

IMPLÉMENTATION DES OPÉRATEURS RELATIONNELS ET OPTIMISATION

Les opérateurs étudiés

On regarde d'abord les opérateurs pris un à un

- La sélection : σ
- La projection : π
- La jointure : \bowtie

ensuite, on s'intéressera aux requêtes composées. Par exemple, on verra qu'il vaut mieux, en général, faire

$$\sigma_{A=cte}(R) \bowtie N$$

que

$$\sigma_{A=cte}(R \bowtie N)$$

Marin(Num_M, Nom_M, Niv_M, Age_M)

Réservation(Num_M, Num_B, Jour, Nom_Pers)

Bateau(Num_B, Nom_B, Cat_B)

Avec les instances respectives M, R et B.

- R : Tuple=40octets, $T_R = 100$ tuples par page, $[R] = 1000$ pages au total.
- M : Tuple=50octets, $T_M = 80$ tuples par page, $[M] = 500$ pages au total.

Jointure avec égalité d'une colonne

```
SELECT *  
FROM Résevation R1, Marin M1  
WHERE R1.Num_M = M1.Num_M
```

- C'est équivalent à un produit cartésien suivi d'une sélection. Mais ce n'est pas efficace.
- La taille du produit est de l'ordre de $[R] * [M]$ (500000). Le coût total est : $[R] + [M] + [R]*[M]$

Pour estimer le coût d'une opération, nous allons considérer le nombre d'E/S en ne tenant pas compte du coût de l'écriture du résultat.

Simple boucle imbriquée

On suppose que la jointure se fait sur l'égalité des attributs A et B de R et M (respectivement)

Pour chaque tuple r dans R faire

Pour chaque tuple m dans M faire

si $r.A = m.B$ alors ajouter $\langle r, m \rangle$ au résultat

- Coût : $[R] + [R] * T_R * [M] = 1000 + 1000 * 100 * 500$
 $= 50001000 \approx 5 * 10^7$
- Si Marin est la relation externe dans la boucle, alors le coût serait : $[M] + [M] * T_M * [R] = 500 + 500 * 80 * 1000$
 $= 40000500 \approx 4 * 10^7$

Boucle imbriquée sur les pages

Pour chaque page P de R

Pour chaque page Q de M

Pour chaque tuple r dans P

Pour chaque tuple m dans M

Si $r.A=m.B$ alors

ajouter $\langle r, m \rangle$ au résultat

Coût = ?

Pour chaque page de R , on va lire toutes les pages de M . Le coût total sera donc

$$[R] + [R] * [M] = 1000 + 1000 * 500 = 501000 \approx 5 * 10^5$$

Si c'est M qui est la relation externe dans la boucle alors le coût sera $[M] + [M] * [R] = 500 + 1000 * 500 = 500500 \approx 5 * 10^5$

Boucle imbriquée utilisant le tampon

On suppose que le tampon peut contenir B pages.

Pour chaque page $B-2$ pages de R

Pour chaque page Q de M

Pour chaque tuple r de R au tampon

Pour chaque tuple m dans M

Si $r.A=m.B$ alors

ajouter $\langle r, m \rangle$ au résultat

Coût= ?

Ici, on réserve 1 zone tampon pour l'écriture du résultat

Pour chaque groupes de $B - 2$ pages de R , on lit toutes les pages de M . Le nombre de groupes est $[R]/(B - 2)$

Si $B = 102$ alors le coût est $[R] + \left([R]/(B - 2) \right) * [M] =$
 $1000 + (1000/100) * 500 = 6000 = 6 * 10^3$

Boucle imbriquée avec Index

Pour chaque tuple r dans R faire

*Pour chaque tuple m dans M où $r.A = m.B$ faire
ajouter $\langle r, m \rangle$ au résultat*

- Si M est indexée, alors le coût est : $[R] + [R] * T_R * \text{coût}$ pour trouver les “bons” tuples dans M
- Si M a un index en arbre B+, l'index aura 3 à 5 niveaux puis au maximum 1 lecture par tuple trouvé (cela dépend du fait que l'index soit primaire ou secondaire)

Jointure avec Tri-Fusion

SORT-MERGE JOIN

Trier R et M

$r \leftarrow$ premier tuple de R, $m \leftarrow$ premier tuple de M

$G_m \leftarrow$ premier tuple de M

Tant que ni fin(R) ni Fin(M) Faire

Tant que $r.A < m.B$ Faire

$r \leftarrow$ tuple suivant dans R

Tant que $m.B < r.A$ Faire

$m \leftarrow$ tuple suivant dans M

Tant que $m.B = r.A$ Faire

$G_m \leftarrow m$

Tant que $r.A = m.B$ Faire

ajouter $\langle r, m \rangle$

$m \leftarrow$ tuple suivant dans M

$r \leftarrow$ tuple suivant dans R

$m \leftarrow G_m$

Fin

Jointure avec Tri-Fusion

R est scannée une seule fois.

Chaque groupe de tuples de M joignables est scanné autant de fois qu'il y a de tuples dans R avec qui ils sont joignables. (De fortes chances, que les pages se trouvent déjà dans le tampon).

Le coût : Le tri de R est de l'ordre $2 * [R] \log_B [R]$ où B est le nombre de zones tampons. En effet, le tri de R a un coût de

$$2 * [R] * \left(\log_{B-1}([R]/B) + 1 \right)$$

On peut donc estimer le coût total à

$$2([R] \log_B [R] + [M] \log_B [M]) + [R] + [M]$$

En considérant $B = 102$ on obtient

$$2(1000 \log_{102}(1000) + 500 \log_{102}(500)) + 1000 + 500 = 5832$$

(cf. la jointure Boucle+ Tampon)

On considère les sélections de la forme

```
SELECT *
```

```
 $\sigma_{Att \text{ op valeur}}(R)$  FROM Reservation R
```

```
WHERE R.Nom_Pers < "C%"
```

- S'il n'y pas d'index et si fichier non trié, coût=[R]=1000
- En utilisant un index, le coût dépend du nombre de tuples retournés et du fait que l'index soit primaire ou non.
- Exemple : un arbre B+, 10% des 100000 tuples sont retournés.
- Le cas primaire : Un peu plus de 100 E/S (10000 tuples résidant dans 100 pages + parcours de l'arbre)
- Le cas non groupé 10000 E/S ! (les 10000 tuples se trouvent éparpillés)

Les marins ayant réservé le bateau 5 après le 8/9/98.

$a \equiv jour > 8/9/94$

$b \equiv Num_B = 5$

$\sigma_{a \wedge b}(R)$

- Si on a un index sur $\langle Jour, Num_B \rangle$ alors on peut l'utiliser. De même qu'un index $\langle Num_B, Jour \rangle$.
- Si l'on a deux index, il vaut mieux commencer par le chemin d'accès le plus sélectif.
- Noter qu'un index sur $\langle Nom_Pers, Jour, Num_B \rangle$ n'est d'aucun secours ici

```
SELECT DISTINCT  R.Num_N, R.Num_B  
FROM            Reservation R
```

Utiliser le tri externe

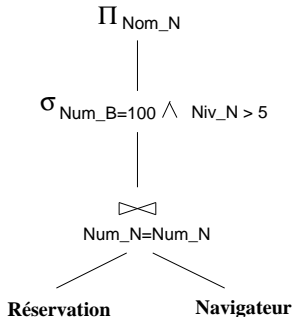
- 1 Modifier la première passe en éliminant les attributs non désirés. En sortie, on a le même nombre de tuples mais avec des tailles plus petites.
Dans l'exemple, si la taille des nouveaux tuples est de $\frac{1}{4}$, alors en sortie de cette première phase, on a 250 pages au lieu de 1000.
- 2 Modifier les phases de fusion de sorte à éliminer les doublons. A chaque phase, en sortie on a moins de tuples.

S'il y a un index sur tous les attributs recherchés, alors parcourir l'index.

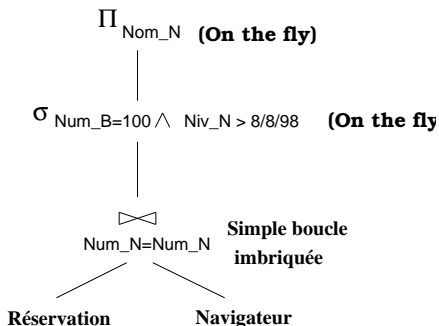
Optimisation

Une requête est une succession d'opérations. Pour une même requête, il y a plusieurs **plans** équivalents.

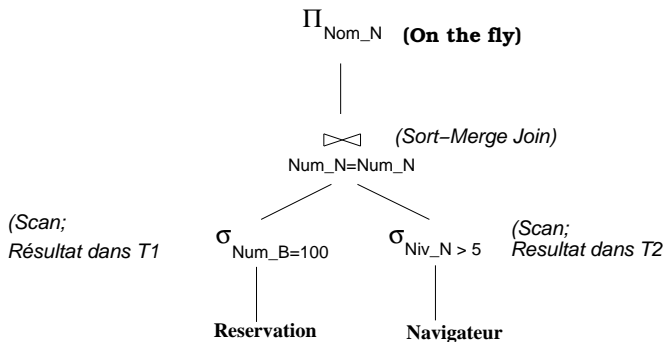
```
SELECT  N.Nom_N
FROM    Navigateur N, Resevation R
WHERE   N.Num_N = R.Num_N AND
        N.Niv_N > 5 AND
        R.Num_B = 100
```



L'idéal : trouver le meilleur plan, en pratique : éviter le pire des plans.



Coût : Ici, il n'y a que la jointure qui nécessite des accès disque.
Donc le coût est égal à celui de la jointure.



La sélection est exécutée au plus tôt.

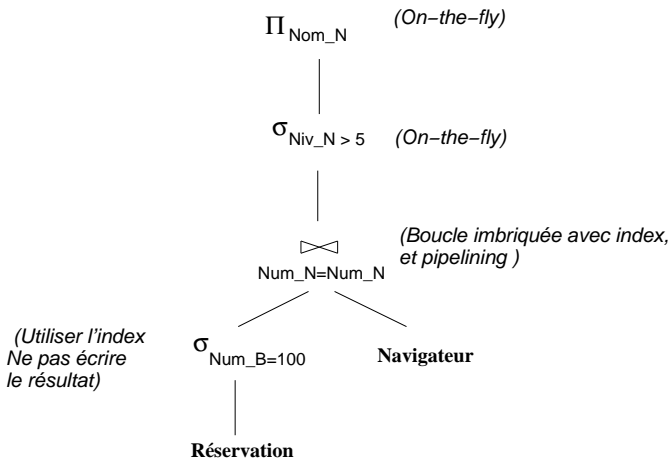
Avec 5 tampons :

- Scan de R = 1000 et écriture de T1 = CT1
- Scan de N = 500 et écriture de T2 = CT2
- Supposons que CT1 = 10 et CT2 = 250. Alors, Tri de T1=2*10, Tri de T2=3*250 et la fusion est = 10 + 250.

Total = 2530

Le coût de CT1 suppose une “ distribution uniforme” et un nombre de bateaux égal à 100 (autant de réservations pour chaque bateau). Pour CT2, on a fait la même hypothèse avec des niveaux dans $[1 \dots 10]$

3ème plan



Avec un index *primaire* sur Num_B et une distribution uniforme, on lit 10 pages dans Réservation (+ un nombre de pages = hauteur de l'arbre). Chaque tuple r du résultat est tout de suite joint avec "Marin" (au max avec un tuple)

Avec un index haché (sans débordement) sur Num_M (la clé) de Marin et comme il y a 1000 tuples à joindre, on fait au max 1000 accès.

Ce qui donne un coût total estimé à **1010**

Paramètres de choix du “meilleur plan”

- Estimer le coût de chaque opération dans le plan
- Estimer la taille des résultats intermédiaires
- Les catalogues contiennent des infos sur
 - # tuples et # pages pour chaque relation
 - Les différents index pour chaque relation
 - # valeurs de clés d'indexation et # pages pour chaque index
 - hauteur de chaque index, les valeurs Max et Min de la clé d'indexation*Mis à jour périodiquement.*

Hypothèses généralement utilisées

SELECT une liste d'attributs
FROM une liste de relations
WHERE *terme1* AND *terme2* ... AND *termeK*

- Le # Max de tuples = le produit du # tuples de chaque relation
- Le nombre de valeurs pour l'attribut A dans la relation R est noté $V_R(A)$
- Un facteur de réduction FR est associé à chaque terme de la clause WHERE
 - $R.A = Val$ a un $FR = 1/V_R(A)$
 - $R.A = S.B$ a un $FR = 1/Max(V_R(A), V_S(B))$
 - $A > Val$ a un $FR = \frac{Max_A(R) - Val}{Max_A(R) - Min_A(R)}$
- La cardinalité du résultat = # Max * $FR_1 * \dots * FR_K$