POST STRATIFICATION SAMPLING AND HORVITZ THOMPSON ESTIMATOR FOR RANGE AGGREGATE QUERIES IN BIG DATA ENVIRONMENTS

S. Barkath Nisha[1] | R. Latha Priyadharshini[2]
[1](Department of CSE, P. A. College of Engineering and Technology, Coimbatore, Tamil Nadu, barkathnisha10cs@gmail.com)
[2](Department of CSE, P. A. College of Engineering and Technology, Coimbatore, Tamil Nadu, priya86755@gmail.com)

Abstract— *Big Data is a collection of large datasets and handling of data is challenging in this environment. Fast Range Aggregate Queries (FastRAQ) approach is used to process the range aggregate queries that consist of aggregate function on all tuples within the query ranges. The query result can be generated from the range cardinality query algorithm. The weight of the sample estimate is calculated using the Post Stratification sampling method and to estimate the total and mean of a super population in a stratified sample, Horvitz Thompson estimator is used. The time complexity is reduced by using the sampling methods.*

Keywords— Balanced partition; Big Data; FastRAQ; Hadoop; Horvitz Thompson; MapReduce; Multidimensional Histogram; Post Stratification; Range Aggregate Query.

1. INTRODUCTION

In Big Data environments, traditional computing techniques are not suitable for processing. The volume of the data refers to the size of the data and the variety determines the type of the content. The velocity determines the speed of the data [3].

Hadoop is an Apache open source framework written in java and is classified in to computation layer and Storage layer. MapReduce parallel programming model is used to write distributed applications for efficient processing of large amount of data. The Hadoop Distributed File System is designed to run on commodity hardware.

Data is initially divided into directories and files. Files are divided into uniform sized blocks and are distributed across various cluster nodes for processing. The sort operation takes place between mapreduce stages and it send the sorted data to a computer and the debugging logs for each job is written.

A. FastRAQ Approach

Range aggregate queries refer to aggregate function on all tuples within the query ranges. The FastRAQ approach is used for processing range aggregate queries and provides accurate results in big data environments. Balanced partitioning algorithm is used to generate a local estimation sketch for each partition [1] using FastRAQ approach. The local estimates from all partitions are summarized to obtain the result.

Range query retrieval in the smaller database is not efficient and the big volume of records consumes more time. The problem is addressed in existing methodology for retrieving accurate and time concerned output for range queries from larger databases [15]. It is achieved by FastRAQ mechanism that attempts to aggregate the query results of number of partition using stratified sampling.

The Prefix-sum Cube method is used in Online Analytical Processing [12], [13] to boost the performance of range aggregate queries. The range aggregate query on a data cube provides the result with less time complexity. A new tuple is added into the cube, the prefix sums is to be recalculated. Online Aggregation (OLA) is an important answering approach to speeding range aggregate queries. The early estimated returns are provided by the OLA [5], [8], [9] systems and the accuracy is improved in every stages. The sampling [2], [4], [10] and histogram approaches are utilized in database environments to support approximate answering or selectivity estimation.

Big data is divided into independent partitions using balanced partitioning algorithm and FastRAQ approach summarizes the local estimates from all partitions to provide accurate results. The stratified sampling model is used to divide all data into different groups and separates each group into multiple partitions.

The cost of distributed range aggregate queries includes the cost of network communication and cost of local files scanning produced by data transmission, synchronization for aggregate operations and to search the selected tuples. The range aggregate query in big data environments gets faster result by minimizing the cost. In each partition, a sample is maintained for values of the aggregation column and a multi dimensional histogram for values of the index-columns. The local result is the product of the sample and an estimated cardinality from the histogram. The cardinality estimator is formulated in (1),

$$\sum_{i=1}^{M} \text{Count}_i * \text{Sample}_i \quad (1)$$

M is the number of partitions, Count_i is the estimated cardinality of queried ranges and Sample_i is the sample for values of aggregation column.

FastRAQ divides numerical value space of an aggregation column into different groups and maintain an estimation sketch in each group. In each partition, the sample and the histogram are updated by the attribute values of the incoming record. The estimate value in each partition is calculated based on the product of the sample and the estimated cardinality [7], [11]. The final return for the request is the sum of all the local estimates from each partition.

FastRAQ supports multi-dimensional range queries and it includes multiple buckets of the histogram. A Range Cardinality Tree (RC-Tree) includes three types of nodes, they are root node, internal nodes, and leaf nodes. The leaf node only keeps the statistical information and a new record arrives, it is written into the hash table. If the number of nodes in the hash table [14] reaches a threshold, the hash table flushes nodes into RC-Tree and appends the temporary files to formal bucket data files.

The processing of range aggregate queries [6] on large amount of data takes long time to provide accurate result. The query can be handled efficiently on big data environments using the FastRAQ approach.

The group identifier is generated for the dataset using stratified sampling and the partitioned dataset is clustered into buckets using K-means clustering algorithm. The range cardinality query algorithm is used to generate the efficient query result from the clustered dataset. The time complexity in retrieving the range aggregate query result is reduced.

2. STRATIFIED SAMPLING METHODS

This section provides the description of Post Stratification sampling and Horvitz Thompson Estimator in Stratified Sampling methods.

POST STRATIFICATION SAMPLING

Stratification is introduced after the sampling phase in a process is called post stratification. If the sampling phase is unknown, Then the post stratification sampling method is used to create a stratifying variable for the necessary information. The random sample method is used to improve the efficiency. The precision of a sample estimation and weight can be calculated with help of that sampling method.

The selected sampled units are classified into groups and the group totals are summed to produce an estimate for the whole population. The groups or post strata are formed and it contains the minimum number of sampled units. The classification of the sampling units is used to improve the precision of the sample estimates.

The post stratification weight is calculated by the use of auxiliary data set to compare with the sample data. It can be computed after collecting all data. Post stratification is used to increase the precision of estimates in unstratified sampling by using additional information of strata weights in the final estimator. It leads to more precise estimations than simple random sampling and involves assignment of units after selection of the sample.

The post stratification is combined with systematic sampling, the gain in precision is small. The assigning of sampling units to strata after observation of the sample, is imposed at the analysis stage. The sample sizes within strata cannot be fixed in advance but must be assumed random depending on the samples actually selected. Post stratification is applied if additional information about strata sizes is available. Post stratification is a method for adjusting the sampling weights for underrepresented groups in the population. Post stratification adjusts the sampling weights and results in smaller variance estimates. The stratified sampling limitations can be overcome with post stratification and it uses the weight variable for efficient data partitioning. An appropriate stratifying variable uses the lack of prior knowledge to implement the post stratification and uses simple random sample. It is a method for adjusting the sampling weights for underrepresented groups in the population.

Post stratification adjusts the sampling and replicate weights so the joint distribution of a set of post stratifying variables matches the known population joint distribution. It is used to reduce the sampling error and variance of estimates of the mean. Post stratification can be combined with stratified random sampling to improve the efficiency of data partitioning.

A. HORVITZ THOMPSON ESTIMATION

The Horvitz Thompson (HT) estimator is an unbiased estimator of the population total. This HT estimator is unbiased for the finite population total in repeated sampling and that can be quite inefficient due to the variation of selection probabilities.

The estimated population total is a combination of probabilities and survey variable values. The HT based estimates from one particular sample may be differing from the true total value and then the probabilities of selection are negatively correlated with the characteristic of interest. It is also used for determining the missing data in large datasets.

The inverse sampling design is used to divide the population into two subpopulations. The subpopulation contains few units of population. The HT estimator is derived for estimating the population mean based on the inverse sampling designs. The subpopulation sizes are known. The post stratification and HT estimator are not location invariant.

The HT estimator is an efficient estimator that finds the mean of the subpopulation is close to zero. The Horvitz Thompson estimator is described in (2),

$$T = \sum_{i=1}^{v} y/\pi_i \qquad (2)$$

sum is taken over the *v* distinct units in sample, v is the effective sample size, i is the unit of population, y is the variable of interest and π_i is the probability of population. It will not depend on the number of times a unit can be selected. The individual unit of the sample is utilized by once. It is used for deriving the variance with help of inclusion probabilities. The cluster formed in HT estimator is expressed in (3),

$$\hat{t}_{HT} = \frac{1}{n}\sum_{i\in S}\frac{\hat{t}_i}{\pi_i/n} = \sum_{i\in S}\frac{\hat{t}_i}{\pi_i} = \sum_{i=1}^{N} Z_i \frac{\hat{t}_i}{\pi_i} \qquad (3)$$

Z_i =1 the cluster *i* is in the sample and otherwise the value of Zi =0. The HT estimator of the total is the sum of the sampled units times and sample weights. It gives the minimum replacement designs that allow sampling units with large relative size measures to be selected more than once. The estimator improves the manageability, availability of large dataset and also used to boost the query processing performance.

ALGORITHM:

POST STRATIFICATION SAMPLING AND HORVITZ THOMPSON ESTIMATOR

Input: Q;

Q: Select sum (AggColumn) other Colname where l_{i1} <$ColName_i$ <l_{i2} opr l_{j1}<$ColName_j$<l_{j2}.

Output: S;

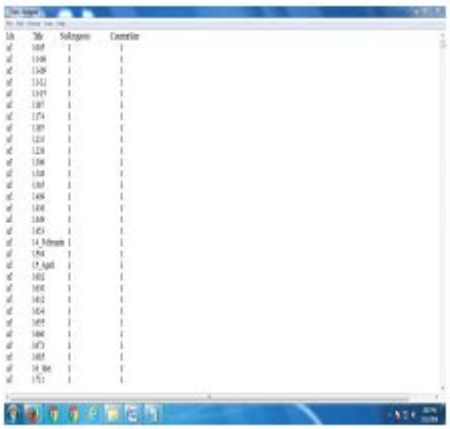
S: Range aggregate query result.

1. Deliver the request Q to all partitions;
2. for each $partition_i$ in partitions do
3. Compute the cardinality estimator of range l_{i1} < $ColName_i$ < l_{i2} from the local histogram, and let CE_i be the estimator of the ith dimensions;
4. Compute the cardinality estimator of range l_{j1} < $ColName_j$ < l_{j2} from the local histogram, and let CE_j be the estimator of the jth dimensions;
5. Merge the estimators CE_i and CE_j by the logical operator Opr, and compute the merged cardinality estimator CE_{merged};
6. $Count_i = ħ\ (CE_{merged})$;
   ħ is a function of cardinality estimation.
7. Compute the sample for AggColumn, and let $Sample_i$ be the sample;
8. $SUM_i = Count_i * Sample_i$;
   SUM_i is a local range aggregate query result;
9. end for
10. Set the approximate answering of FastRAQ as S. Let S= ∑ SUM_i
11. return S.

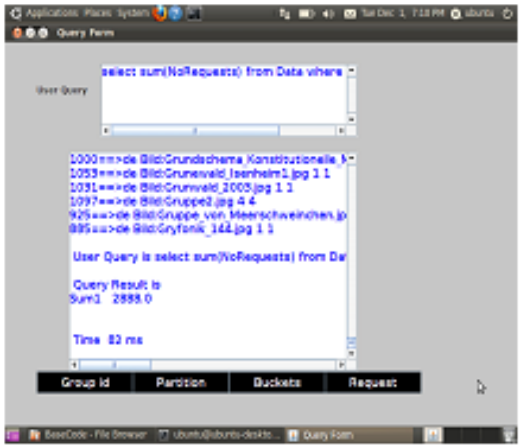
3. RESULTS

The algorithm for range aggregate query is experimented over dataset called Wikipedia page view statistics on 10,000 complex and low latency data are processed based on query and the results are obtained from the fast range cardinality query algorithm. It produces accurate result for range aggregate queries in big data environments and also takes time to provide results. The post stratification sampling method is used to improve the accuracy and to reduce the time complexity in retrieving the query result for range aggregate queries. The HT estimator is used for calculating the mean value. The mean value can be applied for improving the accuracy of the queried results and also used for reducing the time complexity of data updates and range aggregate query results.

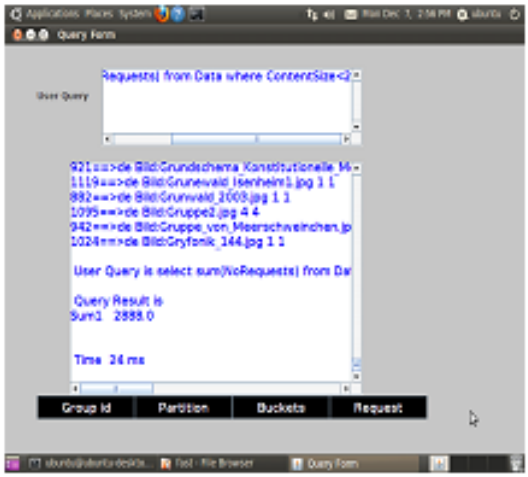
The sample dataset is shown in figure (a). The FastRAQ approach query result is shown in figure (b). The post stratification sampling method query result is shown in figure (c). The HT estimation query result is shown in figure (d). The comparison chart and table based on time complexity of FASTRAQ, Post Stratification Sampling and Horvitz Thompson Estimator approach is shown in figure (e) and figure (f).



(a)

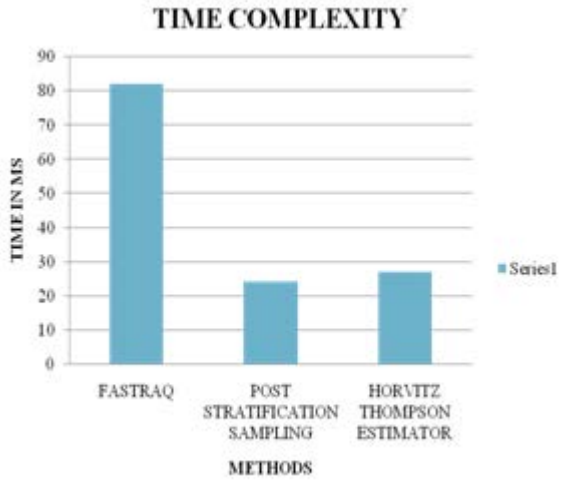


(b)



(c)

(d)



(e)

Comparison Table based on Time Complexity		
FASTRAQ	**POST STRATIFICATION SAMPLING**	**HORVITZ THOMPSON ESTIMATOR**
82 ms	24 ms	27 ms
62ms	17ms	18ms
125ms	38ms	39ms

(f)

Fig.1. Range Aggregate Query Result.
a) Sample Dataset. b) FastRAQ. c)Post Stratification. d) Horvitz Thompson Estimator. e) Comparison Chart. f) Comparison Table

4. CONCLUSION

Big data is a collection of large datasets and to handle the range aggregate queries in big data environments is challenging process. A fast approach to range aggregate queries is introduced to overcome the problems in big data environments. The collected page view statistics dataset from wikipedia is uploaded and it is partitioned and clustered to retrieve the accurate result in less processing time. The partitioning is performed using stratified sampling method and it is efficient in handling the large datasets. The Stratified sampling divides the value

of numerical space into different groups and sub groups. The precision of the sample estimates produced by stratified sampling can be improved using post stratification sampling method by performing stratification after sampling phase. The Horvitz Thompson estimator is used for estimating the total and mean of a super population in a stratified sample. The estimator is applied in survey analyses and used for accounting the missing data.

The post stratification sampling and Horvitz Thompson estimator can be combined with Enhanced K means and Clustering using Representative (CURE) clustering algorithm in future to improve clustering and to reduce the time complexity in retrieving the range aggregate query result.

REFERENCES

[1] Bilal K., Manzano M., Khan S., Calle E., Li K. and Zomaya A. (2013), 'On the characterization of the structural robustness of data center networks', IEEE Transactions on Cloud Computing, volume 1, no. 1, pp. 64–77.

[2] Chaudhuri S., Das G. and Srivastava U. (2004), 'Effective use of block-level sampling in statistics estimation', in Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, pp. 287–298.

[3] Choi H. and Varian H. (2012), 'Predicting the present with Google trends', Economics Record, volume 88, no. s1, pp. 2–9.

[4] Cohen E., Cormode G. and Duffield N. (2011), 'Structure-aware sampling: Flexible and accurate summarization', Proceedings on Very Large Data Bases Endowment, volume 4, no. 11, pp. 819–830.

[5] Condie T., Conway N., Alvaro P., Hellerstein J. M., Gerth J., Talbot J., Elmeleegy K. and Sears R. (2010), 'Online aggregation and continuous query support in MapReduce', in Proceedings of the ACM SIGMOD International Conference on Management of Data, Vienna, pp. 1115–1118.

[6] De Capitani di Vimercati S., Foresti S., Jajodia S., Paraboschi S. and Samarati P. (2013), 'Integrity for join queries in the cloud', IEEE Transactions on Cloud Computing, volume 1, no. 2, pp. 187–200.

[7] Flajolet P., Fusy E., Gandouet O. and Meunier F. (2008), 'Hyperloglog: The analysis of a near-optimal cardinality estimation algorithm', in Proceedings of International Conference on Analysis of Algorithms, Germany, pp. 127–146.

[8] Haas P. J. and Hellerstein J. M. (1999), 'Ripple joins for online aggregation', in ACM SIGMOD Record, volume 28, no. 2, pp. 287–298.

[9] Hellerstein J. M., Haas P. J. and Wang H. J. (1997), 'Online aggregation', ACM SIGMOD Record, volume 26, no. 2, pp. 171–182.

[10] Haas P. J. and Konig C. (2004), 'A bi-level bernoulli scheme for database sampling', in Proceedings of the ACM SIGMOD, International Conference on Management of Data ACM, China, pp. 275–286.

[11] Heule S., Nunkesser M. and Hall A. (2013), 'Hyperloglog in practice: algorithmic engineering of a state of the art cardinality estimation algorithm', in Proceedings of the International Conference Extending Database Technology, New York, pp. 683–692.

[12] Ho C. T., Agrawal R., Megiddo N. and Srikant R. (1997), 'Range queries in OLAP data cubes', ACM SIGMOD Record, volume 26, no. 2, pp. 73–88.

[13] Liang W., Wang H. and Orlowska M. (2000), 'Range queries in dynamic OLAP data cubes', Data Knowledge and Engineering, volume 34, no. 1, pp. 21–38.

[14] Malensek M. and Pallickara S. (2013), 'Polygon-based query evaluation over geospatial data using distributed hash tables', in Proceedings of the IEEE/ACM 6th International Conference on Utility Cloud Computing, New York, pp. 219–226.

[15] Mishne G., Dalton J., Li Z., Sharma A. and Lin J. (2013), 'Fast data in the era of big data: Twitter's real-time related query suggestion architecture," in Proceedings of the International Conference Management of Data ACM SIGMOD, New York, pp. 1147–1158.