INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

EFFICIENT MONITORING OF DATA STREAMS BY HIBERNATION AND PIGGYBACKING

K.Prakash[1], Mr.G.Balakrishnan[2]

[1]M.E. CSE, Krishnasamy College of Engineering and Technology, S.Kumarapuram, Cuddalore, Tamil Nadu – 607109, India, E-mail id: prakashcse02@gmail.com.
[2]Associate Professor, Krishnasamy College of Engineering and Technology, S.Kumarapuram, Cuddalore, Tamil Nadu – 607109, India, E-mail id: gbalki@gmail.com.

Abstract— Monitoring data streams in a distributed system is the focus of much research in recent years. The monitoring problem consists of determining whether the value of a global function defined on the union of all streams crossed a certain threshold. In existing system, a general approach for monitoring heterogeneous streams (HGM) is used, which defines constraints tailored to fit the data distributions at the nodes. HGM provides a practical solution, which reduces the running time by hierarchically clustering nodes with similar data distributions. HGM also present a method for recovering from local violations at the nodes, but it triggers communication only after a safe-zone breach occurs. In our proposed system, we are creating a monitoring tool which secures the database with hibernation. It also monitors the attack, if any found it will raise an alarm and blocks the attacker if necessary. Hibernation will secure the database with object security. The Piggybacking method is used for reducing communication overhead. Our Experiments yield an improvement of over an order of magnitude in communication and data security relative to HGM.

Index Terms— Heterogeneous data streams, Distributed streams, Hibernation, Piggybacking, safe zones.

1 INTRODUCTION

Many emerging applications require processing high volume streams of data. Examples include network traffic monitoring systems, real-time analysis of financial data, distributed intrusion detection systems and sensor networks. For a few years now, processing and monitoring of distributed streams has been emerging as a major effort in data management, with dedicated systems being developed for the task [1]. This paper deals with the monitoring of threshold queries over distributed data streams. Such queries are the building block for many algorithms, such as top-k queries, anomaly detection, and system monitoring. They are also applied in important data processing and data mining tools, including feature selection, decision tree construction, association rule mining, and computing correlations. Another important application is data classification which is often achieved by thresholding a function such as the output of a neural network or support vector machine.

In prior work, each node monitors a convex subset, often referred to as the node's safe-zone, of the domain of these data vectors, as opposed to their range. What is guaranteed in the geometric monitoring approach is that the global function will not cross its specified threshold as long as all data vectors lie within their corresponding safe-zones. Thus each node remains silent as long as its data vector lies within its safe zone. Otherwise in case of a safe-zone breach, communication needs to take place in order to check if the function has truly crossed the given threshold.

A crucial component for reducing the communication required by the geometric method is the design of the safe zone in each node. Nodes remain silent as long as their local vectors remain within their safe-zone. Thus, good safe zones increase the probability that nodes will remain silent, while also guaranteeing correctness: a global threshold violation cannot occur unless at least one node's local vector lies outside the corresponding node's safe-zone.

However prior work on geometric monitoring has failed to take into account the nature of heterogeneous data streams, in which the data distribution of the local vectors at different nodes may vary significantly. This has led to a uniform treatment of all nodes, independently of their characteristics, and the assignment of identical safe-zones (i.e., of the same shape and size) to all nodes.

As we demonstrate here, designing safe-zones that take into account the data distribution of nodes can lead to efficiently monitoring threshold queries at a fraction (requiring an order of magnitude fewer messages) of what prior techniques achieve. However, designing different safe-zones for the nodes is by no means easy. The technique uses hierarchical clustering of nodes, based on the similarity of their data distributions. The main goal of the algorithm is to reduce communication by minimizing the number of safe zone breaches by the local vectors (we refer to such breaches as local violations). It also presents a method for efficiently recovering from local violations, by carefully picking the nodes to communicate with.

2 RELATED WORK

Previous methods for reducing communication in distributed systems include sketching [7]. Other research concerns detecting "Heavy hitters" [8], computing quantiles [10], counting distinct elements, optimal sampling [11], distributed monitoring of decision tree [9] and ranking [12]. Theoretical analysis of the monitoring problem is provided and some nonmonotonic functions of frequency moments are treated [6]. The BBQ system [3] constructs a dynamic probabilistic model for the collection of sensor measurements. The system determines whether it is possible to answer queries only given the model values, or whether it is necessary to poll some of the sensors. In [6], data uncertainty in monitoring linear queries over distributed data is handled by fitting the data with a probabilistic model and devising an optimal monitoring scheme with respect to this model.

A great deal of work was dedicated to distribute monitoring of monotonic functions, usually weighted averages, max and min operators. Querying non-monotonic functions by representing them as a difference of monotonic ones is presented in [6], but for a static database. Ratio queries over streams are treated in [4], based on a dynamic model which monitors the local ratios vis-à-vis optimally chosen local thresholds. We also handle ratio queries, but in a different method, which is based on optimizing with respect to the data distribution at the nodes. The work in [2] does handle the case of issuing an alert if the ratio of two aggregated (over time) dynamic variables crosses a certain threshold, but does not naturally extend to handle instantaneous (based only on the current values) ratios, such as the ones that we target in our experiments.

In [5] it was proved that all existing variants of GM share the following property: each of them defines some convex subset C of A (different methods induce different C's), such that each safe-zone S_i is a translation of C. In [2] the monitoring process, each node N_i is assigned a subset of the data space, denoted as S_i its safe-zone such that, as long as the local vectors are inside their respective safe zones, it is guaranteed that the global function's value did not cross the threshold; thus the node remains silent as long as its local vectors remain within their safe-zone. In case of a local violation (safe-zone breach), communication needs to take place in order to check if the function has truly crossed the given threshold.

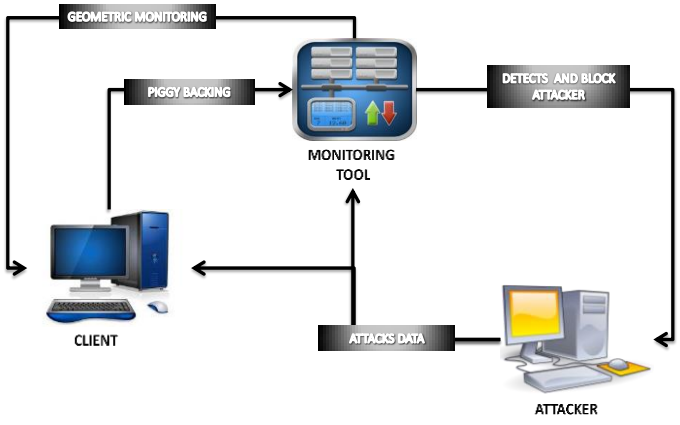
3 PROPOSED WORK

The aim of our work is to provide efficient data security and reduce the communication overhead. Our proposed work uses two different techniques namely Hibernation and Piggybacking. Hibernation will secure the database with object security. The Piggybacking method is used for reducing communication overhead. In our proposed system, we are creating a monitoring server which secures the database with hibernation. It also monitors the attack, once if it found any attack from the attacker then it will raise an alarm and blocks the attacker if necessary.

The monitoring server uses the Geometric Monitoring approach used in prior work to monitor the nodes connected in the network. Every node in the network creates a safe-zone in their database and nodes having similar data distributions are hierarchically clustered using the top down or bottom up approach. Since the constraints at all nodes share an identical structure in the Geometric Monitoring approach, we use the technique of hierarchical clustering of nodes. So that it is not necessary for the monitoring server to monitor all the data distributed in each node, rather it monitors only the safe-zones created in each node. This reduces the running time to communicate with all the nodes connected in the network and efficiently monitors the safe-zones from the attacks.

4 ARCHITECTURE

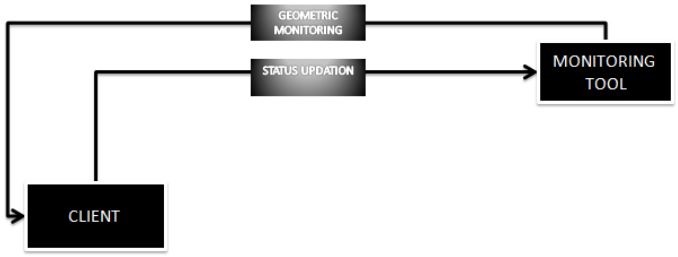
The architecture has three components namely Monitoring tool, client and an attacker. The monitoring server uses the Geometric Monitoring approach to monitor the clients connected in the network. It uses the piggbacking information from the clients to know about the status of the clients in the network and checks if any user attacks any confidential data from the clients, if so it first gives a warning and then blocks the attacker if necessary.



Although if the attacker tries to access the data, the Hibernation method will prevent the attacker by providing object security, so that the attacker can only attack the object and not the data in it. At that time the monitoring tool will block the attacker by disconnecting it from the network.

5 IMPLEMENTATION

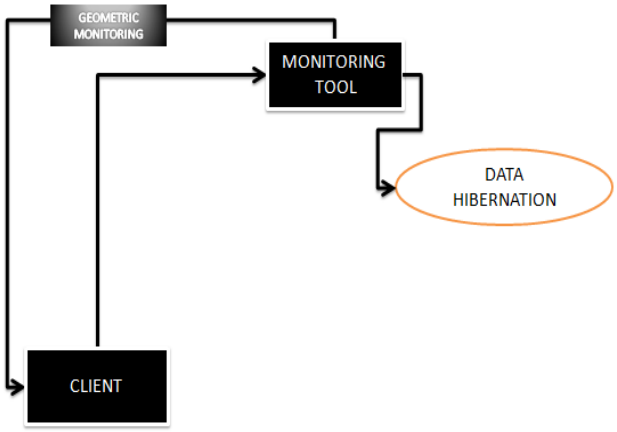
1. MONITORING SERVER CREATION



The monitoring server uses the Geometric Monitoring approach to monitor the clients connected in the network. It uses the piggbacking information from the clients to know about the status of the clients in the network and checks if any user attacks any confidential data from the clients, if so it first gives a warning and then blocks the attacker if necessary Monitoring server will be created to monitor all the clients for security threats. If any threat is detected then an indicator will flash on the screen. We can control the attack by disconnecting the attacker from the network.

2. HIBERNATION

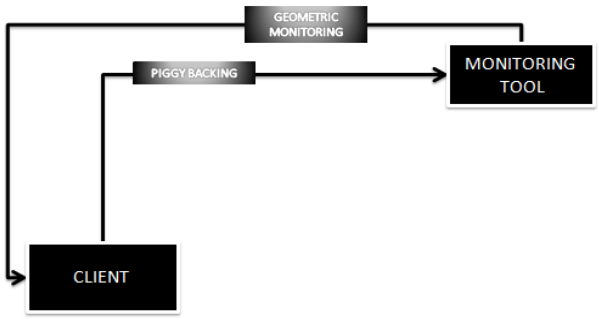
The data being stored in the database and retrieved from the database will be done with the help of hibernation. Hibernating any data will provide high object security to the database. It means that Hibernation will secure the data by covering the data with an object. The object will then act like a shield securing the data from the attacks.



If any client needs to transfer the data to other client or to store and retrieve the data from the database, the data will be hibernated and secured with object security. So that the attacker can only attack the object and not the data in it.

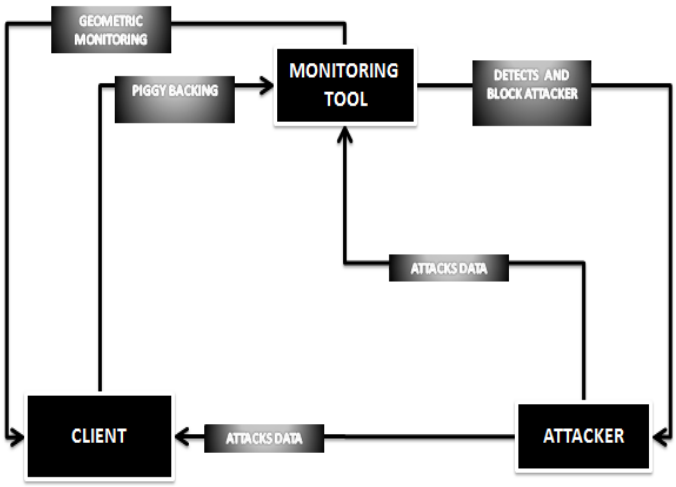
3. PIGGYBACKING

Piggybacking is an efficient tool for communication. It reduces communication overhead. It is mainly used to reduce the traffic in the network. It sends back the needed information along with the acknowledgement. So that the unnecessary communication will be reduced between the server and the client. With this piggbacking technique we can effectively monitor all the clients in the network.



4. ATTACK DETECTION

Attack will be detected with the help of these piggybacking details. These details will be used by the monitoring tool and analyse for possible attacks. Once the attack is detected then the alert will flash according to the intensity of the attack.



When the attack is done externally the above method easily identifies the attacker. But the possibility of Node Replication attack can take place. To recover from the Node Replication attack we are assigning unique name to each node in our network and monitor whether one group member is trying to access confidential data's from other group member, if so we will block the attacker from accessing the data. During Node Replication attack to identify which is the true node, we will send a Tracking code to the nodes and here the two nodes will get clash which helps the server to identify the wrong node and disconnects it from the network.

6 CONCLUSION

Here we conclude that the Hibernation method will efficiently secure the database with object security. The Piggybacking method is used for reducing communication overhead than the prior work. The monitoring server monitors the attack more accurately than the existing system. Hence our Experiments yield an improvement of over an order of magnitude in communication and data security relative to HGM.

REFERENCES

[1] D.J.Abadi et al., "The design of the borealis stream processing engine," in Proc.CIDR, Asilomar, CA, USA, 2005.

[2] I.Sharfman, A.Schuster and D.Keren,"A geometric approach to monitoring threshold functions over distributed data streams," ACM Trans.Database Syst., vol. 32, no. 4, Article 23, Nov. 2007.

[3] S.Burdakis and A.Deligiannakis,"Detecting outliers in sensor networks using the geometric approach," in Proc. ICDE, Washington, DC, USA, 2012, pp. 1108–1119.

[4] N.Giatrakos, A.Deligiannakis, M.N.Garofalakis, I. Sharfman and A.Schuster, "Prediction-based geometric monitoring over distributed data streams," in Proc. SIGMOD, Scottsdale, AZ, USA, 2012.

[5] D.Keren, I.Sharfman, A.Schuster and A.Livne, "Shape sensitive geometric monitoring," IEEE Trans. Knowl.Data Eng., vol. 24, no.8, pp. 1520–1535, Aug 2012.

[6] G.Sagy, D.Keren, I.Sharfman and A.Schuster, "Distributed threshold querying of general functions by a difference of monotonic representation," Proc. VLDB, vol.4, no.2, pp. 46–57, Nov.2010.

[7] G.Cormode and M.N.Garofalakis, "Sketching streams through the net: Distributed approximate query tracking," in Proc. VLDB,Trondheim, Norway, 2005, pp. 13–24.

[8] K.Yi and Q.Zhang, "Optimal tracking of distributed heavy hitters and quantiles," in Proc. PODS, Providence, RI, USA, 2009.

[9] A.Bar-Or, D.Keren, A.Schuster and R.Wolff, "Hierarchical decision tree induction in distributed genomic databases," IEEE Trans. Knowl. Data Eng., vol. 17, no. 8, pp. 1138–1151, Aug. 2005.

[10] A.Arasu and G.S.Manku, "Approximate counts and quantiles over sliding windows," in Proc. PODS, Paris, France, 2004, pp. 286–296.

[11] G.Cormode, S.Muthukrishnan, K.Yi and Q. Zhang, "Optimal sampling from distributed streams," in Proc. PODS, Indianapolis, USA, 2010.

[12] F.Li, K.Yi and J.Jest es, "Ranking distributed probabilistic data," in Proc. SIGMOD, Providence, RI, USA, 2009.

AUTHORS PROFILE:

Prakash K has received his B.E. (CSE) degree in the year 2013. At present he is pursuing M.E. (CSE) in Krishnasamy College of Engineering and Technology, Cuddalore, TamilNadu, India. His research interests lies in the areas of Knowledge and Data Engineering, Data Mining, Network Security and Distributed Computing.

Mr.G.Balakrishnan completed his B.Sc. (Computer science) in St.Joseph's College of Arts & Science, Cuddalore, M.C.A. in Annamalai University, Chidambaram, M.Phil (Computer Science) in Bharathidasan University, Tiruchirappalli. Currently he is working as HOD/ Associate professor in Information Technology in Krishnasamy College of Engineering & Technology, Cuddalore, Tamil Nadu, India. He has published more than 5 research papers in National/International conferences. Also he is a life member of Indian Society for Technical Education (ISTE) and member in Computer Society of India and IAENG (International Association of Engineers). His research interest lies in the areas of Network Security, Data Mining, Knowledge and Data Engineering and Cloud Computing. He attended many workshops & National seminars in various technologies and also attended Faculty Development Programme.