Une approche de répartition des données d'un entrepôt basée sur l'Optimisation par Essaim Particulaire

Hacène Derrar*, Omar Boussaid**, Mohamed Ahmed-Nacer*

* Laboratoire LSI, Département Informatique, USTHB Alger
Bp 32 El Alia, bab Ezzouar/ Ager
Hderrar@yahoo.fr, Anacer@mail.cerist.dz
** Laboratoire ERIC, Université Lyon2
Campus Porte des Alpes, 69676 Bron Cedex
Omar.Boussaid@univ-lyon2.fr

Résumé. Dans le contexte des entrepôts de données, le partitionnement des tables, des index et des vues matérialisées en fragments stockés et consultés séparément apporte des améliorations considérables en terme de gestion des données et de coût de traitement. Lors de leurs conceptions, ces techniques se basent sur l'analyse d'informations statistiques recueillies à partir des requêtes les plus fréquentes. Cependant, en raison de leurs caractéristiques les requêtes décisionnelles rendent ce contexte d'analyse très variable. Ceci rend avec le temps, les schémas de fragmentation réalisés inappropriés et par conséquent une dégradation des performances du système. A partir de ce constat, et considérant que la validité d'une approche de partitionnement est soumise à l'épreuve du temps et dépend des besoins et de l'environnement, on propose dans ce papier une approche de répartition des données de l'entrepôt basée sur une méthode issue des approches biomimétiques.

1 Introduction

Dans le contexte des entrepôts de données, le choix de la méthode et du schéma de fragmentation conjugué avec l'évolution perpétuelle des systèmes, des applications et enfin des données constituent les véritables problèmes qui entravent une meilleure exploitation de cette technique. En effet, une approche de fragmentation inappropriée et mal conçue influe considérablement sur les performances du système et plus particulièrement lors de l'exécution des opérations coûteuses telles que la jointure et la multi-jointure qui caractérisent les requêtes décisionnelles.

L'approche présentée dans ce papier s'inscrit dans le cadre d'une solution aux problèmes induits par l'application d'une stratégie de fragmentation inadaptée aux changements perpétuels du contexte dans lequel est exploité l'entrepôt de données. Cette solution consiste à mettre en œuvre un processus d'automatisation visant la redistribution des données à l'issue d'une dégradation des performances observée lors de l'exploitation de l'entrepôt. Nous proposons dans cet article, une approche de répartition des données basée sur une technique d'optimisation issue des approches biomimétiques. Celle-ci permet, à l'issue d'une phase d'évaluation, de déterminer un schéma de fragmentation optimisant les performances. Cet article est organisé comme suit, dans la première section, nous présentons un état de l'art sur

les approches de fragmentation utilisées. Dans la section 2, nous décrivons d'une manière générale le problème généré par les techniques de partitionnement et nous positionnons notre approche ainsi que l'apport attendu. Dans les sections 3 et 4, nous décrivons respectivement le principe de l'approche, ainsi que son application dans les entrepôts de données. Dans la section 5, nous présentons les résultats expérimentaux obtenus en utilisant le benchmark APB-1 release II implémenté sous Oracle 10g. Enfin, nous terminions cet article par une conclusion et des perspectives.

2 Positionnement

A l'issue d'une étude de l'état de l'art sur les techniques de fragmentation des entrepôts de données, il en ressort que toutes les approches de fragmentation, horizontale, verticale ou mixte, se basent lors de leur conception sur l'analyse d'informations statistiques recueillies à partir de l'exécution des requêtes les plus fréquentes. De ce fait, l'adaptation des techniques de fragmentation aux entrepôts de données s'avère plus délicate en raison principalement de la nature des requêtes analytiques. Ces requêtes sont longues, complexes et nécessitent parfois un grand nombre d'opérations de sélection, d'agrégation. Elles peuvent manipuler des centaines voir des milliers de tuples. Les requêtes analytiques sont extrêmement variables, elles sont généralement composées d'une manière interactive et peuvent être exécutées une ou plusieurs fois. Ce type de requêtes appelées aussi requêtes ad hoc, correspond à des requêtes saisies en ligne sans une longue réflexion préalable (Gardarin, 2005). Toutes ces caractéristiques rendent, avec le temps, le schéma de fragmentation mis en place, inapproprié étant donné qu'il a été conçu à partir d'informations statistiques instables.

Par ailleurs, les travaux qui ont traité l'adaptation des techniques de fragmentation se sont focalisés sur l'aspect logique en le dissociant complètement de l'aspect physique de la conception des entrepôts de données. En effet, la conception du schéma de fragmentation et la stratégie de placement des fragments sont deux approches totalement dépendantes. Elles se basent lors de leurs conceptions sur les mêmes informations recueillies lors de l'exploitation des données, en vue d'accomplir le même objectif, à savoir l'amélioration des performances des systèmes.

De cet état de fait, les techniques de fragmentation initialement conçues pour l'amélioration des performances peuvent dans le cadre des entrepôts de données constituer les principaux obstacles pour une meilleure exploitation des données. En effet, il ne s'agit pas de procéder à une simple application de ces techniques mais également d'assurer une adaptabilité parfaite aux caractéristiques spécifiques des entrepôts de données. Pour bénéficier pleinement de leurs avantages, les techniques de fragmentation et de placement des données doivent être constamment revues et adaptées au contexte de l'exploitation de l'entrepôt. Pour ce faire, nous proposons une approche permettant de concrétiser cette adaptabilité par l'introduction de l'aspect dynamicité qui s'avère être une manière « intelligente » de réorganiser les données en vue d'assurer, en permanence, les meilleures performances en termes de traitement des requêtes, de coût de communication et de capacité de stockage. Notre approche, suppose l'existence d'un schéma de fragmentation déjà mis en place. Elle consiste d'abord, à l'issue d'une dégradation des performances, à rechercher un schéma plus optimal et procéder par la suite à la redistribution des données entre fragments. Dans ce papier, nous définissons un nouveau modèle de coût adapté à notre approche ainsi qu'un Algo-

rithme de sélection d'un schéma de fragmentation optimal inspiré des approches biomimétiques.

3 Une approche de répartition des données dans un entrepôt

On suppose l'existence d'un schéma de fragmentation conçu et implémenté sur un entrepôt de données. Le principe de notre approche consiste, à l'issue d'une dégradation des performances, à déterminer un autre schéma de fragmentation plus optimal et procéder par la suite à la redistribution des données selon ce nouveau schéma. Cependant, cette démarche nécessite, dans un premier temps, à évaluer estimer le schéma de fragmentation existant. Néanmoins, mesurer la qualité d'un schéma de fragmentation dont on ne connaît pas à priori le modèle de coût utilisé est une tâche qui n'est pas toujours évidente. De plus, la majorité des algorithmes de conception logique de la fragmentation sont dirigés par la mesure d'affinité. C'est à dire le calcul des fréquences d'accès des requêtes, uniquement entre une paire d'attributs. Ce qui ne permet pas, par conséquent, de mesurer l'affinité entre tous les attributs d'une partition.

Il s'avère donc nécessaire de définir un nouveau modèle de coût permettant d'évaluer et de mesurer l'affinité d'un schéma de fragmentation. Ce modèle de coût devra être générique et flexible permettant éventuellement de prendre en considération d'autres métriques telles que : le type des requêtes, les informations de placement des données, la capacité de stockage et le coût de transfert des données entre sites.

3.1 Une nouvelle fonction objective

Étant donné que toute approche de fragmentation se conçoit à partir d'informations portants sur les fréquences d'accès des requêtes aux données. L'idée consiste à utiliser ce dénominateur commun entre toutes les approches pour définir une nouvelle fonction de coût. Celle-ci permettra d'évaluer un schéma de fragmentation selon les fréquences d'accès des requêtes aux différents fragments. Pour mesurer donc la qualité d'un schéma de fragmentation, nous adaptons une technique d'estimation utilisée dans le domaine de la statistique à savoir le critère de l'erreur au carrée (*Square-Error*). Elle consiste à évaluer un schéma de fragmentation par le calcul de l'erreur au carrée des fréquences des accès des requêtes sur les attributs des différents fragments.

La formulation, ci-dessous, relative à la définition de notre modèle de coût a été inspirée des travaux de Jain et Dubes (1998) et de Muthuraj (1992), qui l'ont utilisé dans le cadre d'une méthode d'apprentissage non supervisée pour le regroupement d'attributs. Pour des raisons de simplification, on considère dans notre cas un entrepôt de données évoluant dans un contexte de traitement local des requêtes et dont la table de fait est fragmentée selon une approche verticale. Nous considérons que les tables de dimension sont de petite taille et par conséquent, elles ne seront pas fragmentées. De plus, notre approche ne considère pas une matrice d'affinité d'attributs, mais une matrice d'usage d'attributs composée, en colonnes des attributs et en lignes des requêtes fréquentes. Les termes de la matrice sont les fréquences d'accès des requêtes aux attributs. Le calcul de l'erreur au carrée permettra de mesurer l'affinité des partitions de taille différentes. Soit la formulation suivante :

n : nombre total des attributs de la table de faits ;
Q : nombre total des requêtes fréquentes ;
fq : fréquence d'accès de la requête q pour q = 1,2,.., Q ;
M : nombre total des fragments ;
n_i : nombre d'attributs dans le fragment i ;
f_{qj}^{i} : la fréquence d'accès de la requête q à l'attribut j dans le fragment i, avec $f_{qj}^{i} \neq 0$;
A_{ij} : le vecteur attribut de l'attribut j dans le fragment i, où f_{qj}^{i} est une composante de ce vecteur ;
S_{iq} : l'ensemble d'attributs du fragment i accédé par la requête q ; égale à 0 si la requête q n'accède pas au fragment i ;
$| S_{iq} |$: nombre d'attributs du fragment i accédé par la requête q ;

Etant donnée une table de faits F de n attributs fragmentée verticalement en M fragments ($F_1, F_2, \ldots, F_M$) contenant chacun n_i attributs. Ainsi, $\sum_{i=1}^{M} n_i = n$. Le vecteur moyen Vi pour le fragment i est défini par : $V_i = \frac{1}{n_i} \sum_{j=1}^{n_i} A_{ij} \qquad 0 < i < M \qquad (1)$

Le vecteur moyen V représente la moyenne des accès des requêtes à tous les attributs du fragment i. Pour un vecteur d'attributs A_{ij}, (A_{ij} - V_i) est dénommé « le vecteur différence » de l'attribut j dans le fragment i. L'erreur au carrée pour le fragment F_i est la somme des carrées de la longueur des vecteurs différences de tous les attributs dans le fragment i. Il est calculé par la formule suivante : $e_i^2 = \sum_{j=1}^{n_i} (A_{ij} - Vi)^Q (A_{ij} - Vi) \qquad 0 < i < M \qquad (2)$

Si $A_{ij} = V_i$ alors $e_i^2 = 0$. Ce cas signifie : soit qu'il y a un seul attribut dans chaque fragment soit que tous les attributs dans chaque fragments sont nécessaires pour l'exécution de la requête. Dans ce papier on s'intéresse au cas où $A_{ij} \neq V_i$ afin de pouvoir comparer les fragments selon la pertinence des attributs.

L'erreur au carrée du schéma de fragmentation globale est calculée par la formule suivante : $E_M^2 = \sum_{i=1}^{M} e_i^2 \qquad (3)$

D'où : $E_M^2 = \sum_{i=1}^{M} \sum_{j=1}^{n_i} (Aij - Vi)^Q (Aij - Vi) \qquad (4)$

Une autre écriture de l'équation 4 permettra de mieux percevoir la contribution de chaque requête pour le calcul de l'erreur au carrée de chaque fragment. Ainsi, le vecteur moyen V_i pour le fragment i peut être défini comme suit :

$$Vi = \begin{bmatrix} \frac{|s_{i1}| * f_1}{n_i} \\ \frac{|s_{i2}| * f_2}{n_i} \\ \ldots\ldots \\ \ldots\ldots \\ \frac{|s_{iq}| * f_q}{n_i} \end{bmatrix}$$

Le vecteur attribut A_{ij}, dont ses composantes sont les fréquences d'accès, est : $A_{ij} = \begin{bmatrix} f_{1j}^{i} \\ f_{2j}^{i} \\ \ldots\ldots \\ \ldots\ldots \\ f_{qj}^{i} \end{bmatrix}$

D'où : $E_M^2 = \sum_{i=1}^{M} \sum_{j=1}^{n_i} \left[f_{1j}^i - \frac{|s_{i1}| * f_1}{n_i}, \dots, f_{qj}^i - \frac{|s_{iq}| * f_q}{n_i} \right] \begin{bmatrix} f_{1j}^i - \frac{|s_{i1}| * f_1}{n_i} \\ f_{2j}^i - \frac{|s_{i2}| * f_2}{n_i} \\ \dots \dots \\ \dots \dots \\ f_{qj}^i - \frac{|s_{iq}| * f_q}{n_i} \end{bmatrix}$ (5)

Par simplification de l'équation ci-dessus, on aura :

$$E_M^2 = \sum_{i=1}^{M} \sum_{q=1}^{Q} \left[f_q^2 * |s_{iq}| \left(1 - \frac{|s_{iq}|}{n_i}\right) \right] \quad (6)$$

Cette équation est la même que l'équation 4, sous une autre forme. On peut donc percevoir d'après cette équation l'apport des accès aux fragments contenant des attributs qui ne sont requises par les requêtes pour le calcul de E_M^2 .

Ainsi, E_M^2 est notre fonction de coût dans un contexte d'exploitation locale d'un entrepôt de données. Elle signifie que la valeur de E_M^2 est proportionnelle au coût dû à l'accès aux fragments contenants des attributs non pertinents. Plus la valeur de cette erreur se rapproche de 0 plus le schéma de fragmentation est optimal. Il revient donc dans la suite de cet article, à chercher un schéma de fragmentation qui minimise cette valeur.

La prochaine phase consiste donc à rechercher un schéma de fragmentation minimisant la valeur de E_M^2 .

3.2 Un algorithme OEP pour la répartition des données

La fonction coût, en l'occurrence l'erreur au carrée, du schéma de fragmentation mis en place ayant été calculée, il s'agit dans cette phase de déterminer un schéma qui minime cette fonction objective. La recherche de tel optimum a un coût exponentiel en temps de calcul et en espace mémoire. En effet, cela fait partie des problèmes NP-difficiles et la sélection des partitions d'attributs demanderait l'exploration de tout l'espace de recherche. Pour *n* attributs, la recherche exhaustive consiste à explorer 2^n-1 sous-ensembles possibles. Pour remédier à cela, le recours à des heuristiques est nécessaire. Pour ce faire, notre approche se base sur une métaheuristique en l'occurrence l'Optimisation par Essaim Particulaire (OEP).

Les algorithmes OEP ont été introduits par Kennedy et Eberhart (1995) comme une alternative aux algorithmes génétiques standards. Ces algorithmes sont inspirés des essaims d'insectes (ou des bancs de poissons ou des nuées d'oiseaux) et de leurs mouvements coordonnés. En effet, tout comme ces animaux qui se déplacent en groupe pour trouver de la nourriture ou éviter les prédateurs, les algorithmes à essaim de particules recherchent des solutions pour un problème d'optimisation. Les individus de l'algorithme sont appelés particules et la population est appelée essaim.

Dans cet algorithme, une particule décide de son prochain mouvement en fonction de sa propre expérience, qui est dans ce cas la mémoire de la meilleure position qu'elle a rencontrée, et en fonction de son meilleur voisin. Ce voisinage peut être défini spatialement en prenant par exemple la distance euclidienne entre les positions de deux particules ou sociométriquement (position dans l'essaim de l'individu). Les nouvelles vitesse et direction de la particule seront définies en fonction de trois tendances : la propension à suivre son propre chemin, sa tendance à revenir vers sa meilleure position atteinte et sa tendance à aller vers son meilleur voisin. Les algorithmes à essaim de particules peuvent s'appliquer aussi bien à des données discrètes qu'à des données continues.

Les algorithmes à essaim de particules ont été utilisés pour réaliser différentes tâches. Dans le cadre de la sélection d'attributs, Agrafiotis et Cedeno (2002) proposent un algorithme basé sur les essaims de particules pour l'étude de la relation quantitative entre la structure et l'activité de composant chimique (Quantitative Structure Activity Relationship). Omran et al., (2002) utilisent les essaims de particules pour effectuer une classification d'images. Dans le cadre de l'extraction de règles de classification, les OEP ont été comparés aux algorithmes génétiques et à l'algorithme C4.5 (Sousa et al.,2003) et ont été appliqués à la génération de règles pour définir le profil des utilisateurs d'un site web (Ujin et Bentley, 2003) et à l'extraction de règles à partir d'un réseau de neurones (He at al., 1998). Aussi, Xiao et al., (2003) utilisent une méthode hybride basée sur les essaims de particules et l'algorithme de clustering "Self-Organizing Maps" pour réaliser un partitionnement des gènes. D'autres travaux ont porté sur l'extension de cette heuristique en vue d'élargir son champ d'adaptation (Clerc 2004, Gourgand 2007).

Notre choix de l'application de l'OEP par rapport à d'autres méthodes évolutionnaires (typiquement, les algorithmes génétiques) se justifie principalement par le fait que cette heuristique :

- met l'accent sur la coopération plutôt que sur la compétition. Il n'y a pas de sélection. L'idée étant qu'une particule même actuellement médiocre doit être conservée ;
- permet d'effectuer un regroupement distribué donc sans contrôle ;
- s'applique au problème d'optimisation combinatoire dynamique dans le cas ou la fonction objective varie dans le temps ;
- permet d'utiliser des fonctions multi-objectifs.

3.2.1 Principe et algorithme de base

On considère dans l'espace de recherche un essaim de particules. Chaque particule est en mouvement selon une vitesse. A partir des informations dont elle dispose, une particule doit décider de son prochain mouvement, c'est-à-dire décider de sa nouvelle vitesse. Pour ce faire, elle combine linéairement trois informations : sa vitesse actuelle ; sa meilleure performance ; la meilleure performance de ses voisines (ses informatrices).
A l'aide de trois paramètres parfois appelés *coefficients de confiance*, qui pondèrent trois tendances : tendance à suivre sa propre voie ; tendance conservatrice (revenir sur ses pas) ; tendance « panurgienne » (suivre le meilleur voisin) ;

Les équations complètes du mouvement d'une particule peuvent alors s'écrire de la manière suivante : $v(t+1) = c_1 v(t) + c_2(p_i - x(t)) + c_3(p_g - x(t))$ (7)

$$x(t+1) = x(t) + v(t+1) \quad (8)$$

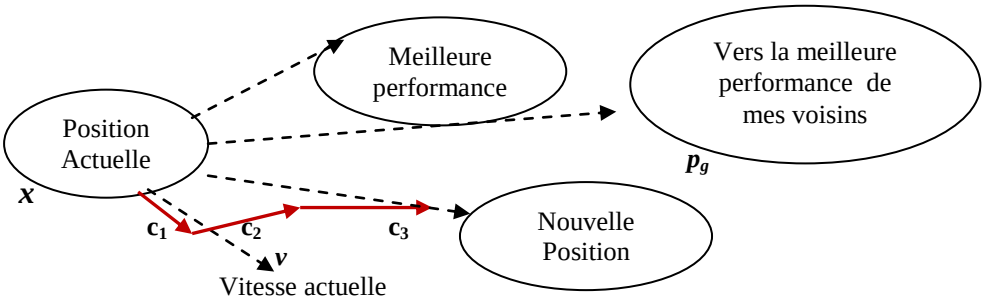

FIG 1. Schéma de principe du déplacement d'une particule.

4 Application à la répartition des données d'un entrepôt

Pour illustrer notre approche, nous considérons un schéma de fragmentation dont la table de faits F de n attributs partitionnée verticalement en M fragments (F_1, F_2, …, F_M) de n_i attributs chacun. L'erreur carrée E_M^2 de ce schéma ayant été calculée, il s'agit donc de déterminer un schéma de fragmentation de la table de faits minimisant la valeur deE_M^2 . Nous considérons une population de n individus où chaque individu représente un attribut de la table de faits se déplaçant dans un espace de dimension 2 en l'occurrence une grille. La taille de la grille G est de forme carrée et sa taille est déterminée automatiquement en fonction du nombre d'individus à traiter. Pour n individus, G comporte L cases par côté avec : $L = \lceil\sqrt{2n}\rceil$. Cette formule permet de s'assurer que le nombre de cases est au moins égal au nombre d'individus. A l'inverse des autres approches qui utilisent les grilles (Lumer et Faieta, 1994), nous permettons à plusieurs individus d'être placés dans une même case, ce qui forme donc une partition.

5 Etude expérimentale

Nous avons implémenté les étapes de notre approche de redistribution des données sous le langage Java. Pour son évaluation, nous avons utilisé le benchmark APB-1 release II Council (1998) implémenté sous *Oracle 10g*. Ce benchmark, utilise un schéma en étoile composé de quatre tables de dimensions (*Prodlevel* de 9000 tuples, *Custlevel* de 900 tuples, *Timelevel* de 24 tuples et *Chanlevel* de 9 tuples) et une table de faits (*Actvars* de 24 000 000 tuples). Pour le calcul des temps d'exécution des requêtes, nous avons utilisé l'utilitaire *Aqua Data Studio 2.0.7*. Pour mener nos tests, nous avons utilisé un ensemble de 50 requêtes englobant différents opérateurs : opérations de jointure, de sélection et des fonctions de calcul et d'agrégations (SUM, COUNT, AVG, MIN, MAX).

Afin de montrer la validité de notre approche, nous avons d'abord commencé par démontrer que les performances se dégradent quand on exécute des requêtes qui ne figurent pas dans la charge de traitement utilisée lors de la fragmentation de l'entrepôt de données. Ainsi, à partir d'une charge de traitement recueillie sur l'exploitation de l'entrepôt de données, nous avons commencé par fragmenter la table de faits selon une approche horizontale et plus particulièrement selon la stratégie de partitionnement par intervalle supportée par Oracle 10g (la

commande *PARTITON BY RANGE*) et nous avons calculé le temps d'exécution des requêtes que nous l'avons comparé avec le temps d'exécution de nouvelles requêtes ne figurant pas dans la charge de traitement.

Notre étude expérimentale s'est déroulée par la suite selon deux étapes. Dans une première étape, nous fragmentons la table de faits selon une approche verticale en utilisant les vues matérialisées et à partir d'un ensemble de requêtes fréquentes nous calculons l'erreur au carré du schéma de fragmentation. Dans une seconde phase, nous appliquons notre algorithme OEP afin de déterminer un nouveau schéma de fragmentation optimal qui minimise la valeur de l'erreur au carrée.

Par ailleurs, différents tests ont été réalisés sur le calcul de l'erreur au carrée selon différents schéma de fragmentation et nous avons constaté que le nombre de fragments est inversement proportionnel à la valeur de l'erreur au carrée. Plus le nombre de fragments est grand plus le coût d'accès aux attributs impertinents devient minime (Fig 3). Ce qui montre également l'apport de la fragmentation verticale si les attributs de fragmentation ont été bien définis.

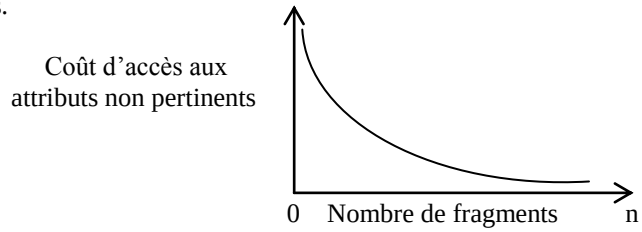

FIG.3- Influence du nombre de fragments sur le coût d'accès aux attributs

En ce qui concerne l'exécution de notre programme d'optimisation, les tests ont été focalisés sur le paramétrage et plus particulièrement sur les coefficients de confiance c_2 et c_3. Pour une meilleure convergence, ces coefficients doivent être puisés dans un intervalle de valeur aléatoire entre 0 et une valeur maximale c. l'équation 12 s'écrira comme suit :

$$v(t+1) = c_1 v(t) + alea(0, c_{max})(p_i - x(t)) + alea(0, c_{max})(p_g - x(t)) \quad (9)$$

Aussi, les valeurs de c_1 et c_{max} ne doivent pas être choisies indépendamment. Selon les différents tests effectués, le premier doit être inférieur à 1 et le second peut être calculé par la formule : $c_{max} = (2/0{,}97725)c_1$. Plus c_1 est proche de 1 plus l'exploration de l'espace de recherche est amélioré.

7 Conclusion et perspectives

Dans cet article nous avons présenté une nouvelle approche de répartition des données basée sur un algorithme d'optimisation inspiré du comportement de certains animaux volant ou nageant qui se déplacent en groupe. Cette approche consiste, à l'issue d'une dégradation des performances, de redistribuer les données selon un nouveau schéma de fragmentation optimal. Nous avons également défini un nouveau modèle de coût, basé sur les fréquences d'accès aux attributs. Ce modèle peut être étendue pour être multi-objectifs en permettant la prise en charge d'autres métriques telles que : le temps d'exécution des requêtes et le temps de transferts des données.

Pour nos travaux futurs, nous envisageons d'abord de continuer les tests expérimentaux portant sur le paramétrage de notre programme et de procéder par la suite à sa validation en comparant les résultats obtenus avec d'autres travaux qui ont utilisé d'autres algorithmes évolutionnaires (génétique, colonie de fourmis artificielles). Nous envisageons également,

d'étendre les tests de notre algorithme sur d'autres approches de fragmentation, le faire passer à l'échelle en considérant un entrepôt de données distribué avec un modèle de coût multi-objectifs.

Références

Agrafiotis,D.K., W. Cedeno (2002). Feature selection for structure-activity correlation using binary particle swarms. *Journal of Medicinal Chemistry*, 45(5) :1098–1107.

Clerc,M., (2004). *Discrete Particle Swarm Optimization*. In G.C. Onwubolu and B.V. Babu, editors,New Optimization Techniques in Engineering, Springer-Verlag,(219-204).

Gardarin,G. (2005). *Base de données*. Edition Eyrolles.

Gourgand,M., S.M.Tchomté (2007). *Une extention de l'optimisation par essaims particulaires*. Séminaire francophone sur l'OEP. France.

He,Z., C.Wei, L. Yang, X. Gao, S. Yao, R.C. Eberhart, et Y. Shi. (1998). Extracting rules from fuzzy neural network by particle swarm optimization. *In Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pages 74–77, Anchorage, Alaska, USA.

Jain,A., R. Dubes (1998). *Algorithms for clustering Data*. Prentice Hall Advanced Reference Series, Englewood Cli_s, NJ.

Kennedy,J., R.C. Eberhart (1995). Particle swarm optimization. *In IEEE Service Center, editor, Proceedings of the 1995 IEEE International Conference on Neural Networks*, pages 1942–1948.

Muthuraj,J.,(1992). *A formal approach to the vertical partitioning problem in distributed database design*. Université de Florida. USA.

Omran,A., Salman, et A.P. Engelbrecht (2002). Image classification using particle swarm optimization. *In Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL)*, pages 370–374.

Sousa,T., A. Neves, A. Silva (2003). Swarm optimisaton as a new tool for data mining. *In Proceedings of NIDICS*, Nice, France.

Ujjin,S., P.J. Bentley (2003). Particle swarm optimization recommender system. *In Proceedings of the IEEE Swarm Intelligence Symposium 2003*, pages 124–131, Indianapolis, Indiana, USA.

Xiao,X., E.Dow, R.Eberhart, Z.BenMiled, et R.J.Oppel (2003). Gene clustering using self-organizing maps and particle swarm optimization. *In Second IEEE International Workshop on High Performance Computational Biology (HICOMB).*

Summary

In the context of the data warehouses, partitioning tables, indexes and materialized views in fragments stored and consulted separately brings considerable improvements in term of management of the data and the cost of treatment. During their conceptions, these techniques base on the analysis of statistical information collected from the most frequent requests. However, because of their characteristics the decisional requests return this context of analysis very variable. This make in time the realized plans of fragmentation inappropriate and consequently a degradation of the performances of the system. From this report and considering that the validity of an approach of partitioning is subjected to the time and depends on needs and on the environment, we propose in this paper an approach of distribution of the data based on a method resulting from the biomimetic approaches.