Data & Knowledge Engineering 70 (2011) 1070–1087

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

An approximate duplicate elimination in RFID data streams

Chun-Hee Lee [a], Chin-Wan Chung [b,*]

[a] Data Analytics Group, SAIT, Samsung Electronics, Yongin, 446–712, Republic of Korea
[b] Department of Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, 305–701, Republic of Korea

ARTICLE INFO

Article history:
Received 6 March 2010
Received in revised form 16 July 2011
Accepted 18 July 2011
Available online 31 July 2011

Keywords:
Duplicate elimination
RFID
Bloom filter
Real-time DBs
Smart cards

ABSTRACT

The RFID technology has been applied to a wide range of areas since it does not require contact in detecting RFID tags. However, due to the multiple readings in many cases in detecting an RFID tag and the deployment of multiple readers, RFID data contains many duplicates. Since RFID data is generated in a streaming fashion, it is difficult to remove duplicates in one pass with limited memory. We propose one pass approximate methods based on Bloom Filters using a small amount of memory. We first devise Time Bloom Filters as a simple extension to Bloom Filters. We then propose Time Interval Bloom Filters to reduce errors. Time Interval Bloom Filters need more space than Time Bloom Filters. We propose a method to reduce space for Time Interval Bloom Filters. Since Time Bloom Filters and Time Interval Bloom Filters are based on Bloom Filters, they do not produce false negative errors. Experimental results show that our approaches can effectively remove duplicates in RFID data streams in one pass with a small amount of memory.

1. Introduction

Recently, due to the advancement of information technology, various kinds of data such as XML, RDF, and RFID data have been generated [18,9,5,8]. Especially, a large amount of RFID data has been generated in many environments since the RFID technology does not require contact in detecting RFID tags and therefore has been used in many areas such as business, military, and medical applications. The RFID adoption in Walmart is a typical RFID example in the business area.

However, the advantage of the RFID technology causes a new problem. Since an RFID tag is detected without contact, if an RFID tag is within a proper range from an RFID reader, the RFID tag will be detected whether we want to or not. Therefore, if RFID tags stay or move slowly in the detection region, much unnecessary data (i.e., duplicate RFID data) will be generated. On the other hand, if RFID tags move fast or many RFID tags move simultaneously in the detection region, one RFID reader may not be able to detect all of them. To prevent missing readings of RFID tags in such cases, several RFID readers are generally deployed in order to monitor a single location [1,2]. When several RFID readers detect one RFID tag at the same time, duplicate data is generated. Therefore, we cannot avoid the generation of duplicate RFID data in RFID applications.

An intelligent RFID reader with the processing capability can eliminate duplicate RFID data generated by the RFID reader. However, duplicate RFID data generated from multiple readers can not be removed with only the self-contained processing capability of RFID readers. Therefore, we need a technique to eliminate duplicate RFID data in the server (RFID middleware) that collects RFID data from various RFID readers.

Consider the example in Fig. 1. There are two RFID readers and two tags pass through the detection region. In this situation, Reader1 and Reader2 detect tags with the identifier ID1 and ID2. Each reader generates the detection information such as <tag ID, location of the reader, time>. Reader1 detects the tag with ID1 and generates RFID data <ID1, Loc1, 1>, <ID1, Loc1, 2>, <ID1, Loc1, 4>. However, the RFID data is duplicate data except <ID1, Loc1, 1>. Also, Reader1 generates RFID data <ID2, Loc1, 3> <ID2, Loc1, 5> for the tag with ID2 and <ID2, Loc1, 5> is duplicate data. In the same way, Reader2 generates RFID data and has duplicates as shown

* Corresponding author. Tel.: +82 42 350 3537; fax: +82 42 350 7737.
E-mail addresses: chunhee1.lee@samsung.com (C.-H. Lee), chungcw@kaist.edu (C.-W. Chung).

0169-023X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2011.07.007

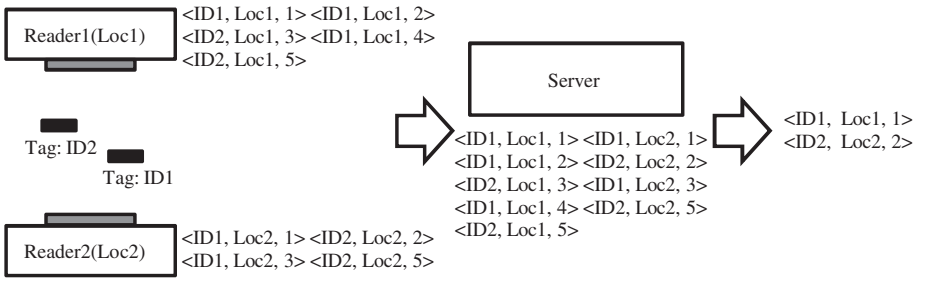

Fig. 1. An example for duplicate RFID data elimination.

in the lower part of Fig. 1. This is because a reader detects a tag continuously within the detection region. RFID data generated in each reader is sent to the server. Then, there are more duplicates in the server since two readers may detect the same tag. For example, <ID1, Loc1, 1> in Reader1 and <ID1, Loc2, 1> in Reader2 are duplicate. Therefore, the server removes duplicate RFID data as preprocessing and the result for preprocessing is <ID1, Loc1, 1> <ID2, Loc2, 2>. This problem seems simple for a small amount of RFID data.

However, the volume of RFID data is generally very big. Also, if a tag is attached to each item, the amount of data generated in a large retailer will exceed terabytes in a day [13]. And, RFID data is produced in a stream. It means that a duplicate elimination method should process data instantly with limited memory. It is difficult to design an exact duplicate elimination method. As an alternative, we can use approximate duplicate elimination techniques for RFID applications that do not require exact answers.

In such an application, we can consider a real-time analysis application for the movement of customers in a large department store. Each store in the department store has RFID readers. Each customer has a unique RFID tag. The manager wants the real-time analysis of the movement of customers such as the number of customers in each store and the store which has the maximum number of customers. For such an analysis, the central server in the department store should remove duplicates. In this environment, so much RFID data comes into the server simultaneously and there are duplicates. Especially, when a customer stays at the same location for a long time, it generates a large number of unnecessary duplicate data. In order to eliminate duplicates exactly, we need to keep all RFID data including duplicates in memory during a long period. Therefore, it is difficult to eliminate duplicates exactly in real-time using a small amount of memory relative to the amount of RFID data. In this application, the manager does not feel uncomfortable even if statistics with allowable errors are provided. Therefore, we propose approximate RFID duplicate elimination techniques in one pass with the limited memory.

Bloom Filters have been widely used as a very compact data structure with an allowable error. To manage RFID data streams with a small amount of memory, we devise Bloom Filter based approaches. However, since Bloom Filters are targeted for static data, we should adapt Bloom Filters to RFID data stream environments. We thus propose Time Bloom Filters as a straightforward adaptation of Bloom Filters. Since Time Bloom Filters are based on Bloom Filters, they do not generate false negatives which are duplicate data contained in the result after filtering. However, they may generate false positives that are non-duplicate data which are not contained in the result after filtering. We provide the false positive probability for Time Bloom Filters. Also, to reduce false positives for Time Bloom Filters, we devise Time Interval Bloom Filters using the concept of the interval.

Our contributions are as follows:

- *The Effective Duplicate Elimination Methods.* In data stream environments, it is not easy to design one pass duplicate elimination algorithm with limited memory. To design such an algorithm, we adapt Bloom Filters to RFID data stream environments. We propose effective approximate duplicate elimination methods, Time Bloom Filters and Time Interval Bloom Filters. They can eliminate duplicates in one pass with a small amount of memory.
- *Space Optimization for Time Interval Bloom Filters.* While the Time Bloom Filter stores one time field, the Time Interval Bloom Filter stores two time fields as a time interval. Therefore, the Time Interval Bloom Filter needs more space than the Time Bloom Filter. We devise a method to reduce space for the Time Interval Bloom Filter.
- *The Formulation of a Duplicate Elimination Problem.* Though many papers mention the duplicate elimination problem in RFID data, they do not formulate it rigorously. In this paper, we formulate the duplicate elimination problem in RFID data streams formally.
- *Parameter Setting.* In Time Bloom Filters and Time Interval Bloom Filters, the number of hash functions k affects errors. We propose a formula to find k and validate it using an experimental evaluation.

1.1. Organization

The rest of this paper is organized as follows. In Section 2, we present related work. In Section 3, we explain Bloom Filters as a preliminary, and in Section 4, we formalize the duplicate elimination problem in RFID data streams. We describe Time Bloom Filters and Time Interval Bloom Filters in Sections 5 and 6, respectively. In Section 7, we discuss parameter setting in Time Bloom Filters and Time Interval Bloom Filters. We measure the effectiveness of Time Bloom Filters and Time Interval Bloom Filters experimentally in Section 8. Finally, we conclude our work in Section 9.