

Movement based Asynchronous Recovery System in Mobile Computing System

Dr. Amit Chaturvedi*, Tariq Ahmad Bhat** & Vikas Kumar***

*Head of MCA Department, Government Engineering College, Ajmer, Rajasthan, INDIA. E-Mail: amit0581@gmail.com

**Research Scholar, Bhagwant University, Ajmer, Rajasthan, INDIA. E-Mail: bhattariq7@gmail.com

***Head of Computer Science Department, Bhagwant University, Ajmer, Rajasthan, INDIA. E-Mail: vikasmca51@gmail.com

Abstract—Mobile Computing System involves mobile communication, mobile hardware and mobile software. It is a distributed system in which one of the processes is known as Mobile Node. The Mobile Computing System has some limitations such as Low Bandwidth of wireless network, lack of stable storage, mobility handling, disconnection of Mobile Nodes (MNs) and limited battery life. Researchers are working to overcome the limitations of asynchronous movement based recovery system, which is based on message logging & checkpointing, and to reduce recovery & transfer cost. In this review paper, we have taken these two objectives and proposed the solutions to overcome these limitations and directions for the future researches such as asynchronous recovery, independent checkpointing and handling of concurrent failures.

Keywords—Asynchronous Recovery; Checkpointing; Message Logging; Mobile Computing System; Movement based Recovery.

Abbreviations—Foreign Agent (FA); Home Agent (HA); Mobile Host (MH); Mobile Node (MN); Mobile Support Station (MSS).

I. INTRODUCTION

MOBILE Computing technology allows transmission of data via a computer without having to be connected to a fixed physical link. Mobile Computing addresses those applications and technical issues that arise when persons move within a specific region or travel between countries and continents. This proves to be the best solution to the biggest problem of business people on the move. A Mobile Computing System is a distributed system which consists of various distinct processes which are geographically separated and communicate with each other by exchanging messages [Prakash & Singhal, 1996; Biswas & Sarmistha, 2010; Gahlan & Kumar, 2010; Khatri, 2012]. In this system some processes are running on Mobile Node (MN)/Mobile Host (MH) that can move. We need to add Mobile Support Stations (MSSs) to communicate with mobile nodes (MNs). Mobile Support Stations (MSSs) communicate with each other through a wired network. The Mobile Node (MN) has some special properties, first it moves from one cell to another and hence, the mobility of MN must be properly and carefully handled [Yeom & Park, 2000; Khatri, 2012]. Also, MN is connected to the MSS via a wireless network. As long as an MN is connected to an MSS, the channel between them also ensures FIFO communication in

both the directions [Prakash & Singhal, 1996; Yeom & Park, 2000]. The Mobile Computing System has some limitations like low bandwidth of wireless network, lack of stable storage, limited battery life of MN, frequent disconnection and limited range of wireless networks [Sapna E. George et al., 2006; Yeom & Park, 2000; Woo et al., 2002; Biswas & Sarmistha, 2010].

II. SYSTEM MODEL

We are considering a mobile computing environment with a network consisting of stationary and mobile nodes. A Mobile Node (MN) [popularly known as Mobile Host (MH)] can change its location and network connection while computations are being processed. Message passing between two nodes is enabled via the Mobile Support Stations (MSS). The MSSs are reliable and are interconnected by a wired network. A MSS handles all communication to and from MNs within its area of influence known as a cell usually determined by the range of wireless transmission. Each MSS has a fixed wireless transmission range known as a cell and an MN can move from one cell to another. Assuming a hexagonal shape for each cell, a hexagonal network coverage model will be formed by a community of cells. Thus sending a message to another MN consist of two one-hop wireless

transmission between the sender and receiver MNs and their respective local MSS in addition to an arbitrary number of hops across the wired infrastructure between the sender's MSS and receiver MSS. At any time, a MN can be connected to at most one MSS. Channels are virtually lossless and they ensure First In- First Out (FIFO) communication and all communication takes place through messages. The interaction between the MN and the network infrastructure most relevant to failure and recovery are handoffs, disconnect and reconnect. Considering that the MN's disk cannot be assumed to be stable, each MSS is equipped with enough volume stable storage to store the state and log information for all the MNs currently in its cell as well as those that were recently in its cell. However due to the fact that MSSs must support multiple concurrent MNs, this storage must be efficiently managed.

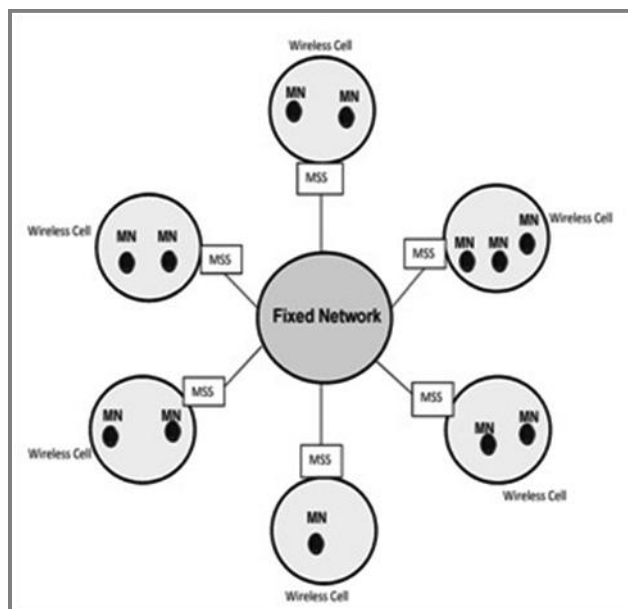


Figure 1: Mobile Computing System Model

III. ASYNCHRONOUS MOVEMENT BASED RECOVERY SYSTEM

When any Mobile Node (MN) gets failure, all it has to do is to just rollback to its latest checkpoint, independent of others. No other Mobile Node (MN) need to rollback together. Whenever a Mobile Node recovers from failure, first of all it obtains the latest checkpoint. In this paper, we have analyzed a movement based asynchronous recovery system based on message logging and checkpointing for mobile computing systems. With the independent checkpointing and message logging, each MN now can perform the rollback – recovery independently. By this recovery, we mean that only the MN which has failed rolls back to its latest checkpoint and replays the logged messages to reach the consistent recovery line, and no other Mobile Nodes (MNs) need to rollback together.

Many recovery schemes have been proposed for the distributed systems, however, these schemes cannot be directly used in the mobile computing environment, as most

of the other distributed services. Especially, the following properties of the mobile computing system give the difficulties on the direct use of the existing schemes: Low bandwidth; Low disk space in MNs; Mobility handling & Disconnected operation [Yeom & Park, 2000; Biswas & Sarmistha, 2010; Tuli & Kumar, 2011; Khatri, 2012]. The scheme proposed by Khatri (2012) and Park (2001) suggests that the MN carrying its recovery information to the nearest MSS and can recover immediately in case of a failure. When the MN moves within the cell range the recovery information of MN remains at the MSS where the information was first saved. However, if the MN does handoff, it transfers the recovery information into the nearby MSS. Thus with the help of this scheme the transfer and recovery cost is reduced. The focus is given on reducing transfer and recovery cost because when MN moves across the cells, its message log gets distributed over a number of MSSs. In this paper we focused on as MN moves across the cells and it is allowed only when it moves out of a particular range.

IV. MESSAGE LOGGING

Message-logging is very popular for building systems that can tolerate process crash failures. Message logging and check-pointing can be used to provide fault tolerance in distributed systems in which all inter-process communication is through messages. Each message received by a process is saved in message log on stable storage. No coordination is required between the checkpointing of different processes or between message logging and check-pointing. The execution of each process is assumed to be deterministic between received messages, and all processes are assumed to execute on fail stop processes. When a process crashes, a new process is created in its place. This new process is given by the appropriate recorded local state, and then the logged messages are replayed in the order the process originally received them. All message-logging protocols require that once a crashed process recovers, its state needs to be consistent with the states of the other processes. Each mobile node N_i takes a checkpoint independent of others. Every time on a new checkpoint, it increments the sequence number of checkpoint. After taking checkpoint say C_i^a , it sends checkpoint with its identifiers $[i, a]$ to its current mobile support station say MSS_p . Also it delivers message sequence number it received last, before taking the checkpoint. When a message M comes for node N_i , MSS_p first stores the M in the message log with sequence number „sn“, and then delivered to N_i . MSS_p also logs the mobility related messages like join, leave, disconnect and reconnect, received from the MNs [Yeom & Park, 2000; Park et al., 2001; Woo et al., 2002; Khatri, 2012]. In the movement based recovery system, the checkpoint and message logs need to be moved into a new MSS during the handover only when the moving distance of the MSS to which MN is connected from the MSS carrying the latest checkpoint exceeds a certain threshold value 'Tv'. As an MN may be lost or permanently damaged, hard drives on Mobile Nodes are not generally considered stable storage.

Therefore they are not suitable as the only location for storing checkpoints or message logs. Traditional checkpointing and message logging algorithms are not directly applicable under such conditions. Previous proposals have suggested that checkpoints be sent back to Home Agents (HA). Others have proposed that stable storage on Mobile Support Stations (MSS) be used for checkpoints and message logs. Because checkpoints and/or message logs are stored on different MSSs as an MN moves from cell to cell, the organization of the distributed process state information is important for successful recovery. A MH is associated with a Home Agent (HA) when it is within its home network, and with a Foreign Agent (FA) when within a foreign network [Neves & Fuchs, 1997; Yao et al., 1999; Chi- Y. Lin et al., 2003].

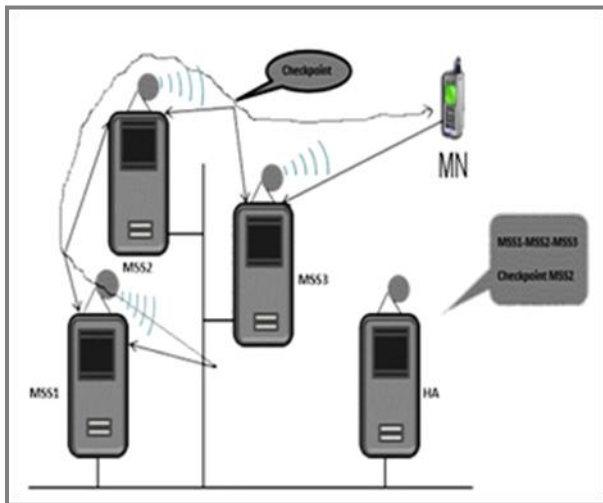


Figure 2: Message Logging

V. CHECKPOINTING

Checkpoint is defined as a designated place in a program where normal processing of a system is interrupted to preserve the status information. A checkpoint is a local state of a process saved on stable storage. It is the process of saving status information. Mobile computing systems often suffer from high failure rates that are transient and independent in nature. To add reliability and high availability to such distributed systems, checkpoint based rollback recovery is one of the widely used techniques for applications such as scientific computing, database, telecommunication applications and mission critical applications. A checkpoint is a local state of a process saved on stable storage [Rachit & Kumar, 2010; Rachit & Kumar, 2010; Khuneta et al., 2011; Xiong et al., 2013]. Local checkpoint is an event that records the state of a process at processor at a given instance. Checkpoint may be local or global depending on taking the checkpoints. Local checkpoint is a process may take a local checkpoint any time during the execution. The local checkpoints of different processes are not coordinated to form a global consistent checkpoint. And Global Checkpoint is a collection of local checkpoints, one from each processor. There are 5 types of checkpoints. Uncoordinating, coordinating, Communication Induced [Khunteta & Praveen,

2010], Diskless Checkpoint, and Double Checkpoint. These types are grouped into two categories: Synchronous and Asynchronous checkpointing [Xiong et al., 2013].

VI. REVIEW ON RELATED WORK

Woo et al., (2002) highlighted an asynchronous recovery scheme in which they tried to cope with space problem of the MHs and to reduce the overhead of stable message logging, also they identified that task of message logging and dependency tracking is performed by the MSSs. The MN (MH) only carry a minimum information so that its mobility can be properly traced by the MSSs. They showed that this scheme requires a small failure free overhead and cost of unnecessary rollback. Khatri (2012) supposed if transferring message log and checkpoints with every handover, it can put extra overhead on network bandwidth and also increases transfer cost, thus here the author tried to restrict transfer of recovery information and allowed only when MN moves out of particular range.

Yeom & Park (2000) presented an asynchronous recovery scheme based on optimistic message logging, here the task of message logging and dependency tracking are performed by the MSSs. And finally the concluded that with the asynchronous recovery, the MHs in the system are never involved in any kind of coordination, and they can take the checkpoints as they want, also this can handle multiple and concurrent failures at the same time. Miraclin Joyce Pamila & Thanushkodi (2009) focused to reduce total cost for recovery from failure, also ensure to reduce handoff cost, log retrieval cost and failure recovery time when mobility rate of mobile node is very high.

Park et al., (2001) expressed message logging and checkpoint based recovery scheme, in which they considers both failure - free operation cost and the recovery cost. Also mentioned when mobile host moves, through this scheme they tried to reduce the recovery and transfer cost. Yi-Wei Ci et al., (2008) improved the scheme by migrating only partial recovery information of a MH when a MH moves out of the range. It means that recovery information of MH which is contained in some MSS due to mobility, is mapped to another set of MSSs. These MSSs are given by route function. The main advantage of this scheme is that one MSS is not burdened by transferring all the information to it.

Sapna E. George et al., (2006) presented an algorithm in which each MH maintains a counter which is incremented by 1 when MH performs a handoff to another cell. Once this counter becomes greater than a predefined value, a checkpoint is taken. This counter depends on the user's mobility rate, failure rate and log arrival rate. Each MH also maintains a set of MSS which stores MH's log after latest checkpoint. When a MH performs a handoff, a new MSS is added to this set if MH sends at least one message in the cell to the new MSS and if MSS has already not been added to the set. MSS logs messages before sending them to the destination. These messages are retrieved from MSS to recover a failure free state of MH after failure occurs. Once a

new checkpoint is successfully taken by MH, set of MSS stored in MH is cleared and a message is sent to the MSS in the set to clear the log related to MH. Thus, storage overhead is reduced. Alvisi & Marzullo (1998) have provided an in-depth treatment of message logging in.

Acharya & Badrinath (1994) describes uncoordinated checkpointing, where multiple MHs can arrive at a global consistent checkpoint without coordination messages. However, neither it takes into account how failure recovery is achieved nor does it address the issue of recovery information management in the face of MH (MN) movement. Rao & Vin (1998) compared the cost of recovery for different message logging methods.

In Higaki & Takizawa (1998) proposed hybrid checkpoint-recovery protocol for mobile systems was by As a mobile host moves between cells, it leaves an agent process on each mobile support station on its itinerary. During recovery, processes on fixed hosts recover from consistent checkpoints and processes on mobile hosts restart from their own checkpoints and roll to a state that is consistent with those on fixed hosts with the help of agent processes.

Pradhan et al., (1996) presented a recovery scheme that combines various checkpointing and logging schemes for different mobile environments. They describe two uncoordinated checkpoint protocols, no-logging and logging and three strategies for recovery information management due to MH mobility, pessimistic, lazy and trickle strategies. In Manivannan & Singhal (1996), authors proposed a low overhead recovery scheme based on a communication induced checkpointing, which allows the processes to take checkpoints asynchronously and uses communication-induced checkpoint coordination for the progression of the recovery line. The scheme also uses selective pessimistic message logging at the receiver to recover the lost messages. However, the recovery scheme can handle only a single failure at a time.

VII. CONCLUSION AND FUTURE SCOPES

For reducing recovery and transfer cost, the algorithms of message logging and checkpointing is a good solution because as MN moves across the cells, its message log becomes distributed over a number of MSSs. If we transfer checkpoints and message log with every handoff then the transfer cost will become very much significant and also puts extra overhead on the network bandwidth.

For movement based asynchronous recovery system, the MN moves within a certain range, recovery information of MN is not moved. However, if the MN moves out of range it transfers the recovery information to the nearby MSS. The scheme proposed in Park et al., (2001) controls the transfer cost as well as the recovery cost. The performance of the proposed scheme shows that this scheme provides various levels of failure-free operation cost and recovery cost by adjusting movement factors.

As a result the reducing recovery and transfer cost of transferring data/information and movement based

asynchronous recovery system with failure-free operation cost and recovery cost are the two very important study areas for future research.

ACKNOWLEDGMENT

We feel grateful to the referees for their valuable suggestions that have helped immensely in preparing the revised manuscript.

REFERENCES

- [1] A. Acharya & B.R. Badrinath (1994), "Checkpointing Distributed Applications on Mobile Computers", *Proceedings of 3rd International Conference on Parallel and Distributed Information System*, Austin, Texas, Pp. 74–80.
- [2] D.K. Pradhan, P. Krishna & N.H. Vaidya (1996), "Recoverable Mobile Environment: Design and Trade-Off Analysis", *Annual Symposium on Fault Tolerant Computing*, Pp. 16–25.
- [3] D. Manivannan & M. Singhal (1996), "Failure Recovery based on Quasi-Synchronous Checkpointing in Mobile Computing Systems", *Department of Computer and Information Science, The Ohio State University*.
- [4] R. Prakash & M. Singhal (1996), "Low Cost Checkpointing & Failure Recovery in Mobile Computing System", *IEEE Transactions on Parallel & Distributed Systems*, Vol. 7, No. 10, Pp. 1035–1048.
- [5] N. Neves & W.K. Fuchs (1997), "Adaptive Recovery for Mobile Environments", *Communications of the ACM*, Vol. 40, No.1, Pp. 68–74.
- [6] L.A.S. Rao & H. Vin (1998), "The Cost of Recovery in Message Logging Protocols", *Proceedings of the 17th Symposium on Reliable Distributed Systems*, Pp. 10–18.
- [7] H. Higaki & M. Takizawa (1998), "Checkpoint-Recovery Protocol for Reliable Mobile Systems", *Proceedings of the 17th Symposium on Reliable Distributed Systems*, Pp. 93–99.
- [8] L. Alvisi & K. Marzullo (1998), "Message Logging: Pessimistic, Optimistic, Causal, and Optimal", *IEEE Transactions on Software Engineering*, Pp. 149–159.
- [9] B. Yao, Kuo-Feng Ssu & Fuchs W. Kent (1999), "Message Logging in Mobile Computing", *29th Annual International Symposium on Fault-Tolerant Computing (FTCS-29)*, Pp. 294–301.
- [10] H.Y. Yeom & T. Park (2000), "An Asynchronous Recovery Scheme based on Optimistic Message Logging for Mobile Computing Systems", *20th International Conference on Distributed Computing Systems (ICDCS 2000)*, Pp. 436–443.
- [11] T. Park, N. Woo & H.Y. Yeom (2001), "An Efficient Recovery Scheme for Mobile Computing Environment", *Proceedings 8th International Conference on Parallel & Distributed Systems (ICPADS)*, Pp. 53–60.
- [12] N. Woo, T. Park & Y.Y. Heon (2002), "An Efficient Optimistic Message Logging Scheme for Recoverable Mobile Computing Systems", *IEEE Transactions on Mobile Computing*, Vol. 1, No. 4, Pp. 265–277.
- [13] Chi- Y. Lin, Szu-Chi-Wang & Sy-Yen Kuo (2003), "An Efficient Time based Checkpointing Protocol for Mobile Computing Systems over Mobile IP", *Mobile Networks and Applications - Mobile networking through IP*, Vol. 8, No. 6, Pp. 687–697.
- [14] Sapna E. George, Ing-Ray Chen & Ying Jin (2006), "Movement-based Checkpointing and Logging for Recovery in Mobile Computing Systems", *Proceedings of the 5th ACM International Workshop on Data Engineering for Wireless and Mobile Access*, Pp. 51–58.

- [15] Yi-Wei Ci, Zhan Zhang, De- Ching Zuo, Zhi- Bowu & Xiaa-Zong Yang (2008), "Area Difference based Recovery Information Placement for Mobile Computing System", *14th IEEE international Conference on Parallel and Distributed Systems*, Pp. 478–484.
- [16] J.C. Miraclin Joyce Pamila & K. Thanushkodi (2009), "Log Management Support for Recovery in Mobile Computing Environment", *International Journal of Computer Science and Information Security (IJCSIS)*, Vol. 3, No. 1.
- [17] P. Gahlan & P. Kumar (2010), "A Low Overhead Minimum Process Coordinated Checkpointing Algorithm for Mobile Distributed System", *International Journal of Computer Applications*, Vol. 3, No. 1, Pp. 17–21.
- [18] S. Biswas & N. Sarmistha (2010), "Mobility based Checkpointing Protocol for Mobile Computing System", *International Journal of Computer Science & Information Technology (IJCSIT)*, Vol. 2, No. 1, Pp. 135–151.
- [19] A. Khunteta & K. Praveen (2010), "An Analysis of Checkpointing Algorithms for Distributed Mobile Systems", *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 02, No. 04, Pp. 1314–1326.
- [20] G. Rachit & P. Kumar (2010), "A Non-blocking Coordinated Checkpointing Algorithm for Mobile Computing Systems", *International Journal of Computer Science Issues (IJCSI)*, Vol. 7, No. 3, Pp. 41–46.
- [21] G. Rachit & P. Kumar (2010), "A Review of Fault Tolerant Checkpointing Protocols for Mobile Computing Systems", *International Journal of Computer Applications*, Vol. 3, No. 2, Pp. 8–19.
- [22] R. Tuli & P. Kumar (2011), "A Survey & Performance Analysis of Checkpointing and Recovery Schemes for Mobile Computing", *International Journal of Computer Science & Communication*, Vol. 2, No. 1, Pp. 89–95.
- [23] A. Khuneta, P. Sharma & R. Garg (2011), "New & Efficient Low Overheads Algorithm for Mobile Distributed Systems", *International Conference & Workshop on Emerging Trends in Technology (ICWET 2011) – TCET*, Mumbai, India.
- [24] Y. Khatri (2012), "Distance based Asynchronous Recovery Approach in Mobile Computing Environment", *International Journal of Distributed and Parallel Systems (IJDPS)*, Vol. 3, No. 3.

- [25] A. Xiong, Guoqiong Liao, Guoqiang Di & Jiali Xia (2013), "A Review of Checkpointing Strategies for Mobile Networks", *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*, Atlantis Press, Paris, France, Pp. 1579–1582.



Dr. Amit Chaturvedi, Head, MCA Department, Government Engineering College, Ajmer have completed Ph.D. (Computer Science) in March, 2012. He has more than 12 years of teaching experience. He have published around 26 research papers in national and international Journals in the area of mobile computing system, spectrum requirement estimation for 4G and cryptography and always ready to teach the subjects to his students, which he does with great finesse.



Tariq Ahmad Bhat received the bachelors degree (with Computer Application and Applied Mathematics) from University of Kashmir, Srinagar, India in 2008. He received the MCA degree from BGSB University, Rajouri, Jammu & Kashmir, India in July 2012. He is presently the lecturer at NCPUL, India and a research scholar. His research interests include Mobile Computing, Cryptography, Parallel and Distributed Computing.



Vikas Kumar, Head, Computer Science Department, Bhagwant Univ, Ajmer has completed M. Tech. (CSE). He is having 9 years of teaching experience.