Contemporary Engineering Sciences, Vol. 7, 2014, no. 14, 725 - 730
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ces.2014.4681

Adaptive Aggregation Scheduling Using Aggregation-degree Control in Sensor Network

Misook Bae

Institute of Information Science and Engineering Research
Mokpo National University, Republic of Korea

Min A Jeong, Seong Ro Lee

Dept. of Information & Electronics Engineering
Mokpo National University, Republic of Korea

Abstract

Data stream processing applications have recently increased. In sensor network these applications use in-network processing or aggregation to process continuous and unbounded data stream. However, it may lead to delaying total execution time and increasing data loss. In previous work, total number of transmitting messages increases, since data aggregator should send refresh messages to acquire the results that satisfy tolerable error bound the client specified. Thus, it needs aggregation scheduling method to minimize reduced accuracy and to reduce the number of transmitting messages. We propose an AASM (Aggressive Aggregates-Scheduling Method) method using the reliable prediction interval to filter just related data and the aggregation-degree control to reset window size according to satisfying with answering client's requirements to tolerable error bounds. Our proposed algorithm simulation resulted in the better performance in terms of reducing total number of transmitting messages and minimizing data accuracy loss.

Keywords: Aggregation, Data accuracy, Network efficiency

1 Introduction

Real time data stream processing applications such as WSN monitoring system, traffic control system or health monitoring system are demanded to analyze continuous, unbounded and geographically distributed data stream and

specialized real-time requirements and infinite data streams. Sensor network is supposed to support reliable data from sensor field to user and is responsible for collecting and monitoring them. Sensors can collect a lot of data spatially as well as temporally. Network sends them to sink node or drops unrelated data, since WSN (wireless sensor network) does not store all of continuous and unbounded data stream in network for effective energy in network. Aggregation collects and combines the data to consume energy efficiently in the network when data correlation is high. It reduces data volume to transmit and increases network life time by decreasing number of transmitting messages. But it may lead to delay total execution time and increase data loss. Therefore, it is necessary for effective aggregation method to obtain minimum data loss and network efficiency.

2 Related work

Nowadays, numerous researches have been conducted to process data stream efficiently. Real time data stream processing applications are various as financial data analysis, large sensor network, etc. The research in [1,2,5] have presented distributed stream processing system and indicated the methods to process them before data exchange include aggregation, data fusion, average, etc. Aggregation is researching much more for method to increase network efficiency and appearing various approaches. In [2,4], authors proposed push-based scheme using data filters at the sources and present method to minimize the refresh message use by assign user' incoherency bounds. In [3,5], authors dealt with grouping or compression to eliminate data redundancy and aggregation scheduling to cluster nodes, and they proposed effective distributed algorithm to produce conflict-free aggregation scheduling in WSN.

3 AASM

We depict the proposed aggregates scheduling method in this section. We now address our proposed algorithm as AASM. This paper restricts to monitoring system that deals with environmental sensing data and represents the aggregating process to aggregate input tuples and to send their results to sink node using proposed aggregation scheduling method to minimize data accuracy loss and total transmitting messages. In monitoring system, sensed data values are composed of similar values within specific interval generally. Fig 1.shows input phase of the input stream to process. A sensed value as input tuple ranges from 5 to 15 in consequence of the analysis result of previous data. It assumes S2 is an arbitrary source node.

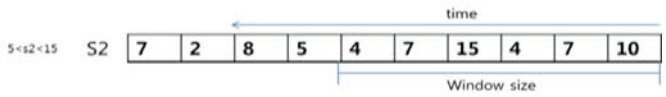

Figure 1. Example of input stream processing

But these input tuple stream may include the values out of its range. This results in decreasing degree of accuracy and increasing delay time due to including unnecessary processing of unrelated data. In addition, existing research generally fixes compression ratio due to restricted memory space, it may result in data accuracy loss. And it may result in increasing the total number of transmitting messages, since it has to refresh in case of not satisfying with answering client's requirements to tolerable error bounds. As the number of refresh messages increased, so do communication cost. To solve these problems, we propose effective aggregation scheduling method to produce reliable prediction interval to increase degree of accuracy and to aggregate the values within its interval using aggregation degree control.

Proposed method produces the confidence interval of cached data and drops the values out of range in its interval. We use two queues, i.e. IQF to be used in collecting the tuples to be filtered unrelated tuples using prediction interval and to be dropped and IQ to be used in collecting the tuples within its interval to be aggregated. Prediction interval uses standard deviation, variance and average are based on cached tuples with a 95 percent confidence level. The values out of range in its interval are dropped, classified abnormal data, transmitted to IQF queue and informed meta data information "abnormal" to sink node. The values within its interval are transmitted to IQ queue and aggregated. Fig 2.show a data aggregator structure to implement the proposed method.

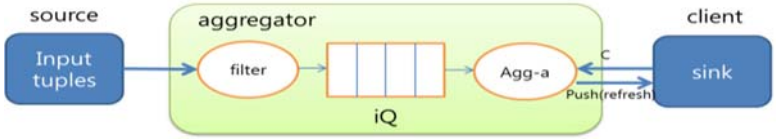

Figure 2. Proposed aggregation scheduling model

This paper collects tuples to window using temporal aggregation characteristic that aggregate values between the tuples acquired from sensor within given time interval and groups tuples in moving window. The p.s.r(partial state record) includes all readings in a window. When a new tuple incomes, window is updated continuously. To satisfy with client's requirements to tolerable error bounds, window size is controlled. The tuples in the window to be reset is finally aggregated, stored the result in r.s.r(resulting state record) and deposit it to output stream. And this paper assumes push based mechanism that data source sends refresh message to client by itself in network data aggregators, the tuples input to input queue regularly, and a client submits query with incoherency bound (Cq). Proposed method composes aggregation set included only values within prediction interval to come close to Cq, i.e. that does not compute average or merge data set collected for a certain period but composes aggregation set included only values within its interval and come close to value less or equal Cq. Next, proposed method inserts an input tuples to satisfy with answering client's requirements to tolerable error bound to IQ queue to aggregate, and drops an input

tuple considered as unrelated values out of range in its interval and sends it to sink node with “abnormal” state information. The following is we aggregate the tuples in IQ in window size, check whether the tuple satisfies client’s incoherency bound or not. If answering client’s requirements to tolerable error bounds satisfies, it transmits aggregated value to sink node, whereas if not, it repeats aggregation processing phase to augment tuples in the window increasing window size until answering client’s requirements to tolerable error bounds is satisfied. In previous work, total number of transmitting messages increases, since data aggregator should resend refresh message after it sends aggregating result, if not satisfy with answering client’s requirements to tolerable error bound. Instead, proposed method check client’s incoherency bound before sending aggregating result. If not satisfying with answering client’s requirements to tolerable error bound, proposed method repeats aggregation process to increase the data set to aggregate until tolerable error bound does not exceed incoherency bound, and after that, sends aggregating result to sink node. Thus, proposed method need not send refresh message and have only to send result messages. After all, our method does not allow data redundancy, thus it avoids memory waste. And filtering process to understand correlation of tuples and drop unrelated tuples and aggregating process using AASM result in minimizing data accuracy loss and decreasing total number of transmitting messages.

4 Performance evaluation

AASM algorithm was implemented using Java Platform (JDK) 7u25. To evaluate the performance of proposed algorithm, we used total number of messages, elapsed time and accuracy ratio as performance evaluation parameters and compared the proposed method (AASM) with the known existing method (EMD) with variation of performances of a system for the changes of the number of input tuples, initial window size, and incoherency bound respectively. The total number of messages means the number of messages used to acquire processed results, i.e. sum of the number of refresh messages and the number of produced result messages. Elapsed time means total time to take to execute from inputting tuples to producing the results to satisfy terms the user requested. Its unit of measure is millisecond. Accuracy ratio means rate of average of the tuples aggregated to average of all tuples. We use the variance of phase incoherency bound (ib):50,100,150,200,250 and 300 and initial window size (ws): 10,50,100,150,200 and 300 respectively, and experiment the evaluation parameter according to the number of input tuples (#tuples): 10000,15000,20000,30000, 40000 and 50000. Firstly, we experimented to measure the effects of incoherency bound change rate on total number of messages and elapsed time as shown in the Fig. 3. AASM is remarkably smaller than EMD in total number of messages for varying incoherency bound regardless of number of input tuples. This paper presented only in case of #tuples:40000 and ws:50 for lack of space and was seen

similar results in all the rest of it.

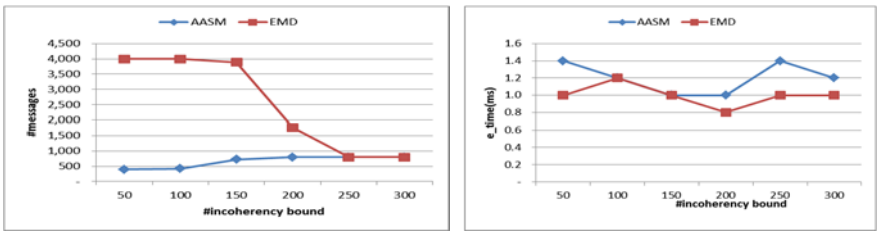

Figure 3. Total number of messages and elapsed time for incoherency bound

Secondly, we experimented to measure the effects of initial window size change rate on total number of messages and elapsed time. In Fig.4, AASM is remarkably smaller than EMD and displays a slow slope in total number of messages for varying initial window size regardless of number of input tuples. This paper presented only in case of #tuples:40000 and ib:100 for lack of space

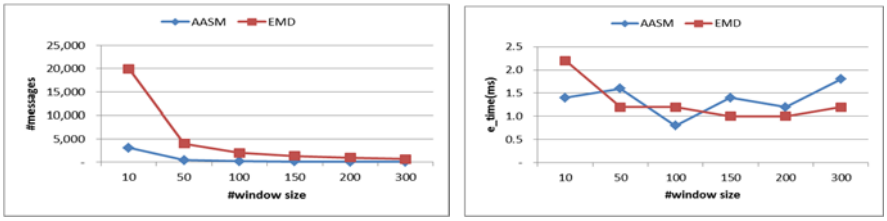

Figure 4. Total number of message and elapsed time for initial window size

Thirdly, we experimented to measure the effects of the number of input tuples change rate on total number of messages and elapsed time. This paper presented only in case of ws:100 and ib:100 for lack of space.

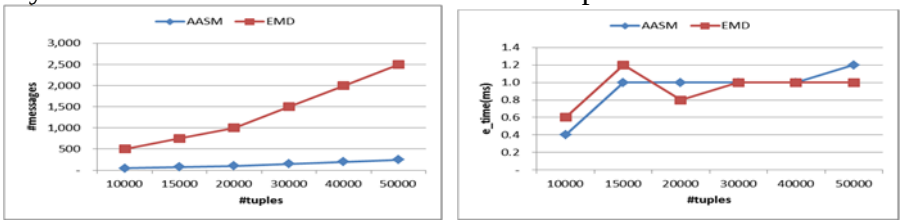

Figure 5. Total number of messages and elapsed time for number of input tuples

As shown in the Fig.5, EMD has a steep incline whereas AASM reduces remarkably total number of messages and ascends gradually whenever number of input tuples increases. A steep incline on number of messages may increase communication cost and lead large overhead of network. After all, our method reduced network energy consumption due to remarkable reduction and gentle incline for total number of messages whenever input tuples increase and it showed AASM takes time to similar to EMD in total elapsed time in this every case. Additionally, in this experiment, it showed the reduced accuracy minimized, since it makes little difference between AASM and EMD in accuracy ratio and omitted.

5 Conclusion

This paper presented an efficient aggregation scheduling method to resolve reduced accuracy of query execution results and increment of number of refresh messages due to aggregation. We have proposed an AASM method using a reliable prediction interval of cached data and the aggregation degree control to reset window size according to satisfying with answering client's requirements to tolerable error bounds and compared an AASM method with the existing method using total number of messages, elapsed time and accuracy ratio as the measure of performance evaluation. Though our AASM lead overhead of the aggregation degree control, it minimized data accuracy loss and increased network efficiency due to reduction of total number of messages. Consequently, we have found out that there is an improvement in the system performance.

Acknowledgements. This work was supported Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2009-0093828) and by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the C-ITRC(Convergence Information Technology Research Center) support program (NIPA-2014-H0401-14-1009) supervised by the NIPA(National IT Industry Promotion Agency).

References

[1] Bo Yu, Jianzhong Li, Yingshu Li, Distributed Data Aggregation Scheduling in Wireless Sensor Networks ,INFOCOM 2009,(2009)
[2] C. Olston, J. Jiang, and J. Widom, Adaptive Filter for Continuous Queries over Distributed Data Streams, Proc. ACM SIGMOD Int'l Conf. Management of Data, (2003)
[3] Gulisano, V. Jiménez-Peris, R. Patiño-Martínez, M. Soriente, C. Valduriez, P. StreamCloud: An Elastic and Scalable Data Streaming System, Parallel and Distributed Systems, IEEE Transactions, (2012), pp.2351 - 2365
[4] Gupta, Rajeev ; Ramamritham, Krithivasan , Scalable Execution of Continuous Aggregation Queries over Web Data, Internet Computing, IEEE Volume: 16 , Issue: 1 , (2011) , pp.43 - 51
[5] Rong He, Zhongzhi Luan, Yuanqiang Huang, Gang Guan, Zhendong Cheng, Depei Qian, Providing high availability for Distributed Stream Processing Application with Replica placement, 15th International Conference on Network-Based Information Systems, (2012), pp.685~690

Received: May 1, 2014