ARTICLE IN PRESS

Journal of Network and Computer Applications ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Load-balancing mechanism for the RFID middleware applications over grid networking

Yi-Wei Ma [a], Han-Chieh Chao [b,*], Jiann-Liang Chen [c], Cheng-Yen Wu [d]

[a] Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan
[b] Institute of Computer Science & Information Engineering, National Ilan University, Ilan, Taiwan
[c] Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan
[d] Department of Electronics Engineering, National Dong-Hwa University, Hualien, Taiwan

ARTICLE INFO

Article history:
Received 9 November 2009
Received in revised form
29 March 2010
Accepted 28 April 2010

Keywords:
RFID networks
Load-balancing mechanism
Grid networking
Overloading problem
Middleware

ABSTRACT

RFID middleware often has to process enormous amounts of data, possibly causes overloading and leading to data errors. Therefore, RFID middleware applications require load-balancing mechanisms. This work proposes a grid-based load-balancing mechanism for RFID middleware applications. The proposed mechanism incorporates functional modules *Buffer Management* and *Load Balancing Management* over a grid networking platform, to buffer the read data and share the middleware loading, thereby solving the overloading issues in RFID applications. Performance analysis results show that the proposed mechanism is more efficient than the existing mechanisms. The mechanism has an average improvement of 29.15% for the processing time and improvement of 88.75% for the packet lost ratio, while the number of middleware hosts is fixed. The proposed RFID load-balancing mechanism is indicated to be very reliable.

1. Introduction

Despite its development over the past half century, Radio Frequency IDentification (RFID) technology has recently attracted renewed interest from many industries including retail, pharmaceutical and health care (Auto-ID Labs, 2009). Industry observers have recognized RFID as a very important technology. RFID is a non-contact automatic identification technology that identifies specific targets and retrieves information of interest using radio signals (EPCglobal Inc., 2009). An RFID system comprises three major parts, namely a tag, a reader and a unique code called an electronic product code (EPC). The RFID tag is associated with the object to be identified. The RFID reader extracts the unique identifier of the object from the tag. And the unique code EPC is provided by the electronic product code global (EPCglobal^{TM}).

Although small, RFID systems have significantly changed in information tracking applications, and are adopted to trace objects and assets worldwide (Weinstein, 2005; Nath et al., 2006). Information applications industries can reduce the investment in management, and can enhance high-quality services by attaching smart tags to objects. An RFID network can track and identify every unique object in a network. The physical environment may have many objects with unique codes that need to be identified. Therefore, the RFID middleware may need to process huge amounts of data, causing it to become overloaded. Hence, RFID middleware applications need to consider the load-balancing problem. This study proposes a load-balancing mechanism based on grid networking for RFID middleware applications.

This investigation described EPC network and RFID middleware architecture, and presented relevant research on methods for load balancing using connection pool in the RFID middleware, the dynamic load-balancing approach on RFID middleware and grid-based job scheduling for RFID middleware applications. Specifically, this study presented a load-balancing mechanism designed to solve the middleware overload. The proposed mechanism incorporates the functional modules, buffer management and load balancing management, over a grid networking platform. The buffer management component is the first to connect with the reader in the middleware host. Meanwhile, load balancing management is the core component of the proposed load-balancing mechanism.

The rest of this work is organized as follows: Section 2 describes the related works on RFID architecture and load-balancing issues. Section 3 draws up a grid networking that could combine with the RFID middleware applications to solve the load-balancing problem. Section 4 presents the performance analysis of the proposed mechanism and compared with the other existing architectures described in Section 2. Conclusions and future works are finally drawn in Section 5.

* Corresponding author. Tel.: +886 277129009.
E-mail address: hcc@niu.edu.tw (H.-C. Chao).

1084-8045/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2010.04.010

2. Background knowledge

The EPCglobal™ is evolved from the Auto-ID Center. EPCglobal™ is developing free and universal standards to regulate the interactions among all the objects including object identification, data transfer and storage. The unique code called the EPC will help uniquely identify each object. The EPC will help the network to find objects by the RFID technology and take the information of the tagged object.

2.1. EPC network architecture

The EPC network architecture is derived from the Auto-ID Center. The EPC network is designed to share business information among trade partners over the Internet, enabling users to obtain any unique product information all over the world. The network comprises of many local EPC networks in various enterprises and industries. The EPC system has several important components. The reader needs to identify and read tag. The middleware processes all the information in tags, and manages the information required to provide services. The information is sent to the EPC information service (EPCIS). The EPCIS shares the information over the Internet. Finally, the object naming service (ONS) sends other tag requests to EPCIS.

The typical EPC network comprises of three main parts, namely middleware, EPCIS and ONS. The reader receives many code numbers from tags that it reads, and transfers these to the middleware, which processes the information and repeat codes (Want, 2006; Atlig et al., 2007; EPCglobal Inc., 2004).

An overall EPC network is composed of many local EPC networks. The EPCIS manages all local EPC networks, and shares all information from these networks with other EPC networks. If the request for information in the local EPC network fails, then the local EPCIS asks the ONS for more information through the Internet, allowing EPC networks to share information with each other via the Internet. The key design aim of the EPC network is to link all useful business information in the world. Linking multiple EPC networks allows information to be shared all over the world, enabling all users to find tag data via the EPC network.

A global network may have many EPCIS. An EPCIS records information about many tags. A node wishing to query tag data asks the EPCIS for information about the tag. However, the nearest EPCIS usually has no record of the tag. In this case, the EPCIS has to ask ONS for more relationship to find the appropriate EPCIS (Petra and Zdenek, 2007; EPCglobal Inc., 2005). If the ONS returns the appropriate EPCIS address to the node, then the node can ask the EPCIS for further information about the tag.

The RFID middleware is located between the reader and the application in an RFID system (Kourouthanassis and Roussos, 2003; Want, 2004). The RFID reader reads data from the tags, and transfers the data to the back-end applications. The back-end system records and interprets the object data to useful business information. The RFID middleware mainly transforms events into meaningful business information. Improvements in RFID reader technology have led to increasingly diverse readers being designed for different applications. Every reader has different commands and regular functions, often with radically different syntax. Therefore, the RFID middleware also must be able to control different readers distributed in the physical environment.

The RFID middleware is also designed to control the overall information received from the reader, and centralize the control of the network. All data obtained from readers should be saved in the database without concern for order: the RFID middleware organizes the data and ensures that information can be obtained easily. Therefore, the RFID middleware contains several fundamental functions, including data filtering, counting and aggregation of tag data, enabling it to handle the huge quantity of data generated by the RFID system. Additionally, the object information read from different readers can be analyzed to help understand the meaningful businesses information.

The RFID middleware consists of two communication interfaces, namely the application and the reader interfaces. The application interface communicates with external applications, while the reader interface handles readers, so that external applications and services co-operate with each reader connected to the RFID middleware. A RFID middleware can have multiple middleware services (MidS), each with its own functionality. External services such as EPCIS, ONS and other services could also interact with MidS to perform certain tasks. Fig. 1 shows the basic organization of a RFID middleware.

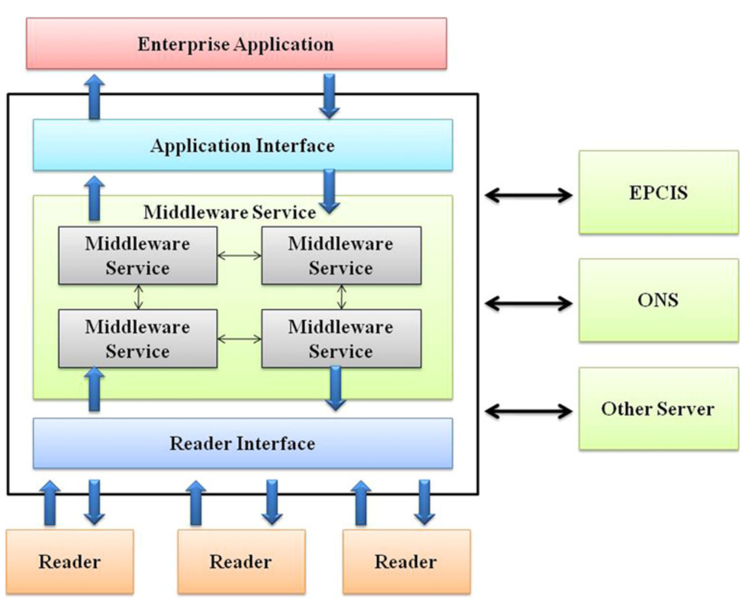

Fig. 1. RFID middleware organization.

2.2. Related research

RFID systems have recently become very popular. The RFID technology and reader performance have been improved significantly. Moreover, many mechanisms have been presented to solve the collision problem, significantly increasing the amount of data handled by readers, and potentially overloading the RFID middleware (Chen et al., 2009; Li, 2006; Christian et al., 2007). Load-balancing mechanisms for RFID middleware have been proposed to solve the problem of overloading. This section discusses various related researches.

2.2.1. Load-balancing method using connection pool in the RFID middleware (Park et al., 2007a, 2007b)

In this research, the component called the connection pool is proposed to deal with the load-balancing issue. Fig. 2 shows the architecture of the connection pool. The pool is built between the readers and the RFID middleware and has several components built inside, connection pool manager (CPM), connection pool table (cpt), middleware manager table (mmt) and interrupt check component (ICC).

CPM allocates tag data to middleware from a connection pool. The tag data from each reader are transmitted to the middleware through connection pool, which then properly distributes the tag data to the connected middleware by the CPM. The proposed method achieves load balancing by the CPM. The CPT finds tag data within the connection pool, which identifies unused buffer, and specifies the tag data to be processed. The MMT is a table that allocates middleware to tag data. The ICC finds malfunctioning middleware, and notifies the MMT.

Tag data input in the connection pool are written into the CPT. The MMT is then checked to determine the middleware to which the data should be allocated. The ICC is then checked to determine whether the chosen middleware is overloaded. If the chosen middleware does not have any problems, then the CPM allocates the data to it.

All RFID readers and middleware hosts are connected with the connection pool in the RFID system; the fact is obvious that the bottleneck is located at the connection pool. Once the connection pool in the RFID system occurs overloading significantly, the system easily suffers from the system crash. While the connection pool system has the huge amount of tag data to be processed, the system will frequently update table information; it may expand the process data time. This proposed architecture is based on a grid-based network as a mean to avoid frequent update of the table information.

2.2.2. Dynamic load-balancing approach on the RFID middleware (Park et al., 2007; Cui and Chae, 2007)

In this research, an agent-based load-balancing approach for the RFID middleware is proposed. Two agents and five policies are developed to deal with the load-balancing problem. Fig. 3 shows the architecture of an agent-based RFID middleware. The proposed two agents are the load gathering agent (lga) and the RFID middleware load-balancing agent (RLBA).

The load-balancing system comprises of five policies concerning information gathering, initiation, job transfer, selection and location. The LGA is a mobile agent, and is compliant with the information gathering policy. The RLBA is a stationary agent, which is compliant with the other four policies. The LGA travels through the RFID middleware to gather the workload information, and maintains information about workload at each RFID middleware host.

The RLBA that resides on each RFID middleware is a stationary agent, and autonomously determines when and where to reallocate overloading readers, and execute the reallocation job. The RLBA is composed of five components, namely load monitor, balancing trigger, middleware chooser, reader reallocator and reader chooser. The load monitor determines who initiates the load-balancing process, and it is compliant with the initiation policy. The balancing trigger determines when the initiator should reallocate jobs to other RFID middleware, and it is compliant with the job transfer policy. The reader chooser determines the readers to be reallocated, and it is compliant with the selection policy. The middleware chooser determines the RFID middleware to which the readers should be reallocated, and it is compliant with the location policy. The reader reallocator reallocates readers to the destination RFID middleware. The proposed LGA and RLBA

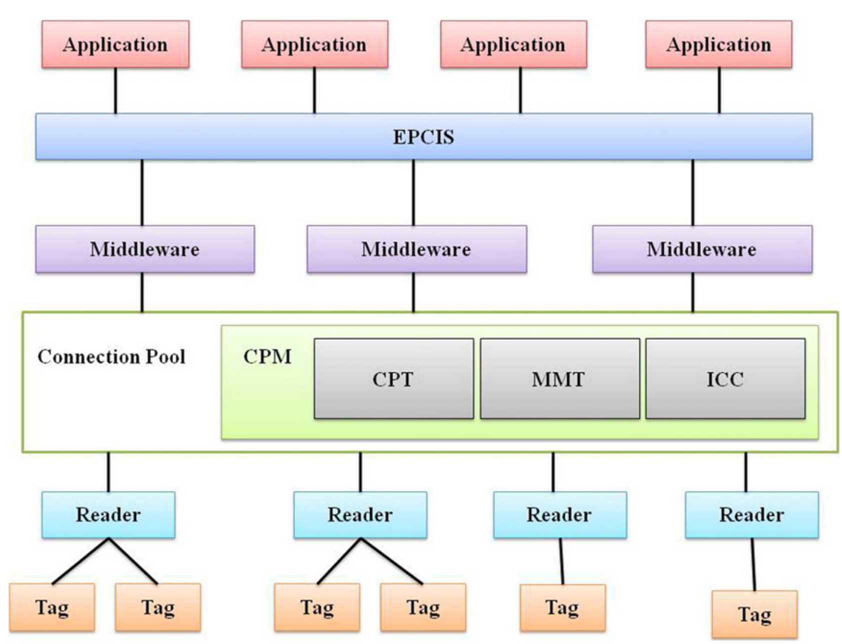

Fig. 2. Architecture of connection pool.