Analysis of Fault Tolerant Techniques in Secure Mobile Agent Paradigm

Parul Arora[1], Sahil Vashist[2], Rajwinder Singh[3], and Rahul Hans [4]
[1]Dept. of CSE, CEC, Landran, Mohali, Punjab, India, parularora2590@gmail.com
[2]Dept. of CSE, CEC, Landran, Mohali, Punjab, India, cecm.cse.sahil@gmail.com
[3]Dept. of CSE, CEC, Landran, Mohali, Punjab, India, rwsingh@yahoo.com
[4]Dept. of IT, D.A.V.I.E.T, Jalandhar, Punjab, India, rahulhans@gmail.com

***Abstract*—Since the past few years, the network domains and mobile agent technology has been the fastest growing and emerging trend as well. But it has to undergo certain challenges and problems, in order to meet the bandwidth requirements. Moreover, it suffers from issues related to reliability like security and fault tolerance. During the agent migration in an itinerary from one server to other common issue is server crash or agent crash. The parameters used for the evaluation of various techniques are agent centric, system centric, fault type, coordination performance analysis, central management and adaptive. Advantages of each mechanism are also described.**

***Index Terms*—Agent Centric, Blocking Attacks, Checkpoint, Fault Tolerance, Monitoring, Rollback Recovery, System Centric.**

I. Introduction

Mobile Agents are capable to run on the machines where the resources are sufficient enough for the completion of particular tasks. During migration, the agent moves to another host, performs the computation and finally transfers results in network. Therefore, network has no congestion issues. A mobile agent is a composition of code, state and data space that moves across the network in an autonomous fashion for interaction [12, 13]. The mobile agent can not only transfer within a particular network but the migration can also be from one network to another. The nature of the agent is based upon its mobility which is of two types:

A. *Weak Mobility*: Migration of only code occurs but not the execution state.

B. *Strong Mobility*: With the execution state, the program itself moves to remote location.

As code and data transportation takes place, it can bring flexibility in the mobile agent systems. Traditionally, the Client-Server paradigm is followed where the client sends the request to the server through the message passing process and the latter provides the service to the former.

The drawback of the model is that the network bandwidth is consumed inefficiently if the client does not get the required service from the server. So, it may send requests to other servers to fulfill its requirement. Some other challenges in this

Architecture may include server crash or agent crash.

Mobile Agent is a promising paradigm due to the fact that these do not depend on the server side operations.

Furthermore, presence of fault is another challenge which violates the reliability of mobile agent systems. Therefore, the concept of fault tolerance is used which resists the effects of faults and unexpected conditions, and aims to result in the successful execution in a graceful fashion.

Moreover fault tolerance is a technique that prevents the complete or partial loss in the information during migration. As far as the execution is concerned, the mobile agent must adhere to the following properties:

A. *Non Blocking:* This property illustrates that in case of any failure, the execution of the agents must not be blocked.

B. *Exactly Once:* This property states that single execution of agent should be done exactly once.

In the mobile agent system the following failures may occur:

A. *Message Loss*: This may occur in case if the network fails.
B. *Server Failure*: This is also known as Node failure. We know that if a server fails, all the agent places are failed. This can be a temporary failure of system.
C. *Agent Failure*: If the computing scheme offered is faulty then this failure may occur.
D. *Incomplete directory of agent fault*: This may result due to component failure in any mobile agent system.
E. *Communication Fault*: This arises due to single link failure of entire link failure.

A system can be made reliable if it can overcome such failures and increase the performance of the system.

The rest of the paper has been organized in the following fashion. Section 2 describes the various

doi:10.4304/jetwi.6.2.231-236

techniques that have been proposed by various authors includes the performance evaluation of these techniques by considering parameters like- agent centric, system centric, fault type, coordination performance analysis, central management and adaptive. Further the advantages of each mechanism are described. Finally Section 4 includes the conclusions and further future work.

II. Fault Tolerance Mechanisms

The basic aim of using fault tolerant techniques is that system must result in successful execution and should remain unaffected from faults and unexpected conditions i.e. it should perform in graceful fashion [11]. Various techniques used are as follows:

A. *Antecedence Graph Approach to Checkpointing for Fault Tolerance in Mobile Agent Systems*

In [1] to overcome certain failures, the authors have proposed an approach for providing efficient fault tolerance in mobile agent systems. Here, the parallel checkpointing approach is used which considers the antecedence graphs. A checkpoint is nothing but just a snapshot of process state which has been saved.

Checkpointing techniques can be classified on the basis of algorithms which can be synchronous, asynchronous or quasi-synchronous. Before an interval of state, the information of the events already occurred is stored in the form of antecedence graphs. These graphs are directed acyclic graphs that record the dependency information.

The identifying time is minimized and time latency is decreased for global checkpointing procedure, as during the beginning the relevant mobile agent is being informed by the initiator simultaneously.

Hence, whenever any unexpected situation or failure occurs, these graphs are regenerated for recovery. The system performance is improved as the proposed approach considers the minimum number of mobile agents.

Further, this approach improves the message overhead, recovery time and execution time as well.

The previous antecedence graphs are deleted after the final checkpointing has been done which in turn reduces the graph size. In this scheme, coordinated checkpointing is used which improves performance as compared to other schemes. Another aim of this approach is that the number of blocked mobile agents are minimized which in turn increases the efficiency as compared to traditional approaches.

The advantage of this technique is that the identifying, execution and recovery time is improved and the message overhead can be reduced. The disadvantage of this technique is that there is an overhead of taking message log and antecedence graph of each message.

B. *Using Host Criticalities for Fault Tolerance in Mobile Agents*

In [2] the authors have presented fault tolerance mechanisms and monitoring in the mobile agent systems. A group of mobile agents are used which in turn monitor including the advantages and disadvantages. Section 3 each host. A prediction mechanism is built to identify critical host agents so as to prevent failure. Therefore, updating of information in the system can be done in a consistent fashion. Here this mechanism includes the weighing calculation i.e., the weights of hosts are considered.

The mobile agents which act as monitoring agents detect the unexpected situations and then help in recovery. Further the host dependence calculation is proposed by the use of an algorithm to update weights. In other words critically of hosts in mobile system is being calculated. The aim is to monitor constantly but in order to obtain reliable behavior the host agents are monitored dynamically which will result in improvement of response of system.

Hence, proposed monitoring mechanism has the following components:

a) *Monitor Agent (MoA):* In the whole module of monitoring, these are at the lower level i.e., they reside on hosts that provide services. These service hosts keep a record of information in the form of log tables. These agents monitor the surrounding atmosphere and gather information of exchanging messages.

b) *Manager Agent (MaM):* This resides at the higher level and controls the agents during monitoring. It can correlate different local events from various monitored hosts in time.

c) *Alert Agent (AlA):* It is made up of many blocks like time stamps that include the alert message creation time, threshold event detection time, alert in case of any fault and alarm information as well. But this works under the MaM.

The concept of checkpointing is also considered for providing fault tolerance. The correlation among mobile agents can be efficient in such a way if the solution is platform independent. The exchange of messages is considered for interaction of agents. The messages that are exchanged between hosts are considered as weights.

Hence the hosts can be detected by such number if host becomes heavy and hence can be prevented. In a nutshell, this proposed mechanism not only recognizes the vulnerability in the system but also increases the fault tolerance.

The advantage of this technique is that fault tolerance can be increased and mobile systems can be made reliable. The disadvantage of this technique is that apart from the parameters that have been considered, some other parameters can be considered in order to accurately measure host vulnerability and to gauge efficiency such that more formal model can be built.

C. *Novel Dynamic Shadow Approach for Fault Tolerance in Mobile Agent Systems*

In [3] the authors have proposed an approach to sort out the agent crash problem. Here the clone of original agent is created. This clone is used in an itinerary to follow the actual agent. So, if any failure occurs in the mobile agent system, the recovery is possible by the clone.

Further if a clone also fails then checkpointing approach can be used. The improvement in performance is evaluated by the total trip time. Here web based e-marketplace is considered. User requests for products information by the use of mobile agents.

The mobile agent migrates and compares the prizes of products according to user requirements. The migrating mobile agents select servers dynamically across the network. The original agent dispatches for retrieval of information in itinerary.

After gathering of information, the agent returns to home or actual server. The clone of this agent is dispatched automatically. It is destroyed when the original agent moves to the next server as the copy of the clone is made at the current server where the agent has moved.

But if both agent and clone get lost then the fault message is sent to home server and finally a replicated copy of agent is sent. In this way fault tolerance can be achieved.

Various experiments are conducted by the authors to calculate total trip time of original agent and clone agent with checkpoints; effect on total trip time in case if both get lost due to any fault; and calculation of overheads due to checkpoints.

The main aim is to limit the rollback by adding checkpoints [14, 15]. No doubt due to checkpoints the overhead might raise but it is considered bearable even in the applications related to real time scenario. The advantage of this technique is that performance can be improved by total trip time, checkpoint time and successful migration time. The disadvantage is that it may sometimes violate the exactly once property of mobile agents. It does not take into account write only and read/write applications.

D. Integrated Mechanism to Prevent Agent Blocking in secure Mobile Agent Platform System

In [4] the authors have proposed an integrated mechanism using SMAPS (Secure Mobile Agent Platform System) to prevent agent blocking in some situations where agent is captured by malicious host. In case of execution of the transactions the partial result retrieval is considered which is further used for tracking the location of mobile agents at any time during the process.

The parameters that have been considered are: fault time, overhead in communication and size of message. The proposed mechanism has been implemented on SMAPS. Web Application based on e-marketplace has been considered for comparison of products' prizes.

The experiments that are conducted by the authors are: effect of acknowledging frequencies and fault tolerant time effect. For originator, the tracking can be possible through acknowledgements. After some fault tolerant time, some partial results are sent back and predefined number of hosts are visited which prevents attacks and hence provide protection and reliability.

Therefore, the main aim is to make the mobile agents suitable for real time applications since it reduces the possibility of complete loss of agents.

The advantage of this technique is improvement in reliability and performance by parameters such as communication overhead, fault tolerance time and message size. The disadvantage is that this technique is suitable only for time sensitive applications.

E. Fault Tolerance Mechanism in Secure Mobile Agent Platform System

In [5] the authors have proposed a mechanism which is based on replication as well as voting for mobile agent platform system. The system considered is Secure Mobile Agent Platform System (SMAPS). The result of reliability and performance is evaluated which helps in increasing the reliability in the mobile agent system. A replication system is presented with voting. The approach used here is transparent in mobile agent. An architectural design is also discussed by using the replication schemes so as to make the mobile agent a fault tolerant system.

Even though the concept of the mobile agent is considered better than the Remote Procedure Call, still there are many issues related to the design, implementation and deployment. Each node is assumed to have an AHS (Agent Host Server) that presents a reliable environment on which one may create new agents that may further get dispatch. AHS agents are light weight implementations as we consider threads rather than processes. The mobile agents follow the autonomous features and the message passing takes place in an asynchronous fashion. In this proposed model, complete isolation is provided against any unauthorized access. Furthermore, much failure can be tolerated by computation offered by replicated agent as against the voting architectural model. Here the experiments conducted were the effect of voting frequencies, effect of replication methods i.e. active or passive. Basic conclusion obtained is that replicated computation with voting can improve the performance. Though there may be slow hosts and delays in the synchronization, still these factors do not affect the performance.

The advantage of this technique is that performance can be improved by replicated computations with voting improve performance as it is not affected by slow hosts. The disadvantage is that it does not make use of features like cryptographic protocols which may in turn increase more security in the system.

F. Improving Fault Tolerance of Distributed Multi-Agent Systems with Mobile Network-Management Agents

In [6] the paper describes the concept of agent tracking technique and to provide a robust system so that efficient management as well as the maintenance can be done in the networks of mobile agent systems. The concept of Multi Agent Systems (MAS) is considered that controls the life cycle of agent.

The Extensible Java-based Agent Framework (XJAF) has also been described by the authors. This framework has to undergo certain challenges for the tree like architecture.

Therefore, the work has been provided to alleviate such challenges and thus to improve the fault tolerance. A

reliable network can be built by following fully connected graph which includes registration of MAS and detection in case of broken MAS. The concept of agent tracking is also laid.

Two types of mobile agents described for providing fault tolerance are:

a) *ConnectionAgent* so as to build reliable network

b) *RemnantAgent that provides agent tracking mechanism.* The advantage of this technique is Maintenance and Efficiency is achieved by robust agent tracking technique. The disadvantage is that replication as well as timely fault detection techniques are not included which may in turn increase the flexibility and robustness in the system.

G. Optimistic Replication Approach for Transactional Mobile Agent fault Tolerance

In [7] the authors have laid the fault tolerant mechanism with respect to the applications concerned with transactions. Partial or complete loss can be prevented by the use of suitable protocols so as to ensure the atomic execution of mobile agents. The design of MAS is described which includes Place, Transaction_Manager, mobile agent, Watch agent and Itinerary.

The proposed protocol provided depends upon the behavior of mobile agent, watch agent and Transaction_Manager. If 'commit at destination' protocol is used, exactly-once execution property is not violated.

The advantage of this technique is Transactional support is provided. The disadvantage is that fault tolerance in case of Transaction Manager (TM) has not been considered. By considering this a non blocking atomic commitment protocol can be integrated in TM to validate global transaction that is being executed by Mobile agent.

H. A Survey of Fault Tolerance Techniques in Mobile Agents and Mobile Agent Systems

In [8] the paper represents the concept of MoCA (Mobile Collaboration Architecture) and CHAMELEON for providing fault tolerance in mobile agent and mobile agent systems respectively. The Agent Centric techniques provide fault tolerance in mobile agents where as the system centric techniques provide fault tolerance in mobile agents whereas system centric techniques provide fault tolerance by the use of mobile agents.

Further the authors have presented an evaluation of Agent Centric Approaches which provide the parameters as exception handling, CAMA (Content Aware Mobile Agents), Transient faults, Improvement by the use of replicated agents, Protection by the use of reference state, FANTOMAS (fault Tolerant Approach for Mobile Agents) and MoCA.

System centric approaches can be evaluated by the use of the parameters: Shadow Approach, CHAMELEON, MAgNET (Mobile Agents for Networked Electronic Trading) and Fault Tolerance Directory Service for Mobile Agents.

Conclusion obtained in MoCA is based on backward recovery technique and it can tackle not only transient but also permanent faults whereas CHAMELEON is based on the centralized management.

The advantage of this technique is follows the Backward recovery technique.

The disadvantage is that this technique does not take into account the dynamic shadow approach.

I. Checkpointing Using Mobile Agents for Mobile Computing System

In [9] the most important used technique that provides reliability in distributed applications is checkpoint based rollback recovery has been described. Here authors have provided an algorithm of rollback recovery based on message logging and independent checkpointing along with a technique to provide management for the same. When the mobile node goes far from the latest checkpoint, the management is done in such a way that checkpoint and message logs can be moved. Hence, the time that mobile agents will need to recover will never exceed threshold value. Authors have also provided the checkpointing protocol to enable mobile agent.

By using this technique Checkpoint Transfer Time (CTT) and total cost of recovery can be calculated and compared to already existing checkpointing techniques. Network bandwidth can be saved as no synchronization needs to be followed since the authors have made use of independent checkpoints. Therefore, this scheme is considered scalable.

The advantage is that this technique is considered as Scalable because message log does not become too large.

J. Agent Tracking: A Reliable Agent Forwarding Mechanism

In [10] authors have laid a mechanism to develop framework to offer reliability for mobile agent systems in distributed environment. Reliability is one of the factors that is necessary during deploying mobile agent systems. The proposed model puts forward the concepts related to network monitoring, routing, resource discovery etc. Agent tracking is necessary in the real world scenario in order to know that where the agent actually is at any particular instant of time.

Agent tracking is necessary to ensure operations that are error free. The possibilities in agent tracking are:

a) ARRIVAL MIGRATE_LOGGING (AML) and
b) REMOTE SEARCHING.

The concept of User Host Tracker is given. The message will be sent by every host to TRACKER which will then forward it to the particular destination. TRACKER can also be made aware of detail of agents at the remote host. Measurement of overhead is also considered.

The advantage of this technique is Application Programmer has the authority to track the agents. The disadvantage is that this technique does not offer fault tolerance and security of mobile agents. The TRACKER is also not aware of the other agents' details at any other remote host.

III. Comparison of Fault Tolerance Mechanisms

Here the mechanisms are compared by the use of some parameters like agent centric, system centric, fault type, coordination, performance analysis, central management and adaptive nature.

A. *Agent Centric:* The fault tolerance which is achieved within agent is agent centric. These are marked by Yes.
B. *System Centric:* Here fault tolerance is achieved with the help of mobile agents. These are marked by Yes.
C. *Fault Type:* It describes that which type of fault is being considered: agent or agent and server.
D. *Coordination:* It shows that how the communication is to be carried out; it can be D (Direct) or IND (Indirect). Communication overhead in case of direct is less. In case of Indirect, communication delays can be possible.
E. *Performance Analysis:* It describes whether the performance analysis has been considered in a particular paper or not. It is marked by Yes or No respectively.
F. *Central Management:* This parameter defines whether the central management i.e., server is used or not. It is described by Yes if central management is used.
G. *Adaptive:* This parameter defines whether the system is adaptive to other platforms and circumstances and is marked by Yes.

IV. Conclusion And Future Scope

The communication as well as computing entities like software agents, computer hosts and application services is becoming mobile and decentralized. So, there is a need of fault tolerance mechanism in the use of mobile agents in order to make the system threat free and less prone to errors. We know, in the emerging era of mobile agents; the survival ability, reliability, scalability and fault tolerance are much are used for the comparison of such trending issues in distributed system.

This paper represents a comparative analysis of the various existing fault tolerant techniques. The various parameters are used for the comparison of such techniques. Advantage of each technique has also been described. The basic aim is to improve the performance, introduce reliability and deliver quality of service as required and to make the system fault tolerant if it fails in case of network or congestion.

So, if an agent fails and is unable to track, location tracking mechanism with fault tolerance can be introduced to overcome such challenge. We intend to work on secure fault tolerance in Agent Tracking i.e. reliable mechanism for forwarding of agents in distributed environment so that improvement can be made in the fault tolerance techniques used in the case of agent tracking mechanism.

TABLE I.

COMPARATIVE ANALYSIS OF TECHNIQUES

Mechanism / Parameters	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
Agent Centric	Yes	No	Yes	No	Yes	Yes	Yes	No	No	Yes
System Centric	No	Yes	No	Yes	No	No	No	Yes	Yes	No
Fault Type	Agent	Agent	Agent & Server	Agent	Agent	Agent	Agent	Agent	Agent	Agent
Coordination	D	D	IND	D	D	IND	D	D	D	D
Performance Analysis	Yes	Yes	Yes	Yes	Yes	No	No	Yes	No	Yes
Central Management	No	No	Yes	No	No	Yes	Yes	No	Yes	No
Adaptive	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No

TABLE II.
ADVANTAGES OF TECHNIQUES

[1]	Improved Identifying, execution and recovery time; reduced message overhead.
[2]	Mobile Systems can be made reliable and fault tolerance can be increased.
[3]	Improved Performance by total trip time, successful migration time and checkpoint time.
[4]	Improvement in reliability and performance by parameters such as communication overhead, fault tolerance
[5]	Replicated computations with voting improve performance as progress is not affected by slow hosts.
[6]	Efficiency and Maintenance is achieved by robust agent tracking technique.
[7]	Transactional support is provided.
[8]	Backward recovery technique is being followed.
[9]	Scalable technique as message log doesn't become too large.
[10]	Application Programmer can track the agents.

REFERENCES

[1] S. Rajwinder and D. Mayank, "Antecedence Graph Approach to Checkpointing for Fault Tolerance in Mobile Agent Systems," In Proc. of IEEE Transactions On Computers, Volume 62, issue 2, Publication IEEE Conference, February 2013.

[2] S. Rajwinder and D. Mayank, "Using Host Criticalities For Fault Tolerance in Mobile Agent Systems," In Proc. of IEEE International Conference on Parallel, Distributed and Grid Computing, 2012.

[3] H. Rahul and K. Ramandeep, "Novel Dynamic Shadow Approach for Fault Tolerance in Mobile Agent Systems," In Proc. of 6th International Conference on Signal Processing Communication Systems, Publication IEEE Conference, 2012.

[4] K.Ramandeep, K.C. Rama and S.Rajwinder, "Integrated mechanism to Prevent Agent Blocking in Secure Mobile Agent Platform System", In Proc. of International Conference on Advances in Computer Engineering, 2010.

[5] S. Rajwinder, A.K.Sarje, K. Navdeep and K.Ramandeep, "Fault Tolerance Mechanism in Secure Mobile Agent Platform System," In Proc. of International Conference on Advance Computing, Publication IEEE Conference, 2009.

[6] M.Dejan, B.Zoran, I.Mirzana and V.Milan, "Improving Fault Tolerance of Distributed Multi-Agent Systems with Mobile Network-Management Agents," In Proc. of the International Multiconference on Computer Science and Information Technology, pp. 217-222, 2010.

[7] L.Zeghach eand N.Badache, " Optimistic Replication Approach for Transactional Mobile Agent Fault Tolerance," In Proc. of 11th CIS Internatioanl Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2010.

[8] Y.Farzana and Z.Zahid, "A Survey of Fault Tolerance Techniques in Mobile Agents and Mobile Agent Systems," In Proc. of Second International Conference on Environment and Computer Science, 2009.

[9] C.Chandreyee and N.Sarmistha, "Checkpointing Using Mobile Agents for Mobile Computing System," In Proc. of International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009.

[10] U.P.Kulkarni, Prof A R Yardi, "Agent Tracking: A Reliable Agent Forwarding Mechanism", Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05).

[11] A.R.NaghshNilchi, A.Vafaei and H. Hamidi, "Evaluation of Security and Fault Tolerance in Mobile Agents," In Proc. of International Conference on Wireless and Optical Communications Networks, Publication IEEE Conference, 2008.

[12] Danny B. Lange, "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, 1998.

[13] S. Beheshti and A. Movaghar, "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, 2006.

[14] H. Rahul and K. Ramandeep, "Fault Tolerance Approach in Mobile Agents for Information Retrieval Applications Using Check Points," In Proc. of International Journal of Computer Science & Communication Networks, Vol. 2, Issue 3, page(s) 347-353, 2012.

[15] P. Marikkannu, J.J. Adri Jovin, T.Purusothaman, "Fault-Tolerant Adaptive Mobile Agent System using Dynamic Role based Access Control," In Proc. of International Journal of Computer Applications Volume 20–No.2, April 2011.