ Especifica un model de serveis sobre el qual basar l'especificació dels SaaS i el desenvolupament dels seus components de *programari*
 - ▶ Inclou els aspectes següents
 - ▶ Models de configuració i de gestió del cicle de vida (incloent les relacions de dependència entre components)
 - Mecanismes de composició, configuració, desplegament i actualització
 - ▶ Model de rendiment
 - Monitoratge automàtic de paràmetres rellevants
 - Expressió de punts d'elasticitat
 - Reconfiguració automatitzada en funció de la càrrega

- ▶ Estructura ideal, en nivells:





4.4. Resum

- ▶ La computació en el núvol (CC) se centra en l'eficiència i la facilitat d'ús:
 - ▶ **Compartició eficient dels recursos**
 - ▶ Consumir solament el que es necessita
 - ▶ Pagar solament pel que s'ha utilitzat
 - ▶ **Adaptació senzilla a una quantitat d'usuaris variable**
 - ▶ **Facilitar formes senzilles per a desenvolupar i proveir un servei**
- ▶ Tres nivells de serveis en el núvol han sigut identificats:
 - ▶ ***Software as a Service (SaaS)***
 - ▶ El seu objectiu és facilitar aplicacions com a servei a un gran nombre d'usuaris
 - ▶ ***Platform as a Service (PaaS)***
 - ▶ Aconsellable per a automatitzar la gestió de recursos per als SaaS i la fàcil creació i desplegament d'aquests serveis
 - ▶ ***Infrastructure as a Service (IaaS)***
 - ▶ Proporciona elasticitat per als sistemes SaaS
- ▶ Des de la perspectiva dels usuaris, el CC és similar a un retorn a l'era dels “mainframes”.



Índex

1. Concepte de sistema distribuït
2. Rellevància
3. Àrees d'aplicació
4. Computació en el núvol
5. Paradigmes de programació
6. Conclusions



5. Paradigmes de programació

- ▶ La forma més comuna d'organitzar un sistema distribuït es basa a convertir cada procés en un “servidor”
 - ▶ Rep peticions, les processa i retorna respostes
- ▶ Els servidors, al seu torn, sol·liciten servei a altres servidors
 - ▶ Poden necessitar aquests serveis per a completar una petició rebuda
- ▶ Per a ser escalable, un servidor no ha de suspendre's mentre gestione una petició
 - ▶ Ha de ser capaç d'acceptar altres peticions



5.1. Servidors concurrents (estat compartit)

- ▶ Programes concurrents (amb múltiples fils)
 - ▶ Cada petició és servida pel seu propi fil
 - ▶ Tots els fils comparteixen un estat global
 - ▶ S'utilitzen mecanismes de control de concurrència per a garantir atomaticitat
- ▶ Avantatges
 - ▶ Els fils poden suspendre's esperant peticions, sense suspendre tot el procés servidor
- ▶ Inconvenients
 - ▶ La programació multi-fil té les seues pròpies sobrecàrregues
 - ▶ Necessita mecanismes de control de concurrència. Implica suspensió.
 - ▶ La programació concurrent amb memòria compartida resulta...
 - ▶ difícil d'implantar sense errors
 - ▶ difícil de raonar sobre com es comporta (i justificar la seua correcció)
- ▶ Entorns predominants:
 - ▶ Java
 - ▶ .NET

5.2. Servidors asincrònics

- ▶ La programació asincrònica (o programació dirigida per esdeveniments)..
 - ▶ correspon fidelment al model de programació guarda/acció.
 - ▶ genera programes amb múltiples activitats, però...
 - ▶ l'estat compartit mai podrà ser accedit concurrentment per aquestes activitats
- ▶ Els “esdeveniments” són les “guardes”
- ▶ Les accions s'estableixen com “*callbacks*” dels esdeveniments
 - ▶ Per a facilitar la programació es relacionen dinàmicament les accions/guardes.
 - ▶ En implantar les accions, certs mecanismes del llenguatge de programació permeten establir quin estat es veurà afectat per elles.
 - ▶ Es redueix la complexitat per a “preparar” l'estat que relacionarà accions internes.
- ▶ Les accions preparades per a execució s’ “encuen”
 - ▶ S'executaran seguint l'ordre FIFO de la cua



5.2. Servidors asincrònics

▶ Avantatges

- ▶ La complexitat en la gestió d'estat compartit desapareix
 - ▶ Però ha de considerar-se l'ordre d'activació (és a dir, d'“encuat”) per a evitar sorpreses
- ▶ Menor sobrecàrrega ja que no es necessita un suport multi-fil
 - ▶ Major escalabilitat
- ▶ Model més pròxim a la forma real de treball en un sistema distribuït: dirigit per esdeveniments
 - ▶ Resulta més fàcil raonar sobre què està ocorrent a cada moment

▶ Inconvenients

- ▶ Es necessita una gestió adequada de l'estat en implantar les accions
- ▶ Es necessita que tot l'entorn siga asincrònic, no solament la comunicació entre processos
 - ▶ Els serveis del sistema operatiu han de ser asincrònics, per a evitar suspensions

▶ Entorns predominants amb suport natiu en el llenguatge:

- ▶ NodeJS
- ▶ Async .NET



Índex

1. Concepte de sistema distribuït
2. Rellevància
3. Àrees d'aplicació
4. Computació en el núvol
5. Paradigmes de programació
6. Conclusions



6. Conclusions

- ▶ Els sistemes en xarxa són sistemes distribuïts
 - ▶ La majoria de la computació es desenvolupa avui dia sobre la xarxa
 - ▶ Per tant, és distribuïda
 - ▶ Un disseny i desenvolupament adequats requereixen un profund coneixement de la programació concurrent i de les architectures utilitzades
- ▶ Ampli conjunt d'àrees d'aplicació ja en explotació
- ▶ La computació en el núvol com a última etapa important en l'evolució de la computació
 - ▶ Caracteritzada per l'eficiència en l'ús dels recursos
 - ▶ Amb un model d'accés de “pagament per ús”
 - ▶ Elasticitat i escalabilitat com a objectius principals
- ▶ Dos paradigmes de programació per a implantar serveis distribuïts:
 - ▶ Servidors concurrents (multi-fil)
 - ▶ Han de gestionar condicions de carrera. Pot hi haver bloquejos.
 - ▶ Servidors asincrònics
 - ▶ Orientats a esdeveniments. Fàcilment escalables.