A Novel Complex Event Mining Network for Monitoring RFID-Enable Application

Tao Ku¹ ²  YunLong Zhu¹  KunYuan Hu¹
¹ Key Laboratory of Industrial Informatics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
² Graduate school of the Chinese Academy of Sciences, Beijing, 100039, China
{kutao, ylzhu, hukunyuan }@sia.cn

Abstract

This paper presents a novel Complex Event Mining Network (CEMN) and defines the fundamentals of radio-frequency identification (RFID)-enabled supply chain event management and discusses how an Complex Event Processing (CEP) can be used to resolve the underlying architecture challenges and complexities of integrating real-time decision support into the supply chain. The proposed CEP architecture is a distributed Event Processing Networks (EPNs) capable middleware infrastructure which enables automatic and real-time routing, caching, filtering, aggregation and processing of RFID events. It provides a global platform for distributed execution and management of RFID-enabled supply chain data. It enables a federated control over supply chain nodes deployed in many different organizations, respecting diverse security requirements while supporting centralized deployment and management of processes and rules. Finally, a distributed complex event detection algorithm based on Master-workers pattern is proposed to detect complex events and trigger correlation actions. The results showed that our proposed approach has more robust and scaleable in large-scale RFID applications.

1. Introduction

Over the past few years, a great deal of attention has been directed towards RFID applications. RFID technology can store and retrieve remotely the unique scanner codes contained within specialized RFID tags by means of electromagnetic radiation. Advances in the field of RFID have brought tremendous benefits for supply chain management, especially in Integrating RFID technology into the supply chain makes it possible to detect and address exceptions in the supply chain in real time and close to the source of the exception. For example, using RFID we can detect mistracted materials or identify counterfeit goods while we still have time to react to the exceptions.

However, the progress has also raised important problems on dealing with large volumes of RFID data. An RFID-enabled supply chain generates a large volume of messages from the edge points of the supply chain. The need to manage this data, coupled with the need for real time decision support to handle exceptions, drive the technical requirement for a distributed architecture that allows for local control and global interoperability among all participants in the supply chain network.

The current researches only focus on the question of RFID data cleaning, such as [1], which mainly resolve RFID data errors and duplicate readings. However, the users of RFID application are always not concerned about specific data of RFID tag, but to find valuable information from data. Complex Event Processing (CEP) [4] as an emerging technology has been attracted increasing attention for mining data semantics information, which usually is applied for building and managing information systems in the event-driven enterprise, such as, in [6] present a RFID middleware based on centralized CEP infrastructure, in [5] present several semantic operators and patterns of event in RFID applications. Our early works mainly focus on the design of the complex event pattern based on semantic operators [2, 3].

In this paper, we propose a novel complex event mining network capable middleware infrastructure which enables automatic and real-time routing, caching, filtering, aggregation and processing of RFID events based on our early work, and a distributed complex event detection algorithm based on Master-workers pattern is proposed to detect complex event and trigger correlation actions. The results showed that the distributed CEP system had more robust and scaleable than centralized approach in large-scale RFID applications.
The paper is organized as follows: we first give a brief overview of CEP technology, in section 3, distributed CEP architecture based on complex event mining network is discussed, in section 4, we discuss distribute complex event detection framework and Master-workers algorithm, followed by conclusions.

2. CEP technology overview

Complex event processing is a new technology for extracting information from distributed message-based systems. This technology allows users of a system to specify the information that is of interest to them. This information can be low level network communication message or high level enterprise management intelligence, which depending upon the role and viewpoint of individual users. CEP engine can use components of adapter, filter, and map to create composite events from numbers of primitive events based on time and causal relationships, and can trigger correlation actions by rule definitions, such as alert or warn.

![Centralized CEP Architecture](image)

Figure 1. Centralized CEP Architecture

Figure 1 shows how a centralized CEP architecture is structured to deal with events from RFID device infrastructure. At the adapter layer, events are being monitored from the RFID device infrastructure layer and adapted to CEP format. The next layer is filter layer that eliminate irrelevant events for any further processing by lightweight filter components that is designed to deal with large event throughput. The filtered events are passed on to a layer of maps that apply the rules involved in the CEP application to trigger processing actions. There exists a great deal of CEP approaches in RFID applications. However, current efforts are only focused on elimination of duplicated data and the aggregation of data in the centralized way. In spite of the potential importance of RFID data processing with the centralized CEP architecture, the RFID applications may have to deal with events that are widely distributed across a large system. In this case, there are many monitoring points on the IT layer. The CEP architecture must be structured to merge executions from various sources to present a coherent global view of system activity.

3. Complex Event Mining Network and Distributed RFID CEP architecture

3.1. Complex Event Mining Network framework

This section describes the EPMN as a conceptual framework, in [8] mentions the concept of an EPN; in [9] provides a definition for this concept. These researches are a means to describe and implement event processing, under this paradigm the events are flowing among autonomic EPAs (Event Processing Agents); each EPA is performing a single operation. However it does neither describe the level of distribution nor the level of parallelism among agents. We present the hierarchical agent organization to distribute the task of detection primitive event and composite event, see figure 2.

![Complex Event Mining Network (CEMN)](image)

Figure 2. Complex Event Mining Network (CEMN)

An EPMN model consists of four components: event producer, event consumer, event-processing agent (EPA), and a connection component called an event channel. An EPMN describes how events received from producers are directed to the consumers through agents that process these events by performing transformation, validation, or enrichment. Any event flowing from one component to another must flow through an event channel.

EPMN has two types of EPAs: local event processing agents (LEPA), and domain event
processing agents (DEPA) (see Figure 2). The former is responsible of detecting primitive events generated by local applications in the same machine while the latter is responsible of detecting composite events which are beyond the LEPA scope of knowledge. One or more producer entities (i.e., RFID reader) are connected to a local LEPA in the same machine. Every group of LEPA related to one domain (geographical or logical domain) is attached to one or more DEPAs. These DEPAs are also connected to higher DEPAs to form a hierarchical structure for exchanging the mining information.

### 3.2. Distributed RFID CEP architecture

Based on above CEMN, we implement a distributed RFID CEP software architecture. The most salient features of the distributed CEP architecture are summarized as follows shown as figure 3. First, we provide facilities that allow RFID information consumers to pose queries on the fly, and some of the common RFID data processing tasks are very well-suited for an SQL-based stream query [7]. These facilities include (1) creation of a user query profile for each RFID query to specify the rule constraints and context of what the user wants in this specific query and (2) generation of a set of virtual interface classes that describe the representation of the resulting objects of the query.

![Figure 3. Distributed RFID CEP Architecture](image)

Second, we describe distributed CEP engines information as CEP profile in terms of load and availability of the system resources. Each of CEP engine profiles is created independently by CEP query interface management at the source registration time to capture the usage and constraints of the available source engine data. By describing each distributed CEP engine independently of the other sources and of the user queries, it enables us to incorporate the new CEP engine into our query scheduling process dynamically and seamlessly.

Third, the most importantly mechanisms are provided to dynamically schedule complex event query plan based on the user's query profile and the CEP engine profiles. The partial results of complex event query will be returned after sub-queries are executed at the station of the geographical distribution. Upon the completion of all sub-queries, some final complex event query will carry out.

### 4. Distributed RFID complex event detection

In contrast to the traditional complex event detection in which the centralized server usually collects events from heterogeneous event resources and executes detection tasks, the distributed complex event detection architecture applies JavaSpaces technology as distributed sharing memory constructor, and decomposes complex event detection tasks into many sub detection tasks by Master-workers parallelism. JavaSpaces is one of many implementations of the so called Linda Spaces distributions concept. The underlying idea is that objects can be thrown into a virtual space and taken out, or simply read, by any object connected with the space. Many distributed platforms have been built using this idea. Master-worker parallelism is a widely used form of parallel application programming, which is well suited for solving problems with the characteristics that the total computational work can be broken into pieces, the results of computing any one piece do not depend on the results from any other pieces, and the order in which pieces are computed is not important.

In our approach, we apply Master-worker parallelism for decomposing and executing detection tasks in a distributed environment. A Master process will solve the \( N \) tasks by looking for Worker processes that can run them. The Master process passes an input of the task to each Worker process. Upon the completion of a task, the Worker passes the result of the task back to the Master. The Master process may carry out some intermediate computation with the results obtained from each Worker as well as some final computation when all the tasks of a given batch are completed. The following will give Master-Slave CEP detection framework, then give Master-Slave based detection algorithm and performance analysis.
Figure 4. Master-Slave CEP detection framework

The proposed framework is conceptually very simple, shown as figure 4, which involves dividing a problem into a number of smaller independent work units which can be distributed to remote worker processes for computation in parallel. A single master process centrally controls both the distribution of work units to worker processes and the return of computed results back to the master process. The method of maintaining a collection of work units in a central location for eventual distribution to remote processors is also referred to in the literature as work queue, task queue, and task farm scheduling.

Figure 5 shows an algorithmic view of complex event detecting based on Master-slave.

Algorithm: distributed complex event detecting based on Master-slave pattern.

Input:
- \( P \), detecting plan
- \( C \), detecting constraints set

Output
- \( R \), detecting results set.

Method:
Create detecting plan sequence
\[
P = \{p_1, \ldots, p_n\}
\]
Create detecting constraints sequence
\[
C = \{c_1, \ldots, c_n\}
\]
Create detecting results sequence
\[
R = \{r_1, \ldots, r_n\}
\]
Create detecting tasks
\[
T = \{t_1 = (p_1, c_1, r_1), \ldots, t_n = (p_n, c_n, r_n)\}
\]
Call Master-slave \((P, C, R, T)\)

Procedure Master-slave \((P, C, R, T)\)
(1) \( R_k \leftarrow \Phi \) initial results sequence
(2) for each detecting tasks \( t_i \in T_k \) do
(3) for each detecting plan \( p_j \in P_k \) do
(4) Send \((p_j)\) into CEP engines \( k \)
(5) Receive \((T_k)\) from CEP engines \( k \)
(6) if \( r_k \in R \) and \( t_j \in T_k \)
(7) then update detecting results sequence
(8) if \( R \neq \Phi \) and satisfy \((R, C)\)
(9) then create complex event
(10) else Master-slave \((P, C, R_{k+1}, T)\)
(11) return;

Figure 5. Master-slave complex event detecting algorithm

Firstly, we decompose detecting plan into independent sub tasks based on numbers and capabilities of distributed CEP engines and independent detecting, then we send and execute sub tasks at the CEP engines resided on distributed servers, and collect sub results into result sequence. After each of tasks is finished, we will finally detect complex event based on sub results and custom constraints included time or casual relationship. Finally, complex event will trigger correlation actions to notify customer or business processing by rules.

The algorithm tests were made on 5 PCs with 1.4GHz Pentium IV processor, running Window 2003 as the operating system with 1 GB of RAM. In our experiment, we deploy complex event processing engines called Esper\] on the 4 PCs, and deploy Javaspace on the master server for summiting detecting tasks. The time of complex event detecting is compared with centralized detecting algorithm on the different of event numbers. The result is shown as Figure 6.

Figure 6 presents the comparison of detecting processing time that taken to perform different algorithms. In general, our algorithm is better than traditional method in terms of capabilities of event processing when the number of events increases. However, our algorithm is bad when the numbers of events are lesser than 500 because of distributed communication overhead. Through simulation studies, as the number of events produced increases, it is proved that the growth rate in time of centralized detecting execution has an exponential tendency. However, our algorithm execution time tends to grow linearly.
5 Related works

Complex event processing came from RAPIDE research project at Stanford. The RAPIDE project aims at processing complex events in distributed systems, which has been studied earlier in Active Database community, under ECA rules, in order to build triggers for a variety of purposes. There are quite a few academic projects involved in the area of complex event processing, such as Snoop, ODE and HiPAC [10]. There are other event processing engines provided in the industry, most of which aims at processing information received from RFID technology, such as IBM situation manager [11], Umbrella Program [12], and Progress Event Engine [13]. Most of these works use the event-driven approach, and service-oriented architecture (SOA) to support business functions.

6. Conclusion

In this paper, we propose a novel distributed CEP processing architecture based on complex event mining network for RFID applications. The performance evaluation on distributed detecting algorithm based on Master-workers pattern shows that our approach has more robust and scaleable than centralized approach in large-scale RFID applications. It can be concluded that leveraging distributed computing and CEP technology in RFID applications can bring more benefit for supply chain management or Logistics application. Future work can be in the area of distributed CEP detecting and query techniques based on intelligent optimization methods.

7. Acknowledgments

This work was supported by the Hi-Tech research and development program of China under contract No. (2006BAH02A09).

7. References


