An Intelligent Complex Event Processing with D-S Evidence Theory in IT Centralized Monitoring

Bin Cao and Jiyun Li

School of Computer Science and Technology
Donghua University
Shanghai, China
2111453@mail.dhu.edu.cn, jyli@dhu.edu.cn

Abstract. For the mass events monitored from IT infrastructures, it is critical that how to use them effectively to get more valuable information is critical, especially for the large-scale enterprises, application performance issues have an immediate impact on customers' experience and satisfaction. The challenge is that the systems are under uncertain and dynamic operating conditions, so it is difficult to identify the complex critical situations from the large numbers of various events. The existing complex event processing (CEP) systems cannot solve the problem efficiently, in this paper, we propose an intelligent approach to solve the complex event correlation. Combining the CEP with Dempster-Shafer (D-S) evidence theory method, we can find the anomalies timely and locate the root cause automatically. It offers leverage by considering the real-time and accuracy in detecting events.

Keywords: event correlation, complex event processing, uncertain, application performance signature, Dempster-Shafer theory.

1 Introduction

In recent years, with the rapid development of the internet and the increasing of the network size, the management is becoming more and more difficult, but vast amounts of data also becomes important information that people can use. Whether it is the potential threat or opportunity, it is necessary to monitor the network timely. As the IT facilities and resources in all walks of life are deployed perfectly, business is increasingly dependents on highly information-oriented integration. To ensure the whole network running well, it is essential to monitore these devices in real time. A sudden slowdown of application can affect large numbers of customers, which will lead to delay projects and ultimately it can result in the company financial loss. In the existing solutions, CEP is an efficient method for surveillance applications in IT Resource Monitoring System.

CEP is used for analyzing the events coming from multiple sources over a specific period of time by detecting complex patterns between events and by making correlations. It can detect meaningful events or pattern of events which signifies either threats or opportunities from the series of events received continuously, and send

M. Pathan, G. Wei, and G. Fortino (Eds.): IDCS 2013, LNCS 8223, pp. 387–398, 2013.

alerts for the same to responsible entity to respond as quickly as possible. Diagnosis or prediction of system failure has been a hot research topic, people have proposed different solutions, but most of the CEP systems are dealing with the flow of events, and the basic premise underlying the design of the CEP queries proposed is that the associated rules of events are predefined. For the uncertain or unexpected events, the root cause is difficult to diagnose. Moreover, many of predefined rules are based on subjective experience, it's not necessarily accurate in practice. Coupled with the impact of environment, the derived common sets of rules are very difficult to apply for the failure analysis. While rule-based reasoning system structure looks simple, the acquisition of knowledge is a big bottleneck in the system, the rules come from experts, so there are not self-learning function. Even in the process of deductive reasoning, there is no full use of the past experience and lack of memory.

In this paper, we propose an intelligent solution to solve the mass integrated data, which is difficult to define the rules in general and detect the failure automatically. It can output the specific events defined by the rules and classify the uncertain events. For the uncertain events, we use the D-S theory based on rule and application performance signature to analyze the root causes of failure. According to the established rules, we can get the judgment quickly.Our main contribution is that: combining the CEP with D-S techniques, we can quickly locate the anomaly for the uncertain events. As many alarm events related to the detected root cause will not send to the user, it will reduce lots of storage space and decrease the detection time compared to artificial screening. For the event whose detection conditions are inaccurate or inadequate that results in the root cause cannot be detected or analyzed timely, we offer leverage by considering the real-time and accuracy in detecting events.

Next, we will do a brief statement on the related work in the section 2, the section 3 introduces the system's architecture and details on how the CEP classify collected events and how to use the D-S theory to determine the root cause for uncertain events. The section 4 is the experiment evaluation. Finally, a summary for the paper is given.

2 Related Work

Modern event processing is fast becoming a foundation of today's information society. This is a technology that emerged in the late 1990s, but event processing is nothing new. In the 1990s, there were one or two university research projects dedicated to developing new principles of event processing, called CEP. After 2000, CEP began to appear in commercial IT applications, and more recently, starting around 2006, the number of event processing applications and products in the market place has grown rapidly[1].It mainly includes two parts: the commercial products , such as Esper[2],STREAM[3], WebSphere[4] etc., and the research-based projects, such as Cayuga[5],Gigascop[6]. These applications have demonstrated their own advantages, but they are almost for the specific application, and there is no uniform language and rule to define events. Moreover, the premise is proposed based on the predefined rules, even the scalability is well, in practical use, many of the rules are difficult to be defined artificially and most of them are dynamic, and they will vary with the change of environment.

So far, CEP systems are almost used in event-driven architectures, business activity monitoring, business process management, enterprise application integration, network and business level security etc. Network management monitoring and related application monitoring are presented, but the research putting them together is quite a few. Among the appeared centralized monitoring tools, the defined rules are simple, it cannot automatically find all the root cause cascading the failure. For event correlation and its root causes analysis, the main methods are the associated rules, Codebook correlation and artificial intelligence[7]. However, these methods have some short comings, they do not have an appropriate combination of the time with the occured correlation and the accuracy of the data returned by a correlation.

Then [8] suggested integrating events generated across different monitoring tools and putting them together to do analysis, which is a valuable research. The different monitoring tools capture certain information to a data center, by using CEP system to associate events automatically, it generates alarm event or potential trend. Some even proposes detecting the root cause of system anomaly automatically. The automated tools are essential for many performance analysis and applications to understand the application behaviors and changes during the application lifecycle. However, the traditional reactive approach is to set thresholds for observed performance metrics and to raise alarms when these thresholds are violated. This approach is not adequate for understanding the performance changes between application updates. Instead, a proactive approach based on the continuous application performance evaluation may help enterprises reduce loss of productivity. By collecting performance data and correlated event logs and environment variables to detect the essential performance changes in applications, many root cause of the anomaly can be captured in real-time. So information integration is necessary, there are mutual effects between different applications. For example, network failure is likely to pose a threat to service application, also may influence the storage. If they are analyzed separately, many root causes will not be detected. Therefore, in order to accurately determine the root cause of failure, we need not only the certain rules definitions, but the machine learning. Adopting the method of artificial intelligence to train the history data set, we can give out the possibility of different events as root cause. It can be obtained the root cause quickly from the least possible event failure information.

In this paper, we will focus on our specific application, namely the centralized monitoring in IT infrastructure. Event correlation is no longer confined to a single application, we collect the different sources through certain rules, such as filtering, aggregation and so on. From the associated event set, we can find out more potential rules and applications. Some alarm events are hard to find the root cause from a single application, combined application is more accurate to locate. For the failure events, we adopt the method of combining rules associated with D-S theory to obtain the root cause by the probability, which can not only solve the randomness and volatility of the event correlation, but also a substantial reduction in manual. Next we will introduce the related work in details.

3 Integrated Monitoring and Correlation Architecture

Before introducing the associated event, we first have a clear understanding of the event. The event is divided into basic event and composite event. Basic event, also known as simple event, atomic event or single-event, is abstract and predefined by the system, embedded in system to be detected or generated by a fixed mechanism. Composite event is composed of other abstraction events which can be basic events or mixed events. They are defined by the uniform rules attributes. The event is expressed by event id, type, operator, value, timestamp and some additional information. The operator includes unary, binary and customized.

3.1 System Architecture

The whole centralized monitoring system architecture is shown as Fig.1, event correlation mainly involves three blocks: event collector engine, CEP engine and D-S evidence theory. First, the event collector engine, collecting events from different monitoring tools by filtering, conversion, aggregation etc., generates a unified formal definition of the event flow. Followed by the CEP engine, the core of the entire system, processing the collected events through the rule definitions, the generated events is sent out directly to particular users, or reused as input events, or used to do further analysis. Finally, the D-S evidence theory method, which is the addition to the CEP results, in view of the detected uncertainty event set, is used to do further analysis. On one hand, it can detect the root cause from the many alarm events, and reduce large numbers of alarm events to produce, on the other hand, the possible root cause is judged automatically, saving time and manpower.

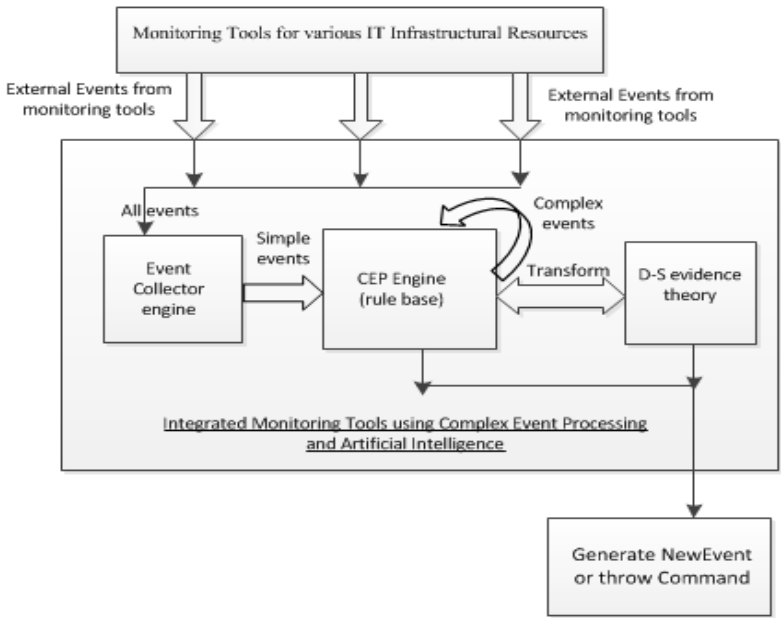

Fig. 1. Intelligent CEP system

Next, we will introduce the key part of the CEP engine, the query model. Then we will focus on the D-S evidence theory method.

3.2 The Query Model

Our event processing system with the modified Cayuga system, which adopts the method of stream processing and event processing, allows the information integration from different monitoring tools. [9] has relevant event model, and made a detailed query engine. Here, we don't do extra statement on event expressions and rule definitions description.We mainly use their framework to process the events, but for the specific rule definition, we do some modification and use the following form:

SELECT [attributes]
FROM [expression]
DO generate New Event or throw Command
Here, the [expression] is constructed as follows:
TYPE Rule content
DECLARE EventClass virtual events list
ON Event Pattern
WHERE Constraints satisfied

TYPE describes the function of the rule. In the rule, we describe the type of detection made. DECLARE lists the virtual events and their classes that are taken as input and the name of the variable representing that class (it is necessary if we have two events of the same class in a rule).ON describes the expected pattern, the declared events are used in the pattern that describes their relationships and the contextual constraints are defined separately. WHERE states the contextual constraints of the rule.

DO describes the actions to be triggered if the rule is activated, 'generate' indicates a new complex event, it will be reused or to be send out, 'throw' indicates a command to do the further analysis.

For different events, we take different rules to process them. We mainly divide them into three cases. For a single application, such as monitoring the service performance, the load of CPU can be detected by stream processing in a certain time interval, output the average CPU utilization. It also can be checked by the rule engine, setting the initial value, an alarm event is generated when a certain threshold is reached. For the complex event from different sources, some rules are very explicit, and it can be obtained by rule matching. For example, all of the applications of network, services and storage send out alarm events. If the network fails, it will cause that the service cannot get request, the store content is empty, so we can define rules to discover automatically the root cause.

For some other events, its root cause is difficult to define, may be the same events in different situations are caused by different events. For example, the server is down, the cause may be: 1. hardware failure; 2. software with a bug; 3. network failure, such as DNS server is down; 4. power outages caused by natural disasters; 5. the fallibility caused by humans. They are all likely to be the cause, and what we can do is to find

out all the possible causes and store them. Then we can do the further analysis. The algorithm for selecting uncertain event set is as follows:

```
1. Initialize HS //HS-Hypothesis Set
2. While Constraints satisfied
Do
3. Select h from HS
4. Use inference rule r∈R
5. Apply r to h to yield h1,h2,…,hn
6. Add h1, h2,…,hn to HS
7. Del h from HS
8. Upon cd // cd-correlation description
9. Adjust HS
10. End
```

3.3 D-S Evidence Theory

The Motivation for Application of the D-S Theory. D-S theory is a generalization of the Bayesian theory of subjective probability [10], a transferable belief model, but unlike Bayesian theory, the D-S theory has the ability to represent uncertainties and lack of knowledge. [11] combined different classifiers' results to get a better result, this application inspired us to apply the D-S theory based on the result of CEP in the monitoring IT infrastructure. For the alarm event and these associated with it at the same time, how to determine the root cause, we adopt epistemic event-based correlation approach to solve it, that is the D-S theory based on rule and application performance signature. By giving the probability of events associated with each other and the decision rules, the root cause can be given out by its reliability. Then choosing the root cause from the given events by their degree of belief, so a lot of manual work is saved and more accurate to locate the root cause. [12] verified the feasibility of the method in the use of network management.

Related Concepts and Definitions. In the D-S based on rule and application performance signature, we use some related concepts and theories, referencing[11, 13].

Definition 1. The frame of discernment: a finite set of all possible hypotheses, expressed in Θ, $\Theta=\{\theta_1,\theta_2,...,\theta_K\}$. Its power set is denoted by 2^{Θ}.

Definition 2. Basic probability assignment (BPA): a basic belief assignment m is a function that assigns a value in [0, 1] to every subset A of Θ and satisfies the following cont:

$$m(\phi)=0, and \sum_{A\subseteq\Theta} m(A)=1 \tag{1}$$

Definition 3. Belief function: the belief function $bel(.)$, associated with the BPA $m(.)$, is a function that assigns a value in [0, 1] to every nonempty subset B of Θ. It is called "degree of belief in B" and is defined by

$$bel(B)=\sum_{A\subseteq B} m(A) \tag{2}$$

Definition 4. Plausibility measure: the plausibility (Pls) is a function that the sum of all the basic probability that overlaps with certain set. The formula is:

$$Pls(A) = \sum_{B \cap A \neq \phi} m(B), \quad \forall A \subseteq \Theta \tag{3}$$

It can be shown that the plausibility of an event equals one minus the belief of the complement of that event, and vice versa. That is $Bel(A) = 1 - Pls(A^c)$.

Definition 5. For $\forall A \subseteq \Theta$, the interval $[Bel(A), pls(A)]$ is referred as the reliability interval in A. The Fig.2 describes the uncertainty of the hypothesis.

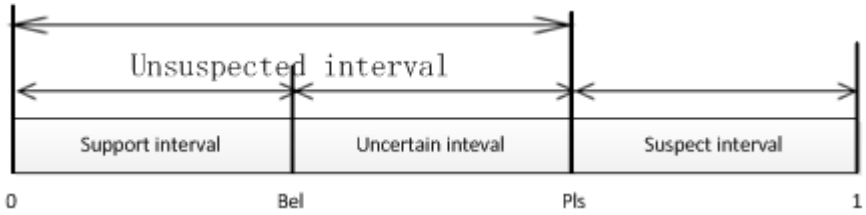

Fig. 2. Uncertainty expression of hypothesis

Definition 6. Focal element: any subset x of the frame of discernment Θ, for which m(x) is non-zero, is called a focal element and represents the exact belief in the proposition depicted by x.

Combination Rule of D-S. Combination rule is a union of different symptom parameters. Given several belief functions based on different rule parameters from unified frame of discernment, and consider the conflicts between the rule parameters. As follows, considering two BPAs m1 (.) and m2 (.) for belief function bel1 (.) and bel2 (.) respectively. Let Aj and Bk denote the focal elements of bel1 (.) and bel2 (.) respectively. Then m1 (.) and m2 (.) can be combined to obtain the belief mass committed to $C \subset \Theta$ according to the following combination or orthogonal sum formula:

$$m(C) = \begin{cases} 0, C = \phi \\ m_1 \oplus m_2(C) = \dfrac{\sum\limits_{j,k,A_j \cap B_k = C} m_1(A_j) m_2(B_k)}{1 - \sum\limits_{j,k,A_j \cap B_k = \phi} m_1(A_j) m_2(B_k)}, C \neq \phi \end{cases} \tag{4}$$

So is the combination of the belief function, that is,

$$((bel_1 \oplus bel_2) \oplus bel_3) \oplus \ldots \oplus bel_n = \bigoplus_{i=1}^{n} bel_i \tag{5}$$

The Method for Failure Diagnosis. First of all, we will estimate the values of $m_n(\theta_k)$ and $m_n(\Theta)$, which represent the belief in hypothesis k that is produced by condition rule for symptom en and the ignorance associated with condition rule for

symptom en. Adopting the method of distance measure, we compute the BPA. Then $m_n(\theta_k)$ and $m_n(\Theta)$ will be estimated according to the following formulas:

$$d_n(\theta_k) = \exp(-\left\| w_k^n - y^n \right\|^2) \tag{6}$$

$d_n(\theta_k)$ is the distance measure used to estimate the BPAs. y^n is the output hypothesis vector produced by condition rule for symptom, w_k^n is a reference vector.

$$m_n(\theta_k) = \frac{d_n(\theta_k)}{\sum_{k=1}^{K} d_n(\theta_k) + g_n} \tag{7}$$

$$m_n(\Theta) = \frac{g_n}{\sum_{k=1}^{K} d_n(\theta_k) + g_n} \tag{8}$$

Where $m_n(\theta_k)$ and $m_n(\Theta)$ are the normalized values of $d_n(\theta_k)$ and g_n respectively.

To minimize the mean square error (MSE) between the output vector M which represents the combined confidence in each hypothesis and the target vector T, we adjust the values of w_k^n and g_n through the training set. The metric is denoted by Err, that is $Err = \left\| \mathrm{M} - T \right\|^2$.

w_k^n and g_n are initialized randomly, their values will be adjusted according to the following formulas:

$$w_k^n[new] = w_k^n[old] - \alpha \frac{\partial Err}{\partial w_k^n[old]} \tag{9}$$

$$g_n[new] = g_n[old] - \beta \frac{\partial Err}{\partial g_n[old]} \tag{10}$$

Where α and β are the learning rates, they are set according to the requirements of precision.

Then we design decision rules to obtain the final result. According to the belief interval $[Bel(A), Pls(A)]$ of all the hypotheses and the uncertain probability $m_i(\Theta)$, we choose the hypothesis having the max belief value as the root cause when satisfying the following rules:

Rule 1: $m(\theta_i) = \max_j \{m(\theta_j)\}$;

$m(\theta_i) - \max_{j \neq i} \{m(\theta_j)\} > \varepsilon$,

Rule 2: $m(\theta_i) - m(\Theta) > \varepsilon$,

$\varepsilon \in R \wedge \varepsilon > 0$;

Rule 3: $m(\Theta) < \gamma, \gamma \in R \wedge \gamma > 0$.

The whole algorithm steps are as follows:

1. $\Theta \leftarrow \mathrm{HS}$; // the hypothesis set of probably root cause
   $E \leftarrow \mathrm{cd}$; // the correlation descriptions for condition rule of symptom
2. initialize w_k^n , g_n , T, and $Ite\_no_{\max}$;
   initialize i=0, $\alpha = 5 \times 10^{-4}, \beta = 10^{-6}$, MSE;
3. While (i< $ite\_no_{\max}$ || $\alpha > 10^{-4}$)
4. For each pattern
5. computer $m_n(\theta_k)$ and $m_n(\Theta)$;
6. computer M and Err_{new} ;
7. adjust w_k^n and g_n ;
8. If $Err_{old} - Err_{new} > MSE$
9. $\alpha = \alpha \times 1.03$;
10. else
    $\alpha = \alpha \times 0.7$;
11. i++;
12. End while
13. If the condition rule satisfied then
14. create composite event CE
    $CE \leftarrow \theta_i$;
15. end if

4 Experiment Evaluation

Our experiments are conducted on a machine with Intel(R) Core(TM) i5-2400 cpu @ 3.10 GHz processor and 4 GB main memory, running MyEclipse 8.6 on Windows7. The algorithms were implemented in Java on standard datasets which come from [14]. Assuming the network congestion, we give several kinds of root causes, here is the teardrop attack, smurf attack and udpstorm attack, because the symptoms they show are alike. Then we classify the information for three segments: 1. network connection based on the content features; 2. the basic characteristics of network connection; 3.network traffic features. Depending on the training data, the BPAs for each possible root cause under different symptoms will be calculated. Finally, based on the diagnosis rule, the possible root cause can be judged. To reduce experimental variance, we perform each experiment for several times, and report the average result we measured.

Our evaluation had two main goals: assessing the validity of the method and evaluating the degree of accuracy. Consequently, we separate the two aspects by first measuring the query result and then analyzing the accuracy rate compared to the fact.

The experiment contains three aspects of the content features used to detect the root cause, the probable hypotheses are three. We select all the hypotheses from the dataset, 280790 records for smurf attack, 979 records for teardrop, and 2 records for udpstorm. The original database is split into two parts--while conserving the proportions of detection failures-one is used for model construction, and the other for model evaluation. The final result is shown below, in Fig.3, r1, r2 and r3 denotes the possible cause, u denotes an unknown cause, the horizontal axis denotes the symptom information for failure.

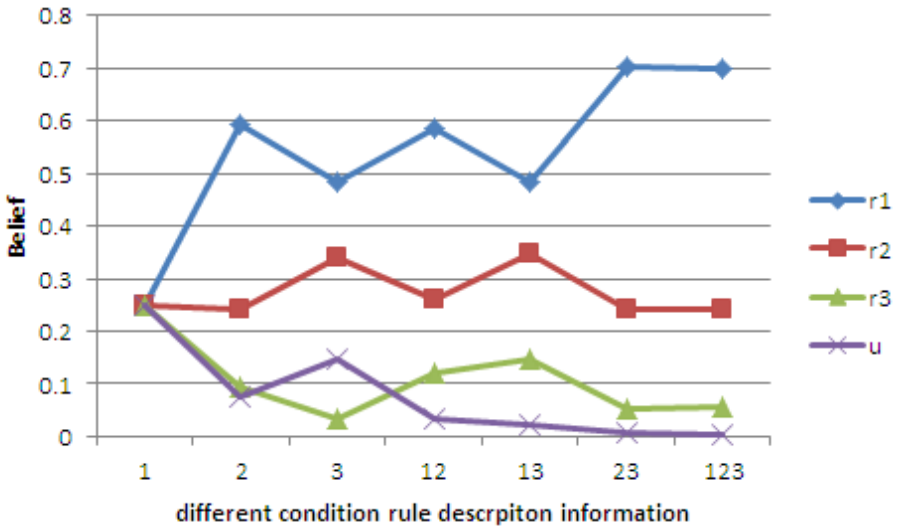

Fig. 3. Changing belief in probable cause with different symptom information

As it can be seen from the graph Fig.3, it is difficult to diagnose failure only using single description information, low belief cannot accurately identify the fault type, and the multi-source information infusion can improve the correct rate for failure detection.

Then we verify the accuracy of the results, using different numbers of training sets to count. The below fig. 4 shows the average accuracy rate. Obviously, the more experimental data, the more accurate the results are. But when it reaches a level, the rate remains the same. This shows that in the practical application, there should be more representative feature selection and rich experience accumulation. It depends on a lot of professional knowledge experience and lots of history records.

Last, it is obvious that the automated detection time and response time are faster than the manual, and because of the further analyses, many alarm events with root event occurred simultaneously will not be generated. A small fault could be a sign of big problem, only trying to avoid it, can the big fault occurrence rate be reduced. We must record all the anomalies and their solutions. So that it can be done the further research.

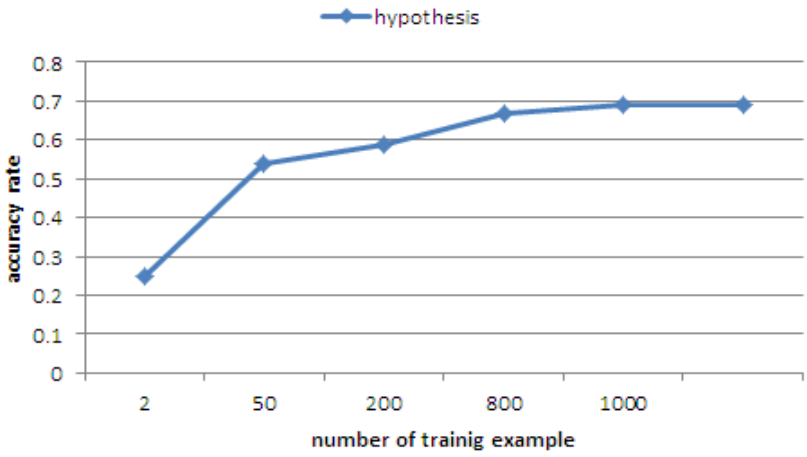

Fig. 4. Detection precision under different training sets

5 Conclusion

In this paper, we propose an intelligent approach for event correlation, particularly in uncertain or unknown events. It reduces large numbers of alarm events to generate finally. The experiments verified theoretically the rationality and validity of the scheme. While this paper also has its shortcoming, namely, when the anomaly or alarm events happen, the fault symptoms are known to us. The further study will focus on the selection of representative and useful performance variables. So far, there is no clear rule features for a lot of anomaly, the efficiency of probability calculation is still a problem to solve, depending on lots of training data. We believe that this will be solved with the in-depth study and the accumulation of a large number of historical record data. In practice, we don't want big failure to happen, but rather to keep timely inspection of every link to ensure the overall work well. So with the rapid expansion of the network, stronger fault self-healing ability has become an important research topic in the next generation of data center platform.

References

1. Etzion, O., Niblett, P.: Event processing in action. Manning Publications Co. (2010)
2. `http://esper.codehaus.org/`
3. Arasu, A., Babu, S., Widom, J.: The CQL continuous query language: semantic foundations and query execution. The VLDB Journal—The International Journal on Very Large Data Bases 15(2), 121–142 (2006)
4. Leymann, F., Roller, D., Schmidt, M.-T.: Web services and business process management. IBM Systems Journal 41(2), 198–211 (2002)
5. Demers, A., et al.: Cayuga: A general purpose event monitoring system. CIDR (2007)

6. Cranor, C., et al.: Gigascope: a stream database for network applications. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. ACM (2003)
7. Tiffany, M., A survey of event correlation techniques and related topics. Research paper. Georgia Institute of Technology (2002)
8. Narayanan, K., Bose, S.K., Rao, S.: Towards' integrated'monitoring and management of DataCenters using complex event processing techniques. In: Proceedings of the Fourth Annual ACM Bangalore Conference. ACM (2011)
9. Hong, M., et al.: Rule-based multi-query optimization. In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology. ACM (2009)
10. http://en.wikipedia.org/wiki/Dempster%E2%80%93Shafer_theory
11. Al-Ani, A., Deriche, M.: A new technique for combining multiple classifiers using the Dempster-Shafer theory of evidence. arXiv preprint arXiv:1107.0018 (2011)
12. Ganapathy, V., et al.: An epistemic event-based correlation approach for managing pervasive networks. International Journal of Network Management 22(1), 81–94 (2012)
13. Mordeson, J.N., Wierman, M.J., Clark, T.D., Pham, A., Redmond, M.: Evidence theory. In: Mordeson, J.N., Wierman, M.J., Clark, T.D., Pham, A., Redmond, M. (eds.) Linear Models in the Mathematics of Uncertainty. SCI, vol. 463, pp. 41–56. Springer, Heidelberg (2013)
14. http://kdd.ics.uci.edu/databases/kddcup99/