

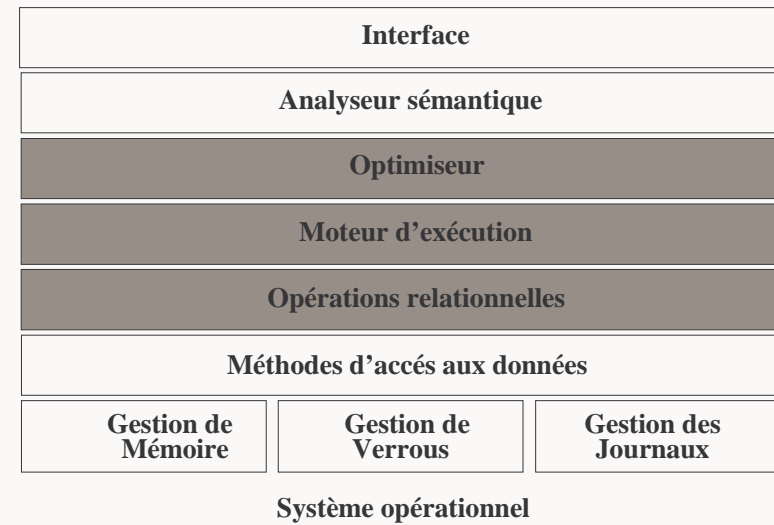
# OPTIMISATION DE QUESTIONS

- OBJECTIFS ET ETAPES DE L'OPTIMISATION
- ANALYSE SYNTAXIQUE ET SEMANTIQUE
- CONSTRUCTION DES PLANS CANDIDATS
- CHOIX DU PLAN OPTIMAL

## Architecture en couche d'un SGBD

Page 2

© Philippe Pucheral, Luc Bouganim

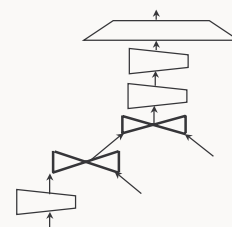
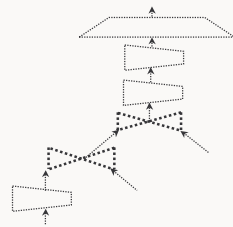


## Optimisation : ???

Page 3

© Philippe Pucheral, Luc Bouganim

Select  
From  
Where



Requête SQL

Arbre logique

Arbre Physique

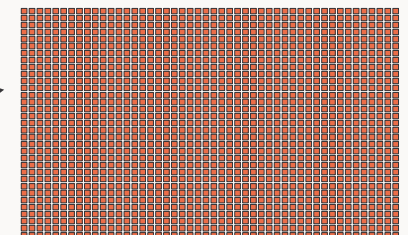
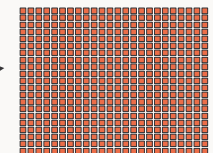
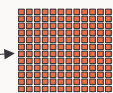
Optimisation

## Le problème : équivalence sémantique

Page 4

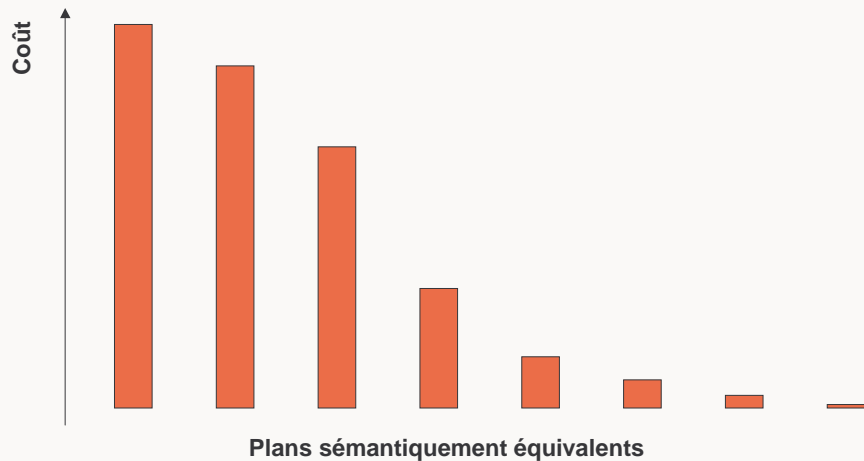
© Philippe Pucheral, Luc Bouganim

- Une question
- Plusieurs expressions équivalentes en SQL
- Plusieurs expressions équivalentes en algèbre
- Plusieurs algorithmes algébriques équivalents



Donner des exemples de requêtes équivalentes en SQL

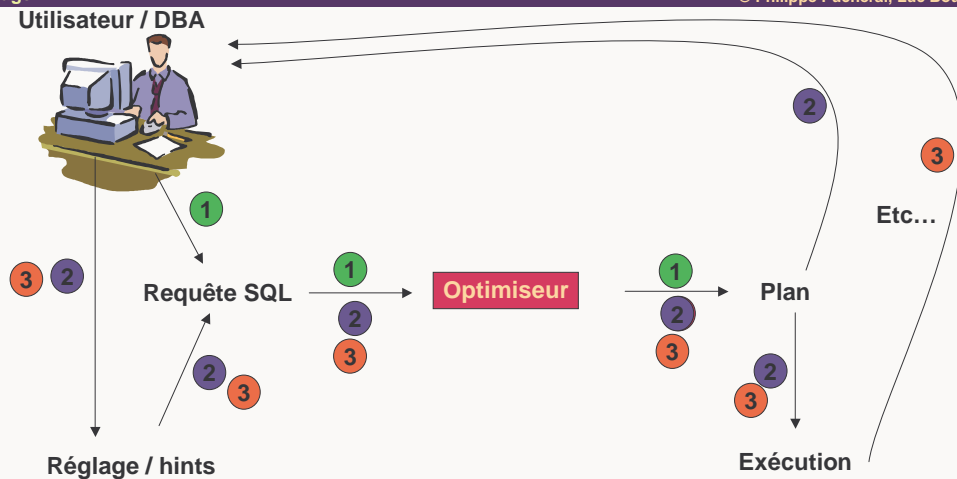
## Le Problème : différence de performances



## Les acteurs de l'optimisation

- **Idéalement :**
  - 2 requêtes équivalentes en SQL (langage déclaratif) doivent, après l'optimisation, produire le même arbre algébrique !
  - Qui plus est, cet arbre doit être le meilleur !
  - Seuls les concepteurs de SGBD (noyau) doivent comprendre l'optimisation et l'exécution
- **Dans la pratique :**
  - 2 requêtes équivalentes ne donnent pas toujours le même plan
  - Le plan n'est pas toujours le meilleur !
  - l'utilisateur (concepteur de l'appli) devra comprendre !!!

## Les acteurs de l'optimisation (suite)



## Objectifs de l'optimisation

Trouver le **meilleur** plan d'exécution .... MEILLEUR ???

- **Donnant les résultats le + vite ....**
  - Optimisation pour le temps de réponse (response time)
- **Minimisant la consommation de ressources**
  - Optimisation du travail total (Total work)
- **Minimisant le temps de délivrance des premiers tuples**
  - Optimisation de la latence (Latency / First tuples ...)



Trouver des applications pour chaque type d'optimisation

# ETAPES DE L'EVALUATION

1. CONTROLE SYNTAXIQUE ET SEMANTIQUE
2. SIMPLIFICATION DE LA REQUETE
3. DECOMPOSITION EN OPERATIONS ELEMENTAIRES ET CONSTRUCTION DES PLANS D'EXECUTION CANDIDATS
4. CALCUL DU COUT DE CHAQUE PLAN, CHOIX DU MEILLEUR ET EXECUTION

ETAPES 1,2 et 3 : INDEPENDANTES DES DONNEES

ETAPE 4: DEPENDANTE DES DONNEES

# Analyse de la question (étape I)

## 1. ANALYSE SYNTAXIQUE

- VERIFIER LA SYNTAXE ET LA COHERENCE DE LA QUESTION PAR RAPPORT AU SCHEMA DE LA BD :
- => EXISTENCE DES RELATIONS ET DES ATTRIBUTS

## 2. ANALYSE SEMANTIQUE

- VERIFIER LA CORRECTION SEMANTIQUE DE LA QUESTION
- RECHERCHER DES QUESTIONS EQUIVALENTES PLUS SIMPLES

# Exemple

BASE DE DONNEES : BUVEUR (NB,NOM,PRENOM,ADRESSE,REGION)  
ABUS (NB,NV,DATE,QUANTITE)  
VIN (NV,CRU,REGION,MILLESIME,DEGRE)

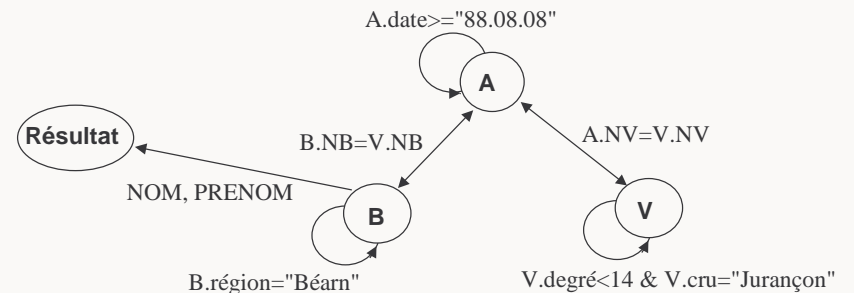
REQUETE : "NOM ET PRENOM DES BUVEURS BEARNAIS AYANT BU DU JURANCON DE 1976 DE DEGRE < 14, LE 8 AOUT 1988"

```
SELECT NOM, PRENOM
FROM VIN V, ABUS A, BUVEUR B WHERE V.MILLESIME = 1976
AND V.DEGRE < 14
AND V.CRU = "JURANCON"
AND A.DATE >= "88-8-8"
AND B.REGION = "BEARN"
AND B.NB = A.NB
AND A.NV = V.NV
```

# Vérification de la correction de la question (1)

## NOTION 1 : GRAPHE DE CONNEXION DES RELATIONS

- UN SOMMET EST ASSOCIE A CHAQUE RELATION
- UNE JOINTURE EST REPRESENTEE PAR UN ARC
- UNE RESTRICTION PAR UNE BOUCLE SUR UNE RELATION
- UNE PROJECTION PAR UN ARC VERS LE NOEUD RESULTAT
- LA QUESTION EST "INCORRECTE" SI LE GRAPHE EST NON CONNEXE



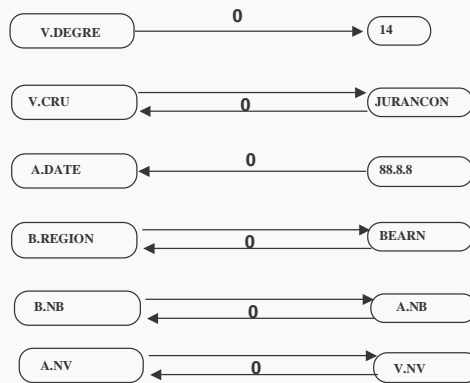
## Vérification de la correction de la question (2)

Page 13

© Philippe Pucheral, Luc Bouganim

### NOTION 2 : GRAPHE DE CONNEXION DES ATTRIBUTS

- UN SOMMET EST ASSOCIE A CHAQUE REFERENCE D'ATTRIBUT OU DE CONSTANCE,
- UN ARC DE LA FORME  $X \xrightarrow{c} Y$  REPRESENTE  $X \leq Y + c$ ,
- UNE EGALITE EST MATERIALISEE PAR DEUX ARCS VALUES PAR 0,
- UNE QUESTION EST CONTRADICTOIRE SI SON GRAPHE DE CONNEXION DES ATTRIBUTS PRESENTE UN CYCLE DONT LA SOMME DES VALUATIONS EST NEGATIVE



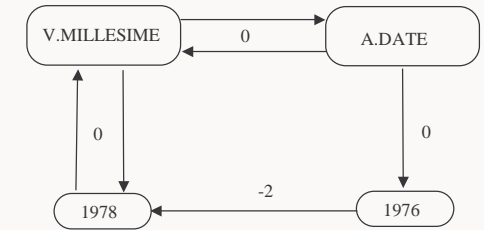
## Vérification de la correction de la question (3)

Page 14

© Philippe Pucheral, Luc Bouganim

EXEMPLE :

```
SELECT * FROM VIN V, ABUS A
WHERE V.MILLESIME = 1978
AND A.DATE <= 1976
AND V.MILLESIME = A.DATE
```



IL EXISTE UN CYCLE DE VALUATION -2:

➔ CETTE SELECTION EST INSATISFAISABLE

## Questions équivalentes (étape II)

Page 15

© Philippe Pucheral, Luc Bouganim

### NOTION 3 : QUESTIONS EQUIVALENTES

DEUX QUESTIONS SONT EQUIVALENTES SSI ELLES DONNENT LE MEME RESULTAT POUR TOUTE EXTENSION DE LA BASE DE DONNEES

2 MOYENS :

- UTILISATION DE LA LOGIQUE DES PREDICATS
- UTILISATION DES CONTRAINTES D'INTEGRITE

## Questions équivalentes : Utilisation de la logique

Page 16

© Philippe Pucheral, Luc Bouganim

EXEMPLE : REQUETE INITIALE :

```
SELECT *
FROM VIN V
WHERE ((V.DEGRE = 12)           P
OR (V.CRU = 'MORGON')         Q
OR (V.CRU = 'CHENAS'))       R
AND NOT ((V.CRU = 'MORGON')   Q
OR (V.CRU = 'CHENAS'))       R
```

CRITERE :  $((P \vee Q \vee R) \wedge \text{NOT}(Q \vee R)) \Leftrightarrow (P \wedge \text{NOT}(Q \vee R))$

REQUETE EQUIVALENTE SIMPLIFIEE :

```
SELECT * FROM VIN V
WHERE (V.DEGRE = 12)           P
AND NOT ((V.CRU = 'MORGON')   Q
OR (V.CRU = 'CHENAS'))       R
```



Pourquoi l'utilisateur écrirait-il une requête idiote comme celle ci ?

## Utilisation des contraintes d'intégrité (1)

Page 17

© Philippe Pucheral, Luc Bouganim

### PRINCIPE :

ETANT DONNE :

- UNE QUESTION DE QUALIFICATION **Q**
- UN ENSEMBLE DE CONTRAINTES D'INTEGRITE  
{I1, I2, ... In}

**SI Q EST CONTRADICTOIRE AVEC L'UNE AU MOINS DES CONTRAINTES I<sub>i</sub>,**

**ALORS LA QUESTION A UNE REPONSE VIDE.**

## Utilisation des contraintes d'intégrité (2)

Page 18

© Philippe Pucheral, Luc Bouganim

### EXEMPLE : REQUETE UTILISATEUR :

```
SELECT * FROM VIN
WHERE CRU = 'JURANCON'
AND DEGRE < 10
```

**CONTRAINTE D'INTEGRITE SUR VIN :** CRU = 'JURANCON' → DEGRE > 12

### REQUETE MODIFIEE :

```
SELECT * FROM VIN
WHERE CRU = 'JURANCON'
AND DEGRE < 10
AND DEGRE > 12 → pas de réponse possible
```

## Utilisation des contraintes d'intégrité (3)

Page 19

© Philippe Pucheral, Luc Bouganim

**Une contrainte d'intégrité peut également servir à compléter le critère de la requête utilisateur afin de profiter d'index**

**EXEMPLE :** Index secondaire déclaré sur l'attribut Vin.degré

**CONTRAINTE D'INTEGRITE SUR VIN :** CRU = 'JURANCON' => DEGRE > 12

### REQUETE UTILISATEUR :

```
SELECT *
FROM VIN
WHERE CRU = 'JURANCON'
```

### REQUETE MODIFIEE :

```
SELECT *
FROM VIN
WHERE CRU = 'JURANCON'
AND DEGRE > 12
```

## Construction et choix du meilleur plan

Page 20

© Philippe Pucheral, Luc Bouganim

**LORSQUE LA REQUETE EST CORRECTE, LE BUT EST D'ASSOCIER UN PLAN D'EXECUTION A LA REQUETE, AFIN QUE LE COUT D'EXECUTION SOIT MINIMUM.**

### 2 METHODES :

- LA RESTRUCTURATION ALGEBRIQUE
- L'EVALUATION DE PLANS

### REMARQUE :

**CES DEUX METHODES NE SONT PAS EXCLUSIVES**

## Importance du choix du meilleur plan (1)

Page 21

© Philippe Pucheral, Luc Bouganim

**BASE DE DONNEES :** BUVEUR (NB,NOM,PRENOM,ADRESSE,REGION)  
ABUS (NB,NV,DATE,QUANTITE)  
VIN (NV,CRU,REGION,MILLESIME,DEGRE)

**REQUETE :** "NOM DES BUVEURS BEARNAIS AYANT BU DU JURANCON"

**REQUETE SQL :**

```
SELECT NOM, NV, DATE
FROM VIN V, ABUS A, BUVEUR B
WHERE B.NB = A.NB
      AND A.NV = V.NV
      AND V.CRU = "JURANCON"
      AND B.REGION = "BEARN"
```

## Importance du choix du meilleur plan (2)

Page 22

© Philippe Pucheral, Luc Bouganim

**DISTRIBUTION DES DONNEES DANS LA BASE :**

- 100 BUVEURS DONT 5 SONT BEARNAIS (20 pages)
- 500 VINS DONT 50 SONT DE CRU JURANCON (50 pages)
- 2000 ABUS (200 pages)

**ON SUPPOSE UNE EQUI-REPARTITION DES DONNEES**

- • EN MOYENNE 20 ABUS PAR BUVEUR ET 4 ABUS PAR VIN
- 10 TUPLES RESULTAT DE LA REQUETE

**INFORMATION SUR LE STOCKAGE**

1 PAGE DISQUE PEUT CONTENIR : - 5 BUVEURS OU 50 NOMS  
- 10 VINS OU 10 ABUS

TOUTES LES RELATIONS SONT TRIEES SUR CLE

COUT DU TRI D'UNE RELATION DE N PAGES =  $2N \log_2 N$

## Importance du choix du meilleur plan (3)

Page 23

© Philippe Pucheral, Luc Bouganim

**STRATEGIE 1:**

- EXECUTER LES OPERATIONS DANS L'ORDRE DE LA REQUETE
- EXECUTER LES JOINTURES PAR PRODUIT CARTESIEN

**1ère PHASE :** JOINTURES BUVEURS >< ABUS >< VINS

NOMBRE DE PAGES LUES  
 $(100/5) * (2000/10) * (500/10) = 20 * 200 * 50 = 200.000$  E/S

NOMBRE DE PAGES ECRITES  
2000 TUPLES RESULTAT = 800 PAGES = 800E/S

**2ème PHASE :** RESTRICTION PUIS PROJECTION

800 LECTURES + 1 ECRITURE = 801 E/S

**TOTAL > 200.000 E/S**

## Importance du choix du meilleur plan (4)

Page 24

© Philippe Pucheral, Luc Bouganim

**STRATEGIE 2:**

- EXECUTER LES OPERATIONS DANS L'ORDRE DE LA REQUETE
- EXECUTER LES JOINTURES PAR TRI-FUSION

**1ère PHASE :** JOINTURE (BUVEURS >< ABUS) → TEMP1

NOMBRE DE PAGES LUES  
 $(100/5) + (2000/10) = 20 + 200 = 220$  E/S

NOMBRE DE PAGES ECRITES  
2000 TUPLES RESULTAT = 670 PAGES = 670E/S

## Importance du choix du meilleur plan (5)

Page 25

© Philippe Pucheral, Luc Bouganim

2<sup>ème</sup> PHASE : TRI DE TEMP1 SUR NV

$$2 * (670 \log_2 670) = 2 * 670 * 10 = 13.400 \text{ E/S}$$

3<sup>ème</sup> PHASE : JOINTURE TEMP1 >< VINS

NOMBRE DE PAGES LUES

$$(670) + (500/10) = 720 \text{ E/S}$$

NOMBRE DE PAGES ECRITES

$$2000 \text{ TUPLES RESULTAT} = 800 \text{ PAGES} = 800 \text{ E/S}$$

4<sup>ème</sup> PHASE : RESTRICTION PUIS PROJECTION

$$800 \text{ LECTURES} + 1 \text{ ECRITURE} = 801 \text{ E/S}$$

**TOTAL = 16.611 E/S**

## Importance du choix du meilleur plan (6)

Page 26

© Philippe Pucheral, Luc Bouganim

**STRATEGIE 3:**

- EXECUTER LES RESTRICTIONS AVANT LES JOINTURES
- EXECUTER LES JOINTURES PAR PRODUIT CARTESIEN

1<sup>ère</sup> PHASE : RESTRICTION BUVEURS → TEMP1

NOMBRE DE PAGES LUES

$$(100/5) = 20 \text{ E/S}$$

NOMBRE DE PAGES ECRITES

$$5 \text{ TUPLES RESULTAT} = 1 \text{ PAGE} = 1 \text{ E/S}$$

2<sup>ème</sup> PHASE : RESTRICTION VINS → TEMP2

NOMBRE DE PAGES LUES

$$(500/10) = 50 \text{ E/S}$$

NOMBRE DE PAGES ECRITES

$$50 \text{ TUPLES RESULTAT} = 5 \text{ PAGES} = 5 \text{ E/S}$$

## Importance du choix du meilleur plan (7)

Page 27

© Philippe Pucheral, Luc Bouganim

3<sup>ème</sup> PHASE : JOINTURE TEMP1 >< ABUS >< TEMP2

NOMBRE DE PAGES LUES

$$1 * (2000/10) * 5 = 1.000 \text{ E/S}$$

NOMBRE DE PAGES ECRITES

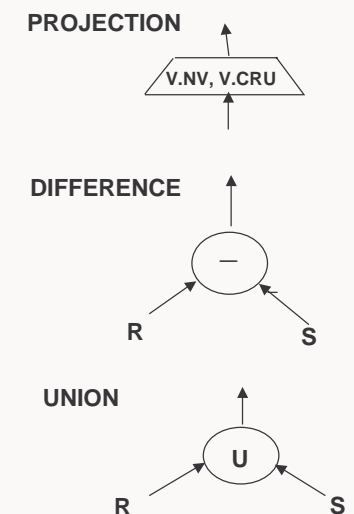
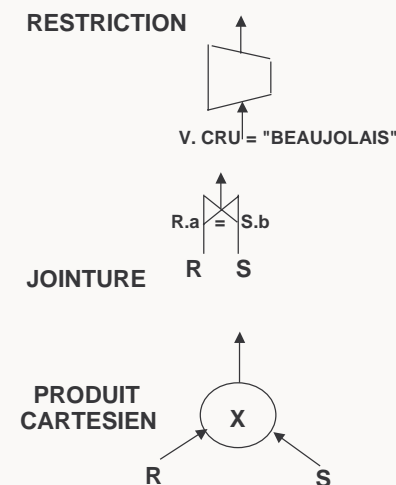
$$10 \text{ TUPLES RESULTAT} = 4 \text{ PAGES} = 4 \text{ E/S}$$

**TOTAL = 1080 E/S**

## ARBRES RELATIONNELS (étape III)

Page 28

© Philippe Pucheral, Luc Bouganim

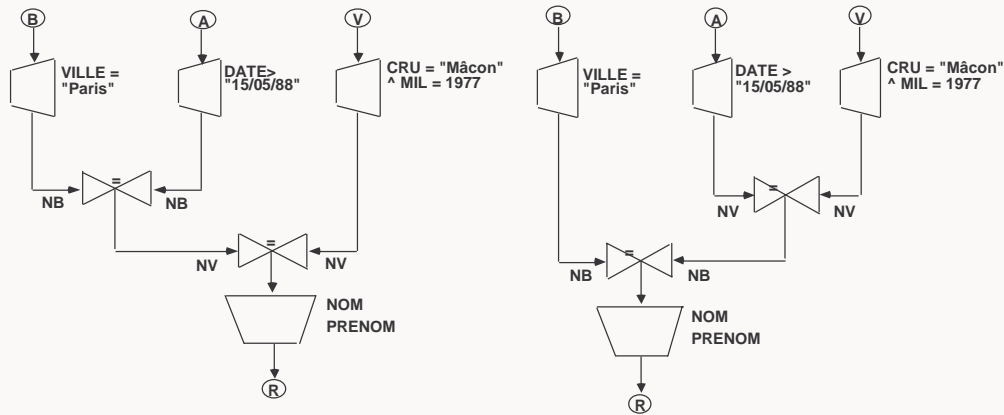


## Exemple (1)

Page 29

© Philippe Pucheral, Luc Bouganim

"Donner les noms et les prénoms des buveurs habitant Paris qui ont bu du Mâcon 1977 après le 15/05/88"

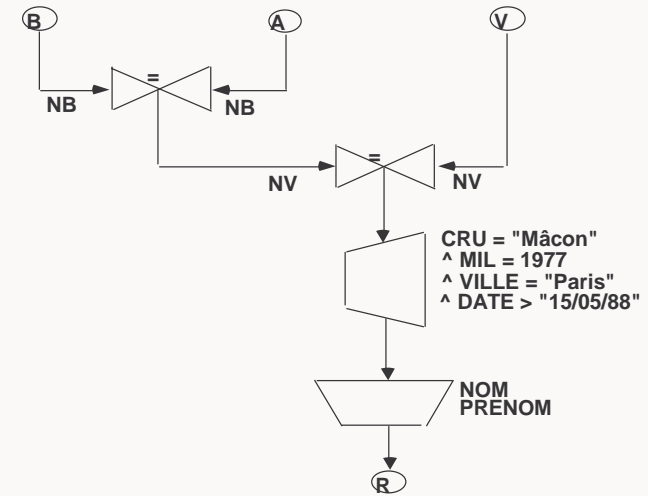


Attention, les arbres se présentent généralement dans l'autre sens !

## Exemple (2)

Page 30

© Philippe Pucheral, Luc Bouganim



De ces trois arbres, quel est le meilleur ??

## Restructuration algébrique

Page 31

© Philippe Pucheral, Luc Bouganim

### PROBLEME :

SUIVANT L'ORDRE DES OPERATEURS ALGEBRIQUES DANS UN ARBRE, LE COUT D'EXECUTION EST DIFFERENT

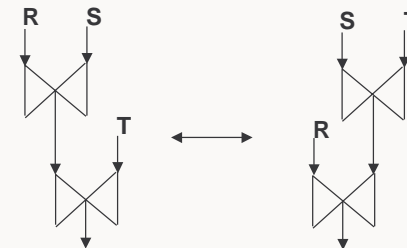
### POURQUOI ?

1. LE COUT DES OPERATEURS VARIENT EN FONCTION DU VOLUME DES DONNEES TRAITEES i.e., PLUS LE NOMBRE DE TUPLE DES RELATIONS TRAITEES EST PETIT, PLUS LES COUTS CPU ET D'E/S SONT MINIMISES
2. CERTAINS OPERATEURS DIMINUENT LE VOLUME DES DONNEES e.g., RESTRICTION ET PROJECTION

## Commutativité/Associativité des Jointures

Page 32

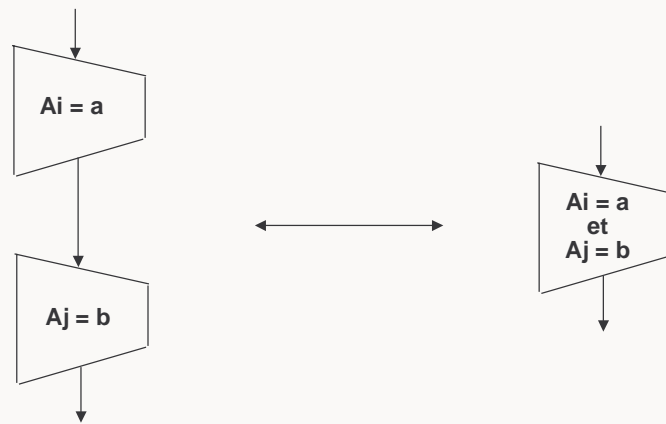
© Philippe Pucheral, Luc Bouganim



## Groupage des Restrictions

Page 33

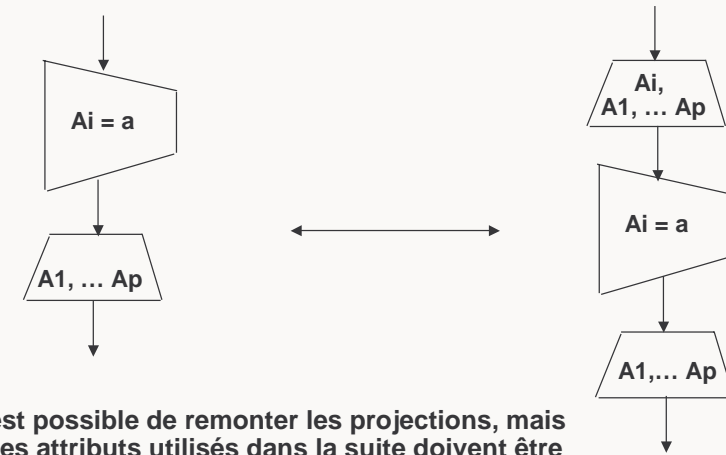
© Philippe Pucheral, Luc Bouganim



## Semi-commutativité des Projections / ...

Page 34

© Philippe Pucheral, Luc Bouganim



Il est possible de remonter les projections, mais les attributs utilisés dans la suite doivent être conservés !!!

## Règles de Restructuration

Page 35

© Philippe Pucheral, Luc Bouganim

- (1) COMMUTATIVITE DES JOINTURES
- (2) ASSOCIATIVITE DES JOINTURES
- (3) GROUPABILITE DES RESTRICTIONS
- (4) SEMI-COMMUTATIVITE DES PROJECTIONS ET RESTRICTIONS
- (5) DISTRIBUTIVITE DES RESTRICTIONS / JOINTURES
- (6) SEMI-DISTRIBUTIVITE DES PROJECTIONS / JOINTURES
- (7) DISTRIBUTIVITE DES RESTRICTIONS / UNIONS OU DIFFERENCES
- (8) DISTRIBUTIVITE DES PROJECTIONS / UNIONS

## Heuristique d'Optimisation (étape IV)

Page 36

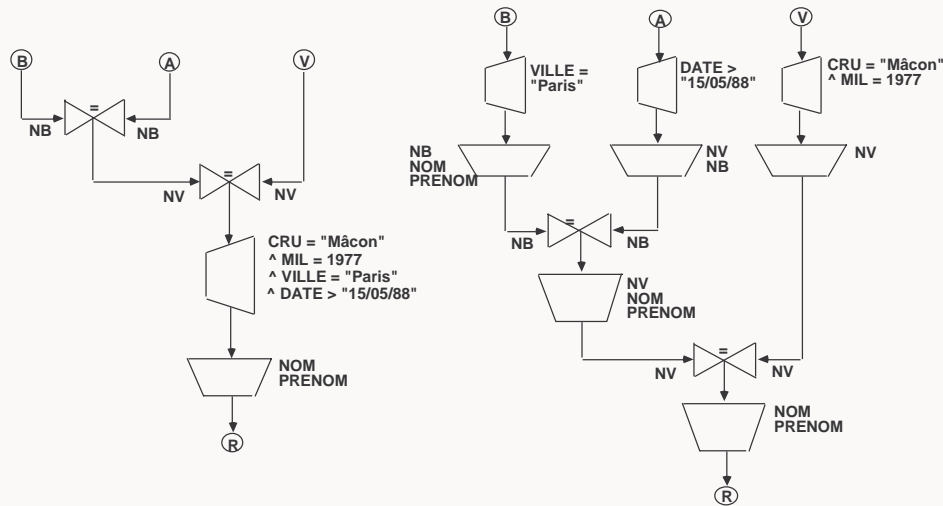
© Philippe Pucheral, Luc Bouganim

Appliquer d'abord les opérations réductrices (restrictions et projections) en les groupant sur chaque relation.

1. Dégroupier les restrictions (Règle 3')
2. Rapprocher les restrictions des feuilles (Règles 4, 5 et 7)
3. Grouper les restrictions aux feuilles (Règle 3)
4. Rapprocher les projections des feuilles (Règles 4, 6 et 8)

L'ordre des unions, différences et jointures reste inchangé !!!

# Exemple d'optimisation



# Problème de l'Ordonnancement

Il faut pouvoir ordonner jointures, union, différence, agrégat, ... en fonction des tailles des relations arguments

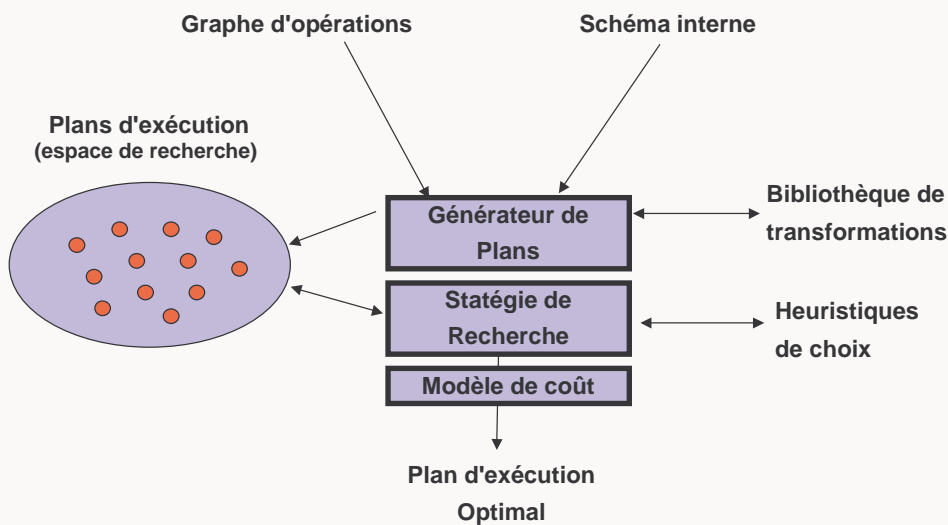
Il faut pouvoir prendre en compte les algorithmes par index afin de les favoriser (sélection, jointure sur index)

➔ Nécessité de développer un modèle de coût général permettant d'évaluer le coût d'un plan, c'est-à-dire d'un arbre annoté par des choix d'algorithmes.

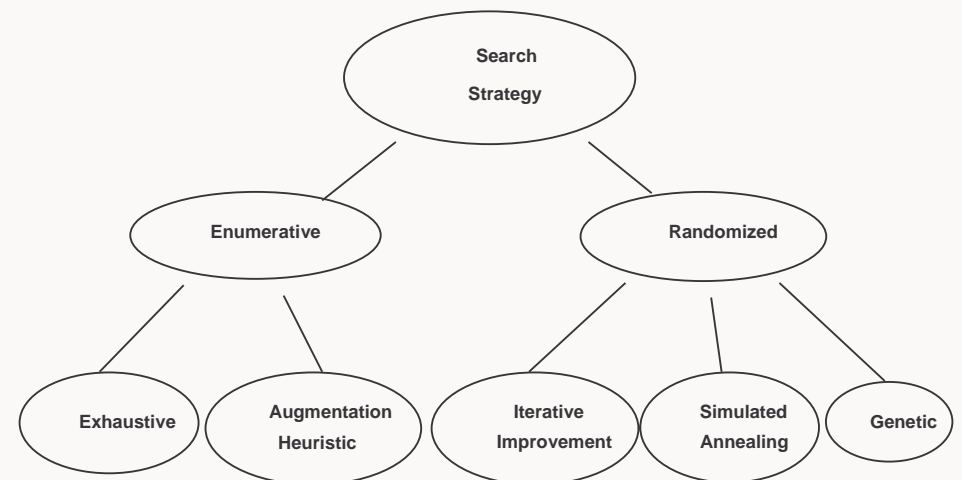
### Annotation:

Marque associée à un noeud indiquant l'algorithme à utiliser pour l'opérateur avec ses paramètres (index, hachage, ...)

# CHOIX DU MEILLEUR PLAN



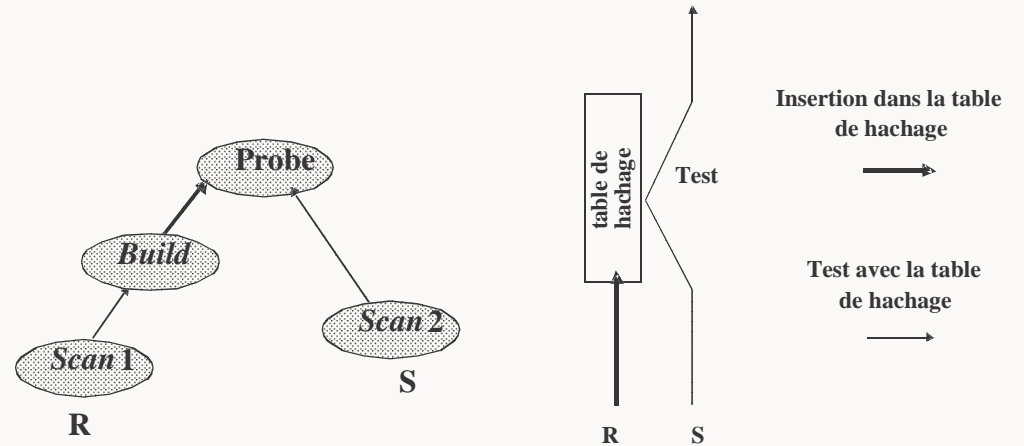
# Différentes Stratégies



# Espace de recherche

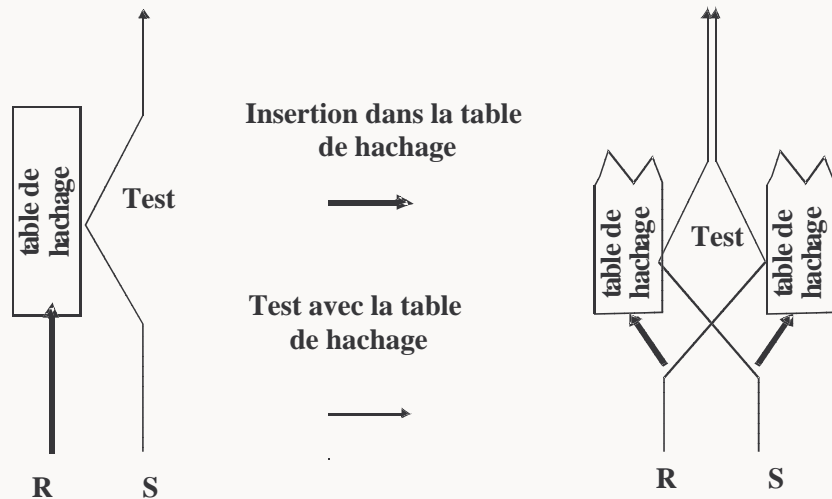
- Les opérations binaires ne sont pas symétriques (au niveau de l'exécution)
  - Notion de pipeline / matérialisation
  - Notion de relation externe / interne
  - Notion de lien bloquant / non bloquant
- ➔ Différentes formes d'arbres sont possibles. Influence sur
  - ✓ La consommation mémoire
  - ✓ Le mode d'exécution
  - ✓ et donc les performances

# Jointure par hachage des relations R et S



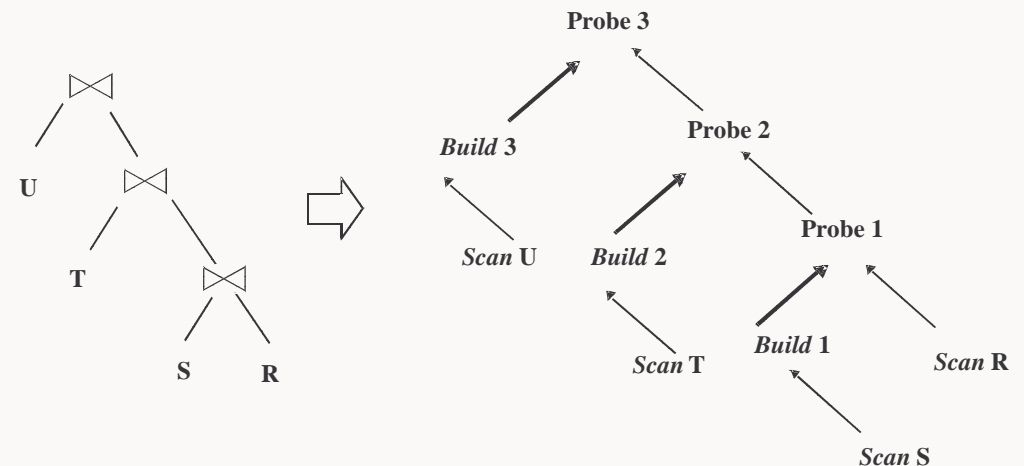
Décrire l'algo par hachage....

# Pipeline simple / Pipeline non bloquant



Applications ??

# Arbre de type linéaire droit (right deep tree)

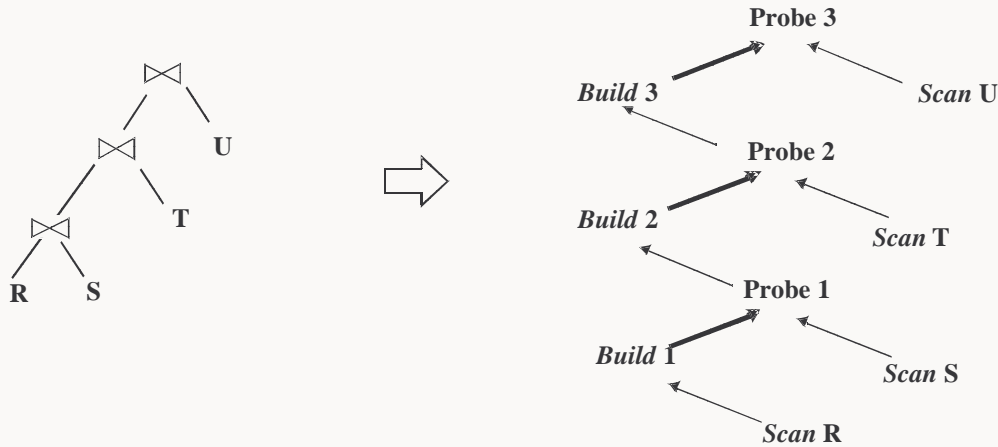


Calculer la taille mémoire nécessaire

## Arbre de type linéaire gauche (left deep tree)

Page 45

© Philippe Pucheral, Luc Bouganim

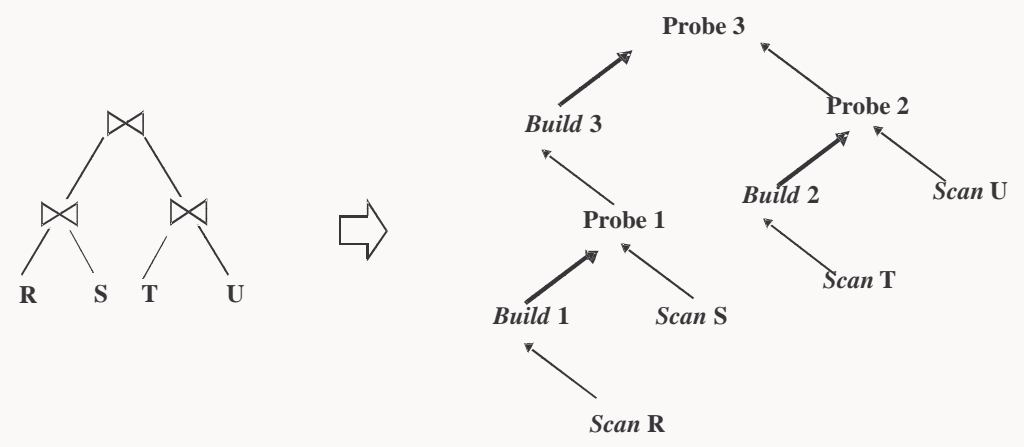


Calculer la taille mémoire nécessaire

## Arbre de type bushy (bushy tree)

Page 46

© Philippe Pucheral, Luc Bouganim



Calculer la taille mémoire nécessaire

## Ordre de grandeur

Page 47

© Philippe Pucheral, Luc Bouganim

- Jointure de 5 relations: (avec produits cartésiens)
  - 120 arbres linéaires différents
  - 1620 arbres bushy différents
- Jointure de 10 relations: (avec produits cartésiens)
  - 3 628 800 arbres linéaires différents
  - 17 643 225 600 arbres bushy différents !!!

## Optimisation physique

Page 48

© Philippe Pucheral, Luc Bouganim

- Choix des meilleurs algorithmes pour les opérations relationnelles
  - Utilisation des index
  - jointure par index, nested loop, etc...
- En fonction de
  - Structures de données existantes
  - Statistiques sur les relations
  - Algorithmes existant

## Modèle de coût

Page 49

© Philippe Pucheral, Luc Bouganim

- Paramètres d'entrée
  - machine (puissance, disques, mémoire, réseau, etc..)
  - Arbre d'exécution
  - Algorithmes relationnels
  - Schéma de la base
  - Statistiques sur les relations
  - Taille, Domaine, Nb valeurs distinctes, Répartition, Histogrammes
- Traitement
  - Evaluation de la taille des résultats intermédiaires
  - Evaluation du coût
- Sortie
  - Un coût en termes d'I/O, CPU, etc... ou un coût global

## Le calcul des tailles

Page 50

© Philippe Pucheral, Luc Bouganim

- Taille des tables de base dans le catalogue
- Calcul des tailles à la compilation
  - application du coefficient de sélectivité
  - hypothèse d'uniformité
- Possibilité d'histogrammes
  - RunStat(<Table>, <attribut>)
  - Stockage dans le catalogue de l'histogramme de distribution de l'attribut
  - Utilisation par le modèle de coût



Sélectivité de prédicats simples, complexes, de jointure (différents cas)

## Jointures par boucles imbriquées (coût I/O)

Page 51

© Philippe Pucheral, Luc Bouganim

- Hypothèse 1: Mémoire ( $|m|$ ) > Plus petite relation ( $|R|$ ) + 2
  - Lire R  $\Rightarrow |R|$  I/O
  - Lire page à page S et joindre  $\Rightarrow |S|$  I/O
  - Total :  $|R| + |S|$  I/O
- Hypothèse 2: Mémoire ( $|m|$ ) < Plus petite relation ( $|R|$ )
  - Lire  $|M| - 2$  pages de R
  - Lire page à page S et joindre  $\Rightarrow |S|$  I/O
  - Recommencer jusqu'à fin de R
  - Total :  $|R| / (|m| - 2) \times |S| + |R|$  I/O
- Hypothèse 3: Mémoire = 3 pages !!!
  - Lire page à page R
  - Lire page à page S et joindre  $\Rightarrow |S|$  I/O
  - Total :  $|R| \times |S| + |R|$  I/O

## Jointure avec index/ tri-fusion

Page 52

© Philippe Pucheral, Luc Bouganim

- Jointure avec index
  - Si il existe un index sur l'attribut de jointure et SI LA RELATION INDEXEE TIENS EN MEMOIRE !!!
    - ✓ Lire une page de R
    - ✓ Parcourir l'index et trouver l(es) attributs joignant
- Jointures par tri-fusion
  - ✓ Trier les 2 relations
  - ✓ Joindre avec 2 pointeurs en parallèle...

## Semi-jointures

Page 53

© Philippe Pucheral, Luc Bouganim

- **Principe:**
  - On projette R sur l'attribut de jointure (sans doublons) dans R'
  - On lit S et on ne retiens que les tuples existant dans R'
  - On joint les tuples résultat avec R
- **Intérêt ?**

## Optimisation dynamique

Page 54

© Philippe Pucheral, Luc Bouganim

- **Normalement l'optimisation se fait à la compilation.**
  - Le coût n'est pas imputé à l'utilisateur
  - Optimisation des requêtes fréquemment posées
  - On peut 'prendre son temps' (dans une certaine mesure)
- **L'optimisation dynamique propose de compléter l'optimisation statique**
  - juste avant l'exécution
  - pendant l'exécution

## Optimisation parallèle

Page 55

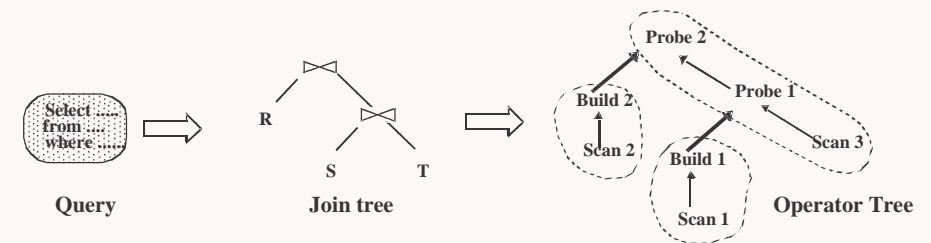
© Philippe Pucheral, Luc Bouganim

- **Stratégie 'two phase optimization'**
  - hypothèse: Le meilleur plan parallèle est une parallélisation du meilleur plan séquentiel
  - simple
- **Stratégie 'one phase optimization'**
  - On produit un espace de recherche contenant des plans parallèles
  - Explosion de l'espace de recherche
  - Complexe (modèle de coût)
- **Problèmes**
  - Localisation des relations → Localisation des opérateurs
  - Degré de parallélisme ?
  - Scheduling ??

## Optimisation parallèle

Page 56

© Philippe Pucheral, Luc Bouganim



### Operator Scheduling

Hash constraints : Build1<Probe1, Build2<Probe2  
Heuristic 1: Build2<Scan3, Build1<Scan3  
Heuristic 2: Build2<Scan1

### Operator homes

home(Scan1, Build1) = Node A  
home(Scan2, Build2) = Node B  
home(Scan3, Probe 1, Probe2) = Node A,B

## Optimisation distribuée

Page 57

© Philippe Pucheral, Luc Bouganim

- **Même type de problématique que pour le parallélisme....**
  - Arbre d'opérateur
  - Localisation des opérateurs
  - Scheduling des opérateurs
- **Dans un contexte différent**
  - Coût réseau importants
  - Duplications fréquente
  - Fragmentation verticale et horizontale
- **Techniques complexes (voir Ozsu/Valduriez)**

## Prise en compte de fonctions

Page 58

© Philippe Pucheral, Luc Bouganim

- **Modélisation des fonctions**
- **Remet en cause certains principes et heuristiques**
- **Nécessite de nouveaux opérateurs**
- **Si les fonctions sont 'coûteuses', nécessite de prendre en compte de nouvelles techniques**
  - Cache
  - Parallélisme
- **Problème 'ouvert'...**

---

## Etude de cas

**Exécution distribuée de requêtes SQL incluant des fonctions 'chères' et manipulant des objets volumineux (BLOB's)**

## Etude de cas

Page 60

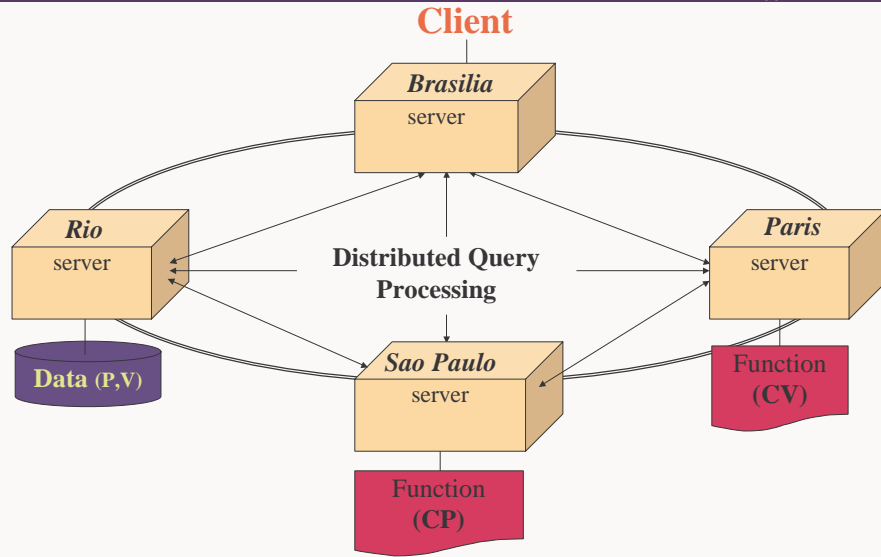
© Philippe Pucheral, Luc Bouganim

- **But :**
  - Appliquer les notions introduites sur un cas 'original' dans un contexte différent...
- **Contexte :**
  - Exécution distribuée de requêtes SQL incluant des fonctions 'chères' et manipulant des objets volumineux (BLOB's)
- **Problèmes :**
  - Optimisation de ces requêtes ???
  - Exécution efficace ???

## Exemple : Architecture

Page 61

© Philippe Pucheral, Luc Bouganim



## Exemple : Scenario

Page 62

© Philippe Pucheral, Luc Bouganim

- **Site de Rio:**
  - Table P contenant les pollutions mesurées :  
 $P(\text{regld:int}, \text{date:int}, \text{value:double}, \dots)$
  - Table V contenant des images (raster) de regions  
 $V(\text{regld:int}, \text{image:Blob}, \dots)$
- **Site de Sao Paulo:**
  - Fonction CP permettant de calculer des indices de pollution en fonction des données brutes :  
 $\text{function CP}(\text{double}) \rightarrow \text{double}$
- **Site de Paris:**
  - Fonction CV permettant de calculer la couverture végétale sur une image raster.  
 $\text{function CV}(\text{Blob}) \rightarrow \text{double}$

## Exemple : Requête

Page 63

© Philippe Pucheral, Luc Bouganim

```

Select  P.regld, CP(P.value), CV(V.image)
From    P, V
Where   P.regld = V.regld
and     CP(P.value) > 1.5
and     CV(V.image) < 0.3
    
```

## Exemple : Paramètres

Page 64

© Philippe Pucheral, Luc Bouganim

Name	Description	Value
CardP	Cardinality of relation P	300 tuples
DistP	Number of distinct pollution measurements in P	150 measurements
CardV	Cardinality of relation V	50 tuples
DistV	Number of distinct images in V	50 images
CardV $\infty$ P	Cardinality of the result of V join P	200 tuples
DistV $\infty$ P	Number of distinct pollution measurements in V join P	100 measurements
DistV $\infty$ P	Number of distinct images in V join P	40 images
ImgTrans	Average image transfer time (with a 1Mb/s network bandwidth)	100 s
CostCP	Average per tuple cost of function CP	30 s
CostCV	Average per tuple cost of function CV	200 s
$\sigma_{pp}$	Average selectivity for predicate $p_p$ ( $CP(P.value) > 1.5$ )	0.5
$\sigma_{pv}$	Average selectivity for predicate $p_v$ ( $CV(V.image) < 0.3$ )	0.8