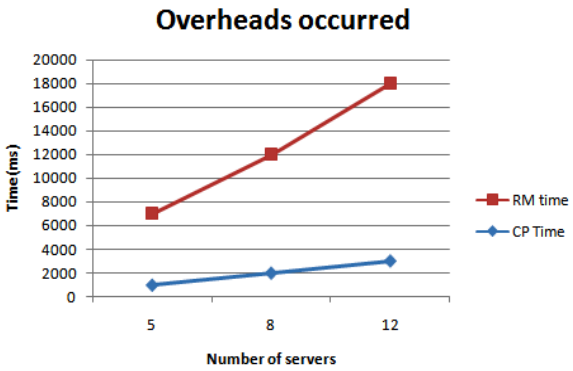
servers the agent should check point the data at the host server.

These overheads are measured in terms of Reliable migration time (RM time) and Check point time(CP time).RM time is the time taken by the agent to complete its itinerary making the faults and CP time is the time taken by agent to go back to the originator or host server to check point the data retrieved and address of the next host in the itinerary.

No. of Servers	5	8	12
CP Time	1000ms	2000ms	3000ms
RM Time	6000ms	10000ms	15000ms

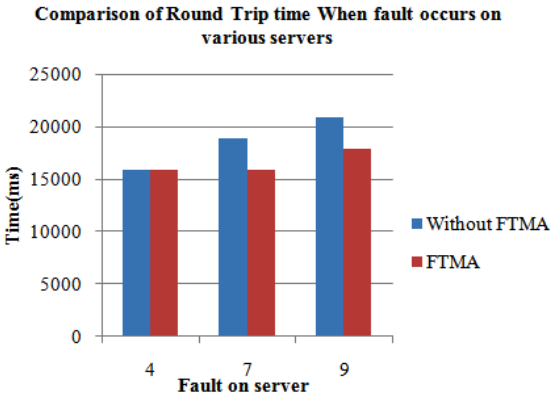


Experiment 2: Effect on Round trip time when fault occurs on any fault.

In this experiment we compare the round trip times of the agent to complete its itinerary when fault occurs on various nodes. The normal round trip time to complete an itinerary when fault occurs on any server RT_{wftm} is more than RT_{ftm} because when fault occurs on any server the host is notified about the fault,it sends a replicated agent or a copy of the original agent $MA_{rep,}$ which starts its itinerary from the beginning that is from the server S_1 again.

The above experiment has been performed on 12 different servers and for $RT_{ftm,}$ We have assumed checkpoint after every three servers.For implementation and result purpose an agent was manually killed by killing the thread of the agent on the particular server to create the fault. The time taken by the agent to visit again all the nodes which have been already visited by the MA that is the original agent adds to the overheads, so the time taken to complete the round trip increases in this case as compared to RT_{ftm}.

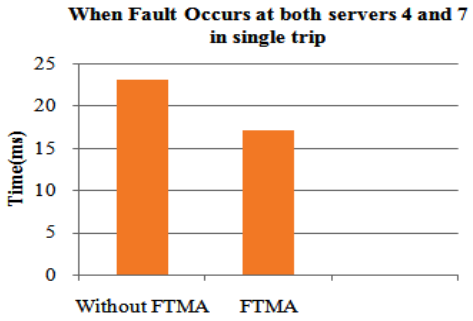
Initially both RT_{ftm} and RT_{wftm} are same when we assume a fault at server 4 but as we assume fault on any server after 4^{th} sever the RT_{ftm} decreases as compared to $RT_{wftm.}$ In case the replicated agent MA_{rep} starts its itinerary from the immediate check point before the faulty server so the there are no overheads to visit again all those servers which have been already visited by the original agent.compare the performance of both the RT_{ftm} and $RT_{wftm.}$ We have taken an itinerary of 12 servers, when fault occurs on the 4^{th} server of the itinerary both RT_{ftm} and RT_{wftm} are same. When fault occurs on 7^{th} server RT_{wftm} increases as compared to RT_{ftm} and same is the case when fault occurs on 9^{th} server.



Server number	4	7	9
Time Without FTMA (RT_{wftm})	16000	19000	21000
Time with FTMA (RT_{ftm})	16000	16000	18000

Experiment 3: Effect on Round trip time when fault occurs on multiple servers in a single trip.

In this experiment we compare the round trip times of the mobile agent without FTMA (RT_{wftm}) and with FTMA .



(RT_{ftm}) when fault occurs on multiple servers in single trip .For RT_{wftm} when agent MA moves on the servers in the itinerary and whenever it finds a faulty server, the replicated agent .MA_{rep} starts from the first server S_1 in the itinerary so the overheads of visiting those servers which have been already visited by original agent MA adds to the total round trip time. For RT_{ftm} the agent does'nt rollback and visits those servers again which are already visited by the by the original agent MA because in this when fault occurs the replicated agent MA_{rep} starts its itinerary from the checkpoint immediately before the faulty server.

Faults on multiple Servers	Server 4 and 7 in single trip
Time Without FTMA (RT_{wftm})	23000 ms
Time with FTMA (RT_{ftm})	17000 ms

6. Conclusions

In this paper, we have proposed a fault tolerance mechanism for the scenarios where the agent stops its execution due to fault on any server in the itinerary. Our approach makes use of check pointing, partial results and the address of last host visited is saved prior before the agent visits the next host in the itinerary.

Whenever a fault occurs, to mask the effect of the fault the host immediately sends the replicated copy of the original agent to the immediate check point before the faulty server. The in-depth analysis of this technique show us good results by improving the round trip time of the agent, Since after occurrence of fault, the replicated agent need not roll back to the first server as it starts moving from the checkpoint immediately before the faulty server.

Check pointing and saving the data repeatedly leads to increase in the communication overhead but for time sensitive applications the overhead may be bearable.

7. Future Work

From the future point of view, whenever an agent does not reaches the desired server due to network congestion the host assumes it to be failed and it sends a replicated copy of it mean while the original agent also reaches the destination which could lead to violation of exactly once property. So this approach should be developed further to avoid violation of exactly once property.

References

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

[2] T. Park, I. Byun, H. Kim, H.Y. Yeom, "The Performance of Checkpointing and Replication Schemes for Fault Tolerant Mobile Agent Systems,"In Proc. of 21st IEEE Symposium on Reliable Distributed Systems, 2002.

[3] K. Rothermel, M. Strasser, "A fault-Tolerant Protocol for Providing the Exactly-Once Property of Mobile Agents," Proc. of 17th IEEE Symposium on Reliable Distributed Systems, Los Alamitos, California, 1998.

[4] M. J. Wooldridge, N. R. Jennings, "Agent theories, architectures and languages: A survey," In ECAI-94 Workshop on Agent Theories, Architectures and Languages, Springer, August 1994.

[5] M. A. J. Jamali, H. E. Shabestar," A New Approach for a Fault Tolerant Mobile Agent System," Proc. of 12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing,2011.

[6] M. Strasser, K. Rothermel, "Reliability concepts for mobile agents," International Journal of Cooperative Information Systems, 1998.

[7] S. Pleisch, A. Schiper, "Modeling fault-tolerant mobile agent execution as a sequence of agreement problems," Proc. of the The19th IEEE Symposium of RDS, October 2000.

[8] A. Budi , I. Alexei, R. Alexander, "On using the CAMA framework for developing open mobile fault tolerant agent systems," Proc. of the 2006 international workshop on Software engineering for large-scale multi-agent systems, May 22-23, 2006,Shanghai, China.

[9] A. Rostami, H.Rashidi, M. S. Zahraie, " Fault Tolerance Mobile Agent System Using Witness Agent in 2-Dimensional Mesh Network," International Journal of Computer Science Issues, Vol. 7, Issue 5, September 201013

[10] S.G. Kumar, "Transient Fault Tolerance in Mobile Agent Based Computing," INFOCOMP Journal of Computer Science, Vol. 4, No. 4, pp. 1-11, 2005.

[11] S. Bagchi, K. Whisnant, Z. Kalbarczyk, R.K Iyer,"Chameleon: Adaptive Fault Tolerance Using Reliable, Mobile Agents,"Proc. of 16th Symposium on Reliable Distributed Systems, ACM New York, NY, USA, 1997.

[12] S. Pears, J. Xu, C. Boldyreff, "Mobile Agent Fault Tolerance for Information Retrieval Applications: An Exception Handling Approach," Proc. of The 6th International Symposium on Autonomous Decentralized Systems, 2003.

[13] D.B. Lange, M.Oshima," Mobile Agents with Java: The Aglet API", Baltzer Science Publishers, The Netherland.

[14] R.Kaur ,R. K.Challa R.Singh,"Integrated Mechanism to Prevent Agent Blocking in Secure Mobile Agent Platform System,"In Proc. of 2010 International Conference on Advances in Computer Engineering.