

## A Fault Tolerant Mobile Agent Information Retrieval System

<sup>1</sup>R. Punithavathi and <sup>2</sup>K. Duraiswamy

<sup>1</sup>Department of Information Technology, Chettinad College of Engineering and Technology,  
Karur (DT), Tamilnadu India 639114

<sup>2</sup>Department of Computer Science and Engineering, KS Rangasamy,  
College of Technology, Tiruchengode, Tamilnadu India 637209

---

**Abstract: Problem statement:** Most of the information retrieval systems used only client-server architectures. The client-server model though powerful, had some limitations. In mobile computing environment which has both wired network and wireless networks with limited communication capabilities, the performance of the system was very low. **Approach:** Mobile agents are considered a suitable technology to develop applications such as information retrieval system for mobile computing environment. Mobile agents are autonomous and dynamic entities that can migrate between various nodes in the network. They offer many advantages over traditional design methodologies like: reduction in network load, overcoming network latency and disconnected operations. Since the mobile agents do not need continuous communication with the mobile host, they are not affected by the sudden disconnection of wireless network and the situation of turning mobile host off for power saving. In order to get the complete benefit of mobile agent system, the system must be fault tolerant. In the context of mobile agents, fault-tolerance prevents a partial or complete loss of the agent. **Results:** Our system in mobile computing environment ensured that the agent arrived at its destination with result and performance of the system improved by the way of reduction in the response time. And also, the system allowed sending more requests by the way of creating many mobile agents without affecting the performance. **Conclusion:** Our research compared the performance of client-server architecture and fault tolerant mobile agent information retrieval system and proved that our system solved the limitations faced by the client server architecture. The system can also be extended to adhoc networks.

**Key words:** Client-server architecture, mobile agent, mobile host, mobile computing environment, fault tolerance, information retrieval system

---

### INTRODUCTION

Most of the present day Internet based information retrieval is web-based and employs the client-server architecture. It uses HTML-forms for user interface, with either Common Gateway Interface (CGI)-scripts or java-servlets for back end processing. The information is requested by the mobile host and the results are submitted back from the server (Lyu *et al.*, 2004).

Client server model has scalability and network bandwidth problems. To overcome that mobile agent based model is the suitable technology for the applications such as information retrieval system (Kannammal *et al.*, 2006). In order to get the complete benefit of mobile agent system, the system must be fault tolerant (Summiya *et al.*, 2006).

In this study, we discuss the performance of fault tolerant mobile agent system and traditional client-server architecture. Generally there are two approaches for fault tolerance mechanism i.e. check pointing and replication (Hevia and Vasa, 2000; Pleisch and Schiper, 2003; 2004). In our system, based on the scenario, the fault tolerant technique is adapted to increase the reliability of the system (Khan *et al.*, 2005).

**Mobile agent system:** Mobile agents departed from MH, migrate and search to get the information appropriate to user's requirements. After they finish their job, MA wants to migrate from Mobile Support Station (MSS) to MH and return the result to MH. The problem can be caused (i) if MH is currently disconnected or (ii) MH is moved to some other network. This can be solved by using replica in our

---

**Corresponding Author:** R. Punithavathi, Department of Information Technology,  
Chettinad College of Engineering and Technology, Karur (DT), Tamilnadu India 639114

proposed system. If MA tries to fail to the connection with MH, it is saved into the replica in MSS. Mobile server has the functionality for monitoring the connection status of the server to a mobile host when the server would like to return a mobile agent to the host. Mobile host has the functionality for transferring mobile agents according to the list of servers provided a directory service and gathering results from the agents. Server receives the mobile agent and processes user's request and produces result. It also dispatches the mobile agent after the processing is completed (Kim and Noh, 2003).

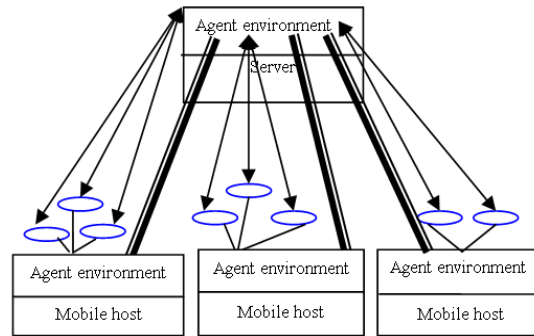


Fig. 1: Mobile agent architecture

### MATERIALS AND METHODS

**Mobile agent architecture:** We proposed the mobile agent architecture to provide application service in mobile computing environment as shown in the Fig. 1.

Even though, our aim of the paper is to compare the performance of client server architecture and fault tolerant mobile agent system, we cared about security aspects also.

When the user wants to enter the request, he/she has to give his/her signature for agent-server protection. A mobile agent is created and dispatched to the certificate authority (Ma *et al.*, 2003).

Depends on the user's allowed privileges; an agent certificate is created and dispatched to the mobile host. Now, if the user gives his request, it is accepted or rejected based on the details in the agent certificate. After the user request is accepted, mobile agent is dispatched to the mobile server. Mobile server gets the location of the server system corresponds to the user request from server selection database. Mobile agent is dispatched to the server system to do the required processing. It is retracted from the mobile server and replica agent is created. It also has the functionality for monitoring the connection with the mobile host when it would like to return a mobile agent. In scenario 1, if the mobile host is currently disconnected, the replica agent is set to the inactive status. After the mobile host is connected to mobile server, replica agent is set to active status and it is migrated from the mobile server to the mobile host. In scenario 2, mobile agent wants to jump to mobile host, but mobile host already departed from the region of the mobile server. Mobile server knows that mobile host had already moved to the cell of another mobile server using IETF's IP mobility Support (Brown and Singh, 1996).

The pseudo code for the above two scenarios are given in Fig. 2.

```

Scenario 1: Mobile host is not alive
Initiate MA with user request at Mobile Host;
Dispatch MA to the server;
Execute at server;
Retract MA at MSS;
If (MA@ MSS == True)
    Make replica of MA to make fault tolerant;
    If (IsMobileHost alive)
        Activate MA;
    Else
        Deactivate MA;
Dispatch all active MAs to Mobile Host;
Scenario 2: Mobile Host is in roaming
If (MA@ MSS == True)
    If (IsMobileHost roaming)
        Find COA of Mobile Host;
        Make replica of MA to make fault tolerant;
        If (IsMobileHost alive)
            Activate MA;
        Else
            Deactivate MA;
Dispatch all active MAs to COA;
    
```

Fig. 2: Pseudo Code for Scenarios

Mobile agent is jumped to the currently connected mobile server using IP mobility support. The remaining tasks are just the same with the first case. After the user entered his/her request, he/she is able to get the reply and continue the service, which is provided by the system even when the mobile is handoff or switched off either due to power saving or voluntary disconnection.

**Mobile host:** Mobile host has the functionality for transferring mobile agents corresponding to user's request in parallel and gathering results from the agents.

After sending the mobile agent, the mobile host may or may not be connected. If the mobile host connects to the network after the disconnection also, it will be available to gather the result from the agents.

**Server:** Server receives the mobile agent and processes user's request and produces result. It also dispatches the mobile agent after the processing is completed.

**RESULTS AND DISCUSSION**

To implement, we selected Aglet as a mobile agent platform after analyzing different mobile agent platforms such as Aglets, Grasshopper, Voyager etc (Trillo *et al.*, 2007). The proposed and implemented system in Java works properly in Aglet and Windows XP environment (Lange and Oshima, 1998). System overview of mobile agent architecture is shown in Fig. 3. Mobile agent architecture and user interface are installed in a mobile host. Agent technology allows simultaneous request from more users.

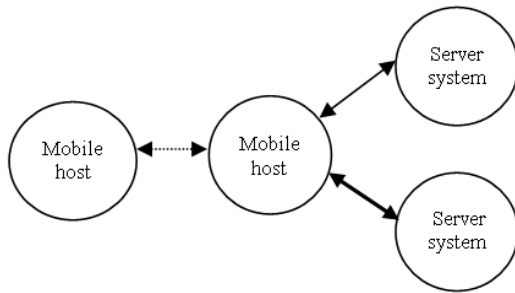


Fig. 3: System overview

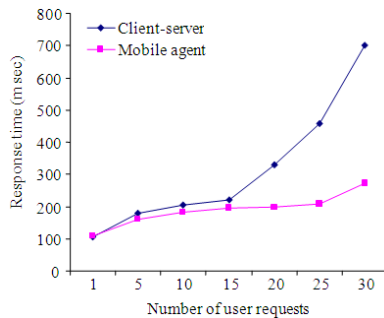


Fig. 4: Performance of client-server Vs mobile agent system

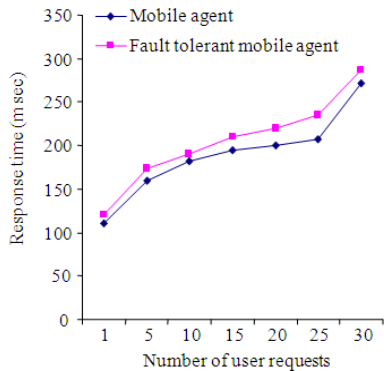


Fig. 5: Performance of mobile agent system Vs fault tolerant mobile agent system

As shown in the Fig. 4, the response time for smaller number of user requests is almost similar. But as the number of user requests increases, the performance of mobile agent system performs better.

As shown in the Fig. 5, the response time for all the requests in fault tolerant mobile agent system is higher than the mobile agent system. This is because of the creation of replica agent and the processing required for implementing fault tolerance.

In client server architecture, if the connection is lost during the transaction, the user has to send his request once again to get the results.

The proposed system overcomes that problem as well as the system is fault tolerant. And also, the system allows sending more user requests by the way of creating many mobile agents without affecting the performance.

**CONCLUSION**

Our fault tolerant mobile agent information retrieval system, which uses multi agent concept, provides a more flexible approach against existing client-server implementations. Our system supports better utilization of network bandwidth, improves user's response time avoiding network delays and thus increases network performance. In our study, we developed a fault tolerant system for mobile computing environment which is having wireless network. The system can be extended for adhoc network.

**REFERENCES**

Brown, K and S. Singh, 1996. A network architecture for mobile computing. Proceeding of the 15th Annual Joint Conference of the Networking the Next Generation, Mar. 1996, IEEE Computer Society, USA., pp: 1388-1396. DOI: 10.1109/INFCOM.1996.493087

Hevia, A. and A. Vasa, 2000. Fault tolerant protocols for mobile agents: A survey. <https://www-cse.ucsd.edu/classes/sp00/cse221/reports/hev-vas.ps>

Kannammal, A., V. Ramachandran and N.C.S.N. Iyengar, 2006. Reliable multi agent system for e-business applications. Acad. Open Internet J., 18. <http://www.acadjournal.com/2006/v18/part6/p2/>

Khan, Z.A., S. Shahid, H.F. Ahmad, A. Ali and H. Suguri, 2005. Decentralized architecture for fault tolerant multi agent system. Proceedings of Autonomous Decentralized Systems, Apr. 4-8, IEEE Computer Society, USA., pp: 167-174. DOI: 10.1109/ISADS.2005.1452043

- Kim, P.J. and Y.J. Noh, 2003. Mobile agent system architecture for supporting mobile market application service in mobile computing environment. Proceedings of IEEE Conference on Geometric Modeling and Graphics, July 16-18, IEEE Computer Society, USA., pp: 149-153.
- Lange, D.B. and M. Oshima, 1998, Programming and Deploying Mobile Agents with Aglets. Addison-Wesley, USA., ISBN: 020132582.
- Lyu, M.R., X. Chen and T.Y. Wong, 2004. Design and evaluation of a fault tolerant mobile agent system. IEEE Intell. Syst., 19: 32-38. DOI: 10.1109/MIS.2004.40
- Ma, Q., W. Hao, W. Li, M. Tu and I.L. Yen, 2003. PeAgent-a mobile agent system to support secure internet and web applications. Proceedings of Annual International Computer Software and Applications Conference, Nov. 3-6, IEEE Computer Society, USA., pp: 690. DOI: 10.1109/CMPSAC.2003.1245418
- Pleisch, S. and A. Schiper, 2003. Fault-tolerant mobile agent execution. IEEE Trans. Comput., 52: 209-222. DOI: 10.1109/TC.2003.1176987
- Pleisch, S. and A. Schiper, 2004. Approaches to fault tolerant and transactional mobile agent execution-an algorithmic view. ACM Comput. Surveys, 36: 219-262. DOI: 10.1145/1035570.1035571
- Summiya, S., K. Ijaz, U. Manzoor and A.A. Shahid, 2006. A fault tolerant infrastructure for mobile agents. Proceeding of the International Conference on Computational Intelligence for Modeling Control and Automation and International Conference on IEEE Intelligent Agents, Web Technologies and Internet Commerce, Nov. 2006, IEEE Computer Society, USA., pp: 235. DOI: 10.1109/CIMCA.2006.5
- Trillo, R., S. Ilarri and E. Mena, 2007. Comparison and performance evaluation of mobile agent platforms. Proceedings of the 3rd International Conference on Autonomic and Autonomous Systems, June 19-25, IEEE Computer Society, USA., pp: 41. DOI: 10.1109/CONIELECOMP.2007.66