TD3 - Optimisation de requêtes

- Parcours séquentiel d'une table R :

$$P(R) = \frac{card(R)}{\underbrace{T_{page} / largeur(R)}_{\text{nombre de n-uplets de R dans une page.}}} , \quad T_{page} : \text{taille d'une page en octets}$$

$$\underbrace{coût(R) = P(R) \times c}_{\text{lecture séquentielle (Table Access FULL)}} \quad \text{où} \begin{cases} c < 1 & \text{si pages contigues } (\approx 0{,}27 \text{ oracle}) \\ c = 1 & \text{sinon} \end{cases}$$

- Sélection σ :
  - si E expression composé : $coût(\sigma_{p(A)}(E)) = coût(E)$
  - si T table et A pas indexé : $coût(\sigma_{p(A)}(T)) = P(T)$

- Sélection index non plaçant :

$$coût(\sigma_{p(A)}(R)) = C_{index} + C_{rowid} + card(\sigma_{p(A)}(R)) \times \frac{CF}{card(R)}$$

où : $C_{index} = \begin{cases} 0 & \text{si l'index tient en mémoire} \\ \text{hauteur de l'arbre} - 1 & \text{sinon} \end{cases}$

$$C_{rowid} = \begin{cases} 0 & \text{index unique scan} \\ \left\lceil \frac{card(\sigma_{p(A)}(R))}{card(R)} \times nb\ pages \right\rceil & \text{index range scan} \end{cases}$$

$CF = \frac{card(R)}{N/P}$ où N rowid font référence à P pages à lire

↳ clustering factor

cas général : $coût(\sigma_{p(A)}(R)) = card(\sigma_{p(A)}(R))$

- Sélection index plaçant

$$coût(\sigma_{p(A)}(R)) = \lceil P(R) \times SF(p(A)) \rceil$$

$$= P(R) \times \frac{card(\sigma_{p(A)}(R))}{card(R)}$$

- Jointure par boucles imbriquées

→ index : $coût(R \bowtie_{r.a = s.a} S) = coût(R) + card(R) \times coût(\sigma_{a=v}(S))$

$$\text{avec } coût(\sigma_{a=v}(S)) = \begin{cases} 1 & \text{si a est une clé de S} \\ C_{rowid} + card(\sigma_{a=v}(S)) \times \frac{CF}{card(S)} & \text{si index non plaçant} \\ \lceil P(S) \times SF(a=v) \rceil & \text{si index plaçant} \end{cases}$$

→ table sans index : $coût(R \bowtie S) = coût(R) + P(R) \times P(S)$

→ si S est une expression : $coût(R \bowtie S) = coût(S) + P(S) + coût(R) + P(R) \times P(S)$

- Jointure par hachage : hypothèse $P(R) \geq P(S)$

$$\text{coût}(R \bowtie S) = \text{coût}(S) + \text{coût}(R)$$

cas hachage externe (S ne tient pas en mémoire)

$$\text{coût}(R \bowtie S) = 2\lfloor \log_k(P(S)) \rfloor (P(R) + P(S)) + P(R) + P(S)$$

avec taille mémoire = $k+1$ pages.

- Tri externe (merge-sort)
- E ≠ table : $\text{coût}(\text{tri}(E)) = \text{coût}(E) + 2 \times P(E) \times (s-1)$
- $\text{coût}(\text{tri}(R)) = 2 \cdot P(R)(s-1) + P(R)$ si on ne matérialise pas le résultat (on lit seulement)
- $\text{coût}(\text{tri}(R)) = 2 \cdot P(R) \cdot s$ coût total

avec $s = \lceil \log_k(P(R)) \rceil$, s tq $k^s \geq P(R)$.

- cardinalité d'une sélection :

$$\text{cand}(\sigma_{p(A)}(R)) = SF(p(A)) \times \text{cand}(R)$$

→ si égalité : $SF(p(A)) = \dfrac{\text{nbval}}{D(R,A)}$ ← nombre valeurs distinctes $= \pi_A(R)$

→ si inégalité : $SF(p(A)) = \dfrac{\text{longueur du segment}}{L(A)} = \dfrac{\text{longueur du segment}}{\max(A) - \min(A)}$

où borne : $A \in [\min(A), \max(A)]$

$$\text{longueur du segment} = \begin{cases} v - \min(A) & \text{si } A < v \\ \max(A) - v & \text{si } A > v \\ v_2 - v_1 & \text{si } A \geq v_1 \text{ and } A \leq v_2 \\ \max(A) - v_2 + v_1 - \min(A) & \text{si } A < v_1 \text{ or } A \geq v_2 \end{cases}$$

- sélection sur plusieurs attributs
- $SF(p(A) \text{ and } p(B)) = SF(p(A)) \times SF(p(B))$
- $SF(p(A) \text{ or } p(B)) = SF(p(A)) + SF(p(B)) - SF(p(A)) \times SF(p(B))$ il faut l'enlever
- $SF(\text{not } p(A)) = 1 - SF(p(A))$

- cardinalité sur les opérations :
- $\text{cand}(\pi_A(R)) = \text{cand}(R)$
- $\text{cand}(R \times S) = \text{cand}(R) \times \text{cand}(S)$
- $\text{cand}(R \cup S) \in [\max(\text{cand}(R), \text{cand}(S)), \text{cand}(R) + \text{cand}(S)]$
- $\text{cand}(R - S) \in [0, \text{cand}(R)]$

- cardinalité jointure

table : $\text{cand}(R \bowtie_A S) = \text{cand}(S) \times \underbrace{\text{cand}(R) \times \dfrac{1}{D(\text{entité}, A)}}_{=1 \text{ si jointure naturelle}}$

entité : table avec primary key

entre 2 sélections :

$$\text{cand}(\sigma_{p(A)}(R) \bowtie_A \sigma_{p(B)}(S)) = SF(p(A)) \times SF(p(B)) \times \text{cand}(R \bowtie_A S)$$