Gestion de données à large échelle

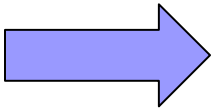
Anne Doucet
LIP6
Université Paris 6

Plan

- Contexte
- Les réseaux P2P
  - Non structurés
  - Structurés
  - Hybrides
- Localisation efficace et Interrogation complète et exacte des données.
  - Organisation des P2P en groupes
  - Interrogation et gestion

Contexte

- Les données sont complexes:
  - Format (ex. multimédia)
  - Structure (ex. documents)
  - Contenu (ex. incomplet, imprécis)
  - Taille
  - Sémantique associée (ex. métadonnées)
  - Code associé
- Et distribuées à large échelle



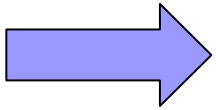
Les SGBD ont trouvé leurs limites

Systèmes Pair-à-pair

- Réseaux dont les nœuds sont à la fois client et serveur
- Propriétés :
  - Pas de coordination centralisée
  - Aucun nœud n'a une vision globale du système
  - Le comportement global émerge à partir des interactions locales
  - Tous les services et données sont accessibles de n'importe quel nœud
  - Les nœuds sont autonomes
  - Les nœuds et les connexions ne sont pas fiables

Systèmes Pair-à-pair

- Avantages :
  - Souplesse d'utilisation
  - Auto-organisation
  - Équilibrage de charge
  - Tolérance aux fautes
  - Disponibilité
  - Faible coût
  - Pas de limitation en nombre d'utilisateurs, de serveurs



Permet le partage de données à large échelle

Gestion de données à large échelle

- Problèmes ouverts :
  - Découverte et interrogation des ressources
  - Sécurité
  - Performances
  - Qualité de service
  - Gestion des mises à jour

Recherche et interrogation

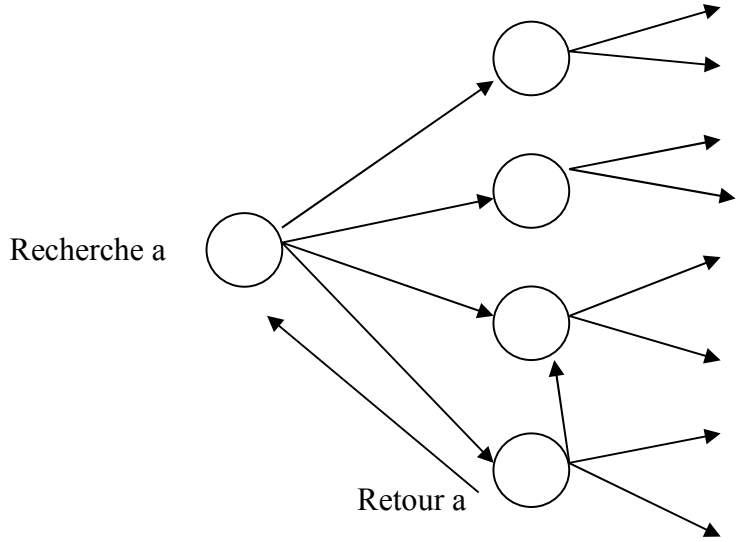
- Expressivité du langage
  - Recherche par clé
  - Recherche par mots-clés
  - Classement des résultats par importance
  - Requêtes sur intervalles
  - Requêtes avec agrégats
  - Requêtes SQL (pb. de schéma)
- Efficacité
- Autonomie
  - choix des ressources à stocker, choix des nœuds successeurs
- Robustesse
  - tolérance aux pannes, résistance aux attaques
- Qualité des réponses
  - Une réponse, toutes les réponses, uniquement les bonnes réponses, les k meilleures

Classes de systèmes P2P

- P2P non structurés (ex: Freenet, Gnutella)
  - Pas d'index global, pas de coordination centrale, le comportement global vient des interactions locales, etc.
- P2P structurés (ex: CHORD, CAN, P-GRID)
  - Répartition des ressources sur les différents nœuds
  - Utilisation de tables de hachage distribuées
- P2P hybrides
  - super peers (ex: Kazaa)
    - Mélange de client/serveur et P2P
  - P2P sémantiques (ex: SON, Routing Indices)
    - P2P « pur » avec routage basé sur une information sémantique

Gnutella

- Chaque nœud propage la requête à ses voisins (en général 4)
- Le nombre de propagations est limité (en général à 7)
- Détection de cycles grâce à l'identificateur des paquets

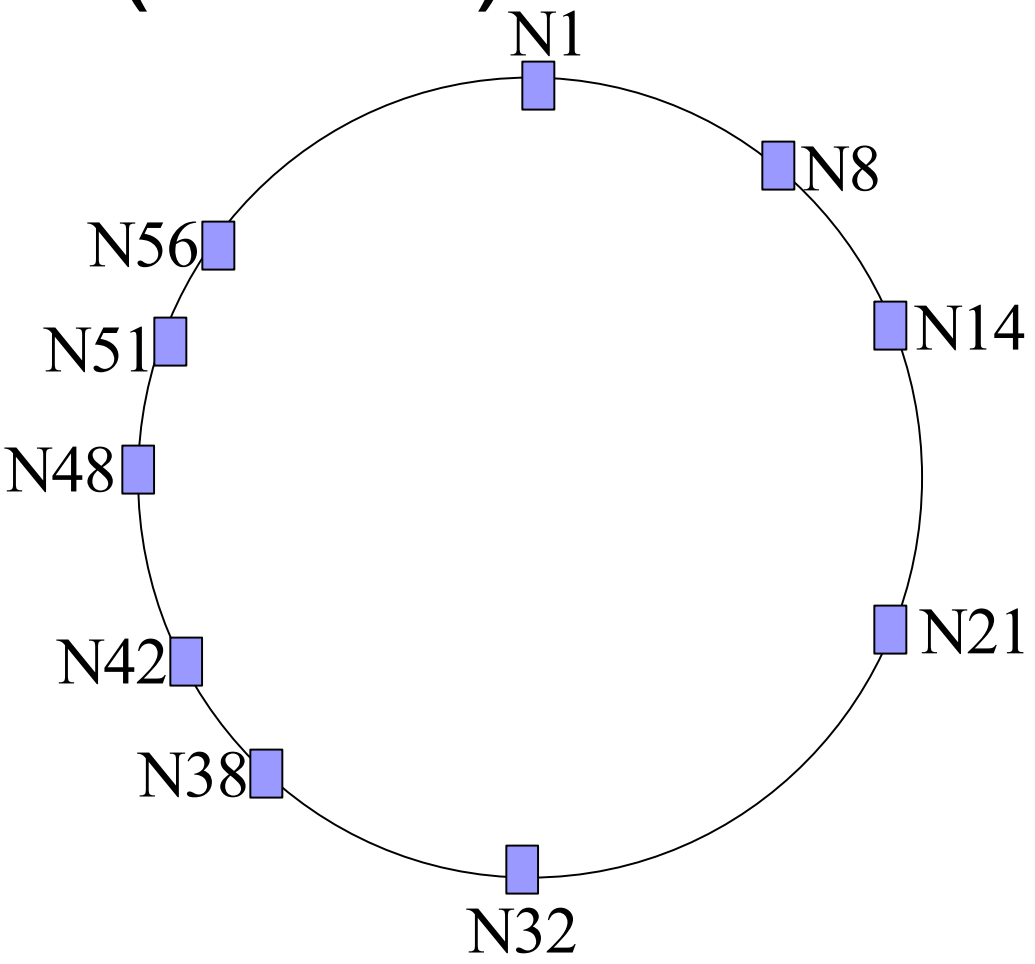


Bilan de Gnutella

- Complètement décentralisé
- Très tolérant aux fautes
- S'adapte bien à la dynamique du réseau
- Gros consommateur de bande passante
- Pas de garantie de succès, ni d'estimation de la durée des requêtes
- Pas de sécurité, ni de réputation (pas de notion de qualité des pairs ni des données fournies)
- Simple, robuste

P2P structuré (Chord)

Chaque nœud est alloué sur l'anneau à l'aide d'une fonction de hachage Hash(IP)



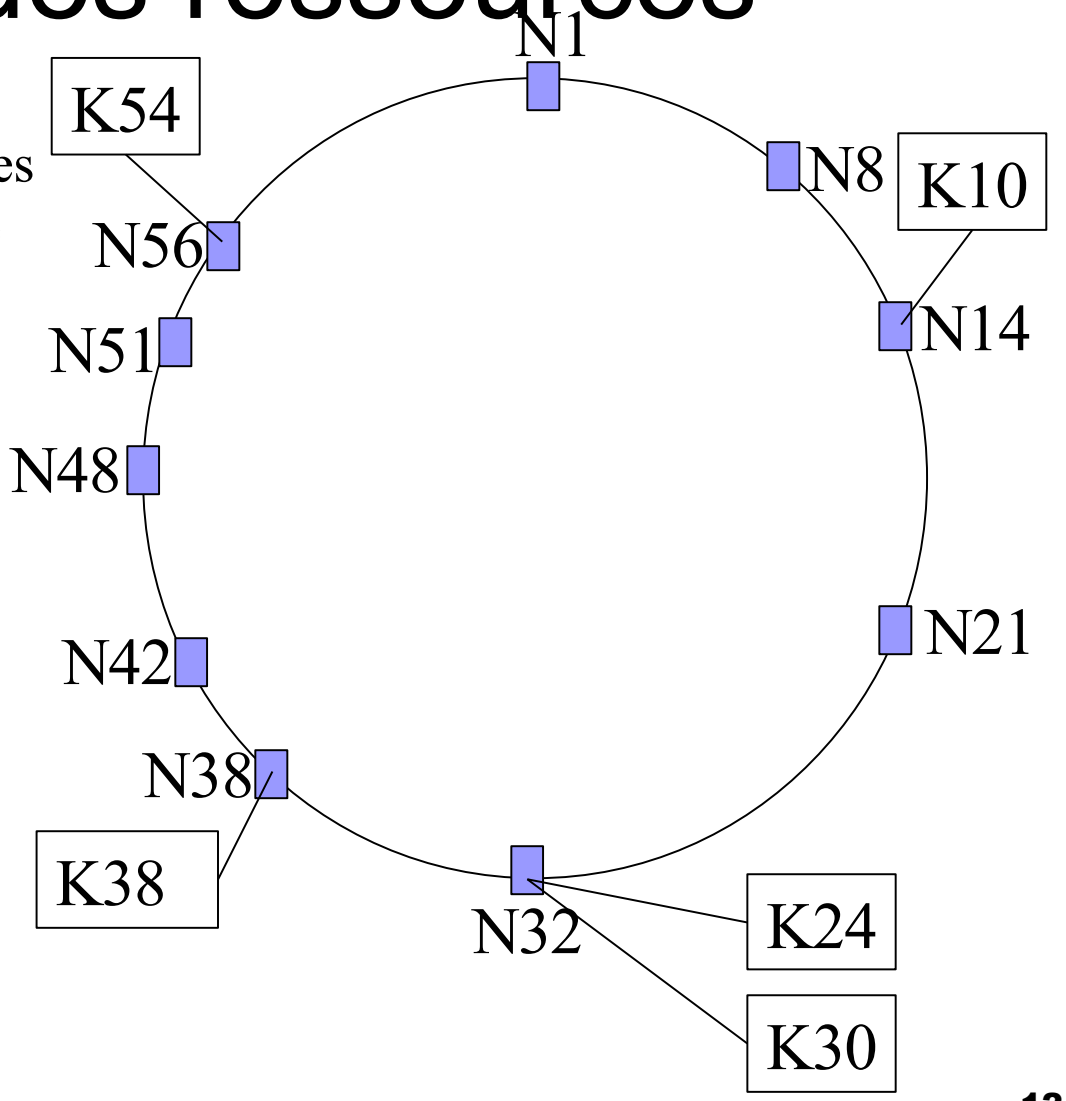
Au plus 2^m nœuds

Placement des ressources

Les ressources sont distribuées uniformément sur l'ensemble des nœuds

Hash(ressource)=k

k placé sur successeur(k)
successeur(k)=nœud immédiatement supérieur (ou égal) à k



Exemple de recherche

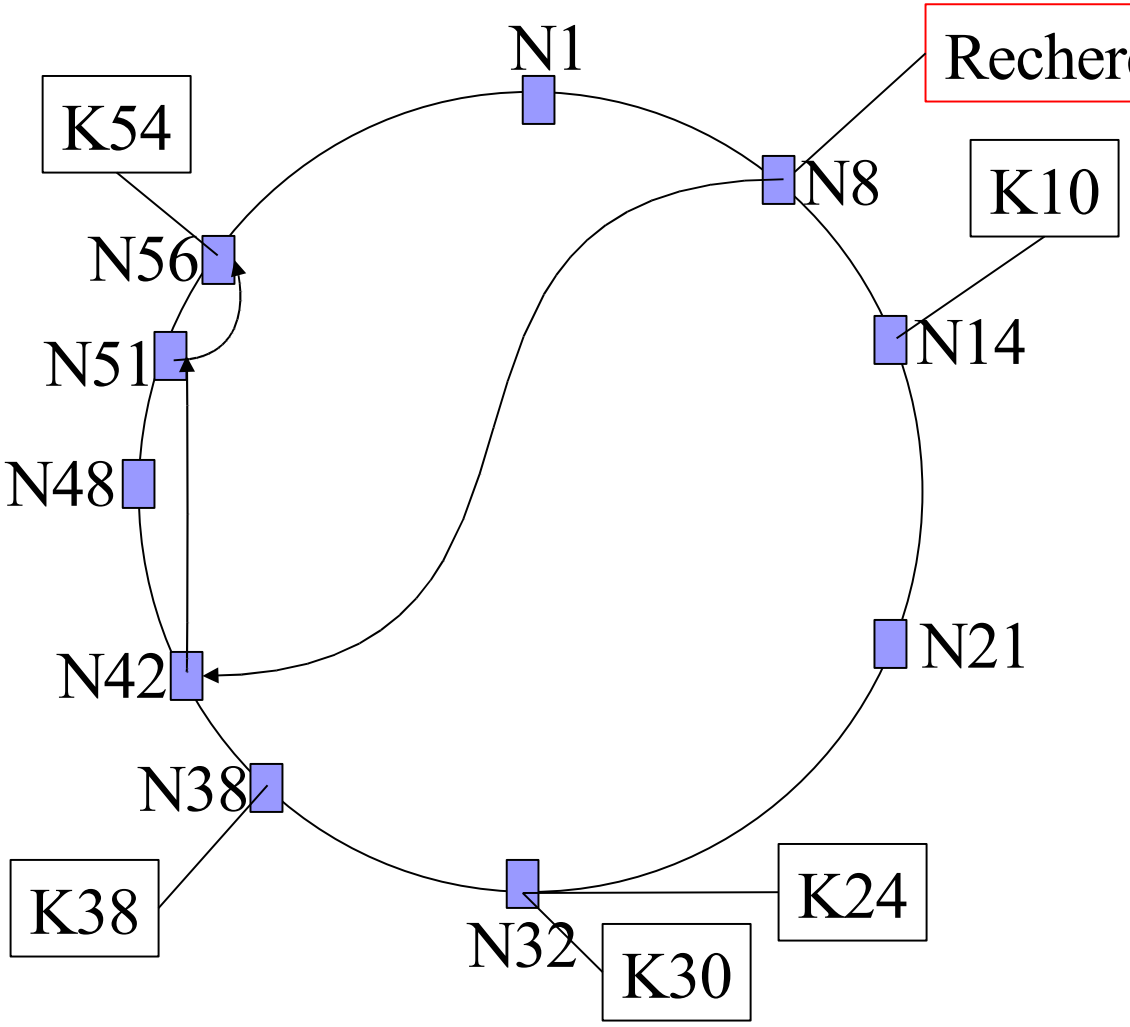


Table routage N8

N8+1	N14
N8+2	N14
N8+4	N14
N8+8	N21
N8+16	N32
N8+32	N42

Bilan de Chord

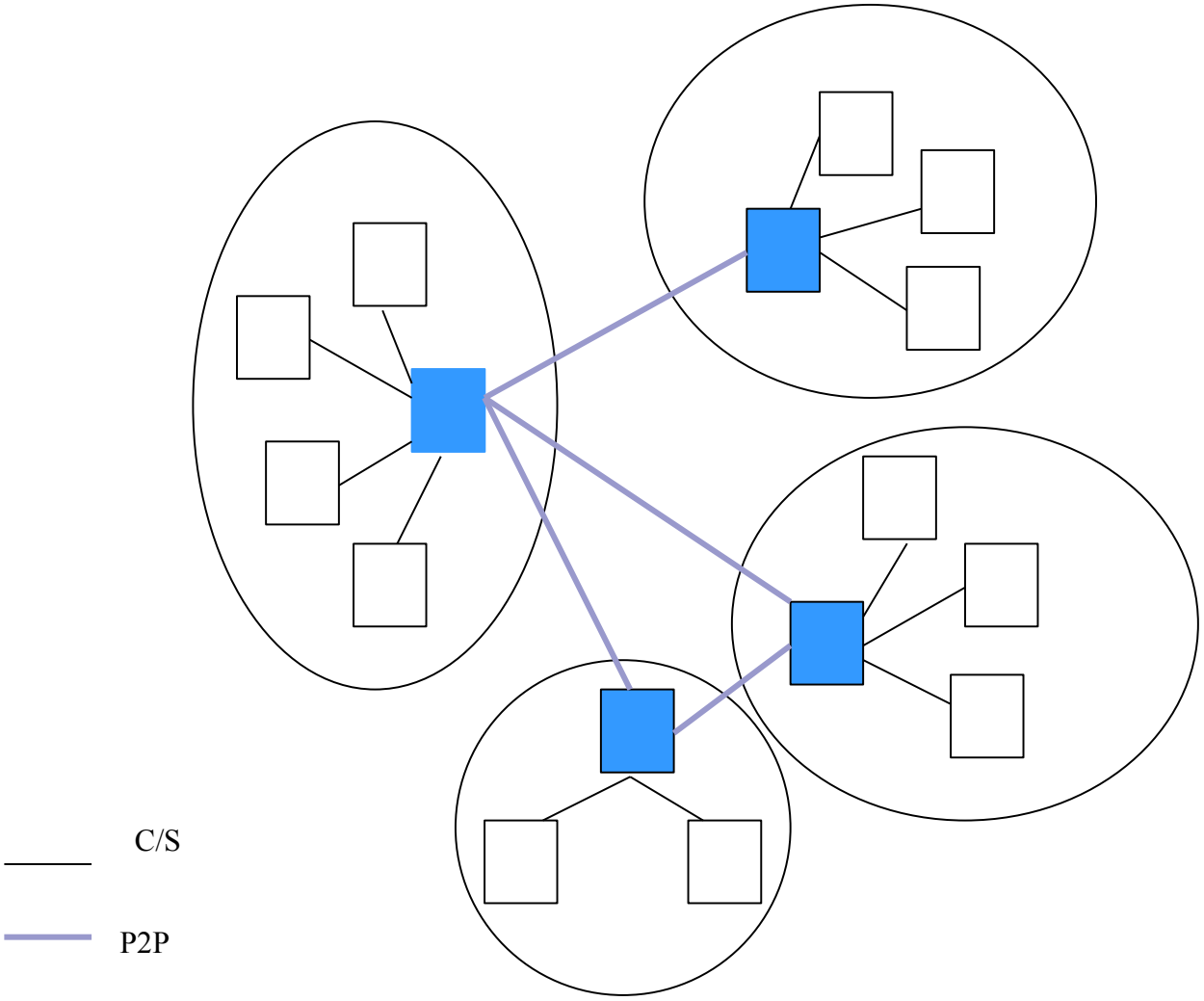
- Recherche efficace : O(Log(n))
- Coût de la mise à jour : O(Log(n))
- Coût de l'ajout d'un nœud : $O(Log_2(n))$
- Algorithme assez simple, avec de bonnes propriétés démontrables
- Résultats expérimentaux confirment

- Type de recherche : égalité
- Pas d'autonomie de stockage et de routage
- Problème de latence :
  - Fonction de recherche minimise le nombre de sauts, mais tous les sauts n'ont pas forcément le même prix (traversée transatlantique e.g)
  - Besoin d'utiliser de l'information sur la distance entre les nœuds

Super-peers (ex. kazaa)

- Eviter les problèmes dus à l'hétérogénéité de la bande passante des nœuds
- Tous les nœuds ne sont plus égaux
  - Les nœuds avec une bonne bande passante sont organisés en P2P : les super-peers
  - Les nœuds avec faible bande passante sont rattachés en mode client/serveur à un super-peer (cluster)
  - Les super-peers disposent d'un index de ressources de leur cluster

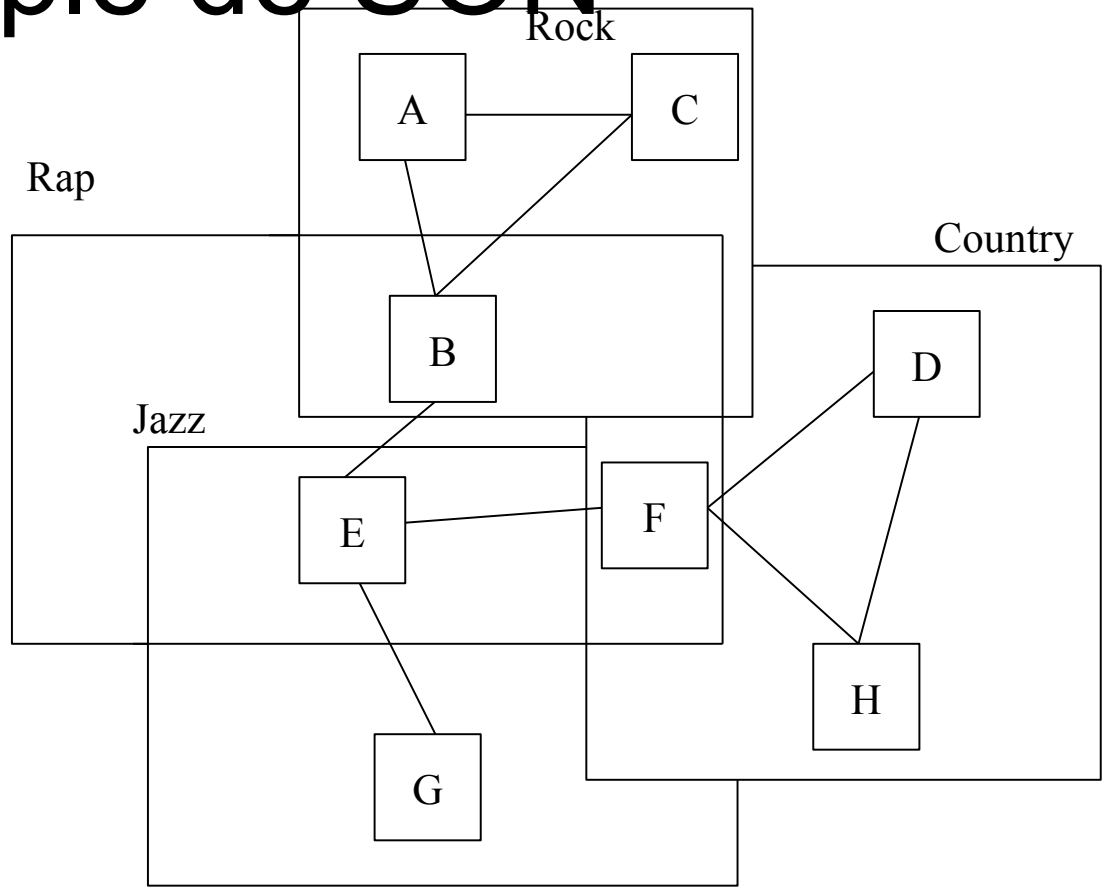
Exemple de super-peer



P2P « sémantique » (ex. SON)

- Ajouter de l'information aux tables de routage
  - Sur le contenu des nœuds (index)
    - Routage par comparaison requête-index
  - Sur le réseau (clustérisation)
    - Regrouper logiquement les nœuds avec même « sémantique »
  - Sur les requêtes
  - Sur les utilisateurs
- Généraliste vs spécifique (à une application)
- Information doit pouvoir être maintenue dynamiquement
- Équilibre entre taille et précision

Exemple de SON



Un nœud est logiquement relié à un autre, par un lien : (n_i, n_j, l_k). Ex : (A, B, 'Rock')

Les nœuds ayant le même *l* forment un SON

Bilan SON

- Repose sur la classification (sémantique)
- Lié à un domaine précis
- Favorise la précision, mais pas la complétude
- Peut se paralléliser pour obtenir rapidement des réponses (requête lancée dans chaque SON sélectionné par le classifieur)
- Les résultats expérimentaux montrent une amélioration notable en nombre de messages par rapport à Gnutella.

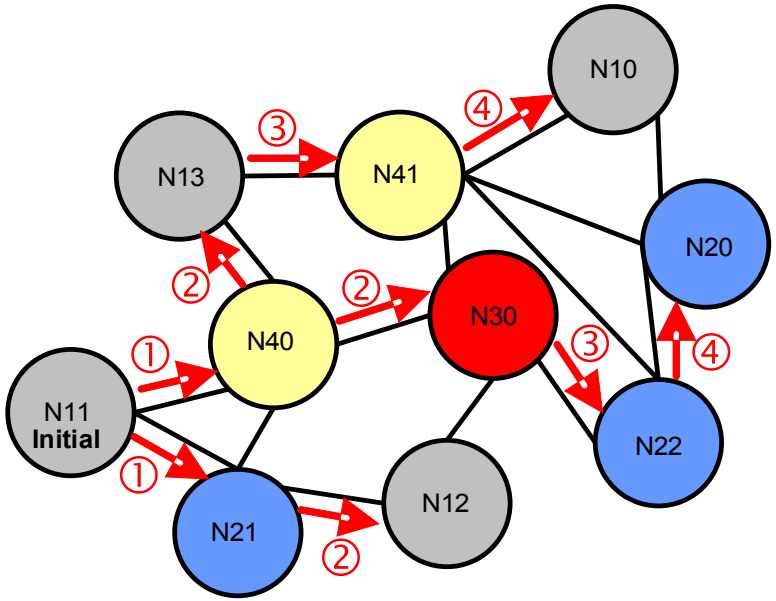
Bilan des architectures

	Non structuré	structuré	hybride
autonomie	Forte	Faible	Moyenne
efficacité	Faible	Forte	Forte
Qualité des réponses	Faible	Forte	Forte
Expressivité du langage	Forte	Faible	Forte
Tolérance aux pannes	Forte	Forte	Faible

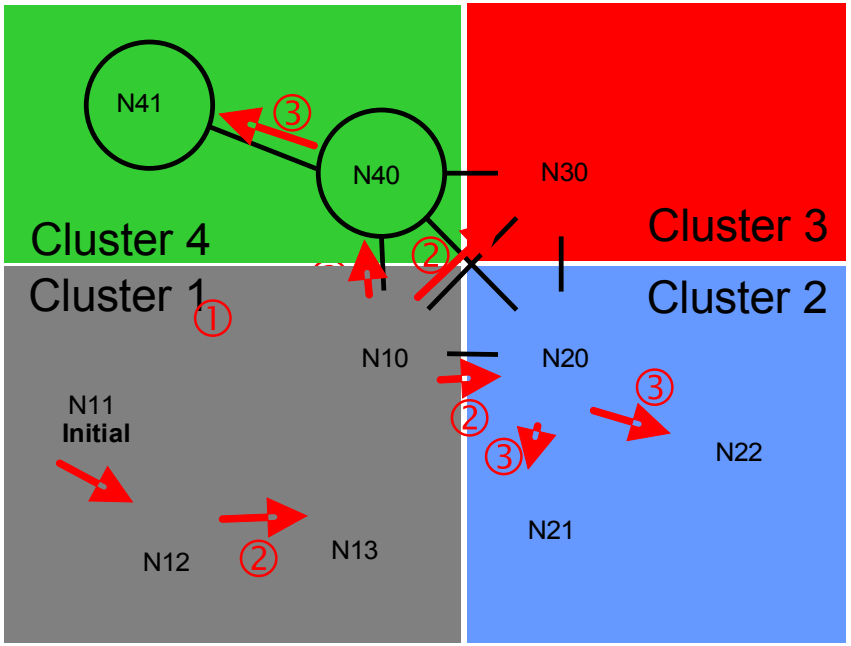
Regroupement logique du réseau

- Rapprocher les nœuds en fonction de critères
  - Sémantiques (Padoue)
  - Géographiques
  - Habitudes, usages
  - Communauté d'intérêts
  - Sécurité
  - …
- Intérêt
  - Reflète mieux la réalité
  - Améliore la recherche des données
    - Réduisant le nombre de sauts
    - Ciblant la requête
    - Améliorant la précision

Exemple



Construction aléatoire du voisinage



Construction du voisinage par sélection des voisins les plus pertinents

Problèmes

- Trouver la bonne topologie
  - Choix du critère
  - Un seul critère ? Plusieurs ?
  - Taille des réseaux logiques (groupes)
- Construire les groupes
- Gestion, maintenance des groupes
- Entrée d'un nouveau nœud
- Interrogation
- Évaluation des performances (benchmark)

Hypothèses

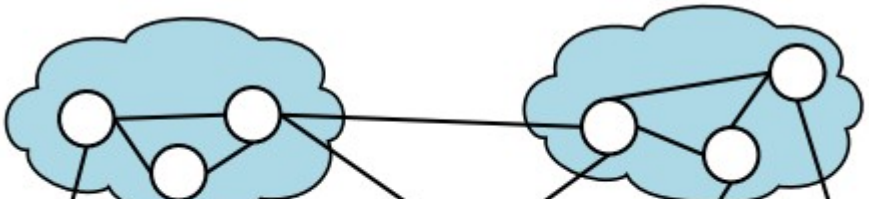
- Partage de données structurées (XML)
- Interrogation : XPath, XQuery
- Réseau P2P non structuré

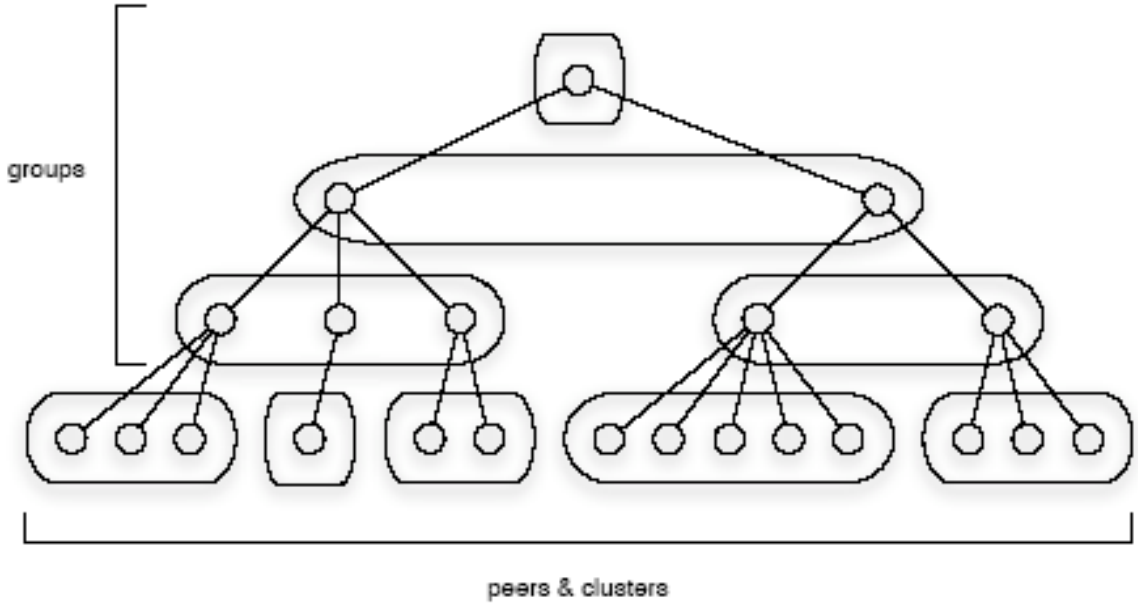
Regroupement de pairs

- Trois problèmes principaux :
  - Méthode et critère(s) de regroupement
  - Organisation du réseau de groupes
  - Routage des requêtes

Architectures existantes

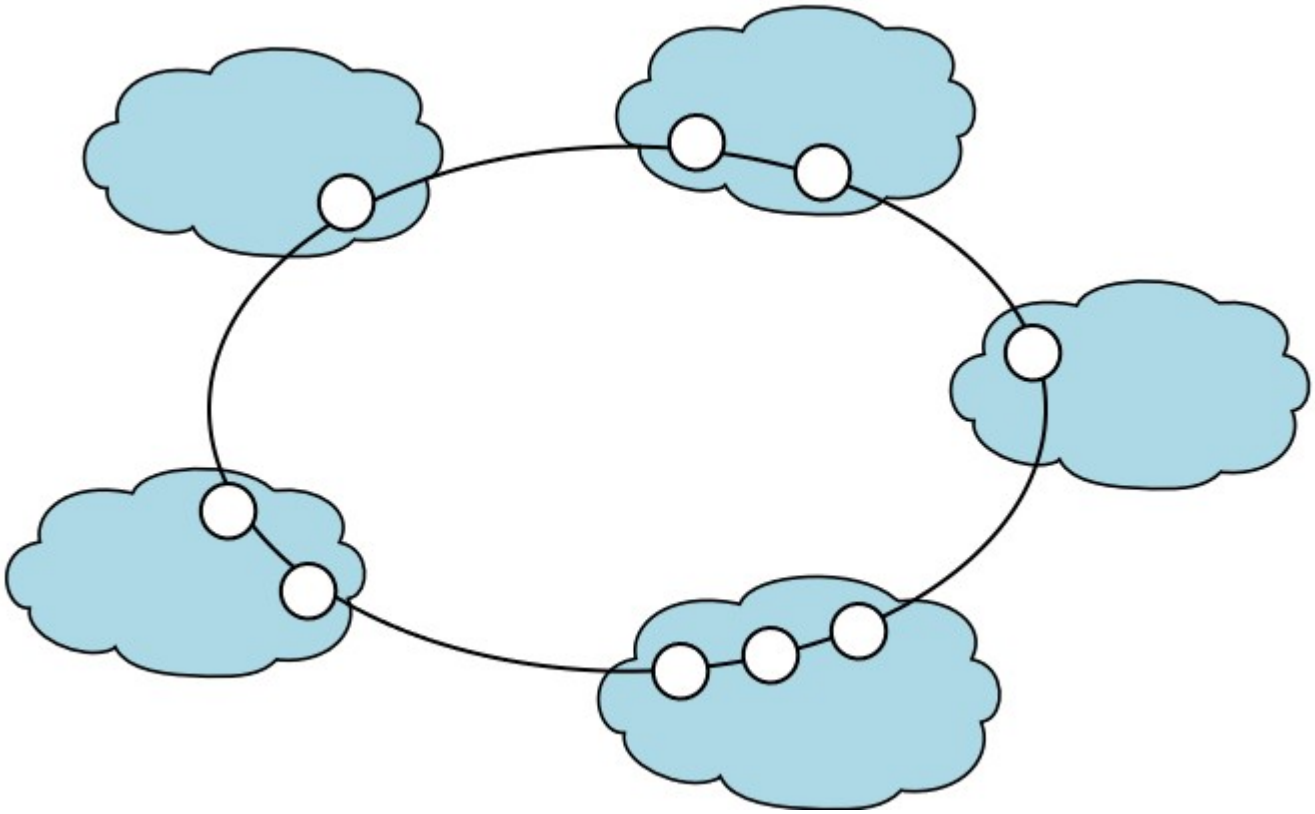
- Adlib (Stanford Peers Group, 2005)
- MediaPeer (PRISM, 2005)
- XPe





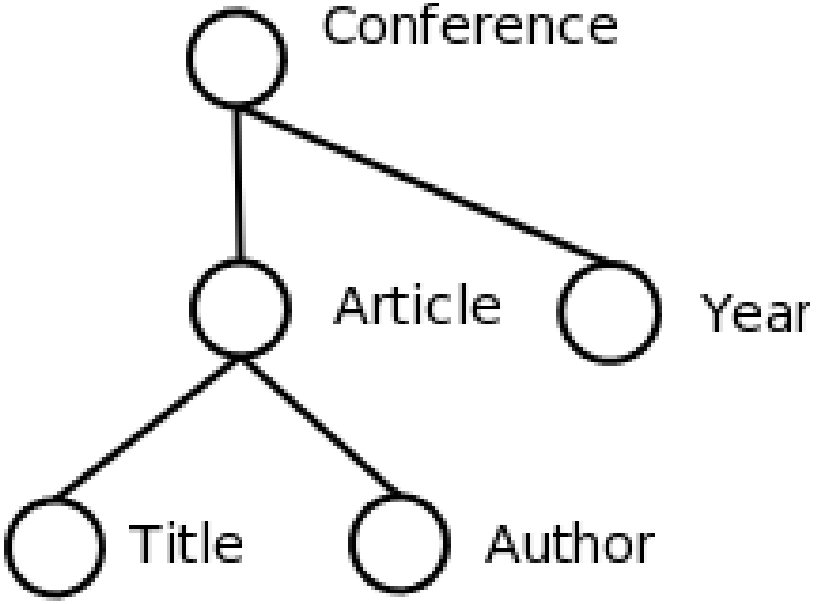
Réseau de groupes structuré

- Utiliser une DHT pour relier les différents groupes et indexer la structure de leurs documents.



Modèle de pair

- Connecté au réseau, partage un ensemble de documents XML, décrits par des structures arborescentes.



Groupe de pairs

- Ensemble de pairs interconnectés, fournit des services au réseau P2P
- Un groupe possède un ID et des pairs de bootstrap (points d'entrée).
- Un pair peut appartenir à plusieurs groupes
- Groupes formés en fonction des structures des documents : un arbre ou une forêt est associé à un groupe.

Réseau de groupes

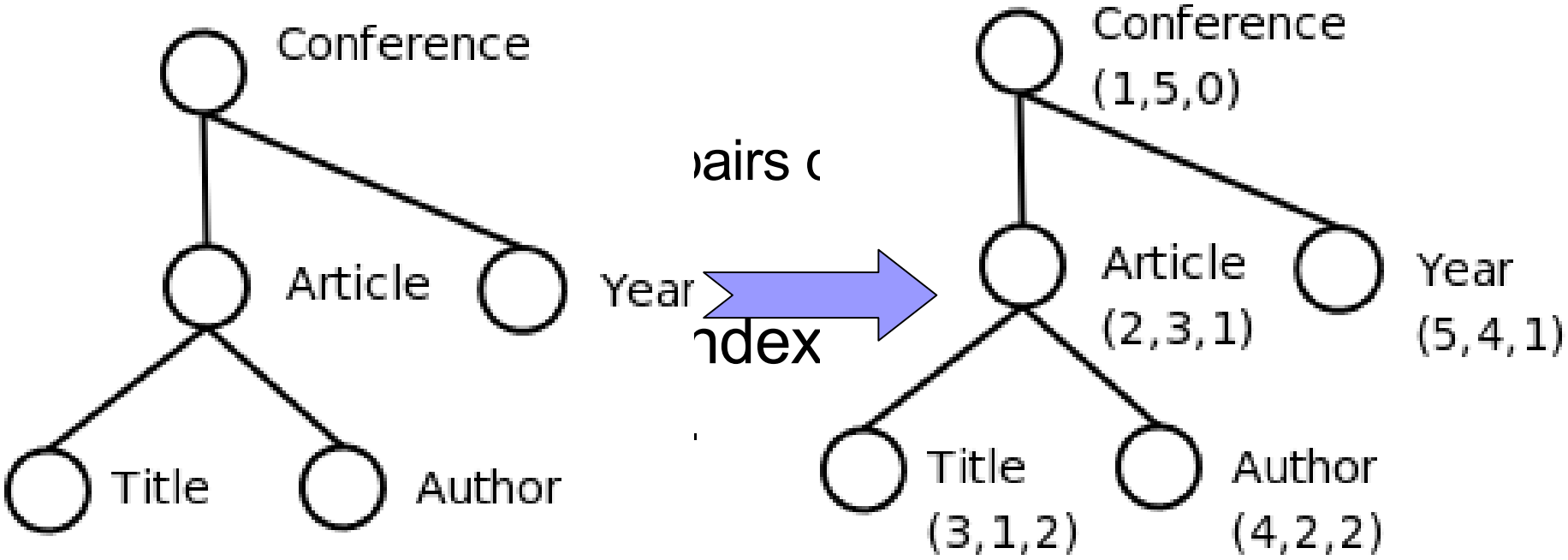
- DHT maintenue par tout ou partie des pairs du réseau.

- Doit permettre :

  - sélection de groupes pertinents pour une requête
  - sélection de groupes pour l'insertion d'un nouveau nœud
  - publication de nouveaux groupes

Couples { clé, valeur }

- { h(Elément), [Schéma, GroupID, Pair] }
  - Représentation des schémas : Dietz
  - Maintenues par tous les groupes



Evaluation d'une requête

Pair Emetteur

DHT

For $b in document

Where $b/auteur/nom/text() = «Amann»

Return

<publi>

$b//titre

</publi>

get (hash(« nom »))

Obtention des triplets

[Schema, GroupID, Pair]

Interrogation des groupes associés.

Localement

Sélection des schémas satisfaisant :

$b/auteur/nom

$b//titre

Arrivée d'un pair

- Pour chaque schéma hébergé,
  - Interrogation de la DHT avec le premier élément
  - Sélection pour chaque groupe obtenu du plus grand schéma associé
  - Si un de ces schémas contient ou est contenu dans le schéma courant, rejoindre le groupe associé.
  - Sinon, créer un nouveau groupe pour ce schéma.
- Publication dans la DHT des {clé,valeur} associés aux éléments et aux nouveaux groupes

Gestion avancée des groupes

- Limiter la prolifération des groupes :
  - Introduire une notion de proximité entre les schémas pour le regroupement.
- Division :
  - Sur quels critères ?
    - Contenu, Localisation …
- Indexation intra-groupe :
  - Type d'index ? (ex : filtres de bloom)
  - Sur quel(s) éléments indexer ?
    - Cout de maintien Vs Gain induit
    - Indexation en fonction des requêtes ?

Conclusion & Perspectives

- Définition d'une architecture de réseau de groupes
  - Utilisation d'une DHT pour lier les groupes et indexer la structure des documents
- Perspectives :
  - Estimation du coût et du gain induit
  - Gestion avancée des groupes
  - Attribution des rôles (bootstrap, DHT)
  - Equilibrage de charge dans la DHT (VLDB 2003)
  - Diffusion de schémas (MenT2)
  - …

Travail réalisé dans le cadre du projet RESPIRE

(Ressources Et Services Pair-à-pair, Interrogation, Réplication)

Projet ARA Masse de données

http://www-poleia.lip6.fr/~tanguy/respires/

MERCI