A Novel Distributed Complex Event Processing for RFID Application

Tao Ku[1, 2] YunLong Zhu[1] KunYuan Hu[1] Lin Nan[1]
[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, nl}@sia.cn

Abstract

Radio-Frequency Identification (RFID) technology has brought tremendous benefits for business processing, especially in incorporating RFID data into Supply Chain Management (SCM). With the explosion of RFID data, there has raised important problem how to mine valuable information from tremendous and potentially infinite volumes of RFID data. Complex Event Processing (CEP) technology used to process RFID data has attracted increasing attention. However, the current researches usually focus on centralized CEP architecture, which requires greater bandwidth and computational capability, and Lacks of robustness and scalability because of single point failure or network break. This paper first proposes a novel distributed CEP architecture, which spreads centralized CEP tasks load over multiple communicating stations by a space-based communication paradigm over a distributed message broker architecture based on Jini network. A distributed complex event detection algorithm based on Master-workers pattern is proposed. The results of experiment show that the performance of our approach is remarkable in the 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 business processing, especially in incorporating RFID data into supply chain management, but the progress has also raised important problems [1] how to deal with large volumes of RFID data. The current researches usually focus on the question of RFID data cleaning, such as [9], which mainly resolve the problems of RFID data errors in respects of false negative reading, false positive reading, and duplicate readings. However, the users of RFID application are always not concerned about specific data of RFID tag, but how to find valuable information from data, for example, ''whether the goods ordered from suppliers have storage". Addressing this complex issue of semantics of data processing, the Complex Event Processing (CEP) [4] as an emerging technology has been attracted increasing attention, which usually is applied for building and managing information systems in the event-driven enterprise. Currently, CEP has been integrated into the RFID applications in many research projects, such as, in [7, 8] present a RFID middleware based on centralized CEP infrastructure, in [5, 6] present several semantic operators and patterns of event in RFID applications. However, these researches mostly adopt traditional centralized CEP architecture, which requires greater bandwidth and computational capability as the number of clients increase, and the system is hardly robust and scaleable because of single point failure or network break.

According these challenges, we have proposed a novel complex event processing pattern for RFID applications [2] in the context of RFID middleware project. Our early work mainly focuses on the design of the complex event pattern based on semantic operators [3]. In this paper, we propose a novel distributed CEP architecture for RFID applications 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 is discussed, in section 4, we discuss distribute complex event detection

978-0-7695-3407-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ICCIT.2008.31

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.

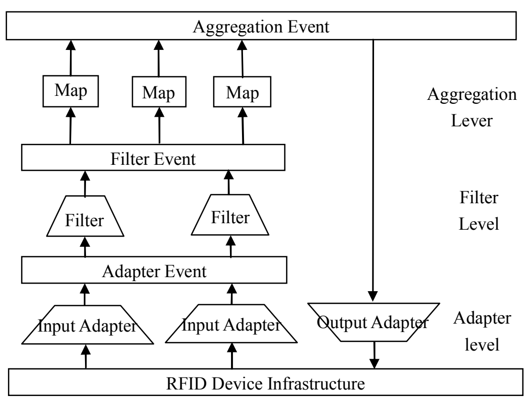

Figure.1. Centralized CEP Architecture

Figure1 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.

In the subsequent sections, we first propose a novel distributed RFID CEP architecture currently being implemented in the RFID Middleware project. Then we describe distributed complex event detection framework and discuss each of the major steps in generating a distributed detection algorithm.

3. Distributed RFID CEP architecture

The most salient features of the distributed CEP architecture are summarized as follows shown as figure 2. 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 [10]. 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.

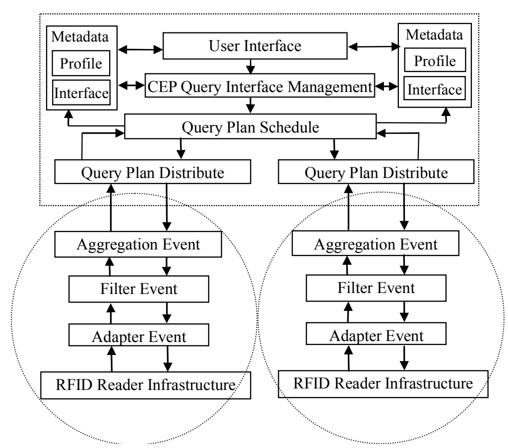

Figure.2. Distributed RFID CEP Architecture

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 Javaspace [14] technology as distributed sharing memory constructor, and decomposes complex event detection tasks into many sub detection tasks by Master-workers parallelism [15]. 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.

4.1. Master-Slave detection framework

The proposed framework is conceptually very simple, shown as figure 3, 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.

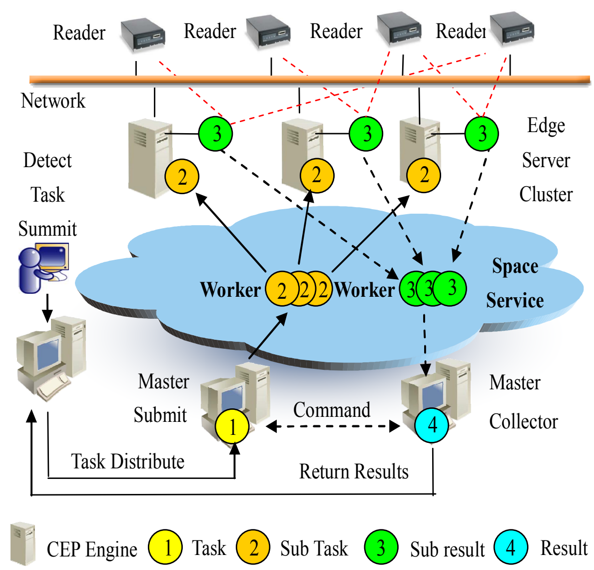

Figure.3. Master-Slave CEP detection framework

4.2. Master-Slave detection algorithm

In this section, we will discuss the methods for detecting complex events. Currently, there exist many algorithms for detecting complex event, such as [11, 12]. The traditional algorithms based on centralized CEP architecture used graph methods to construct event tree for detecting complex events in a bottom-up fashion. However, the centralized host must collect primitive events from many event source nodes for construct event tree, which always needs highly computing and storage capabilities. In order to take full advantage of computing and storage capability residing on distributed CEP engines, we propose distributed detecting architecture.

In contrast to the simple centralized detection, the distributed detecting algorithm is different in sub-detection computing stage. First, Master process distributes N tasks of sub-detection into geographical distribution CEP engines. When CEP engines finish detection tasks, Master process collects partially results of sub-detection for some intermediate computation, and generates final result, and triggering correlation actions by rules. The Master-slave detection flowchart is shown as figure 4.

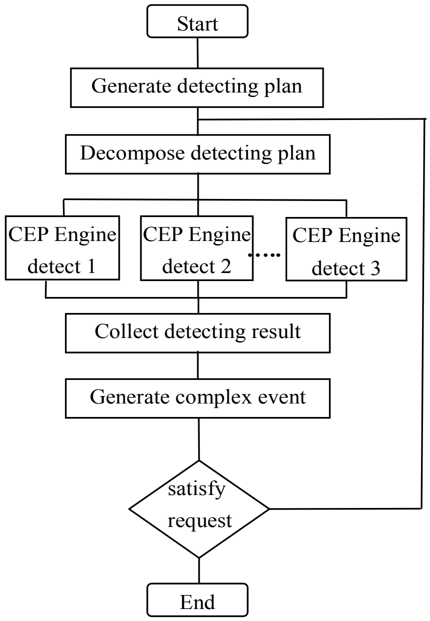

Figure.4. Master-Slave detecting flowchart

Figure 5 shows an algorithmic view of complex event detecting based on Master-slave. First, we decompose detecting plan into independent sub tasks based on numbers and capabilities of distributed CEP engines and independent detecting. It is impartment that plan can be decomposed independently, which likes Sql decomposing into sub sqls, 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.

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(r_k) from CEP engines k
(6) if $r_k \in R$ and $t_i \in T_k$
(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

4.3. Algorithm analysis

The algorithm tests were made on 5 PCs with 1.4GMHz 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 [13] 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.

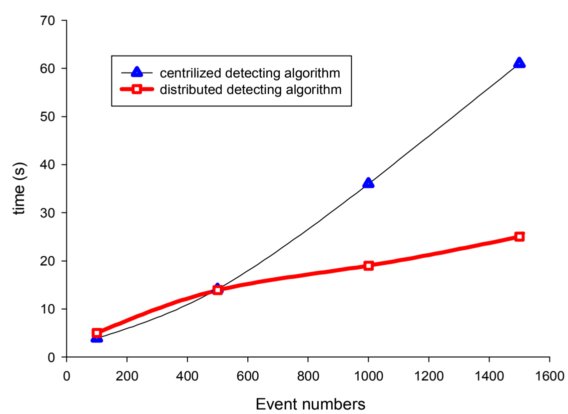

Fig.6. time of detecting complex events in algorithm evaluation

In figure 6 presents the comparison of detecting processing time that taken to perform different algorithms. The horizontal axis corresponds to the quantity of events processed, and the vertical axis corresponds to the amount of time (in seconds) taken in the evaluation of algorithm.

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. Conclusion

In this paper, we propose a novel distributed CEP processing architecture 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.

6. Acknowledgments

This work was supported by the Hi-Tech research and development program of China under contract No. (20060104A1118), National Science and Technology Key Project of China under contract No. (2006BAH02A09).

7. References

[1] M. Palmer. Seven Principles of Effective RFID Data Management. www.objectstore.com/docs/articles/7principles rfid mgmnt.pdf, Aug. 2004.

[2] Ku, T., Zhu, Y. L, and Hu, K. Y. 2007. "Semantics-Based Complex Event Processing for RFID Data Streams" In Proceedings of the First international Symposium on Data, Privacy, and E-Commerce (November 01 - 03, 2007). ISDPE. IEEE Computer Society, Washington, DC, 50-52.

[3] Ku, T, Zhu, Y. L, Hu, K. Y and Lv, C. X, "A Novel Pattern for Complex Event Processing in RFID Applications", *Enterprise Interoperability III*, Springer, 2008, pp. 595-607, DOI = 10.1007/978-1-84800-221-0_47.

[4] David C. Luckham, The power of events: an introduction to complex event processing in distributed enterprise systems [M].Boston: Addison Wesley, 2002.

[5] Wang, F., Liu, S., Liu, P., Bai, Y.: Bridging Physical and Virtual World: Complex Event Processing for RFID data Streams. In: Proc. of the 10th International Conference on Extending Database Technology, Munich Germany (2006)

[6] Daniel Gyllstrom, Eugene Wu, et al, SASE: Complex Event Processing over Streams. In Proceedings of the Third Biennial Conference on Innovative Data Systems Research, (CIDR 2007), Asilomar, CA, January 2007.

[7] C. Zang, Y. Fan, Complex event processing in enterprise information systems based on RFID, Enterprise Information Systems, Volume 1, Issue 1 February 2007, pages 3-23

[8] Liang Dong, Dong Wang, Huanye Sheng, Design of RFID Middleware Based on Complex Event Processing, in: Cybernetics and Intelligent Systems, 2006 IEEE Conference on, June 2006 On page(s): 1-6

[9] Jeffrey R， Alonso G， Franklin M， et al. A pipelined framework for online cleaning of sensor data streams[C]/Proc of ICDE，2006：773- 778.

[10]Q.Jiang, R.Adaikkalavan, and S.Chakravarthy, "Towards an Integrated Model for Event and Stream Processing," TR CSE-2004-10, CSE Dept, Univ. of Texas at Arlington, 2004.

[11] S. Chakravarthy, V. Krishnaprasad, E. Anwar, and S.-K. Kim, "Composite Events for Active Databases: Semantics, Contexts, and Detection," in 20th International Conference on Very Large Databases (VLDB), 1994, pp. 606–617.

[12] Shyh-Kwei Chen, J.-J.J., Henry Chang. Complex Event Processing using Simple Rule-based Event Correlation Engines for Business Performance Management. In Proceedings of the 8th IEEE International Conference on E-Commerce Technology on Enterprise Computing, E-Commerce, and E-Services. 2006.

[13] Bernhardt, T., Esper, Available online at: http://esper.codehaus.org (accessed 21 November 2005).

[14] Eric Freeman, Susanne Hupfer, Ken Arnold: JavaSpaces Principles, Patterns, and Practice. Addison-Wesley Professional, 1. June 1999, ISBN 0-201-30955-6

[15] J. P. Goux, S. Kulkarni, J. Linderoth, M. Yoder, "An enabling framework for masterworker applications on the computational grid", Tech. Report, University of Wisconsin Madison, March, 2000.