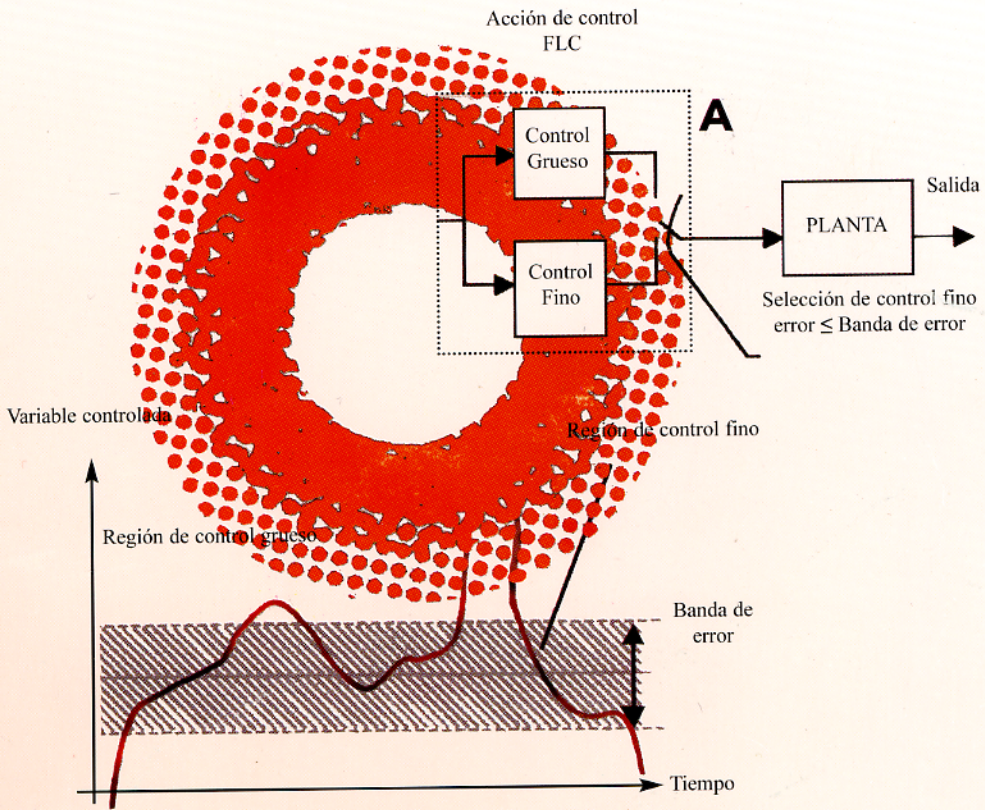


Ingeniería

Revista de la Universidad de Costa Rica
Enero/Diciembre 1999 VOLUMEN 9 Nos. 1 y 2



INGENIERIA

Revista Semestral de la Universidad de Costa Rica
Volumen 9, Enero/Diciembre 1999 Números 1 y 2

DIRECTOR

Rodolfo Herrera J.

CONSEJO EDITORIAL

Víctor Hugo Chacón P.

Ismael Mazón G.

Domingo Riggioni C.

CORRESPONDENCIA Y SUSCRIPCIONES

Editorial de la Universidad de Costa Rica
Apartado Postal 75
2060 Ciudad Universitaria Rodrigo Facio
San José, Costa Rica

CANJES

Universidad de Costa Rica
Sistema de Bibliotecas, Documentación e Información
Unidad de Selección y Aquisiciones-CANJE
Ciudad Universitaria Rodrigo Facio
San José, Costa Rica

Suscripción anual:

Costa Rica: ₡ 1 000,00

Otros países: US \$ 30,00

Número suelto:

Costa Rica: ₡ 750,00

Otros países: \$ 20,00



FUNCIÓN DE LA EVALUACIÓN DE LAS PARTICIONES EN BASES DE DATOS DISTRIBUIDAS ORIENTADAS A OBJETOS

Elzbieta Malinowski G.¹

Resumen

Las bases de datos orientadas a objetos se utilizan cada vez con más frecuencia. Estas bases de datos heredan muchos de los conceptos desarrollados para las bases de datos relacionales. De modo similar, las bases de datos distribuidas relacionales tienen estrecha conexión con bases de datos distribuidas orientadas a objetos. En este artículo se presenta el desarrollo de una nueva función llamada *Partition Evaluator for Object-Oriented Databases* que permite evaluar las particiones "verticales" en bases de datos distribuidas orientadas a objetos. Se demuestra que esta función es la generalización del así llamado *Partition Evaluator* desarrollado para bases de datos distribuidas relacionales.

Summary

Object-oriented databases are used with more frequency now than before. These databases inherit many concepts developed for relational databases. In similar way, distributed relational databases have strong connection with object-oriented distributed databases. In this article we present the development of new function called *Partition Evaluator for Object-Oriented Databases* that allows to evaluate "vertical" partitions in object-oriented databases. We show that this function is the generalization of so called *Partition Evaluator* developed for distributed relational databases.

1. INTRODUCCIÓN

Las bases de datos distribuidas relacionales (BDDR) fueron investigadas durante muchos años. Sin embargo, el interés por su desarrollo creció conforme las compañías se expandieron geográficamente y se implementaron las redes de computadoras. Son presentadas como una colección de múltiples y lógicamente relacionadas bases de datos (BD). Deben ser diseñadas tomando en cuenta todos los aspectos de la BD centralizada y además incluir los aspectos relacionados con la fragmentación, la replicación y la localización para mejorar su desempeño [15].

Uno de los aspectos ampliamente investigados en BDDR es la fragmentación. El enfoque relacional distingue dos tipos de

fragmentación: horizontal, vertical y también la fragmentación híbrida que está formada por los dos tipos de fragmentación mencionados anteriormente. Cada una de ellas tiene sus propios criterios para determinar el mejor esquema de partición para el sistema relacional ([2, 3, 4, 12, 13, 14]).

Por otro lado, aunque se han hecho un número considerable de investigaciones con respecto a las bases de datos orientadas a objetos (BDOO), muy pocas se relacionan con la distribución de los objetos y específicamente con la fragmentación de éstos. Esta última es estudiada por Ezeife y Barker [5] quienes mencionan que sus beneficios deberían ser también reconocidos en BDOO. Aunque existe una diferencia significativa entre las BD relacionales y BDOO, debe ser claro que el

¹ Ing. M.Sc. Prof. Esc. Ciencias Comp.e Informática Univ. de Costa Rica

enfoque relacional puede ser visto (ignorando por la simplicidad la existencia de los métodos) como un caso especial de sistemas orientados a objetos sin la jerarquía de clases y atributos complejos (contienen la referencia a otros objetos). Por consiguiente, muchos de los algoritmos desarrollados para BDDR pueden ser generalizados y aplicados para BDOO, como lo es el evaluador de particiones (*PEOO*, *partition evaluator for object-oriented databases*) que puede ser aplicado en la búsqueda exhaustiva para evaluar cada una de las combinaciones de la fragmentación vertical de los objetos o para evaluar la "bondad" de algoritmos de fragmentación. Este evaluador de particiones es la generalización del evaluador de particiones (*PE*, *partition evaluator*) desarrollado para las bases de datos relacionales y presentado por Chakravarthy et al. [1].

El presente artículo menciona los aspectos e investigaciones hechas con respecto a la fragmentación tanto en bases de datos relacionales como orientadas a objetos. Además, en forma breve se presenta la motivación para el desarrollo de *PEOO* y se especifica el desarrollo de la función objetiva, presentando las notaciones usadas y los pasos para obtener la fórmula para el costo local y el costo remoto.

2. ASPECTOS DE LA FRAGMENTACION EN BASES DE DATOS RELACIONALES Y BASES DE DATOS ORIENTADAS A OBJETOS

La fragmentación vertical permite al diseñador agrupar los atributos de cada relación en registros más pequeños para mejorar la eficiencia de las transacciones en bases de datos. Sin embargo, el número de las particiones posibles con m atributos presentes es igual a m número de Bell; para bases de datos grandes este número es igual a m^m [13]. Para hacer este número más pequeño, en la mayoría de las investigaciones se usa un

enfoque heurístico. Una de las suposiciones más usadas en este enfoque es que cada fragmento debe estar muy cerca de los requerimientos de la transacción que lo ocupa [13]. Esto significa, que en el caso ideal el fragmento puede tener sólo los atributos que la transacción local necesita y ésta no necesita acceder ninguno de ellos de otro nodo remoto, lo cual no es real. La mejor forma de partir los atributos para una transacción no es necesariamente la mejor forma para otra transacción. La meta factible es maximizar el desempeño de todas las transacciones tomándolas como un solo grupo, pues, el diseñador debe asegurar el acceso mínimo a los atributos remotos para todas las transacciones ejecutadas en cada uno de los sitios.

El seguimiento de estos conceptos se puede ver en las investigaciones presentadas sobre la fragmentación vertical en BDDR. Actualmente existe un número considerable de algoritmos para la fragmentación vertical en BD relacionales (Hoffer y Severance [8], Hammer y Niamir [7], Navathe et al. [13], Cornell y Yu [4], Navathe y Ra [14], Chu y Jeong [3]). En la mayoría de estos algoritmos se usa la matriz de afinidad de los atributos (*AAM*, *attribute affinity matrix*). Sin embargo, en este artículo se enfatiza más el enfoque presentado por Chakravarthy et al. [1] que defiende el uso de la matriz del uso de atributos (*AUM*, *attribute usage matrix*) en lugar de *AAM*. Chakravarthy et al. opinan que *AAM* solamente señala la "afinidad" entre las parejas de los atributos y las relaciones entre más de dos de ellos no se pueden medir en esta matriz. Basándose en la *AUM* ellos establecen la función que puede medir la "bondad" de un esquema de fragmentación. Esta función se puede usar para el proceso de fragmentación (búsqueda exhaustiva) o puede ser aplicada después de usar algún otro algoritmo de fragmentación.

Aunque los conceptos de fragmentación desarrollados para BDDR son ampliamente

aceptados, los algoritmos de la fragmentación presentados para BDOO, en general, no se basan en ellos. Algunos autores presentan algoritmos desarrollados e implementados para fragmentar BDOO (Ezeife y Barker [5], Gruber y Amsleg [6]), estableciendo nuevos conceptos como afinidad entre objetos, fragmentos derivados, ligas de jerarquía, y otros. Otros autores analizan en forma general la posibilidad de la fragmentación sin ninguna implementación (Karlalalem y Li [10], Karlalalem et al. [11]). Esta falta de unificación de la visión con respecto a lo que es la fragmentación en BDOO hace difícil la motivación de los investigadores para profundizar en esta área.

3. MOTIVACIÓN PARA EL DESARROLLO DE LA NUEVA HERRAMIENTA PARA LA FRAGMENTACION VERTICAL EN BDOO

En el enfoque relacional los algoritmos de fragmentación vertical distribuyen los atributos. Sin embargo, en el enfoque orientado a objetos el esquema conceptual puede ser mucho más complejo, lejos de factores usados en el enfoque relacional (frecuencias de las transacciones, uso de los atributos, tipo de predicado, y otros). Adicionalmente deben ser considerados algunos aspectos nuevos, como: métodos, estructura jerárquica y los atributos complejos.

Además, la AUM usada en la fragmentación vertical de los trabajos previos en BDDR, solamente considera el caso cuando las frecuencias de acceso de los atributos por una transacción específica son iguales para todos los atributos. Sin embargo, en el caso de BDOO, la AUM puede tener diferentes frecuencias de atributos para la misma transacción; por ejemplo, en el caso de atributos complejos, cuando el objeto hacia el cual dicho atributo tiene referencia puede ser accesado indirectamente por medio de este objeto complejo o directamente desde alguna

otra transacción. Pues, el evaluador de particiones propuesto por Chakravarthy et al. [1] no puede ser aplicado sin modificaciones para el caso de BDOO.

Otro aspecto importante es que la mayoría de los algoritmos para la fragmentación vertical en BDDR usan la matriz de "afinidad" de los atributos como entrada. Dicha matriz no puede expresar la "afinidad" entre más de dos atributos, por consiguiente, una herramienta que usa la matriz de uso de atributos puede con más exactitud reflejar el comportamiento real del sistema. Además, algunos algoritmos existentes se declaran como los mejores y entregan las soluciones "óptimas" diferentes para los mismos datos de entrada. De este forma, el resultado de un algoritmo no puede ser comparado con los resultados de los otros; de aquí la necesidad de tener una herramienta para evaluar en qué medida los resultados son "óptimos". Sin embargo, esta herramienta fue desarrollada para BDDR después de que muchos de los algoritmos de fragmentación fueron presentados. Dicha herramienta para BDOO permite a los diseñadores evaluar los nuevos algoritmos de fragmentación conforme ellos se hacen disponibles.

4. DESARROLLO DE LA FUNCIÓN OBJETIVA

La meta de la fragmentación de los atributos es obtener el mínimo costo de procesamiento para un conjunto dado de transacciones con respecto al acceso a sus respectivos atributos. Es poco probable obtener un esquema de fragmentación ideal donde cada transacción accesa localmente solo los atributos que necesita y no necesitará acceder ninguno de los atributos de los lugares remotos. La función objetiva propuesta aquí trata de balancear el costo del acceso local y remoto para una transacción específica. De acuerdo con Chakravarthy et al. [1] el costo general en el ambiente distribuido que debemos minimizar consiste de dos elementos:

1. Costo de acceso de los atributos locales irrelevantes del sitio local, asumiendo que todos los datos requeridos por la transacción son localmente accesibles.

2. Costo de acceso de los atributos remotos relevantes requeridos por la transacción desde los sitios remotos.

Chakravarthy et al. [1] han demostrado el comportamiento esperado de la función objetiva usando los componentes especificados anteriormente:

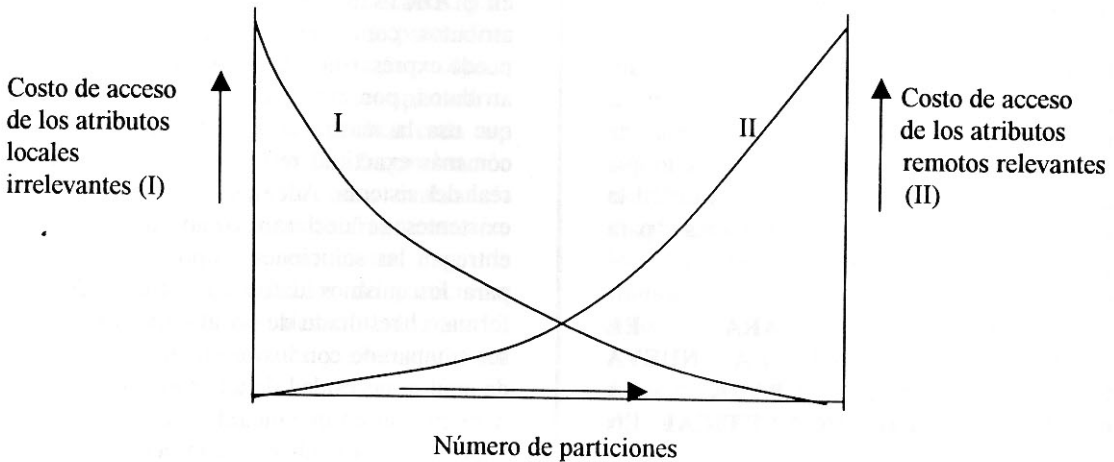


Figura No. 1

Individualmente, el costo de acceso de los atributos locales irrelevantes debería tener el valor máximo para un fragmento, donde están todos los atributos del sistema, y cero para la partición donde la cantidad de fragmentos es igual al número de atributos. Ocurre lo contrario con el costo del acceso a los atributos remotos relevantes con valor cero y máximo para estos dos extremos respectivamente. Además, como lo especifican los autores mencionados anteriormente, el costo local debería ser más sensible para los tamaños de particiones pequeñas, lo contrario del costo remoto donde éste debería ser más sensible para los fragmentos grandes.

Es evidente que el evaluador de particiones para BDOO debería tener el mismo comportamiento que se presentó antes para BDDR. Sin embargo, el costo del acceso de los atributos locales irrelevantes tiene un componente adicional. Debido que las frecuencias de uso de los atributos por la misma transacción pueden ser diferentes, se penaliza el acceso de algunos atributos locales relevantes cuando son accedados y no usados en la transacción. En este caso, estos atributos relevantes son tratados de igual manera que los atributos irrelevantes. Esta penalidad será incluida en el costo local.

4.1 SUPOSICIONES Y ANOTACIONES

Para considerar el enfoque orientado a objetos se debe representar en forma diferente la matriz original AUM. La nueva matriz llamada TAUM (*transaction-attribute usage matrix*, la matriz de uso transacciones - atributos) tiene las frecuencias de acceso diferentes, contrario a la original AUM, en la cual se usan los mismos valores de las frecuencias para todos los atributos accedados por la misma transacción. Eso indica que la matriz (1) la cual se presenta para el caso relacional, debería ser reemplazada por la matriz (2), desarrollado para las BDOO.

<i>trans/atrs</i>	at_1	at_2	\dots	at_n	
tr_1	q_1	q_1	\dots	q_1	
tr_2	q_2	q_2	\dots	q_2	
\vdots	\vdots	\vdots	\vdots	\vdots	
tr_T	q_T	q_T	\dots	q_T	(1)

donde q_t representa la frecuencia de la transacción $t = 1, 2, \dots, T$

<i>trans/atrs</i>	$at_{i,j}^1$	$at_{i,j}^2$	\dots	$at_{i,j}^n$	
tr_1	$f_{1,1}$	$f_{1,2}$	\dots	$f_{1,n}$	
tr_2	$f_{2,1}$	$f_{2,2}$	\dots	$f_{2,n}$	
\vdots	\vdots	\vdots	\vdots	\vdots	
tr_T	$f_{T,1}$	$f_{T,2}$	\dots	$f_{T,n}$	(2)

donde $f_{i,j}$ representa la frecuencia con la cual la transacción inaccesa el atributo j .

Aunque es posible considerar la replicación de los datos, en este primer paso del desarrollo de *PEOO* se asume que no existen réplicas y que la fase de fragmentación sigue con la fase de localización de los fragmentos. Adicionalmente, se considera que cada uno de estos fragmentos tiene los atributos no sobrepuestos, lo cual indica que n atributos se particionan en M fragmentos (P_1, P_2, \dots, P_M) con n_i atributos en cada fragmento. Por lo

tanto $\sum_{i=1}^M n_i = n$. Estas suposiciones permiten

al lector y al autor de este artículo simplificar los pasos del desarrollo de *PEOO* sin perder su generalidad. Adicionalmente, para simplificar se asume que el acceso simple al dato en el fragmento local corresponde a la unidad del costo.

Como la localización de los datos no es conocida durante la partición, el costo relacionado con los atributos remotos relevantes se calcula asumiendo que los fragmentos de datos necesitados por la transacción son localizados en diferentes

sitios. Es importante notificar que la contribución para el costo del acceso remoto de los atributos irrelevantes está ya incluida en el primer término.

No se hace ninguna suposición con respecto a la entrada de datos.

Los siguientes son los parámetros propuestos por Chakravarthy et al. [1] y también usados aquí con pequeñas modificaciones:

- n Número total de atributos en la relación considerada para la partición.
- T Número de transacciones que se considera.
- M Número total de fragmentos en la partición.
- n_i Número de atributos en el fragmento i .
- n_{ikt}^r Número total de atributos en el fragmento k accesados remotamente con respecto al fragmento i por la transacción t .
- f_{ij}^i Frecuencia de la transacción t accedendo el atributo j en el fragmento i ; note que f_{ij}^i tiene el valor de 0 o de q_t para el enfoque relacional.

- A_{ij} Vector de Atributos para el atributo j en el fragmento i .
- S_{it} Conjunto de los atributos relevantes en el fragmento i que la transacción t accesa; es vacío si la transacción t no necesita el fragmento i .
- $|S_{it}|$ Número de atributos relevantes en el fragmento i que la transacción t accesa.
- I_{it} Conjunto de los atributos irrelevantes en el fragmento i que la transacción t accesa.
- $|I_{it}|$ Número de atributos irrelevantes en el fragmento i que la transacción t accesa.
- R_{ik} Conjunto de los atributos relevantes en el fragmento k accedidos remotamente con respecto al fragmento i por la transacción t .
- $|R_{ik}|$ Número de atributos relevantes en el fragmento k accedidos remotamente con respecto al fragmento i por la transacción t .

- CC_{ik} Costo de comunicación entre los sitios i y k .

4.2 COSTO DE ACCESO LOCAL

Para el componente de costo de acceso local se usa el mismo criterio de error cuadrático que fue usado por Chakravarthy et al. [1] basado en Jain y Dubes [9]. Este criterio ofrece el factor de penalidad cada vez que un atributo irrelevante es accedido. El mejor costo de procesamiento local es cuando este valor es mínimo.

El vector de atributos introducido por Chakravarthy et al. [1] se presenta de la siguiente forma:

$$A_{ij} = [f_{1j}^i \quad f_{2j}^i \quad \dots \quad f_{ij}^i]^T \tag{3}$$

El vector media V_i para el fragmento i se puede definir de la siguiente forma:

$$V_i = \left[\sum_{j=1}^{|S_{i1}|} f_{1j}^i / n_i \quad \sum_{j=1}^{|S_{i2}|} f_{2j}^i / n_i \quad \dots \quad \sum_{j=1}^{|S_{in}|} f_{ij}^i / n_i \right]^T \tag{4}$$

donde T denota matriz transpuesta.

Además $|S_{it}|$ es el número de atributos en la partición i que la transacción t accesa y n_i es el número de los atributos en la partición i . (Esta suposición debería tener el límite superior igual a n_i . Sin embargo, f_{ij}^i es igual a 0 para todos los atributos irrelevantes y se puede usar el valor de $|S_{it}|$ en lugar de n_i).

El error cuadrático e_i^2 para el fragmento P_i y el error cuadrático para el esquema de

partición total E_L^2 , son definidos en la misma manera como para el enfoque relacional:

$$e_i^2 = \sum_{j=1}^{n_i} (A_{ij} - V_i)^T (A_{ij} - V_i) \tag{5}$$

$$E_L^2 = \sum_{i=1}^M e_i^2 \tag{6}$$

En el desarrollo de PE para el enfoque relacional se encontró el componente de

penalidad irrelevantes dos elementos

- 1.
- 2.

$$E_L^2 = \sum_{i=1}^M \sum_{j=1}^n$$

Esta fórmula

$$E_L^2 = \sum_{i=1}^M$$

Después

$$E_L^2 = \sum_{i=1}^M$$

Reescribiendo

penalidad por el acceso de los atributos locales irrelevantes. Sin embargo, *PEOO* va a tener dos elementos:

1. La penalidad por acceso de los atributos locales irrelevantes (similar al enfoque relacional).
2. La penalidad por el acceso de los atributos relevantes, la cual aparece por las diferentes frecuencias del

acceso de los atributos por la misma transacción (se debe pagar la penalidad por acceder los atributos más veces de lo necesario).

Reescribiendo E_L^2 de forma distinta se demuestra la contribución de estos dos componentes. Al sustituir V_i y A_{ij} en la ecuación (6) para E_L^2 se obtiene que:

$$E_L^2 = \sum_{i=1}^M \sum_{j=1}^{n_i} \left[f_{1j}^i - \sum_{p=1}^{|S_{11}|} f_{1p}^i / n_i \quad f_{2j}^i - \sum_{p=1}^{|S_{21}|} f_{2p}^i / n_i \quad \dots \quad f_{tj}^i - \sum_{p=1}^{|S_{it}|} f_{tp}^i / n_i \right] \left[f_{1j}^i - \sum_{p=1}^{|S_{11}|} f_{1p}^i / n_i \quad f_{2j}^i - \sum_{p=1}^{|S_{21}|} f_{2p}^i / n_i \quad \dots \quad f_{tj}^i - \sum_{p=1}^{|S_{it}|} f_{tp}^i / n_i \right]^T \quad (7)$$

Esta fórmula puede ser reducida a:

$$E_L^2 = \sum_{i=1}^M \sum_{j=1}^{n_i} \sum_{t=1}^T \left[\left[f_{tj}^i - \frac{\sum_{p=1}^{|S_{it}|} f_{tp}^i}{n_i} \right] * \left[f_{tj}^i - \frac{\sum_{p=1}^{|S_{it}|} f_{tp}^i}{n_i} \right] \right] \quad (8)$$

Después de la expansión de los términos internos se recibe:

$$E_L^2 = \sum_{i=1}^M \sum_{j=1}^{n_i} \sum_{t=1}^T \left[\left(f_{tj}^i \right)^2 + \frac{\sum_{p=1}^{|S_{it}|} f_{tp}^i}{n_i} - 2 * \left[f_{tj}^i * \frac{\sum_{p=1}^{|S_{it}|} f_{tp}^i}{n_i} \right] \right] \quad (9)$$

Reescribiendo lo anteriormente presentado de distinta forma:

$$E_L^2 = \sum_{i=1}^M \sum_{t=1}^T \left[\sum_{j=1}^{n_i} (f_{tj}^i)^2 + \sum_{j=1}^{n_i} \left[\frac{\sum_{p=1}^{|S_{it}|} f_{tp}^i}{n_i} \right] - 2 * \sum_{j=1}^{n_i} \left[f_{tj}^i * \frac{\sum_{p=1}^{|S_{it}|} f_{tp}^i}{n_i} \right] \right] \quad (10)$$

Ahora se puede dividir $\sum_{j=1}^{n_i}$ en dos partes: $\sum_{j=1}^{|S_{it}|}$ y $\sum_{j=1}^{|I_{it}|}$ donde $I_{it} = n_i - S_{it}$

$$E_L^2 = \sum_{i=1}^M \sum_{t=1}^T \left[\sum_{j=1}^{|S_{it}|} (f_{tj}^i)^2 + \sum_{j=1}^{|I_{it}|} (f_{tj}^i)^2 + \sum_{j=1}^{|S_{it}|} \left[\frac{\sum_{p=1}^{|S_{it}|} f_{tp}^i}{n_i} \right]^2 + \sum_{j=1}^{|I_{it}|} \left[\frac{\sum_{p=1}^{|I_{it}|} f_{tp}^i}{n_i} \right]^2 - 2 * \sum_{j=1}^{|S_{it}|} \left[f_{tj}^i * \frac{\sum_{p=1}^{|S_{it}|} f_{tp}^i}{n_i} \right] - 2 * \sum_{j=1}^{|I_{it}|} \left[f_{tj}^i * \frac{\sum_{p=1}^{|I_{it}|} f_{tp}^i}{n_i} \right] \right] \quad (11)$$

Agrupando los términos para los atributos relevantes e irrelevantes se obtiene:

$$E_L^2 = \sum_{i=1}^M \sum_{t=1}^T \left[\sum_{j=1}^{|S_{it}|} (f_{tj}^i)^2 + \sum_{j=1}^{|I_{it}|} (f_{tj}^i)^2 + \sum_{j=1}^{|S_{it}|} \left[\frac{\sum_{p=1}^{|S_{it}|} f_{tp}^i}{n_i} \right]^2 + \sum_{j=1}^{|I_{it}|} \left[\frac{\sum_{p=1}^{|I_{it}|} f_{tp}^i}{n_i} \right]^2 - 2 * \sum_{j=1}^{|S_{it}|} \left[f_{tj}^i * \frac{\sum_{p=1}^{|S_{it}|} f_{tp}^i}{n_i} \right] - 2 * \sum_{j=1}^{|I_{it}|} \left[f_{tj}^i * \frac{\sum_{p=1}^{|I_{it}|} f_{tp}^i}{n_i} \right] \right] \quad (12)$$

Al final se tiene:

$$E_L^2 = \sum_{i=1}^M \sum_{t=1}^T \left[\sum_{j=1}^{|S_{it}|} \left[f_{tj}^i - \frac{\sum_{p=1}^{|S_{it}|} f_{tp}^i}{n_i} \right]^2 + \sum_{j=1}^{|I_{it}|} \left[f_{tj}^i - \frac{\sum_{p=1}^{|S_{it}|} f_{tp}^i}{n_i} \right]^2 \right] \quad (13)$$

El primer componente representa el costo de la penalidad por acceder los atributos relevantes que tienen diferentes frecuencias. El segundo término representa el costo de la penalidad por acceder los atributos irrelevantes.

Se puede transformar fácilmente esta fórmula para el caso relacional. Para este caso se tiene $f_{tj}^i = q_t$ para los atributos relevantes y $f_{tj}^i = 0$ para los atributos irrelevantes.

También $\sum_{j=1}^{|S_{it}|}$ es igual al valor de $|S_{it}|$ y $\sum_{j=1}^{|I_{it}|}$ igual al valor $n_i - |S_{it}|$. Esto da:

$$E_L^2 = \sum_{i=1}^M \sum_{t=1}^T \left[|S_{it}| * q_t^2 \left(1 - \frac{|S_{it}|}{n_i} \right)^2 + (n_i - |S_{it}|) \left(q_t * \frac{|S_{it}|}{n_i} \right)^2 \right] \quad (14)$$

la misma fórmula intermedia presentada por Chakravarthy et al. [1] y puede ser transformada hacia su versión final dando el costo penal por acceder los atributos irrelevantes en BDDR.

$$E_L^2 = \sum_{i=1}^M \sum_{t=1}^T \left[q_t^2 * |S_{it}| \left(1 - \frac{|S_{it}|}{n_i} \right)^2 \right] \quad (15)$$

Así se puede ver que la fórmula para el costo local de acceso para el enfoque relacional es un caso especial de fórmula presentada anteriormente para el enfoque orientado a objetos. De hecho, la nueva fórmula puede ser usada para el modelo relacional cuando las frecuencias de acceso no son iguales para todos los atributos usados por alguna transacción.

4.3 COSTO DE ACCESO REMOTO

Basándose en la explicación presentada por Chakravarthy et al. [1] y en la parte previa de este trabajo, la fórmula de costo de penalidad de acceso remoto a los atributos se puede calcular de la siguiente manera: dado el conjunto de particiones, para cada transacción ejecutándose en la partición se calcula la proporción del número de atributos remotos accedidos y el número total de atributos en cada una de las particiones remotas. Aquí, en contraste con la fórmula presentada para el enfoque relacional, se usa las frecuencias individuales de acceso a cada atributo debido a la posibilidad de tener diferentes frecuencias de acceso de los atributos por la misma transacción. Además, adicionalmente se considera el costo de comunicación que no se incluyó en el enfoque relacional.

$$E_R^2 = \sum_{t=1}^T \Delta_{i=1}^M \sum_{k=1}^i \left[CC_{ik} * \sum_{p_k} (f_{tp_k}^k)^2 * \frac{|R_{itk}|}{n_{itk}^{rem}} \right] \quad (16)$$

donde p_k indica los atributos relevantes en el fragmento k accedidos remotamente con respecto al fragmento i por la transacción t . Similarmente con la fórmula relacional PE , Δ es el operador de mínimum , máximum o de promedio sobre todos fragmentos i , presentando el costo de acceso optimista, pesimista o promedio respectivamente.

Esta fórmula es la generalización de la fórmula presentada por Chakravarthy et al. [1] para BDDR. Con las suposiciones

$$\sum p_k = |R_{itk}|, f_{tp_k}^k = q_k \text{ y } CC_{ik} = 1$$

para todos los sitios en el modelo relacional, se obtiene la misma fórmula del costo de acceso remoto, presentada en el artículo anteriormente mencionado.

El evaluador de particiones para BDOO, $PEOO$, se presenta de la siguiente forma:

$$PEOO = E_L^2 + E_R^2 \quad (17)$$

Basándose en las fórmulas (11) y (14) y en el análisis similar presentado por Chakravarthy et al. [1] el comportamiento esperado de esta función es igual al presentado en la figura No. 1.

5. CONCLUSIONES Y TRABAJOS FUTUROS

Las BDDR permiten a los usuarios distribuir los datos de acuerdo a su uso. Esto disminuye el costo de procesamiento de transacciones y mejora el rendimiento del sistema. El uso extenso de las redes de computadoras aumentó

el interés con respecto a las BDDR. Sin embargo, cuando parecía que estas BD reciben mucha atención por medio de los implementadores, surgió una nueva tendencia de BDOO. Las valiosas investigaciones y algoritmos de fragmentación hechos para las BDDR no se podrían aplicar en forma directa a este nuevo enfoque.

El desarrollo de la función de evaluación de particiones permite demostrar la posibilidad de generalizar y adaptar algunos conceptos del enfoque relacional al enfoque orientado a objetos. En este artículo se pudo señalar que la función objetiva desarrollada para bases de datos relacionales es un caso particular de la función objetiva establecida para el enfoque orientado a objetos. Esto permite dirigir la investigación sobre BDOO hacia el aprovechamiento de trabajos existentes desarrollados para BDDR, así como contar con $PEOO$ da la posibilidad de evaluar los nuevos algoritmos de fragmentación conforme éstos surgen.

Los resultados del modelaje y de la fórmula obtenida de $PEOO$ se usaron para examinar el comportamiento de la función objetiva. Se implementó el algoritmo usando lenguaje C++ en ambiente UNIX y se confirmaron los resultados esperados que se presentaron en la figura No. 1.

La limitación de este modelo consiste que en la $PEOO$ se necesita tener como entrada todas posibles fragmentaciones. Esta búsqueda exhaustiva consume recursos y tiempo. Así es imposible, por ahora, realizar la implementación contra los casos reales de BD con muchos atributos.

El acercamiento presentado en este artículo asume la presencia de las estadísticas sobre las frecuencias de las transacciones, métodos y atributos junto con la información sobre la estructura jerárquica del sistema. En el caso cuando otra información está disponible, el $PEOO$ puede ser adaptado para las siguientes situaciones específicas:

1. El tamaño fijo de la partición: Si el número de fragmentos descados es conocido con anticipación, el programa puede ser fácilmente modificado para esta situación en especial.
2. Replicación de los objetos: El *PEOO* no considera la replicación de los objetos. Sin embargo, si la replicación de los datos es recomendada, se puede aplicar y los datos correspondientes serán recuperados desde el nodo local. En este caso el *PEOO* se mantiene igual. Sin embargo, el costo total no va a incluir el costo adicional de actualizar las réplicas y se debería hacer un análisis adicional si este costo se considera importante para los diseñadores del sistema.
3. Localización: Para hacer el *PEOO* más universal se debería hacer una extensión para poder analizar la localización de los fragmentos en los sitios específicos.

Por las limitaciones del espacio no es posible presentar en este artículo el proceso de modelaje de jerarquías de objetos para obtener la matriz TAUM que corresponde a la matriz AUM del enfoque relacional.

6. REFERENCIAS

- [1] Chakravarthy S., Muthural J., Varadarajan R., y Navathe S.B. *An Objective Function for Vertically Partitioning Relations in Distributed Databases and its Analysis*. Distributed and Parallel Databases, Vol. 2, No. 1, 1993.
- [2] Chu Pai-Cheng. *A transaction Oriented Approach to Attribute Partitioning*. Information Systems, Vol. 17, No.4, 1992.
- [3] Chu W., y Jeong, I.T. *A Transaction Based Approach to Vertical Partitioning for Relational Database Systems*. IEEE Transactions on Software Engineering, Vol. 19, No. 8, Agosto 1993.
- [4] Cornell D., y Yu P. *A Vertical Partitioning Algorithm for Relational Databases*. Proc. Third International Conference on Data Engineering, Febrero 1987.
- [5] Ezeife C.I., y Barker K. *A Comprehensive Approach to Horizontal Class Fragmentation in a Distributed Object Based System*. Technical report, Advanced Database Systems Laboratory, Department of Computer Science, University of Manitoba, Canada, October 1994.
- [6] Gruber O., y Amsaleg L. *Object Grouping in EOS*. M.T. Ozsu, U.Dayal y P.Valduriez (eds.) Distributed Object Management. Morgan Kaufman Publishers, San Mateo, Ca 1994.
- [7] Hammer M., y Niammir B. *A Heuristic Approach to Attribute Partitioning*. Proc. ACM SIGMOD International Conference on Management of Data. Boston, MA, 1979.
- [8] Hoffer J.A. y Severeance D.G. *The Use of Cluster Analysis in Physical Database Design*. Proc. First International Conference on Very Large Data Bases. Framingham, MA, Setiembre 1975.
- [9] Jain A. y Dubes R. *Algorithms for Clustering Data*. Prentice Hall Advanced Reference Series, Englewood Cliffs, NJ 1988.
- [10] Karlapalem K., y Li O. *Partitioning Schemes for Object Oriented Databases*. Technical report. University of Science and Technology, Department of Computer Science Clear Water Bay, Kowloon, Honk Kong, Agosto 1994a.

- [11] Karlapalem K., Navathe A.B. y Morsi M. *Issues in Distribution Design*. M.T. Ozsu, U.Dayal y P.Valduriez (eds.) Distributed Object Management. Morgan Kaufman Publishers, San Mateo, Ca 1994.
- [12] Lin X., Orlowska M. y Zhang Y. *A Graph Based Cluter Approach for Vertical Partitioning in Database System*. Data & Knowledge Engineering, Vol. 11, 1993.
- [13] Navathe S., Ceri G., Wiederhold. y Dou J. *Vertical Partitioning Algorithm for Database Design*. ACM Transaction on Database Systems, Vol.9, No.4, Diciembre 1984.
- [14] Navathe S., y Ra M. *Vertical Partitioning for Database Design: A Graphical Algorithm*. ACM SIGMOD, Portland, June 1989.
- [15] Özsu M. Y Valduriez P. *Principles of Distributed Database Systems*. Prentice Hall, 1991.

Elzbieta Malinowski G.,
emalinow@cariari.ucr.ac.cr