J Electr Eng Technol.2018; 13(2): 989-997
http://doi.org/10.5370/JEET.2018.13.2.989

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

An Efficient Complex Event Detection Algorithm based on NFA_HTS for Massive RFID Event Stream

Jianhua Wang[†,*], Jun Liu, Yubin Lan* and Lianglun Cheng*****

Abstract – Massive event stream brings us great challenges in its volume, velocity, variety, value and veracity. Picking up some valuable information from it often faces with long detection time, high memory consumption and low detection efficiency. Aiming to solve the problems above, an efficient complex event detection method based on NFA_HTS (Nondeterministic Finite Automaton_Hash Table Structure) is proposed in this paper. The achievement of this paper lies that we successfully use NFA_HTS to realize the detection of complex event from massive RFID event stream. Specially, in our scheme, after using NFA to capture the related RFID primitive events, we use HTS to store and process the large matched results, as a result, our scheme can effectively solve the problems above existed in current methods by reducing lots of search, storage and computation operations on the basis of taking advantage of the quick classification and storage technologies of hash table structure. The simulation results show that our proposed NFA_HTS scheme in this paper outperforms some general processing methods in reducing detection time, lowering memory consumption and improving event throughput.

Keywords: Complex event detection, Hash table structure, NFA, RFID event stream

1. Introduction

RFID (Radio frequency identification) technology is non-contact automatic identification technology, which mainly uses radio frequency communications to achieve a event acquisition. And with its rapid development, RFID technology are gaining adoption on an increasing scale in many fields, ranging from RFID tracking for supply chain management [1,2] to industrial manufacturing monitoring [3,4]. The wide usage of RFID devices generates massive RFID event streams. Since the massive RFID event streams have the following characteristics: large volume, big velocity, many varieties, small value, and high veracity, etc, that make it difficult to quickly find out some valuable information from them, thus affecting its extensive application. So how to quickly obtain a valuable information from massive RFID event stream become a very important challenge when processing the massive event stream.

Since CED (Complex Event Detection) [5] technology can quickly pick up some valuable information from massive RFID event stream above by taking advantage of the correlation between event attributes, matching rules and algebraic operations, therefore, it gets an increasing application in massive event streams processing recently and has become an important issue in our current research.

In recent, many research works have been carried out for CED. In the detection systems aspects, some general complex event detection systems, such as Cayuga [6], Esper [7], Estream [8], PQS [9], have been developed to provide the basic detection functions for CED. In the detection methods aspects, four general complex event detection methods, for example, complex event detection method based on petri-net [10], complex event detection method based on diagram [11], complex event detection method based on tree [12], complex event detection method based on finite automaton [13], etc, and some of their improved algorithms, such as complex event detection method based on timed petri-net [14], complex event detection method based on optimized directed graph [15], complex event detection method based on compressed composition tree [16], complex event detection method based on pushdown automata [17], and so on, have been developed to detect a complex event from event streams. But since these general detection systems or methods need to execute many redundant search, storage and computation operations when detecting a complex event, thus leading to their long detection time, high memory consumption and low event throughput.

Aiming to solve the problems above, an efficient complex event detection algorithm based on NFA_HTS is proposed in this paper. The achievement of this paper lies that we successfully use NFA_HTS to realize the detection of a complex event from massive RFID event streams,

† Corresponding Author: College of Electronic Engineering, South China Agricultural University, Guangzhou, China. (123chihua@163.com)
* National Center for International Collaboration Research on Precision Agricultural Aviation Pesticides Spraying Technology, Guangzhou, China. (ylan@scau.edu.cn)
** College of Automation, Guangdong Polytechnic Norma University, Guangzhou, China. (liujun7700@163.com)
*** College of Automation, Guangdong University of Technology, Guangzhou, China. (llcheng@gdut.edu.cn)
Received: October 18, 2016; Accepted: November 15, 2017

which can effectively solve the problems above. Specially, in our scheme, after using NFA to capture the related primitive events, we use HTS to store and process the large matched results, as a result, our scheme can effectively reduce lots of redundant search, storage and computation operations by using the quick classification and storage technologies of hash table. The simulation results show that our proposed scheme based on NFA_HTS in this paper outperforms some current general methods in reducing detection time, lowering memory consumption and improving event throughput.

The rest of this paper is organized as follows. In section 2, the related works of complex event detection are introduced. Our proposed complex event detection method is presented in section 3. The experimental results and analysis with our proposed scheme are shown in section 4. In section 5, we give our conclusions.

2. Related Works

In recent years, a lot of related research works have been developed for the detection of complex event. In the detection systems aspects, Cornell University developed a Cayuge system [6], which mainly used customized automatic machine to realize the detection of a complex event from a event stream. Esper Tech company developed an Esper system [7], which mainly used event match rules, event listeners technology, event type inheritance and polymorphism technology, dynamical property settings technology, and so on, to realize the detection of a complex event from an event stream. The Arlington of Texas University developed an ESstream system [8], which mainly took advantage of integrated successive events interrogator and predefined rules to realize the detection of a complex event from event streams. Dartmout University developed a PQS system [9], which mainly used nondeterministic finite automaton, invisible markov processing method, and so on, to realize the detection of a complex event from a event stream. The Berkeley of California University developed a SASE system [18], which mainly used nondeterministic finite automaton and active instance stack to realize the detection of a complex event from real-time RFID event stream.

In the detection methods aspects, four general detection methods of complex event, for example, wang et al. [10] presented a complex event detection method based on petri-net. Bai et al. [11] proposed a complex event detection method based on graph. Sun et al. [12] suggested a complex event detection method based on tree. Mei et al. [13] proposed a complex event detection method based on finite automaton, and some of their optimized algorithms, for example, Jin et al. [14] proposed a complex event detection method based on timed petri-nets. Wang et al. [15] presented a complex event detection method based on optimized directed graph. Li et al. [16] proposed a complex event detection method based on condensed composition tree. Cao et al. [17] proposed a complex event detection method based on pushdown automata structure, and so on, have been developed to detect a complex event from event stream. Eugene et al. [18] presented a high-performance complex event detection method based on NFA_AIS for real-time RFID event stream. Zhang et al. [19] developed a complex event based on improved matching tree structure for distributed uncertain stream. Bok et al. [20] proposed a complex event detection scheme based on minimum conditions for distributed event streams. Wang et al [21] developed a complex event detection method based on multile DAG for multi-probability RFID event stream. Peng et al[22] proposed a scalable event detection method based on NFA state for high speed event stream. Wang et al. proposed a complex event detection method based on probabilistic NFA and AIS to detect a complex event from single distributed probabilistic event streams in the work [23]. Peng et al. proposed a complex event detection scheme based on event selectivity to detect a complex event from live archived event streams in the paper [24]. Moon et al. developed a RFID business aware framework based on business rules to extend RFID events in the paper [25]. Wang et al. proposed a complex event detection algorithm based on RCEDA for RFID event stream in this work [26]. Liu et al proposed a complex event detection engine called DSMS to handle the volume RFID event stream in the paper [27]. In this paper[28], a new parallelization model and three parallel processing strategies are proposed for distributed complex event processing systems to address the difficulties of implementing parallel processing. In this paper[29], a distributed query-plan of complex event processing structure and algorithm based on directed acyclic graph are developed to solve the problem of a single complex event or a small quantity of events. In this paper[30], an intelligent complex event processing method with D numbers under fuzzy environment is proposed to address the issues of intrinsic uncertainty in pattern rules. In this paper [31], a novel complex event detection based on joint max margin and semantic is presented to address the limitations of semantic and temporal features.

Besides, some research works have also been developed for RFID network or networked control system aspects. In order to improve network traffic prediction accuracy, a network traffic hybrid prediction model optimized based on improved harmony search algorithm is proposed in this paper [32]. In order to improve the compensation effect of implicit generalized predictive controller, a time-delay compensation method for networked control system based on time-delay prediction and implicit PIGPC is proposed in this paper [33]. In order to solve the problem that the random time delay in a networked control system can usually deteriorate the control performance and stability of system, a networked control system random time-delay compensation method based on time-delay prediction and improved implicit generalized predictive control is put forward in this paper[34]. In order to solve this problem that stochastic delay in a network can affect its performance, or make the system unstable, a network teleoperation robot system control based on fuzzy sliding mode is proposed in this paper [35].

However, all of detection methods above need to execute a lot of unnecessary search, storage and computation operations when detecting a complex event, thus leading to their long detection time, high memory consumption and low event throughput. Aiming to solve the problems above, an efficient complex event detection method based on NFA_HTS (Nondeterministic Finite Automaton_Hash Table Structure) are presented in this paper under the basis of the analyzing and studying the current detection methods above. In our scheme, we mainly take advantage of the quick classification and storage technologies of hash table to reduce many redundant search, storage and computation operations, and then reduce their long detection time, high memory consumption and improve their low event throughput.

3. Related Works

In this section, an efficient complex event detection algorithm based on NFA_HTS is proposed to solve the problems above.

3.1. Motivation resource

Since the existing methods need to execute a lot of repeatable search, storage and computation operations when detecting a complex event from massive RFID event stream, thus leading to their long detection time, high memory consumption and low event throughput. In order to solve the problems above, we use a hash table structure to store and process the large intermediate matched results by taking advantage of its quick classification and storage technologies under the basis of the analyzing and studying current detection methods after capturing the related primitive event by using NFA from massive RFID event stream. In order to rapidly output the detection results, hash table search technology is used to output the desired results instead of depth first search algorithm in our scheme.

3.2. Working principle

The basic working principle of NFA_HTS algorithm in this paper is that, after using NFA to match the related primitive events from massive RFID event stream, we use hash table structure to store and process the large intermediate matched results, which can reduce lots of search, storage and computation operations existed in current methods by taking advantage of the quick classification and storage technologies of hash table

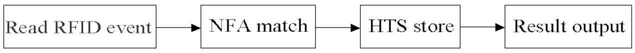


Fig.1. Composition structure of our suggested algorithm

structure, thus reducing detection time, saving memory consumption and improving event throughput for current methods. Fig. 1 is the composition structure of our suggested method.

From Fig. 1, we can see clearly that our suggested NFA-HTS algorithm mainly includes the following four part contents: Read RFID event, NFA match, HTS store and result output. Where, read RFID event mainly takes the reading operations for primitive events from massive RFID event streams. NFA match mainly executes the capturing operations for related primitive events from the read RFID primitive events. HTS store mainly takes the storage operations for matched RFID primitive events by using hash table structure. Result output mainly executes the output operations for desired complex event sequences by using hash table search technology instead of the depth first search technology in current. Fig. 2 is the detailed detection process for our proposed NFA-HTS algorithm.

In Fig. 2, the capital letters (e.g., *E*) represent for event types. The lower-case letters (e.g., *a*) stand for event instances. For example, a_f represents event instance of RFID event type E_a with attributes f; $a_{f,}$, b_f, c_f represent RFID primitive event instance of event type E_a, E_b, E_c with the same associated attributes. The RFID event mainly offers the RFID event resource. It consists of series of various event instances. The timestamp refers to the received timestamp of RFID primitive event.

The NFA is mainly used to capture the related RFID event. Its construction processing mainly includes the following two parts: initialization and construct NFA model. Where Initialization mainly makes some necessary preparations before the program starts to run. Construct NFA model primarily establishes corresponding NFA models through given complex event pattern expressions, and realizes its construction function of NFA model for RFID event. The ①, ②, ③ … stand for automata states.

In our proposed scheme, hash table mapping function mainly realizes the mapping function for mapping the same associated attributes of RFID primitive event into hash table address through some predefined rules. As the related RFID primitive events in pattern expression of complex event have the same associated attributes, for example, a_f, b_f, c_f, we can mappings a_f, b_f, c_f event instance into the same hash table address through some predefined rules.

The C in main node represents the counter, which is used to calculate the number of related RFID primitive event in main node. In our proposed scheme, we need to execute the output operation only if when the C value of main node in main chain is equal to the length value of INFA. The t_{min} represents the minimum time of RFID primitive event in child chain. In our scheme, we can output complex event by hash table search technology when the timestamp of the

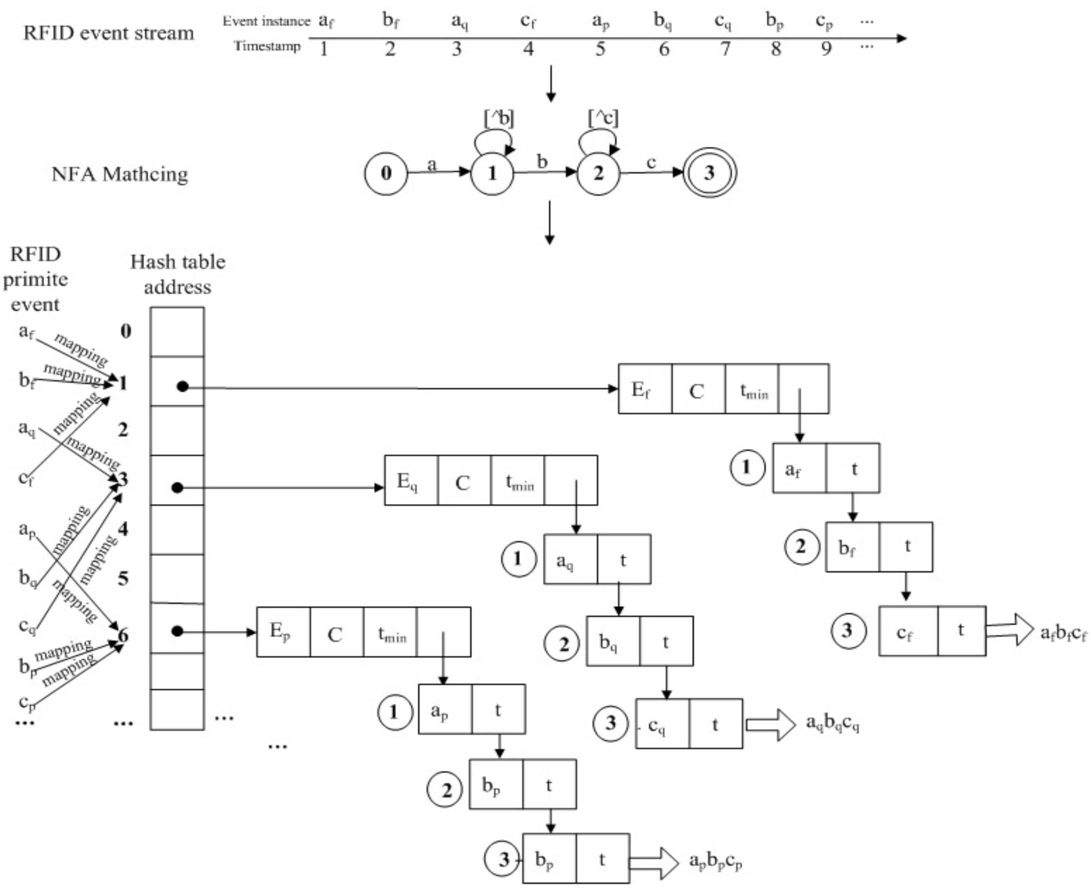


Fig. 2. Detection process of NFA-HTS algorithm

current RFID primitive event-the minimum timestamp t_{min} of main node in main chain < Time Window. The designed main node consists of event type E, counter C, minimum timestamp t_{min} and pointers. The child node includes event instance, timestamp of event instance and pointers.

In our scheme, after using NFA(Nondeterministic Finite Automaton) to capture the related primitive events from RFID event stream, we utilize HTS(Hash Table Structure) to store and process the large intermediate matched result, at last we use hash table searching technology to output the complex event sequences. RFID event mapping, RFID event search, RFID event insertion, RFID event storage and RFID event delete and RFID event output technologies are used to reduce many searching, inserting, sorting, comparing, outputting operations, thus improving its processing performance. We realize the quick classification operations of RFID events by comparing the timestamp of RFID event on the basis of taking good advantage of hash table search and hash table insertion technology in this paper.

Note that, in our suggested algorithm, HTS is created at the beginning running of detection program and then it is used to store and process the lots of events that can trigger transitions of NFA at each NFA state. Hash table search is used to replace the function of depth first search for the construction of complex event sequence when ased to output detection result with the search complexity O(logN), where N is the number of the states in the NFA.

3.3. Realizing steps

The realizing steps of our proposed NFA-HTS algorithm can be summed up the following steps:

Step 1, build the corresponding NFA according to the given event detection expression, and calculate its length, create and initialize new hash table;

Step 2, read a primitive event from massive RFID event streams;

Step 3, judge whether the primitive event is accepted by NFA. If yes, jump step 4 to execute; otherwise, skip step 2 to execute;

Step 4, map the event type of primitive event to corresponding hash array by hash mapping function, and judge whether it has already existed in the hash array. If no, firstly, detection program adds a main node on the primitive event in a main chain of the hash array, which includes event type, counter C of primitive event and minimum time stamp t_{min} of the primitive event. Secondly, detection program adds a child node about the primitive event in a child chain, which only includes event type and the time stamp of the primitive event. Thirdly, detection program updates the value of counter C. If yes, primitive event only adds a child node in child chain with event type and the time stamp of primitive event, and update the value of counter C.

Step 5, judge whether the value of counter C in main chain is equal to the length value of NFA. If yes, jump step 6 to execute; Otherwise, skip step 2 to execute.

Step 6, judge the minimum time stamp t_{min} of main node in main chain + Time Window > the time stamp of the primitive event. If yes, output complex event by hash table search technology. Otherwise, skip step 2 to execute.

The part pseudo codes of NFA-HTS algorithm is as follows.

Part pseudo code of NFA-HTS algorithm

Algorithm1 : complex event processing algorithm based on NFA_HTC
Input: RFID stream S, matching expression Q , hash function H
Output: complex event sequence M

Method:

```
initialized
Build_NFA ←  read(Q );
Calculate_NFA_length();
Create_initialize_hashtable();
read  e_i ← S;
if( e_i is accepted by NFA) then
    hash array←H(e_i);
    search E(e_i)←hash array;
    if(E(e_i) is not in hast array) then
        add Mainnode ←{ Ee_i,C,T_mix};
        add childNode←{ Ee_i,t_ei};
        T_mix ← t_ei;
        Counter+1;
    else then
        add childNode ←{ E_ei,t_ei};
        Counter +1;
    if( value of C == length of NFA) then
        if(t_ei - T_min<Time Window) then
            M ←  Search_hash_table() ;
            return M;
        end if
    end if
end if
go to (5);
```

Noting that, in this section, the event type refers to event type of this primitive event. The counter C of primitive event refers to the calculation function for the total number of primitive events in child node. The minimum time stamp t_{min} refers to the recording function for the primitive event with minimum timestamp in child node.

3.4. Case application

Take detecting complex event matching expression SEQ (A, B, C) from massive RFID event stream for example, to illustrate the detailed realization process of our proposed method.

In step 1, build the corresponding NFA according to the matching expression SEQ (A, B, C), which is shown in fig. 2, and calculate its length for 3, create and initialize a new hash table.

In step 2, read primitive event a_f from massive RFID event stream;

In step 3, since the primitive event a_f can be accepted by NFA, detection program jumps step 4 to execute;

In step 4, map the event type E_f of primitive event a_f to corresponding hash array by hash mapping function. Since there are no event type E_f in the hash array, detection program adds an main node of the primitive event a_f in the main chain of corresponding hash array, which includes event type E_f of primitive event a_f, counter C of primitive event and minimum time stamp t_{min} of primitive event a_f. Then detection program adds an child node in the child chain with event type a_{af} of primitive event a_f and time stamp t_a. At last, detection program adds the value of primitive counter C for 1, updates the minimum time stamp t_{min} for t_a.

In step 5, as the value of counter C in main node (1) is not equal to the length value of NFA(3), detection program jump step 2 to execute. The detection process for primitive event b_f and c_f are the similar way as primitive event a_f .

In step 6, since the minimum time stamp t_{min} in main node+ time window > the time stamp t_c of the primitive event c_f, detection program searches child chain using hash table search technology and output complex event: a_f b_f c_f.

The other primitive event in the RFID event stream, such as a_f, a_p, and so on, can be detected by the similar detection way above.

4. Experimental Results and Analysis

In this section, in order to verify the effectiveness of our proposed method above, some experiments are taken for the purpose. Our designed experiments mainly include four parts: build experimental environment, test detection time, memory consumption and event throughput in different event stream scale respectively.

4.1. Build experimental environment

We implemented our simulation experiment on Microsoft Windows 7 operating systems, AMD A6-3420M 4 core CPU Processor, 2G memory, 500G Hard disk. In our experiments, we use the Visual C++ 6.0 tool to develop a event generator, and then use it to generate different kinds of RFID primitive event streams by controlling some parameters. The main configured parameters in our experiment are listed in Table 1.

Table 1. Main setting parameters in our experiment

Setting parameters	Parameter value
Number of event types	10
Attributes number of per event	2
Scale of event stream	10^5
Length of detection sequence	7
Sliding window size	40
Detection expression	SEQ(A, B, C, D, E, F, G)

In our experiments, the comparison indicators used are detection time, memory consumption and event throughput respectively. The comparison algorithms are three important algorithms in current: NFA_RSS method [36], NFA_AIS method [18] and Zhang' method [37]. Where NFA_RSS algorithm mainly uses NFA and RSS(runtime stack structure) to achieve the detection of complex events; NFA_AIS algorithm mainly uses NFA and AIS(active instance stacks) to realize the detection of complex events; Zhang' method mainly uses a series of optimized algorithms based on nondeterministic automata model to reduce its bottlenecks and realize the detection of complex events after analyzing its complexity of expensive queries of complex event, while our scheme mainly uses NFA and HTS(Hash Table Structure) to realize the detection of complex events. The detailed experimental results are as follows.

4.2. Test detection time

In this subsection, we evaluate the detection time of our proposed scheme with other three methods. Fig. 3 is the experimental results in different event stream scale.

From Fig. 3, we can see clearly that our algorithm shows the least detection time in four methods. Zhang' method followed. NFA_RSS method costs the most detection time. The main reason for that lies that, in our scheme, we use hash table structure to process the large intermediate matched results after capturing the related primitive event by NFA from RFID event stream, which can make full use of the quick classification and storage technologies of hash table to reduce many search, storage and computation operations, thus saving much detection time. Zhang' method spends less detection time lies in its use of series of optimized algorithms based on nondeterministic automata model in its process of complex event. It has the similar performance as our scheme when the scale of event stream is low, but with the increase of event stream, it generates a large amount of runtime intermediate state sequences, which needs to wait for further matching computations due to its redundant intermediate processing results, Therefore consuming more detection time compared with our proposed method in high event stream scale. NFA_AIS method costs more detect time compared lies in the use of previous stack in AIS, which needs many storage and search operations for the more recent instances. NFA_RSS method consumes the most detection time mainly due to its lots of creating operations of virtual edges in RSS.

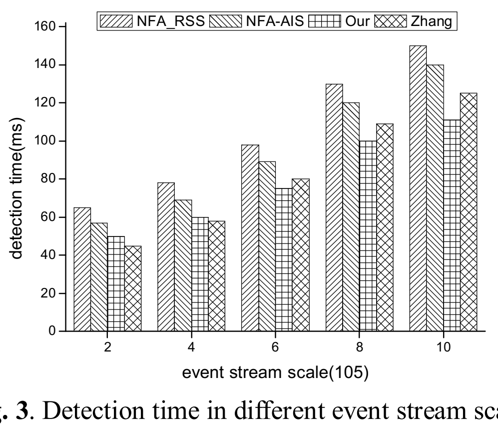


Fig. 3. Detection time in different event stream scale

4.3. Test memory consumption

In this subsection, we test the memory consumption for our suggested method. The testing experimental result is shown in Fig. 4.

Fig. 4 reveals that our proposed algorithm has a significant improvement in saving memory consumption compared with other three methods. Zhang' method followed. NFA_RSS method shows the largest memory consumption. The main reason for them are that, we use hash table structure to store and process the large intermediate matched result in our method, which can reduce many repetitive search, storage and computation operations for many the same RFID primitive events by using hash table structure after capturing the related primitive event from RFID event stream, therefore saving a lot of memory consumption. In Zhang' method, since it uses of series of optimized algorithms based on nondeterministic automata model to realize its complex event detection, which can generate less runtime intermediate state sequences when the scale of event stream is low, but with increase of event stream scale, a large amount of runtime intermediate state sequences are generated, which needs to consume much memory due to its redundant intermediate processing results, therefore consuming more memory consumption in high event stream scale. NFA_AIS algorithm consumes much memory consumption lies in the use of previous stack in its AIS,which needs to take many storage and search operation for more recent instances. NFA_RSS method costs the most memory consumption due to the creating operations of many virtual edges in its RSS, thus having the poor performance of memory consumption saving.

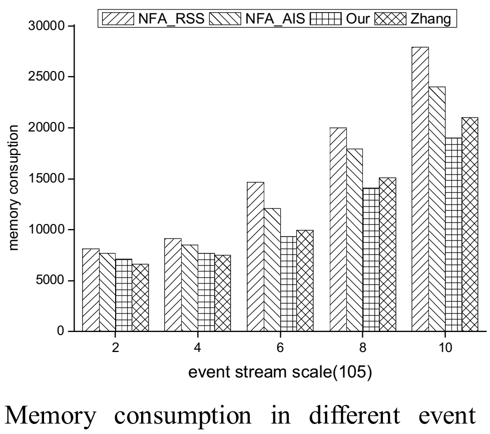


Fig. 4. Memory consumption in different event stream scale

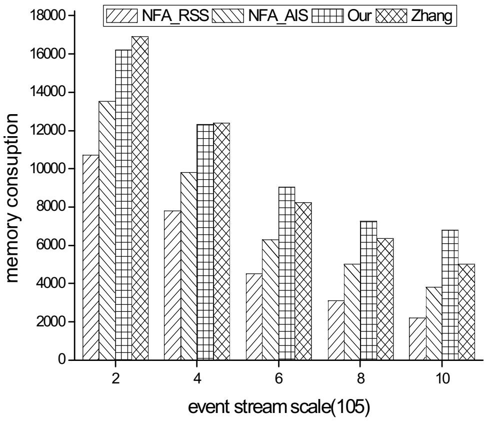


Fig.5. Event throughput in different event stream scale

4.4. Test event throughput

In this subsection, the event throughputs with four algorithms are tested. The experimental results are shown in the following Fig. 5.

From Fig. 5, we can clearly see that that our proposed algorithm in this paper shows good event throughput processing capability compared with other three algorithms in different event stream scale. Zhang' method and NFA_AIS method followed. NFA_RSS method presents the worst processing performance. The main reasons for them are the use of hash table structure in our algorithm. In our scheme, we mainly use hash table structure to store and process the related events of complex event from RFID event stream, which can reduce many search, storage and computation operations for many the same RFID events, thus improving the event throughput. Zhang' method has a better processing capability because of its uses of series of optimized algorithms based on nondeterministic automata model in its process of complex event. It also presents the similar processing performance as our proposed method when the scale of event stream is very low, however, with the increase of event stream scale, it starts to present poor performance compared with our suggested method because a large amount of runtime intermediate sequences generated cannot be processed in time due to its redundant intermediate processing results, therefore leading to low event throughput compared with our proposed method in high event stream scale. NFA_AIS method shows low processing performance lies in the use of previous stack in its AIS, which needs to take many storage and search operations for many more recent instances. NFA_RSS method shows the worst processing performance in four algorithms due to its lots of creating operations of virtual edges in its RSS.

From Fig. 5, we can also observe that four algorithms show litter difference in event throughput when the scales of event stream is relatively smaller, while the large difference between our proposed scheme and other three algorithms will be presented with the growing of event stream scale.

5. Conclusion

In this paper, an efficient complex event detection scheme based on NFA_HTS is presented for massive RFID event stream. In our scheme, after using NFA to capture the related primitive event from RFID event stream, we use hash table structure to store and process the large intermediate matched results, which can effectively improve the detection efficiency by taking advantage of quick classification and storage technologies of hash table structure to reduce lots of search, storage and computation operations. The simulation results show that our proposed NFA_HTS scheme in this paper outperforms some general methods in reducing detection time, lowering memory consumption and improving event throughput as a whole through various experiments.

The limitation of the work is that the proposed method in this paper is mainly applied to ordered RFID event streams. It is difficult to directly apply to disordered RFID event streams and uncertain RFID event streams. So we plan to continue our work for future research in the following directions. Firstly, we will extend our proposed method to suit event matching for out-of-order RFID event stream. Secondly, it will be very useful to develop our proposed scheme for event detection in uncertain RFID event stream. Finally, for deployment in RFID-based applications, we will also enhance our proposed method with support for simultaneous queries, distributed event processing, and so on.

Acknowledgment

The work was supported by the National Natural Science Foundation of China (No.61602187) and (No. 61502110) and (No.61601189); the National Key Research and Development Plan (No.2016YFD0200700) and (2016 YFD0200701); the Guangdong Science and Technology Projects (No.2016A020209007) and (No.2016A020210088); the Guangzhou Science and Technology Project (N0. 201707010482).

References

[1] Virgilio R D, Milicchio F. Physical design for distributed RFID-based supply chain management. DISTRIB PARALLEL DA, vol. 34, no. 1, pp. 3-32, 2016.

[2] Piramuthu S, Zhou W. 5. RFID in food supply-chain management. RFID and Sensor Network Automation in the Food Industry: Ensuring Quality and Safety through Supply Chain Visibility. John Wiley & Sons, Ltd, 2016.

[3] Zhai C, Zou Z, Chen Q, et al. Delay-Aware and

Reliability-Aware Contention-Free MF-TDMA Protocol for Automated RFID Monitoring in Industrial IoT. Journal of Industrial Information Integration, vol. 3, pp. 8-19, 2016.

[4] Ding K, Jiang P, Su S. RFID-enabled social manufacturing system for inter-enterprise monitoring and dispatching of integrated production and transportation tasks. ROBOT CIM-INT MANU, vol. 49, pp. 120-133, 2018.

[5] Del G, Eugene W, Hee J, et.al. “SASE: Complex Event Proeessing over Streams,” In 3rd Biennial Conference on Innovative Data Systems Researeh. pp. 407-411, 2007.

[6] Demers A, Gehlke J, Panda B, etal. “Cayuga:A General Purpose Event Monitoring System. Comell University,” In Proe of 3rd Biennial Conference on Innovative Data Systems Research, Asilomar, USA, pp. 412-422, 2007.

[7] Esper Introduction. http://esper.codehaus.org

[8] Garg V. “Estream: an Integration of Event and Stream Proeessing,” Master Thesis, University of Texasat Arlington, 2005.

[9] Cybenko G, Berk H. “Proeess Query System,” *IEEE Computer*. vol. 40, no. l, pp. 62-70, 2007.

[10] Wang F, Liu S, Liu P. “Bridging physical and virtual worlds: complex event processing for RFID data streams,” LNCS 3896: the 10th International Conference on EDBT. March. pp. 588- 607, 2006.

[11] Bai L, Lao S, Smeaton A F, et al. “Semantic analysis of field sports video using apetrinet of audio-visual concepts,” Computer, vol. 52, no. 7, pp. 808-823, 2009

[12] Sun X W, Chen R, Du Z J. “Composite Event Detection Based on Automata”, in proc of 2009 IEEE International Conference on Intelligent Human-Machine Systems and Cybernetics. Hangzhou, China, pp. 160-163, Aug, 2009.

[13] Mei Y, Madden S. “ZStream: A cost-based query processor for adaptively detecting composite events”, in Proceedings of the SIGMOD2009, USA, pp. 193-206, July, 2009.

[14] Jin X, Lee X, Kong N. “Efficient complex event processing over RFID data stream”, in proceedings of the Seventh IEEE/ACIS International Conferenceon, Portland, pp. 75-81, May, 2008.

[15] Wang F, Liu S, Liu P. “Bridging physical and virtual worlds:complex event processing for RFID data streams,” In proceedings of the 10th International Conference on Extending Database Technology, EDBT’ Munich, Germany, pp. 588-607, March, 2006.

[16] Li Y, Wang J, Feng L. “Accelerating sequence event detection through condensed composition,” Proceedings of the 5th International Conferenceon Ubiquitous Information Technologies & Applications. Sanya, pp. 6-10, 2010.

[17] Cao J, Wei X, Liu Y Q, et al. “LogCEP-Complex event processing based on pushdown automaton,” International Journal of Hybrid Information Technology. vol. 7, no. 6, pp. 71-82, 2014.

[18] Eugene W, Yanlei D, Shariq R. “High-performance Complex Event Proeessing over Streams,” In proe of the SIGMOD conf. Chicago, USA, pp. 407-418, June, 2006,

[19] Zhang X L, Wang Y H, Zhang X M. “Complex Event Processing over Distributed Uncertain Event Streams,” International Conference on Computer Science and Service System. pp. 357-361, july, 2014.

[20] Kyoung S B, Myung H Y, Byoung Y L, et al. “Efficient Complex Event Processing over RFID Streams,” INT J DISTRIB SENS N. vol. 2012, pp. 1-9, 2012.

[21] Wang J H, Cheng L L, Liu J.” A Complex Event Detection Method for Multi-probability RFID Event Stream,” Journal of Software. vol. 9, nol.24, pp. 834-840, (2014).

[22] Peng S L; Liu H X, Guo X L, et al. “State-based event detection optimization for Complex Event Processing,” *Sensors and Transducers*, vol. 164, nol.2, pp. 242-248, 2014.

[23] Wang, Y. H., Cao, K., Zhang, X. M., “Complex event processing over distributed probabilistic event streams,” *Computers and Mathematics with Applications*, vol. 66 no. 10, pp. 1808-1821, 2013.

[24] Peng S L, Li Z H, Cheng, et al. “Complex event processing over live archived data streams,” *journal of computers*, vol. 35, no. 3, pp. 540-554, 2012.

[25] M. Moon, S. Kim, K. Yeom, and H. Choi, “Framework for extending RFID events with business rule, ” in Proceedings of the International Conference on Database Systems for Advanced Applications, pp. 955-961, 2007.

[26] H. Kawashima, H. Kitagawa, and X. Li, “Complex event processing over uncertain data streams,” in Proceedings of the International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 521-526, 2010.

[27] Y. Liu and D. Wang, “Complex event processing engine for large volume of RFID data,” in Proceedings of the 2nd Inter-national Workshop on Education Technology and Computer Science, pp. 429-432, March, 2010.

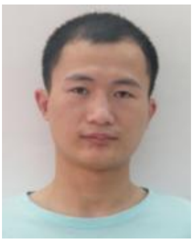
[28] Xiao F, Zhan C, Lai H, et al. New parallel processing strategies in complex event processing systems with data streams. International Journal of Distributed Sensor Networks, vol. 13, no. 8, 2017.

[29] Yuan L, Xu D, Ge G, et al. Study on distributed complex event processing in Internet of Things based on query plan. IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems. IEEE, pp. 666-670, 2015.

[30] Xiao F. An Intelligent Complex Event Processing with D Numbers under Fuzzy Environment. Mathe-

matical Problems in Engineering, vol. 2016, no. 1, pp. 1-10, 2016.
[31] Abbasnejad I, Sridharan S, Denman S, et al. Complex Event Detection Using Joint Max Margin and Semantic Features. International Conference on Digital Image Computing: Techniques and Applications. IEEE, pp. 1-8, 2016.
[32] Z Tian, S Li, Y Wang, X Wang. A network traffic hybrid prediction model optimized by improved harmony search algorithm. Neural Network World, vol. 25, no. 6, pp. 669-686, 2015.
[33] ZD Tian, XW Gao, BL Gong, T Shi. Time-delay compensation method for networked control system based on time-delay prediction and implicit PIGPC. International Journal of Automation and Computing, vol. 12, no. 6, pp. 648-656, 2015.
[34] ZD Tian, SJ Li, YH Wang, HX Yu .Networked Control System Time-Delay Compensation Based on Time-Delay Prediction and Improved Implicit GPC. Algorithms, vol. 8, no. 1, pp. 3-18, 2015.
[35] Z Tian, X Gao, P Guo. Network Teleoperation Robot System Control based on Fuzzy Sliding Mode, Journal of Advanced Computational Intelligence and Intelligent Informatics, vol. 20, no. 5, pp. 828-835, 2016.
[36] Diao, Y., Altinel, M., Zhang, H., Franklin, M.J., and Fischer, P.M. Path sharing and predicate evaluation for high-performance XML filtering. TODS, vol. 28, pp. 4, pp. 467-516, Dec. 2003.
[37] H. Zhang, Y. Diao, and N. Immerman, "On complexity and optimization of expensive queries in complex event processing," in Proc. ACM SIGMOD, Snowbird, UT, USA, pp. 217-228, 2014.

JianHua Wang was born on February 6, 1982 in Guangdong, China. He received his B.S degree in Electronic Information Science and Technology from Shaoguan University, Guangdong, China, in 2006. He received his Ph.D degree in Control Science and Engineering at Guangdong University of Technology, Guangdong, China, in 2015. Currently he is a teacher of college of electronic engineering, south china agricultural university, Guangzhou, China. His research interests include wireless video transmission, cyber-physical systems and IoT.



Jun Liu was born on October 11, 1986 in Hubei, China. He received his M.S degree in Control Science and Engineering from Guangdong University of Technology, Guangdong, China, in 2012. Currently he is a teacher of College of Automation, Guangdong Polytechnic Norma University, Guang-zhou, China. His research interests include wireless sensor networks and cyber-physical systems.

Lan Yubin was born on in Jilin, the special expert of national "thousand person plan", professor, deputy director of the agricultural aviation branch of China Agricultural Engineering Society. His research interests include precision agriculture aviation, aerial pesticide application technology and aerial remote sensing technology.

LiangLun Cheng was born on August 22, 1964 in Hubei. He received his M.S and Ph.D degrees from Huazhong University of Science and Technology, Hubei, China in 1992 and Chinese academy of Sciences Jilin, china in 1999 respectively. He is a Prof and doctoral supervisor of Guangdong University of Technology. His research interests include RFID and WSN, IoT and CPS, production equipment and automation of the production process, embedded system, the complex system modeling and its optimization control, software of automation and information, etc