*International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*
*Volume 4 Issue 6, June 2015*

# Mining for Data Cube and Computing Interesting Measures

**Miss.Madhuri S. Magar**
**Student, Department of Computer Engg .**
**BSIOTR, Savitribai Phule Pune University**

**Prof. Sonali A. Patil**
**Professor, Department of Computer Engg.**
**BSIOTR, Savitribai Phule Pune University**

*Abstract*— **Efficient computation of aggregations plays important part in Data Warehouse systems. Data Cube is introduced which is a way of structuring data in N-dimensions so as to perform analysis over some measure of interest. Data cube computation is a key task in data warehouse. For many important analyses done in the real world, it is critical to compute interesting measures for data cubes and subsequent mining of interesting cube groups over massive data sets. For analyzing the multidimensional data cube analysis is one of the important tool. There are several methods and techniques for cube computation but having some limitations. To take advantage of parallel computing, MapReduce framework based cube computation is proposed. To work on holistic measure for cube analysis the subset of holistic measure is identified and used for MapReduce based framework. The MapReduce based approach is used for efficient cube computation and mining for identifying interesting cube group for given subset of holistic measure. The problem of extreme data skew arise while implementing MapReduce based cube computation i.e. MR-Cube. The extreme data skew is detected and handle using Log-Frequency Sketch which is compressed counting data structure.**

*Index Terms*— ***Data Cube; MapReduce; Cube Mining; Log Frequency Sketch; Holistic Measures***.

## I. Introduction

Data cube is very important concept and research area in OLAP (Online Analytical Processing). The researches on the data cube consider the aspect related to cube compression and cube storage and also, consider how to choose the cube and materialize them. Massive data size and high dimensionality of big data introduce computational challenge. Storing all the data cubes needs a lot of resources and space. In order to help analyst to get effective data, need to find the best method for choosing cube and materialize data cubes and perform mining to finding out interesting information for analyst.

The Data cube is the N-dimensional generalization of simple aggregate functions [3]. In OLAP systems, a data cube is a way of organizing data in N-dimensions so as to perform analysis over some measure of interest. Measure is a term used for numerical facts that can be non-algebraic (DISTINCT, TOP-K etc.) or algebraic (SUM, COUNT etc.).Supporting multiple aggregates in OLAP databases Data cube is used. It requires computing group-bys on all possible combinations of a list of attributes, and is equivalent to the union of a number of standard group-by operations. The cube problem is to compute all of the aggregates as efficiently as possible. As many techniques are proposed for efficient cube computation. Data cube analysis is a powerful tool for analyzing multidimensional data stored in a data warehouse that maintains huge information. A lot's of studies have been devoted to designing techniques for efficiently computing the cube. A variety of algorithms on Cube computations and Storage are suggested by various Researchers. There are several methods for cube computation, several strategies to cube materialization and some specific computation algorithms, namely Star Cubing, Multiway array aggregation, Bottom Up Cubing, the computation of shell fragments and parallel algorithms, But these techniques have limitation so MapReduce based approach is used in proposed system.

Users consider the data as multidimensional data cubes. Data cube construction is important operation in data warehouses. Each cell of the data cube is nothing but a view consisting of an aggregation of interest. The values of many of these cells are dependent's on the values of other cells in the data cube. Commercial systems having different approaches to materializing the data cube .Cube analysis is good way to discover insights from the data by computing different aggregate measures. Data analysis applications typically aggregate data across many dimensions and they are looking for strange patterns. It is the process for extracting useful patterns from the large database.

In this age of data explosion, parallel processing is very important to processing a large volume of data, for that MapReduce based approach is used. MapReduce based Approach used for data cube materialization and mining over massive data sets .Important subset of holistic measure is identified and measure is called as Partially Algebraic Measure. These measures are easy to compute in parallel as compare to Algebraic. The partitioning mechanism added to balance the data load and for efficiently distribute data. To effectively distribute computations and balancing intermediate data produced, Batch Areas are formed. Finally three phase cube computation algorithm called **MR-Cube** which successfully used for cube materialization and mining for interesting cube groups. The Problem of extreme data skew is handled using compressed counting structures as **Log-Frequency Sketch**. Analyst of system analyses over real data available from query logs. MR-Cube gives more efficiency and scalability. This paper is organized as follows: the next section reviews the related work on cube computation, materialization and mining for interesting cube groups. Section III describes the implementation details of proposed system architecture, Section IV details about possible results obtained by analyst of proposed system. In section V describe and concludes this study

## II. RELATED WORK

Data mining is the very important branch of knowledge discovery in database. Data mining is the application of efficient algorithms to detect the desired patterns contained within the given data. In data warehousing, Data cube computation, materialization and mining are most essential but more costly procedures. As many study was devoted for data cube, and there are many techniques have been proposed for efficient cube computation. There are several methods for cube computation, several strategies to cube materialization and some specific computation algorithms, namely Multiway array aggregation, BUC, , top-down versus bottom-up cubing, Star Cubing, the computation of shell fragments and parallel algorithms[13] But, these techniques have limitation as:

1. They are designed for a small data size with sequential processing. Now a days, at many companies the data accumulation rate is very high (e.g., terabytes per day) and increases exponentially. It is difficult to process such data with sequential manner so need to take advantage of parallel processing.
2. Many of the established techniques take the measure of type algebraic and use this property to avoid processing groups with a large number of tuples. This allows parallelized aggregation of data subsets and its results are then post processed to obtain the final result. Computation of holistic (i.e., non-algebraic) measures is required for analyses over logs such as top-k most frequent queries or the distinct number of users.

To address these limitations, MapReduce based approach is used [1]. Our proposed system works on cube materialization and mining for holistic measures. It uses the MapReduce Paradigm. MapReduce based Approach used for data cube materialization and mining over big data sets using important subset of holistic measure called Partially Algebraic Measure. The work of the proposed system is related to efficient cube computation. Extreme data skew is one of the important problems that need to be handle. This is handled using compressed counting structures as LFS. The input data used is the query logs. The analysis of the result obtained will be done by computing aggregates with holistic measures.

## III. IMPLEMENTATION DETAILS

The proposed system works on cube materialization and mining task on web-scale data set. Data dimensions and measures are given as input. The objective is to find out important subset of holistic measure i.e. Partially Algebraic Measure that are easy to compute in parallel as compare to holistic measure and implement Map Reduce based approach for Cube computation and mining. The system also identifies the interesting cube group for given holistic measure. The Extreme data skew is handled using the compressed counting data structure as Log-frequency Sketch.

### *A. System Architecture*

Figure shows the proposed system architecture. The Raw Dataset is taken as input to our system. There are the following modules:

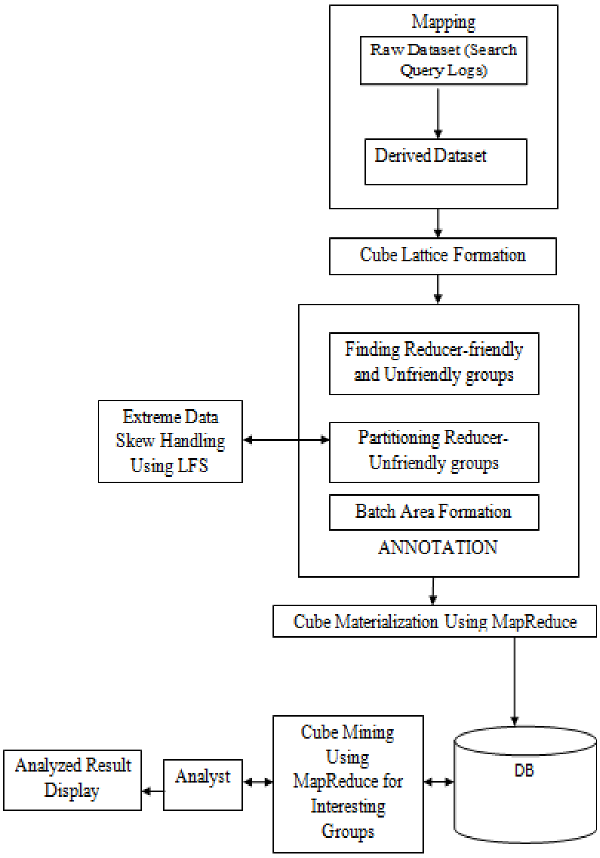


Figure 1. Proposed System Architecture

*1. Preparation of Derived Data set:*

Pre-process the existing raw data. Convert given raw data into some readable and useful format. According to that determine measure and dimensions for construction of data cube. Raw data are stored as a set of tuples. Each tuple consists of a set of raw attributes, such as ip and query. For many analyses, it is more desirable to provide mapping for some raw attributes into a fixed number of derived attributes through a mapping function provided by user. For our example, ip can be mapped to city, state, and country. Similarly, query can be mapped to subcategory, category, and topic

Ip→city, state, country
Query→topic, category, subcategory

  

*2. Lattice Formation:*

Determine the dimensional attribute that user wants to analyze and based on those attributes, a cube lattice can be formed representing all possible grouping(s) of the attributes. A cube lattice is generated, where the dimension attributes include the six attributes derived from ip and query. A node in the lattice represent cube region and a cube groups are the actual groups belonging to cube region.

The following modules belong to the use of Map Reduce based approach for large scale cube computation using user specified holistic measure. The goal is to divide the computations into the pieces such that no reducer can handle large data group and intermediate data size is controlled.

*3. Determine Partially Algebraic Measure:*

First find a subset of holistic measure called as Partially Algebraic Measure. These measures are easy to compute in parallel as compare to Algebraic. It is either provided by analyst or detected by system from few frequently used measures. In proposed system it is provide by the analyst. To detect whether holistic measure M is partially algebraic or not, Detection Principle is used.

Detection Principle:

If there exists an aggregation A based on attribute a and an algebraic measure M', such that M (D) = M' (A (D)), where D is the original data, then M is partially algebraic on attribute a. For the measure reach, the aggregation is "group by uid" and the algebraic measure is count.
Example- Reach (unique number of users) and TOP-K are partially algebraic above data set.

*4. Finding Reducer Unfriendly Groups:*

The large cube groups are required to partition base on the algebraic attribute. In order to determine which groups to be partitioned the proposed system distribute cube group into two categories as Reducer-Friendly and Reducer-Unfriendly. The Sampling Approach is used to differentiate it.

Sampling Approach:

The group G to be Reducer-Unfriendly if it is observe more than 0.75rN tuples of G in the sample data, where N is the sample size and r =c/|D| denotes the maximum number of tuples a single reducer can handle (c) as percentage of the overall data size(|D|).
Using this approach Reducer-Unfriendliness of each cube region is estimated. The region is Reducer-Unfriendly if it contains at least one Reducer-Unfriendly group. In proposed system sampling is done by performing cube computation on sample random dataset of derived data with count as measure.

*5. Partitioning Reducer Unfriendly Groups And Data Skew Handling:*

The Partitioning is performed only on the Reducer-Unfriendly groups. The region is annotated with appropriate partition factor. It is integer closest to s/rN. Each Reducer-Unfriendly region is partitioned using the partition factor.
Data Skew Handling using LFS:

If a few cube groups are unusually large even when they belong to a cube region at the top of the lattice .This causes value partitioning to be applied to the entire cube it is called as problem of extreme data skew, which reduces the efficiency of proposed MR-Cube algorithm.

Value partitioning on a region-by-region basis: if a cube region is estimated to contain a reducer-unfriendly group, all groups within the region are value partitioned, many of which may not be necessary. This approach works well until there is extreme data skew which can lead to most cube regions being value partitioned. Alternative approach of marking reducer unfriendly groups instead of regions is proposed since the number of groups can be very large; it may not be feasible to compute quickly or maintain some statistics in the mapper's memory, as can be easily done for regions. So to overcome this problem compressed counting data structure such as Log-Frequency Sketch is used as solution to this.

Log-Frequency Sketch:

"Frequency based sketches" are concerned with summarizing the observed frequency distribution of a dataset. From these sketches, accurate estimations of individual frequencies can be extracted. This leads to algorithms to find the approximate heavy hitters (items which account for a large fraction of the frequency mass) and quintiles' (the median and its generalizations). The same sketches are also used to estimate (equi) join sizes between relations, self-join sizes and range queries. A different style of sketch construction leads to sketches for distinct-value queries such as COUNT DISTINCT query.

Practical data analysis relies on the ability to count observations of objects *succinctly* and *efficiently*. Unfortunately the space usage of an exact estimator grows with the size of the *a priori* set from which objects are drawn while the time required maintaining such an estimator grows with the size of the data set. We present static and on-line approximation schemes that avoid these limitations when approximate frequency estimates are acceptable. Our *Log-Frequency Sketch* [14] extends the *approximate counting* algorithm of Morris [14] to estimate frequencies with bounded relative error via a single pass over a data set. It uses *constant space per object* when the frequencies follow a power law and can be maintained in *constant time per observation*.

The Log-Frequency Sketch estimates the frequencies of objects $x \in U$ via a single pass over a data set $D$ with the following guarantees.
(i) (*Bounded relative error*) Estimates are within a constant factor of their true frequencies with high probability.
(ii) (*Constant space*) Estimates are maintained using constant space per object independent of $|U|$.
(iii) (*Constant update complexity*) Estimates are maintained in constant time per observation independent of $|D|$.

*6. Batch Area Formation:*

Partitioning technique is used for effectively distribute computation. It is also called as Batch Area. To balance the intermediate data and pruning unnecessary data Batch Areas are used here. Each batch area represents a collection of regions which share a common ancestor region. We suggest combining regions into batch areas.

  

Constraints for formation of Batch Area:
*(i)* A region with at least one parent that is also reducer friendly must belong to a batch area that contains at least one of its parents.
*(ii)* No two regions whose parents are reducer-unfriendly can belong to the same batch area.
*(iii)* The difference in the number of regions of two batch areas cannot be more than two, a heuristic used to balance the workload of each batch area.

The process of identifying reducer unfriendly groups, partitioning these groups and formation of Batch Area is called as ANNOTATE. The lattice formed is called annotate lattice. The following figure shows the annotate lattice for our data set where Each Reducer-Unfriendly region is value partitioned using a partitioning factor estimated from sampling approach.

*7. Cube materialization Using Map-Reduced Approach:*

Materializing the cube means computing measures for all cube groups satisfying the pruning conditions. The generated annotated lattice take as input to the cube materialization using map reduce based approach. In map reduced based approach, mappers are allocated to each batch area and it emits key: value pairs for each batch area. In required, keys are appended with a hash based on value partitioning, then in shuffle phase sorts them by key. The BUC Algorithm is run on each reducer, and the cube aggregates are generated. All value partitioned groups need to be aggregated to compute the final measures. Then cube is loaded into DB for future exploration.

*8. Mining for Interesting Cube Group:*

"A group Gi is said to be interesting if the measure for that group is higher than any of its sibling groups Gj with respect to a dimension d". The proposed system cube mining is done as separate MapReduce which is post materialization step.

As user is interested in particular cube groups in dataset, searching that interesting pattern is necessary. For that purpose cube materialization need to be completed. Finding interested patterns in a huge data will be hectic, but computation of cube saves efforts. Algorithm 5 explains how mining will be done so that user can get interesting patterns. By using the proposed system, it is feasible to perform both large-scale cube materialization and mining in the same framework

*B. Algorithms*

*Algorithm - Overall MR-Cube Computation (MR-Cube) Algorithm*

MR-Cube (Cube Lattice C, Data set D, Measure M)
1 $D_{sample}$=SAMPLE(D)
2 RegionSize R=ESTIMATE-MAPREDUCE ($D_{sample}$, C)
3 $C_a$=ANNOTATE(R, C) # *partitioning . & batching*
4 **while** (D)
5 **do** R <− R ∪ MR-CUBE-MAPREDUCE
6 D <− D′ # *retry failed groups D′ from MR-Cube-Reduce*
7 $C_a$ <− INCREASE-PARTITIONING(Ca)
8 Result <− MERGE(R)# *post-aggregate value partitions*
*9 return **Result***

The overall MR Cube algorithm breaks into following phases:
1. Annotation MapReduce
2. Materialization MapReduce
3. Aggregation MapReduce
4. Mining MapReduce

All the above steps along with its algorithm explained in the below section:

*Algorithm .MR-Cube Phase 1: Annotation MapReduce*

ESTIMATE-MAP(e)
1 *# e is a tuple in the data*
2 let C be the Cube Lattice;
3 **for each** $c_i$ **in** C
4 do EMITE($c_i$,$c_i$(e)⇒ 1) *# the group is the secondary key*
ESTIMATE-REDUCE/COMBINE(<r,g>, {e1,e2,...})
1 *#<r, g> are the primary/secondary keys*
*2* MaxSize S <− {}
3 **for each** r,g
4 **do** S[r] <− MAX(S[r],|g|)
5 *#|g| is the number of tuples{$e_i$,....$e_j$}∈ g*
6 **return** S

*Algorithm. MR-Cube Phase2: Materialization MapReduce*

MR-CUBE-MAP(e)
1 *# e is tupele in the data*
*2 let* $C_a$ be the Annotaed Cube Lattice
3 **for each** $b_i$ **in** $C_a$.batch_areas
4 **do** s<− $b_i$[0].partition_factor
5 EMIT (e.SLICE($b_i$[0])+e.id%s⇒ e)
6 *# paritioning:'e.id present s' is appended to primary key*
MR-CUBE-REDUCE(k,V)
1 let $C_a$ be the Annotated Cube Lattice
2 let M be the measure function
3 cube <− BUC(DIMENSIONS($C_a$,k), V, M)
4 EMIT-ALL(k, cube)
5 **if** (MEMORY-EXCEPTION)
6 **then** D′ <− D′ ∪ (k,V)

*Algorithm .MR-Cube Phase 3: Aggregation MapReduce*

AGGREGATION-MAP(g, p, m)
1 *# g, p, m are group label, partition id & measure value*
2 EMIT(g⇒ m)
AGGREGATION-COMBINE/REDUCE(g, M)
1 *# key: group label g & measures M*
2 EMIT(g, $\sum M_i$)

*Algorithm **.** MR-Cube Phase 3: Mining MapReduce*

MINING-MAP(g, p, m)
1 *# g, p, m are group label, partition id & measure value*
*2* **for each** parent $p_i$ , dimension $d_i$ , **in** PARENT(g)
3 **do** EMIT(($p_i$ ,$d_i$ ),g ⇒m)
*4 # g is secondary key*
MINING-COMBINE(((p, d), V<g, m>))
1 **for each** group $g_i$ **in** V
2 do EMIT((p, d), $g_i$⇒ $\sum m_j : m_j \in g_i$ )
MINING-REDUCE((p, d), V<g, m>)
1 *# key: parent p & dimension d*
*2 # values: measures m ordered by $2^{dary}$ key, group g*
*3* $g_{best}$ = **null**
4 $m_{best}$ = **null**

  

**for each** group $g_i$ , measure $m_{j \in} g_i$ , **in** V
**do if** $(\Sigma m_j > m_{best})$
**then** $g_{best} = g_i$
$m_{best} = \sum m_j$
EMIT(p, d, $g_{best}$, $m_{best}$ )

*C. Mathematical Model*

Let S be the system or application.

A. $S = \{I, O, \delta, A, q_0, F,$

*Where,*

I = set of inputs,
O = set of output,
$\delta$ = transition,
A = set of algorithms,
$q_0$ = initial set,
F = final set,
$P_M$ = performance measure
I = { Real Dataset, Partially Algebraic Measure
O = {Analysis Result, Performance Statistics}
$\delta$ = {v1,v2,v3,v4,v5,v6,v7,v8 ,v9}
{Cube lattice, Sampling approach, Reducer-Friendly group, Reducer-Unfriendly groups, Partitioning, Data Skew Handling, Batch Area Formation, Cube materialization, Cube Mining}
A = {MR-cube computation , sampling Approach, LFS}
$q_0$ = {Data Preprocessing}
F = {Comparative Result}
$P_M$ = {Time, Graph Analysis}

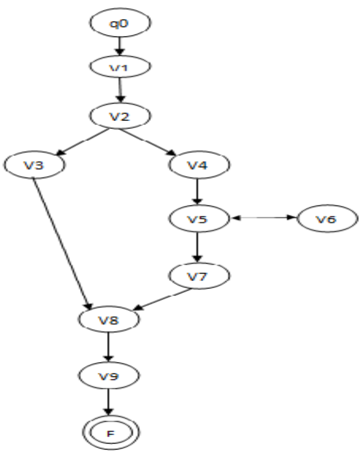


Figure 2. Mathematical Model

*D. Experimental Setup*

The standard configuration required to built the system is
(a) Ubuntu 12.04
(a) MYSQL 5.0
(b) Java (JDK 1.6)/ Eclipse IDE
(c)Hadoop Framework (version 1.0.2)

## IV. RESULTS

*Data Set*

We have used raw dataset of Web Search query logs in the form "id, time, userid, ip, query". For detail analysis, it is necessary to locate raw attributes into fixed number of derived attributes. We used tables for mapping. For example, query can be mapped to topic, category, subcategory and ip can be mapped into country, state, city. Three dimensions are established with six level. Data set is consist of id, date, uid, country, state, city, topic, category, subcategory. The full data set contains thousands of click tuples.

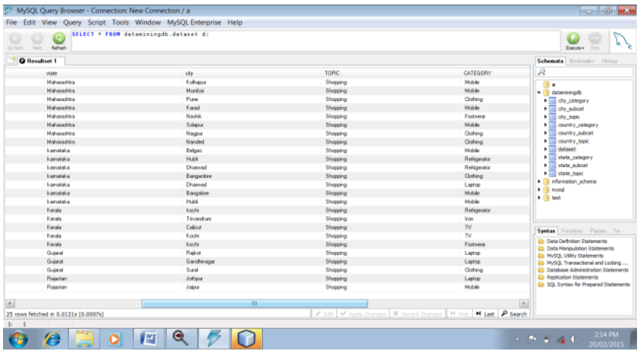

Figure 3.Dataset

*Cube computation Tasks*

1. Computing user reach- This measure computes the number of distinct users within the set of tuples for each cube group. It is partially algebraic on uid.
2. Computing Top-K queries-for our work we compute the Top-5 most popular queries. It is partially algebraic on query.

*Final Results*

The proposed system mainly works on cube materialization and mining with extreme data skew handling using the Log-Frequency Sketch as solution for it. User reach and Top-K used as Measures. The comparative results are obtained for these two measures with data skew handling using LFS and without data skew handling. The time is used as performance measure. The time required for cube computation and mining using MR-Cube and LFS is minimum as compare to without using LFS. The approximate final results for measure Reach given by the following graph.

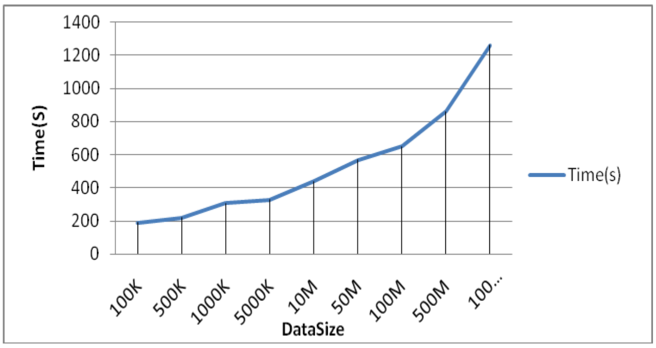


Figure 4. Performance of system without skew handling For measure Reach

  

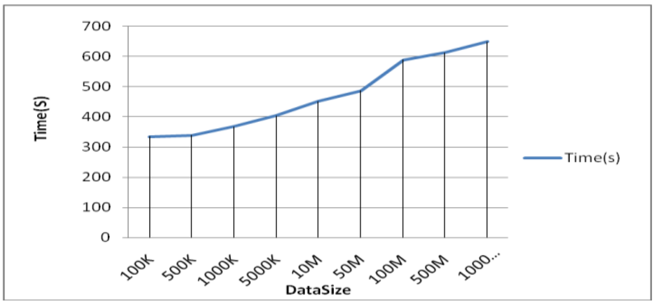


Figure 5. Performance of system with skew handling For measure Reach

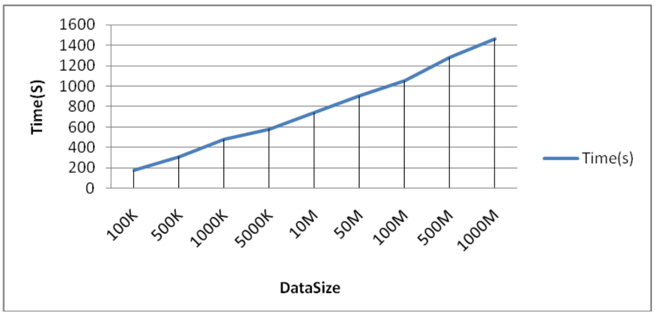


Figure 6. Performance of system without skew handling for measure Top-K

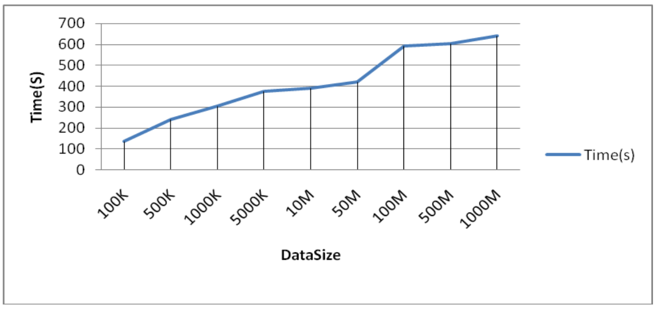


Figure 7. Performance of system with skew handling for measure Top-K

## V. CONCLUSION AND FUTURE SCOPE

In this paper we presented a workflow for cube computation with holistic measures and mining of interesting cube groups. cube materialization and subsequent mining of holistic measures over extremely large data such as search logs using the MapReduce based approach is done. A system is designed to identify a subset of holistic measures that are partially algebraic and propose sampling approach for value partitioning to make them easy to compute in parallel. The algorithms(MR-Cube) are implemented to partition the cube lattice into batch areas for effectively exploit both the parallel processing power of MapReduce and the pruning power of cube materialization algorithms.MR-Cube algorithm efficiently distributes the computation in parallel and is able to complete cubing tasks at a scale where previous algorithms fail. System uses the reach and Top-K as measure. System handle the problem of extreme data skew using LFS as solution.

Comparative results of MR-Cube with data skew handling using LFS and without Data skew handling for both Reach and Top-K are obtain. System with skew handling gives more accuracy than system with without skew handling.
In this work it is assume that the algebraic attribute are provided by analyst in our case it is Reach and Top-K. Future work can be done as system automatically deciding whether given holistic measure is partially algebraic or not and detecting its algebraic measure. Use of some other than LFS sketch techniques such as Count Minimum Sketch (CM-Sketch) [2] for data skew handling is also part of future work.

## ACKNOWLEDGMENT

We would like to thank the researchers as well as publishers for making their resources available and the teachers for their guidance. We are thankful to authorities of Savitribai Phule Pune University for their constant guidelines and support. We also thank the college authorities of BSIOTR, Wagholi,Pune for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to all friends and family members.

## REFERENCES

[1] Arnab Nandi, Cong Yu, Philip Bohannon, and Raghu Ramakrishnan, Fellow, IEEE, "DataCube Materialization and Mining over MapReduce" IEEE transactions on knowledge and data engineering, vol. 24, no. 10, October 2012

[2]Amrutha A, Viji Mohan A"large-scale data mining and materialization using CM-Sketch and MR Cube algorithm" Proceedings of 2nd IRAJ International Conference, 13th October 2013, Trivandrum, India. ISBN: 978-93-82702-33-7

[3] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh, "Data Cube: A Relational Operator Generalizing Group- By, Cross-Tab and Sub-Totals," Proc. 12th Int'l Conf. Data Eng. (ICDE), 1996.

[4] S. Agarwal, R. Agrawal, P. Deshpande, A. Gupta, J. Naughton, R. Ramakrishnan, and S. Sarawagi, "On the Computation of Multidimensional Aggregates," Proc. 22nd Int'l Conf. Very Large Data Bases (VLDB), 1996.

[5] K. Ross and D. Srivastava, "Fast Computation of Sparse Datacubes," Proc. 23rd Int'l Conf. Very Large Data Bases (VLDB), 1997.

[6] K. Beyer and R. Ramakrishnan, "Bottom-Up Computation of Sparse and Iceberg CUBEs," Proc. ACM SIGMOD Int'l Conf. Management of Data, 1999.

[7] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, and J.D. Ullman, "ComputingIceberg Queries Efficiently," Proc. 24th Int'l Conf. Very Large Data Bases (VLDB), 1998.

[8] J. Hah, J. Pei, G. Dong, and K. Wang, "Efficient Computation of Iceberg Cubes with Complex Measures," Proc. ACM SIGMOD Int'l Conf. Management of Data,2001.

[9] R.T. Ng, A.S.Wagner, and Y. Yin, "Iceberg-Cube Computation with PC Clusters,"Proc. ACM SIGMOD Int'l Conf. Management of Data, 2001.

[10] D. Xin et al., "Star-Cubing: Computing Iceberg Cubes by Top- Down And Bottom-Up Integration," Proc. 29th Int'l Conf. Very Large Data Bases (VLDB), 2003.

[11] R. Jin, K. Vaidyanathan, G. Yang, and G. Agrawal, "Communication Memory Optimal Parallel Datacube Construction," IEEE Trans. Parallel and Distributed Systems, vol. 16, no. 12, pp. 1105- 1119, Dec. 2005.

[12] Y. Chen, F.K.H.A. Dehne, T. Eavis, and A. Rau-Chaplin, "PnP: Sequential, External Memory and Parallel Iceberg Cube Computation," J. Distributed and Parallel Databases, vol. 23, pp. 99-126, 2008.

[13] K. Sergey and K. Yury, "Applying Map-Reduce Paradigm for Parallel Closed Cube Computation," Proc. First Int'l Conf. Advances in Databases, Knowledge, and Data Applications (DBKDA), 2009

[14] D. Talbot, "Succinct Approximate Counting of Skewed Data," Proc. 21st Int'l Joint Conf. Artificial Intelligence (IJCAI), 2009.

## Author Profile

**Madhuri S. Magar** received the B.E degree in Computer Science & Engineering Information Technology from Bharati Vidyapeeth's Collage of Engineering, Kolhapur in 2013.She is currently pursuing her master's degree in computer engineering from Bhivrabai Sawant Institute of Technology and Research, Wagholi, Pune

  