PHÂN MẢNH DỮ LIỆU TRONG THIẾT KẾ CƠ SỞ DỮ LIỆU PHÂN TÁN DỰA VÀO KỸ THUẬT PHÂN CỤM HƯỚNG TRI THỨC

FRAGMENTATION IN DISTRIBUTED DATABASE DESIGN BASED ON KNOWLEDGE-ORIENTED CLUSTERING TECHNIQUE

Lê Văn Sơn[1], **Lương Văn Nghĩa**[2]

[1]*Trường Đại học Sư phạm, Đại học Đà Nẵng; Email: levansupham2004@yahoo.com*
[2]*Trường Đại học Phạm Văn Đồng; Email: nghia.itq@gmail.com*

Tóm tắt – Bài toán tối ưu hóa cơ sở dữ liệu phân tán bao gồm các bài toán: phân mảnh và định vị dữ liệu. Có nhiều phương pháp tiếp cận khác nhau và nhiều thuật toán được đề xuất để giải quyết các bài toán này. Tuy nhiên, độ phức tạp của thuật toán vẫn còn là một thách thức. Trong bài báo này, chúng tôi sử dụng kỹ thuật phân cụm hướng tri thức cho cả hai bài toán phân mảnh ngang và phân mảnh dọc dữ liệu. Độ đo tương tự sử dụng trong hai thuật toán là các độ đo được phát triển từ các độ đo cổ điển. Kết quả thử nghiệm trên các tập dữ liệu nhỏ hoàn toàn trùng khớp với kết quả phân mảnh dựa vào các thuật toán cổ điển. Thời gian thực hiên phân mảnh dữ liệu cũng được giảm đáng kể (mặc dù độ phức tạp thuật toán trong trường hợp tổng quát vẫn chưa thay đổi).

Từ khóa – cơ sở dữ liệu phân tán; phân mảnh; định vị; độ đo tương tự, phân cụm; kỹ thuật phân cụm hướng tri thức.

Abstract – The optimization problem of data fragmentation is requiring to several interrelated problems including: Data fragmentation and Data allocation. Although we had many different algorithms to approach solving problems, the complexity of algorithm is always a big challenge to solve. In this paper, we presented a knowledge-oriented clustering technique that is applying both of vertical fragmentation and horizontal fragmentation problems. Similarity measures are used in both of algorithms which were built in the traditional measures. The experimental result of small data files and the fragmentation result based-on traditional algorithm are similar. The execution time of fragmented data is significantly reduced. (Although, the complexity of algorithm in the general case is still un-changed).

Key words – distributed database; fragmentation; allocation; similarity measures; clustering; knowledge-oriented clustering technique.

1. Đặt vấn đề

Trong môi trường phân tán, mỗi đơn vị dữ liệu (item) được truy xuất tại các trạm (site) thường không phải là một quan hệ mà chỉ là một bộ phận của quan hệ. Vì vậy, để tối ưu hóa quá trình thực hiện các truy vấn, các quan hệ của lược đồ toàn cục *(global scheme)* được phân mảnh thành các đơn vị dữ liệu.

Các loại phân mảnh dữ liệu bao gồm phân mảnh dọc, phân mảnh ngang, phân mảnh hỗn hợp *(mixed)* và phân mảnh suy dẫn *(derivate)*. Hai thuật toán cổ điển gắn liền với phân mảnh ngang và phân mảnh dọc lần lượt là thuật toán PHORIZONTAL và thuật toán BEA [5]. Nhiều tác giả đã đề xuất các thuật toán cải biên hai thuật toán này như Navathe và đồng sự (1984), Cornell và Yu (1987), Chakravarthy và đồng sự (1994), Bellatreche (2000), Schewe (2002),.. Tuy nhiên, độ phức tạp của các thuật toán này là khá lớn, phân mảnh dọc là $O(n^2)$ với n là số lượng thuộc tính, phân mảnh ngang là $O(2^m)$ với m là số bản ghi [5][8].

Trong thời gian gần đây, một số tác giả đã kết hợp giải bài toán phân mảnh và bài toán định vị bằng các thuật toán tối ưu [9][14] hay sử dụng các thuật toán heuristic [1][9], thời gian thực hiện các thuật toán này giảm đáng kể so với các thuật toán cổ điển mặc dù độ phức tạp của giải thuật trong trường hợp tổng quát vẫn chưa được cải thiện. Sử dụng kỹ thuật luật kết hợp trong khai phá dữ liệu để phân mảnh dọc dữ liệu đã được đề cập [10], tuy vậy các kỹ thuật khai phá dữ liệu khác cũng chưa được các tác giả quan tâm ứng dụng.

Trong bài báo này, chúng tôi đề xuất sử dụng thuật toán phân cụm hướng tri thức cho 2 bài toán phân mảnh dọc và phân mảnh ngang. Các độ đo tương đồng *(similarity)* được phát triển dựa trên các độ đo đã có trong các thuật toán cổ điển và khai phá dữ liệu [2].

Nội dung chính của bài báo được tổ chức như sau: Các khái niệm cơ sở được trình bày trong Mục 2. Mục 3, trình bày thuật toán phân cụm hướng tri thức. Mục 4, 5 lần lượt trình bày thuật toán phân mảnh dọc, phân mảnh ngang đề xuất. Mục 6 là phần kết luận.

2. Một số khái niệm cơ sở

2.1. Phân mảnh dọc

Phân mảnh dọc là phân rã tập thuộc tính của lược đồ quan hệ R thành các lược đồ con $R_1, R_2, .., R_m$, sao cho các thuộc tính trong mỗi lược đồ con là thường được truy vấn cùng nhau. Để thể hiện mức độ hay cùng được truy vấn cùng nhau, Hoffer và Severance đưa ra khái niệm ái lực thuộc tính *(attribute affinity)* [13].

Nếu $Q = q_1, q_2, .., q_m$ là tập các ứng dụng, $R(A_1, A_2, .., A_n)$ là một lược đồ quan hệ. Mối quan hệ giữa ứng dụng q_i và thuộc tính A_j được xác định bởi giá trị sử dụng [2]:

$$\text{use}(q_i, A_j) = \begin{cases} 1, A_j \text{ có tham gia } q_i \\ 0, A_j \text{ không có tham gia } q_i \end{cases} \quad (1)$$

Đặt $Q(A, B) = q \in Q | \text{use}(q, A).\text{use}(q, B) = 1$. Ái lực giữa 2 thuộc tính A_i, A_j:

$$\text{Aff}(A_i, A_j) = \sum_{q \in Q(A_i, A_j)} \left(\sum_{\forall S_l} \text{ref}_l(q) * \text{acc}_l(q) \right) \quad (2)$$

Trong đó, $\text{ref}_l(q)$: số lần cặp thuộc tính (A_i, A_j) được tham chiếu trong ứng dụng q tại trạm Sl; accl(q): tần số truy xuất ứng dụng q đến các thuộc tính (A_i, A_j) tại trạm S_l. Thuật toán BEA thực hiện gồm 2 giai đoạn chính:

(1). Hoán vị hàng, cột của ma trân ái lực thuộc tính sinh ra 1 ma trận ái lực tụ thuộc tính CA (*Cluster Affintity matrix*) có số đo ái lực chung AM (*global affinity measure*) là lớn nhất [5].

(2). Tìm điểm phân hoạch tập thuộc tính từ ma trận tụ thuộc tính CA bằng phương pháp vét cạn, sao cho [8]:

$Z = CTQ * CBQ - COQ^2$ đạt cực đại, với:

$$CTQ = \sum_{q \in TQ} \sum_{\forall Sj} ref_j(q_j) acc_j(q_i)$$

$$CBQ = \sum_{q \in BQ} \sum_{\forall Sj} ref_j(q_j) acc_j(q_i)$$

$$COQ = \sum_{q \in OQ} \sum_{\forall Sj} ref_j(q_j) acc_j(q_i)$$

	A_1	A_2	..	A_i	A_{i+1}	..	A_n
A_1							
..			**TA**				
A_i							
A_{i+1}							
						BA	
A_n							

Hình 1: *Ma trận tụ thuộc tính CA*

Trong đó,

$AQ(q_i) = \{A_j | use(q_i, A_j) = 1\};$
$TQ = \{q_i | AQ(q_i) \subseteq TA\};$
$BQ = \{q_i | AQ(q_i) \subseteq BA\};$
$OQ = Q\{TQ \cup BQ\}.$

Độ phức tạp của thuật toán tỉ lệ với n^2.

2.2. Phân mảnh ngang

Phân mảnh ngang là phân chia tập các bản ghi thành các tập bản ghi nhỏ hơn. Phân mảnh ngang dựa vào các điều kiện truy vấn được thể hiện qua các vị từ đơn giản có dạng:

$P : A_j \theta$<giá trị>

Đặt $P_r = \{p_1, p_2, .., p_k\}$ là tập các vị từ đơn giản được trích ra từ tập các ứng dụng. Một hội vị từ được xây dựng từ Pr có dạng:

$$p_1 * \wedge p_2 * \wedge .. \wedge p_n * \quad (3)$$

Trong đó p_i*là vị từ mang 1 trong giá trị là pi hay $\neg p_i$

Thuật toán PHORIZONTAL sử dụng các hội vị từ có thể xây dựng từ P_r, để tìm các điều kiện phân mảnh ngang dữ liệu [11]. Quan hệ r(R) sẽ được phân mảnh thành $\{r_1(R), r_2(R), .., r_k(R)\}$, với $r_i(R) = \sigma_{Fi}(r(R))$, $1 \leq i \leq k$; Fi là một vị từ hội sơ cấp (m_j).

2.3. Hệ thống thông tin và quan hệ không phân biệt

Hệ thống thông tin là một cặp SI=(U, A), trong đó U là tập hữu hạn các đối tượng U=$\{t_1, t_2, .., t_n\}$, A là tập hữu hạn khác rỗng các thuộc tính.

Một quan hệ tương đương (quan hệ 2 ngôi thỏa các tính chất phản xạ, đối xứng và bắc cầu) xác định trên U được gọi là một quan hệ không phân biệt trên U.

3. Thuật toán phân cụm hướng tri thức

Thuật toán phân cụm hướng tri thức KO-Knowledge-Oriented Clustering dựa vào lý thuyết tập thô đầu tiên được đề xuất bởi nhóm tác giả Shoji Hirano and Shusaku Tsumoto (2001) [12]. Đây là thuật toán phân cụm tự động xác định số cụm dựa vào bộ dữ liệu khảo sát. Ý tưởng chính của thuật toán phân cụm này gồm 2 giai đoạn [3]:

1. Xây dựng 1 quan hệ tương đương ban đầu trên tập các đối tượng cần phân cụm.
2. Hiệu chỉnh các quan hệ tương đương bằng cách sử dụng một ngưỡng Tk dựa trên độ không phân biệt. Quá trình lặp cập nhật lại Tk cho đến khi thu được phân cụm tốt nhất.

Thuật toán này được nhóm tác giả C.L Bean, C.Kambhampati hiệu chỉnh và thử nghiêm (2008) [4] (*bài báo của nhóm tác giả chỉ xây dựng lại quan hệ tương đương ban đầu dựa vào ý tưởng đường "đẳng trọng" so với cách xây dựng dựa trên gradient của nhóm tác giả Shoji Hiran, Shusaku Tsumoto*). Các kết quả thử nghiệm của 2 nhóm tác giả trên nhằm minh họa cho thuật toán, chưa đưa ra các ứng dụng trong thực tế.

Sử dụng thuật toán này để phân mảnh dữ liệu, chúng tôi đã đề xuất quan hệ tương đương ban đầu dựa trên khoảng cách trung bình giữa các đối tượng.

Thuật toán phân cụm hướng tri thức chúng tôi sử dụng cụ thể như sau:

Input: U= Tập các đối tượng cần phân cụm

(Mỗi đối tượng phải được mô tả các thông tin cần thiết để xây dựng độ tương tự).

Output: Các phân cụm *(tương ứng với các phân mảnh dữ liệu)*

Method:

Bước 1: *Xây dựng ma trận độ tương tự S=S(t_i, t_j) giữa tất cả các cặp đối tượng(t_i, t_j).*

Bước 2: *Chỉ định một quan hệ không phân biệt ban đầu Ri cho từng đối tượng. Tổng hợp để có được một phân cụm ban đầu.*

Bước 3: *Xây dựng ma trận bất khả phân biệt* $\Gamma = \nu(t_i, t_j)$ *để đánh giá chất lượng phân cụm.*

Bước 4: *Sửa đổi phân cụm theo một quan hệ bất khả phân biệt mới* R_i^{mod} *cho từng đối tượng để đạt được một phân cụm sửa đổi.*

Bước 5: *Lặp lại bước 3 và 4 cho đến khi thu được một phân cụm ổn định.*

Chi tiết của thuật toán có thể tham khảo [4][12]. Điểm cần lưu ý là chúng tôi đề xuất cách xây dựng quan hệ không phân biệt ban đầu khác với 2 nhóm tác giả Shoji Hirano, Shusaku Tsumoto và C.L Bean, C.Kambhampati như sau:

Quan hệ không phân biệt ứng với thuộc tính thứ i:

$R_i = \{(t_i, t_j) \in U \times U : d(t_i, t_j) \leq Th_i$ với $j = 1, 2, \ldots, n\}$

Trong đó $d(t_i, t_j)$ là khoảng cách giữa 2 đối tượng tham gia phân cụm.

Ngưỡng Th_i được xác định như sau:

$$Th_i = \left[\sum_{j=1, j\neq i}^{n} (1 - s(t_i, t_j)) \right] / (n-1) \qquad (4)$$

Với $s(t_i, t_j)$ là độ tương tự của 2 đối tượng t_i, t_j. [2].

4. Phân mảnh dọc hệ cơ sở dữ liệu phân tán dựa vào thuật toán phân cụm hướng tri thức

Để chuyển bài toán phân mảnh dọc trọng hệ cơ sở dữ liệu phân tán, các giả thiết bài toán được chuyển đổi sang giả thiết bài toán phân cụm dựa vào các khái niệm sau:

4.1. Thuộc tính và Vector đặc trưng tham chiếu

Định nghĩa 1. Độ đo tham chiếu của giao tác qi với thuộc tính Aj ký hiệu M(qi, Aj) hay Mij là tần suất giao tác qi tham chiếu đến thuộc tính A_j được xác định bởi giá trị

$M(q_i, A_j) = M_{ij} = use(q_i, A_i) * f_i$

Với: f_i là tần suất thực hiện của giao tác q_i và $use(q_i, A_j)$ xác định bởi công thức (1).

Định nghĩa 2. Vector đặc trưng tham chiếu VAj của thuộc tính Aj ứng với tham chiếu của các giao tác $(q_1, q_2, .., q_m)$ được xác định như sau:

VA_j=	q_1	q_2	...	q_m
	M_{1j}	M_{2j}	...	M_{mj}

4.2. Độ đo tương đồng của 2 thuộc tính

Định nghĩa 3. Độ đo tương đồng của 2 thuộc tính A_k, A_l có 2 vector đặc trưng tham chiếu tương ứng với bộ các giao tác $(q_1, q_2, .., q_m)$:

$VA_k = (M_{1k}, M_{2k}, .., M_{mk})$
$VA_l = (M_{1l}, M_{2l}, .., M_{ml})$

được xác định bởi độ đo cosin:

$$s(A_k, A_l) = \frac{VA_k * VA_l}{\|VA_k\| * \|VA_l\|} = \frac{\sum_{i=1}^{m} M_{ik} * M_{il}}{\sqrt{\sum_{i=1}^{m} M_{ik}^2} * \sqrt{\sum_{i=1}^{m} M_{il}^2}} \qquad (5)$$

4.3. Phân mảnh dọc dựa vào thuật toán phân cụm hướng tri thức

Để minh họa phân mảnh dọc dựa vào thuật toán phân cụm hướng tri thức, sử dụng lại giả thiết ví dụ bài toán phân mảnh dọc dựa vào thuật toán BEA được trình bày trong [2]:

- Tập các thuộc tính Att = $\{A_1, A_2, A_3, A_4\}$
- Tập các giao tác Q = $\{q_1, q_2, q_3, q_4\}$
  Ma trận sử dụng:

	A_1	A_2	A_3	A_4
q_1	1	0	1	0
q_2	0	1	1	0
q_3	0	1	0	1
q_4	0	0	1	1

- Tập tần suất thực hiện ứng với tập các giao tác $\{q_1, q_2, q_3, q_4\}$, và F = $\{f_1, f_2, f_3, f_4\}$ = {45, 5, 75, 3}.

Từ giả thiết ta có các vector đặc trưng tham chiếu:

	q_1	q_2	q_3	q_4
VA_1=	45	0	0	0
VA_2=	0	5	75	0
VA_3=	45	5	0	3
VA_4=	0	0	75	3

Ma trận độ tương tự $S_{4x4} = (s(A_k, A_l))_{k=1,4;l=1,4}$

	A_1	A_2	A_3	A_4
A_1	1	0	0.9918	0
A_2		1	0.0073	0.9970
A_3			1	0.0026
A_4				1

Kết quả phân mảnh bằng thuật toán phân cụm hướng tri thức thu được:

Cụm	Tập thuộc tính
1	$\{A_1, A_3\}$
2	$\{A_2, A_4\}$

Nhận xét:

Kết quả phân mảnh này trùng khớp với kết quả phân mảnh dọc theo thuật toán BEA.

5. Phân mảnh ngang hệ cơ sở dữ liệu phân tán dựa vào thuật toán phân cụm hướng tri thức

Tương tự như phân mảnh dọc trong hệ cơ sở dữ liệu phân tán, việc chuyển đổi giả thiết phân mảnh ngang [2] từ thuật toán PHORIZONTAL dựa trên các khái niệm cơ sở sau:

5.1. Vector hóa các bản ghi của một quan hệ

Xét quan hệ r(R)=$\{T_1, T_2, .., T_l\}$, tập các vị từ đơn giản rút trích từ các ứng dụng trên r(R) là $Pr = \{Pr_1, Pr_2, .., Pr_m\}$. Vector hóa nhị phân các bản ghi theo qui tắc:

	Pr_1	Pr_2	...	Pr_j	...	Pr_m
T_1	a_{11}	a_{12}	...	a_{1j}	...	a_{1m}
...	...	...	...	...	...	...
T_i	a_{i1}	a_{i2}	...	a_{ij}	...	a_{im}
...	...	...	...	...	...	...
T_l	a_{l1}	a_{l2}	...	a_{lj}	...	a_{lm}

$$\forall a_{ij} = \begin{cases} 1, \text{ khi } Ti[Pr_j] = true \\ 0, \text{ khi } Ti[Pr_j] = false \end{cases}$$

5.2. Độ đo tương đồng của 2 vector nhị phân

Xét 2 vector x_i và x_j, được biểu diễn bằng các biến nhị phân. Giả sử các biến nhị phân có cùng trọng số. Ta có bảng sự kiện như Bảng 1. Trong đó q là số các biến nhị phân bằng 1 đối với cả 2 vector x_i và x_j, s là số các biến nhị phân bằng 0 đối với x_i nhưng bằng 1 đối với x_j, r là số các biến nhị phân bằng 1 đối với x_i nhưng bằng 0 đối với x_j, t là số các biến nhị phân bằng 0 đối với cả 2 vector x_i và x_j [2].

Bảng 1: *Bảng sự kiện cho biến nhị phân*

		Đối tượng j		
		1	***0***	*Tổng*
Đối tượng i	***1***	q	r	q+r
	0	s	t	s+t
	Tổng	q+s	r+t	p

Sự khác nhau của 2 vector x_i và x_j dựa trên các biến nhị phân đối xứng *(symmetric binary dissimilarity)* là:

$$d(x_i, x_j) = \frac{r + s}{q + r + s + t} \quad (6)$$

Sự khác nhau của 2 vector x_i và x_j dựa trên các biến nhị phân bất đối xứng *(asymmetric binary dissimilarity)*

$$d(x_i, x_j) = \frac{r + s}{q + r + s} \quad (7)$$

Độ đo tương đồng *(similarity)* giữa 2 vector x_i và x_j, được xác định bởi hệ số Jaccard:

$$sim(x_i, x_j) = 1 - d(x_i, x_j) \quad (8)$$

5.3. Phân mảnh ngang dựa vào thuật toán phân cụm hướng tri thức

Sử dụng lại giả thiết ví dụ bài toán phân mảnh ngang dựa vào thuật toán PHORIZONTAL được trình bày trong [2]:

Giả sử có một quan hệ Emp

	ENO	ENAME	TITLE
T_1	E_1	Jjoe	Elect-Eng
T_2	E_2	M.Smith	Syst-Analyst
T_3	E_3	A.Lee	Mech-Eng
T_4	E_4	J.Smith	Programmer
T_5	E_5	B.Casey	Syst-Analyst
T_6	E_6	L.Chu	Elect-Eng
T_7	E_7	R.David	Mech-Eng
T_8	E_8	J.Jone	Syst-Analyst

Xét 2 vị từ đơn giản:

- p_1=(TITLE>"Programmer") và
- p_2=(TITLE<"Programmer") với quy tắc so sánh chuỗi theo thứ tự a,b,c...

Vector hóa các bản ghi theo 2 vị từ p_1 và p_2:

	p_1	p_2
T_1	1	0
T_2	0	1
T_3	1	0
T_4	0	0
T_5	0	1
T_6	1	0
T_7	1	0
T_8	0	1

5.4. Kết quả phân mảnh ngang quan hệ Emp theo thật toán phân cụm hướng tri thức

Cụm	Tập các bản ghi
1	T_1, T_3, T_6, T_7
2	T_2, T_5, T_8
3	T_4

Nhận xét: Kết quả phân mảnh này trùng khớp với kết quả phân mảnh ngang theo thuật toán PHORIZONTAL.

6. Kết luận

Trong bài báo này chúng tôi trình bày giải pháp phân mảnh dọc và ngang của hệ cơ sở dữ liệu phân tán dựa vào thuật toán phân cụm hướng tri thức. Với giải pháp này chúng tôi đã đề xuất được cách chuyển đổi giả thiết các bài toán phân mảnh cổ điển trở về giả thuyết cho bài toán phân cụm.

Trong thuật toán phân cụm hướng tri thức, chúng tôi có đề xuất cách xây dựng quan hệ tương đương ban đầu dựa vào ngưỡng khoảng cách, đây là điểm khác biệt với 2 thuật toán của 2 nhóm tác giả Shoji Hirano, Shusaku Tsumoto và C.L Bean, C.Kambhampati đã đề xuất trước đó.

Kết quả thử nghiệm trên các bộ dữ liệu trong [13] cho thấy kết quả trùng lắp với kết quả có được từ 2 thuật toán phân mảnh cổ điển là PHORIZONTAL và thuật toán BEA. Ngoài dữ liệu thử nghiệm như đã trình bày, chúng tôi còn thử nghiệm trên một số bộ dữ liệu khác, kết quả có được cũng tương đồng với 2 thuật toán cổ điển nêu trên.

Tuy các bộ cơ sở dữ liệu phân tán thử nghiệm còn nhỏ nhưng lại phù hợp với các thử nghiệm trên [2] cho các phân mảnh cổ điển, đồng thời so sánh được với kết quả trên thuật toán phân cụm hướng tri thức đã đề xuất. Thời gian tới, chúng tôi sẽ sưu tập các bộ dữ liệu đã được công bố có kích thước lớn hơn để có thử nghiệm so sánh tính khả dụng của giải pháp đề xuất với việc kết hợp các kỹ thuật phân cụm khác dựa vào thuật toán di truyền, tiến hóa hay các thuật toán heuristic kết hợp với các công cụ toán học như lý thuyết tập thô, tập mờ [3][4] để đánh giá và chỉ ra các giải pháp phân mảnh dữ liệu với hiệu năng cao.

Tài liệu tham khảo

[1] Adrian Runceanu, Towards Vertical Fragmention in Distributed Databases, *International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE 2007) Conference*; 01/2007.

[2] Lương Văn Nghĩa (2013), Phân đoạn dọc, ngang trong thiết kế cơ sở dữ liệu phân tán dựa trên kỹ thuật phân cụm, *Tạp chí Khoa học và Công nghệ, Đại học Đà Nẵng*, số 3(64).2013.

[3] Lương Văn Nghĩa (2012), Khai phá dữ liệu theo tiếp cận tập thô nhằm tìm thuộc tính hạt nhân và chọn đặc trưng trên tập cơ sở dữ liệu lớn, *Tạp chí KH&CN, ISSN 0866-7659, Đại học Phạm Văn Đồng*, số (01), 12/2012, pp 46-54.

[4] C.L Bean, C.Kambhampati (2008), Automonous Clustering Using Rough Set Theory, *International Journal of Automation and Computing*, Vol.5 (No.1). pp. 90-102. ISSN 1476-8186.

[5] S. Chakravarthy, J. Muthuraj, R. Varadarajan, and S. Navathe, Anobjective function for vertically partitioning relations in distributed databases and its analysis, *Tech Rep. UF-CIS-TR-92-045*, 1994.

[6] Hui Ma, Klaus-Dieter Schewe and Markus Kirchberg, A Heuristic Approach to Vertical Fragmentation, *Proceedings of the 2007 conference on Databases and Information Systems IV: Selected Papers from the Seventh International Baltic Conference DB&IS'2006*, page 103-116.

[7] I. Lungu, T. Vatuiu, A. G. Fodor (2006), Fragmentation solutions used in the projection of Distributed Database System, P*roceedings of the 6th International Conference "ELEKTRO 2006"*, pp. 44-48, Edis-Zilina University Publishers.

[8] Navathe S, Ceri S, Wiederhold G, Dou J (1984), Vertical partitioning algorithms for database design, A*CM Trans Database Syst*, 9(4).

[9] Marwa F. F. Areed, Ali I.El-Dosouky, Hesham A. Ali, A Heuristic Approach for Horizontal Fragmentation and Allocation in DOODB, infos2008.fci.cu.edu.eg/infos/DB-02-P009-016.pdf.

[10] Narasimhaiah Gorla, Pang Wing Yan Betty, Vertical Fragmentation in Databases Using Data-Mining Technique, www.irmainternational.org/chapter/vertical-fragmentation-databases-usingdata/40404/.

[11] Shahidul Islam Khan, A. S. M. Latiful Hoque (2010), A New Technique for Database Fragmentation in Distributed Systems, *International Journal of Computer Applications (0975 – 8887)*, pp. 20-24, Volume 5– No.9.

[12] Shoji Hirano and Shusaku Tsumoto, A Knowledge-Oriented Clustering Technique Based on Rough Sets, *Computer Software and Applications Conference,* 2001. COMPSAC 2001.

[13] Tamer O., Valduriez P.. (1999), Principles of Distributed Database Systems, *Prentice Hall Englewood Cliffs*, Second Edition, New Jersey 07362.

[14] Yin-Fu Huang, Jyh-her Chen, Fragment Allocation in Distributed Database Design, *Journal of Information Science and Engineering*, 491-506(2001).

(BBT nhận bài: 21/12/2013, phản biện xong: 22/01/2014)