

Nom :	Prénom :	<i>page 1</i>
--------------	-----------------	---------------

Université Pierre et Marie Curie - Paris 6 - UFR 922 - Master d'informatique (SAR)

Module Bases de Données Réparties

Partiel du 5 avril 2005

Les documents ne sont pas autorisés – Durée : 2h.

Répondre aux questions sur la feuille du sujet dans les cadres appropriés. La taille des cadres suggère celle de la réponse attendue. Utiliser le dos de la feuille précédente si la réponse déborde du cadre. Le barème est donné à titre indicatif. La qualité de la rédaction sera prise en compte. Ecrire à l'encre bleue ou noire. Ne pas dégrafer le sujet.

Exercice 1 : Questions diverses

4 pts

Question 1. Pour optimiser le traitement des requêtes, on utilise parfois la technique du pipeline. Expliquez en quoi consiste cette technique. Quels en sont les avantages ?

Question 2. Donnez un exemple où on ne peut pas utiliser la technique du pipeline. Expliquez pourquoi.

Question 3 Expliquez en quoi l'architecture sans partage est intéressante pour les bases de données parallèles ?

Question 4. Comparez les avantages et inconvénients de l'architecture sans partage avec l'architecture à mémoire partagée.

Exercice 2 : Optimisation de requêtes**10 pts**

Soit le schéma suivant :

Emp (<u>ne</u> , salaire, age, ns)	-- un employé est identifié par son numéro <i>ne</i>
Service (<u>ns</u> , <u>np</u> , budget, statut)	-- <i>ns</i> fait référence à un numéro de service.
Projet (<u>np</u> , code, libellé)	-- un projet est identifié par son numéro <i>np</i>

La taille d'un n-uplet est de 20 octets pour Emp, 40 octets pour Service, 2000 octets pour Projet. Les attributs *ne*, *ns* et *np* ont chacun 4 octets. La cardinalité des relations est de 20 000 pour Employés, 5000 pour Service et 1000 pour Projet. La relation Service représente l'**association N-M** d'un service avec un projet. La clé de Service est composée des attributs *ns* et *np* (ainsi, l'attribut *ns* n'est pas unique dans Service).

Chaque Service, identifié par *ns* a en moyenne 10 Projets. Les données sont stockées sur disque dans des pages de 4000 octets. Les attributs sont indépendants et leur distribution est uniforme.

Soient les fonctions auxiliaires :

- $ntp(R)$ le nombre de tuples par pages pour la relation *R*,
- $D(R, c)$ le nombre de valeurs distinctes du domaine de l'attribut *R.c*,
- $largeur(R)$ la taille d'un tuple de *R*,
- et $arr(x)$ l'arrondi de *x* par excès à une valeur entière,

Un index sur l'attribut *c* est dit *plaçant* si les données sont **triées** sur le disque dans l'ordre des valeurs de *c*. Un index sur l'attribut *c* est dit *non plaçant* si les données ne sont **pas triées** sur le disque dans l'ordre de *c*.

L'estimation du coût des opérations repose sur le modèle suivant. Le modèle de coût estime le nombre de pages à lire et écrire, sans prendre en compte l'écriture du résultat final.

- Le coût d'une lecture séquentielle de la relation *R* est égal au nombre de pages de *R* (noté $page(R)$).

Le coût d'une sélection avec un prédicat *pred* de la forme $a \text{ op } v$ où *a* est un attribut numérique et *op* est l'opérateur = (égal), < (inférieur à) ou > (supérieur à), est :

$\text{coût}(\sigma_{pred}(R)) = \text{card}(\sigma_{pred}(R))$ si l'attribut *a* est indexé par un index non plaçant,

$\text{coût}(\sigma_{pred}(R)) = \text{arr}(page(R) * \text{card}(\sigma_{pred}(R)) / \text{card}(R))$ si l'attribut *a* est indexé par un index plaçant,

$\text{coût}(\sigma_{pred}(R)) = page(R)$ si l'attribut *a* n'est pas indexé.

- Le coût d'une jointure naturelle avec un prédicat de jointure *pred* de la forme $R.c = S.c$ dépend de l'algorithme utilisé et de la présence d'index sur les attributs de jointure. Les algorithmes de jointure sont numérotés J1 à J9 :

- J1: jointure par boucles imbriquées, l'itération sur *R* imbrique l'itération sur *S*, sans utiliser aucun index.
- J2: jointure avec itération sur *R* et accès par index plaçant sur *S.c*
- J3: jointure avec itération sur *R* et accès par index **non** plaçant sur *S.c*
- J4 : jointure par tri de *R* et *S* selon *c*, puis fusion sans utiliser aucun index
- J5 : jointure par fusion, avec des index plaçants sur *R.c* et sur *S.c*
- J6 : jointure par fusion, avec des index **non** plaçants sur *R.c* et sur *S.c*
- J7 : jointure par tri de *R* selon *c*, puis fusion en utilisant un index plaçant sur *S.c*
- J8 : jointure par tri de *R* selon *c*, puis fusion en utilisant un index **non** plaçant sur *S.c*

- J9 : création d'une table de hachage (non plaçante) de R sur c , puis jointure avec itération sur S et accès à R par la table de hachage.

On propose 9 formules de coût :

$$C1: \text{coût}(R \bowtie_{\text{pred}} S) = \text{page}(R) + \text{card}(R) \times \text{card}(S) / D(S, c)$$

$$C2: \text{coût}(R \bowtie_{\text{pred}} S) = 3 \times \text{page}(R) + \text{card}(S)$$

$$C3: \text{coût}(R \bowtie_{\text{pred}} S) = 3 \times (\text{page}(R) + \text{page}(S))$$

$$C4: \text{coût}(R \bowtie_{\text{pred}} S) = \text{page}(R) + \text{page}(S)$$

$$C5: \text{coût}(R \bowtie_{\text{pred}} S) = \text{page}(R) + \text{card}(R) \times \text{page}(S)$$

$$C6: \text{coût}(R \bowtie_{\text{pred}} S) = \text{page}(R) + \text{card}(R) \times \text{arr}(\text{page}(S) / D(S, c))$$

$$C7: \text{coût}(R \bowtie_{\text{pred}} S) = \text{page}(R) + \text{page}(S) + \text{card}(S) \times \text{card}(R) / D(R, c)$$

$$C8: \text{coût}(R \bowtie_{\text{pred}} S) = 3 \times \text{page}(R) + \text{page}(S)$$

$$C9: \text{coût}(R \bowtie_{\text{pred}} S) = \text{card}(R) + \text{card}(S)$$

Précisez clairement toute hypothèse supplémentaire que vous jugez nécessaire.

Question 1

1) Donner le nombre de services distincts et le nombre moyen de services par projet

nb de services distincts : $D(\text{Service}, ns) =$

nombre moyen de Services par Projet :

2) Donner le nombre de tuples par page des 3 relations.

$\text{ntp}(\text{Emp}) =$

$\text{ntp}(\text{Service}) =$

$\text{ntp}(\text{Projet}) =$

3) Donner la taille des relations en nombre de pages.

$\text{page}(\text{Emp}) =$

$\text{page}(\text{Service}) =$

$\text{page}(\text{Projet}) =$

4) Combien de pages sont nécessaires pour stocker conjointement tous les tuples de R qui ont une même valeur pour l'attribut c ? Donner une formule en fonction de R , c et des notations définies ci-dessus.

$f(R, c) = \dots$

Question 2

1) Parmi C1 à C9, quelles sont les formules symétriques (*i.e.*, pour lesquelles R et S ont le même rôle)

Formules symétriques :

2) Parmi J1 à J9, quels sont les algorithmes symétriques (*i.e.*, pour lesquels R et S ont le même rôle)

Algorithmes symétriques :

3) Associer chaque algorithme de jointure avec la formule la plus appropriée. Expliquer vos choix en une phrase. Répondre en complétant le tableau. Indication :

Coût	Explication
J1: C5	page(R) : lire R une seule fois card(R) * page(S): lire S autant de fois qu'il y a de tuples dans R
J2	
J3	
J4:	
J5 :	
J6 :	
J7 :	

J8 :	
J9 :	

Question 3.

Soit la requête R1:

Select *

From Emp e, Service s

Where e.ns = s.ns

1) On suppose qu'il existe seulement un index plaçant sur Emp.ns. Dire, pour chaque algorithme J1 à J9, s'il est utilisable pour traiter R1? Si oui, donner le coût du plan d'exécution de R1. Sinon expliquer brièvement pourquoi.

J1	J2
J3	J4
J5	J6
J7	J8
J9	

2) On suppose maintenant qu'il existe un index plaçant sur Emp.ns et un index plaçant sur Service.ns. Quel est l'algorithme de coût minimal ? Donner son coût.

<p>J...</p> <p>Coût = ...</p>

3) On suppose maintenant qu'il existe seulement un index plaçant sur Service.ns. Le SGBD possédant seulement 7 pages disponibles en mémoire, on choisit de traiter R1 avec l'algorithme suivant :

Etape 1: Lire Emp par blocs de 6 pages (1 page mémoire étant réservée pour écrire le résultat du tri), et créer des sous-listes triées.

Etape 2: Tant que le nombre de sous-listes est supérieur ou égal à 6 :
fusionner 6 sous-listes en 1 seule liste, écrire le résultat sur disque.

Etape 3: Fusionner les listes restantes avec la relation Service

a) Donner le coût de R1, en utilisant cet algorithme, en nombre de pages (lues ou écrites).

Coût de l'étape 1 = ...

Coût de l'étape 2 = ...

Coût de l'étape 3 = ...

Total :.....

b) Combien faut-il de pages mémoire au minimum pour pouvoir utiliser J7 ?

Question 4.

Rappel des formules évaluant la cardinalité de la sélection et de la jointure :

$$card(\sigma_F(R)) = SF(\sigma_{(F)}) * card(R)$$

$$\text{où } SF(\sigma_{A = valeur}) = 1 / D(R, A)$$

$$SF(\sigma_{A > valeur}) = (max(A) - valeur) / (max(A) - min(A))$$

$$SF(\sigma_{A < valeur}) = (valeur - min(A)) / (max(A) - min(A))$$

$card(R \bowtie_{A=B} S) = card(S)$ si A est clé de R, et B est clé étrangère de S.

sinon $card(R \bowtie_{A=B} S) = SFj * card(S) * card(R)$ où $SFj = 1 / \max(D(R, A), D(S, B))$

On donne $SFj = 1/500$ pour la jointure entre Emp et Serv.

Soit la requête **R2** :

Select E.ne, S.ns, P.np

From Emp E, Service S, Projet P

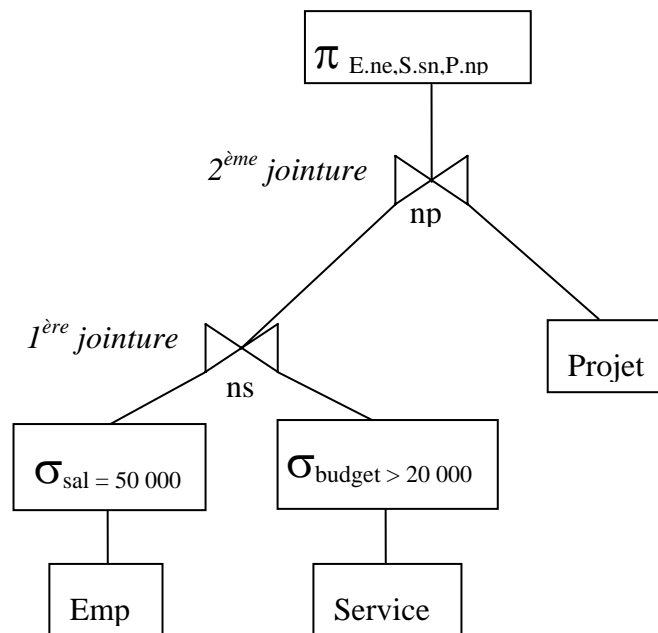
Where E.Salaire = 50000

and S.budget > 20000

and E.ns=S.ns and S.np=P.np

On suppose une répartition uniforme des salaires (par tranche de 5000 dans l'intervalle [10000, 105000]) et des budgets (dans l'intervalle [10000, 30000]). Il y a un index plaçant sur l'attribut salaire de Emp, sur l'attribut ns de Service et sur l'attribut np de Projet.

On considère le plan d'exécution **P1** suivant :



1) Estimez le nombre de n-uplets résultant de chacune des opérations.

$\text{card} (\sigma_{\text{sal} = 50\,000} (\text{Emp})) = \dots$

$\text{card} (\sigma_{\text{budget} > 20\,000} (\text{Service})) = \dots$

$\text{card} ((\sigma_{\text{sal} = 50\,000} (\text{Emp})) \bowtie_{\text{ns}} (\sigma_{\text{budget} > 20\,000} (\text{Service}))) = \dots$

$\text{card} (\text{résultat final}) =$

2) Calculez le coût du plan d'exécution P1, en précisant les algorithmes de jointure utilisés.

$\text{coût} (\sigma_{\text{sal} = 50\,000} (\text{Emp})) = \dots$

$\text{coût} (\sigma_{\text{budget} > 20\,000} (\text{Service})) = \dots$

coût (1^{ère} jointure) = ...

coût (2^{ème} jointure) = ...

coût total = ...

3) Donnez un plan d'exécution de coût optimal pour la requête R2 et calculez son coût.

Plan optimal:

coût total = ...

4) On suppose maintenant que l'index sur l'attribut *np* de *Projet* est non plaçant. Calculez le coût du plan optimal sous cette hypothèse.

5) Même question en supposant que les index sur *salaire* et sur l'attribut *ns* de *Service* sont non-plaçants. Justifiez votre réponse.

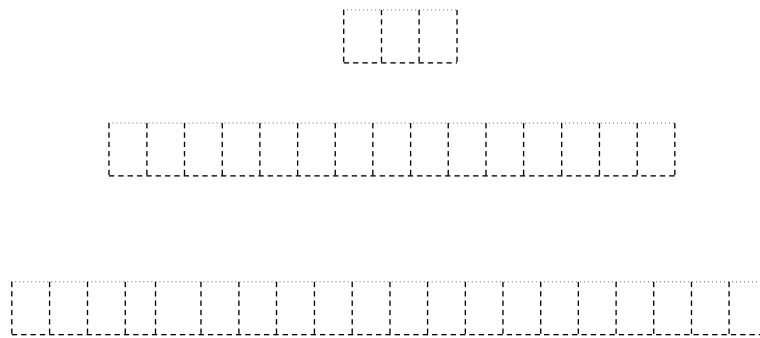
Exercice 3 : Arbres B+**4 pts**

On considère un arbre B+ tel qu'un nœud quelconque peut contenir de **1 à 3** clés.

Question 1

a) Dessiner l'arbre résultant de l'insertion successive des 10 clés 1, 2, ..., 10, dans l'ordre croissant, dans un arbre initialement vide. En cas de débordement d'une feuille, l'éclater en 2 feuilles ayant chacune le même nombre de clés. Utiliser la trame quadrillée pour dessiner les nœuds et laisser un seul espace entre deux nœuds.

Arbre B+ après les 10 insertions :



b) On considère l'insertion successive de N clés 1, 2, ..., N , dans l'ordre croissant, dans un arbre initialement vide. Donner la valeur maximale de N lorsque l'arbre obtenu est de profondeur 3 (un seul niveau intermédiaire).

$N_{\max} =$

c) On considère l'arbre de profondeur 3 qui a le nombre maximum de clés (sans se préoccuper de la façon d'obtenir cet arbre), et M le nombre total de clés dans ses feuilles. N_{\max} est-il égal à M ? Justifier.

Question 2

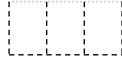
Donner une suite de 10 clés à insérer successivement, dans un arbre vide afin d'obtenir un arbre de profondeur 2 (pas de niveau intermédiaire), en choisissant la valeur des clés parmi les entiers dans $[1, 10]$. Puis représenter l'arbre obtenu.

Indication : Répondre de manière à insérer autant que possible les plus petites valeurs en premier. Parmi toutes les réponses possibles, donner celle qui est la plus "proche" d'une suite croissante. C'est-à-dire, donner la suite qui est composée d'un nombre minimum de sous-suites croissantes.

Suite de clés à insérer:

.....

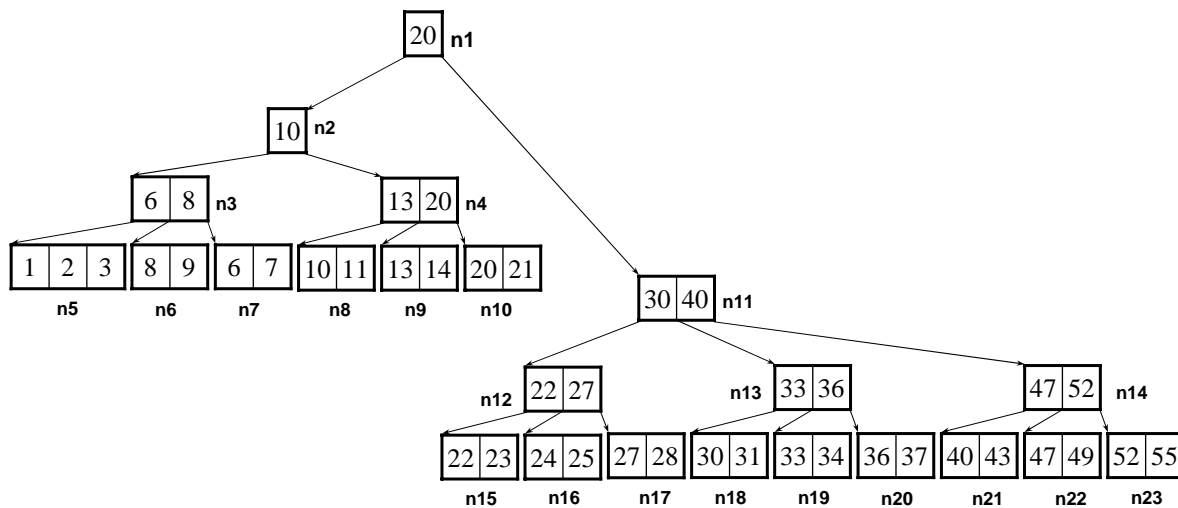
Arbre obtenu :



Exercice 4 : Arbres B+

2 pts

On considère l'arbre B+ d'ordre 2 suivant :



Trouver toutes les erreurs dans cet arbre. Indiquer le numéro n_i du noeud erroné et expliquer brièvement l'erreur. S'il est possible de corriger l'erreur sans restructurer l'arbre, mais en modifiant seulement des valeurs de clés, alors suggérer une correction.

1^{ère} erreur :

2^{ème} erreur :

3^{ème} erreur

4^{ème} erreur

....