Effective RFID Data Indexing in Supply Chain Management using R+ Tree Approach.

Maria Anu V.[1], G. S. Anandha Mala[2],

Faculty of Computing, Sathyabama University, Chennai, India
Department of Computer Science, Easwari Engineering College, Chennai, India
mariaanu18@gmail.com.

Abstract:—

RFID is an emerging technology and effectively applied to different sectors like retail warehouses, manufacturing, supply chain management, health care applications and wall-mart. Specifically in the area of supply chain management. RFID techniques are very useful. On the other hand, the quantity of RFID data in such an environment is very large. Thus more computation and time is needed to take out valuable information from RFID data storage for the process of object tracking in the supply chain management. Therefore, this research work focuses on the indexing of the huge amount of RFID tag data query retrieval. Here, the R+ tree indexing technique is introduced to prompt the RFID data retrieval from the RFID data warehouse. This technique consists of the root node, intermediate and leaf node. Each leaf node is formed in the form of Minimum Bounding Rectangle (MBR) that includes data object and its value. Also leaf node stores the minimum bounding rectangle (MBR) and doesn't store the actual RFID data object. At last, our proposed indexing technique is conducted with various experiments to prove their effective and efficient.

Keywords: Data Storage Management; Indexing; Query Processing; R+ Tree; RFID Technology.

I. INTRODUCTION

Emerging RFID technology offers a lot of benefits over the conventional object tracking and it is adopted and deployed in real time applications widely. Conventional object tracking using WSN involves sensors [1]. But RFID technology provides a cheap and simple method for object tracking. This application also produces the huge volume of streaming data and these data is processed with the three steps. A first step to filter the streaming data automatically, next step is to process the data and the final step is to transform into semantic data. Finally, RFID data's are integrated into business applications.

In the RFID storage management, many multidimensional indexing methods were proposed but very few of the methods are adapted by the main database retailers because of their cost of integration and complexity. Many of the techniques need unsupported access to block the manager so they cannot be readily constructed. Other techniques are possible as external indexes that do not take over the industrial recovery and concurrency of the RFID database management system.

The most commonly proposed multidimensional indexes need the kernel unsupported augmentation and it is not readily available to maintain existing RFID applications in business database systems. Recently bitmap indexes, inbuilt the B-tree methods, one dimensional transformation methods and approximation methods are used to access and support the multidimensional vector data in business database system, Mostly the bitmap indexes methods have been used to handle the RFID data [2]. Here, the space necessities are condensed with the help of the bitmap data type that efficiently distinguish a set of identifiers. Though, these techniques may not fit either when the data in the same cluster are not constant or in applications that do not provide themselves to cluster based on a usual property [3].

One dimensional transformation methods use the single dimensional index structures. For example B-tree used to index the data depend on the scalar key. This technique uses the two stage query filter and it is like spatial querying to estimate the results and false hits removed to create a final result set. Examples of One dimensional tree technique include the pyramid-tree technique [4], i-distance [7], iMinMax [5], the p+ tree [6].

These methods are usually not a goal, purpose and which are lossy in nature. Thus, they rely more greatly on the exact filter than bounded regions in the feature space. Their throughput has an effect on object tracking. Where false hits take place due to objects being far away in the original space, but near in the transformed space. In case of i-distance methods there may be many objects are near close to the reference point for distance information. But it is not likely to close together and many objects require add during the exact filter.

Thus, a novel approach known as modified R+ is introduced in this proposed work in order to overcome the object tracking issues in RFID technology,. This approach has multidimensional indexing structure used to track the moving object and it has MBR. This indexing technology mostly used for the purpose of indexing and query processing over RFID streaming data. In the result section, the performance comparison is evaluated between the proposed indexing technology and R+ tree. From the result point of view, It is observed that the proposed indexing scheme process the queries and index RFID data effectively and provides better performance than the existing R+ tree.

II. RELATED WORK

A lot of research work is done on the RFID data storage management for increasing the performance of the RFID data retrieval. A few techniques are discussed briefly in this subsection. A novel warehousing model is proposed to process the path selection query and for the object transition, where tables are joined many times to obtain the collective parameters on the path. Hence, these techniques utilize the compression to decrease the join cost, but this existing technique is not applicable in case of products not move jointly in huge groups.

When objects move from one location to another location, objects' positions were reported periodically. The database which stores the histories of the spatiotemporal coordinates object of each moving object is called as Moving object database. This database must sustain queries of constantly moving objects for retrieving the location and trajectory. There is many indexing scheme approaches such as HR-tree [11], MV3R-tree [12] and 3D R-tree [10] for efficient query processing about the historical positions of the moving object. All these indexing schemes are known as the locality based indexing schemes, which are uses the R-tree technique [14]. The R-tree structure is a hierarchy of probably overlapping index regions which encircle the child nodes in Minimum Bounding Box. This MBB consists of all child node entries in n-dimensional domain due to its hierarchical structure depend on position. These methods illustrate better performance for time scale and distance range queries. However, the position property guides to traversal of a huge number of index paths in the trajectory queries. These trajectory queries are retrieved using the trajectory indexing preserving scheme in order to enhance the query performance. Which includes an indexing scheme like OP-tree [13] and TB-tree [17], all of these indexing schemes uses the property known as stores the only one path per leaf node. This property helps these indexing schemes to better output location based indexing scheme for efficient query processing Trajectory queries.

One more main property of trajectory preservation generates the dead space among the MBBs of non-leaf nodes. Even though, overlap among the node is large. This outperforms in poorer quality performance in distance range queries than the locality based indexing schemes.

RFID tag moves in the similar way as moving objects, thus the existing indexing schemes can be used to build for tracing tag trajectories. On the other hand, these techniques locate the RFID tags which enter into automobile industry, but do not leave a reader's position. To overcome this issue, data model technique discussed in [15] corresponds to tag route as time parameterized time dependent intervals; it is probable to recover a tag that remains at a reader, irrespective of the indexing method. Also the [15] introduced the one more index scheme called Time Parameterized Interval R-tree (TPIR) using the concept of R-tree [16] in order to retrieve and store RFID tag path effectively. By split and insert algorithms for the time-parameterized period, this technique had a develop query performance when comparing with the existing index scheme for moving objects.

An RFID system presented in [8], it is designed as a high-level of a large scale. Specific applications are utilized to compress and store RFID data into RFID databases. In these databases, data purification is integrated with query processing, high level information is recovered from partial, highly improvised data compression techniques are available, prior statistical knowledge and noisy data by exploiting known constraints. Person and object tracking concentrate on the moving objects in the case of association among the object identities and observed features is uncertain.

However, RFID system the object identity is given as a measurement of the readings. The main motive of the system is to transfer high volumes of noisy readings into accurate location events.

VG-curve introduced to multidimensional dynamic clustering primary index technique that can efficiently access and control RFID data [9]. The proposed idea is not responsive to the order or number of proportions of dimensions limited in the query and is simply created within present commercial database management system and it is able to process the huge volume of multidimensional RFID data. Storing the RFID data in multidimensional space permit processing of a wide range of queries. On the other hand, this method doesn't provide full RFID data recovery performance when increases the size of multidimensionality.

III. PROPOSED WORK

This section is briefly exploring the concept of RFID data indexing method in supply chain management.

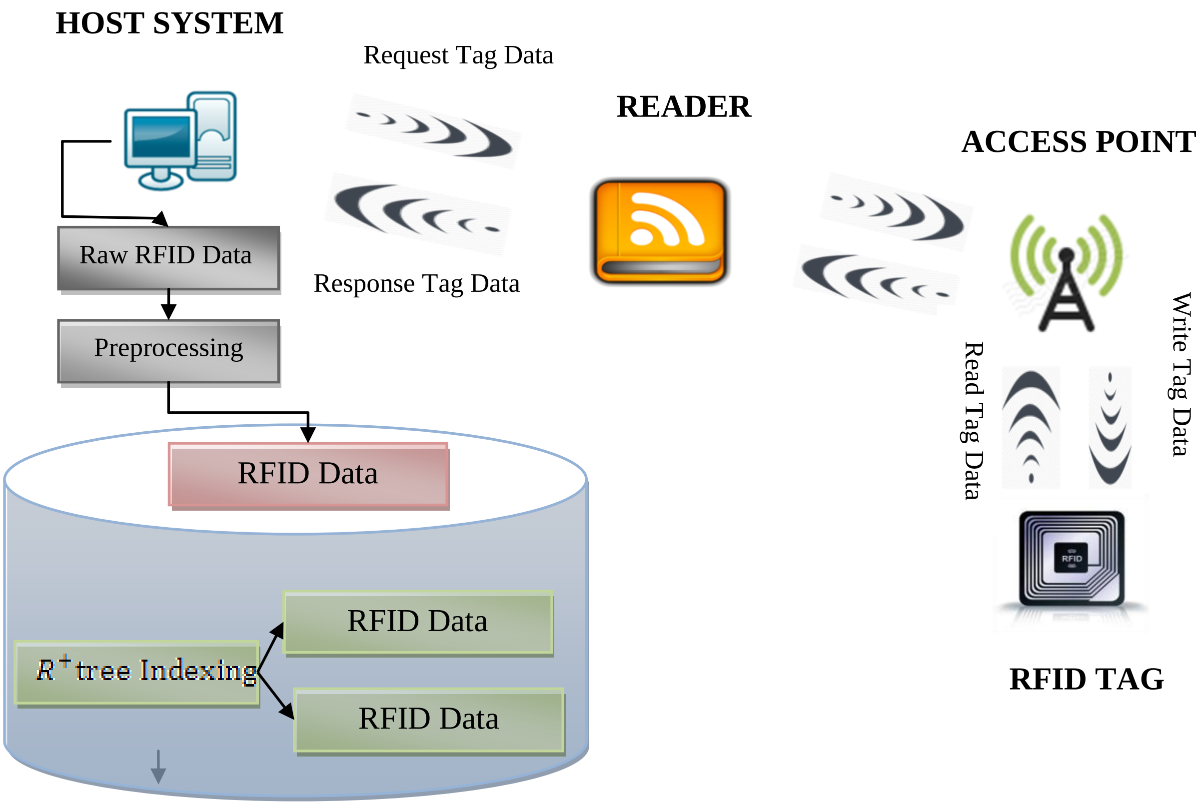

Fig 1. Proposed RFID Data Indexing Architecture

A. *Overview of Proposed Architecture*

The illustrated above Fig.1 shows the architecture diagram of the proposed system. Initially, the host system is frequently sending the RFID tag request to RFID tag reader device. The tag reader device is getting the raw RFID tag from the moving objects. Then, the tag reader forwards the received tag to the host server. After that, the received raw RFID data is pre-processed to remove unnecessary information from the RFID tag. Finally, the pre-processed RFID tag will match with stored RFID data to update the location or tracing the moving object location.

The proposed model also has two main data structures; the first is the R+ tree indexing implementation and the second one is query search processing. They are merged to facilitate the search process to the user and help him to get the full information of RFID tag data. As an object tracking and monitoring applications distributed into many geographically separate sites and millions of objects, the sheer volume of data poses a scalability challenge. A centralized approach requires all data to be transferred to a single location for processing. This approach incurs both delay of answering queries and high communication costs. In this work, we propose a novel index approach for object tracking and monitoring, which performs queries like where an object (and data) is located? using R+ tree technique. Thus performs inference and query processing on local RFID streams as objects are observed. An object event stream is produced from raw RFID data stream, describing the location and container of each object. Query processing runs continuously on the object event stream and other sensor streams to return all answers.

Input: the RFID tag id of a moving object **O**
Output: the indexing information of object **O**
1: **if O** is stored locally
2: return its indexing information
3: i ← 1
4: p =**O**. id. sub(1, Lp − i)
5: **while true**
6: Res= **O** at node (hash (p))
7: **if** Res is not nil return Res
8: **else**
9: i++, p =**O**. id. Sub (1, Lp + i)

Fig. 2: Proposed R^+ tree Algorithm for RFID Data Storage

The fig.2 shows our R+ tree indexing steps where initially the RFID tag ID is considered for indexing the RFID data. The input RFID data are checked initially if it is present in the local maximum bound rectangle then return the indexing results. Otherwise, compute the hash value of indexing RFID data until it finds it corresponding RFID tag from the data warehouse.

The following system configuration is used to implement our proposed indexing method: Intel corei3 processor, Clock Speed-2.53 GHz, RAM-4GB, Hard Disk-320GB. MySQL server 2008 is also used to store and processing the RFID tag retrieval queries on the RFID tag Data. To apply the indexing and querying, proposed method used the R+ tree that is widely used for the multi-dimensional space; And also, the performance of the R+-tree is compared with the original B-tree. The R+ tree a query index for the aggregate transformed space (RID, TIDcell, CID), The R+ -tree is implemented as the query index for four-dimensional space (RID, manager, product, serial), and CQI and VCR indexes are the existing query indexes in two-dimensional space (RID, TID). The proposed method applied approximations to the data of RFID indexes because it is impossible to store all the segments of query data. All indexes are kept in the main memory to support real time processing.

IV.RESULTS AND DISCUSSIONS

In this section presents experimental results of the proposed indexing method and query processing. The efficiency of our proposed model and methods can be illustrated by the average response time and average maintenance cost per query result. The maintenance cost means the cost to maintain and update our proposed data structure because of incoming RFID readings which can be incurred without any query. The response time can be evaluated by the query processing cost and the maintenance cost at the time point when a query is triggered.

Scalability: The proposed method further test the scalability of our inference system by using larger numbers of objects in the simulation with above mentioned system configuration. Our RFID data processing system can scale to 100,000 items per warehouse while keeping up with stream speed, totaling 1 million objects over 10 databases. The above reported results for accuracy and communication costs stay true. One way to support more objects is to use mobile readers for scanning objects on shelves (which is a more cost-effective deployment than static readers). Additionally, the proposed method uses an RFID reader to scan each isle of 90 shelves. The RFID reader reads every second and spends 10 seconds scanning each shelf. Given such reduced shelf readings, our proposed system can scale to 1.21 million items per warehouse while running at stream speed, totaling 12.1 million objects over 10 warehouses.

Fig.3(a) shows the indexing rates of the two techniques with respect to the number of readings in the index. As expected, serialization has the highest throughput, and B tree has the lowest. Fig.3(b) shows the query throughput of the two techniques with respect to the number of readings in the index.

The indexing cost, measured by the total volume of queries and their results transferred over the network, is considered to study the feasibility of the proposed R+ tree indexing algorithm and related data structures. Initially, the proposed method concentrated on the indexing cost to verify that the architecture is scalable. In this test, continuously added new nodes into the network until the size of the network reached 512. When a new node joined the network, a specific number of objects were generated on the node and the indexing process started. After a random period of time, the new node moved 10% of the objects to a randomly chosen destination. The proposed system collected the total volume of messages transferred during the whole process. It also ran the test 10 times and for the ith ($1 \leq i \leq 10$) time, the number of objects generated at each node is 500 i. As we can see from Fig.3(a), the indexing cost of the R+ tree indexing algorithm increases much slower than the individual indexing algorithm. This section also studied the performance of the indexing algorithms under different network sizes. In the experiment, the number of objects generated at each node is fixed to 5000 and the network size varies from 64, 128, 256, to 512. The result in Fig.3(b) also shows that with the increase in network size, the indexing cost of the R+ tree indexing algorithm increases while the individual indexing algorithm shows a horizontal line (due to the fixed data volume). The reason is that when the network size increases, Lp increases as well, which leads to an increase in the number of groups (2Lp) and the messages to be transferred. Generally, the performance degrades when the ratio of data volume to network size becomes smaller. As we can see, when the size of the network increases, the indexing cost of the R+ tree indexing algorithm becomes closer to that for the individual indexing algorithm. However, in reality, particularly large-scale applications, there are many more objects than nodes. These applications can take advantage of the benefits brought by our proposed R+ tree indexing approach.

V. CONCLUSION

This research work introduces novel RFID tag based R+ tree indexing technique and query processing to the RFID tag tracking applications. The proposed R+ tree indexing technique is utilized and considering their features of the RFIDs stream data that has been working with previous RFID tag tracking systems. The proposed query processing techniques minimize the computation state transferred across RFID tag warehouses while approximating the accuracy of centralized processing. The conducted experimental results illustrated the accuracy, efficiency, and scalability of the proposed indexing technique. In future work, our work extends to probabilistic query processing, exploit on-board tag memory to hold object state and enable anytime, anywhere querying, and explore smoothing over object relations for other data cleaning problems. The experimental results show the performance improvement of the proposed technique, compared with existing query indexes.

Table 1. Number of RFID Tags Retrieved from R+ tree Index Database

S.No	Number of requested Readings	Retrieved Query Results
1	122000	121000
2	250000	249000
3	300000	299078
4	500000	490450
5	700000	699800

Table 2. Number of RFID tag Reading Vs Query Processing Time

S.No	Number of Requested Readings	Query Processing Time(ms)
1	122000	330013
2	250000	660025
3	300000	740043
4	500000	1100100
5	700000	1501200

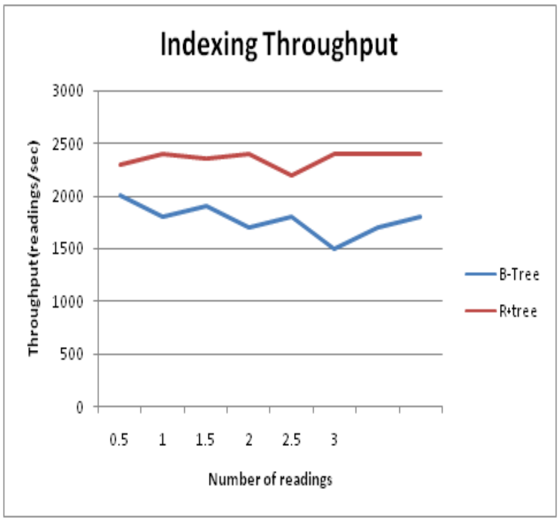

Figure 3 a). Indexing Throughput

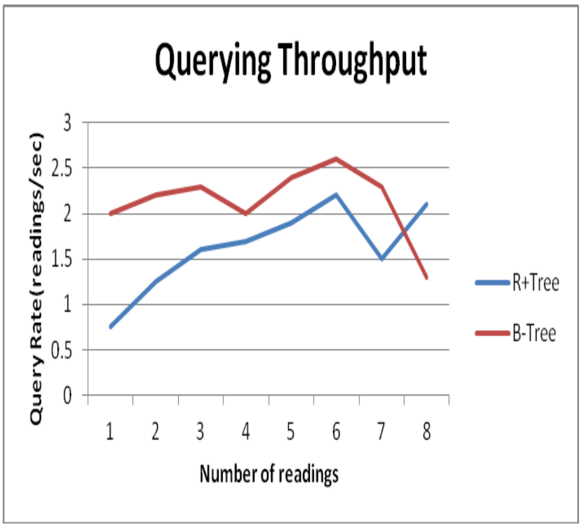

Figure 3 b). Querying Rate VsThroughput

ACKNOWLEDGMENT

We would like to express our thanks to Sathyabama University for encouraging our work. We would like to thank the anonymous commentators for their valuable comments.

REFERENCES

[1] R. Rathna, A. Sivasubramanian, "Energy Conservation in Radiation Monitoring" in the "Journal of Engineering (JER)", Kuwait University, Vol.2, No.2, 2014. Pages. 123-138.

[2] Hu, Y., Sundara, S., Chorma, T. & Srinivasan, J. (2005), Supporting RFID-based Item Tracking Applications in Oracle DBMS Using a Bitmap Datatype, in 'Proceedings of Very Large Data Bases VLDB', pp. 1140–1151.

[3] Lin, D., Elmongui, H. G., Bertino, E. & Ooi, B. C. (2007), Data management, RFID applications, in 'DEXA', pp. 434–444.

[4] Berchtold, S., Bohm, C. & Kriegel, H.-P. (1998), 'The Pyramid-Tree: Breaking the Curse of Dimensionality', Proceedings ACM SIGMOD pp. 142–153.

[5] Ooi, B. C., Tan, K.-L., Yu, C. & Bressan, S. (2000), Indexing the Edges - A Simple and Yet Efficient Approach to High-Dimensional Indexing, in 'Symposium on Principles of Database Systems',pp. 166–174.

[6] Zhang, R., Ooi, B. C. & Tan, K.-L. (2004), Making the Pyramid Technique Robust to Query Types and Workloads, in 'ICDE '04: Proceedings of the 20th International Conference on Data Engineering'. Proceedings of the Twenty-Third Australasian Database Conference (ADC 2012), Melbourne, Australia 125

[7] Jagadish, H. V., Ooi, B. C., Tan, K.-L., Yu, C. & Zhang, R. (2005), 'iDistance: An Adaptive B+-tree Based Indexing Method for Nearest NeighborSearch', ACM Trans. Database Syst. 30(2), 364–397.

[8] Thanh Tran, Univ. of Massachusetts,Amherst, MA. Sutton, .; Cocci, R.; Yanming Nie: Yanlei Daio : Shenoy, P., "Probabilistic Inference over RFID Streams in Mobile Environments", IEEE 25th International Conference on Date of Conference: March 29 2009-April 2 2009.

[9] Justin Terry Bela Stantic Abdul Sattar Institute for Integrated and Intelligent Systems (IIIS) Griffith University, Gold Coast, Queensland, Australia," Indexing RFID data using the VG-curve" Proceedings of the Twenty-Third Australasian Database Conference (ADC 2012), Melbourne, Australia.

[10] Y. Theodoridis, M. Vazirgiannis, and T. Sellis, "Spatio-temporal indexing for large multimedia applications," in Proceedings of International Conference on Multimedia Computing and Systems, 1996, pp. 441-448.

[11] T. Yufei and D. Papadias, "MV3R-Tree: A spatio-temporal access method for timestamp and interval queries," in Proceedings of the 27th International Conference on Very Large Data Bases, 2001, pp. 431-440.

[12] M. A. Nascimento and J. R. O. Silva, "Towards historical R-trees," in Proceedings of the ACM Symposium on Applied Computing, 1998, pp. 235-240

[13] H. Zhu, J. Su, and O. H. Ibarra, "Trajectory queries and octagons in moving object databases," in Proceedings of the 11th International Conference on Information and Knowledge Management, 2002, pp. 413-421.

[14] Guttman, "R-trees: A dynamic index structure for spatial searching," in Proceedings of ACM SIGMOD International Conference on Management of Data, 1984, pp.47-54.

[15] H. Ban, B. H. Hong, and D. H. Kim, "Time parameterized interval R-tree for tracing tags in RFID systems," in Proceedings of the 16th International Conference on Database and Expert Systems Applications, Vol. 3588, 2005, pp. 503-513.

[16] N. Beckmann and H. P. Kriegel, "The R*-tree: An efficient and robust access method for points and rectangles," in Proceedings ACM SIGMOD International Conference on Management of Data, 1990, pp. 322-331.

[17] D. Pfoser, C. S. Jensen, and Y. Theodoridis, "Novel approaches to the indexing of moving object trajectories," in Proceedings of the 26th International Conference on Very Large Data Bases, 2000, pp. 395-406.

[18] Timos K. Sellis, Nick Roussopoulos , Christos Faloutsos, "The R+-Tree: A Dynamic Index for Multi-Dimensional Objects", Proceedings of the 13th International Conference on Very Large Data Bases, p.507-518, September 01-04, 1987.