An Analogous Study of Fault Tolerance Techniques in Mobile Agents

Preeti
(Department of Computer Science
Gurukul Kangri Vishwavidyalaya, Uttarakhand, India
preetishivach2009@gmail.com)

Praveena Chaturvedi
(Department of Computer Science
Gurukul Kangri Vishwavidyalaya, Uttarakhand, India
praveena_c1@rediffmail.com)

ABSTARCT: The Internet has make innovation the computer and communications world like nothing before. The explosive growth of Internet as a medium for communication, business and e-commerce has stimulated the flourishing interest in agent-based systems. A mobile agent is a type of software system which acts on one's behalf with the feature of autonomy, learning ability and most importantly mobility. For the mobile agent's technology to survive, it is necessary that the mobile agents should be reliable. This paper evaluates fault tolerance techniques used in mobile agents. These techniques are analyzed on the basis of some defined criterion.

I. INTRODUCTION

The Internet is at once a world-wide broadcasting competence, a mechanism for information propagation, and a medium for collaboration and interaction between individuals and their computers without regard for geographic sites. The Internet represents one of the most successful examples of the benefits of sustained investment and commitment to research and development of information infrastructure. The explosive growth of Internet as a medium for communication, business and e-commerce has fostered the growing interest in agent-based systems [1]. A mobile agent is a type of software system which acts "**intelligently**" on one's behalf with the feature of autonomy, learning ability and most importantly mobility [2]. There are seven reasons for using mobile agents; these are mobile agent reduce network load, overcome network latency, encapsulate protocols, execute asynchronously and autonomously, adapt dynamically, naturally heterogeneous and robustness and fault-tolerant. Mobile Agent application areas include network management, remote software management, electronic commerce, distributed information retrieval etc. There are no such applications; instead there are application that benefit from using mobile agents [3].

In mobile agent computing paradigm not only data but the code acting on that data is also transported among the hosts. This makes the applications more flexible. A mobile agent can stops its execution at a point (when it is necessary to migrate or to access the services available at another server), saves its state then migrate itself to another server or node and resumes its execution from the point where it was stopped. So migration of code, data and state is done in mobile agent paradigm.

There are various issues related to mobile agent paradigm. The main are reliability, security and of course mobility. In this paper we are focusing fault tolerance of mobile agent, which comes under reliability issues. Reliability issues are like an agent should not fail due to any failure in software or hardware components. Agent can fail if host fails or agent might not reach the desired server. These failures may lead to a partial or complete loss of the agent. So the fault tolerant mobile agent system should be created.

In this paper, we are performing a comparative analysis of existing fault tolerance techniques used in mobile agent and mobile agent systems. The comparison is done on the basis of some parameters like type of failure handled, agent type, failure detection and healing mechanism etc. The rest of the paper is organized as follows: next section gives an introduction to fault tolerance in mobile agents; section 3 defined the parameter based on which comparative analysis is conducted; in section 4 analyses the techniques and finally in the last section we conclude the paper.

II. FAULT TOLERANCE IN MOBILE AGENT

In mobile agent computing environment any component of the network machine, link, or agent may fail at any time, thus may preventing mobile agents from continuing their executions. Therefore, fault-tolerance is a vital issue for the deployment of mobile agent systems. The main goal of an agent fault tolerant protocol is to detect communication or node failures and recover lost agents [4]. Fault-tolerant mobile agent execution removes this uncertainty and ensures that the agent eventually reaches its destination or at least notifies the agent owner of a potential problem. Many techniques have been developed to add reliability and high availability to distributed system which can be broadly classified into two kinds: replication and check pointing.

In checkpointing, the snapshot of the agent state is saved on non-volatile storage, so that whenever failure occur in the system the agent can resume its execution from the saved state. The latest checkpoint is used to restore the agent. The check pointing scheme also shows a very stable performance and it is sensitive only to the blocking time when the failure rate is high [5]. For recovery a mobile agent need to rollback to its consistent state. Message logging for rollback recovery requires that the agent saves its state time to time and log the sent and received message [6].

Replication is a kind of redundancy. Replicated servers are used to tolerate the fault. When a server is down, we can use another server which provides the same services and continue the computation. In this way computation is not blocked when a server failure occurs. However it is not cost-effective to use multiple physical servers for a single logical server. On the other hand, the data on the replicated server must be consistent. Thus it is not easy to preserve data consistency on the servers, when the servers are widely separated or agents in a group communicate to each other to provide fault tolerant agent execution. If failure occurs, they take appropriate action to recover from fault. Replication avoids the blocking problem. Adding redundancy masks failures and enables the agent to continue the execution despite failures [7].

III. PARMETER RECOGNITION

After taking a step towards a small survey of existing fault tolerance techniques, we found some common parameters. A comparative analysis of techniques is performed on the basis of these parameters. In this section, we present the identified parameters and their definitions.

A. Failure: Failure may be an agent failure, node failure, fault of component of the agent system, communication failure or network failure. This parameter shows that a particular technique deals with what type of failure or failures.

B. Failure Discovery Mechanism: This parameter shows that what mechanism is used to detect the agent failure. It is very important factor which should be considered, especially if the agent execution should retain the exactly-once property of the agent. Because, if we mistakenly detect a slow communication or link as an agent failure. The replicated agent gets executed, which violate the exactly-once property of the agent. The two main property of mobile agent are: non- blocking and exactly-once execution.

C. Failure Healing Mechanism: Error recovery is performed using backward recovery or forward recovery. In backward recovery, the system is restored or rolled back to a previously saved state. On the other hand, the forward recovery mechanism attempts to do this by finding a new state from which the system can continue operation. This state may be a degraded mode of the pervious error-free state. Redundant software processes are executed in parallel.

D. Administration: It tells that whether the mechanism uses centralized administration or not. It is an important factor from implementation point of view and the cost also.

E. Adaptivity: It shows that whether the technique allows mobile agent to change its behavior in response to change in its operating environment [8]. If it is, then this makes the system more flexible at cost of complexity and of course extra effort.

F. Coordination: This shows that whether the coordination is direct or indirect. Because direct coordination takes less communication overhead and comparably less complex then the indirect one. So it is also an important factor in analyzing the fault tolerance techniques.

IV. INVESTIGATION OF TECHNIQUES BASED ON IDENTIFIED PARAMETERS

Optimistic Replication is based on checkpointing, chain control and message passing to detect and recover failed agent [7]. Author does not guarantee of perfect failure discovery. This approach tolerates agent failure, place failure and also network partition. It is also non-blocking protocol. Failure administration is not centralized; discovery is done by witness agent by heartbeat messages. Coordination between these two types of agent is direct. Performance is not measured by the author.

Transient faults in mobile agents, [9] use enhanced single version programming to cope with agent failure. No centralized administration, agent coordinates transparently and directly with its replicas to tolerate bit errors. Can be blocking if bit error cannot be recovered by any of the replicas but probability of the situation is very small. Technique provides high performance as provide fault tolerance at low level.

Witness Agent Approach [10] copes with failure by using a special kind of agent. This agent is a copy of agent called witness agent. Actual agent and the witness agent communicate by passing messages.

It deals with agent failure, server failure, network failure and loss of messages also. Author guarantees that all failures are detected and recovered. Discovery of failure is done by witness agent and recovery is message log based recovery and also checkpointing. Coordination is direct and indirect. Performance of the technique is high. Fault administration is centralized.

Using Checkpointing for Information Retrieval, In [11] agent put its computation results on the home server (the server that originates the agent) after completing its task on first three servers in its itinerary. If agent stops its execution due to any fault on the server and unable to move in its itinerary, Agent sends a message to home server, which then sends the replicated copy of the original agent to the immediate checkpoint before the faulty server. Failure administration is centralized. Recovery is backward because after fault occurrence the system is in a prior state. The author measures the performance of the approach.

Using Group Communication Based Services, In [12] agents in a group communicate to give reliable service. Only the leader of the group performs computation and broadcast result to other members of the group. Author considers two scenarios; group leader crash and non-leader member of the group crash. Administration is not centralized. Discovery is done via check-alive messages. Coordination is direct. Recovery mechanism is forward. Performance is measured. Group on the previously visited server is deleted when the agent migrates from one server to another.

Adaptive Mobile Agent System Using Dynamic Roll Based Access Control, in this approach [13] mobile agents are assigned roles to perform a task. This is done by role generator component. When an agent failure occurs, other agents share the task of failed agent.

Failure administration is not centralized. Agents coordinate to recover from failure. This approach is good from performance point of view. Recovery is done by sharing the failed agent load by other agents. This technique deals with permanent failure. And as the name indicate the mobile agent is adaptive because the roles of mobile agent changes dynamically when a failure occurs.

Exception handling for mobile agents copes with agents and nodes failure. Two exception handlers are proposed. The first exists at the agent server that created the mobile agent. The second operates at the previous agent server visited by the mobile agent [14]. Author uses both backward and forward recovery. Coordination among the replicas of the agent is direct. Author claims it to be highly dependable and efficient technique.

Generic Framework proposed by Bassey et al, is a hierarchical approach for mobile agent fault tolerance. It constitutes structure of operation that is used to plan, decide and execute fault tolerance mechanism [15]. The execution phases include Monitor/Detection, Action Planning, and Execution of action and continue service Phase. These phases are abstraction of the traditional fault tolerance framework in mobile agents and apply to any fault tolerance mechanism in mobile agent execution at any specific stage in the agent's itinerary.

This framework can help tolerate failure of agent, node, communication and component. Failure administration is distributed. It is not implemented yet, from performance point of view we can't say whether it is good or not. No direct communication and coordination among agents. System can be blocking based on recovery mechanism used.

Fault Tolerance in Cooperative Mobile Agent Systems, author in uses three types of agents which are actual agent, supervisor agent and the replicas. Failure detection and recovery is done by message passing mechanism among these three kinds of agents [16]. Administration is centralized. Permanent failures are considered. Recovery mechanism used is of both type replication and checkpointing. According to the author this method in system with high needed to reliability have better performance than existing methods.

A Fault Tolerance Infrastructure for Mobile Agents, the proposed algorithm is similar to sliding window model [17]. Agents of different types collaborate to provide fault tolerant behavior. Five agents Transaction Manager, Observer Agent, Ping Agent, Transaction Agent and Statistical Agent are the backbone of the infrastructure. Approach is non-blocking as there is always observer agent, which is not on the same server or node. Crash failure, communication failure, slow processor and agent failure are handled. Administration is not centralized. Clones are used to take up responsibility of failed agents. Performance analysis is done by author but not compared by other approaches.

CAMA supports system fault tolerance through exception handling and structured agent coordination. It deals with application level faults. Exceptions are propagated between agents in reliable and secure way by propagation mechanism [18]. It handles agent and connection disconnection problems. Coordination is direct between the agents. The approach can be blocking. According to author it provides high performance and dependable framework.

FANTOMAS uses Logger Agent to provide fault tolerant system. It does not participate in any computation, so needs only small fraction of CPU capacity [19]. Logger Agent is always at the server which is just visited by the user agent (actual agent which perform computation). Both the agents monitor each other and whenever there is a fault in one, it is rebuild by the other one from its local information. This approach deals with node and agent failures. Administration is not centralized. System is not blocked if failure occurs. It is an efficient technique according to the author.

Rear Guard Agent Approach Johansen et al. detect and recover from faulty migrations inside the TACOMA agent system [20]. When an agent migrates, a *rear-guard agent* is created that stays on the origin node. It monitors the migrated agent on the destination node. This very simple concept does not tolerate network partitioning.

In the table given below, the analysis is represented in tabular form:

Table 1. Analysis of Fault Tolerance Techniques in Mobile Agents

Parameters Techniques	Failure	Discovery Mechanism	Healing Mechanism	Administration	Adaptivity	Coordination
[7]	Agent, Place and Network Partition	Witness Agent	Backward	No	No	Direct
[9]	Agent	Error Detection Mechanism (exception, memory protection mechanism, checksum)	Forward	No	Yes	Direct
[10]	Agent, Server, Network and Loss of Messages	Witness Agent	Backward	Yes	No	Direct
[11]	Agent and Server	Clone Agent	Backward	Yes	No	Indirect
[12]	Agent	Check-alive Message	Forward	No	No	Direct
[13]	Agent	Heart Beat Signal	Forward	No	Yes	Indirect
[14]	Agent and Node	Timeout and Shadow	Both	No	No	Direct
[15]	Agent, Node, Communication and Component	Flexible	Both	No	May be	Indirect
[16]	Agent and Server	Message	Both	Yes	No	Direct
[17]	Crash, Communication and Agent	Clone	Forward	No	No	Direct
[18]	Agent and Link	Timeout	Backward	No	No	Direct
[19]	Agent and Node	Logger Agent	Backward	No	No	Direct
[20]	Agent	Rear Guard Agent	Forward	No	No	Direct

V. CONCLUSION

Mobile Agents are becoming popular day by day with the growth of Internet. They change the trends of distributed computing. So mobile Agents are required to provide service even in the presence of failure that is the mobile agent should be fault tolerant. Prior to this paper we have gone through some fault tolerance techniques proposed for mobile agent. We have concluded that FANTOMAS and Optimistic Replication based Approach are more reliable and suitable from fault tolerance point of view. Both are based on backward recovery. Optimistic Replication approach can cater with permanent faults only. Both techniques do not use any centralized administration and uses direct communication among agents hence do not increase the complexity.

REFERENCES

[1] L. L. Kassab and J. Voas, "Towards Fault-Tolerant Mobile Agents".

[2] Y. Singh, K. Gulati and S. Niranjan, "Dimension and Issues of Mobile Agent Technology", International Journal of Artificial Intelligence & Application, Vol.3, No. 5, Sep. 2012.

[3] D.B. Lange and M. Oshima. "Seven Good Reasons for Mobile Agents", Communications of the ACM, Vol. 42, No. 3, pp. 88-89, 1999.

[4] R. Hans, R. Kaur, "Novel Dynamic Shadow Approach for Fault Tolerance in Mobile Agent Systems", IEEE, 2012.

[5] R. Hans, "Fault Tolerance Techniques in Mobile Agents: A Survey", International Conference on Computing and Control Engineering, 12 & 13 April, 2012.

[6] Preeti, P.Chaturvedi, "Fault Tolerance in Mobile Agents: An Overview", Second International Conference on Recent Trends in Computing, India, Oct, 2013.

[7] Z. Linda and B. Nadjib, "Optimistic Replication Approach for Transactional Mobile Agent Fault Tolerace", Proceeding of 11th ACIS, International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2010.

[8] F. Yousuf, Z. Zaman, "A Survey of Fault Tolerance Techniques in Mobile Agents and Mobile Agent Systems", 2009 Second International Conference on Environmental and Computer Science, 2009.

[9] G. K. Saha, "Transient Fault Tolerance in Mobile Agent Based Computing", 2005.

[10] M. R. LYU and T. WONG, "A Progressive Fault Tolerance Mechanism in Mobile agent System".

[11] Rahul Hans, Ramandeep Kaur, "Fault Tolerance Approach in Mobile Agents for Information Retrieval Applications Using Check Points", International Journal of Computer Science & Communication Networks, Vol2(3), 347-353, 2012.
[12] Alaa Eldeen Sayed Ahmed, Rania Ramadan Abd El-dayem, "$62.5 US The Modeling and Implementation of Reliable Mobile Agent Systems using Group Communication Services", International Journal of Computer Application (0975-888) Volume 48- No. 6, June 2012.
[13] P. Marikkannu, J. J. Adri Jovin and T. Purusothaman, "Fault-Tolerance Adaptive Mobile Agent System using Dynamic Role based Access Control", International Journal of Computer Application Vol 20, No. 2, April 2011.
[14] S. Pears, J. Xu and C. Boldyreff, "Mobile Agent Fault Tolerance for Information Retrieval Applications: An Exception Handling Approach", 2003.
[15] B. E. Isong and O. O. Ekabua, "A Generic Framework for Mobile Agent's Fault Tolerance", IEEE, 2011.
[16] M. A. J. Jamali and H. E. Shabestar, "A New Approach for a Fault Tolerant Mobile Agent System", 12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2011.
[17] Summiya, K. Ijaz, U. Manzoor and A. A. Shahid, "A Fault Tolerant Infrastructure for Mobile Agents", International Conference on Computing Intelligence for Modeling Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), IEEE.
[18] A. Budi., I. Alexei., R. Alexander., "On using the CAMA framework for developing open mobile fault tolerant agent systems", In Proceedings of the 2006 international workshop on Software engineering for large-scale multi-agent systems, May 22-23, 2006,Shanghai, China.
[19] P. Holger. , P. Stefan ., G. Claus., "FANTOMAS: Fault Tolerance for Mobile Agents in Clusters", In Proceedings of the 15 IPDPS 2000 Workshops on Parallel and Distributed Processing, pp.1236-1247, May 01-05, 2000.
[20] J. Dag, R. V. Renesse and F. B. Schneider, " Operating System Support for Mobile Agents", Position paper for 5th IEEE Workshop on Hot Topics in Operating Systems.