www.ijarcsse.com

Volume 4, Issue 9, September 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Performance Analysis of Database Indexing Techniques used in Large Biological Dataset

[1]Garima Munjal ,[2]Chandana
JCDV Sirsa,
Haryana, India

Abstract: Biomedical databases are different from traditional data warehouse as it contains non transactional information like bimolecular (protein, RNA, DNA, lipid, carbohydrate). A large amount of data is accumulated which is needed to be accessed in least amount of time when complex queries are executed in present biomedical data warehouses. When the Biological data is compare with the other proteins/structures data, searching takes the lot times for check the any similarities, relation between biological data. There is need of index for access of data in less time. In this paper we are going to Compare and analyze the different indexing techniques on Large Dataset. Different indexing techniques are clustered Index, Non-Clustered Index and Full Text Index.

Keywords: Indexing, clustered Index, Non-Clustered Index, Full Text Index

I. INTRODUCTION

The main operations on biomedical databases includes searching of protein and matching of certain patterns of data which is very complex and tedious task to handle with because finding a particular pattern of RNA/DNA in warehouse can even take days, so pattern matching or searching becomes difficult in the systems as it takes considerable amount of time and memory consumption depending upon the queries which are complex and iterative. Consider scenario of forensic science which deals with crime issues which are increasing day by day, a large amount of data is accumulated which is needed to be accessed in least amount of time when complex queries are executed in present biomedical data warehouses which is a major challenge. The ability to answer these complex queries efficiently depends upon a major factor 'Index'. Indexing of a data warehouse is complex and if there are few indexes, the data loads quickly but the query response is slows. If there are many indexes, the data loads slowly and there will be more storage requirements but the query response is good. This is true with large tables and complex queries that involve table joins. Considerable amount of time is taken by the query to be processed is more due to large size of both tables and attributes. Index's space and time play an important role in choosing an indexing technique in data warehouse. Usually if the space used by an index is large then the results are achieved in short time and on the other side, if the space used by the index space is small then the results are achieved in greater amount of time. So there is a tradeoff between the time consumed and the space used by a particular index. Important factors which are need to be improved:

1. Response time
2. Searching time/Scan time
3. Memory Usage

Different indexing techniques are clustered Index, Non-Clustered Index and Full Text Index.

Memory Consumption: Indexes which have been built on character based columns consumed much memory.
CPU Time: CPU time is the sum of Compilation time and Query execution time. There are number of indexes like clustered index, non clustered index and Full text index which having different execution time and corresponding, the different CPU time. Indexes have large CPU time depends on Index.

Working of Indexes : There are different types of indexes which can be implemented in different scenarios and before this, there is need to analyze the working, and memory consumption, etc. When the indexes are created, then the database engines stores and sorted the records in B-Tree Structure which reduces the memory consumption and disk reads during retrieve of data. The B-Tree structure is having the leaf nodes, root nodes and intermediate nodes. The bottom level of nodes is the leaf nodes, top level of nodes are root nodes and levels between the bottom and top nodes are intermediate levels. It contains the data pages and index pages. The index pages stores the records and index rows and different ID's for the specific indexes along with pointers.

II. LITERATURE REVIEW

In 2007, Han-Chieh Wei, Scott Dancer, Srinivas Kolluru,Erich Peterson assimilated the knowledge about the traditional index structures B+ tree and R-tree in accordance to applications which containing the hundreds of terabytes(TB) and difficult to maintain and retrieve subsets from the different databases, network scenarios and grid environment. According to author, the bitmap indexing is efficient multidimensional index for the ad hoc queries. There is pool of data which is generated day by day and staggered and due to customer-centric and e-science applications. The data is contained in storage systems and difficulty to collected, retrieve subsets and analyze, simulate data from storage pool. The ad-hoc queries are easy to apply on datasets and it can be multi dimensional queries. The data has been spread over the network and grid environment. The indexes such as B-Tree and R-Tree are optimized for the multi-dimension. It has been analyzed that the bit-map indexes are efficient multi-dimensional index structures for handling complex ad hoc queries in read mostly environment. This paper explains the concept of bitmap index and its mechanism which contains different types of indexes: Simple Bitmap Index, Binary Encoded Bitmap and Range Encoded Bitmap. The bitmap implementation consist the different steps

1. fileParser.parse( )
2. bitmapParser.parse( )
3. SimpleBitmap( ), EncodedBitmap( ), RangeBitmap( )
4. queryInterface
5. queryInterface.prompt( ):

There were number of methods proposed from which Word-aligned Hybrid (WAH) bitmap compression technique performed well. This WAH bitmap compression is based upon run-length encoding.

In 2008, Janya Sainui, Sirirut Vanichayobon and Niwan Wattanakitrungroj et al.,(2008) assimilated the knowledge about the advance indexing technique for improve the query processing time and apply data mining technique called frequent item set mining. This paper describes the Encoded Bitmap FIM (Frequent item sets mining) algorithm consists of number of steps and explained the advantages of algorithm by use Encoded Bitmap FIM algorithm. The concept of Apriori algorithm is used for frequent item datasets analyze .The performance of Optimizing Encoded Bitmap Index using frequent item sets mining is better than those found by existing techniques for a membership query from the point of view of space-time trade-off. Bitmap representations of indexing techniques are the best optimizing techniques. There is significantly improved query processing time by utilized multiple index scans and Boolean operations and, execute queries by perform the predicate conditions on the indexes level prior to the primary source. This paper explains: If there is need to optimize the existing Encoded Bitmap Index, and then apply techniques of data mining named FIM for finding an enhanced encoding scheme, and it leads to improving query processing and execution time. This paper explains the comparative study which shows that: the FIM techniques are consumed less time

In 2008, Stephane Azefack, Kamel Aouiche and Jerome Darmont, assimilated the knowledge about the automatic, dynamic index selection method for data warehouses that is based on incremental frequent item set mining from a given query workload. This paper describes the advantages of dynamic index selection method. The main advantage is the dynamic update of index method by identifies the workload. The issues are that the queries are complex and iterative which are executed against data warehouse and it contains the joins (Outer join, inner join, cross join or Cartesian product) and it results very costly in term of memory and processing time. This paper explains the advantages of indexing and so; the indexing is defined. This paper defines the frequent item set and provides efficiency. It helps to updates the indexes whenever workload evolves. This paper explained the Dynamic index selection strategy which contains number of steps. This paper has critical issue when using automatic, dynamic optimization strategies is to master system overhead, and in particular determine when the administrator should run the incremental index update process.

In 2010 Nur'Aini Binti Abdul Rashid, Rana Ghadban, Hazrina Yusof Hamdani,AtheerA-Abdulrazaqa assimilated the knowledge about the exhaustive searching technique in which explanation about the enhanced CAFE indexing structure/algorithm using the hash function and this reduces the space of the index structure and this speed up than the original CAFE Index. The improvement in the index is purposed with help of hash function for speed up the retrieve the records. The datasets are having of different records in which the the maximum number of comparisons is generated with help of graphical representation between the original and enhanced algorithm of CAFE technique.

III. PURPOSED WORK

There are different indexing techniques has been proposed and need to indentify the best indexing technique in terms of time and memory. The existing techniques can be good for timely searching the records of DNA/RNA structure, protein structure and it is observed that existing techniques takes much time. The proposed work will be implemented in the SQL Server 2008 by generate the dataset of variation records.

There are many database banks that provide the public database. There are some resources from National Center for Biotechnology Information (NCBI) which gives the data for proteins, Structure information and sequences information.

1. Collection of Protein Data.
2. Generate Dataset of different collection of records.
3. Apply different indexing technique clustered Index, Non-Clustered Index and Full Text Index.
4. Perform SQL Queries for Calculate the CPU Time and Memory Consumption.
5. Generate the Graphs and Analyze the Results.

Example: GenBank contains the datasets of proteins and structure information.

IV. RESULT AND DISCUSSIONS

Exact Match

In SQL, ‘WHERE’ key word can use for searching of exact keyword in tables of dataset. The Full text time consumption is increasing rapidly but on the contrary, Non Cluster index time consumption is much less nearly 400 ms approx. Cluster index takes less time for searching even the records in datasets are increasing. Cluster index gives better searching in exact matching of string and results good performance.

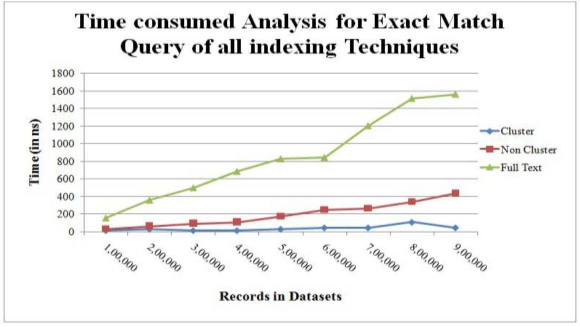

Figure 1: CPU time analysis for Exact Match Query

The non-cluster index and clustered index having the equal elapsed time approx but full text indexing elapsed time having the much elapsed time as the records are increasing from 1 Lakhs to 9 Lakhs.

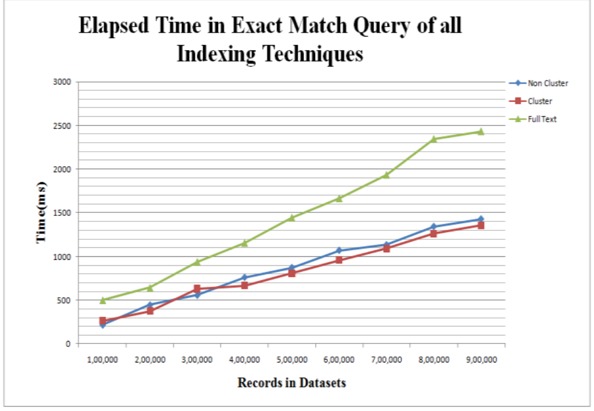

Figure 1: Elapsed Time of Exact Match Query Analysis

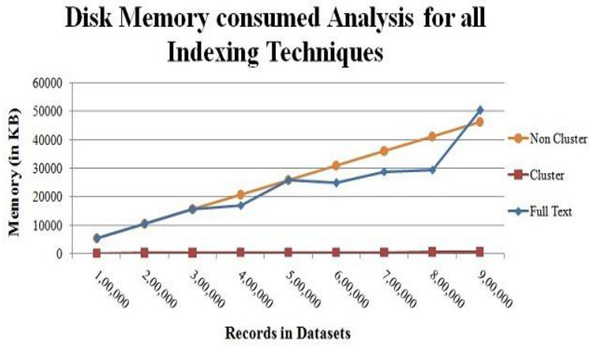

Figure 3: Indexes Disk Memory Consumption

REFERENCES

[1] Aho A. V., J. E. Hopcroft, and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley: An Imprint of Addison WesleyLongman, Inc., Reading Massachusetts, 1999.

[2] J. Kratika, I. Ljubic, and D. Tosic, "A genetic algorithm for the index selection problem", Applications of Evolutionary Computing (EvoWorkshops 03), Essex, UK; LNCS, Vol. 2611, Springer, Heidelberg, 2003, pp. 281-291.

[3] Khalid Jaber, Rosni Abdullah and Nur'Aini Abdul Rashid (2009) "Indexing Protein Sequence/Structure Databases Using Decision Tree: A Preliminary Study", Information Technology (ITSim), 2010 International Symposium, IEEE Computer Society Conference.

[4] M. Golfarelli, S. Rizzi, and E. Saltarelli, "Index selection for data warehousing", 4th International Workshop on Design and Management of Data Warehouses (DMDW 02), Toronto, Canada; CEUR Workshop Proceedings, Vol. 58, CEURWS. org, Aachen, 2002, pp. 33-42.

[5] P. O'Neil and D. Quass, "Improved Query Performance with Variant Indexes", SIGMOD, 1997

[6] R. Kimball, L. Reeves, M. Ross and W. Thornthwaite, "The Data Warehouse Lifecycle Toolkit : Expert Methods for Designing, Developing, and Deploying Data Warehouses", John Wiley & Sons, Aug. 1998

[7] Sankalap Arora, Priyanka Anand, Kirandeep Singh," An Efficient Indexing Technique Used In Telemedicine Data Warehouse", (2010)

[8] Safavian, S.R., Landgrebe (1991), "D.A survey of decision tree classifier methodology", Systems, Man and Cybernetics, IEEE Transactions, Page 660-674.

[9] Shawana Jamil and Rashda Ibrahim (2009) "Performance analysis of Indexing Techniques in Data Warehousing", Emerging Technologies, 2009. ICET 2009 International Conference, Page 57-61

[10] Sreerama K. Murthy (1998)," Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey", Association for Computing Machinery

[11] Thabasu Kannan, Dr.K.Iyakutti (2009) "A Clustered Indexing Method for Optimizing the Query for Biological Databases", GCC Conference & Exhibition, 2009 5th IEEE, page 1-6.