

**FIB**Facultat d'Informàtica
de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

CONCEPTES AVANÇATS DE SISTEMES OPERATIUS
Departament d'Arquitectura de Computadors

Looking Up Data in P2P Systems

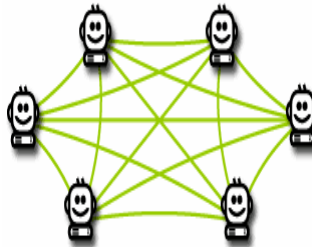
(Seminaris de CASO)

Autors

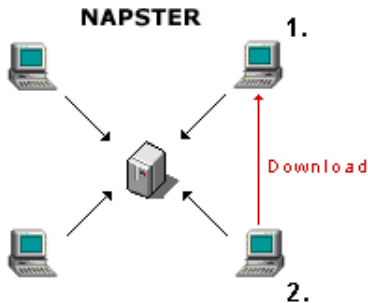
David Domínguez Sal
David Otero Cuadros

Què és una xarxa P2P ?

- Sistema distribuït.
- Software equivalent a tots el nodes (mateixa funcionalitat).
- Màquines no especialitzades: funcionament de la xarxa amb PC's.
- Possibilitat de servei:
 - Emmagatzemament i compartició de fitxers.
 - Computació distribuïda.
 - Xarxes robustes a atacs o caigudes.
- No existeix el concepte clàssic de servidor... **Com buscar la informació?**



Napster: Primera gran xarxa "P2P"



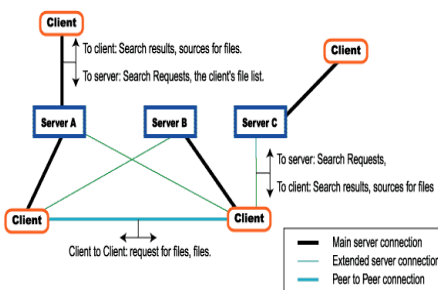
- Semi - P2P: Un Servidor Central ☹.
- Compartició música.
- Búsqueda de fitxers:
 - Cerca de fitxers per nom!!
 - Base de Dades centralitzada.
 - Al connectar, informar al servidor central dels fitxers de què es disposen.
 - L'usuari pregunta per un nom al servidor central i li retorna un conjunt d'adreces d'altres clients amb el fitxer.

☞ Problemes

Escalabilitat i Fiabilitat.

Edonkey: Multiples Servidors

- Gran nombre de petits servidors (PC's domèstics).
- Compartició de fitxers: musica, documents, aplicacions...
- Búsqueda de fitxers:
 - Identificació de fitxers per hash: md5.
 - Cada client es connecta a un servidor d'una llista.
 - Es fa la petició de cerca al servidor i aquest retorna els noms de fitxers amb els hash de cadascun. El client pot fer peticions de cerca a altres servidors.
 - El client fa peticions del hash seleccionat als servidors i li retornen adreces de clients que el tenen.
- Manteniment llistat servidors:
 - Els servidors intercanvien missatges amb llistes de servidors.
 - El client al connectar-se al servidor de cerques rep la llista d'altres servidors



☞ Problemes

Necessita servidors dedicats

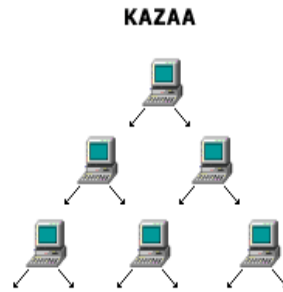
Kazaa: Estructura jeràrquica

- Estructura en forma d'arbre
- No hi ha servidors dedicats! Tots els nodes corren el mateix software: són clients i servidors.
- Búsqueda:
 - La cerca comença a l'arrel de l'arbre.
 - Cada node al preguntar-li pel fitxer mira si el té i en cas contrari pregunta als seus descendents.
 - Un cop trobat es retorna la traça desfent el camí seguit anteriorment.

Problemes

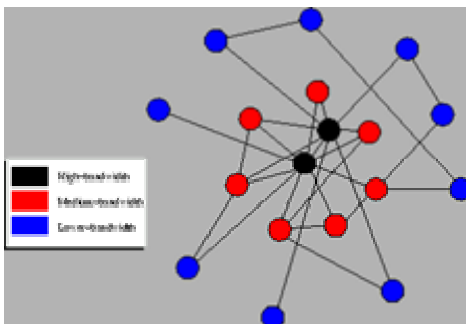
Hardware dels nodes superiors potent.

Problemes de connectivitat en cas de caiguda de nodes superiors de l'arbre.



Gnutella: Flooding

- No existeix servidor: estructura en forma de graf
- Búsqueda de fitxers:
 - Cada client té una llista de veïns
 - Per cercar es llença un missatge broadcast als seus veïns. Cada veï reenvia el missatge si no l'havia reenviat abans (el graf pot tenir cicles).
 - Dependent de la velocitat de connexió el nombre de veïns es mes gran o més petit



Problemes

Consum gran d'amplada de banda: flooding

Cicles en el graf

Propostes d'Algoritmes de cerca

- Característiques:
 - Cerca ràpida
 - Connectivitat
 - Escalabilitat
 - Tolerabilitat a fallades de nodes
 - No privacitat
 - Peer to Peer pur: sense servidors
- Basats en la Distributed Hash Table (DHT)
 - CAN
 - CHORD
 - Pastry
 - Tapestry

Què és la DHT? (I)

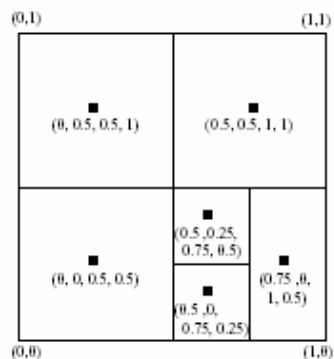
- És una taula de hash no centralitzada.
- Implementa una única operació: **lookup(key)**
 - La clau es genera amb una funció de hash. Ex. SHA-1
- En una aplicació distribuïda d'enmagatzemament per publicar un fitxer:
 - Codifica el nom del fitxer amb el hash de forma "unívoca"
 - Cridar lookup(key) i enviar el fitxer al node resultant.
- ... I per obtenir el fitxer des d'un node qualsevol:
 - Obtenir el nom del fitxer
 - Convertir-ho a una key
 - Fer el lookup(key) i demanar al node resultant una còpia del fitxer

Què és la DHT? (II)

- Diferència fonamental amb els sistemes actuals: la informació no es troba necessàriament en el node que la publica sino en el que li pertoca pel hash.
- Per motius de robustesa i/o popularitat pot replicar-se la informació a diferents nodes i que el retorn de la key sigui múltiple.
- Què falta per decidir als algoritmes?
 - Mapejar les claus als nodes de forma equilibrada
 - Encaminament d'una cerca cap al node que conté el fitxer
 - Construcció de taules d'enrutament.

CAN (I)

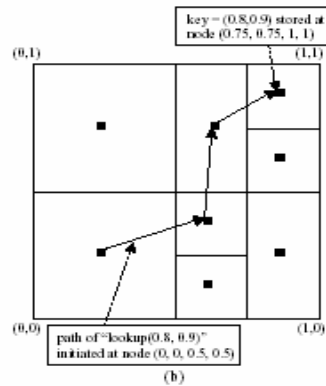
- Reparteix els nodes en un espai cartesià d-dimensional per implementar l'abstracció de la DHT (cada node té 'd' coordenades).
- Cada node té un hipercub del que s'encarrega i el node que estarà al seu centre queda a càrrec del hipercub.
- Cada node manté una taula d'enrutament dels seus veïns a l'espai (dos nodes són veïns si les seves zones comparteixen un hiperplà (d - 1) dimensional).



Esquema del model CAN amb dues dimensions ($d=2$) i sis nodes repartint-se l'espai.

CAN (II)

- Per buscar un fitxer: (cost $O(d \cdot N^{1/d})$)
 - Fem un hash sobre el nom del fitxer i retorna un punt de l'espai.
 - Dels nostres veïns preguntem al més proper a aquest punt.
 - Recursivament els veïns van preguntant als seus veïns fins arribar al node que conté el fitxer i ens l'enviarà.
- Al connectar-se un node:
 - Obté una posició de l'espai.
 - Busca qui estava a càrrec del punt.
 - Parteixen l'antic domini a l'espai per la meitat.
- Al desconectar-se un node:
 - Avisa als seus veïns.
 - Un dels seus veïns annexa aquest espai al seu propi.
 - Un programa en background agafa els nodes que no responen i annexa la seva zona a veïns.



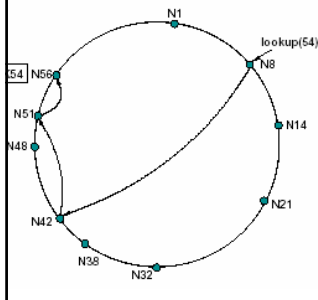
El node (0,0,0.5,0.5) busca el fitxer indexat al punt (0.8,0.9)

FIB

Seminari de CONCEPTES AVANÇATS DE SISTEMES OPERATIUS
Departament. d'Arquitectura de Computadors - UPC

11

Chord (I)



Estructura circular de les claus a Chord. Cerca del node 8 per la clau 54

- Espai de claus unidimensional. Imaginar l'espai com a circular.
- El node responsable per la clau k es el **successor** de k . És el node el qual el seu identificador segueix el de k .
- Cada node té una taula d'enrutament de $\log(N)$ entrades. Conté la IP del node diametralment oposat (meitat del cercle), d'un quart de cercle, ... en decrements de potències de 2.
 - A més per evitar pèrdues de connectivitat a la xarxa, cada node té les adreces dels r nodes següents.

FIB

Seminari de CONCEPTES AVANÇATS DE SISTEMES OPERATIUS
Departament. d'Arquitectura de Computadors - UPC

12

Chord (II)

- Per buscar un fitxer: (cost $O(\log N)$)
 - Fem un hash sobre el nom del fitxer i retorna un ID.
 - Fem servir els apuntadors de la taula d'enrutament per saltar el major nombre de nodes sense passar-nos del destí.
- Al connectar-se un node n:
 - És fa el hash de n i pregunta a un node del anell on va.
 - Actualitza necessàriament actualitzar la llista de successors del predecessor i la pròpia. El node pren carrec dels nodes que es responsable.
 - Per mantenir el rendiment es van actualitzant en background les taules d'enrutament dels nodes ja existents.
- Al desconectar-se un node:
 - No fa falta avisar abans de desconectar (per tant no hi ha diferencia entre la desconnexió i la caiguda d'un node).
 - Quan un node detecta que un dels seus enllaços no és present busca un altre similar per agafar la seva posició a la taula d'enrutament.
- Per saber més: <http://www.pdos.lcs.mit.edu/chord>

Altres algoritmes: Pastry & Tapestry

- Pastry:
 - Basat en un espai dimensional circular.
 - Similar a Chord. Les taules d'enrutament contenen a més a més enllaços cap als nodes predecessors.
 - Per saber més: <http://www.cs.rise.edu/CS/Systems/Pastry>.
- Tapestry:
 - Cada node té una clau numèrica que l'identifica.
 - Cada node té una taula d'enrutament cap a almenys un node amb un identificador que té com a prefix l'id del node actual amb un dígit diferent. Ex. el node 234 té enllaços cap el 1xx,2xx,3xx..., 21x,22x..., 231, 232...239.
 - Per fer una cerca cada node va afegint un dígit aproximant-se al id que es busca.

Bibliografia

- Què és p2p?
 - <http://www.peer-to-peerwg.org/whatis/>
- Funcionament sistemes actuals
 - http://www.kent.ac.uk/law/undergraduate/modules/ip/handouts/Kazaa_essay.doc
 - <http://www.iespana.es/cuadernos/downloads/edonkey2000.doc>
 - <http://www.thedonkeynetwork.com/pl/tdn.pl?site=clients>
- Sistemes basats en DHT:
 - Looking Up Data in p2p systems:
<http://www.utoronto.ca/~rosselet/cscd58s/tut03/pres03/p2p-lookups.pdf>