Vertical Partitioning in Object Oriented Databases Using Intelligent Agents

Rajan John[†] **and** **Dr. V. Saravanan**[††],
Ph.D Research Scholar **Professor & Head**

Department of Computer Applications,
Karunya University, Coimbatore-641 114, INDIA

Summary

Vertical partitioning is an important technique in which attributes of a relation assigned to partitions, is aimed at improving database performance. The complexity of object-oriented databases models due to subclass hierarchy and class composition hierarchy complicates the definition and representation of vertical partitioning of the classes. In this research paper, we propose a new algorithm for vertical partitioning in object oriented databases using intelligent agents based on attributes and methods.

Key words:
Vertical Partitioning, Software agents, object oriented databases.

1. Introduction

Object oriented databases (OODBs) have gained a considerable attention mainly because they reduce the gap between real world concepts and data representation models. The world surrounding us generates various types of data in abundance. The partitioning of related objects should be performed before clustering for an efficient access in object oriented databases. In databases, the clustering of data is needed to store and retrieve related data together. Generally the clustering of data can be divided into two phases. In the first phase partitioning of related data is done. The second phase is rearrange data in the partition block so that data which are more likely accessed together are located closely to increase the performance.

Partitioning in database design is the process of assigning a logical object (relation) from the logical schema of the database to several physical objects (files) in a stored database. Vertical partitioning subdivides the attributes into groups and assigns each group to a physical object. In other words vertical partitioning refers to the dissection of a relation into a set of relations, each containing a subset of attributes of the original relation. Horizontal partitioning subdivides object instances (tuples) into groups, all having the same attributes of the original object. In other words, horizontal partitioning refers to the dissection of a relation so that each smaller relation contains the same number of attributes as the original relation but with different tuples in each partition. Here the physical objects are a result of vertical or horizontal partitioning as horizontal or vertical fragments.

The OO features such as encapsulation, ISA/class-composition hierarchies and the presence of simple and complex methods add to the complexity of the partitioning problem. Vertical partitioning is inherently more difficult than horizontal partitioning because of its large solution space [26]

Several papers published on the clustering of object-oriented databases [1, 3, 15, 16]. All the papers assumed that there were partition blocks already. It is important to derive a partition block prior to clustering. The information for partitioning databases can be given by users. The information from users is not always correct and sufficient. Therefore, an automatic partition of databases is desirable. In this paper the vertical partitioning problem is considered. The research work also presents a method for vertical partitioning in Object Oriented Databases Using Intelligent Agents.

2. Ontologies

Objects are used in a broad sense. The object may represent a single attribute in a relational database or a complex object in an object oriented database.

Object: Object is the principal building blocks of object-oriented programs. Each object is a programming unit consisting of attribute (instance variables) and behavior (instance methods). An object is a software bundle of variables and related methods.

Clusters: Clusters are groups of objects linked together according to some rules. In abstract terms, a cluster can be

Manuscript received October 5, 2008
Manuscript revised October 20, 2008

viewed as a group of "similar" objects: the degree of similarity between members of the same group is higher than the degree of similarity between members of different groups.

Cluster analysis: Cluster analysis is a generic name for techniques that group items based on some similarity criteria between them. Various clustering techniques can be obtained by choosing different similarity measures and grouping procedures. Cluster analysis refers to the generation of groups, or clusters, of objects that fit a set of definitions.

Intelligent Agents: An intelligent agent is a software that assists people and act on their behalf. Intelligent agents work by allowing people to delegate work that they could have done, to the agent software. Agents can perform repetitive tasks, summarize complex data, learn from you and even make recommendations to you.

3. Related Work: Vertical Clustering in Object Oriented Databases

Most of the research related to partitioning has been carried out in the context of relational databases. There are various ways of partitioning a relation include vertical, horizontal and hybrid / mixed [26]. Because of the criticality of the database performance, several researchers have contributed enormously to vertical Partitioning. Database partitioning has been applied in centralized relational databases [4,8,19,25,28], distributed databases [2,5,8,19,22,26], Data Warehouse Design [10,13,18], and Object-Oriented Database design [12,14]. There has been a increasing demand on the performance of object oriented database systems (ODBSs) which resulted in the adoption of partitioning techniques from relational databases. Gorla [14] used genetic algorithm to determine the instance variables that should be stored in each class/ subclass in a subclass hierarchy, so that the total cost of database operations is minimized in the Object Oriented Database.

More recently, Ailamaki, A; Dewitt, D.J.; Hill, M.D and Skounakis M [1] proposed Partition Attributes Across (PAX) model by improving cache performance, while Ramamurthy et al [27] proposed fractured mirrors partitioning scheme based on Decomposition Storage Model and N-ary Storage Model. Fung, Karlapalem, and Li [12] analyze vertical partitioning of classes/ subclasses for class composition hierarchy and subclass hierarchy and develop the associated cost functions for query processing under the cases of large memory and small memory availability. Ng et al [25] proposed a combined vertical partitioning and tuple clustering using genetic algorithm. In addition to vertical [2, 3, 16], horizontal [16, 17] partitioning, there are new ways specific to object oriented databases such as path partitioning [16] and method-induced partitioning [18]. A vertical partitioning has been reported in [16]. In this representation, attributes are partitioned first and then relevant methods are inserted in the partitions afterwards.

Partitioning based on attributes has been studied earlier in [6], [7], [9], [11]. In [9] DBMS that carries out attribute clustering [9]. It was observed [33] that the major obstacle to widespread use of object-oriented systems would be that their execution may be intrinsically inefficient due to excessive overhead. To reduce this overhead, [33] proposed a vertically partitioned structure for design and implementation of object-oriented systems. Because of the gaining popularity of object-oriented databases in [35] proposed class fragmentation and allocation schemes in order to minimize data transfer in distributed object database systems with complex attributes and methods. Hwang and Yang [34] addressed the necessity of component and data distribution in designing a distributed workflow management system (WFMS).

4. Vertical Partitioning Procedure

Object oriented data model supports basic features such as class, encapsulation, inheritance, and unique object identifier. Each object of a class has a state and behavior. The state represented by attributes and behavior is represented by methods. Objects in a class can inherit attributes and methods from other classes. For each class in the object oriented data, it is important to know the following information: (i) its attributes, their classifications (simple or complex) and the referred classes of each (for complex attributes); (ii) its methods, their classifications (simple or complex) and the referred classes of each (for complex methods); and (iii) its relationships, their cardinalities and the referred classes. Simple attributes acquire values from an atomic domain, such as integer or character. Complex attributes can acquire values from the set of OIDs of objects in the database. Complex methods can invoke other methods but simple methods cannot.

A vertical partitioning of a class C in an object-oriented database defines both structural and behavioral properties. The structural properties are represented by a set of instance variables $I= \{i_1, i_2, \ldots i_n\}$ results in a set of vertical class partitioning $V= \{v_1, v_2, v_3, \ldots, v_n\}$ while behavioral properties are embodied by a set of methods $M= \{m_1, m_2, \ldots m_n\}$; the latter are used to access and manipulate objects in class C's warehouse. For the former, each instance variable of an object is instantiated by using a value from its domain class. Vertical class partition V_j has

a non empty subset of instance variables $i^{vj} = \{i^{vj}_1, i^{vj}_2, \ldots i^{vj}_{nj}\}$ and each $i^{vj}q \in I$, $q=1, \ldots, n_j$, where n_j is the number of instance variables in the j th vertical partition.

4.1 Internal Representation of Vertical Fragments

The research work considers a student data with the following attributes. The attributes of the class partitioned vertically based on the dependency factor. The dependency here refers how each attributes are related or closely related to one another. The related attributes taken together then grouped together to form a vertically partitioned class. For example the above class is vertically partitioned into V1, V2 & V3. The vertical partition V1 partitioned vertically based on the personal details of the class student. Similarly partition V2 is partitioned based on the family details, and V3 is partitioned based on the address. Agents are used to identify the related or dependent attribute and partition it appropriately in respective partitions. The following diagrammatic representation shows clearly how these attributes are partitioned:

```
Class Student
{ Registernumber Char [10];
Initial Char [5];
FirstName char [20];
MiddleName char[20];
LastName char[20];
Dob char[20];
Gender char[5];
Country char [20];
Religion char[20];
Denomination char[25];
Language char[25];
FatherName char [25];
FDesignation char[25];
Mothername char[25];
MDesignation char[25];
Phonenumber number [10];
DegreeClass char [30];
Degree char[30];
Area char[30];
Address char[50];
City char[30]; }
```

Figure 1: Student data

4.2 Strategy and Algorithm Used

Two basic forms of vertical partitioning schemes are possible. They are method-based partitioning and attribute based partitioning.

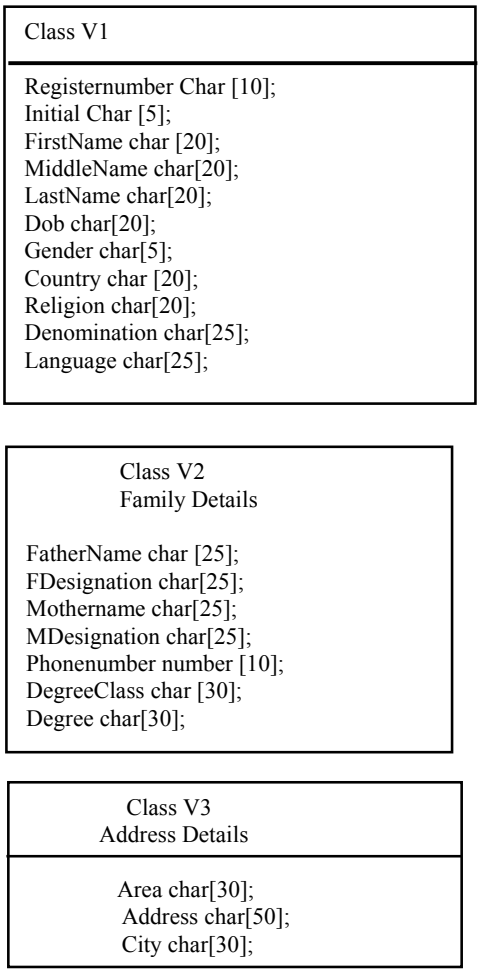

Fig 2: Possible Vertical Class partitions V1, V2 & V3

In the method based partitioning methods in a class becomes basic unit for partitioning. Methods partitioned first and the attributes inserted afterwards. In the attribute based partitioning scheme, the attributes are partitioned first and methods are inserted afterwards.

The first step analyzes the vertical partitioning of a set of classes and its set of operations. Each attributes are accessed in the class, to identify the attributes that are closely related or related to one another and their corresponding methods. For each attribute if there is a relation between the two then it is attached to the existing class. New partitions are created if there are no related methods or attributes. Due to some constraints this research work is based on only attributes. (ie) the related/closely dependent attribute are vertically partitioned to a class.

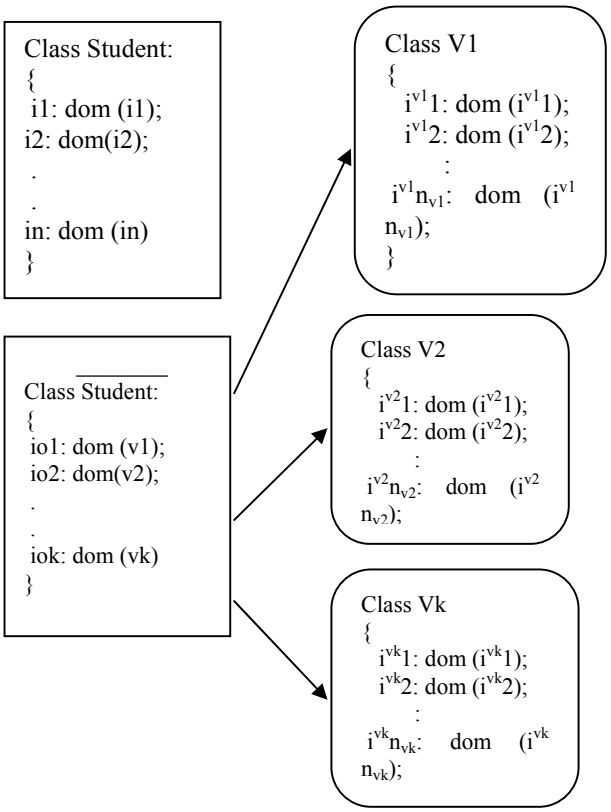

Figure 3: Internal representation of Vertical Class Partitioning of class C

5. Vertical Partitioning Algorithm

The following algorithm is a general algorithm for vertically partition an object-oriented database based on methods or attributes.

5.1 Method based Partitioning Scheme

Methods in a class can be used for partitioning based on the relationship between different methods in a class. In method based partitioning, methods are partitioned in the first place and the attributes are inserted afterwards based on the relations with the methods. In [29] method based partitioning can be further classified:

Method – Method affinity (MMA) is used to select methods for grouping in this scheme. The methods can be grouped together based on either high or low affinities among themselves. The affinities could be estimated on the basis of transaction method access patterns over a predetermined set of transactions or on the basis of complex methods that invoke other methods.

Common attribute affinity (CAA) There is a possibility of grouping the methods which access the same subset of attributes in a partition along with the relevant attributes. Such a grouping can reduce data shipping in an environment where partitions are mapped onto different sites (processors).

```
Function VerticalPartitioning (Cv: set of classes to be vertically partitioned, O: the set of operations)

returns Fv: set of vertical class partitions
begin
 for each Ck that is in Cv do
   for each Oi that is in O do
    for each element (attribute or method) ei of ck that is accessed
      by Oi  do
       for each element (attribute or method) ej of ck that is accessed by Oi do
         if there is a relation between ei and ej then
           link it into already existing class (methods or attributes)
         else
           create a method or class between ei and ej
             and partition the attribute or methods  into a new class
return Fv
end
```

6. Role of Intelligent Agents

An agent is a physical or virtual entity, which runs approximately as follows: [30]

- Which is capable of acting in an environment
- Which can communicate directly with other agents
- Which is driven by a set of tendencies
- Which possesses resources of its own
- Which is capable of perceiving its environment
- Which possesses skills and can offer services

We can make the agent, learn to group the similar attributes into vertical partition based on the class. Also using agents we can select grouping the methods in the vertical partitioning scheme. Intelligent Agents are used in the following steps.

1 Identifying Objects

Agent identifies the object by listing candidate objects found in the written requirements specification. The next step is to identify relevant objects from the application domain. The object identification agent manages the task concerning the object identification.

6.2 Identifying Classes

A class identification agent used to identify the various classes found in the object-oriented databases. The class identification agent begins by listing the available classes

in the object-oriented databases. The class identification agent manages the task concerning the identification of classes.

6.3 Identifying Methods

A method identification agent used to identify the various methods used inside the class / object oriented database. It begins by listing the various methods based on the dependency and the association between them.

6.4 Identifying Attributes

An attribute identification agent begins by listing the attributes in the object oriented databases / class. The next step is to identify relevant attributes. Attributes are properties of individual objects such as name, weight, velocity, or color. The attribute identification agent manages the task concerning the identification of object attributes.

7. Conclusion

Vertical partitioning is a well-known technique in object-oriented database. The challenges of Vertical Partitioning in Object Oriented databases are, to identify the objects and their attributes. This research paper introduces intelligent agents, to automatically manage the various tasks of identification of objects, identification of classes, identification of methods, and identification of attributes. This work defines two different partitioning schemes for object-oriented databases. In this research work intelligent agents is used to vertically partition the object-oriented databases.

References

[1]. Ailamaki, A; Dewitt, D.J; Hill M.D, and Skounakis M, " Weaving Relations for Cache Performance" , Proceedings of the 27th VLDB Conference, 2001.

[2] Baiao, F; Mattoso, M and Zaverucha, G., "A Distribution Design Methodology for Object DBMS," Journal of Distributed and Parallel Databases, 16 (6), 2004, 45-90

[3] E.E. Chang, and R.H. Katz, "Exploiting Inheritance and Structure Semantics for Effective Clustering and Buffering in an Object-oriented DBMS," *Proc. of ACM SIGMOD Conference*, pp. 348-357, 1989.

[4] Ceri, S., Navathe, S., and Wiederhold, G., "Distribution Design of Logical Database Schemas", IEEE Trans. Soft. Eng. SE-9, 4, (July 1983)

[5] Cheng, C-H; Lee, W-K; Wong, K-F, "A Genetic Algorithm-Based Clustering Approach for Database Partitioning," *IEEE Transactions on Systems, Man, and Cybernetics,* 32(3), 2002, 215-230.

[6] M. Babad. " A record and file partitioning model communication", ACM 20, Jan 1977.

[7] F. Baiao, "A Methodology and Algorithms for the Design of Distributed Databases using Theory Revision", D.Sc Thesis, COPPE/UFRJ Dec 2001.

[8] Cornell, D.W. and Yu, P.S., "An Effective Approach to Vertical Partitioning for Physical Design of Relational Databases", IEEE Transactions on Software Engineering, 16-2, (Feb 1990)

[9] J. Hoffer, " An integer programming formulation of computer database design problems", Inf. Science, 1976.

[10] Ezeife, C.I., "Selecting and materializing horizontally partitioned warehouse views," Data and Knowledge Engineering, 36, 2001, pp 185-210

[11] S. Navathe, S. Ceri, G. Weiderhold, and J. Dou, " Vertical Partitioning Algorithms for Database Design", ACM Transactions on Database Systems, vol 9, No. 4, 1984.

[12] Fung, C-w; Karlapalem, K. and Li, Q., "An Evaluation of Vertical Class Partitioning for Query Processing in Object-Oriented Databases," *IEEE Transactions on Knowledge and Data Engineering,* JCS&T Vol. 7 No. 3 October 2007,14(5), 2002, 1095-1118.

[13] Furtado, C; Lima, A.A.B.; Pacitti, E; Valduriez, P. and Mattoso, M., "Physical and virtual partitioning in OLAP database cluster," 17th International Symposium on Computer Architecture and High Performance Computing, 2005, pp 143-150

[14] Gorla, N., "An Object-oriented database design for improved performance," Data & Knowledge Engineering, 2001.

[15] J.R. Cheng,. and A.R. Hurson, "Effective clustering of complex objects in object-oriented databases," *Proc. of ACM SIGMOD Conference*, pp. 22-31, 1991.

[16] Karlapalem, K. and Li, Q, "Partitioning schemes for object oriented databases", In proc 5th International Workshop on Research Issues in Data Engineering- Distributed Object Management (RIDE-DOM'95), Taipei, pp, 42-49,1995.

[17] Karlapalem, K, Navathe, S.B, and Morsi, M.M.A, " Issues in distribution design of Object Oriented Databases in Distributed Object Management, Tamer Ozsu, M.et al (eds), Morgan Kaufman Publishers, San Mateo, pp 148-164, 1994.

[18] Karlapalem, K, Li,Q. and Vieweg,S, " Method induced partitioning schemes in object oriented databases, In Proceedings 16th Int. Conf on Distributed Computing Systems, Hong Kong, pp 377-384, 1996.

[19] M.M. Tsangaris, and J.F. Naughton, "A Stochastic Approach for Clustering in Object Bases," *Proc. of ACM SIGMOD Conference*, pp. 12-21, 1991.

[20] S.B. Navathe, S. Ceri, G. Wiederhold and J. Dou, "Vertical Partitioning Algorithms for Database Design", ACM Transactions on Database Systems, Vol 9, No.4, December 1984.

[21] K. Karlapalem and Q.Li, " Partitioning schemes for Object Oriented Databases", in Proceedings RIDE-DOM '95, page 42, 1995.

[22] Yonglei Tao, " Using Expert Systems to Understand Object-Oriented Behaviour", The 26th SISCSE Technical Symposium on Computer Science Education, 1995.

[23] Rebecca Wirfs-Brock, Brian Wilkerson, and Lauren Wiener, "Designing Object-Oriented Software", Prentice Hall, 1990.

[24] Y. Liang, M.A Newton, and H.M. Robinson, " Analysis of Information Systems using Object Oriented Methodologies",

Proceeding of BCS ISM Sg and BSS Joint Conference on the Theory, Use and Integrative Aspects of IS Methodologies, pp 55-70,1993.
[25] Ng, V; Gorla, N.; Law, D.M. and Chan, C.K., "Applying Genetic Algorithms in Database Partitioning," Proceedings of the 2003 ACM Symposium on Applied Computing (SAC) 2003, pp 544-549.
[26] Tamer Ozsu, M. and Valduriez, P, Principles of Distributed Database Systems, Prentice Hall, Englewood Cliffs, NJ, 1991
[27] Ramamurthy, R; Dewitt, D.J. and Su, Q., "A Case for Fractured Mirrors," *Proceedings of the 28th VLDB Conference*, 2002
[28] E. Bertino and L. Martino, "Object-Oriented Database Management Systems : Concepts and Issues," *Computer*, Vol. 24, No. 4, pp. 33-47, April 1991.
[29] Gajanan S, Chinchwadkar and Angela Goh, " An Overview of Vertical Partitioning in Object Oriented Databases", The Computer Journal, Vol 42, No 1, 1999.
[30] Jacques Ferber, " Multi-agent Systems An Introduction to Distributed Artificial Intelligence", Addison –Wesley, 1999.
[31] Ying Liang, Daune West and Frank A Stowell, " An Approach to Object Identification, Selection and Specification in Object Oriented Analysis", Information Systems Journal, Vol 8, No 2, 1998, pp 163-180, Blackwell Science Ltd,1998.
[32] J. Banerjee, W. Kim, S.J. Kim, and J.F. Garza, "Clustering a DAG Databases," *IEEE*
Transactions on Software Engineering , Vol. 14, No. 11, Nov. 1988
[33] Hufnagel, S.P, Browne J.C, "Performance properties of vertically partitioned object-oriented systems", IEEE Transactions on Software Engineering 15 (8), 935-946, 1989.
[34] Hwang, S, Yang, C, "Component and data distribution in a distributed workflow management system,",Proceedings of IEEE Software Engineering Conference, pp 244-251,1998.
[35] Bellatreche, L, Simonet A, " Vertical Fragmentation in distributed object database systems with complex attributes and methods", The Seventh International Workshop on Database and Expert Systems Applications, pp 15-21, 1996.

Rajan John obtained his Bachelors degree in Physics from Bharathidasan University during 1995 and Masters Degree in Computer Applications from Bharathiar University during 1999. He is currently pursuing his PhD in Computer Applications in Data Mining in the Department of Computer Applications, Karunya University, Coimbatore, India. He is working in the area of Automated Data Mining and Intelligent Agents. He worked in Karunya University as a Lecturer for 3 years, and then working at All Nations University, Ghana, West Africa as a Senior Lecturer for 3 years. At Present he is a Full Time Research Scholar in the Department of Computer Applications of School of Science of Humanities in Karunya University, Coimbatore.

Dr. V Saravanan obtained his Bachelors degree in Mathematics from University of Madras during 1996 and Masters Degree in Computer Applications from Bharathiar University during 1999. He has completed his PhD in Computer Science in the Department of Computer Science and Engineering, Bharathiar University during 2004. He specialized on automated and unified data mining using intelligent agents. His research area includes data warehousing and mining, software agents and cognitive systems. He has presented many research papers in National, International conferences and Journals and also guiding many researchers leading to their PhD degree. He has totally 10 years experience in teaching including 3 years as researcher in Bharathiar University. He is the life member of Computer Society of India, Indian Association of Research in Computing Sciences and many professional bodies. At present, he is working as Professor & Head of the Department of Computer Applications of Karunya School of Science and Humanities in Karunya University, Coimbatore.