An Improvement on Fragmentation in Distribution Database Design Based on Clustering Techniques

Van Nghia Luong¹, Ha Huy Cuong Nguyen² and Van Son Le³
¹Department of Information Technology, Pham Van Dong University, Quang Ngai, Viet Nam
²Department of Information Technology, Quang Nam University, Quang Nam, Viet Nam
³Department of Information Technology, Da Nang University of Education, Da Nang, Viet Nam

Abstract— Distributed database design plays an important role in the design of distributed applications in general. The problem of optimizing distributed database includes the following problems: data fragmentation and data allocation. There are many different approaches to solve these problems. It also means that the design of distributed databases is difficult to carry out. This paper presents two algorithms (vertical and horizontal fragmentation) in distributed databases based on clustering techniques. The similar measures used in the two algorithms are developed from classical metrics. The experimental results show that segmentation results by two algorithms proposed are equivalent to the results of the classical algorithms.

Index Terms— Distributed Database, Fragmentation, Allocation, Similar Measure and Clustering

I. INTRODUCTION

In distributed computing environments, each unit of data (item) which is accessed at the station, (site) is not usually a relationship but part of the relationship. Therefore, to optimize the performance of the query, the relations of global schema are fragmented into items.

There are several types of data fragmentation that are use vertical fragmentation, horizontal fragmentation, mixed fragmentation and derived fragments. Two classical algorithms associated with horizontal fragmentation and vertical fragmentation are PHORIZONTAL and BEA respectively [11]. Many authors have proposed solutions improved the above two algorithms, as Navathe, et al., (1984) [13]. Chakravarthy, et al., (1994) [3]... However, the complexity of this algorithm is quite large, with vertical fragmentation problem is O(n²), where n is the number of attributes and horizontal fragmentation is O(2ᵐ), where m is the number of records [9], [11].

In recent years, several authors have incorporated to solve the problem of fragmentation and positioning, using the optimal algorithm [5]-[6], [10] or using the heuristic method [4], [7]. The execution time of algorithms significantly reduced compared with the classical algorithm.

The used technical association rules in data mining to vertical fragmentation has been mentioned in [8]. However, the data mining techniques do not attract many authors.

In this paper, we use knowledge-oriented clustering techniques for vertical and horizontal fragmentation problem. The measure of similarity was developed based on from the available measure of the classical algorithms in data mining.

In the clustering algorithm based on knowledge-oriented, we propose an algorithm that builds the initial equivalence relation based on the distance threshold. This approach differs from the previous works proposed by Hirano et al., [10] and Bean et al.,[2], in that the proposed algorithm automatically determines the number of clusters based on the data set of survey.

The paper is organized as follows: section 2 presents a brief overview of the basic concepts and the related work. We detail with the proposed vertical and horizontal fragmentation algorithms, in section 3 and section 4, respectively. We then discuss the main contributions of proposed approach in section 5.

II. BASIC CONCEPTS AND APPLICATION

A. Vertical Fragmentation

Vertical fragmentation is the collective decay properties of the relational schema R into the sub schema R₁, R₂, ...Rₘ, such that each attribute in these sub schemas is often accessed together.

To show how often the same queries together, Hoffer and Severance introduced the concept attribute affinity [11].

If \( Q = \{q_1, q_2, ... q_m\} \) is a set of applications, \( R(A_1, A_2, ... A_n) \) is a relational schemas. The relationship between \( q_i \) and attributes \( A_j \) is determined by using the values:

\[
use(q_i, A_j) = \begin{cases} 
1, & A_j \text{ is engaged in } q_i \\
0, & A_j \text{ is not engaged in } q_i 
\end{cases}
\]  

(1)

Put \( \{A_i, A_j\} = \{q \in Q \mid use(q, A_i) \cdot use(q, A_j) = 1\} \). Attribute affinity between \( A_i \) and \( A_j \) is:
Van Nghia Luong et al.

\[
\text{Aff}(A_i, A_j) = \sum_{q \in Q(A_i)} \left( \sum_{q \in S_j} \text{ref}_i(q) \times \text{acc}_j(q) \right) \tag{2}
\]

In particular, \( \text{ref}_i(q) \): the number of pairs of attributes \((A_i, A_j)\) is referenced in the application \( q \) at station \( S_j \); \( \text{acc}_j(q) \): frequency of access to applications \( q \) in station \( S_j \).

BEA algorithm consists of two main phases:

1) Permutations row, column affinity matrix of attribute to obtain the cluster affinity matrix (CA) which has global affinity measure AM (global affinity measure) \([11]\) is the largest.

2) Find the partition of the set of attributes from the matrix CA by exhaustive method, so that:

\[
Z = \text{CTQ} \times \text{CBQ} - \text{COQ} \tag{2}
\]

The complexity of the algorithm is proportional to \( n^2 \).

Vertical fragmentation problem is converted to the clustering problem, based on the following concepts:

1) Attribute and the reference feature vector

**Definition 1:** The reference measure of transaction \( q_i \) on attribute \( A_j \), denoted by \( M(q_i, A_j) \):

\[
M_{ij} = M(q_i, A_j) = \text{use}(q_i, A_j) \times f_i
\]

In which \( M_{ij} \) is the frequency with which transactions \( q_i \) reference to attribute \( A_j \). With \( f_i \) is the frequency of transactions \( q_i \) and \( \text{use}(q_i, A_j) \) is defined by formula (1).

**Definition 2:** \( VA_j \) reference feature vector of attribute \( A_j \) with reference transactions \((q_1, q_2, \ldots, q_m)\) is defined as follows:

\[
VA_j = \begin{bmatrix}
M_{ij} \\
M_{j2} \\
\vdots \\
M_{mj}
\end{bmatrix}
\]

2) The similarity measure of two properties

**Definition 3:** The similarity measure of two attributes \( A_k, A_1 \) has two feature vectors corresponding to the reference transactions \((q_1, q_2, \ldots, q_m)\):

\[
VA_k = (M_{ik}, M_{2k}, \ldots, M_{mk})
\]

\[
VA_1 = (M_{i1}, M_{21}, \ldots, M_{m1})
\]

Is determined by the cosine measure:

\[
s(A_k, A_1) = \frac{\sum_{i=1}^{m} M_{ik} \times M_{il}}{\sqrt{\sum_{i=1}^{m} M_{ik}^2} \times \sqrt{\sum_{i=1}^{m} M_{il}^2}} \tag{3}
\]

**B. Horizontal Fragmentation**

Horizontal fragmentation divided set records into a smaller set of records. Horizontal fragmentation is based on the query conditions, which are expressed through simple predicates of the form: \( A_i < \text{value} \).

Set \( P_i = \{Pr_1, Pr_2, \ldots, Pr_k\} \) is a set of simple predicates extracted from a set of applications. A conjunction of the predicates, which is built from \( P_i \), will have the form:

\[
p_1 \land p_2 \land \ldots \land p_i
\]

Where \( p_i \) is a predicate, which received one of \( p_i \) or \( \neg p_i \) values.

**PHORIzONTAL** algorithm uses the conjunction of the predicates \( p_1 \land p_2 \land \ldots \land p_n \) to find the conditions for horizontal fragmentation of data \([9]\). The relation \( r(R) \) will be fragmented into \( \{r_1(R), r_2(R), \ldots, r_k(R)\} \), with \( r_i(R) = \sigma_{p_i}(r(R)), 1 \leq i \leq k; F_i \) is a predicate, which forms the conjunction of the primary predicates \([9]\).

**C. Technical data clustering**

Data clustering is similar gathering data together into clusters. Some classic clustering algorithms commonly used as K-Means (Mac Queen-1976), K-Medoids (Kaufman and Rousseau 1987), clustering stack area (agglomerative Hierachical Clustering) \([3]\). Depending on the type of data clustering that have the appropriate level of similarity measure.

The similarity measure to the objects represented by the binary data type variable:

Consider two vectors \( x_i \) and \( x_j \), being represented by binary variables. Assuming binary variables have the same weight. We have event tables as Table II. Where \( q \) is the number of binary variables equal to 1 for the two vectors \( x_i \) and \( x_j \), \( s \) is the number of binary variables equal to 0 for \( x_i \) but equal to 1

<table>
<thead>
<tr>
<th>A_1</th>
<th>A_2</th>
<th>..</th>
<th>A_i</th>
<th>A_isl</th>
<th>..</th>
<th>A_n</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>..</td>
<td>TA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A_1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A_isl</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A_n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In which,

\[
AQ(q_i) = \{ A_j \land \text{use}(q_i, A_j) = 1 \};
\]

\[
TQ = \{q_i | AQ(q_i) \subseteq TA \};
\]

\[
BQ = \{q_i | AQ(q_i) \subseteq BA \};
\]

\[
OQ = \{q_i | TQ \land \neg BQ \}
\]
for $x_i$, $r$ is the number of binary variables equal to 1 for $x_i$, but is 0 for $x_j$, $t$ is the number of binary variables equal to 0 for all vectors $x_i$ and $x_j$.

| TABLE II | EVENT TABLE FOR BINARY VARIABLES |
|---|---|---|---|
| Object j | $I$ | $0$ | Sum |
| Object i | $I$ | $q$ | $r$ | $q+r$ |
|          | $0$ | $s$ | $t$ | $s+t$ |
| Sum      | $q+s$ | $r+t$ | $p$ |

- The difference of two vectors $x_i$ and $x_j$ based on the symmetric binary dissimilarity are:

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

(4)

- The similarity measure between two vectors $x_i$ and $x_j$ is defined by the Jaccard coefficient:

$$sim(x_i, x_j) = 1 - d(x_i, x_j)$$

(5)

### III. VERTICAL FRAGMENTATION ALGORITHM IMPROVEMENTS (VFC)

#### A. Proposition

At the relational schema $R(A_1, A_2, \ldots, A_n)$, mapping $s:RxR \rightarrow [0,1]$ defined by the formula:

$$s(A_i, A_j) =
\begin{cases}
1, & \text{when } i = j \\
0, & \text{when } Aff(A_i, A_j) = 0 \\
\frac{Aff(A_i, A_j)}{L}, & \text{when } Aff(A_i, A_j) \neq 0,i \neq j
\end{cases}$$

(6)

Is a similar measure on $R$.

#### B. Vertical fragmentation algorithm improvements (VFC)

**Input:** Relational schema $R(A_1, A_2, \ldots, A_n)$

An affinity matrix properties ($Aff(A_i, A_j)$) compute

**Output:** The vertical fragmentation of the relation $R$

**Method:** Applied techniques clustering of similarity measure (6) combinations $R$.

**Example 1:** Considering the relational schema $R(A_1, A_2, A_3, A_4)$, with Table III. Matrix an affinity following attributes:

| TABLE III | MATRIX AN AFFINITY |
|---|---|---|---|
| $A_1$ | $A_2$ | $A_3$ | $A_4$ |
| $A_1$ | 45 | 0 | 45 | 0 |
| $A_2$ | 0 | 80 | 5 | 75 |
| $A_3$ | 45 | 5 | 53 | 3 |
| $A_4$ | 0 | 75 | 3 | 78 |

### IV. HORIZONTAL FRAGMENTATION ALGORITHM IMPROVEMENTS (HFC)

#### A. Vectorization binary of records

Considering relations $r(R)=\{T_1, T_2, \ldots, T_l\}$, set of simple predicates extracted from applications on $r(R)$ là $Pr=\{Pr_1, Pr_2, \ldots, Pr_n\}$. Vectorized binary of records under the rule:

| TABLE IV | VECTORIZATION BINAR |
|---|---|---|---|---|
| $Pr_1$ | $Pr_2$ | $\ldots$ | $Pr_j$ | $\ldots$ | $Pr_m$ |
| $T_1$ | $a_{11}$ | $a_{12}$ | $\ldots$ | $a_{1j}$ | $\ldots$ | $a_{1m}$ |
| $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ |
| $T_1$ | $a_{l1}$ | $a_{l2}$ | $\ldots$ | $a_{lj}$ | $\ldots$ | $a_{lm}$ |
| $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ |

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

#### B. Horizontal fragmentation algorithm improvements (HFC)

**Input:** Relations $r(R)=\{T_1, T_2, \ldots, T_l\}$ ;

$Pr=\{Pr_1, Pr_2, \ldots, Pr_m\}$: A set of simple predicates

**Output:** The Horizontal fragmentation of the relation $r(R)$ corresponding to $Pr$.

**Method:** Applied techniques clustering of similarity measure (5) combinations $R$

**Example 2:** Assuming there is a relation

| TABLE V | RELATIONS EMP |
|---|---|---|
| ENO | ENAME | TITLE |
| E1 | Joe | Elect-Eng |
| E2 | M.Smith | Syst-Analyst |
| E3 | A.Lee | Mech-Eng |
| E4 | J.Smith | Programmer |
| E5 | B.Casey | Syst-Analyst |
| E6 | L.Chu | Elect-Eng |
| E7 | R.David | Mech-Eng |
| E8 | J.Jone | Syst-Analyst |

Considering two predicate logic association:

$P_1=(\text{TITLE}="Programmer")$;

$P_2=(\text{TITLE}="Programmer")$. 
From association relationships and predicate logic given, the input data set vectorization:

| TABLE VI. VECTORIZATION RECORDS |
|---|---|---|---|---|
| P1 | –P1 | P2 | –P2 |
| E1 | 1 | 0 | 0 | 1 |
| E2 | 0 | 1 | 1 | 0 |
| E3 | 1 | 0 | 0 | 1 |
| E4 | 0 | 1 | 0 | 1 |
| E5 | 0 | 1 | 1 | 0 |
| E6 | 1 | 0 | 0 | 1 |
| E7 | 0 | 1 | 0 | 1 |
| E8 | 0 | 1 | 1 | 0 |

Implementing clustering algorithm on K-Medoid the vectorization with similarity measure (5):
\[
sim(x_i, x_j) = 1 - d(x_i, x_j),
\]
which \(d(x_i, x_j)\) calculated using the formula (4), the results:

| TABLE VII. RESULTS-BASED SEGMENTATION CLUSTERING TECHNIQUES HORIZONTAL |
|---|---|---|
| K=2 | K=3 | K=4 |
| Cluster 1: E1, E3, E6, E7 | Cluster 1: E1, E3, E6, E7 | Cluster 1: E1, E3 |
| Cluster 2: E2, E4, E5, E8 | Cluster 2: E2, E5, E8 | Cluster 2: E2, E5, E8 |
| Cluster 3: E4 | Cluster 3: E4 | Cluster 3: E4 |
| | Cluster 4: E6, E7 | |

V. CONCLUSION

Apart from the results of tests on, try ghiem with data in [1], [2], the algorithm HFC, VFC have matches with the existing algorithm. It is noted that the horizontal segmentation algorithm based on predicate logic assembly, the cluster segment is due to the algorithm automatically determined; HFC algorithms, VFC using classical clustering techniques, the number of clustering is dependent on the user. So should be combined with expert knowledge to choose the appropriate number of clusters.

The complexity of algorithms HFC, VFC is the complexity of the integrated clustering algorithms, this complexity is less than the complexity of the classical algorithm segments in distributed databases.

In future, the two algorithms HFC, VFC will be improved by redefining the relationship between the measured attributes, records; Integrated high-performance algorithms.

REFERENCES

[8] Jiawei Han, Micheline Kambel (2012), Data Mining: Concepts and Techniques, 3rd ed, Mor-gan Kaufmann Publishers.