ISSN 23499842(Online), Volume 2, issue 1 March 2017
International Journal of Innovative Trends and Emerging Technologies

SURVEY ON VARIOUS AGGREGATION MECHANISM IN CONTINUOUS QUERY PLAN

M.Madhankumara[1] and Dr. C. Sureshganadhas[2]
[1]*Research Scholar, Department of Computer Science and Engineering, Manonmaiam Sundaranar University, Tamil Nadu,*
[2]*Professor& Head, Department of Computer Science and Engineering, Vivekanandhan College of Engineering for Women, Elayampalayam, Thiruchencode, Tamil Nadu.*
E-Mail :[1] manomadhanbe@gmail.com, [2]sureshc.me@gmail.com

Abstract
Continuous queries monitor the dynamic changes in continuous time data. The monitoring is carried out to provide a top – k decision making query node in the continuous data system. Since, the user obtains the query result from averaging aggregation function from a continuous set of data. The major challenge is to monitor and find the dynamic query plan node in distributed environment. The coherency, completeness and reliability of the query data is important in such cases. The ideal investigation over these constraints that includes techniques that helps in improving the query plan strategy in specified dataset. This paper investigates various data aggregation mechanism that employs techniques related to best query plain in continuous aggregation data. The study has concentrated majorly on the achievements done in distributed databases, constraints due to multiple query and overhead problems and the data-dependent model.

Index terms: Continuous Data, Query Plan, Dataset, Data Aggregation, Optimization.

1. Introduction

With increasing online application, dynamic data is increasing significantly and the interest in automated systems that delivers the updates is increasing prominently. If a user inputs a query to a system, the query plan from various sources are aggregated and provides an average results to the user. This method is considered as the query aggregation mechanism in continuous query data. The aggregation data is a long run data that changes continuously and when the criteria is achieved, the notification is sent to the user. The periodical update or refreshing of query is done repeatedly in continuous query applications. The users prominently receives inaccurate results from the query search and use of effective mechanisms helps in providing accurate results that matches the user's requirement [20]. This is done by finding the best or top-k data from the real world or synthetic distributed datasets and producing it to the user.

Initially, the filtering of incoming data streams has to be initiated continuously that requires time and space constraints. The elimination of such filtering at initial stage, leads to complex data in the streams. This increases the complexity and reduces the robustness of the query mechanism in terms of its effectiveness and scalability. The data selected after the query at front end is collected and grouped using similarity measuring techniques. Then the average value of the aggregated value is selected or top-k value is selected based on varying mechanism. The selected averaged or top-k data is then sent to the user to satisfy his/her request. However, the queries searches through the database available worldwide, efficiency of query plan is still a threat to such mining systems. So, a need of optimal mechanism or optimization of querying plain is required for effective query retrieval.

The processing of aggregate queries is done by sending periodically the partial aggregates from source node to sink node and finally the aggregates are computed. In certain application, the queries varies, such that the aggregators or the techniques also vary. These includes various sliding windows, various aggregation operators and various predictors. The summary information over data streams may be summed, counted or averaged. However, systems having multiple queries sends aggregators for each individual data and leads to scalability issues. Many mechanism had been proposed to overcome the shortfalls occurs in querying the aggregate data.

The main advantage of the continuous query plan model is that it reduces the processing time and the cost of the system. The processing is carried out within incoherency bound and this leads to elimination of messages that are redundant to the users. This method achieves high throughput and reduces the energy consumption in certain sensors related environment, and low latency. Optimization involves a greater degree of assured relevant data in redundant space. The examples of such systems include its application in telecom fraud detection, traffic surveillance, stock market exchange, router and sensor fluctuating rates, etc. [2].

This paper provides a deep study of recent query planning mechanisms in continuous online query data. The method includes both real world and synthetic datasets. The technique is discussed based on problems occurring in multiple queries, correctness, distributed database and effectiveness of data-dependent model in distributed environment over real-world datasets.

The remaining part of the paper is as follows: Section 2 discusses the deep insights of recent techniques proposed in querying continuous aggregators. Section 3 provides the summary on the above study.

2. Related Works

2.1 Multiple Query Problems:

The attention over distributed stream processing systems has increased widely that involves generation of data streams by distributed source nodes. Here, the queries are executed continuously and multiple queries are processed effectively. However, processing such multiple streams is a bit tougher task. A periodical partial aggregate is sent by source node that involves computation and results of local stream to sink node. This leads to communication overhead, reduced scalability and performance, since each the multiple queries are aggregated and processed.

Lee et al. (2008) proposed attribute selection construct scheme for selecting the predicates either indexed or grouped continuous queries. The scheme divides the data stream attribute domain into a disjoint set, depending on the selected predicates. The disjoint sets are pre-computed with the existing selected predicate results. The main advantage of the scheme is that it can compute the entire selected predicates at same instant of time. Also, it can monitor the evaluation criteria like dropping ratio of tuple and its selectivity in a dynamic manner. A run-time matching algorithm is designed to evaluate the performance of multiple queries through periodical capturing of dropping ratio [1].

Patroumpas and Sellis (2011) studied monotonic semantic existence over rich windows set for effective management over dynamic data. Additionally, the investigations are carried out in common windows variants and adaptations of operators like union, join and aggregation. Syntactic equivalent expressions for optimizing queries that involves windows is validated against the datasets that holds algebraic functions. The investigations over the windowed patterns helped in improving the insights on stream processing mechanism [2].

Similar to [2], Jeon et al. (2014) [5] proposed collective processing mechanism for aggregating multiple queries. A small partial set is found that aggregates and reduces the message transmission between the source node and sink node. The inclusion and exclusion of query node is handled using on the fly mechanism.

Han et al. (2014) [6] proposed join aggregation operation that returns approximate query rather than exact query though combining two or more tables. A p, ε-approximate join aggregation algorithm is designed to obtain join results over a specified confidence interval. Further, JRS and JPIPT is used to improve the response time by joining the tuples and the random tuples are obtained by sampling algorithm that determines the correctness and speed of retrieval.

2.2 Overhead Problems:

Lange and Naumann, (2013) proposed query planning mechanism that specifies the use of similarity index over thresholds and combination of the results. The similarity search is carried out in non-static querying environment that selects the query with maximum similar results within the query limit. The use of compile and query time procedure improved the completeness and reduces the cost involving in similarity comparisons in real time datasets [4].

Xu and Pottinger, (2014) uses query translation using decomposition aggregation query three layer structure. The aggregate queries are created using this semantic integration technique over heterogeneous data sources. Aggregation rewriting algorithm plays a wide role in query translation and creation of new semantics. The overhead processing during query answering and duplication removal through data source determination towards the translated decomposition query. Various solvers are used to optimize the source selection and NP-hard problems against real- world and synthetic data sets [7].

Bae et al. (2015) proposed aggregation over cryptographic algorithm in smart grid technology. The technique helps in reducing the network time and delay due to large messages and overheads. Additionally, the user privacy is preserved and helped in overhead reduction, thus reducing complexity using homomorphic part encryption algorithm [8]

Cui et al. (2015) proposed data aggregation in boundary collision using entropy histogram threshold and probability distribution function. The former technique finds the boundary collisions and the size of zones and then the distribution function is applied over this data. At last, the collision density ratio is used to aggregate the collided data near boundaries into neighbourhoods [9] could reduce the overhead problems. However, the application of collision in boundary region (divided into zones) could be used in real and synthetic datasets. Panah et al. (2016) maintained the integrity and authentication of messages using detectable hidden marks and verification of trust of each data sources. This method is constructed using watermarking technique that uses synchronization marks in data aggregation streams and protection of data at end points at data layer. The proposed method is tested over both real and synthetic sensor data environment [18].

2.3 Distributed database:

Sagy et al. (2011) proposed top-k aggregation mechanism to find the top scoring objects in distributed database. The score of each object is found by applying aggregation and scoring function over attribute vectors, thus forming aggregated attribute vector. The combination of skyline and geometric mechanism is adopted over the framework that defines the local constraints over real datasets. Using such framework, a top-k algorithm is designed with four phases that involves elimination and improvement over lower bound objects and finding out the top-k objects over it. Further, it also determines the upper bound objects using threshold level for reducing the computation cost and processing time [3].

Mishra and Singh, (2015) proposed parameterless optimization method to reduce the computational overhead due to storage of query relations at various sites. This reduces the computational complexity in exploring all the query plans at stored locations. Teaching learner optimization with multiple objective function is proposed to produce less complexity in constrained and unconstrained standard databases. Top – k query plans [3] is obtained by using multi-objective genetic algorithm over distributed database with higher query relations [10]. Bustince et al. (2016) discussed the problems in data aggregations due to improper penalty functions. A standard function related to averaging of aggregation functions are proposed as penalty functions. The method implies mainly on continuous aggregation data and moreover, quasi penalty function is implied over non-monotonic averaging functions [13].

Mousazadeh and Ladani, (2015) proposed secure pull adjust gossip algorithm in fault and dynamic distributed data systems. Here, the data aggregation is carried out with a full protection to the source messages and removes the malicious nodes. The attacks are carried out in hostile environments by finding out the malicious nodes using this averaging algorithm and replacing the nodes with new nodes. The distortion of

aggregation values are reduced largely with this method in real-world datasets [11]. Nechifor et al. (2015) proposed data aggregation method over electrical application that collects the data related to phasor measurements in Nottingham and Manchester locations [12]. Similar sensor data measurement in wireless sensor networks is proposed by Wu et al. (2016). This method is called as scalable privacy preservation big data aggregation that pre-establishes the topologies, nodes and clusters. The sensor data is aggregated based on the sink configuration message. The privacy preservation employs both inter and intra clustering data aggregation that reduces the energy consumption [19]. Furthermore, to improve the security in data aggregation environment, Duque (2016) [14] proposed data aggregation in affine spaces through shifted projection protocols. This works under the suitability of parameter in the distributed data environment.

2.4. Data-dependent model

Elkano et al, (2016) proposed a t-norm replacement in interval valued aggregation and formed integer based fuzzy aggregation function. This method is meant to study the effectiveness of aggregation in inference process, rule generation and matrix generalization of inferences. This method proved to produce results with better clarity and reduces the error occurring due to aggregation function [15]. Garg and Garg, (2016) proposed a similar fuzzy related to ordered weighted aggregation function. The technique is employed in time series domain and in forecast model, which possess vague, imprecise and uncertain data. Monotonic quantifiers are used with priority matrix that reduces the root means square and average error rate [16]. The method proposed in [15] works well with cardiovascular risk prediction and [16] works well with forecasting data. Further, a relational model is proposed by Kao (2016), which divides/decomposes the network system relationship to a weighted function. The weighted factor is adjusted with a factor and proves best aggregation of data in baseball example [17].

3. SUMMARY

The study mainly concentrated on the effectiveness of continuous query techniques in distributed datasets. Mainly, the techniques considered are related to multiple query problems, distributed database, overhead problems that includes its effectiveness and finally the data-dependent model. The methods conclude that a solution could be achieved, when there is a problem associated with continuous query plan with query manipulators. The present study concentrated mainly on continuous query data or plan in real-world and synthetic datasets. Finally, the mechanism in data-dependent model could be used in proposing a new problem formulation for constraints associated with query plan.

7. References

[1]. Lee, H. H., Yun, E. W., & Lee, W. S. (2008). Attribute-based evaluation of multiple continuous queries for filtering incoming tuples of a data stream.Information Sciences, 178(11).

[2]. Patroumpas, K., & Sellis, T. (2011). Maintaining consistent results of continuous queries under diverse window specifications. Information Systems, 36(1), 42-61.

[3]. Sagy, G., Sharfman, I., Keren, D., & Schuster, A. (2011). Top-k vectorial aggregation queries in a distributed environment. Journal of Parallel and Distributed Computing, 71(2), 302-315.

[4]. Lange, D., & Naumann, F. (2013). Cost-aware query planning for similarity search. Information Systems, 38(4), 455-469.

[5]. Jeon, J. H., Lee, K. Y., & Kim, M. H. (2014). Communication-efficient processing of multiple continuous aggregate queries. Information Sciences, 284, 1-17.

[6]. Han, X., Li, J., & Gao, H. (2014). Efficiently processing (p, ε)-approximate join aggregation on massive data. Information Sciences, 278, 773-792.

[7]. Xu, J., & Pottinger, R. (2014). Integrating domain heterogeneous data sources using decomposition aggregation queries. Information Systems, 39, 80-107.

[8]. Bae, M., Kim, K., & Kim, H. (2016). Preserving privacy and efficiency in data communication and aggregation for AMI network. Journal of Network and Computer Applications, 59, 333-344.

[9]. Cui, G., Wang, X., & Kwon, D. W. (2015). A framework of boundary collision data aggregation into neighbourhoods. Accident Analysis & Prevention, 83, 1-17

[10]. Mishra, V., & Singh, V. (2015). Generating optimal query plans for distributed query processing using teacher-learner based optimization. Procedia Computer Science, 54, 281-290.

[11]. Mousazadeh, M., & Ladani, B. T. (2015). Gossip-based data aggregation in hostile environments. Computer Communications, 62, 1-12.

[12]. Nechifor, A., Albu, M., Hair, R., & Terzija, V. (2015). A flexible platform for synchronized measurements, data aggregation and information retrieval.Electric Power Systems Research, 120, 20-31.

[13]. Bustince, H., Beliakov, G., Dimuroa, G. P., Bedregal, B., & Mesiare, R. (2016). On the Definition of Penalty Functions in Data Aggregation. Fuzzy Sets and Systems.

[14]. Fernández-Duque, D. (2016). Perfectly secure data aggregation via shifted projections. Information Sciences, 354, 153-164.

[15]. Elkano, M., Sanz, J. A., Galar, M., Pękala, B., Bentkowska, U., & Bustince, H. (2016). Composition of interval-valued fuzzy relations using aggregation functions. Information Sciences, 369, 690-703.

[16]. Garg, B., & Garg, R. (2016). Enhanced accuracy of fuzzy time series model using ordered weighted aggregation. Applied Soft Computing, 48, 265-280.

[17]. Kao, C. (2016). Efficiency decomposition and aggregation in network data envelopment analysis. European Journal of Operational Research.

[18]. Panah, A. S., van Schyndel, R., & Sellis, T. (2016). Towards an asynchronous aggregation-capable watermark for end-to-end protection of big data streams. Future Generation Computer Systems.

[19]. Wu, D., Yang, B., & Wang, R. (2016). Scalable privacy-preserving big data aggregation mechanism. Digital Communications and Networks, 2(3), 122-129.
[20]. Gupta, R., & Ramamritham, K. (2012). Query planning for continuous aggregation queries over a network of data aggregators. IEEE Transactions on knowledge and Data Engineering, 24(6), 1065-1079.