Recherche approchée d'information dans une base de documents semi-structurés : une application REMIX

Eugen Popovici, Gildas Ménier, Pierre-François Marteau

{eugen.popovici, gildas.menier, pierre-francois.marteau} @ univ-ubs.fr

PaRISTIC - Bordeaux, 21- 23 Novembre 2005

Laboratoire VALORIA

Université de Bretagne-Sud

VALORIA

A PRIM

Le projet ReMIX

REconfigurable Memory for massive data IndeXing

- Projet ACI Masses de Données (septembre 2003 à septembre 2006)
- Recherche par le contenu dans de grandes masses de données
- Conception d'un système matériel, dédié à l'indexation, basé sur une mémoire vive de très grande taille et sur des composants reconfigurables (FPGA) pour accélérer :
  - l'accès aux données
  - et les traitements à réaliser sur les données lues
- Conception d'un mini-système de fichiers dédié
- Réalisation d'un environnement de programmation ad-hoc
- Validation par trois domaines d'application :
  - en indexation d'images
  - en génomique
  - **en recherche documentaire**

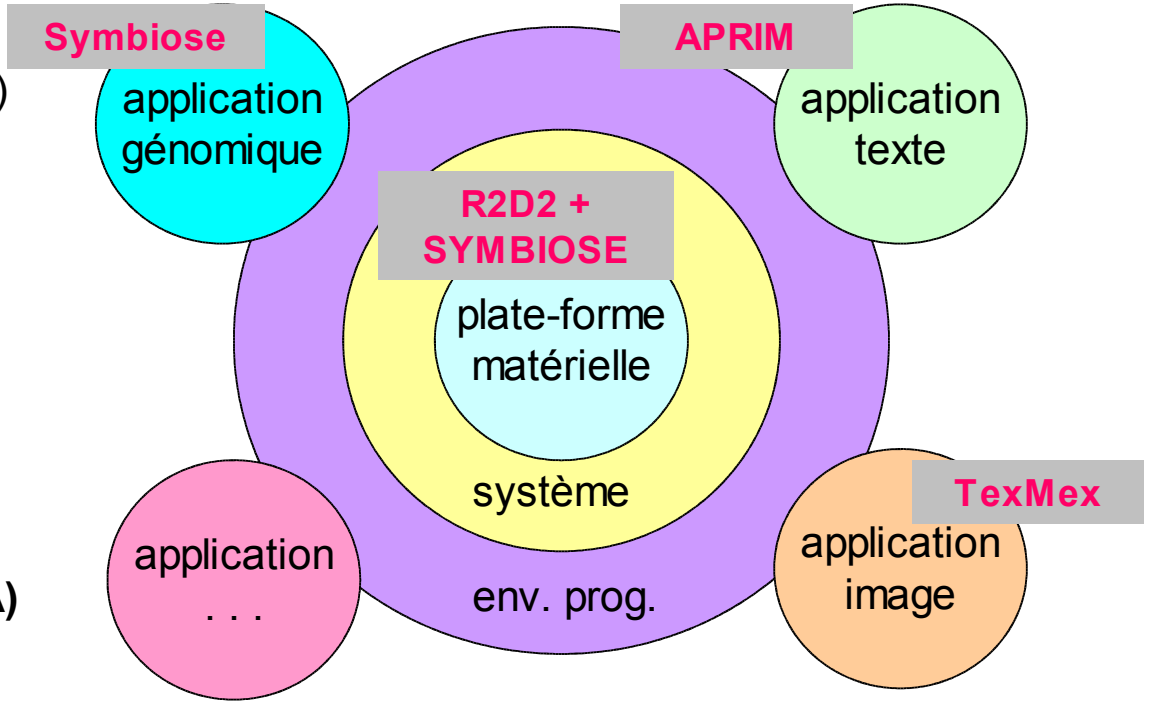
Participants du projet ReMIX

Equipe Symbiose (IRISA)
- **D. Lavenier** (DR)
- S. Rubini (MCF)
- J. Xianyang (Post-doc)

Equipe R2D2 (IRISA)
- F. Charot (CR)
- S. Derrien (MCF)
- G. Georges (ING)

Equipe TexMex (IRISA)
- L. Amsaleg (CR)

Equipe APRIM (VALORIA)
- P-F. Marteau (PR)
- G. Ménier (MCF)
- F. Raimbault (MCF)
- E. Popovici (DOC)



Plan

1. Présentation du projet ReMIX
   i. La recherche par le contenu et l'indexation
   ii. Le verrou technologique de la mémoire
   iii. La solution explorée dans ReMIX
2. Système d'indexation et de recherche approchée
3. Portage de l'application sur ReMIX
4. Avancement

i. La recherche par le contenu et l'indexation

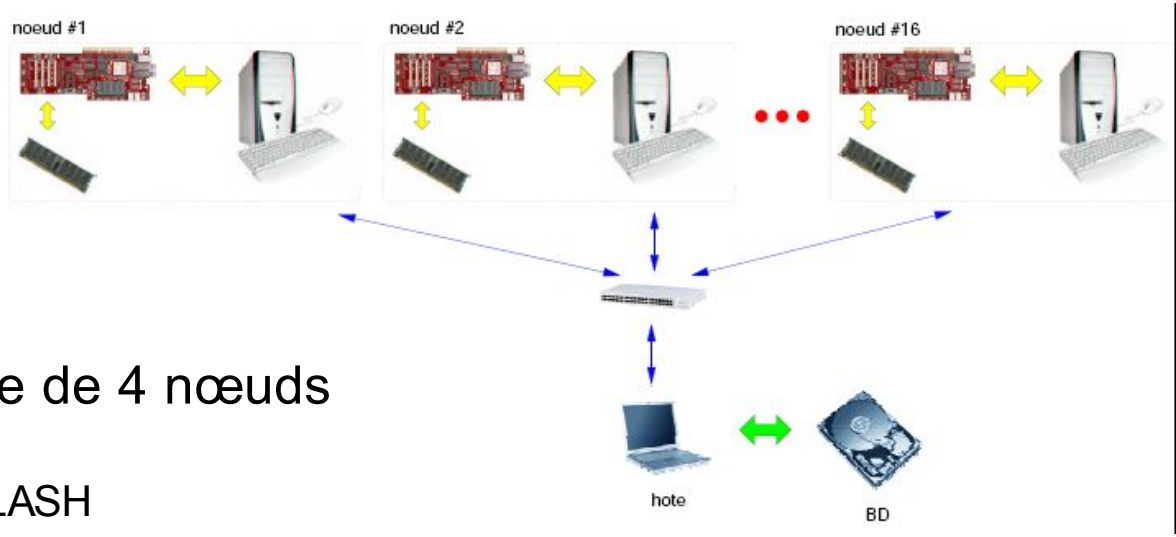
- Caractéristiques des applications visées
  - volumes des données conséquents (xxx Go)
  - requête de recherche par le contenu
  - extraction d'un ensemble de données similaires
  - calcul de distance (coûteux)
- Utilisation d'un index
  - association de plusieurs données à chaque entrée
  - calcul d'une (ou plusieurs) entrée(s) à partir de la requête
  - limitation de l'espace de recherche aux données indexées
- Contribution de ReMIX :
  - **associer mémoire d'index et opérateurs de calcul pour l'extraction rapide des données pertinentes**

ii. Verrou technologique de la mémoire

- Taille de l'index supérieure au volume des données (×10, 20)
  - l'index ne tient pas en totalité dans la RAM
- Accroissement du volume des données supérieur à l'accroissement des capacités d'intégration des RAM
- La mémoire secondaire est 1000 fois plus lente
- Prise en compte des niveaux de hiérarchie mémoire (registres, cache niv.1, cache niv.2, RAM, disques, . . .)
- Complexification des algorithmes
  - pas de solution générale (portabilité faible)
  - performances aléatoires
    - coût de lecture non constant
    - influence des accès antérieurs

iii. Plate-forme matérielle

- Architecture développée pour ReMiX :
  - carte RMEM : (FPGA + connecteur PCI) mémoire FLASH (64Go)
  - système ReMIX : réseau de PC dotés de carte(s) RMEM
- Opérationnelle en décembre
- Mémoire :
  - 512 Go FLASH
  - 1,28 To DD
- Système parallèle de 4 nœuds
  - un nœud =
    - 2 x 64 Go FLASH
    - 2 x FPGA Virtex Pro
    - 2 x 160 Giga octets (disques durs)
    - 2 Go RAM – PC 3 GHz.



iii. Plate-forme logicielle

- Système
  - mini système de fichiers
    - dispositif de traduction d'adresses pour masquer les cellules mémoires défecteuses
- Environnement de Programmation
  - opérationnel / émulateur (utilisable sur un cluster de PC)
  - Java / framework
  - serveur de requêtes
  - le parallélisme est caché

Plan

1. Présentation du projet ReMIX
2. Système d'indexation et de recherche approchée
   i. Contexte
   ii. Schéma de recherche
   iii. Mécanisme d'indexation
   iv. Évaluation
3. Portage de l'application sur ReMIX
4. Avancement

i. Contexte

- Base de documents
  - Différentes sources
  - Différentes structures
  - Différentes utilisations
  - Volumes croissants

- *XML eXtended Meta Language*
  - SGML (*Standard General Meta Language*) simplifié
  - Balises
  - Contraintes structurelles : DTD / Schema
  - Texte ou références externes (images, sons, séquences (à dépendance temporelle) : séries financières, courbes des températures, ...)

- Problématique

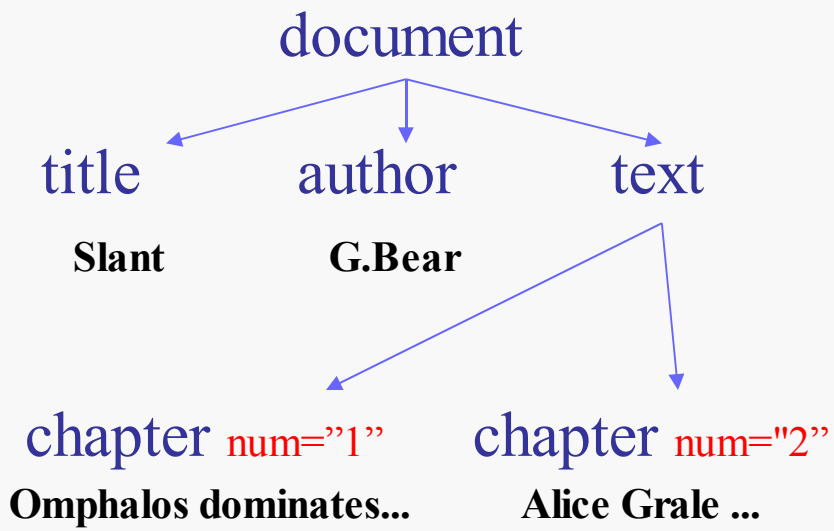
  Besoin d'outils capables :
  - d'analyser, d'indexer et d'interroger ces nouveaux types hétérogènes de documents,
  - de gérer des changements d'échelle liés à l'accroissement des volumes documentaires accessibles

i. Contexte

Document XML

```
<document>
   <title> Slant </title>
   <author> G.Bear </author>
   <text>
         <chapter num="1" >
Omphalos dominates ...
         </chapter>
         <chapter num="2" >
Alice Grale believes ...
         </chapter>
   </text>
</document>
```

DOM Tree
Document Object Model



i. Contexte

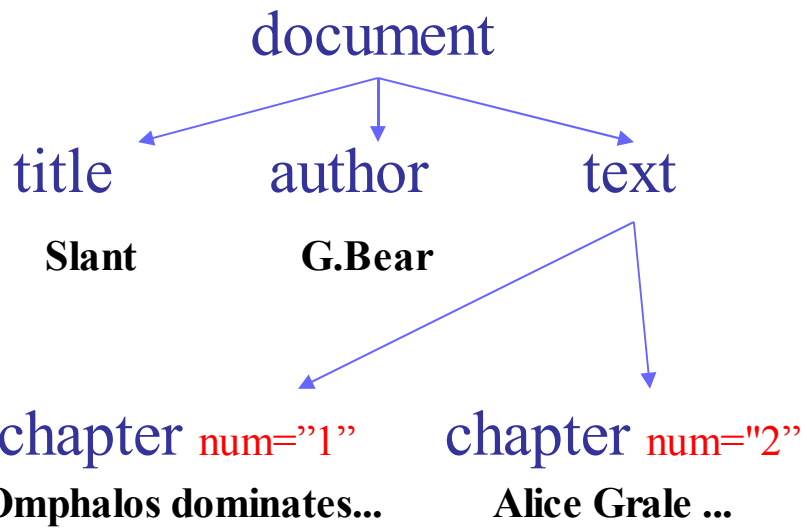
- Requêtes
  - 'Hors contexte' :
    - trouver les documents qui contiennent 'Bear' (n'importe ou)
  - 'En contexte' :
    - trouver les documents qui ont 'Bear' comme auteur
    - trouver les documents dans lesquels 'Alice' apparaît dans un chapitre
  - 'Structure' :
    - trouver les documents qui possèdent exactement la structure XYZ
    - trouver les documents qui possèdent **à peu près** la structure XYZ

i. Contexte XML

Chaque doc XML peut être associé à un arbre DOM 'D'…

… qui est un ensemble de chemins $\{p_i^D\}$ …

… un chemin *p* Def. : sequence de noms d'éléments associés aux pairs (attributs,valeurs).



$\{p_i^D\}=\{$
/document/
/document/title/
/document/title/{Slant}
/document/author/
/document/author/{G.Bear}
/document/text/
/document/text/chapter(num="1")/
/document/text/chapter(num="1")/{Omphalos dominates… }
/document/text/chapter(num="2")/
/document/text/chapter(num="2")/{Alice Grale … }
$\}$

L'intérêt : maîtrise de la complexité de l'algorithme d'alignement d'arbres
→ pb. d'alignement de chemins d'arbres

ii. Schéma de recherche

Requête élémentaire p^r : Un chemin (**evt. incomplet**) avec des conditions sur les éléments et les attributs, terminé par du texte (feuilles)

Exemples :

/document/title/Slant

Recherche des documents avec le mot 'Slant' dans l'élément title de l'élément document.

/document/text/chapter(num >= 1)/Alice

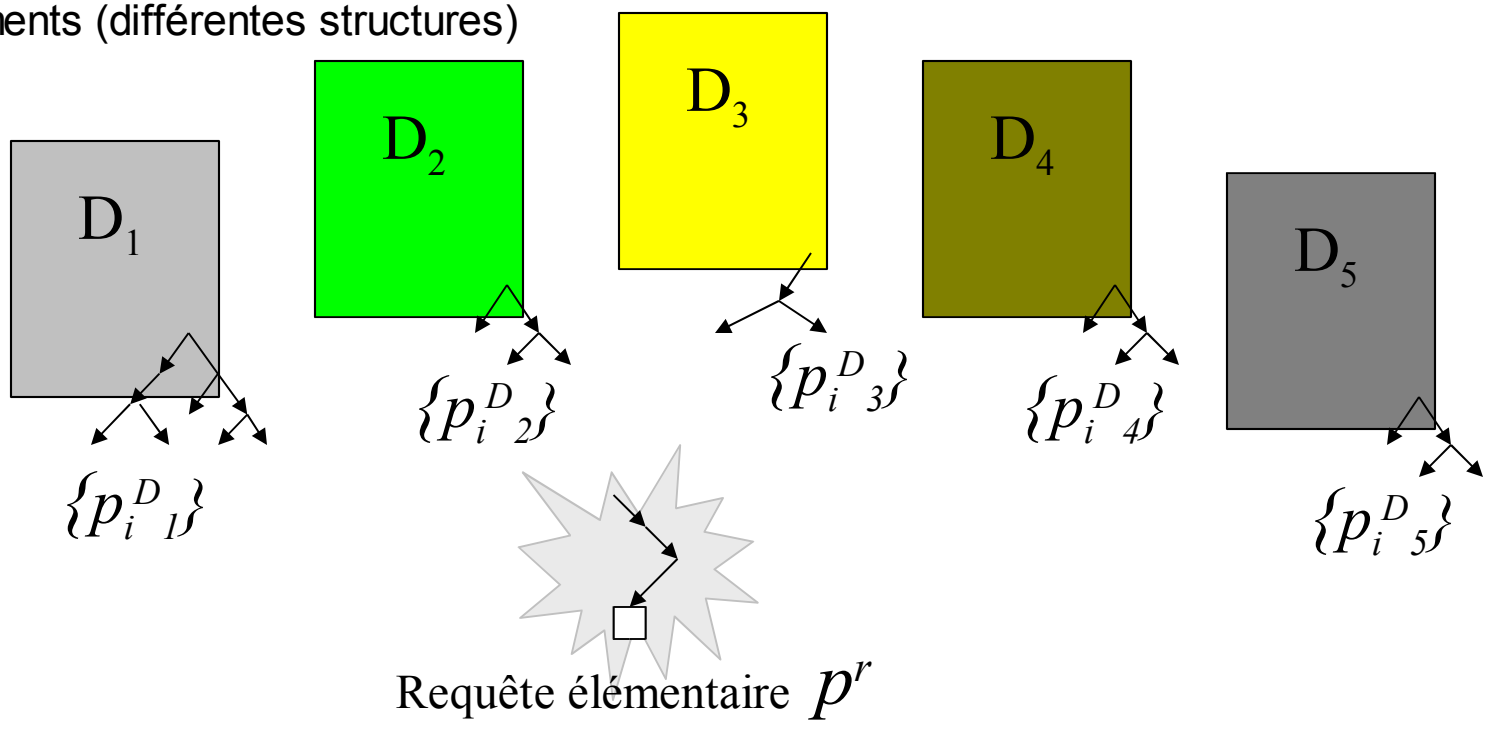
Rechercher les documents avec Alice dans un chapitre de num >= 1

/document/text/chapter/

Recherche des docs avec / à peu près la structure indiquée.

ii. Schéma de recherche

Documents (différentes structures)



Soit une requête élémentaire p^r et un ensemble de doc XML, trouver les meilleurs documents qui possèdent des chemins proches de p^r dans $\{p_i^D\}$

Calcul de distance entre une req p^r et un chemin p_i^D

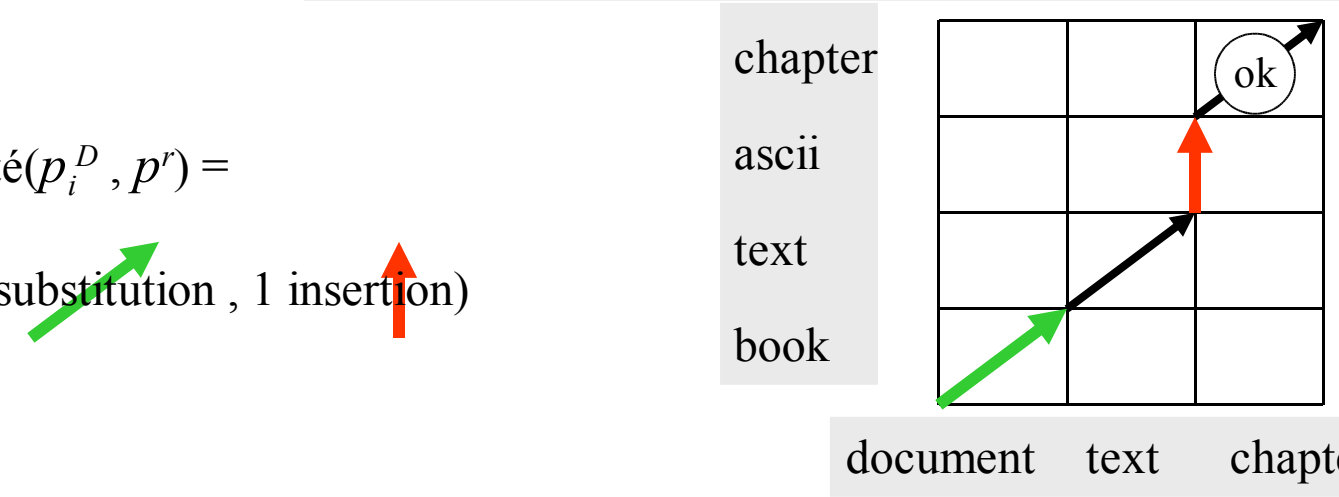
ii. Schéma de recherche

Levenshtein Editing Distance
Wagner&Fisher Algorithm : *O(nm), n et m longueurs de S_1 et S_2.*

chemin p_i^D = /book/text/ascii/chapter(num == 1)/{ … Alice … }

Similarité(p_i^D , p^r) =

= Φ(1 substitution , 1 insertion)



requête p^r = /document/text/chapter(AND (num >= 1) (num < 4))/Alice

ii. Schéma de recherche

A partir des requêtes élémentaires on peux exprimer des **requêtes complexes**

Évaluation des conditions sur les attributs pour chaque nœud + op ensemblistes

(or

/**book***(or (== author bob) (== author fred))* / **chapitre***(or (== num 5) (== num 7)*)/computer

/**article***(or (== author bob) (<> author Franck)*)/computer

)

Coût pour une requête élémentaire : $o(n_{ri*eval(i,m)}*m)$

Coût pour une requête complexe : $o(N_R*n_{ri*eval(i,m)}*m)$

Hypothèse : $eval(i,m)>=n_r$

$$o > o(N_R*n_{ri}^2*m)$$

iii. Mécanisme d'indexation

Schéma d 'indexation

/book/text/ascii/chapter(num == 1)/{ … Alice … }

$/E_1,\{ (a_{11},v_{11}) ,... (a_{1n(1)},v_{1n(1)}) \}/.../E_h,\{(a_{h1},v_{h1}) ,... (a_{hn(h)},v_{hn(h)}) \}/[\ldots \text{term}_k \ldots]$

Liste Inverse

(Recherche ~ $O(n^{0.85})$, Espace ~ $O(n^{0.85})$, Construction O(n), Suppression O(n))

term_k -> $/E_1,\{ (a_{11},v_{11}) ,... (a_{1n(1)},v_{1n(1)}) \}/.../E_h,\{(a_{h1},v_{h1}) ,... (a_{hn(h)},v_{hn(h)}) \}/$

terminal spécial pour l'indexation structurelle (#siriusempty#)

Espace de stockage des tables/listes inverses

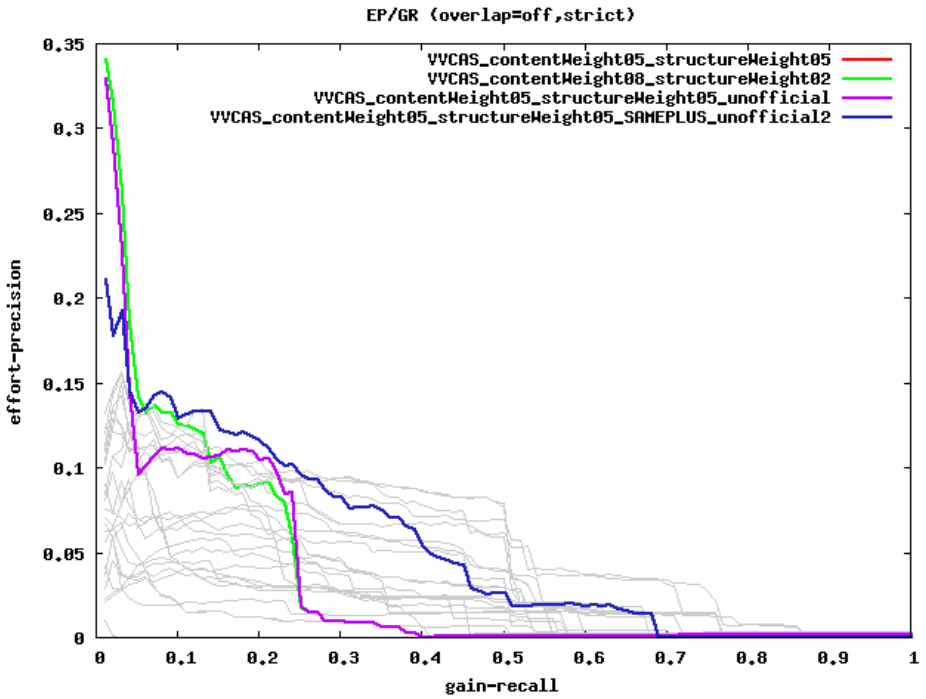
- dépend de la base de documents
- en moyenne

taille indexe = 4 à 6 fois la taille de la base initiale

iv. Approche évaluée à INEX 2005

- Pertinence de l'approche évaluée et validée à INEX* 2005 pour l'interrogation flexible de documents XML (VVCAS task)

  *) INitiative for the Evaluation of XML Retrieval



Système d'indexation et de recherche approchée

Interrogation approchée de la base de documents :

Pour évaluer la requête complexe :

parcourir l'arbre de la requête complexe
feuille = requête élémentaire

Stockage
Accès rapide → **recherche des listes inverses pour un terminal**

Calcul rapide → **évaluation de toutes les distances**

classement

opérations ensemblistes

Plan

1. Présentation du projet ReMIX

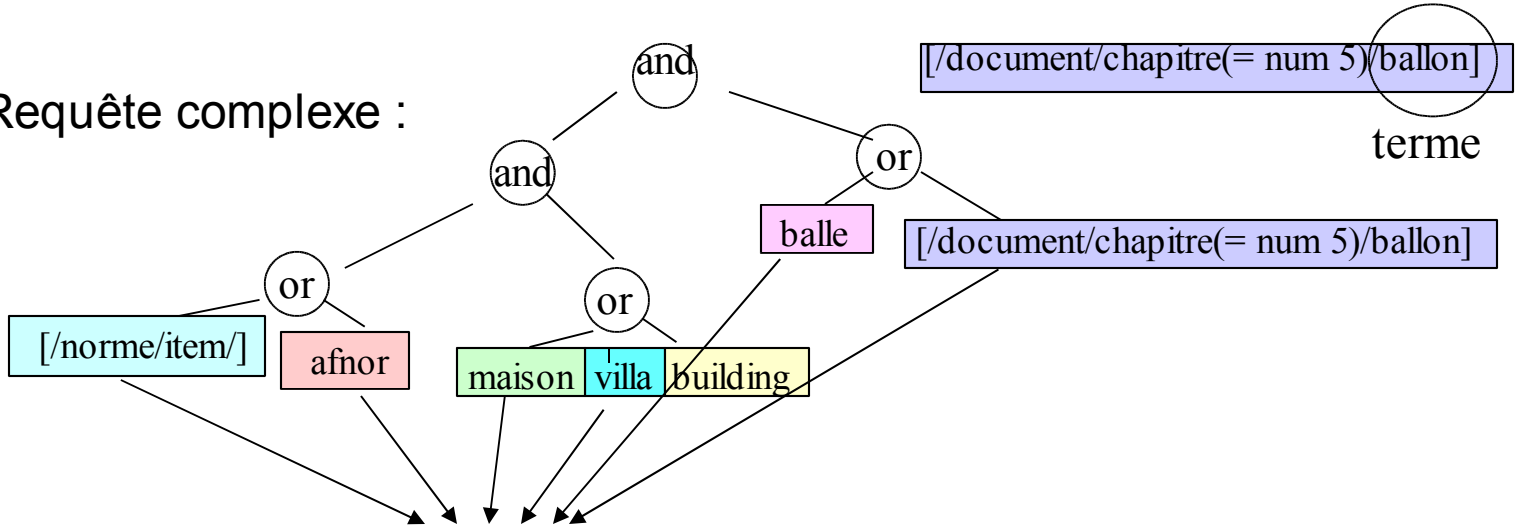
2. Système d'indexation et de recherche approchée

3. Portage de l'application sur ReMIX
   i. Principe d'implantation
   ii. Recherche
   iii. Répartition mémoire

4. Avancement

i. ReMIX : implantation

Requête complexe :

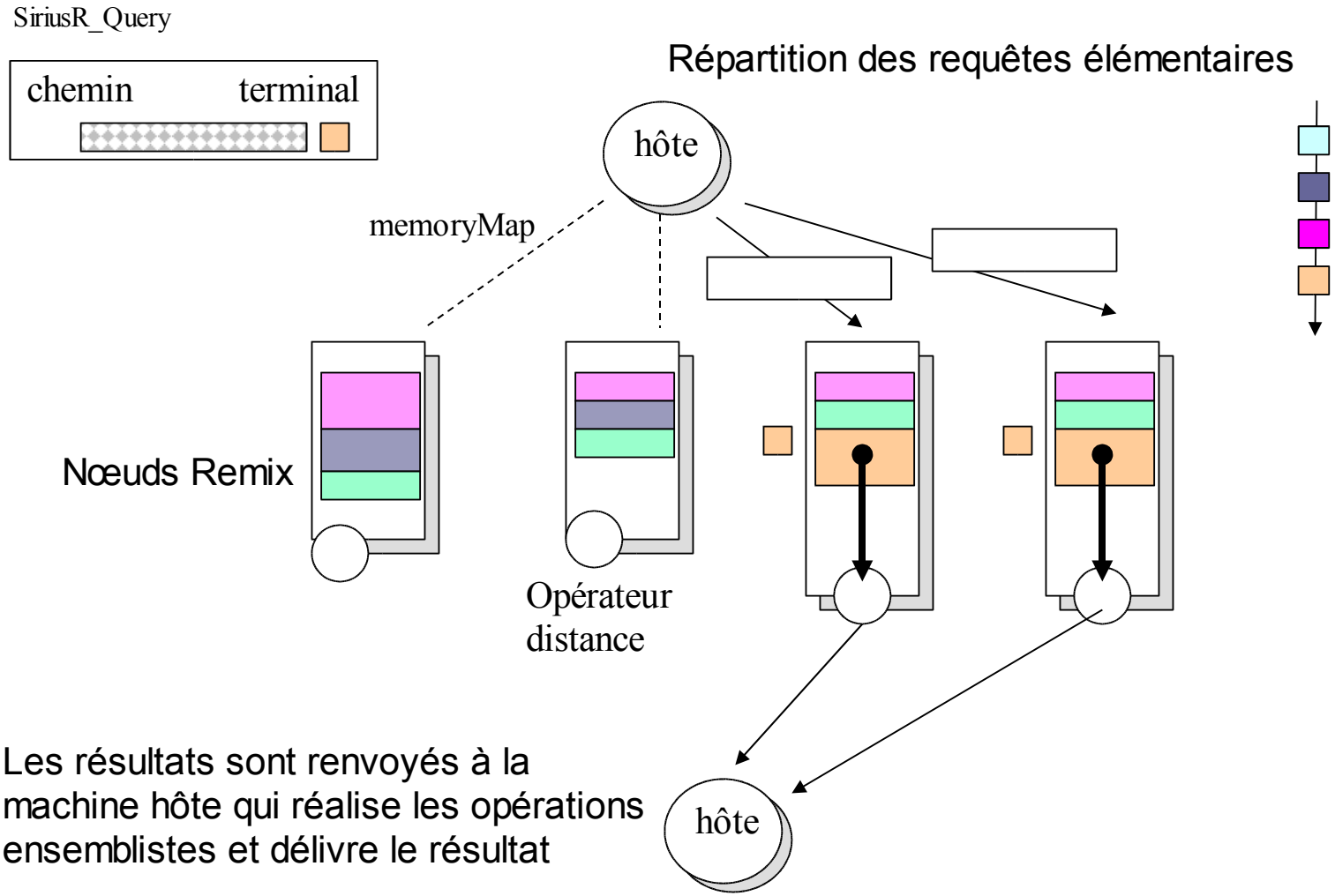


Requête élémentaire

Indépendance entre chaque requête élémentaire ( )
répartition des requêtes élémentaires sur les nœuds

Opérateurs ensemblistes avec fusion ( )
sur la machine ' hôte '

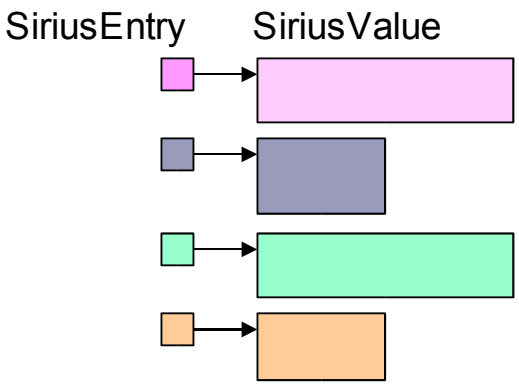
ii. ReMIX : Recherche

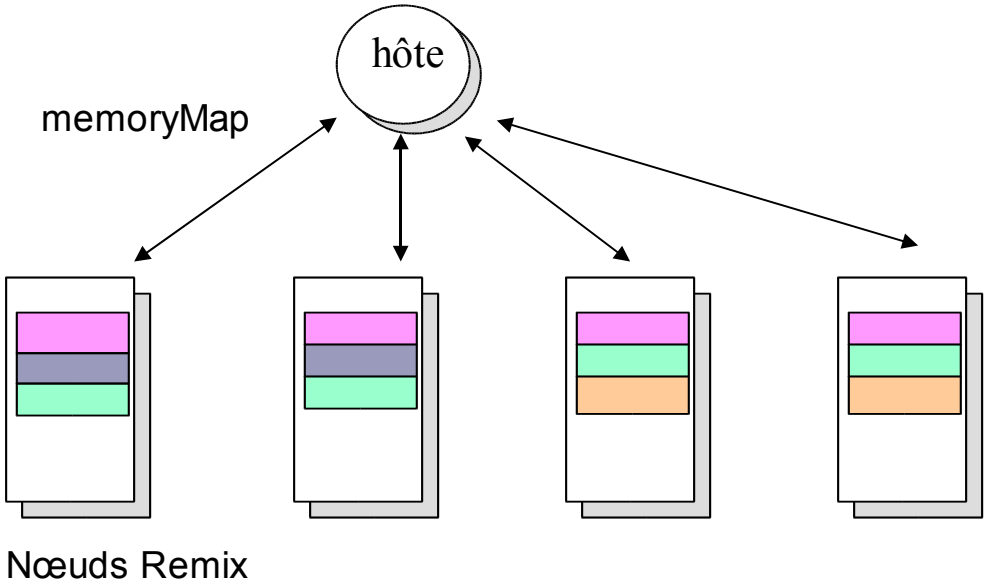


Les résultats sont renvoyés à la machine hôte qui réalise les opérations ensemblistes et délivre le résultat

iii. ReMIX : Organisation mémoire

La construction d'index est réalisée indépendamment de ReMIX...





4. Avancements

- Évaluation de la pertinence de l'approche pour la recherche flexible des documents XML à INEX 2005 :
  "SIRIUS: A Lightweight XML Indexing and Approximate Search System at INEX 2005", E.Popovici, G. Ménier, P.F. Marteau, INEX 2005 workshop pre-proceedings
- Extension pour gérer les informations séquentielles ou à dépendances temporelles :
  "Information Retrieval of Sequential Data in Heterogeneous XML Databases", E. Popovici, PF. Marteau, G. Ménier, AMR 2005
- Un premier prototype adapté au framework ReMIX a été réalisé et est en cours d'évaluation
- Plateforme matérielle quasi opérationnelle => Portage des prémiéres applications sur ReMIX à partir de décembre 2005.

?