Visualization for IBQ Applications

V. Chandra Shekhar Rao
Research Scalar
Department of CSE,
JNTU, Hyderabad,
Telangana, India

P. Sammulal, PhD
Assistant Professor
Department of CSE
JNTU Hyderabad,
Tekangana, India

ABSTRACT

Iceberg query (IBQ) is a special class of aggregation query which compute aggregations upon user provided threshold (T). In data mining area, efficient evaluation of iceberg queries has been attracted by many researchers due to enormous production of data in industries and commercial sectors. Decision support database and discovery of knowledge related systems mainly compute aggregate values of interesting attributes by handling a big quantity of data in large databases. In literature, different strategies were found for IBQ evaluation, but using compressed bitmap index technique provides efficient strategy among all. In this paper, we propose a new strategy for computing IBQ, which builds a set for each attribute value, contains its occurrences in the attribute column and performs set operations for producing result. An experimentation on synthetic dataset demonstrates our approach is efficient than existing strategies for lower thresholds. We suggested set operations[11] in place of bitwise-AND operations to reduce execution time for different threshold values. And we developed effective GUI for aggregation of Different item pairs

Keywords

Database, iceberg query, threshold,, set operations

1. INTRODUCTION

Business insight and knowledge discovery from Operational databases/warehouses always powerful weapons for gaining competitive advantages in the present business world .An Iceberg Query(IBQ) is a special class of an aggregation query that computes aggregate values to a user specified threshold (T) value. It is of the special interest to the users in extracting high aggregate values that often carry more significant in formation. The Syntax of an Iceberg query on a relation REL (C1, C2…Cn) is shown below:

SELECT Ci, Cj, …, Cm, AGG(*)

FROM R GROUP BY Ti,

Tj…, Tm HAVING AGG (*) > = T

Here aggregation functions, Where Ci, Cj,…,Tm represents a subset of attributes in R and referred to as aggregate attributes. Aggregation functions such as COUNT(),COUNT(*),MIN,MAX,SUM and AVG. The greater than or equal to (>=) is a symbol used as a comparison predicate. So me of the applications of iceberg queries are : Market-basket analysis in data mining, Retrieval of Telecom information, Web searching engines, Data warehousing, Copy detection and Clustering, Decision support.

A Market Basket Analysis: Market analysts execute market basket queries on large data warehouses that store customer sales transactions. These queries identifies user buying patterns, by sending item pairs (and triples) that are brought together by many customers. Since these queries operate on very large datasets. We use the market basket query find commonly occurring word pairs.

B Advantages of Set Operations Are: Set operations help to evaluate the Iceberg Queries in decreased execution time when compared to remaining IBQ techniques such as tuple scan-based approach and dynamic pruning. It reduces the number of iterations between set pairs of two distinct attributes by performing set difference thereby pruning the sets which doesn’t satisfy the support value. It also helps in pruning the sets initially thereby reducing the iterations when the threshold condition doesn’t satisfy.

Graphical User Interface (GUI) is a point of interaction between the user and the computer software. The success and failure of a software application depends on the GUI. The Graphical User Interface is important in designing the educational software. If software is difficult to use, it forces users to do mistakes, or if it frustrates users efforts to accomplish his/her goals, he/she will dislike it, regardless of the computational power it exhibits or the functionality it offers; because it turns a user’s perception of the software, so the interface has to be right . When we design systems, we must consider the intended users, including profiles of their age, education, sex, physical abilities, cultural or ethnic background, motivation, goals and personality. Thus, a user interface design may not be useful for all computer users, while it may be just useful to specific users.

The remaining part is organized as follows: The collection of literature of different research papers that used in our project is mentioned in section 2. The proposed system is shown in section 3. The implemented GUI concepts are shown with neat diagram in section 4. We represent the GUI and results in section 5.We conclude our conclusion in section 6.

2. RELATED WORK

Some of the techniques that can be used for smaller database are: 1. Array of counters in memory - one for each distinct target, or 2. Sorting REL on disk then passes over it, aggregating and selecting the above threshold values. These techniques do not scale to large data sets. Hence, other techniques are needed. Some of them are:

2.1 Sampling

This procedure samples a small number of records fro m the relation, aggregates and extracts the records “candidates for the final solution” that relatively (the sample size) pass the threshold

2.2 Bucket counting

Rather than allocating a counter for each distinct value, allocate a counter for a group of distinct values, using a hash function to divide the values into groups. These building blocks produce false positive, values that are considered as candidates for the final solution but do not exceed the threshold. Sampling also causes false negatives, values that should be the final answer but are not found as candidates. These techniques follow tuple scan-based approach

2.3 Tuple scan based approach

Most existing query optimization techniques for processing iceberg queries can be categorized as the tuple scan base approach, which requires at least one table scan to read data from disk. They focus on reducing the number of passes when the data size is large. None has effectively leveraged the property of iceberg queries for efficient processing. Such a tuple-scan-based scheme often takes a long time to answer iceberg queries, especially when the table is very large. Besides these tuple-scan-based approaches, designed a two-level bit map index which can be leveraged fo r processing iceberg queries.

There are many different data structures used in data base to create indexes used to quickly evaluate queries. Each one has different strengths and weaknesses based on tradeoffs they make on memory, CPU and storage. One of these types of indexes is called a bit map index. Bit map indices are known to be efficient, especially for read mostly or append only data, and are commonly used in the data warehousing applications and column stores Using bitmap indices, we only need to access bitmap indices of the aggregate attributes. Second, bitmap indices operate on bits rather than real tuple values. Bitwise operations are very fast to execute and can often be accelerated by hardware. Third, bit map indices have the advantage of leveraging the anti monotone property of iceberg queries to enable aggressive index pruning strategies. Iceberg queries have an intriguing anti monotone property for many of the aggregation functions and predicates.

For GUI we followed Ben Shneiderman's eight golden principles .The principles are very heuristic and used in the design of interactive systems. These are strive for consistency, enable frequent users to use shortcuts, offer informative feedback, design yield closure, offer simple error handling, permit easy reversal of actions ,support internal locus of control, reduce short term memory load. When we are developing an user interface design, need to follow some important context in human computer interaction(HCI).These are, Nature of HCI :vision of HCI as a form of art, Social organization and work : Read the social structure developing computer games through online, Human processing of information : How to present information to users that are new to a field, Ergonomics : How posture affects efficiency and Designing a system fit for the disabled, Input and Output devices : New input and output devices make our lives easier, Design approaches: Success rates of products that use speech recognition and a waterfall model And effective use of color in interface design, Comparing the various system development processes.

3. PROPOSED SYSTEM

3.1 GUI

The aggregated attributes shown in the Iceberg query read from the database table-1 and generate equivalent bitmaps of them. Then index positions of 1's of bit maps Am and Bm are retrieved. The retrieved position values of Am and Bm are stored into a set called A Set and B Set respectively. With the use of count function the number of 1-bit positions are counted and stored in sets. Now each set which contains the number of bit positions is verified with the threshold. If the threshold is passed then the intersection operation is done between the pairs of satisfied A Set and B Set. If the result of intersection set has the count greater than the threshold, confirm this vector pair as an iceberg result and include them into iceberg result set. Then the Am and Bm Set positions are updated by conducting the difference operation for future reference. The updated bitmap vectors of Am and Bm to compare bit index positions with the threshold, if they are above the threshold the above process will be continued. Otherwise, we prune the bitmap Am and Bm Set if count not passes the threshold. Continue the same process until all the vector pairs are completed.

Table 1 bitmap index

X	X_1	X_1	X_3	X_2	X_3	X_3	X_2	X_3	X_2	X_1	X_1	X_2
Y	Y_1	Y_2	Y_1	Y_2	Y_3	Y_1	Y_2	Y_3	Y_1	Y_2	Y_3	Y_2

Step 1: Creating bit map indices of aggregated attributes mentioned in the query

X_1: 1 1 0 0 0 0 0 0 0 1 1 0

X_2: 0 0 0 1 0 0 1 0 1 0 0 1

X_3: 0 0 1 0 1 1 0 1 0 0 0 0

Y_1: 1 0 1 0 0 1 0 0 1 0 0 0

Y_2: 0 1 0 1 0 0 1 0 0 1 0 1

Y_3: 0 0 0 0 1 0 0 1 0 0 1 0

Step 2: Extraction of sets: Fetch an index position of 1 bit from each bitmap. Set X_1: {0,1,9,10} X_1.count=4,

Set X_2: {3,6,8,11} X_2.count=4,Set X_3: {2,4,5,7} A 3.count=4, Set Y_1: {0,2,5,8} Y_1.count=4,Set Y_2: {1,3,6,9,11}

Y_2.count=5,Set Y_3: {4,7, 10} Y_3.count=3

Step :3 Evaluation of IBQ using set operations along with optimization techniques and threshold value =2

Set intersection: (SX_1 ∩ SY_1): {0,1,9,10} ∩ {0,2,5,8} = {0}, so r.count = 1 As r.count doesn't satisfy threshold.

Hence, (X_1,Y_1) declared as not an Iceberg result. Set Difference: rd 1= (SX_1-r) = {0,1,9,10}-{0} = {1,9,10}.

SX_1.count=3 This is passing threshold. Hence, set X_1 is added back for further evaluation, otherwise pruned.

Similarly, rd2= (SY1-r) = {0,2,5,8}-{0} = {2,5,8}. SY_1.count=3 which is passing threshold. Hence, set Y1 is also added back for further evaluation. Resultant pairs are above (X_1,Y_2),(X_2,Y_2),(X_3,Y_3).

3.2 The following section discusses how we implemented

1. Creating database using two attributes: In this first module we will be creating the database with two attributes by randomly inserting the rows into the database.

2. Creating bitmap indices: In the second module based on the above database table, we will be generating the bit maps of 0's and 1's. By using these bits maps only we will be moving for further approaches. for i=1 to Table Size, if value of attr1 at row i is a then, attr2[i]=1 Else attr2[i]=0

3. Extraction of Sets: In this third module we will be extracting the sets based on the generation of bitmaps.

By using these sets we will be performing set operations like set intersection and set difference on these extracted sets in the further steps.

4. Evaluation if IBQ using set operations: In this fourth module, iceberg queries are performed on sets using the set Operations like set intersection and set difference.

The below algorithm shows the functionality of Evaluation, if IBQ using set Operations.

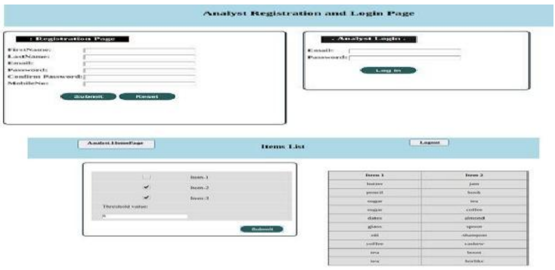
```
ALGORITHM IBQ( attributes A, attribute  B )

For each set A1 of attribute A, Store its position in set as set element in Sorted Set.

Push vectors of attribute A into Priority Queue based on its first 1st bit position if theirr size is greater than given threshold

    Ex. if A1.size >= T then

        SA.push(A1)

for each set $B_1$ of attribute B, store its position in set as set element in sorted set

Push vectors of attribute B into Priority Queue based on its first 1st bit position if their size is greater than given threshold

    Ex. if $B_1$. size >= T then

        SB.push( B1)

Let iceberg result R = null

Repeat following steps while both priority queues are notempty

Fetch aligned sets S1 and S2 from queues SA, SB

Let S3=S1

Compute S1=S1-S2

Compute S2=S2-S3

Compute c=size of S3-size of S1

If c>T add vectors with count c into result R.

Push sets into corresponding Priority Queues if

their set size is greater than given threshold.

Return result R
```

4. IMPLEMENTATION

4.1 This paragraph is a repeat of 3.1

Please use a 9-point Times Roman font, or other Roman font with serifs, as close as possible in appearance to Times Roman in which these guidelines have been set. The goal is to have a 9-point text, as you see here. Please use sans-serif or non-proportional fonts only for special purposes, such as distinguishing source code text. If Times Roman is not available, try the font named Computer Modern Roman. On a Macintosh, use the font named Times. Right margins should be justified, not ragged.

Implementation of GUI: We used the technologies CSS, HTM L, PHP, Javascript and JQuery for developing user interface . And MySQL database and Apache server. For PHP MySQL, and Apache server control panel is used which actually provides MySQL, Apache, and PHP. It is the home page that is look by the Analyst, Manager, and administrator. New analyst need to register through registration page and stores data into database.



Analyst Registration and Login Page

: Registration Page

FirstName:
LastName:
Email:
Password:
Confirm Password:
MobileNo:

Submit Reset

- Analyst Login -

Email:
Password:

Log in

Analyst HomePage Items List Logout

□ Item-1
☑ Item-2
☑ Item-3

Threshold value:

Submit

Item 1	Item 2
butter	jam
pencil	book
sugar	tea
sugar	coffee
dates	almond
glass	spoon
oil	shampoo
coffee	cashew
tea	bread
tea	biscuit

Fig 1: Registration Page

After registration he/she directly login through login page, Analyst homepage contains two fields . These are

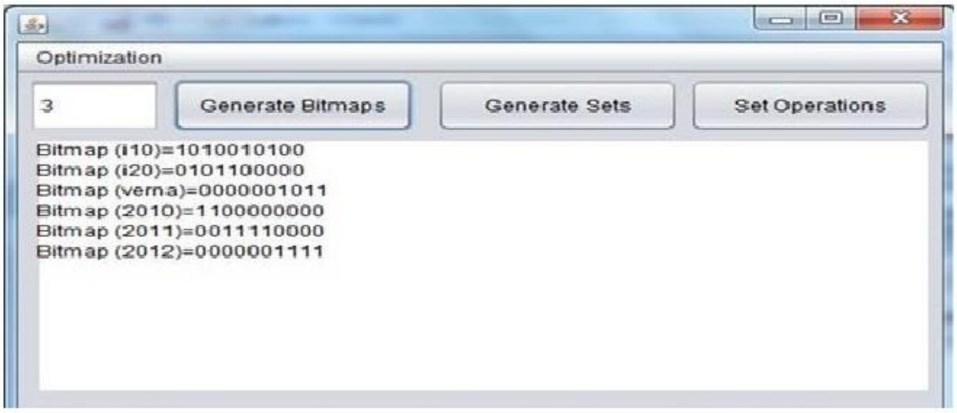
1. Item aggregation

2. Items list

Implementation of Items List page: This page contains number of items in the database . These items transactions are stored in database when customer doing shopping. Items database maintained by database administrator. This total items list storing in database and maintained by admin, using this database analyst will do aggregation on items list and display output as which pair of items satisfies threshold value. This item list represents number of items available in grocery store. Consumer will purchase items based on availability of items.

Implementation of Items Aggregation: This is the final resultant page. In this page, it display output as what pair of items satisfying the threshold value. Threshold value means some value like 1,2,3,4….etc based on size of database. If database size is high then threshold value like 100,200,300 etc. Items aggregation page display pair of items satisfying threshold value. Database contains so many attributes; in that attribute list every time took two attributes for performing aggregation. Pair of items displays as output which pair of items reached support value.

5. REESULTS



3

```
Bitmap (i10)=1010010100
Bitmap (i20)=0101100000
Bitmap (verna)=0000001011
Bitmap (2010)=1100000000
Bitmap (2011)=0011110000
Bitmap (2012)=0000001111
```

Fig 2: Generates bitmaps

Table 2: Performance on various thresholds

Threshold value	**Basic IBQ**	**Dynamic IBQ**	**Set IBQ**
100	23.2	21.0	21.56
200	12.7	10.8	13.8
300	8.28	7.4	8.52
400	5.77	4.9	6.3
500	4.94	4.25	4.98
600	4.23	3.95	4.52
700	3.59	3.53	3.6
800	3.1	2.8	3.2
900	2.47	2.23	2.22
1000	2.	2.21	2.41

6. CONCLUSION AND FUTURE SCOPE

This paper presents a new IBQ evaluation for processing of two columns using set representation method. The sets are used for processing of IBQ by conducting set intersection operation between aligned sets only. The experimental results are demonstrated and observed that IBQ evaluation time from first two attributes. The future research direction of this work may be multiple attributes with reduction of number of set operations and applying dynamic approach in choosing the attributes which may further optimizes the execution time of evaluation of iceberg queries.

7. ACKNOWLEDGMENTS

I acknowledge my guide, my colleagues and .my friends who helped my work.

8. REFERENCES

[1] Gossen, Tatiana, Marcus Nitsche, and Andreas Nurnberger. "Knowledge journey: A web search interface for young users." Proceedings of the Symposium on Hu man-Co mputer Interaction and Information Retrieval. A CM, 2012.

[2] R. Agrawal, T. Imielinski, and A.N. Swami, “Mining Association Rules between Sets of Items in Large Databases,” Proc. A CM SIGM OD Int’l Conf. Management of Data, pp. 207-216, 1993.

[3] Shneiderman. B and Plaisant. C, “The Eight Golden Rules of Interface Design”, Designing the User Interface: Strategies for Effective Hu man Co mputer Interaction: Fifth Edit ion, Addison- Wesley Publ. Co., MA, Ch.2,Sec 2.3.4, 2010

[4] http://wiki.up.ac.za/ index.php/HCI_research_topics

[5] Bin He, Hu i-I Hsiao, Ziyang Liu, Yu Huang and Yi Chen, “Efficient Iceberg Query Evaluation Using Co

mpressed Bit map Index”, IEEE Transactions On Knowledge and Data Engineering, vol 24, issue 9, sept 2011, pp.1570-1589.

[6] Dayal, Nab ilKamel, GunterSchlageter,andKyu-Young Whang, editors,VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10-14, 2000, Cairo, Egypt, pages 329–338. Morgan Kaufmann, 2000.

[7] www.w3schools.com/php/default.asp

[8] Scott W. A mbler "User Interface Design Tips, Techniques, and Princip les."

[9] R.C.C. Guntupalli, "User Interface Design-Methods and Qualities of Good User Interface Design", M.Thesis, University West, Sweden, Jun. 2008.

[10] Alan Dix, Janet Finalay, Gregory D. Abowd, Russell Beale "The Hu man, The Co mputer, The Interaction", Hu man Co mputer Interaction, 3rd edition,England,Pearson Publishers, Ch.1-3, pp.11-123, 2004

[11] Rao V.C.S, Sammulal P, “Efficient Iceberg Query Evaluation Using Set Representation”, IEEE Explorer, INDICON, Pune, INDIA, ISBN 978-1-4799-5362-2, December 2014, pp.1-5