Review On Fault-tolerant Mobile Agent-based Mechanism for Distributed Networks

[1]S.Haritha, [2]L.Raghavender Raju, [3]D.Jamuna

1.INTRODUCTION

ABSTRACT

Thanks to asynchronous and dynamic natures of mobile agents, a certain number of mobile agent-based monitoring mechanisms have actively been developed to monitor large-scale and dynamic distributed networked systems adaptively and efficiently. Among them, some mechanisms attempt to adapt to dynamic changes in various aspects such as network traffic patterns, resource addition and deletion, network topology and so on. However, failures of some domain managers are very critical to providing correct, real-time and efficient monitoring functionality in a large-scale mobile agent-based distributed monitoring system. In this paper, we present a novel faulttolerance mechanism to have the following advantageous features appropriate for large-scale and dynamic hierarchical mobile agent-based monitoring organizations. It supports fast failure detection functionality with low failure-free overhead by each domain manager transmitting heart-beat messages to its immediate higher-level manager. Also, it minimizes the number of non-faulty monitoring managers affected by failures of domain managers. Moreover, it allows consistent failure detection actions to be performed continuously in case of agent creation, migration and termination, and is able to execute consistent takeover actions even in concurrent failures of domain managers.

Keywords: *Distributed Network, Fault-tolerance, Mobile Agent, Scalability, Takeover.*

Distributed Networking is a distributed computing network system, said to be "distributed" when the computer programming and the data to be worked on are spread out over more than one computer, usually over a network.

A mobile agent is a process that can transport its state from one environment to another, with its data intact, and be capable of performing appropriately in the new environment. Mobile agents decide when and where to move. Movement is often evolved from RPC methods. Just as a user directs an Internet browser to "visit" a website similarly, a mobile agent accomplishes a move through data duplication. When a mobile agent decides to move, it saves its own state, transports this saved state to the new host, and resumes execution from the saved state. A mobile agent is a specific form of mobile code, within the field of code mobility. However, in contrast to the Remote evaluation and Code on demand programming paradigms, mobile agents are active in that they can choose to migrate between computers at any time during their execution. This makes them a powerful tool for implementing distributed applications in a computer network.

Mobile agent-based monitoring mechanisms are classified as follows: centralized and hierarchical distributed monitoring mechanisms. Centralized monitoring model and divided into two categories, single mobile agent-based and segment-based mechanisms. A single management station creates a mobile agent and allows the agent to sequentially visit the required

nodes in a particular order. In segment based collection and filtering of the management information for monitored nodes can be performed in parallel per domain.It is very difficult to perform the procedure to obtain and filter the monitoring information in real-time called as scalability problem.

2. STATIC CENTRALISED MONITORING

In this case there is a single monitoring station with which all the monitored systems communicate directly. The monitoring station is in charge of collecting, aggregating and processing raw network data.

This model is widely used to manage relatively static, small networks using (simple network management protocol) SNMP. The model has been criticized for its limited responsiveness, accuracy and lack of scalability. The concentration of management intelligence in a single point results in processing and communication bottlenecks, limiting the number of elements that can be monitored and the rate at which information can be gathered. Furthermore SNMP favors a polling approach which limits the ability to track problems in a timely manner while requiring management traffic even if no significant change has occurred.

To overcome the shortcomings of polling, the alternate technique of event reporting may be used. With event reporting, the monitored systems take he initiative to inform the manager according to pre-determined rules set by the manager. Event reports are generated within the monitored systems either periodically or as and when an critical event occurs.

Periodic reporting provides the manager with status information in a summarized manner and is more efficient than requesting the same information via polling. On the other hand alarm reporting is used for detecting problems as soon as they occur. The problem with alarm reporting is that the types of alarms need to be thought-out in advance, standardized and supported by vendors. Event reporting requires an increased level of intelligence in the monitored systems.

Typical systems employ both polling and event reporting although in practice the telecommunications systems rely more on event reporting and SNMP-based management systems

3. STATIC DECENTRALIZED MONITORING

One way to increase performance and scalability is to adopt a hierarchical management architecture which uses multiple systems with one system acting as a main monitoring station and the others working as area monitors. Hierarchical monitoring is used in Telecommunications Management Network (TMN). In context of SNMP, simple monitoring and statistical probes can be introduced using RMON, which is equivalent to an area monitor that collects monitoring information about a number of elements within a sub-network. More recently, other forms of decentralization based on distributed object technologies such as CORBA and JAVA RMI have become popular in management. An extensive review of management paradigms and technologies can be found.

The common denominator of the above approaches in the adoption of simple, pre-defined functionality that can actually be decentralized is restrained to operations such as low-level filtering of monitoring data, generation of alarms on the basis of simple conditions, and collection of rudimentary statistical information. In addition, these decentralized area monitors operate in pre-defined network locations, which mean that they cannot easily adapt to network changes.

Therefore, conventional static decentralized schemes, despite coping with the scalability problem to a certain extent, inherit the other problems of centralized management and cannot easily cope with frequently changing, dynamic environments.

So far we have been discussing the operations and problems concerned with the static systems. Hence to overcome the cons of static systems, we are looking into the dynamic aspect of methodologies of which the programmable decentralized monitoring poses some shortcomings not satisfying the requirements.

4.IMPLEMENTATION

To solve the scalability problem, mobile agent-based mechanisms using hierarchical monitoring structure were proposed. They allow a network to be partitioned into a set of domains organized hierarchically and deploy a new monitoring agent to each domain. In this hierarchy, a main manager is at the top-level (level 1) and delegates monitoring tasks with monitoring agents to the lower level domain managers. Each manager clones and dispatches a monitoring agent to the appropriate domain manager node considering load redistribution of monitoring tasks. In this case, each domain manager collects the management information from the lower-level managers and filters and delivers the processed information to its higher-level manager. The original hierarchical monitoring mechanisms were almost based on a static manager organization model. In other words, each network administrator configures a tree of network domains according to its initial monitoring policy and then the main manager at the root domain creates and migrates monitoring manager agents to other domains.

DATA STRUCTURES :

AIDα: it is the agent identifier of domain manager α.

MMaddrα: it is the main manager's identifier needed when domain   manager α is created or the organization of its lower-level managers changes.

IHMaddrα: it is the immediate higher-level manager's   identifier   of   domain manager α.

ptrα: it is the root node of a tree for saving the identifier and timer of every lower-level manager of main or domain monitoring manager α. Its node is a tuple (aid, tinterval, ptr). tinterval for each lower-level manager aid is used so that monitoring manager α detects whether its lower-level manager aid is alive or failed, and is initialized to τ . ptr for its lower-level manager aid is the next-level node maintaining references for all lower-level managers of the domain manager aid in a hierarchical manner.

Every domain manager α periodically transmits each heartbeat message only to its immediate higher-level manager IHMaddrα. Therefore, each monitoring manager can know which ones fail or are alive among its immediate lower level managers by their periodic notification. In our mechanism, the manager α decrements the timer-t interval for its corresponding immediate lower-level manager aid in ptr α by one every certain time interval. If α has not received any heart-beat message from the lower-level manager until the timer expires, it suspects that the lower-level manager crashes. This behavior results in low failure-free overhead incurred by failure detection by utilizing the tree-like organization of monitoring managers effectively.

5. CONCLUSIONS

This paper presented a novel fault-tolerance mechanism to have the following advantageous features appropriate for large-scale and dynamic hierarchical mobile agent-based monitoring organizations. It supports fast failure detection functionality with low failure-free overhead by each domain manager transmitting heart-beat messages to its immediate higher-level manager. Also, it minimizes the number of non-faulty monitoring managers affected by failures of domain managers. Moreover, it allows consistent failure detection actions to be performed continuously in case of agent creation, migration and termination, and is able to execute consistent takeover actions even in concurrent failures of domain managers

REFERENCES

1.Fault-tolerant Mobile Agent-based Monitoring Mechanism for Highly Dynamic Distributed Networks by **Jinho Ahn**

2. Mobile Agent Based Distributed System Computing In Network by Prof. S. N. Gujar, Prof. G.R.Bamnote , Prof. R.S.Apare , Prof. M.A.Pund , Mr. S.R.Gupta.

3. New Mobile Agent-Based Intrusion Detection Systems For Distributed Networks by Pranita Jain, Sandeep Raghuwanshi And Pateria Rk

S.Haritha Pursuing M.Tech from Jaya Prakash Narayan college of engineering Mahabubnagar, Andrapradesh . Her areas of interest include study in mobile adhoc networks.

Mr. L.Raghavendra Raju B.Tech(CSE),M.Tech(CSE) Associate Professor in Jaya Prakash Narayan College of Engineering, mahabubnagar, Andrapradesh. His areas of interest include Wireless networks, Information Security currently focusing on IP Networks .

Prof.D.Jamuna ,M.Tech, (,Ph.D). Professor & HOD. At Jaya Prakash Narayan College of Engineer ing, mahabub nagar, Andra pradesh. M.Tech. degree in SE from School of Information Technology, JNTU, Hyd and Pursuing Ph.D from Rayalaseema University, Kurnool. Her areas of interest include Wireless networks, Information Security currently focusing on IP Networks