Efficient Protocols for Collecting Histograms in Large-Scale RFID Systems

Lei Xie, Member, IEEE, Hao Han, Member, IEEE, Qun Li, Member, IEEE, Jie Wu, Fellow, IEEE, and Sanglu Lu, Member, IEEE

Abstract—Collecting histograms over RFID tags is an essential premise for effective aggregate queries and analysis in large-scale RFID-based applications. In this paper we consider an efficient collection of histograms from the massive number of RFID tags, without the need to read all tag data. In order to achieve time efficiency, we propose a novel, ensemble sampling-based method to simultaneously estimate the tag size for a number of categories. We first consider the problem of basic histogram collection, and propose an efficient algorithm based on the idea of ensemble sampling. We further consider the problems of advanced histogram collection, respectively, with an iceberg query and a top-k query. Efficient algorithms are proposed to tackle the above problems such that the qualified/unqualified categories can be quickly identified. This ensemble sampling-based framework is very flexible and compatible to current tag-counting estimators, which can be efficiently leveraged to estimate the tag size for each category. Experiment results indicate that our ensemble sampling-based solutions can achieve a much better performance than the basic estimation/identification schemes.

Index Terms—Algorithms, RFID, time efficiency, histogram

1 INTRODUCTION

With the rapid proliferation of RFID-based applications, RFID tags have been deployed into pervasive spaces in increasingly large numbers. In applications like warehouse monitoring, the items are attached with RFID tags, and are densely packed into boxes. As the maximum scanning range of a UHF RFID reader is usually 6-10 m, the overall number of tags within this three-dimensional space can be up to tens of thousands in a dense deployment scenario, as envisioned in [1], [2], [3]. Many tag identification protocols [4], [5], [6], [7], [8] are proposed to uniquely identify the tags one by one through anti-collision schemes. However, in a number of applications, only some useful statistical information is essential to be collected, such as the overall tag size [2], [9], [10], popular categories [11] and the histogram. In particular, histograms capture distribution statistics in a space-efficient fashion. In some applications, such as a grocery store or a shipping portal, items are categorized according to some specified metrics, such as types of merchandise, manufacturers, etc. A histogram is used to illustrate the number of items in each category.

In practice, tags are typically attached to objects belonging to different categories, e.g., different brands and models of clothes in a large clothing store, different titles of books in a book store, etc. Collecting histogram can be used to illustrate the tag population belonging to each category, and determine whether the number of tags in a category is above or below any desired threshold. By setting this threshold, it is easy to find popular merchandise and control stock, e.g., automatically signaling when more products need to be put on the shelf. Furthermore, the histogram can be used for approximate answering of aggregate queries [12], [13], as well as preprocessing and mining association rules in data mining [14]. Therefore, collecting histograms over RFID tags is an essential premise for effective queries and analysis in conventional RFID-based applications. Fig. 1 shows an example for collecting histogram over the RFID tags deployed in the application scenarios.

While dealing with a large scale deployment with thousands of tags, the traditional tag identification scheme is not suitable for histogram collection, since the scanning time is proportional to the number of tags, which can be in the order of several minutes. As the overall tag size grows, reading each tag one by one can be rather time-consuming, which is not scalable at all. As in most applications, the tags are frequently moving into and out of the effective scanning area. In order to capture the distribution statistics in time, it is essential to sacrifice some accuracy so that the main distribution can be obtained within a short time window—in the order of several seconds. Therefore, we seek to propose an estimation scheme to quickly count the tag sizes of each category while achieving the accuracy requirement.

In most cases, the tag sizes of various categories are subject to some skewed distribution with a “long tail”, such as the Gaussian distribution. The long tail represents a large number of categories, each of which occupies a rather small percentage among the total categories. While handling the massive number of tags, in the order of several thousands, the overall number of categories in the long tail could be in
scanning time of the SC strategy and the ES strategy grows slowly. The reason is as follows: as the standard deviation $\sigma$ increases, the number of qualified categories is increasing, thus more slots are essential to verify the categories for accuracy; besides, fewer categories have tag sizes close to the threshold, thus fewer slots are required to verify the population constraint. In all, the overall scanning time increases rather slowly.

We evaluate the performance of our PT-Top$k$ algorithm. In Fig. 6c, we compare the scanning time with various values of $k$. We observe that as $k$ increases from 20 to 120, the scanning time of the ES strategy increases from $1.5 \times 10^5$ to $2.5 \times 10^5$, and then decreases to $2 \times 10^5$. The reason is that, as the value of $k$ increases, the exact threshold is reduced, and more categories are identified as qualified, thus more slots are essential to verify the categories for accuracy. Then, as the value of $k$ further increases, more qualified categories with large tag sizes can be quickly wiped out in the threshold estimation, and thus fewer slots are required in the threshold estimation, and the overall scanning time is decreased. In Fig. 6d, we evaluate the convergence for estimating the threshold $t$. We set $m = 200, \mu = 500, \sigma = 200, k = 20$. We observe that the width of the range $[\hat{t}, \bar{t}]$, i.e., $y$, is continuously decreasing as the scanning time increases. When the scanning time reaches $1.8 \times 10^6$, the value of $y$ is below the required threshold in the dash line, then the iteration ends.

10 Conclusion
Collecting histograms over RFID tags is an essential premise for effective aggregate queries and analysis in large-scale RFID-based applications. We believe this is the first paper considering the problem of collecting histograms over RFID tags. Based on the ensemble sampling method, we respectively propose effective solutions for the basic histogram collection, iceberg query problem, and top-$k$ query problem. Simulation results show that our solution achieves a much better performance than others.

Acknowledgments
This work was supported in part by National Natural Science Foundation of China under Grant No. 61100196, 61472185, 61321491, 91218302, 61373129; Jiangsu Natural Science Foundation under Grant No. BK2011559; Key Project of Jiangsu Research Program under Grant No. BE2011316; EU FP7 IRSES MobileCloud Project under Grant No. 612212. The work of Qun Li was supported in part by US NSF Grants CNS-1117412, CNS-1320453, and CAREER Award CNS-0747108. The work of Jie Wu was supported in part by US NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167. Lei Xie is the corresponding author.

References