Modèles et Systèmes de programmation distribuée

Nhan LE THANH

Cours LpSIL – Spécialité IDEE
Octobre 2004

Rappel du plan du cours

- I. États de l'art
- 1- Objectifs, caractéristiques
- 2- Communication en réseaux
- 3- Structures logiques
- 4- Problématique
- 5- Développement, déploiement

- II. Modèles d'exécution
- 1- Modèles client-serveur
- 2- Modèles de communication par messages
- 3- Modèles de communication par évènements
- 4- Autres modèles

III. Systèmes pair à pair
1- Principes et composantes
2- Calcul distribué Pair à Pair
3- Échange de données Pair à Pair
4- Évolution

IV. Systèmes transactionnels
1- Transactions et transactions réparties
2- Systèmes transactionnels
3- Programmation transactionnelle
4- SGBD répartis

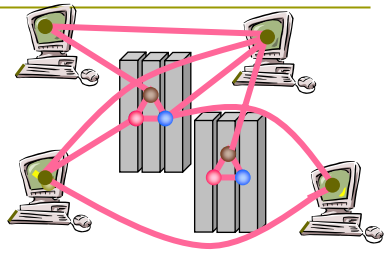
Part 3 : Système Pair à Paire

Plan

1- Principes et composantes
2- Calcul distribué Pair à Pair
3- Échange de données Pair à Pair
4- Évolution

III. Systèmes Pair à Pair
1. Principe et composantes

- Introduction
  - Besoin de partage de ressources
  - Besoins de calcul



- Historique
  - P2P sont nés en 98 avec le système Napster (auteur : Shawn Fanning qui n'avait que 17 ans …)
  - P2P sont pleinement dans son développement
  - P2P ont besoin un contrat communautaire moral

III. Systèmes Pair à Pair
1. Principe et composantes

- Définitions ?
  - "Sharing of Computing Resources via Direct Exchange Between Computers" [Intel, 2000]
  - "Every entity is both a client and a server" - L'isotropie n'est pas fondamentale (Kazaa)
  - "Computing from the edge of Internet" - Aussi "[…] *from the dark matter of Internet*"
  - « P2P is a class of applications that takes advantage of resources -- storage, cycles, content, human presence -- available at the edges of the Internet » [Shirky]

III. Systèmes Pair à Pair
1. Principe et composantes

- Définitions ?

« Le Pair à Pair est une classe de systèmes auto organisants qui permettent de réaliser une fonction globale de manière décentralisée, en tirant parti du partage des ressources et de la puissance de calcul disponibles sur un grand nombre d'extrémités de l'Internet »

Bruno Richard - HP Laboratories Grenoble

III. Systèmes Pair à Pair
1. Principe et composantes

- Principes généraux de P2P
  - Chaque membre est à la fois client et serveur
  - Dimensionnement dynamique
  - Dispose des mécanismes de stockage et de mise en disponibilité de ressources
  - Dispose des mécanismes de recherche d'information
  - Dispose des mécanismes récupération de ressources

III. Systèmes Pair à Pair
1. Principe et composantes

- Exemples

- Systèmes P2P
  - À base de recherche
    - Napster
    - Glutella
    - Kazaa
  - MSN Messenger
  - À base d'archivage
    - Can
    - Chord
    - Freenet

- Système non P2P
  - Téléphone
  - NNTP
  - DNS
  - Microsoft .NET
  - SMTP
  - MOM

III. Systèmes Pair à Pair
1. Principe et composantes

- Caractéristiques :
  - Système d'auto organisation
    - Facilité d'utilisation, d'installation et de démarrage
      - Pas d'administrateur local
      - Pas d'administrateur global
    - Gestion automatique de la dynamicité
      - Volatilité des machines
      - Déconnexions violentes
      - Hétérogénéité
    - Découverte de ressources automatique et annonce automatique
    - Gestion du réseau
      - Adaptation aux conditions (modem, LAN, mobile, …)
      - Support de la mobilité
      - Accès aux ressources à distance

III. Systèmes Pair à Pair
1. Principe et composantes

- Caractéristiques :
  - Anonymat
    - Réseau insaisissable
      - Pas le cas de Napster
    - Publication anonyme
    - Édition anonyme
    - Récupération anonyme
  - Mais aussi possible : groupes privés et fermé

III. Systèmes Pair à Pair
1. Principe et composantes

- Problème de dimensionnement
  - Grand nombre de machines
  - Jusqu'à plusieurs millions
  - Le système doit rester efficace
    - Un serveur central doit être bien dimensionné
      - Travail >= O(N)
      - CPU
      - Réseau
      - Mémoire (dont stockage)
  - Un système P2P doit être en O(log(N)) au moins sur sa fonction essentielle

III. Systèmes Pair à Pair
1. Principe et composantes

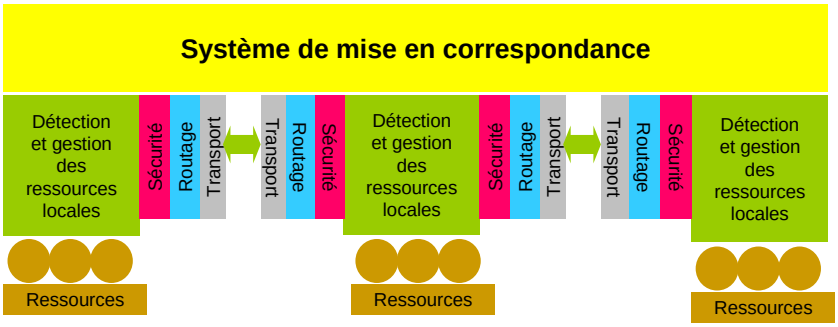
- La liberté et la loi : aspect social
  - Le P2P nécessite un *contrat communautaire* moral
  - Coopération pour le partage de ressources
    - On utilise les ressources des autres
    - On offre ses ressources locales : Fichiers, CPU, stockage, bande passante
  - Pas de *comportement mal sain*
    - Récupération de données sans en partager
    - Rupture du contrat
    - Provoque un déséquilibre du système

III. Systèmes Pair à Pair
1. Principe et composantes

- La liberté et la loi : aspects légaux
  - Est-on responsable du contenu de ses disques ?
    - Certaines jurisprudences semblent le montrer
      - Vente d'objets nazis sur eBay
      - Groupes pédophiles sur Compuserve en Allemagne
  - Peut-on faire une copie de ses propres documents ?
    - A usage personnel pour l'instant
    - Une tentative de limitation est en cours en France
  - Mise des gens en relation
    - Passible de poursuites : Exemple de Napster
      - Limité à une fonction d'annuaire - Rien d'illégal sur les serveurs
      - Attaque de Métallica – Plus de MP3 que d'albums vendus ! 250000 adresses IP tracées
    - A été interdit en Mai 2002
  - Digital Millenium Copyright Act (DMCA)
    - Emprise forte des éditeurs (Hollywood business)
    - Recording Industry Association of America (RIAA) (SACEM aux US)

III. Systèmes Pair à Pair
1. Principe et composantes

- Composantes principales



-**Transport direct: plus performant !** (Napster, Glutella, e-donkey2000)

- Transport indirect : anonymat, prévention de censure !
(Freenet, OceanStore, SETI@home)

III. Systèmes Pair à Pair
1. Principe et composantes

- Architectures principales
  - Plusieurs architectures possibles
    - Centralisée
      - Napster, SETI@home
    - Serveurs distribués
      - eDonkey2000
    - Superpeers
      - Catalogues hiérarchiques
      - Election automatique
      - FastTrack, Kazaa, BearShare
    - Distribuée
      - Gnutella, Freenet, Chord

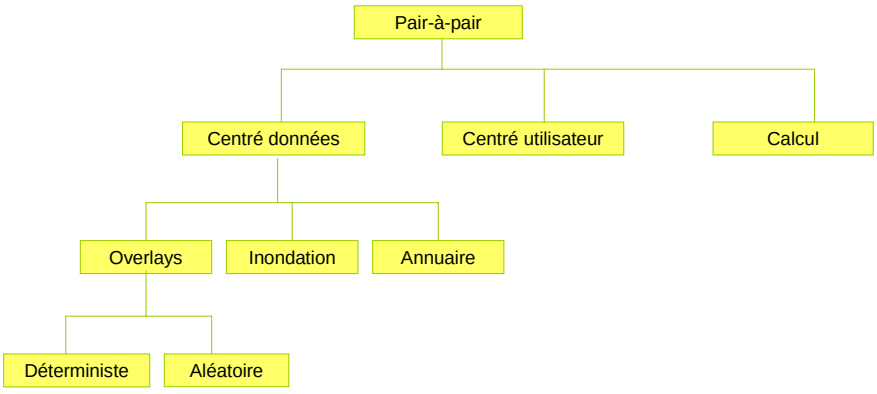
P2P centralisés

P2P hybrides

P2P pures

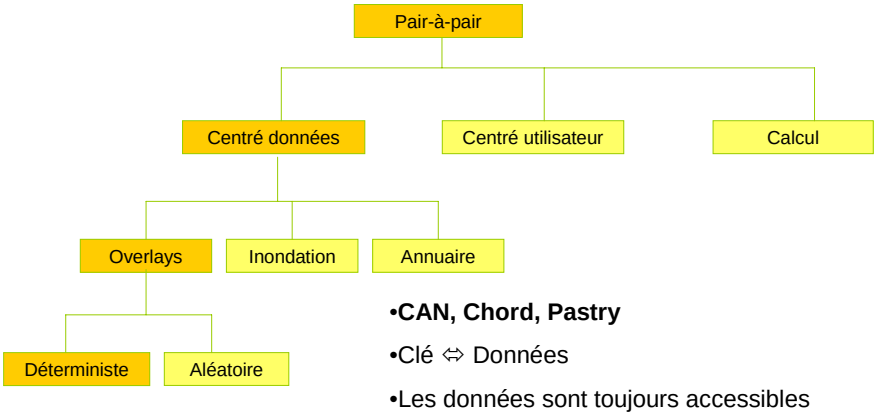
III. Systèmes Pair à Pair
1. Principe et composantes

- Taxonomie



III. Systèmes Pair à Pair
1. Principe et composantes

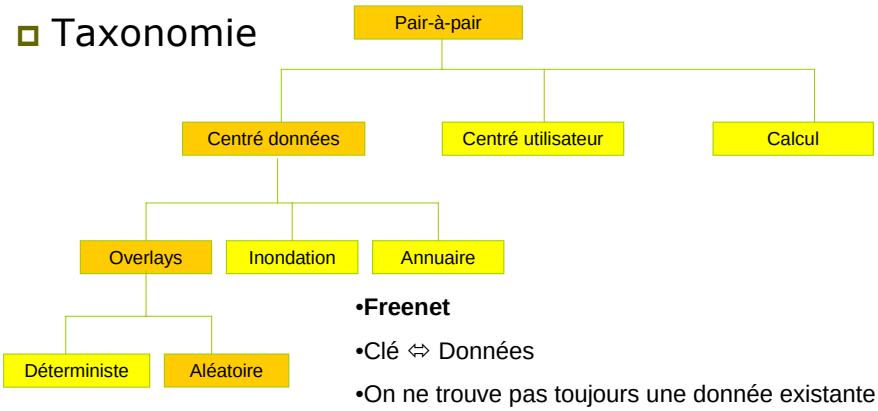
- Taxonomie



- **CAN, Chord, Pastry**
- Clé ⇔ Données
- Les données sont toujours accessibles

III. Systèmes Pair à Pair
1. Principe et composantes

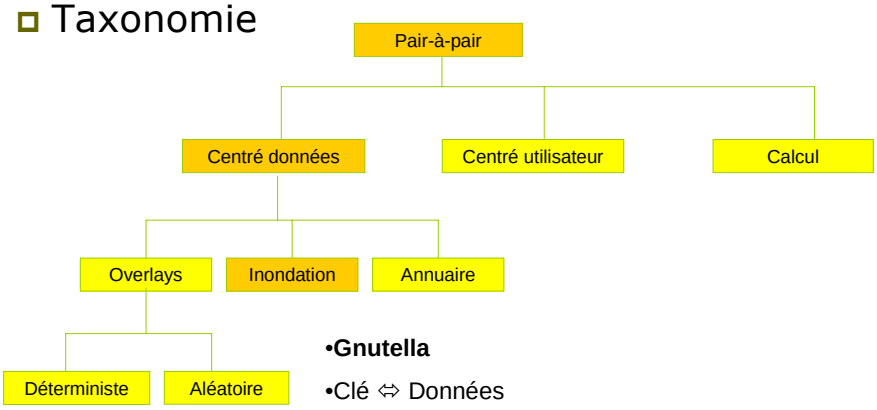
- Taxonomie



- **Freenet**
- Clé ⇔ Données
- On ne trouve pas toujours une donnée existante
- Les données sont répliquées à l'utilisation
- Résiste aux pics de charge

III. Systèmes Pair à Pair
1. Principe et composantes

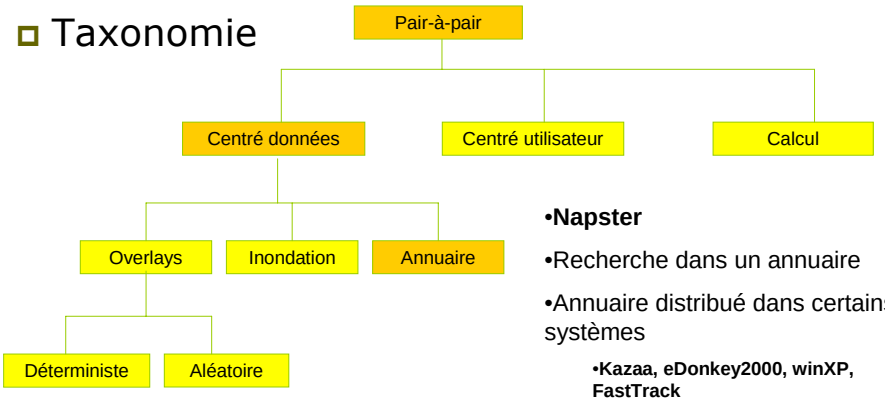
- Taxonomie



- **Gnutella**
- Clé ⇔ Données
- Recherche des données par requêtes aux voisins

III. Systèmes Pair à Pair
1. Principe et composantes

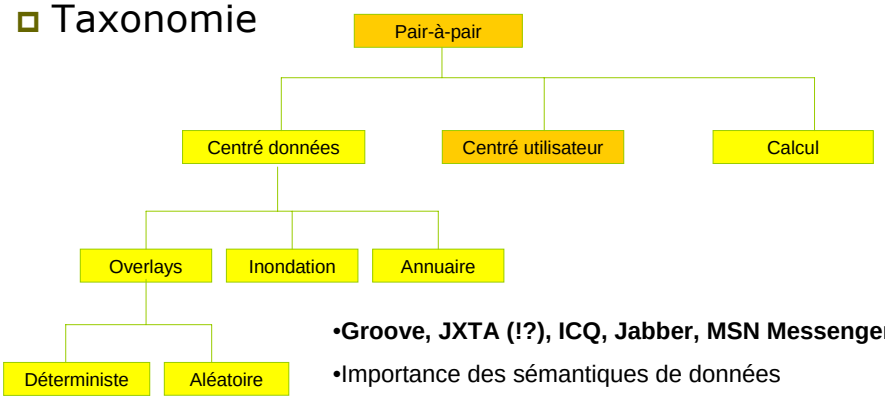
- Taxonomie



- **Napster**
- Recherche dans un annuaire
- Annuaire distribué dans certains systèmes
  - **Kazaa, eDonkey2000, winXP, FastTrack**
- Résistance à la censure
- Super-peers

III. Systèmes Pair à Pair
1. Principe et composantes

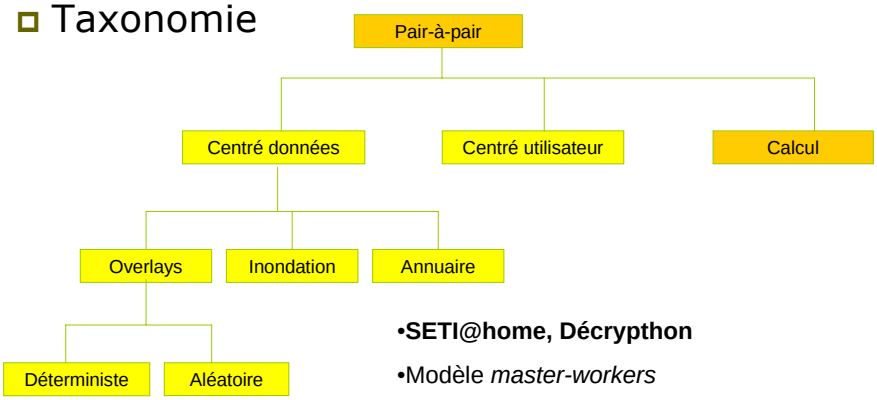
- Taxonomie



- **Groove, JXTA (!?), ICQ, Jabber, MSN Messenger**
- Importance des sémantiques de données
- Informations mutables et/ou « jetables » (voix, SMS,…)

III. Systèmes Pair à Pair
1. Principe et composantes

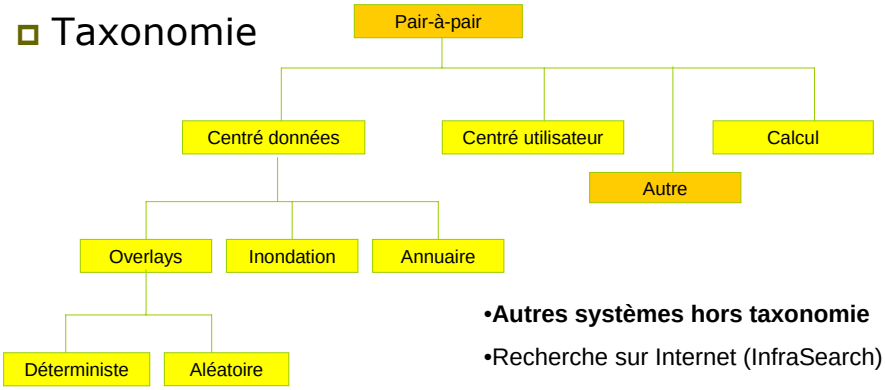
- Taxonomie



- **SETI@home, Décrypthon**
- Modèle *master-workers*
- Gros calcul exécuté par petits morceaux

III. Systèmes Pair à Pair
1. Principe et composantes

- Taxonomie

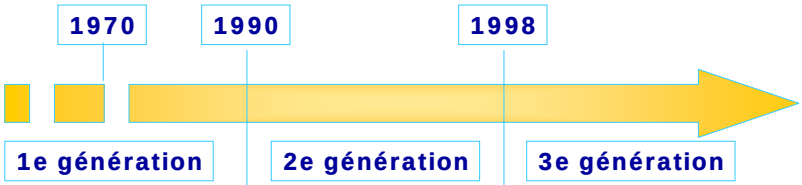


- **Autres systèmes hors taxonomie**
- Recherche sur Internet (InfraSearch)
- Evaluation de performance de sites Web http://www.porivo.com/

Modèle de business viable ?

III. Systèmes Pair à Pair
1. Principe et composantes

- Passé et présent



- Accès direct entre les machines (Arpanet)
  telnet, ftp, gopher, wais
- Les protocoles sont ouverts et *amicaux*

- World Wide Web
  Tim Berners-Lee
- Protocoles HTTP, HTML, Hyperliens de page à page
- Internet devient une toile (Web)

Technologies pair-à-pair
L'information n'est plus liée au réseau .Plus de notion de serveur
Administration complexe et coûteuse. Volatilité des ressources physique

III. Systèmes Pair à Pair
1. Principe et composantes

- Enjeux et l'avenir

Utilisateurs :
Grand intérêt du partage de ressources gratuites sur Internet

Entreprises :
Grande question : qui y trouve l'intérêt ?
-Pas besoin de serveurs -> moins de vente de matériels !
-Difficile à facturer les services !
-Qui paye le développement

Quels seront les modèles économiques ?

III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

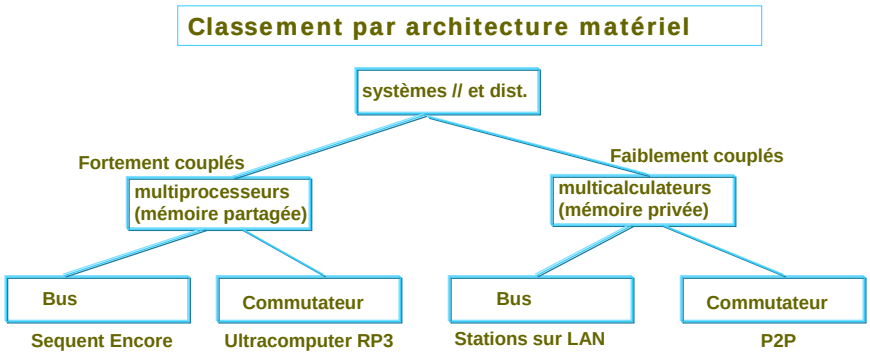
- Introduction au calcul distribué

Taxonomie de Flynn (1972)
- nombre des flux d'instructions
- nombre des flux de données

SISD :	un seul flux d'instruction et un seul flux de données (ordinateurs centralisé)
SIMD :	un seul flux d'instruction et multiples flux de données (machines // vectorielles)
MISD :	multiples flux d'instruction et un seul flux de données (pas de machine réelle)
MIMD :	multiples flux d'instruction et multiples flux de données

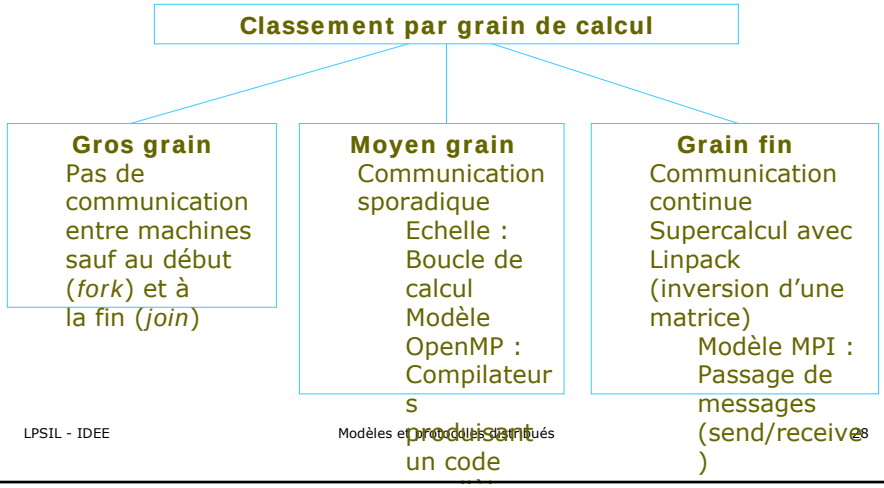
III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

- Introduction au calcul distribué



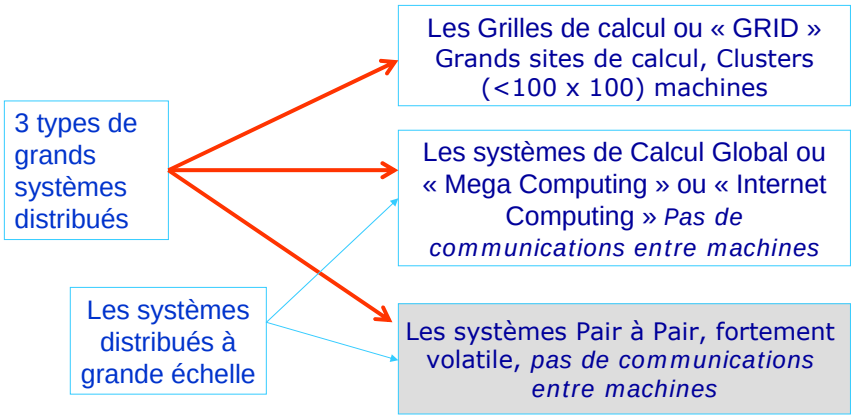
III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

- Introduction au calcul distribué



III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

- Introduction au calcul distribué



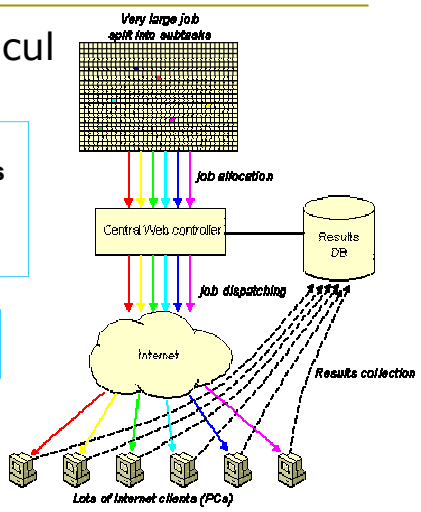
III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

- Principe de grille de calcul (Metacomputing)

Principe
Des milliards de calculs **indépendants** effectués sur les PCs de "volontaires". Offrir ainsi un service telle qu'il est offert par les producteurs d'énergie.

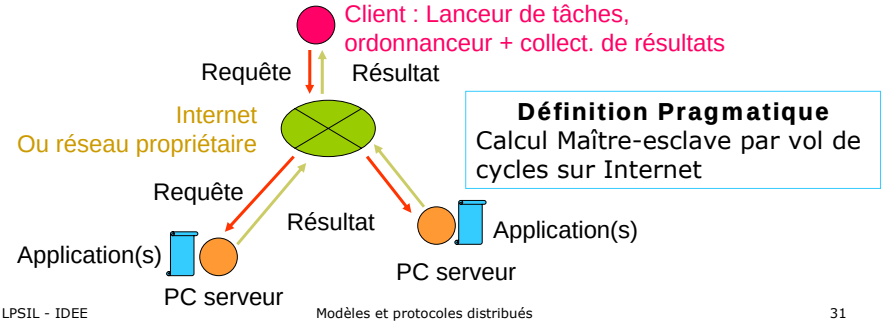
Calcul sur la base du volontariat
Utilisation des ressources inexploitées

Projets en cours : Globus, Harness, DataGRID, Legion, EuroGRID, GénoGRID...



III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

- Principe du calcul global (global computing)
  - Modèle Client-Serveur inversé : 1 client et *n* serveurs
  - L'application exécutée sur les serveurs est fournie par le client
  - Type de services : principalement calcul distribué (SETI@home)

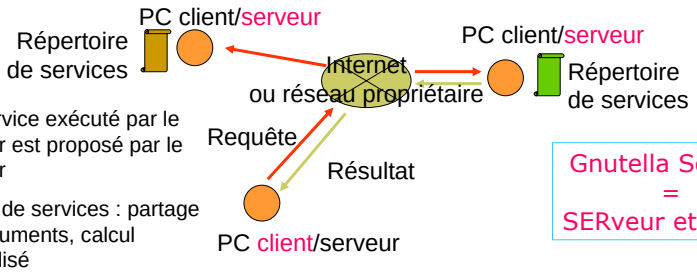


III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

- Principe du calcul global (global computing)
  - Objectif pratique
    - des millions de PC en attente…
    - récupération des cycles processeurs inutilisés (environ 47% en moyenne dans une entreprise*) via un économiseur d'écran)
  - Exemples
    - SETI@home (ce n'est pas du P2P !)
      - Recherche de signaux extra-terrestres
      - 33,79 Teraflop/s (à comparer aux 12,3 Teraflop/s de l'ordinateur le plus puissant au monde au LLNL !)
    - Décrypthon
      - Etablir la carte des 500 000 protéines du vivant
    - RSA-155
      - Casser des codes cryptographiques

III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

- Principe du calcul pair à pair



•Le service exécuté par le serveur est proposé par le serveur

• Type de services : partage de documents, calcul délocalisé

Gnutella Servent = SERveur et cliENT

Pas de consensus autour d'une définition
Un système dans lequel toutes les ressources peuvent agir comme des clients, des serveurs et/ou maintiennent le système lui même

III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

- Principe du calcul pair à pair

XtremWeb : Calcul global Pair à Pair

Systèmes Pair à Pair

Mode d'interaction inter-ressource	Mode de communication	Mode d'organisation système
Toutes les ressources sont à la fois client et serveur	Après la phase de découverte (mise en relation) les nœuds communiquent directement	Système sans serveur centralisé. Système auto-organisé (découverte de ressources,

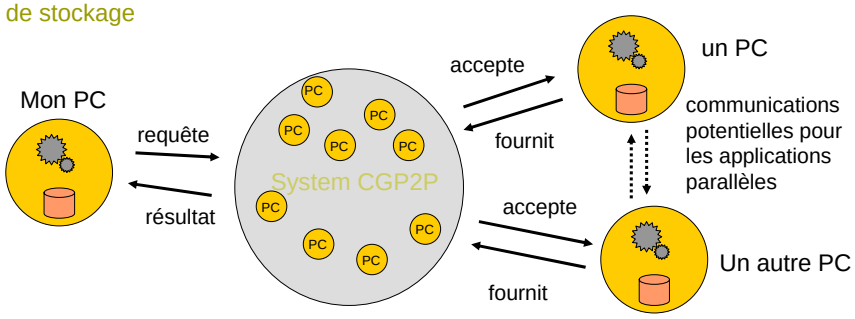
III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

- Principe du calcul pair à pair
  - Seul le gros grain est aujourd'hui possible en P2P.
    - Les **latences réseau** sont trop grandes : le transport est Internet.
    - L'**hétérogénéité des machines** est difficile à gérer: de Pentium 66 aux supercalculateurs multiprocesseurs
    - Les capacités CPU sont très supérieures aux capacités réseau

III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

- Global Calcul Pair à Pair (GCP2P)

Les requêtes correspondent à des demandes de calcul et/ou à des demandes de stockage



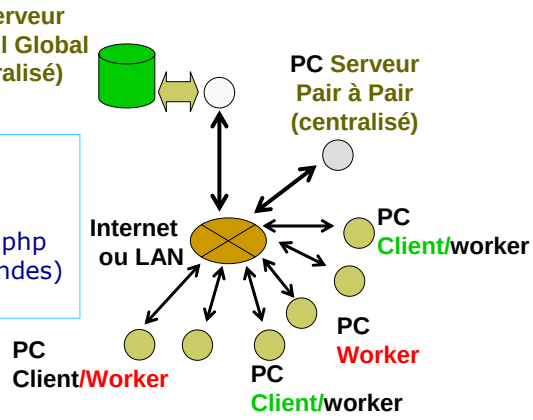
Un environnement de recherche offrant une image système unique à partir de l'agrégation de ressources faiblement couplées

III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

- Global Calcul Pair à Pair (GCP2P)

Projet Xtremweb – open source
LRI, Paris XI

Download :
www.xtremweb.net
Équipement :
RedHat 7.1+mysql+apache+php - RPM (installation en 5 secondes)
serveur + worker + demo



III. Systèmes Pair à Pair
2. Calcul distribué pair à pair

- Perspectives
  - Besoin d'environnements de programmation
    - Aujourd'hui les applications sont figées
  - Amélioration de la sécurité
    - Comment se protéger des fautes
    - Comportements byzantins et malicieux
  - Limité à quelques applications
    - Oui, mais un business assez juteux
    - Intel, AMD utilisent Netbatch
  - I-Cluster
    - Reconnaissance de clusters de machines au repos
  - Un domaine de recherche très actif

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Quelques rappels
  - Modèle client/serveur symétriques et dynamiques : un nœud peut jouer le rôle client ou serveur ou les deux
  - Dimensionnement dynamique et arbitraire
  - Trois types de systèmes :
    - annuaire : échanges indirects et routage indéterministe
    - Inondation : échange direct et le routage indéterministe
    - Overlay : routage déterministe

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Quelques critères d'évaluations
  - Algorithmes distribués déterministes
  - Extensibilité : évolution dynamique
  - Scalabilité : la recherche des informations sur N pairs ne doit pas dépasser $O(\log_2 N)$
  - Mesures d'optimisation :exemple : par réplication de données
  - Mesures de protections de données
  - Modèle de cohérence de données
  - Etc.

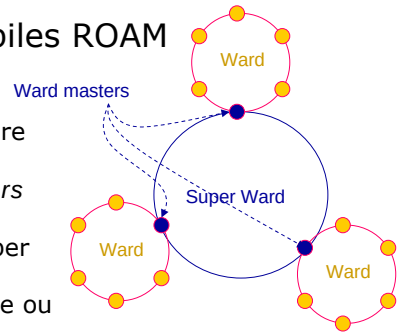
III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Précurseurs du P2P
  - Systèmes de fichiers répliquant :
    - on réplique la totalité des fichiers sur un ensemble de serveurs interconnectés (Scalabilité O(N))
    - Un client peut connecter à importe quel serveur pour accéder (read) à un fichier. Il peut librement modifier le fichier (write)
    - Le système dispose d'un mécanisme de détection des conflits des mises à jours (une requête de vérification des contraintes d'intégrité du système avant un write)
    - Si un conflit est détecter, le système dispose d'un mécanisme de de résolution globale (merging) qui consiste à ordonner les « writes » selon leur estampille et effectuer les mises à jour des « writes » valides dans cet ordre sur tous les serveur
  - Systèmes : CODA (Satyanarayanan, CMU, 1990), Ficus (Popek, UCLA, 1990), Bayou (Petersen, Xerox, 1995), "automatic hoarding" (Popek, UCLA, 1997)
  - Utilisation dans : Oracle, Lotus

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Précurseurs du P2P
  - Systèmes de fichiers mobiles ROAM
    D.H. Ratner (UCLA 1998)

-Système adapté au topologie cellulaire de téléphonie mobile : *Ward model*
-Election automatique de *Ward Masters* qui sont Maîtres de l'anneau local
- Les ward masters participent au super anneau
- Propriétés de localité : Géographique ou topologique
Scalabilité : $\log_2 N$



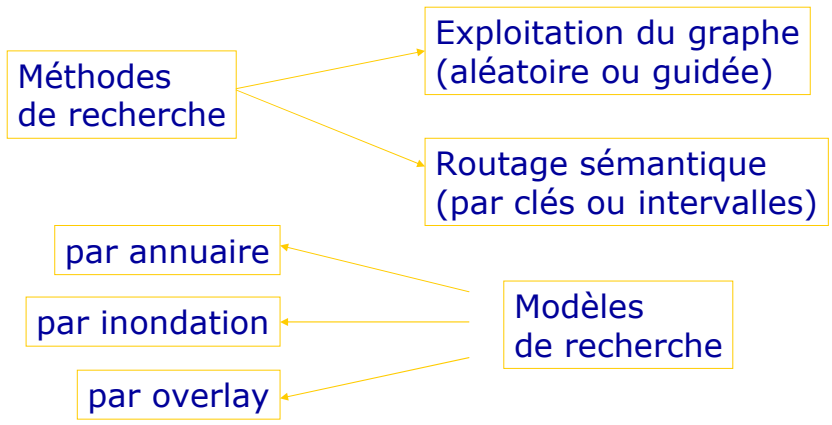
Projet est en développement

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Systèmes P2P récents : totalement distribués
  - Deux domaines d'étude
    - Topologie : quelle est la structure du système ?
    - Routage : Comment circulent les informations dans le système ?
  - Plusieurs objectifs :
    - Minimiser l'utilisation des liens du réseau (ce qui peuvent être traités localement doivent être traités localement !)
    - Minimiser le temps de recherche (réduire le diamètre du système)
    - Minimiser la complexité de reconstruction de la topologie de réseau en présence de volatilité (réduire le temps de réorganisation du système)
    - Préserver les propriétés du routage en présence de volatilité (résistance à la présence aléatoire des nœuds)
    - Préserver la cohérence du système et la sécurité de données (cohérence de données répliquées, résistance aux attaques)

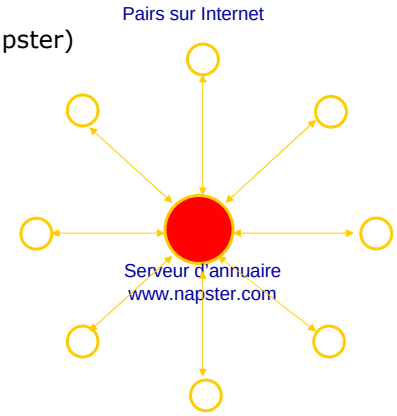
III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Systèmes P2P récents :recherche P2P



III. Systèmes Pair à Pair
3. Échange de données pair à pair

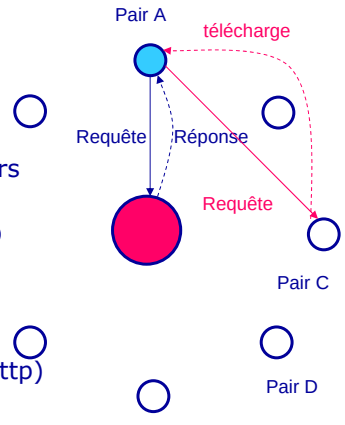
- Modèle par annuaire :
  - Annuaire centralisé (exemple Napster)
    - Un seul serveur central
      - Annuaire
      - NomDeFichier <-> Pair
    - Liens statiques pairs-serveur
      - Annonce de présence
      - Ressources locales
      - Capacités de communication
      - Charge en cours
    - Communication continue
      - Heartbeat
  - Annuaire distribués :
    ex. OceanStore
  - Autres systèmes :
    Kazaa, eDonkey2000, WinXP



III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par annuaire :
  - Routage de requête dans Napster

Le Pair A veut un fichier
1/ A envoie une requête au serveur central sur lequel dispose une liste complète des pairs
ex. « Search La-mer.mp3 »
2/Le serveur répond à A
En donnant l'adresse C et D qui sont des machines actives ayant cette ressource
3/ A établit 1 connexion directe à C
« Get La-mer.mp3 »
avec un protocole de *download* direct (port http)
4/ Remarques
Gestion d'erreurs triviale, pas de gestion des reprises, pas de *hammering*, une seule source

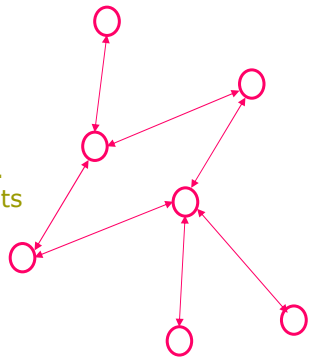


III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par inondation :
  - Modèle isotropique
    - Les pairs sont libres et égaux en dignité et en droit
    - Le système est totalement volatile : tout les pairs sont libres à connecter ou déconnecter
  - Chacun dispose de données *locales*
    - Aucune notion de réplication
    - Chaque recherche repart de zéro
  - Diffusion en vague d'une requête
    - Pas de scalabilité : temps $O(N^2)$
    - Pas d'optimisation

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par inondation :
  - Exemple Glutella (2000)



1/ Liste de pairs est locale à chacun contenant quelques centaines « voisins ». Elle est évolutive en fonction d'événements divers

2/ Sessions permanentes sur 5 à 9 pairs avec un routage des messages continuel

3/ Messages de base :

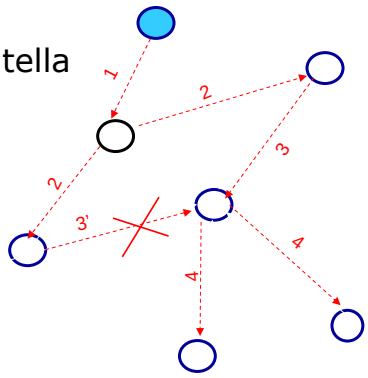
- Ping : Annonce de présence
- Pong : Réponse au ping avec IP/port du réceptionnaire attachée, File padding
- Query : Requête de recherche - Vitesse minimum du répondant
- QueryHits : Réponse au message Query. Nombre de fichiers en réponse avec leurs index

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par inondation :
  - Exemple du routage de Glutella

Query

- Requête en vague (dite inondation) vers tous les voisins
- Chacun fait une recherche locale…
…et relaie la requête à tous ses voisins
- Les messages déjà reçus sont ignorés pour éviter les cycles
Chaque message a un Time-To-Live

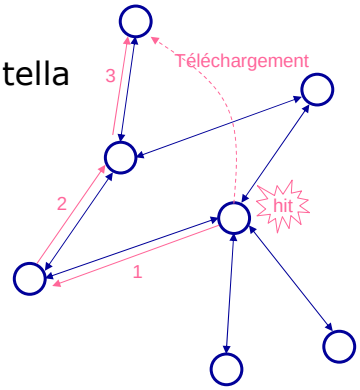


III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par inondation :
  - Exemple du routage de Glutella

QueryHit

1/ Recherche locale par chaque pair lors du Query
2/ Relais inverse des réponses positives avec Adresse / Descripteur de fichier. L'aggrégation des retours avec *Padding* ou *stuffing*
3/ L'émetteur original choisit un fichier (et le pair associé)
4/ Le téléchargement se fait en direct



III. Systèmes Pair à Pair
3. Échange de données pair à pair

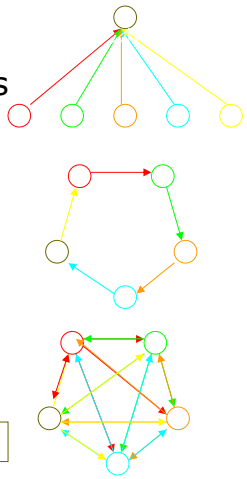
- Modèle par inondation :
  - Problèmes liés au modèle
    - Inefficace à une large d'échelle : Au passage des 20000 utilisateurs, le système GLUTELLA s'écroule (une observation en été 2000 - « *Why Gnutella Can't Scale. No, Really.* » [Ritter2000]
    - Le routage est non optimal : $O(N^2)$
  - Nécessité de modéliser/simuler/émuler
  - Solution de Hubs BearShare
    - SuperPeers permettant d'étendre le système
    - Mais pas en O(log(N)) !

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay :
  - Principe des overlays :
    - Soit une communauté représentée par un graphe d'interconnexion
    - Chacun nœud membre (pair) est à la racine d'un arbre de profondeur 1, d'arité variable. Cet arbre est un plongement du graphe total du réseau
    - Chaque pair possède un ID (identifiant) unique
    - Un algorithme local permet le routage
      - de proche en proche
      - vers un des nœuds de l'arbre local
      - global vers un ID quelconque de l'espace des ID
    - Aussi appelé *Distributed Hash Tables* (DHT)
    - La mode : Les *Plaxton Meshes* (Pastry)
    - Ancêtre : Routage par hypercubes

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay :
  - systèmes P2P par overlay simples
    - Modèle client serveur
      - L'arbre local est limité à la connaissance du serveur
      - L'arbre du serveur est complet
    - Anneau simple
      - L'arbre est limite au nœud suivant
    - Graphe complet
      - L'arbre local est le graphe complet



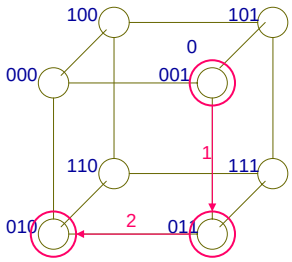
Ces systèmes ne sont que des cas d'école

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay :
  - Routage par hypercube
    - Utilisé en architectures parallèles (années 1980-90)
    - Cube virtuel en dimension N
    - 2^N participants (hypercube complet)
    - Chaque nœud a un ID sur N bits
    - Chaque nœud a N voisins directs
    - Routage très simple : de X = {x0 x1 x2} vers Y = {y0 y1 y2} :
    - Pour i = 0 à N-1
      - Si xi != yi
        - router x0..xi..xN-1 vers x0..xj..xN-1
        - Sortie de la boucle
      - FinSi
    - FinPour

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay :
  - Routage par hypercube : exemple 3-cube
    - Routage de proche en proche
      - Exemple : De 001 a 010
        - 001 -> 011
        - 011 -> 010
    - Algorithmique simple et efficace
      - 65536 nœuds en dimension 16
    - Inconvénients
      - Chacun ***doit*** être présent
      - Pas de volatilité des nœuds possible



III. Systèmes Pair à Pair
3. Échange de données pair à pair

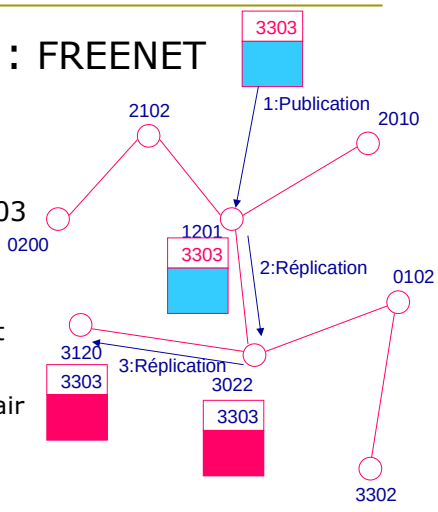
- Modèle par overlay :
  - Notion d'UUID (Universally Unique Identifier)
    - Identifiant de 128 bits (parfois 160)
    - Statistiquement unique
      - Probabilité de 2^{64} pour tomber sur deux UUID égaux
    - Généré à partir de divers composants
      - Sources uniques
        - Adresse IP
        - Hachage du contenu d'un fichier
        - Nom du fichier
      - Sources dynamiques
        - Estampille temporelle
      - Sources aléatoires
        - Tirage au sort
    - Secure Hash Algorithm (SHA-1)
    - Fonction non inversible en un temps non polynomial
    - Utilisation pour l'identification des pairs et de sources de données

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : FREENET
  - Un système d'échange de fichiers P2P
    - Algorithmique très simple, voire canonique
    - Décentralisation complète
    - Isotropique
  - Anonymat
    - Pour la publication d'informations
    - Pour la récupération des informations
    - Résistance à la censure
  - Open source
    - Beaucoup de sous projets, contributions, analyses, recherches
  - Résistance aux Hot spots (*slashdot effect*)
    - La réplication est basée sur la popularité des fichiers
    - Plus un fichier est récupéré, plus il est répliqué

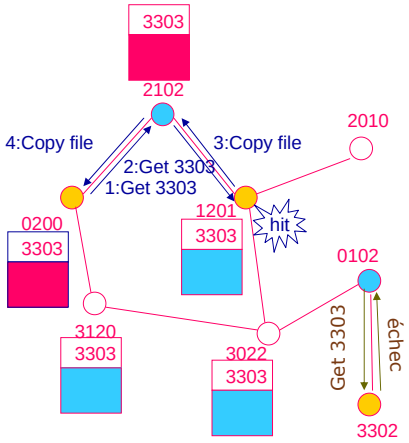
III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : FREENET
  - Publication d'un fichier
- Chaque pair a un ID
- Le Fichier est publié sur 1201
- L'ID du fichier est calculé : 3303
  - Hachage du nom de fichier
- Processus récursif :
  - Le fichier est caché localement
  - Si ∃ pair avec ID plus proche lexicalement de 3303 que l'ID local alors réplication sur ce pair



III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : FREENET
  - Routage de requête
- Demande du fichier d'ID 3303 depuis 0200
- Processus récursif :
  - Si ∃ pair voisin avec ID plus proche lexicalement de 3303 que l'ID local alors requête transmise vers ce pair
  - Si le fichier est trouvé il est copié vers l'appelant
- Risque de non trouver
  - Ex. depuis noeud 0102 (backtracking : repart d'un 2e voisin avec un TTL (Time To Live))



III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : FREENET
  - Gestion d'extension :
    - L'ajout d'un nœud dans Freenet se fait par connaissance d'un *seed*
    - C'est un nœud déjà membre d'une communauté Freenet
    - Il va « transférer » sa liste de voisins au nouveau
    - Cette liste évoluera ensuite aléatoirement
  - La destruction des nœuds est automatique
    - Les nœuds les plus anciens ne sont plus référencés

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : FREENET
  - Quelques problèmes
    - Nécessite de connaître les clés
    - *Garbage collection automatique*
      ⇒Disparition des vieux fichiers
    - On ne trouve pas toujours un fichier existant
    - Charge importante (réplication systématique)
      - En bande passante réseau
      - En stockage local (cache des fichiers)

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : Chord
  - Description générale
    - Lab of Computer science, MIT (2001-2004)
    - Ressources (Key, Value)
    - Système complètement distribué
    - Ne garantit pas l'anonymat mais le temps de recherche est prévisible
    - Coût de communication et de gestion en $\log_2(N)$
    - Equilibrage de charge
      - Fonction de hachage distribuée équitablement
    - Changement dynamique des tables
      - Pour le support de la volatilité des nœuds
    - L'espace de clé est plat
      - Pas de hiérarchie

III. Systèmes Pair à Pair
3. Échange de données pair à pair

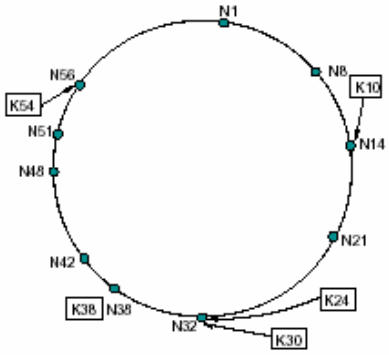
- Modèle par overlay : Chord
  - Distribution de clés sur les nœuds
    - Les nœuds (paires) identifiés sont ordonnés sur cercle modulo de 2^m.
    - Une clé k sera assignée au *premier nœud* ayant identifiant égal supérieur à k. Ce nœud s'appelle *successeur* de la clé k et dénoté successeur(k).
    - Si les nœuds sont représentés comme un cercle de $0..2^m-1$, alors le successeur(k) sera le premier nœud à partir de k dans le sens d'aiguille de la montre
    - Un tel cercle ayant des nœuds identifiés et attribuées des clés est appelé « anneau de Chord » (chord ring)

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : Chord
  - Distribution de clés sur les nœuds

Exemple

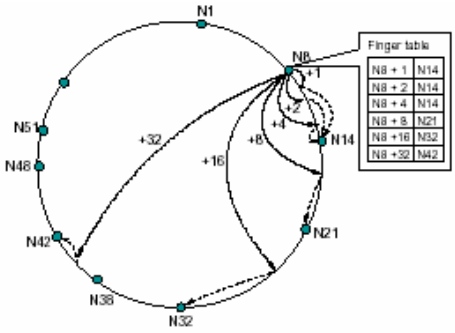
Soit un Chord ring avec m = 6, sur lequel il y a 10 nœuds identifiés et 5 clés.
Voici la distribution de clés :
- Le successeur de la clé 10 est 14 car la clé 10 est stockée dans le nœud 14.
- De la manière similaire, la clé 24 et 30 ont leur successeur noeud 32, la 38 nœud 38, et la clé 54 a le successeur le nœud 56.



III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : Chord
  - Recherche d'une clé depuis d'un noeud

- Soit m bits de l'identifiant.
- Chaque nœud n (mod 2^m) possède une « finger table » comprenant m entrée dont l'entrée i ($1 \le i \le m$]) contient l'adresse du premier nœud (actif) qui peut recevoir la clé [$(n+2^{i-1})$ mod 2^m]. On dénote ce nœud par n.finger[i]
- le successeur du nœud n (ne pas confondre à la succession de clé) est n.finger[1] (le nœud identifié suivant)

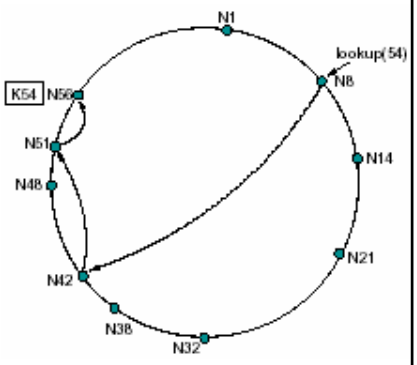


III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : Chord
  - Recherche d'une clé depuis d'un noeud

Exemple

- Nœud N8 lance la requête de recherche de la clé K54
- En examinant sa « finger table », il trouve le nœud N42 qui est plus proche à K54. Il se diriger sa requête à ce nœud.
- Le nœud N42 trouve à son tour, le nœud N51 qui est le plus proche (et inférieur) à K54. Il l'envoie la requête au nœud N51.
- Le nœud N51 sait que son successeur N56 doit être le successeur de la clé K54 (par définition. Il envoie l'adresse de N56 à N8.

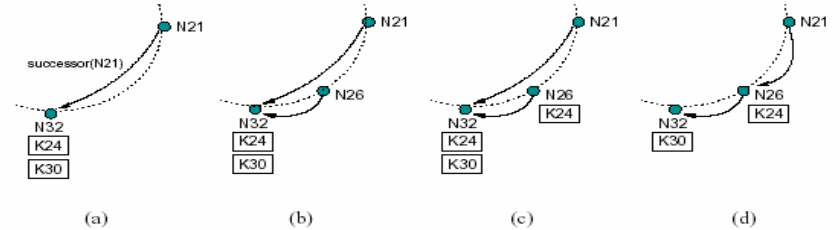


III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : Chord
  - Gestion de l'extension
    - Ajout d'un nouveau nœud (pair)
      - Construire une « finger-table » en contactant un nœud n' déjà connu (n'.find.successor(n))
      - Réactualiser les finger-tables des nœuds
      - Transférer les clés dont le nouveau nœud est responsable
    - Retrait d'un nœud
      - Tous les nœuds font une mise à jours de leur finger-table en remplaçant le nœud en retrait par son successeur
      - Possibilité de rupture de système quand tous les successeurs d'un nœud prennent simultanément leur retrait (exemple : N14, N21 et N32 sont à retraite, N8 ne peut savoir que N38 est son nouveau successeur)

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : Chord
  - Redistribution de clés



N26 se rejoigne l'anneau entre N21 et N32 : a/ étape initial N21 pointe sur N32; b/ N26 cherche sont successeur et point sur 32; c/ N26 copie tous les clés inférieures ou égales à 26 depuis N32; d/ La procédure de stabilisation fait la mise à jour le successeur de N21 à N26

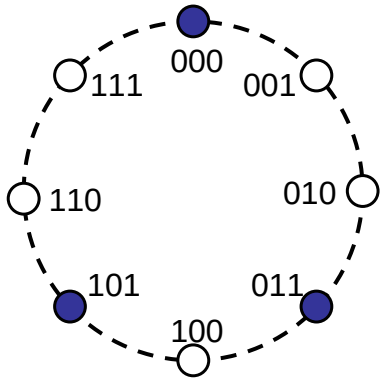
III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : Pastry
  - Description générale
    - Microsoft Research, University of Rice, Cambridge
    - « Scalable decentralized object location and routing »
    - Table de hachage distribuée
    - Système complètement distribué
    - Propriétés de localité dans le routage
    - Applications basées sur PASTRY
      - PAST (Persistent Storage utility)
      - SCRIBE (event notification infrastructure)
      - Squirrel (Decentralized web cache)

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : Pastry
  - Distribution de clés

1/ Chaque nœud possède un NodeID, un identifiant unique et aléatoire
2/ Espace de nœuds circulaire 2^{128} nœuds
3/ Assignation uniforme, des numéro adjacents concernent des nœuds très différents



III. Systèmes Pair à Pair
3. Échange de données pair à pair

- **Modèle par overlay : Pastry**

- Routage d'une requête
  - Vers le nœud qui possède le NodeID le plus proche de la clé
  - Nœuds vivants seulement
  - Routage par proximité de clé
  - les clés et NodeIDs sont des séquences de chiffres de base 2^b
- Dans la table chaque NodeID est associé à une adresse IP
- Exemple
  - b = 2
  - Nœuds connus de 10233102
  - Table de routage
    - Entrées à la ligne n partagent n premiers chiffres avec le NodeID
    - *($log2^b$ n)* lignes
    - *(2^b –1)* colonnes
- Délai de routage $O(log2^b n)$

NodeID 10233102

Smaller		Larger	
10233033	10233021	10233120	10233122
10233001	10233000	10233230	10233232

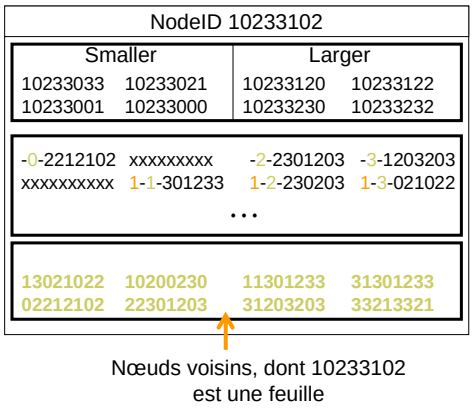
-0-2212102	1	-2-2301203	-3-1203203
0	1-1-301233	1-2-230203	1-3-021022
10-0-31203	10-1-32102	2	10-3-23302
102-0-0230	102-1-1302	102-2-2302	3
1023-0-322	1023-1-000	1023-2-121	3
10233-0-01	1	10233-2-32	xxxxxxxxxx
xxxxxxxxxx	xxxxxxxxx	102331-2-0	0

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- **Modèle par overlay : Pastry**
  - Ajout d'un nœud
    - Un nouveau nœud de NodeID X entre dans Pastry
      1. Il contacte un nœud A
      2. X demande à A de router le message *Join(X)*
      3. Pastry route le message jusqu'au nœud Z
      4. Les nœuds traversés par la requête envoient leurs tables à X
      5. X initialise sa table à partir de ces informations
      6. X informe les autres nœuds de sa présence
    - Puisque Z est proche numériquement de X, X et Z partagent les même feuilles.

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : Pastry
  - Retrait d'un nœud
- Heartbeat entre voisins
  - Messages périodiques
  - Timeout sur inactivité
  - Détection des disparitions
- Un nœud peut mourir
  - Ses voisins le détectent
  - Ils collectent les informations des nœuds proches
  - Ils remettent leur table à jour



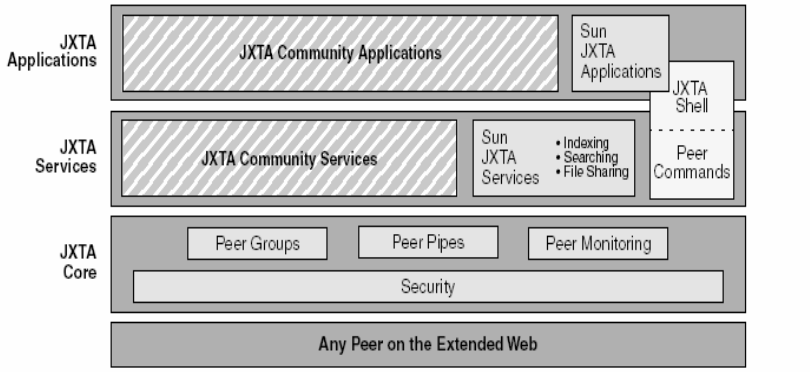
Nœuds voisins, dont 10233102 est une feuille

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : JXTA
  - Description générale
    - Développement par Sun basé sur Java
      - Succès mitigé
    - 4 composants principaux
      - Peer pipes
        - Connexion entre pairs
        - Partage d'informations sur le réseau et de manière distribuée
      - Peer groups
        - Groupes virtuels
        - Dynamiques
        - Cohérents
      - Peer Monitoring
        - Surveillance et mesure des interactions
        - Politiques de contrôle entre pairs
      - Sécurité
        - Mécanismes d'identification, de contrôle d'accès et de confidentialité

III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Modèle par overlay : JXTA
  - Composants



III. Systèmes Pair à Pair
3. Échange de données pair à pair

- Perspectives
  - Mutabilité
    - Mise à jour des données
    - Écrivains multiples
    - Cohérence
  - Topologie réseau
    - Connaissance de la localisation des machines sur le réseau
    - Minimalisation de la consommation réseau
    - Minimalisation des temps de récupération des données
  - Sécurité
    - Authentification
    - Contrôle d'accès
  - Observation
    - Simulation ou émulation
    - Visualisation

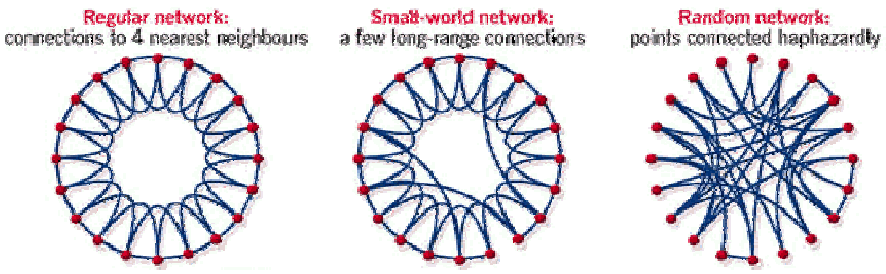
III. Systèmes Pair à Pair
4. Évolution , problématique

- Small World
  - Deux propriétés essentielles
    - Regroupement maximal
      - Mes voisins se connaissent entre eux
    - Chemin minimal
      - La distance entre deux nœuds du graphe est faible
      - Dans l'idéal, pour un graphe G=(E,V), $\max_{i,j \in V}$ (min(Path(i,j)) est le plus petit possible
  - Ces propriétés permettent de mettre en œuvre des *hubs* d'information
    - Des nœuds très communicants, connaissant beaucoup de voisins lointains
    - Ce sont les autoroutes (ou les avions) utiles pour aller loin
  - Permet de rester « *scalable* », en O(log(N))
  - Les systèmes P2P doivent (devraient) essayer de faire converger ces paramètres
    - Ce n'est pas toujours le cas

III. Systèmes Pair à Pair
4. Évolution , problématique

- Small World
- Stanley Millgram 1967
  - Diamètre des américains : 6
  - graphe des accointances très connexe
- Lois de distributions

Université notre-Dame
Diamètre du WWW : 19



III. Systèmes Pair à Pair
4. Évolution , problématique

- Lois distribuées
  - Zipf law
  - Extended Zipf
  - Power law
  - Pareto
  - Exponential
  - Lognormal
- Ces lois caractérisent une population

III. Systèmes Pair à Pair
4. Évolution , problématique

- Sécurité
  - Comment construire une chaîne de trust ?
  - Nécessitent des *certificate authorities*
    - *E.g. Kerberos*
  - Pas possible d'avoir un serveur central en P2P
  - Comment répudier un membre du groupe ?
    - Consensus distribué
  - Comment éviter le « déni de service » (DoS attacks) ?
  - Des recherches en cours
    - Mais assez peu
    - Systèmes de votes statistiques, à la PGP
    - Ce sont des domaines *difficiles*

III. Systèmes Pair à Pair
4. Évolution, problématique

- Réputation :
  - Comment juger de la qualité d'un pair ?
    - Volume de ressources offertes
    - Qualité des ressources
    - Détection des *Freeloaders*
  - Problème classique eBay (*auction sites*)
    - Il est facile de tricher pour augmenter sa réputation
  - Pas de réponse satisfaisante coté recherche

III. Systèmes Pair à Pair
4. Évolution, problématique

- Comptabilité
  - Comment faire payer au service ?
    - Paiement par fichier downloadé
    - Par unité de calcul
  - Difficile sans serveur central
    - Dons pas adapté au P2P isotrope
  - Encore une fois, pas de bonne réponse

III. Systèmes Pair à Pair
4. Évolution, problématique

- Mutabilité et cohérence
  - Modification des données
    - Mutabilité
    - Cohérence
  - Résilience
    - Checkpointing
    - Tolérance aux pannes
  - Beaucoup de travaux en cours

III. Systèmes Pair à Pair
4. Évolution, problématique

- Évolution
  - Des systèmes très intéressants
    - Techniquement
    - Pour l'utilisateur
      - Plein de propriétés séduisantes
  - Réponse à des problèmes très communs
  - Un *business model* à trouver
    - L'avenir répondra

Bibliographie

- Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan, Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications. To Appear in IEEE/ACM Transactions on Networking
- Franck Cappello « Calcul Global Pair-à-Pair. Journée ACI GRID, 22 novembre 2001
- Bruno Richard, Technologie Pair à Pair, Laboratoire HP de Grenobre
- Roland Balter, Modèles de structuration des applications réparties
- A. Birrell, B. Nelson, "Implementing Remote Procedure Calls", ACM Trans. on Computer Systems, Vol. 2, 1984
- J-M. Busca, F. Picconi, P. Sens, PASTIS : Système de fichiers P2P multi-écrivain, IRISA, juin 2004
- Michel Riveill, Modèles Client Serveur, INP Grenoble / ENSIMAG, 1999
- S. Krakowiak, Systèmes répartis : état de l'art, IMAG, Grenoble