A Distributed Network Fault Diagnosis Platform based on Web Services

LI Bo
Department of Compute Science and Engineering
PLA University of Science and Technology
Nanjing, China

ZHANG Guo-min
Department of Compute Science and Engineering
PLA University of Science and Technology
Nanjing, China

WEI Xiang-lin
Department of Compute Science and Engineering
PLA University of Science and Technology
Nanjing, China

CHEN Ming
Department of Compute Science and Engineering
PLA University of Science and Technology
Nanjing, China

LI Bing
Department of Compute Science and Engineering
PLA University of Science and Technology
Nanjing, China

***Abstract*—Large-scale network services bring great challenges to fault diagnosis, which include the unpredictability of the fault, the difficulty of collecting the fault information, high cost and technical complexity of diagnosis system deployment. To address these challenges, a web services-based distributed network fault diagnosis platform is built in this paper, which makes full use of the existing regional network management systems' resources. The platform integrates passive diagnosis and active probing techniques. Therefore, it can obtain more network information, which provides sufficient evidence for passive fault diagnosis, and can select optimal active actions to enhance the passive diagnosis whenever necessary. The experimental results show that our platform can efficiently and accurately localize the root causes of faults.**

Keywords-network fault diagnosis; distributed; web services; platform

I. INTRODUCTION

With the rapid growth of network applications, fault management is playing a crucial role in network service. Moreover, fault diagnosis is the most critical component in fault management since it can identify the root causes that can best explain observed network disorders [1-2]. Passive diagnosis [2-3] and active probing [4-6] are two commonly used fault reasoning and localization approaches. Owning to the facts which include system scalability, fault unpredictability and symptoms irregularity in a large-scale network, both the existing passive and active fault localization approaches are insufficient for efficient and accurate fault diagnosis. The rest of the paper is organized as follows. Distributed network fault diagnosis platform model is put forward in Section II, in Section III architecture of the platform is introduced, and in Section IV the fault diagnosis process of the platform is described. Experiments that validate our platform will be shown in Section V. Finally, we conclude our main works in Section VI.

It is faced with high cost, technical complexity, and other problems that deploy a dedicated network fault diagnosis system for large-scale network service. To take a massively multiplayer online game for an example, the game operator usually only regards the situation and quality of the local network in which the servers are, and it is helpless with the network fault and abnormity which the players are up against. Although the user experience is an important criterion when evaluating the quality of network service, apparently they are not willing to deploy a network-wide fault diagnosis system specifically for the players. Therefore, a platform which can take full advantage of the existing regional network management systems to building a large-scale distributed network fault diagnosis platform is needed, in which the management resources are shared and the passive diagnosis and active probing techniques are integrated.

II. DISTRIBUTED NETWORK FAULT DIAGNOSIS PLATFORM MODEL: AN OVERVIEW

Firstly, the method to build the distributed network fault diagnosis platforms based on the existing regional network management systems for large-scale network services is studied in this paper. Secondly, a distributed network fault diagnosis platform is put forward based on this method, which supports exchange of information. This makes the platform can obtain more network information, provide more evidence for passive fault diagnosis, and select optimal active actions to improve the diagnosis accuracy whenever necessary. It is faced with the following problems that building distributed network fault diagnosis platforms:

- How to exchange and share the observation of the regional network management systems?
- How to communicate and collaborate among the systems?

© 2012. The authors - Published by Atlantis Press

- How to define the fault management resources?
- How to integrate the passive diagnosis and active probing techniques to improve the efficiency and accuracy of fault diagnosis?

Web services technology provides good support to solve the above problems, which is based on SOA (Service-Oriented Architecture), and supplies services through the standard web protocols to ensure interoperability of heterogeneous applications [7]. Web services are described by WSDL (Web Service Description Language), which affords clear specification for publishing and invoking services. Therefore, the network management resources and fault information can be described in a web service-based way and encapsulated in web services, then, a distributed network fault diagnosis platform can be build with web services technology.

Figure 1 describes the web service-based distributed network fault diagnosis platform model. The platform mainly contains two parts: many Managers and one Management Center. The fault management resources are encapsulated in web services and are published to the management center by the managers of the regional network management systems. As defined in the web services architecture, the managers are service requesters as well as service providers, and the management center is the service registry [8].

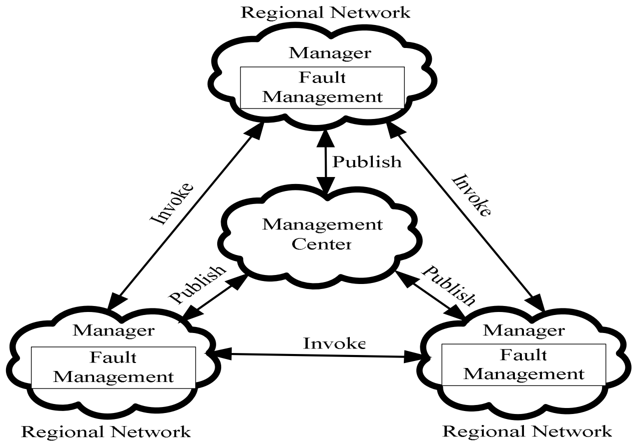

Figure 1. Distributed network fault diagnosis platform model

III. PLATFORM ARCHITECTURE

A. The architecture of the platform

Based on the platform model described in Figure 1, Figure 2 shows the architecture of the web services-based distributed network fault diagnosis platform, which is designed in detail from the manager and the management center.

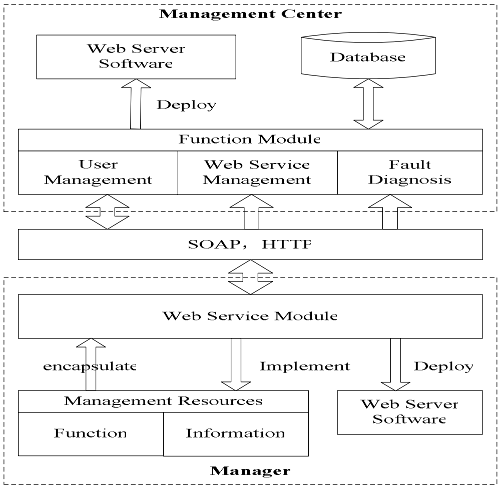

Figure 2. Architecture of the web services-based platform

The Manager contains Management Resources module, web server software and web services module. The fault management resource module provides fault management information and functions. The web server software, such as Internet Information Service (IIS) or Apache, Tomcat and so on, provides support for the web applications. The web services module, deployed on the web server software, includes various fault management resources, which are encapsulated in web services and can be invoked by other managers via standard protocols such as SOAP and HTTP.

The Management Center mainly plays a role as service registry in the web services system. It is the resource and service center of the platform, as well as the organizer and coordinator of the managers and the network management system. The management center includes the web server software, database and function module. The web server software is used to deploy the function module. The database is used to store user information, network status information and a variety of web service information. The function module includes three sub-modules: user management, web services management and fault diagnosis. The management center is a logical center, can be deployed in any location in the network. Considering the reliability and robustness of the platform, several mirror sites of the management center can be deployed in the network and synchronize with each other through the database maintenance.

B. Implementation of platform architecture

Based on the design of platform architecture, the key technical problems in implementing platform architecture are discussed from the following two aspects.

1) Encapsulating the fault management resources in web services.

Before sharing resources among the managers, the resources should be transformed into global resources in the platform. Therefore, the first step for implementing the distributed network fault diagnosis functions is encapsulating the fault management resources in web services.

The fault management resources of the manager contain network performance information and fault management functions. Network performance information includes a variety of network anomalies and fault information. The information can be encapsulated in web services directly and invoked by other managers in real time, just like invoking the network weather forecasting services.

The mode of invoking the web services directly and obtaining the results at once is not appropriate for fault management functions, which need to modify the structure and implementation of functions as web services. The right way is to regard the web service as the channel of invoking the functions rather than the implementation of the functions; the management systems receive the web service request and then pass the request to the function module. The web services play the role as a middleware in this process.

2) Users and web services management: Users and web services management works as the service registry in the web services system and contains the following functions.

a) Manager registration and certification: The managers can join the platform by registering in the user management module, and the management center certifies the managers' identity, then, the managers are allowed to take action on the management center, such as publishing web services, querying as well as invoking web services, and so on.

b) Publishing Web services: After registration, the managers can publish web services to the management center. In this process, the managers need to describe and explain the web services so that other mangers can find and invoke them.

c) Querying and invoking web service: The user can query the web services on the management center. The platform offers two search methods: directory querying and conditions querying. The managers can browse the service directory tree to find the web services step by step, in accordance with the hierarchical classification of the web services category and the location of the web services. The managers can also input some words as conditions to find web services. Then, the managers can send the web service requests to the manager whom the requested web services own to, invoke the web service and get the results.

IV. WORK PROCESS OF DISTRIBUTED NETWORK FAULT DIAGNOSIS

Distributed network fault diagnosis capabilities of the platform are provided by the fault diagnosis module of the Management Center. Inspired by the fault diagnosis method proposed by Tang et al [9], the fault diagnosis module is a coarse-grained fault diagnosis technology framework and can seamlessly integrate passive and active monitoring techniques into one framework. Firstly, the module analyzes the observed symptoms for a coarse grained diagnosis to rapidly narrow down the fault search scope in a large-scale network. Secondly, it conducts a passive heuristic fault reasoning with finer granularity to pinpoint the root faults that highly likely occurred. Since the passive diagnosis may not be sufficient, a set of optimally selected probing actions are conducted to collect the most relevant but unobserved symptoms. The feedback from active probing can be incrementally integrated with the previous reasoning results until the confidence on the diagnosis results is satisfactory.

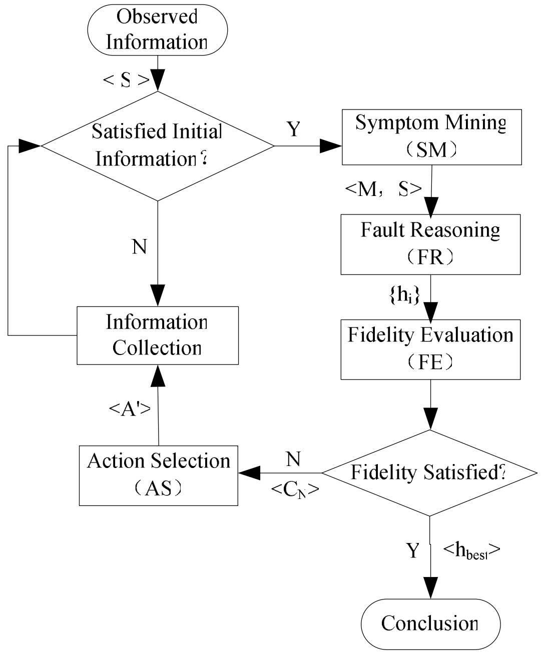

Figure 3. Fault diagnosis process

Figure 3 describes the fault diagnosis process, which mainly consists of four modules: Symptom Mining (SM), Fault Reasoning (FR), Fidelity Evaluation (FE), and Action Selection (AS).

1) Symptom Mining

When the managers discover any anomalies in a network, it first reasons the fault root through its own observation information. If the information is sufficient to localize the root causes of faults, the fault diagnosis process terminates. Otherwise, the managers invoke the web services of other managers to get more network symptoms related. We denote each observed symptom as S_i, and find out the corresponding network path, which is marked by the unique path identifier (UPI). An UPI consists of a sequence of network components (e.g., network nodes or regional network system), and each component is denoted as C_i, then each symptom S_i can be represented by a sequence of components as $\{C_1, C_2, \ldots, C_n\}$. We define *Maximum Identifiable Component* (MIC) as the longest common component shared by the maximum number of symptoms.

Denote each element in MIC as m_i and the probability of m_i as $p(m_i)$ that indicates the inherent likelihood that m_i becomes faulty. The probability $p(m_i|S_i)$ provides critical information on the likelihood measure that m_i is faulty when observing the symptom S_i. We assume the occurrence of any given symptom S_i is triggered by at least one faulty component m_k. For a given m_i, $p(m_i|S_i)$ can be caculated.

2) Fault Reasoning

Given all the $p(m_i|S_i)$, the next task is to find the most likely root causes that can best explain all observed symptoms. A new parameter called *Symptom Contribution* (SC)[9] is introduced in this paper:

$$SC(mi) = \frac{1 - \prod_{Si \in S_{mi}} (1 - p(mi|si))}{|mi|} \qquad (1)$$

The value of SC can arbitrarily vary in (0, 1), it can be used to evaluate the correlation likelihood between m_i and a set of observed symptoms Sm_i. $SC(m_i)$ evaluates the possibility that we can diagnose faults with the given symptoms. The higher $SC(m_i)$is, the more likely m_i is faulty if Sm_i observed. For each m_k, if $SC(m_k) = max_{mi \in M}\{SC(m_i)\}$, then m_k could be put in the fault hypothesis set Φ, which may contains many fault hypotheses(h_1, h_2,…, h_n). Each hypothesis includes a set of fault components and attempts to explain the observed symptoms.

3) Fidelity Evaluation

The task of the fidelity evaluation is to measure the fidelity of the hypothesis created in the passive reasoning phase. In order to objectively evaluate the reasoning result, we design a fidelity function $FD(h)$[9] to measure the fidelity of hypothesis h, which essentially measures the likelihood of hypothesis h when S observed. We assume that the occurrence of each faulty component is independent.

$$FD(h) = \prod_{mi \in h} (1 - \prod_{si \in S} (1 - p(mi \mid si))) \qquad (2)$$

The managers can define a fidelity threshold based on their relatively long term observation and previous experience. Generally, the fidelity threshold would exceed 50%. The fidelity evaluation function is used to evaluate each hypothesis and decides if a hypothesis is satisfactory. If the best hypothesis (i.e., the highest fidelity), whose fidelity exceeds the fidelity threshold, the fault diagnosis process terminates. Otherwise, the best available hypothesis and a non-empty set of components (C_N) will be verified via the selected probing actions to find a satisfactory hypothesis in the next iteration.

4) Action Selection

The probing action results can directly collect more relevant symptoms that can be used to gather more evidence to increase SC of the observed symptoms, or to verify the correctness of a given hypothesis. Every action, denoted as A_i, has its own coverage, which is a set of involved components. If A_i returns a positive result, it indicates that all components are in positive status. Otherwise, if A_i returns a negative result, it returns a negative symptom showing that at least one related component is in negative status. Every action also has its cost that can be measured differently. In this paper, we simply use the number of hops.

The Action-Component correlation graph can be represented as a 3-tuple (A, C, E). Given a set of components $C = \{C_1, C_2,…, C_N\}$, a set of actions $A = \{A_1, A_2,..., A_M\}$, and the coverage of each action denoted as: $A_i = \{C_x, C_y,…, C_z\}$($Ai \in A$), every edge e in E connects a vertex A_j with a vertex Ci with a corresponding weight (w_{ij}) to denote that A_j can verify Ci with cost $Wij = W(Ci, Aj)$. The action selection module is to find a set of actions A' such that: $\forall Ci \in C$, $\exists Aj \in A'$, $W_{ij} > 0$; and $\sum_{Ai \in A', Cj \in C} Wij$ is the minimum.

V. EXPERIMENT AND ANALYSIS

1) Experimental Environment

The experimental environment of the distributed fault diagnosis platform is shown in Figure 4.

The regional networks N_a, N_b, N_c, N_d and N_e are connected together via network nodes R_A and R_B. M_1, M_2 and M_3, running the Windows XP operating system, are deployed in N_c, N_d and N_e, are the mangers of network management system, and have been incorporated into the distributed network fault diagnosis platform, they can get network symptom information from other network management systems. n_1 and n_2, running the Windows 2003 operating system, are deployed in N_a and N_b respectively and are the servers running web applications. Without the support of distributed network fault diagnosis platform, M_1, M_2 and M_3 can only cover the local network N_c, N_d and N_e, obtain the local network information, as well as detect the health of the local network, but be unable to get the situation of the external network.

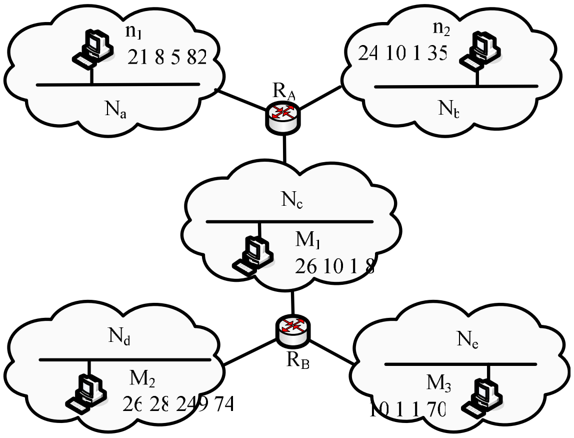

Figure 4. Experimental environment

2) Experimental process and results analysis

Based on the experimental environment described in Figure 4, we design the following experiment. At some particular time, the users in N_d find that the connection to n_1 is abnormal; at the same time, the users in N_e find that the connection to n_1 is abnormal. Then, the Fault diagnosis process begins with the fidelity threshold set to 50%.

a) Symptom Mining

There are two observed symptoms, which can be denoted as S_1 and S_2. The regional networks N_a, N_b, N_c, N_d and N_e can be regard as the components of the large-scale network. Then, the network paths related to S_1 and S_2 are denoted as the sets $\{N_d, R_B, N_c, R_A, N_a, n_1\}$ and $\{N_e, R_B, N_c, R_A, N_b, n_2\}$, accordingly, the MIC, M_{S1} and M_{S2} are described as the following:

$$MIC(S_1, S_2) = \{(C_d); (C_e); (C_B, C_c, C_A); (C_a); (C_b); (C_1); (C_2)\}$$
$$= \{m_1, m_2, m_3, m_4, m_5, m_6, m_7\}.$$

In the experiment, we can't get the probability of m_i, so we set the value of each $p(m_i)$ to 15%, then we can get that $p(m_i|S_i)$=0.15 for each m_i and S_i.

b) Fault Reasoning and Fidelity Evaluation

According to equation (1), we can get that $max_{mi\in M}\{SC(m_i)\} = SC(m_3) = 0.04$, and the hypothesis set $\Phi = \{m_3\}$. Then we can get that $FD(m_3)$=28% according to equation (2), the value is too small to exceed the fidelity threshold , so that the probing actions are necessary to get more symptoms and information.

3) Action Selection

Based on the set of components $C = \{C_B, C_c, C_A\}$, which related to m_3, the following actions can be selected according to Figure 4:

$$A_1 = \{M_1, N_c, R_A, N_a, n_1\},$$
$$A_2 = \{M_1, N_c, R_A, N_b, n_2\},$$
$$A_3 = \{M_2, N_d, R_B, N_c, M_1\},$$
$$A_4 = \{M_2, N_d, R_B, N_c, R_A, N_b, n_2\},$$
$$A_5 = \{M_3, N_3, R_B, N_c, M_1\},$$
$$A_6 = \{M_3, N_e, R_B, N_c, R_A, N_a, n_1\}.$$

According to a heuristic greedy set-covering approximation algorithm for the weighted set-covering problem [10], we get the initial set of actions $A_1' = \{A_4\}$ or $A_2' = \{A_6\}$. Considering that both of them contain only one action and the initial hypothesis set $\Phi = \{m_3\}$, the feedback from A_1' and A_2' is clearly not conducive to narrow down the fault search scope. Therefore, $A_3' = \{A_1, A_3\}$ can be selected.

4) Experiment results

Both A_1 and A_3 return negative results, the symptoms are denoted as S_3 and S_4, then we get that $max_{mi\in M}\{SC(m_i)\} = SC(m_3) = 0.16$, the hypothesis set $\Phi = \{m\} = \{(N_c)\}$, and $FD(m)$=56%, which exceed the fidelity threshold, so the fault diagnosis process terminates. Through the detection and investigation, we find that the reason of the network anomaly is that the administrator of the network N_c temporary shut the export to the R_A and R_B.

5) Experiment analysis

Though we can't get the organization and topology details inside of each regional network and don't clear about the actual failure rate of each regional network and node, we narrow down the fault scope and provide the possibility to find out the root causes of fault in support of the characteristics observed initially in combination with corresponding active probing action.

In the large-scale networks, we can estimate the fault rate of some node or component based on long-term network observation information and experience. In the actual network environment, the infrastructure of network is usually stable. We often cannot obtain the earlier fault rate of a network component in troubleshooting, and then we could set a smaller value to the fault rate for the infrastructures of network comparing to the nodes of network application, that helps to localize the root causes of faults quickly and reduce the active probing actions.

VI. Conclusion

Effective network fault management is critical to ensure the quality of network service in large-scale distributed network applications. But the deployment of large-scale network fault diagnosis system is faced with the problems of technical complexity and high cost which bring difficulty to the fault management in large-scale network. This paper presents a web services-based method for building the distributed network fault diagnosis platform, which makes full use of the fault management resources of existing regional network management systems to obtain more network symptoms, offers multi-faceted and multi-angle probing actions, as well as provides good support to integrate the passive diagnosis and active detection techniques so that helps to narrow down the fault scope quickly and localize the root causes accurately. The platform maintains the function structure of original network management systems in the greatest degree when utilizing various network resources, and takes on excellent scalability and versatility with low cost.

References

[1] M. Steinder, A. S. Sethi. Increasing robustness of fault localization through analysis of lost, spurious, and positive symptoms . In Proc. of IEEE INFOCOM, New York, NY, 2002.

[2] M. Steinder, A. S. Sethi. Probabilistic Fault Diagnosis in Communication Systems Through Incremental Hypothesis Updating. Computer Networks, vol. 45, 4 pp. 537-562, 2004.

[3] K. APPLETY et al. Yemanja - a layered event correlation system for multi-domain computing utilities, Journal of Network and SystemsManagement, (2002).

[4] I. Rish, M. Brodie, N. Odintsova, S. Ma, G. Grabarnik. Realtime Problem Determination in Distributed Systems using Active Probing [C]. IEEE/IFIP (NOMS), Soul, Korea, 2004.

[5] M. Brodie, I. Rish, S. Ma. Optimizing Probe Selection for Fault Localization. IEEE/IFIP (DSOM), 2001.

[6] J. Guo, G. Kar, P. Kermani. Approaches to Building Self Healing System using Dependency Analysis. IEEE/IFIP (NOMS), Soul, Korea, 2004.

[7] B. Jeong, D. Lee, J. Lee, et al. Support for seamless data exchanges between Web services through information mapping analysis using kernel methods. Expert Systems with Applications, vol. 36, 1 pp. 358-365, 2009.

[8] R. Aboolian, Y. Sun, G. J. Koehler. A location- allocation problem for a Web services provider in a competitive market. European Journal of Operational Research, vol. 194, 1 pp. 64-77, 2009,.

[9] Yongning Tang, Guang Cheng, and Zhiwei Xu. Probabilistic and Reactive Fault Diagnosis for Dynamic Overlay Networks. Peer to Peer Networking and Applications, vol. 4, 4 pp. 439-452, 2011.

[10] Qiang Zheng, Guohong Cao. Minimizing Probing Cost and Achieving Identify ability in Network Link Monitoring. IEEE ICDCS, June 2010.