DECISION TREE CLUSTERING: A COLUMN-STORES TUPLE RECONSTRUCTION

Tejaswini Apte[1], Dr. Maya Ingle[2] and Dr. A.K. Goyal[2]

[1]Symbiosis Institute of Computer Studies and Research
apte.tejaswini@gmail.com
[2]Devi Ahilya Vishwavidyalaya, Indore
maya_ingle@rediffmail.com, goyalkcg@yahoo.com

ABSTRACT

Column-Stores has gained market share due to promising physical storage alternative for analytical queries. However, for multi-attribute queries column-stores pays performance penalties due to on-the-fly tuple reconstruction. This paper presents an adaptive approach for reducing tuple reconstruction time. Proposed approach exploits decision tree algorithm to cluster attributes for each projection and also eliminates frequent database scanning. Experimentations with TPC-H data shows the effectiveness of proposed approach.

GENERAL TERMS

Performance, Clustering, Projection

KEYWORDS

Tuple Reconstruction, Support

1. INTRODUCTION

Tuple reconstruction time is a major factor to affect the query performance in column-stores [1, 2]. For large data requirement, traditional row-id based tuple construction time pays heavy performance penalty. Two materialization strategies for tuple reconstruction exists in literature namely; Early materialization and Late Materialization. Early Materialization is a technique to stitch the columns into partial tuples as early as possible, Late materialization provides significant performance advantages for analytical queries, since there are fewer row-ids to fetch off the disk for each join [16], hence results in significant disk I/O savings for large tables. In late materialization, schedule to materialize columns according to need, requires awareness in the optimizer, execution engine, and cost model during query optimization. For the very large tables, late materialization involves multiple I/O operations other than the join. In contrast, an early materialized plan involves single scan of complete data.

Sundarapandian et al. (Eds) : ICAITA, SAI, SEAS, CDKP, CMCA-2013
pp. 295–303, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3824

Therefore, researching and designing a time effective tuple reconstruction approach adapted to column-stores is of great significance. Exploiting projections to support tuple reconstruction is included in C-Store. A relation is divided according to required attributes by query, into sub-relations called projections. Each attribute may appear in more than one projection and stored on different sorted order. Since all the attributes are not the part of projection, thus tuple reconstruction pays penalty [4].

Decision tree algorithm is a popular approach for classifying data of various classes. A purity function to partition the data space into several different classes is a requirement for Decision tree algorithm. Although, as datasets have no pre-assigned labels, the decision tree partitioning approach is not directly applicable to clustering. The partitioning of data space into cluster and empty regions is carried through CLTree, as it efficiently finds cluster in large high dimensional spaces [15]. Proposed approach inherits CLTree to reduce the tuple reconstruction time by clustering frequently used attributes with available projection techniques.

It has been observed that attribute projection plays vital role for tuple reconstruction. Proposed approach i.e. Decision Tree Frequent Clustering Algorithm (DTFCA) uses some existing projection techniques to reduce tuple reconstruction time. These techniques are discussed in Section 2. Some notations and terminology are necessary to review the correlation amongst query attributes and are discussed in Section 3. Methodology to understand DTFCA is presented in Section 4. Section 5 presents the detail description of DTFCA. To exploit DTFCA, experimental data based on TPC-H schema is used as an input in suitable experimental environment. The experimental results thus obtained are analyzed, and discussed in Section 6. Finally, paper conclude with concluding remarks in Section 7.

2. RELATED WORK

All column-stores databases require tuple reconstruction to process multi-column queries. Literature reveals that much work has been performed to present the materialization strategies and their trade-offs [2]. Column-Stores database series C-Store and MonetDB has gained popularity due to their good performance for analytical queries [4, 5]. Projections are exploited to logically organize the attributes of the base table. Multiple attributes are involved in each projection. The objective of projections is to reduce the tuple reconstruction time. MonetDB uses late tuple materialization. Though partitioning strategies do not guarantee about the better projection, a good solution is to cluster attributes into sub-relations based on the usage patterns [5]. CLTree clustering technique performs clustering by partitioning the data space into dense and sparse regions [15].

The Bond Energy Algorithm (BEA) is used to cluster attributes based on Attribute Usage Matrix [1, 10, 11]. MonetDB proposed self-organization tuple reconstruction strategy in a main memory structure called cracker map [5, 7]. Dividing and combining the pieces of existing cracker map optimize the query performance. But for the large databases, cracker map pays performance penalty due to high maintenance cost for memory [5]. Therefore, the cracker map is only adapted to the main memory database systems such as MonetDB.

3. DEFINITIONS AND NOTATIONS

In relational Online Analytical Processing Applications, the common format of a query from the fact table(s) and dimension tables is depicted as follows:
Select <attribute list>
From <Table list>
Where <condition>
Group by <attribute list>
Order by <attribute list>

Definition 1 Strong Correlation

k attributes A1, A2,…, Ak of relation R are strongly correlated if, and only if they appear in the conjunction in conditional clause and target list of an access query.

Definition 2 Weak Correlation

In a collection of accesses A={a1, a2,…, am}, every ai($1 \leq i \leq m$) is an access to a query. If the correlation among query target attributes is smaller than P, where P is the threshold, the attributes are considered for weak correlation.

In column-stores, two critical tasks are needed to process a query: First, to determine qualifying rows based on the conditions in Where clause, and to generate a set of row identifiers; Second, to merge the qualifying columns of identical row identifiers, and to construct the target rows according to target expression in Select clause. The attributes appear in conjunction in conditional clause and query target are considered to be strongly correlative. The strong correlations in the access list drives the creation of frequent attribute set.

4. METHODOLOGY

This section covers the search space for DTFCA.

4.1 Decision Tree Construction

Divide and conquer strategy has been used to recursively partition the data to build decision tree [15]. In order to obtain purer regions each successive step chooses the cut to partition the space. A commonly used criterion for choosing the best cut is the minimum support. It evaluates every possible value (or cut point) for all projected attributes to find the cut point that gives the best gain (Figure 1).

for each attribute Ai ∈{A1, A2, …, Ad} of the dataset D do
for each value x of Ai in D do

/* each value is considered as a possible cut */
Compute the information gain at x
end
Select the cut that satisfy the best information gain according to minimum support

Figure 1: The procedure to find the appropriate cluster

4.2 Determining Cluster Attributes

This sub-section focuses on determining attributes for tuple reconstruction. A cluster is created for each data point in the original dataset, and introduce some attributes for support greater than minimum support. The number of attributes to be added for the cluster E is determined by the following rule:

If the number of attributes inherited from the parent cluster projection is less than the number of projected attributes in E then
 the number of inherited attributes is increased to E
else
the number of inherited attributes is used for E

The recursive partitioning method will divide the data space until each cluster contains only attribute of a single class, results in very complex tree that partitions the space more than necessary. Hence the decision tree needs to be pruned to produce meaningful clusters. This problem is similar to classification problem, however pruning methods used for classification, cannot be applied for clustering. Subjective measures are required for pruning, since clustering, to certain extent is a subjective task [12, 13]. The tree is pruned using the minimum support values. After pruning, algorithm summarizes the clusters by retrieving only those attributes.

4.3 Scaling-up decision tree algorithm

For the decision tree, whole data must resides in memory. For the large data set, SPRINT, a scalable technique is proposed by decision tree algorithm, for eliminating the need to have the entire dataset in memory [23]. A statistical technique to construct a consistent tree based on small subset of data is discussed in BOAT [22]. Proposed algorithm inherits these techniques to determine the cluster.

5. DECISION TREE FREQUENT CLUSTERING ALGORITHM (DTFCA)

This section describes the implementation of DTFCA in the MonetDB analytic platform [5]. DTFCA is used to determine the clustering of frequent attribute set. DTFCA algorithm combines multiple iterations of the original loop into a single loop body, and rearranges the frequent attribute set.

Let each relation R be grouped into several projections based on empirical knowledge and users' requirement. After the system is used by users for a period, a set of accesses A={a1, a2,…, am} are collected by the system. DTFCA uses mainly a set of accesses to cluster frequently projected attribute-set which are strongly correlated. The input to DTFCA is a variant of AUM (Table 1). Each element in the output forms attributes of a projection. DTFCA clustering results truly reflect the strong correlativity between attributes implied in previous collection of accesses for queries, and conform to the meanings of projection in column-stores.

```
function   DTFCA(String S)
{
/* Decision Tree Frequent Clustering Algorithm*/
```

Input : a relation R(U={A[1],A[2],…,A[n]}) , a collection of strongly correlated accesses A={a1,a2,…,am}, the support threshold min_sup.

Output: clusters of strongly correlated attributes Cl1,Cl2,…,Clk, which holds support no smaller than min_sup.

```
for each access in A      //Processing an element per iteration
{
compute frequent attribute set for support > minimum support;  // Checking for the limit for traversal
for i=0 to N-1
{
string ResF[200];
string ResT[200];
ResT [i] = A[i].getName(); //Getting the item name of tree node for frequent attribute set
strcat(ResF,ResT) //Concat columns to build the tuples
}
visit the matching build tuple to compare keys and produce output tuple;
}
```

6. EXPERIMENT DETAILS

The objective of the experiment is to compare the execution time of existing tuple reconstruction method with DTFCA for TPC-H schema queries on column-stores DBMS.

6.1. Experimental Environment

Experiments are conducted on 2.20 GHz Intel® Core™2 Duo Processor, 2M Cache, 1G Memory, 5400 RPM Hard Drive, Monet DB, a column oriented database and Windows® XP operating system.

6.2. Experimental Data

TPC-H data set is used as the experiment data set, which is generated by the data generator. Given a TPC-H Schema, fourteen different queries are accessed 140 times during a given window period.

6.3. Experimental Analysis

Let us consider a relation with four attributes namely; S_Supplierkey (A1), P_partkey (A2), O_orderdate (A3), and L_orderkey (A4). These attributes are accessed by fourteen different queries of TPC-H schema for varying minimum support. For each attribute AUM has access frequency generated by the system. The access frequency is derived from three parameters namely; Minimum access frequency (Min_fq), Maximum access frequency (Max_fq), and Average access frequency (Avg_fq). Minimum access frequency of query attributes is computed for initial access of attributes. Maximum access frequency of query attributes is computed on the system with more processing of queries. Average frequency is computed on the system with less processing of queries. DTFCA uses access frequency to cluster frequent attribute-set. Frequent attribute-set refers to the set for frequency no smaller than the minimum support.

Table 1: Attribute Usage Matrix

Attr Que	S_supplierkey (A1)				P_partkey (A2)				O_orderdate (A3)				l_orderkey (A4)			
	Min_ fq	Max_ Fq	Ave_ fq	Acc_ Fq	Min_ fq	Max _fq	Ave _fq	Acc _fq	Min_ fq	Max_ Fq	Ave_ Fq	Acc_ Fq	Min_ Fq	Max_ fq	Ave_ Fq	Acc_ Fq
2	1	1	1	100^{123}	1	1	1	100^{123}	0	0	0	0	1	0	0	33^{1}
3	0	0	0	0	0	1	0	33^{1}	1	1	1	100^{123}	0	1	0	33^{1}
4	0	1	0	33^{1}	1	0	0	33^{1}	1	1	1	100^{123}	1	1	1	100^{123}
5	1	0	0	33^{1}	0	0	0	0	1	1	1	100^{123}	0	1	0	33^{1}
6	0	1	0	33^{1}	1	1	1	100^{123}	0	0	0	0	0	0	0	0
7	1	0	0	33^{1}	0	1	0	33^{1}	0	1	0	33^{1}	1	1	1	100^{123}
8	1	0	0	33^{1}	0	1	0	33^{1}	1	1	1	100^{123}	1	1	1	100^{123}
9	0	1	0	33^{1}	1	1	1	100^{123}	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1	0	0	33^{1}	0	1	0	33^{1}
12	0	1	0	33^{1}	0	1	0	33^{1}	0	1	0	33^{1}	1	1	1	100^{123}
16	0	0	0	0	1	1	1	100^{123}	0	1	0	33^{1}	0	1	1	66^{12}
17	0	1	0	33^{1}	0	1	1	66^{12}	0	1	0	33^{1}	0	0	0	0
19	0	0	0	0	1	1	1	100^{123}	0	1	0	33^{1}	1	1	1	100^{123}
21	0	1	0	33^{1}	0	1	0	33^{1}	0	0	0	0	0	1	0	33^{1}

Now, we explain the execution of DTFCA algorithm for different cases of minimum support. We denote the attribute frequency as a candidate of frequent attribute-set by (int_val) xyz, where int_val is the integer value of access frequency. Also, x, y and z represent the case variable numbers respectively. For the given minimum support, the frequent attribute-set will be a combination of four attributes (A1, A2, A3, and A4).

Case-Study

Let Case 1, Case 2 and Case 3 denote three minimum support case variables. For DTFCA experiment analysis, let the minimum support values for these variables be 30, 50, and 60 respectively. For the implementation of Case 1, user gains the control to determine the strong correlation among attributes, and hence formulating the frequent attribute-set. By referring to Table 1, frequent attribute-sets generated are higher than other cases i.e. out of fourteen pairs or patterns, thirteen patterns were frequent. This result may be considered as moderate and reduces tuple reconstruction time for column-stores. However, with Case 2 and Case 3, generated frequent attribute sets are four and one respectively. The clustering result of DTFCA is more in conformity with the idea of projection in column-stores.

The experiments are performed on TPC-H dataset queries with the same selectivity and varying minimum support. As shown in Figure 2 (horizontal axis denotes the minimum support, vertical axis denotes the execution time), the execution time of TPC-H query schema is inversely proportional to minimum support, and the system may perform well for query attributes which are strongly correlative (refer Section 3). As shown in Figure 3, the execution time under DTFCA is much lower than traditional tuple reconstruction method; hence DTFCA could minimize the tuple reconstruction time.

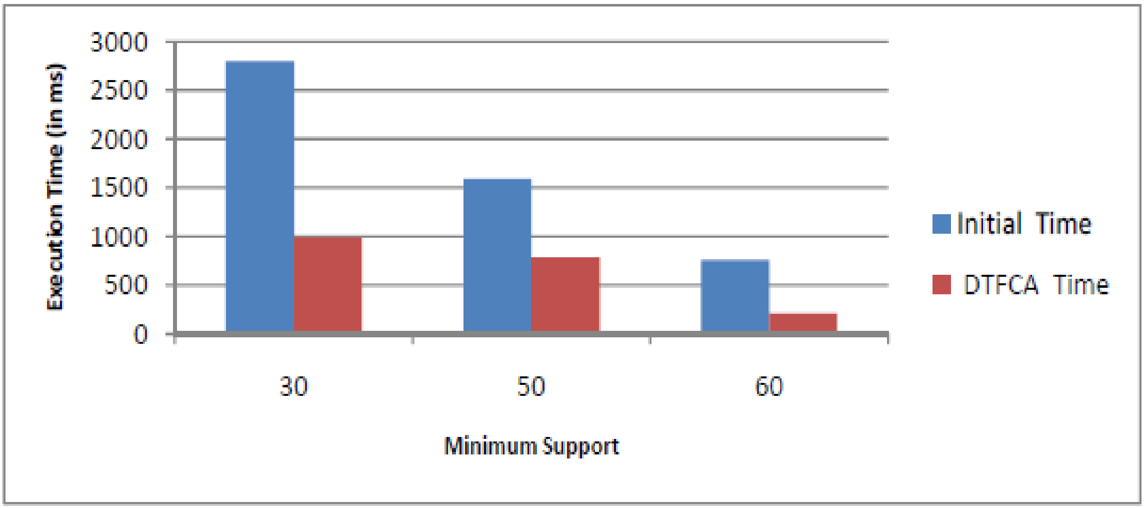

Figure 2: Minimum Support Time Cost

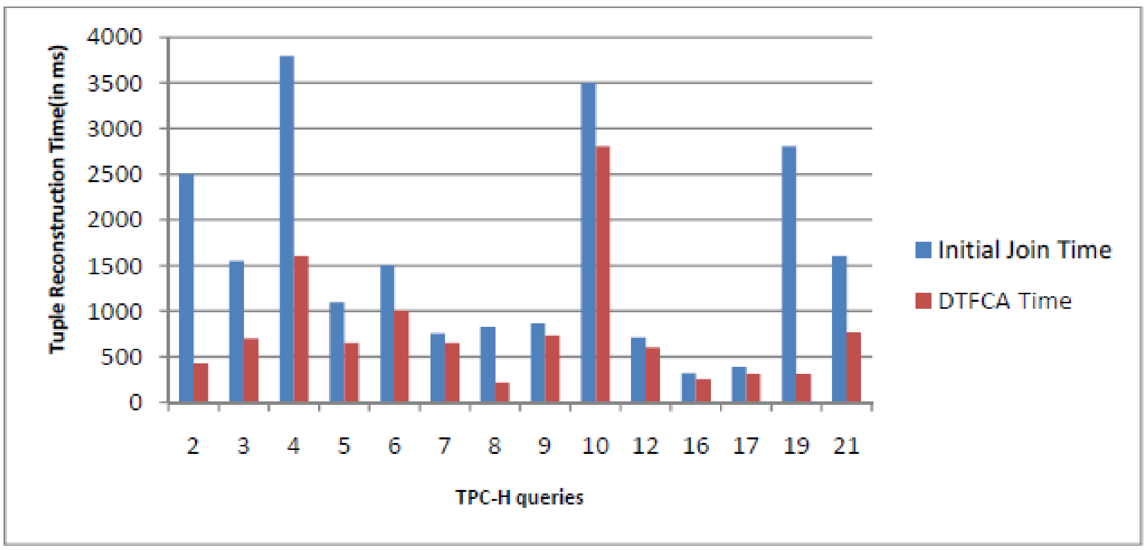

Figure 3: Tuple Reconstruction Time Cost

7. CONCLUSION

DTFCA approach, exploits decision tree to cluster frequently accessed attributes of a relation. All the attributes in each cluster form a projection. The experiment shows that proposed approach is beneficial to cluster projection in column-stores and hence reduces the tuple reconstruction time. The output of DTFCA is not a partition, but a group of attributes for the clustering into column-stores.

REFERENCES

[1] D. J. Abadi."Query execution in column-oriented database systems,". MIT PhD Dissertation, 2008.
[2] S. Harizopoulos, V.Liang, D.J.Abadi, and S.Madden, "Performance tradeoffs in read-optimized databases," Proc. the 32th international conference on very large data bases, VLDB Endowment, 2006, pp. 487-498.
[3] D. 1 Abadi, D. S. Myers, D. 1 DeWitt, and S. Madden, "Materialization strategies in a column-oriented DBMS," Proc. the 23th international conference on data engineering, ICDE, 2007, pp.466-475
[4] M. Stonebraker, D.J.Abadi, A.Batkin, X. Chen, and M. Cherniack, et al. "C-Store: a column-oriented DBMS," Proc. the 31 st international conference on very large data bases, VLDB Endowment, 2005, pp. 553-564
[5] P.Boncz, M.Zukowski, and N.Nes, "MonetDB/X100: hyper-pipelining query execution," Proc. CIOR 2005, VLDB Endowment, 2005, pp. 225-237
[6] R. MacNicol and B. French, "Sybase IQ multiplex-designed for analytics," Proc. the 30th international conference on very large data bases, VLDB Endowment, 2004, pp. 1227-1230
[7] S. Idreos, M.L.Kersten, and S.Manegold, "Self organizing tuple reconstruction in column-stores," Proc. the 35th SIGMOD international conference on management of data, ACMPress, 2009, pp. 297-308
[8] G. P. Copeland, and S.N. Khoshafian, "A decomposition storage model," Proc. the 1985 ACM SIGMOD international conference on management of data, ACM Press, 1985, pp. 268-279
[9] D.J. Abadi, S.R.Madden, and M.C. Ferreira,"Integrating compression and execution in column-oriented database systems," Proc. the 2006 ACM SIGMOD international conference on management of data, ACM Press, 2006, pp. 671-682
[10] H. I. Abdalla. "Vertical partitioning impact on performance and manageability of distributed database systems (A Comparative study of some vertical partitioning algorithms)". Proc. the 18th national computer conference 2006, Saudi Computer Society.
[11] S. Navathe, and M. Ra. "Vertical partitioning for database design: a graphical algorithm". ACM SIGMOD, Portland, June 1989:440-450
[12] R. Agrawal, T. Imielinski, and A.Swami. "Mining association rules between sets of items in large databases". Proc. of the ACM SIGMOD conference on manage of data May 1993:207-216
[13] R. Agrawal and R. Srikanr."Fast algorithms for mining association rules,". Proc. the 20th international conference on very large data bases,Santiago,Chile, 1994.
[14] Electronic Publication: P.O'Neil and X. Chen, The Star Schema Benchmark Revision 3, Jan. 2007, doi: 10. I. 1.80.4071 VIO-589.
[15] Bing Liu, Yiyuan Xia, Philip S. Yu "Clustering Through Decision Tree Construction, CIKM 2000, McLean, VA USA
[16] D. Abadi, D. Myers, D. DeWitt, and S. Madden, "Materialization strategies in a column-oriented DBMS," in Proc. ICDE, 2007, pp. 466–475
[17] W. Yan and P.-A. Larson, "Eager aggregation and lazy aggregation," in VLDB, 1995, pp. 345–357.
[18] J. R. Quinlan. C4.5: program for machine learning.Morgan Kaufmann, 1992.
[19] R. Dubes and A. K. Jain, "Clustering techniques: the user's dilemma." Pattern Recognition, 8:247-260, 1976.
[20] A. K. Jain and R. C. Dubes. Algorithms for clustering data. Prentice Hall, 1988.
[21] J. Gehrke, R. Ramakrishnan, V. Ganti. "RainForest - A framework for fast decision tree construction of large datasets." VLDB-98, 1998.
[22] J. Gehrke, V. Ganti, R. Ramakrishnan & W-Y. Loh,"BOAT – Optimistic decision tree construction." SIGMOD-99, 1999.
[23] J.C. Shafer, R. Agrawal, & M. Mehta. "SPRINT: A scalable parallel classifier for data mining." VLDB-96, 1996.

Authors

Tejaswini Apte received her M.C.A(CSE) from Banasthali Vidyapith Rajasthan. She is pursuing research in the area of Machine Learning from DAU, Indore,(M.P.),INDIA.. Presently she is working as an ASSISTANT PROFESSOR in Symbiosis Institute of Computer Studies and Research at Pune. She has 4 papers published in International/National Journals and Conferences. Her areas of interest include Databases, Data Warehouse and Query Optimization.
Mrs. Tejaswini Apte has a professional membership of ACM

Maya Ingle did her Ph.D in Computer Science from DAU, Indore (M.P.) , M.Tech (CS) from IIT, Kharagpur, INDIA, Post Graduate Diploma in Automatic Computation, University of Roorkee, INDIA, M.Sc. (Statistics) from DAU, Indore (M.P.) INDIA. She is presently working as PROFESSOR, School of Computer Science and Information Technology, DAU, Indore (M.P.) INDIA. She has over 100 research papers published in various International / National Journals and Conferences. Her areas of interest include Software Engineering, Statistical, Natural Language Processing, Usability Engineering, Agile computing, Natural Language Processing, Object Oriented Software Engineering. interest include Software Engineering, Statistical Natural Language Processing, Usability Engineering, Agile computing, Natural Language Processing, Object Oriented Software Engineering.
Dr. Maya Ingle has a lifetime membership of ISTE, CSI, IETE. She has received best teaching Award by All India Association of Information Technology in Nov 2007.

Dr. Arvind Goyal did his Ph.D. in Physiscs from Bhopal University,Bhopal. (M.P.), M.C.M from DAU, Indore, M.Sc. (Maths) from U.P. He is presently working as SENIOR PROGRAMMER, School of Computer Science and Information Technology, DAU, Indore (M.P.). He has over 10 research papers published in various International / National Journals and Conferences. His areas of interest include databases and data structure. Dr. Arvind Goyal has lifetime membership of All India Association of Education Technology