RFID-Miner: Warehousing and Mining Massive RFID Data Sets*

(SIGMOD'06 Demo Proposal)

Hector Gonzalez Jiawei Han Xiaolei Li Diego Klabjan
University of Illinois at Urbana-Champaign & Massachusetts Institute of Technology
{hagonzal, hanj, xli10}@uiuc.edu, klabjan@mit.edu

Abstract

Radio Frequency Identification (RFID) applications will play an essential role in object tracking and supply chain management systems. In the near future, it is expected that every major retailer will use RFID systems to track the product movements from suppliers to warehouses, store shelves, and eventually to points of sale. The volume of information generated by such systems can be enormous as each individual object (a pallet, a case, or an individual item) will leave a trail of data as it moves through different locations. The movement trails of such RFID data form a gigantic repository that contains rich information about the characteristics, changes, trends, and outliers of commodity flows.

In this demo, we will present an RFID-Miner system that constructs RFID data warehouses and mines knowledge from such data in the following aspects: (1) RFID data from multiple sites are integrated and cleansed to reduce data redundancy and errors; (2) RFID data warehouses are constructed, with data compressed by taking advantage of the fact that objects usually move together in large groups through early stages in the system (e.g., distribution centers) and only in later stages (e.g., stores) do they move in smaller groups; (3) RFID object flowgraphs are constructed to characterize the general trend of object movements; and (4) based on such general flowgraphs, data mining is performed in RFID data warehouses to discover trends, changes, and outliers in commodity data flow. Beside using synthetically generated data sets to demonstrate the efficiency and scalability of the system, some real RFID repository data sets from a retail chain will be used to show its usefulness in our demonstration.

* The work was supported in part by the U.S. National Science Foundation NSF IIS-02-09199/IIS-03-08215. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

1 Introduction

Radio Frequency Identification (RFID) is a technology that allows a transponder (RFID reader) to read, from a distance and without line of sight, a unique identifier that is provided (via a radio signal) by an "inexpensive" tag attached to an item [4, 10, 3, 5]. RFID offers a possible alternative to bar code identification systems and it facilitates applications like item tracking and inventory management in the supply chain. The technology holds the promise to streamline supply chain management, facilitate routing and distribution of products, and reduce costs by improving efficiency.

Large retailers like WalMart, Target, and Albertsons have already begun implementing RFID systems in their warehouses and distribution centers, and are requiring their suppliers to tag products at the pallet or case levels. Individual tag prices are expected to fall from around 25 cents per unit to 5 cents per unit by 2007. At that price level, we can expect tags to be placed at the individual item level for many products. The main challenge then becomes how to handle, interpret, and analyze the enormous volume of data that an RFID application will generate.

Example 1. Suppose a retailer with 3,000 stores sells 10,000 items a day per store. Assume that we record each item movement with a tuple of the form: $(EPC, location, time)$, where EPC[1] is an Electronic Product Code which uniquely identifies each item. If each item leaves only 10 traces before leaving the store by going through different locations, this application will generate at least 300 million tuples per day. A manager may ask queries on the duration of paths like (Q_1): "*List the average shelf life of dairy products in 2005 by manufacturer*", or on the structure of the paths like (Q_2): "*What is the average time that it took coffeemakers to move from the warehouse to the shelf and*

[1] We will use the terms EPC and RFID tag interchangeably.

finally to the point of sale in January of 2006?" or some general trend and outliers like (Q_3): "*Find the products whose turn-around time is unusually short in comparison with its past history and with other similar products in December 2005.*" ∎

Such enormous amount of low-level data and flexible high-level queries pose great challenges to traditional relational and data warehouse technologies since the processing may involve retrieval and reasoning over a large number of inter-related tuples through different stages of object movements. No matter how the objects are sorted and clustered, it is difficult to support various kinds of high-level queries and data mining requests in a uniform and efficient way. A nontrivial number of queries may even require a full scan of the entire RFID database.

In our recent study [6], we have proposed a new RFID data warehouse model [7, 8, 11, 9] to integrate, cleanse, compress and aggregate RFID data in an organized way such that a wide range of queries can be answered efficiently. Moreover, in another recent study, we have worked out an effective flowgraph analysis model so that RFID data flow can be modeled and analyzed systematically. Such RFID data warehouse and data analysis methods set up a solid foundation for the RFID-Miner system that is being implemented and will be demonstrated in the conference.

The rest of the paper is organized as follows. Section 2 presents the architecture of the RFID-Miner system. Section 3 introduces the general principles in the design and implementation of the RFID-Miner system. Finally, Section 4 describes our demonstration plan.

2 Architecture of RFID-Miner

The architecture of RFID-Miner is shown in Fig. 1. Data in several RFID data repositories at multiple sites are integrated to form an RFID data warehouse, where data cleansing, redundancy removal, data integration, and specialized RFID data consolidation and aggregation are performed in order to form a consistent, concise and OLAP-efficient data warehouse [6]. Then data mining is performed on the RFID data stored in the warehouse to extract general data flowgraphs as well as exceptions which are represented as a set of exceptional data flow subgraphs, fragments, and rules. Such extracted rules are stored in the RFID data flowgraph/exception repository. A user-friendly interface is then used to interact with the RFID system users and data analyzers. The system takes users' queries or data mining task requests, which are posed either to data flowgraph repository or data warehouse, based on the nature of the query. The mining/query results will then be returned to the user.

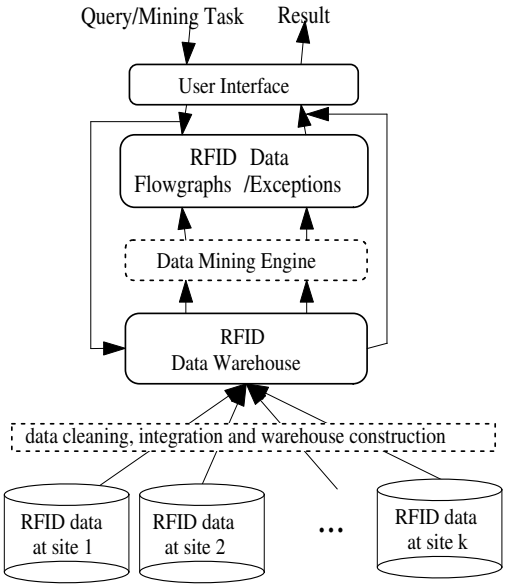

Figure 1. The Architecture of RFID-Miner

3 Principles in the design and development of RFID-Miner

The design of RFID-Miner consists of the following major stages: (1) data cleaning, (2) warehouse construction, (3) data mining, and (4) retrieval/OLAP/mining query processing.

3.1 Data cleaning: Redundancy elimination and noise reduction

There exists much redundancy in real-world RFID data. The first task is redundancy removal. Each reader provides tuples of the form $(EPC, location, time)$ at fixed time intervals. When an item stays at the same location, for a period of time, multiple tuples will be generated. These tuples should be grouped into a single one of the form $(EPC, location, time\_in, time\_out)$. For example, if a supermarket has readers on each shelf, so called smart shelves, that scan the items every minute, and items stay on the shelf on average for one day, we get a 1,440 to 1 reduction in size without loss of information. Similarly, data at higher levels of abstraction will be merged and collapsed as well. In the meantime, some missing records can be filled based on the principle that many items move together, especially if they re-appear in later stages.

3.2 RFID data warehouse construction

Our RFID warehouse consists of (1) a fact table, *stay*, composed of cleansed RFID records; (2) an information table, *info*, that stores path-independent information for each item, i.e., SKU information that is constant regardless of the location of the item such as manufacturer, lot number, color, etc.; and (3) a *map* table that links together different records in the fact table that form a path. Figure 2 shows a logical view into the RFID warehouse schema. We call the *stay*, *info*, and *map* tables aggregated at a given abstraction level an *RFID-Cuboid*.

A major difference between the RFID warehouse and a traditional warehouse is the presence of the map table linking records from the fact table (*stay*) in order to preserve the original structure of the data.

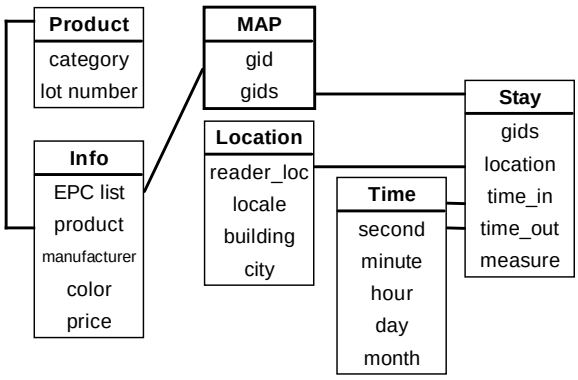

Figure 2. RFID Warehouse: Logical Schema

Several general ideas are explored for constructing a highly compact RFID data warehouse.

1. Since a large number of items travel and stay together through several stages, it is important to represent such a collective movement by a single record no matter how many items were originally collected.
2. Since many users are only interested in data at a relatively high abstraction level, data compression can be explored to group, merge, and compress data records.
3. In many analysis tasks, certain path segments, e.g., non-essential object movements, can be ignored or merged for simplicity of analysis.

Each dimension in the *stay* and *info* tables has an associated concept hierarchy. Thus the data warehouse consists of a hierarchy of RFID-Cuboids. In order to provide fast response to queries specified at various levels of abstraction, it is important to pre-compute some *RFID-Cuboids* at different levels of the concept hierarchies for the dimensions of the *info* and *stay* tables [2, 8]. In our design, we suggest to compute a set of *RFID-Cuboids* at the minimal interesting level at which users will be interested in inquiring the database, and a small set of higher level structures that are frequently requested and that can be used to quickly compute non-materialized *RFID-Cuboids*.

In the construction of the RFID Warehouse, an efficient method has been developed to identify path segments that can be collapsed by simply merging parent/child nodes that correspond to the same location.

3.3 Mining flowgraphs from RFID data warehouses

To study the RFID data flow and discover the trends and outliers of the object flow, it is beneficial to construct RFID flowgraphs that reflect the major probabilistic [2] flow of RFID objects. Figure 3 shows a fragment of a top-level RFID flowgraph extracted from an RFID data warehouse. It indicates that a producer $P1$ sends its products to distribution centers $A1$, $A2$, and $A3$ with the probability of 0.2, 0.6 and 0.15 respectively (and the remaining 0.05 could be distributed to other distribution centers but with the probability too small to be listed in this major top-level flowgraph. In turn, the product $P1$ at the distribution center $A1$ is distributed to stores $S1$, $S2$ and $S4$ with the probability of 0.2, 0.65 and 0.13, respectively. The remaining part of the graph can be interpreted similarly.

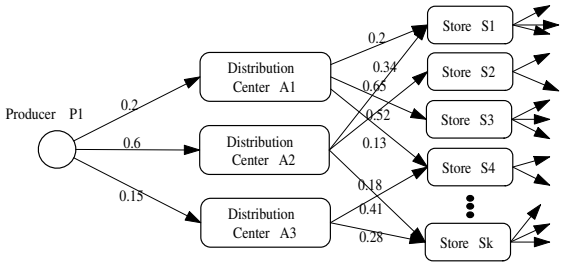

Figure 3. A fragment of an RFID flowgraph

Besides finding the general trends, it is important to associate with each node, edge, fragment, and subgraphs the corresponding exceptional flow with probability attached as well. In general, such association may include additional conditions such as time, duration, location, object category, and other spatiotemporal or categorical information. The general principle

[2] when objects move deterministically, probability should be interpreted the fraction of items taking the transition

is that if the behavior can be approximately derived (within certain range of expected values), there is no need to associate such "redundant information" with the higher-level flowgraph information, However, if the occurrence of the exception substantially deviates from the upper level fragment and is still over the minimal support threshold, such information should be stored in the corresponding "exceptional flow information repository" to facilitate trend and outlier analysis.

Efficient methods have been developed to extract such information from the RFID data warehouse with minimal scan of the corresponding cuboid of the data warehouse [1].

3.4 Query and Mining Processing

With the above "preprocessing", retrieval, OLAP, and mining queries can be processed on RFID data warehouses efficiently.

The retrieval and slice/dice operations can be implemented efficiently by using relational query execution and optimization techniques.

Path queries, which ask about information related to the structure of object traversal paths, are unique to the RFID warehouse since the concept of object movements is not modeled in traditional data warehouses. It is essential to allow users to inquire about an aggregate measure computed based on a predefined sequence of locations (path). One such example could be: "What is the average time for milk to go from farms to stores in Illinois?".

For mining queries directly related to certain nodes, edges and paths at multiple levels, one can take advantages of constructed data warehouse and map queries to the related cuboids at the corresponding levels of abstraction. For mining queries that need reasoning on the flowgraphs, such as finding anomaly of object movements with respect to certain particular category of products in a specified duration/region, one can take advantages of the precomputed general and exceptional object flowgraphs and present answers when available or further retrieve data based on the paths and regions guided by such flowgrpahs.

4 System demonstration plan

We plan to use two kinds of data in our demonstration. The first set is RFID data, obtained by means of a simulation where we generate RFID object movements, based on the study of the product flow of a big retail chain store. The second set is a subset of real RFID data, with confidential and private information removed, obtained from a retail chain company.

We will construct a Web-based, graphical user interface, with which users can input queries and get responses. The queries will be partitioned into retrieval queries, OLAP queries and mining queries. Mining currently is confined to trend, outlier, and flow-graph analysis. Other functions may be added time permitting. The output will be presented in different forms based on the query results, including pie-charts, histograms, flowgraphs, curves, and scatter plots.

References

[1] anonymous authors. Undisclosed title. In *submitted for evaluation*, Nov. 2005.

[2] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26:65–74, 1997.

[3] S. Chawathe, V. Krishnamurthy, S. Ramachandran, and S. Sarma. Managing RFID data. In *Proc. Intl. Conf. on Very Large Databases (VLDB'04)*.

[4] K. Finkenzeller. *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. John Wiley and Sons, 2003.

[5] C. Floerkemeier, D. Anarkat, T. Osinski, and M. Harrison. PML core specification 1.0. White paper, MIT Auto-ID Center, `http://www.epcglobalinc.org/standards_technology/Secure/v1.0/PML_Core_Specification_v1.0.pdf`, 2003.

[6] H. Gonzalez, J. Han, X. Li, and D. Klabjan. Warehousing and analysis of massive RFID data sets. In *Proc. 2006 Int. Conf. Data Engineering (ICDE'06)*, Atlanta, Georgia, April 2006.

[7] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. *Data Mining and Knowledge Discovery*, 1:29–54, 1997.

[8] V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. In *Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'96)*.

[9] X. Li, J. Han, and H. Gonzalez. High-dimensional OLAP: A minimal cubing approach. In *Proc. Intl. Conf. on Very Large Databases (VLDB'04)*, 2004.

[10] S. Sarma. Integrating RFID. *ACM Queue*, 2(7):50–57, October 2004.

[11] A. Shukla, P. Deshpande, and J. F. Naughton. Materialized view selection for multidimensional datasets. In *Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98)*.