

# Exploration of Various Fault Tolerance Techniques

Payal Chhabra  
PG Student

Department of Information Technology  
College of Technology  
G.B Pant University of Agriculture and Technology  
Pantnagar, Uttarakhand, India

Hardwari Lal Mandoria  
Professor

Department of Information Technology  
College of Technology  
G.B Pant University of Agriculture and Technology  
Pantnagar, Uttarakhand, India

## ABSTRACT

In the past few years, Mobile agent technology is one of the emerging, fastest growing and promising concepts for the development of applications. A mobile agent is composition of computer program that proceed autonomously for a user or application moving in the heterogeneous communication environment. This survey presents “pure fault tolerance technique “that enables a system to continue operating their functions properly while, occurrence of the failure of some of its components in mobile agent systems. It is mandatory that the mobile agents should be reliable so that mobile agent’s technology continues. To verify fault and failure free process of mobile agent based applications, consistency need to be addressed. Fault tolerance is important to the further expansion of mobile agent applications. In the framework of mobile agents, fault-tolerance prevents an incomplete or entire loss of the agent, i.e., it confirms that the agent arrives at its target.

## Keywords

Mobile Agent System, Fault Tolerance System, Fault Tolerance Techniques.

## 1. INTRODUCTION

Mobile agents are programming paradigm for distributed applications [3]. Mobile agent approach is different from others because it transports data as well as code acting on the data among the nodes. Because of the moving of the code, flexibility of application is improved. For building distributed applications, Mobile agents are a useful technology. Advantages of mobile agents are autonomy, flexibility, and effective usage of network bandwidth. On the basis of these features, they require as an enabling technology for mobile, wireless and pervasive computing.

A mobile agent is a successively program that can travel from host to another within network and decides when and where to move [12]. Mobile agents are one method of mobile code. In its humblest form, the formation of mobile code involves the dynamic installation of code on a remote host. In Web applications, common forms of mobile code are servlet and applets. The mobile code concept also seems in “Remote evaluation systems”, which transfer executable software code from a client to a server for following execution at the server. After the code has completed its execution, then results are sent back to the client. A mobile agent runs in one place, moves from one host (with its state) to another host, and continues at that host.

Mobile code and mobile objects are normally relocated by an exterior entity; mobile agents have feature of migration autonomy. As Compared to client-server approaches, mobile agents can keep away from transmitting of much data across the network. The mobile agent can move with restricted results, from one server to another until it has completed its

task, and then return back to the originating host, which may freely be disconnected for the duration of the agent’s travels.

- **Computation Bundles** - mobile agent move the code near the data sources instead of pulling the data into the code, which reduces network load.
- **Parallel Processing** - continuous open connection requires a high cost so mobile agent allowed tasks to execute independently and asynchronous on multiple heterogeneous network hosts
- **Dynamic Adaptation** - Mobile agents can observe the changes in environment and proceed dynamically.
- **Tolerant to Network Faults** - able to operate when a client machine can be disconnected from the network
- **Flexible Maintenance** - need to update only source to change an agent’s events (rather than the computation hosts).

A **Fault-Tolerant Design** [13] allows a system to tolerate its intended operation, possibly at a reduced level, quite failing completely, when some part of the system is abortive. The term is usually used to designate computer systems designed to remain more or less fully operational with, possibly, a decline in throughput or an increment in response time whenever some partial failure. That is, the system which is not stopped because of hardware or the software problem.

Fault tolerance system is a crucial issue in distributed computing; it keeps the system in a running state in subject to failure. The most essential point of it is to keep the system operative even if any of its part goes rotten or not working.

Failures in a mobile agent system can incompletely or completely fail the agent. To make a system fault-tolerance, the agent owner (i.e. the person or application create and configure the agent) can try to distinguish the failure of its agent, and upon such an event launch a new agent. In contrast, it has capability to properly identify the crash of the agent, i.e., to make a distinction between a failed agent and delayed agent specifically delayed by slow processors or communication links.

## 2. EXPLORATION OF FAULT TOLERANCE TECHNIQUES

Agent centric technique: The Agent centric fault tolerance techniques [1, 2, 7, 8 and 14] are responsible for fault tolerance within mobile agents.

System centric technique: The System centric fault tolerance techniques [3, 4, 5, and 6] are responsible for fault tolerance using mobile agents.

### **2.1 An Exception handling approach**

When there is a failure of agent server, consequently terminating all active mobile agents to conserve the availability of mobile agents by using two exception handler [1] strategies. First handler is occurs at the server which created the mobile agent and uses a timeout mechanism. The handler waits for a timeout period and resends mobile agents that did not answer or return. Second handler uses mobile shadow scheme, which creates a pair of replica mobile agents, master and shadow, to continue remote agent server crashes. If a master crash is identified, the shadow spawns a new shadow and converts to new master, else the shadow receives a die message and terminates. The advantage of this approach is that there is message passing mechanism among the replicas agents.

### **2.2 Antecedence Graph Approach**

This approach [14] logged the dependency information or recording the dependency relation between mobile agents in the form of antecedence graphs by sharing mobile agents of mobile agent group. Checkpointed method is parallel used for fault tolerance in the form of antecedence graphs. Whenever failures occur, Message logs combined with Antecedence graph are used and they both are regenerated by using Checkpointed information for recovery and then continuous its normal procedure. Checkpointed antecedence graphs are stored into stable storage which is finally used for recovery and fault tolerance. The suggested algorithm can decrease the time latency, improves the message overhead, recovery time, execution time, and considers the minimum number of mobile agent, which expands the system performance.

### **2.3 Host Criticalities Approach**

To introduce fault tolerance this approach [3] collects global or local information of the mobile agent system to identify critical host agents which is determined by calculating weights i.e., the weights of hosts are considered. Now, for making a decision of checkpointing these weights are used. Mobile agents continuously monitor each host in the system based on updating of weights from time to time. Further we propose the calculation of host dependence (i.e. criticality of hosts) by using various parameters such as executed tasks and type and number of messages exchange in system using. These weights may be convenient in identifying which host becomes too heavy or if system is loading few hosts. Greater the dependency, higher is the important role of host. From experimental research result it can be inferred that it increases fault tolerance, response of system time by actual acknowledgment of openness in system.

### **2.4 Novel Dynamic Shadow Approach:**

It challenges the problem of server crash by creating a duplicated copy of the original agent. The replica [8] monitors the original agent in the itinerary, whenever failure occurs; it can be recovered from its clone. This approach correspondingly added checkpointing, so that when original agent and its clone fail then it roll back to the home agent and finally a replicated copy of agent is conducted. The original agent dispatches for retrieval of information in itinerary. Once collecting of information, the agent returns to home or actual server. The clone dispatched automatically and cracked after the original agent transfers to the next server. The results show that by the total trip time, checkpoint time and

successful migration time there is an enhancement in the performance.

### **2.5 Witness Agents in Linear Network**

In this approach [4, 5] new agent is created called witness agent who observes the actual agent (who accomplishes programs for its owner) and monitor whether the actual agent is alive or dead. Checkpointed data or log information is used to recover the lost agent. In this design, agents are able of transferring messages to each other. As a witness agent constantly lags at the back of actual agent so, actual agent can imagine that witness agent is at the server that is just earlier visited by actual agent. Furthermore, the actual agent always acquainted with the address of the earlier visited server. Hence, the peer-to-peer message passing mechanism can be established. If the witness-dependency is preserved, then failure detection and recovery can be achieved for agent.

### **2.6 Adaptive Mobile Agent System using Dynamic Role based Access Control**

Adaptive Mobile Agents are working inside a special environment called context-aware environment which accomplishes the task of sharing and allocating the roles to the mobile agents exist in the environment. Mobile agents obtain roles based on the information known by the environment. The Adaptive Mobile [6] essentially works together with one another and with the environment to acquire roles. Roles are allocated to restrict or grant access to a resource. The way of restricting or granting access to a resource is called Role Based Access Control (RBAC) which conserves the confidentiality and integrity of information. Dynamic Rule Generator is used to assigned roles based on rules. The communication among various components is accepted by communication messages. The advantage of this technique stays that as mobile agents are already inside the system, it does not involve any sort of external communication. As a result, there is no need to create and dispatch a new mobile agent which turns response time into a smaller amount.

### **2.7 Region-based Stage Construction Protocol**

RBSC protocol is used for fault tolerant execution of mobile agents in a multi-region mobile agent computing environment [2]. It uses new notions of quasi-participant and sub stage in order to construct a stage according to region. Each action  $a_i$  that execute on a place  $P_i$  is called a step. Each step contains a set of places called a stage  $S_i$ .  $P_i^w$  at  $S_i$  is called a worker, the others are called participants. As soon as a worker fails; one of participants is designated as a new worker and takes over the achievement of the previous worker. Consequently, this protocol moderates the total execution time and reduces the overhead of stage works and the drawback is that an overhead occurs for stage constructed in the same region.

### **2.8 FANTOMAS**

This is logger agent based fault-tolerant mobile agent system; FANTOMAS [7] practices the novel "agent dependent" approach in which logger agent is combined into the user agent and travels with it. This protocol is responsible for fault tolerance activities with the agent. Both the agents continuously monitor each other and whenever failure occurs, it is recovered by other one by using its local information. FANTOMAS is java based and supports Objectspace's Voyager mobile agent platform. The communication of a user-defined agent with the FTE generates a fault tolerant mobile agent. This has the important advantage that existing mobile agent platforms do not need to be modified. The cost

of FANTOMAS is reasonable and provides: non-blocking and exactly once mobile agent execution.

### 3. CONCLUSION

Mobile agents systems have enhanced the tendencies of distributed computing. Their acceptance is on the rise progressively. This paper analyses different fault tolerance techniques for the scenarios where the agent stops its execution because of fault on any server. These techniques are proposed for fault tolerant systems presented by various authors. According to conducted survey it is conclude that if the methods are agent based but in case the same agent

becomes faulty it becomes difficult to overcome failure. As to achieve reliability, enhance the tolerance factor by dynamically allocating the task of the failure node to other nodes which are active at present time.

In future, it improves the performance, reliability and deliver quality of service as required and to make the system fault tolerant if it fails in case of network or congestion. In the developing era of mobile agents; the subsistence ability, reliability, scalability and fault tolerance are used for such trending concerns in distributed system.

**Table 1. Comparison of various techniques**

Parameters Techniques	Agent Centric	System Centric	Description	Drawback
An Exception handling approach	Yes	No	Mobile time out design and mobile shadow design are used to achieve fault tolerance.	It is not a Harsher failure model, suffers Blocking
Antecedence Graph Approach	Yes	No	Logged or recording the dependency relation between mobile agents in the form of antecedence graphs.	Increase the overhead of captivating message log and antecedence graph of every message.
Host Criticalities Approach	No	Yes	Critically of hosts in mobile system is being considered.	To accurately measure host vulnerability and efficiency such that more formal model can be assembled.
Dynamic Shadow Approach:	Yes	No	Clone of original agent is created. This clone is used for recovery.	Sometimes disturb the exactly once property of mobile agents.
Witness Agents in Linear Network	No	Yes	Fault tolerance is attained by mutual aid of agents with each other.	Consumes a lot of resources along the itinerary.
Adaptive Mobile Agent System using Dynamic Role based Access Control	No	Yes	Adaptive Mobile Agents accept additional roles while working inside an environment.	Overheads are there in updating the Access Control List.
Region-based Stage Construction Protocol	Yes	No	Used in a multi-region mobile agent computing and reduces the overheads of stage works.	Reduced reliability and security. At each time worker fails, quasi participant substitutes it's position with real- participant
FANTOMAS	Yes	No	Allow fault – tolerant agent execution without having to modify the underlying mobile agent platform.	Overhead acquaint with the replication and with increasing the number of stages and size of the agent.

### 4. REFERENCES

- [1]. S. Pears, J. Xu, C. Boldyreff.2003. Mobile Agent Fault Tolerance for Information Retrieval Applications: An Exception Handling Approach," Proc. of the 6th International Symposium on Autonomous Decentralized Systems.
- [2]. S. J. Choi, M. S. Baik, H. S. Kim, J. W. Yoon, J. G. Shon, C. S. Hwang.2004. Region-based Stage Construction Protocol for Fault tolerant Execution of Mobile Agent. Proc. of the 18th International Conference on Advanced Information Networking and Application.
- [3]. S. Rajwinder and D. Mayank.2012.Using Host Criticalities for Fault Tolerance in Mobile Agent Systems. In Proc. of IEEE International Conference on Parallel, Distributed and Grid Computing.
- [4]. A.Rostami, H.Rashidi, M. S. Zahraie. September 2010. Fault Tolerance Mobile Agent System Using Witness Agent in 2-Dimensional Mesh Network. International Journal of Computer Science Issues, Vol. 7, Issue 5.
- [5]. S. Beheshti and A. Movaghar.2006.Fault tolerance in Mobile Agent Systems by Cooperating the Witness Agents. In Proc. of International Conference on Information and Communication Technologies, Vol. 2, Page(s): 3018 – 2, Publication IEEE Conference.
- [6]. P. Marikkannu, J.J. Adri Jovin, T.Purusothaman. April 2011.Fault-Tolerant Adaptive Mobile Agent System using Dynamic Role based Access Control. International Journal of Computer Applications Volume 20– No.2.
- [7]. P.Holger, P.Stefan, G. Claus.2000.FANTOMAS: Fault Tolerance for Mobile Agents in Clusters .In Proceedings of the 15 IPDPS 2000 Workshops on Parallel and Distributed Processing, pp.1236-1247.

- [8]. R. Hans, R. Kaur.2012.Novel Dynamic Shadow Approach for Fault Tolerance in Mobile Agent Systems. IEEE.
- [9]. M. Strasser, K. Rothermel.1998.Reliability concepts for mobile agents. International Journal of Cooperative Information Systems.
- [10].Shahnawaz Husain, SC Gupta, Mukesh Chand, HL Mandoria .2010.A proposed model for intrusion detection system for mobile adhoc network”, Computer and Communication Technology (ICCCT), 2010 International Conference.
- [11].Ashok Kumar, HL Mandoria, BK Pandey.2016.Study and Comparison of Distributed Energy Efficient Clustering Protocols in Wireless Sensor Network: A Review. i-Manager's Journal on Wireless Communication Network.
- [12].Danny B. Lange.1998.Mobile Objects and Mobile Agents: The Future of Distributed Computing?” This paper is based on a chapter of a book by Lange and Oshima entitled Programming and Deploying Java™ Mobile Agents with Aglets™, Addison-Wesley, ISBN: 0-201-32582-9.
- [13].P. Jalote.1994. Fault Tolerance in Distributed Systems. T R Prentice Hall, Englewood Cliffs, New Jersey, ISBN 0-13- 301367.
- [14].M. Masud Khokhar, Aamer Nadeem, Omer Mansoor Paracha.2006. An Antecedence Graph Approach for Fault Tolerance in a Multi-Agent System. Proceedings of the 7th International Conference on Mobile Data Management (MDM'06) 0-7695-2526-1/06 IEEE.