Recent Trends on Consistent Global Snapshot Algorithms for Distributed Mobile Environments

Vijaya Kapoor
Shri Venkateshwara University
Gajraula, UP (INDIA)

Parveen Kumar
Amity University
Gurgaon, Haryana (INDIA)

ABSTRACT
Mobile computing allows omnipresent and incessant access to computing resources while the users is on the move In the recent years the mobility issues is one of the most significant development and effect the converging areas of computing and telecommunications The way to compute and communicate is changing rapidly. Failure that are rare with fixed hosts become common , host disconnection and mobility makes the fault detection & message coordination difficult. The various distributed applications of mobile / wireless environments are e-commerce , national defense , emergency & disaster management, telecommunications , the background studies report that snapshot is the technique used to tolerate failures in distributed system and thus well suited for mobile environments. There are different approaches for failure free executions of a nodes providing fault tolerance to the existing distributed system without fault tolerance, the application programs or software executing in a multiprocessor environment in a distributed system could fail entirely if even a single process executing part of it. An efficient recovery mechanism for distributed mobile environment is required to maintain the continuity of computation in the event of node failure. During the study it has been analyzed , to meet the requirement of mobile environment the recovery algorithm should meet the low energy consumption, reduced storage overhead having low communication & band width constraints.

Keywords
Fault tolerance, Coordinated snapshot, Message logging and Mobile Distributed Systems

1. INTRODUCTION
System with more than one processor are known as multiprocessor systems the term distributed system , parallel system and multiprocessor system are used interchangeable. As the number of processor increasing the system failure probability is also high , it has been found that over 80 % of the failure in the system are transient & intermittent . parallel & distributed systems comprises of computing and communication & storage resources where execution speed, storage capacity & communication band width , reliability & resilience are critical issues. The fault causes can be due to environmental inferences, software bugs , physical failure of components , violation in security , operator error. Fault tolerance in multiprocessor systems can be addressed at two levels. At network level faults can be handled by using efficient fault tolerant channel allocation algorithms and fault tolerant location management .At operating system level, Snapshot & recovery techniques are useful. While designing a protocol having mobile hosts the limited &vulnerable MH local storage, cost of location, energy consumption ,open systems , low bandwidth & high channel contention, voluntary disconnection /connection are the Constraints of snapshot have to be Taken into account [2]. Frequency of snapshot, contents of snapshot, methods of snapshot are the aspects of snapshot . The frequency of snapshot is essential in which the substantial computation will not lose, which necessitates frequent snapshot and consequently significant overhead. The number of snapshots initiated should be such that the cost of information loss due to failure is small [2]. The content of snapshot includes code, data and stack segments along the environment and the register contents. The environment has the information about the various files currently in use. The Methods of snapshot used in multiprocessor systems should incorporate explicit coordination unlike uni processor systems. Two overheads such as coordination overhead, context saving overhead are the overheads in Snapshot algorithms

In coordination overhead special messages & piggybacked information with regular messages are used to obtain coordination among processes while in context saving overhead, the time taken to save the global context of a computation is defined as context saving overhead.

2. RELATED WORK
Kamani p. [1] designed and implemented a new approach for adding fault tolerance to distributed mobile computing environment using a new general model based on the token ring methodology . The methods used here are transparent and add only the little overhead to the existing system . The proposed algorithm out performs the existing fault tolerance algorithm. The goal is to develop a non blocking coordinated snapshot protocol well suitable for MCS. The snapshot request is passed as a token from higher priority process to the lower priority process. The token comprises of dependency information of the current process. The processes which receives the token, takes a snapshot & updated information is stored in the token ,with the deletion of information not required in future. When the token becomes empty , the process informs it to the coordinator process by a special trigger message . it is assumed that each process has a priority value which is based on the nature of the application. Token reaches from the last process to the initiator and it is end of snapshot . Two versions of token ring DTRC & STRC are used.

Sneha & Ramtek [3] presented a non blocking snapshot coordination , a remote snapshot technique is preferred in wireless environments in which there is a remote snapshot server where snapshot data of mobile consumer device is kept instead of the mobile consumer device[27]. The proposed methods also deals with optimal CI to minimum power consumption in wireless remote snapshot environments by

considering environmental parameters as the battery power is most critical resource for mobile consumer device, it is significant to identify optimal CI that reduces power consumption . The proposed algorithm presented an approximation method to the energy optimal CI to minimize the expected energy expenditure using Poisson method for mobile device.

Sarmistha Neogy[4] in the paper presented a proposal for achieving fault tolerance in wireless as well as for mobile computing system . The algorithm proposed used a coordinated snapshot strategy with a well known technique triple modular redundancy for wireless environments known as WTMR where two MSSs and a MH will act as replicas and second using snapshot to achieve rollback recovery from fault . This approach causes 1)lowering of the network traffic 2) tackles the situation of intermittent failure due to disconnection, wireless channel saturation with traffic low power of devices in wireless / mobile computing environments 3) reducing power & communication overheads using the proposed technique of the WTMR snapshot , an architecture for a wireless system is possible without any extra overhead added . The approximated cost of this scheme remains low as compared with traditional coordinated snapshot & approach taken by Byun & kim [5]

The total cost is c + (mss-1) *(cr + mss-1) * MH *M*Cr. It can handle 2 faults simultaneously.

l.kumar[6] presented an efficient low cost synchronous min process snapshot algorithm which fit into mobile environment having following characteristic:

1. Maintains exact dependencies by direct dependencies & transitive dependency

2. It piggybacks the message information onto normal messages

3 To reduce searching cost, it maintain the information related to updation of location of mobile host.

4 To reduce the useless snapshots , the trigger set is send after taking forced snapshot.

5 To Maintaining record of multiple forced snapshot after conversion of forced snapshot into permanent one.

The algorithm makes use of snapshot interval CI .As the number of MHs increases , the searching cost is slightly increasing 2 times as compared to Cao-Singhal algorithm. It uses min set so, less number of MH of in active mode consumes more power . In this algorithm , MHS take less Number of snapshots nearest to minimum , which causes the efficient resource utilization of mobile system , it requires very minimum interaction between originator MH & others. Algorithm use simple data structure due to all above said features made this algorithm more suitable for mobile computing environment

Parveen Kumar & Preeti Gupta [8] to optimize the blocking of processes & minimum loss of snapshot effort the algorithm proposed a mechanism which delay the processing of selective message at the receiver end, only during the snapshot period. Here min process algorithm similar to cao-singhal , proposed scheme keeps track of direct dependency of processes. Originator collects direct dependency vector , compute minimum set may receive some message which causes addition of new members to already computed minset, in order to keep the computed min set as it is The two classification of message received during blocking period.1)Message that change the dependency set of the receiver process.2) Message that do not change the dependency set of the receiver process. Solution to 1) type message need to be delayed 2) type of problem is normal.

Helary [9] proposed a snapshot algorithm that uses the concept of message waves. A wave is a flow of control messages such that every process in the system is visited exactly once by a control message and at least one process in the system can determine when this flow of control messages terminates. Wave sequences may be implemented by various traversal structures such as a ring. On visited by the wave control message, a process begins recording the local snapshot

Wang and Fuchs lazy snapshot coordination [10] proposed a coordinated snapshot scheme in which they incorporated the technique of lazy snapshot coordination into an uncoordinated snapshot protocol for bounding rollback propagation. Recovery line progression is made by performing communication induced snapshot coordination only when predetermined consistency criterion is violated. The notation of laziness provides a tradeoff between extra snapshots during normal execution and average rollback distance for recovery.

Higaki-Takizawa Hybrid Snapshot [22] proposed a hybrid snapshot protocol for mobile computing system. It is a hybrid of independent and coordinated snapshot. The mobile Hosts take the snapshot independently whereas the fixed stations take the coordinated snapshot. The messages sent and received by Mobile Hosts are stored in corresponding MSS. The algorithm has two demerits. First, using independent snapshot protocol may cause the domino effect. Second, coordinated and independent snapshot protocols perform independently in mobile support stations and mobile hosts, and do not negotiate with each other. Therefore, it is difficult to obtain consistent global snapshots.

Rao and Naidu Snapshot Scheme with selective sender based message logging [23] proposed a new snapshot protocol combined with selective sender based message logging .The protocol is free from the problem of lost messages .The term ‘selective’ means that messages are logged only within specified interval known as active snapshot interval, in this manner reducing message logging overhead .All the processes take snapshots at the end of their respective active snapshot intervals forming a consistent global state Outside the active snapshot interval there is no snapshot of process state. This protocol minimizes different overheads i.e. snapshot overhead, recovery overhead, blocking overhead. In this protocol there exists $P_{initiator}$,which coordinates with all the processes to take a consistent global snapshot. $P_{initiator}$ is responsible for invoking the snapshot operation periodically .It sends control messages ,prepare snapshot and take snapshot messages to all other processes .Here the concept of active interval is introduced .The time that elapses between two events sending ‘prepare snapshot and ‘take snapshot’ messages by $P_{initiator}$ to all the process is referred to as an active interval of $P_{initiator}$. Similarly ,the time that elapses between two events of receiving ‘prepare snapshot and ‘take snapshot’ messages by any process is referred to as an active interval of that process .The maximum transmission delay incurred by any message to reach the destination is assumed to be t . It is also assumed that T>3t,Since snapshot interval is obviously greater than active interval and length of active interval is bound to be at least ‘3t’ to survive the transmission delay of control messages and to enable logging of computational messages. If any process wants to send a message inside the active interval, To continue with the process execution, initially system has to be

logged. The proposed protocol enables the system to handle the lost messages. There are two counters maintained by each process namely message received count (MRC) and message send count (MSC). These counters are initialized to zero at the start of active interval .The counts of MRC and MSC are incremented only within the active interval .Outside the active interval there will not be any change in their values. At time K*T + 3*t ,the initiator sends 'take' snapshot' signal to other processes .Afterwards it takes the snapshot and exits from active interval. In response to take snapshot, the rest of process will take snapshot and exits from the respective active intervals. These snapshots forms consistent global state .After exiting from the active interval, all the processes follow their normal operation .It implies that there is no snapshot and logging of messages outside the active snapshot interval .In case of any failure ,every process rolls back to its latest snapshot and necessary messages will be replayed from stable storage to reconstruct the previous state of the whole system .If failure occurs after all processes exited from their respective active intervals, then the application rolls back to the latest consistent global state namely 'g'; else if failure occurs before one of the processes exits from their respective active intervals ,then the application rolls back to previous global state namely 'g-1'.

Juang- Venkatesan [24] proposed an asynchronous snapshot scheme for distributed systems. Since their algorithm is based on asynchronous snapshot. The main issue in the recovery is to find a consistent set of snapshots to which the system can be restored. The recovery algorithm achieves this by making each process keep track of both the number of messages it has sent to other processes as well as the number of messages it has received from other processes. Recovery may engage numerous iterations of rollbacks by processes whenever a process rolls back. It is obligatory for all other processes to find out if any message sent by the rolled back process has become an orphan message Orphan messages are discovered by comparing the number of messages sent to and received from adjoining processes. If the number of messages received by processor P1 from process P1 is greater than the number of message sent by process P2 to process P2 according to the current states of the processes. Then one or more messages at process P1 are orphan messages. In this case process P1 must roll back to a state where the number of messages received agrees with the number of messages sent.

Basu et al Mobility Based Scheme [25] proposed a snapshot algorithm in which they took the mobility of the nodes as the basis of the algorithms. They considered both location distance between MSSs and mobility to take snapshot decisions. They showed that the recovery probability increases as the failure rate increases in a distributed mobile system and the movement of mobile hosts does not affect the recovery time much. Wireless media is vulnerable to different types of attacks. There may be various malicious nodes trying to enter into secured network. In this algorithm, an additional attempt of incorporating security is made by authentication of a Mobile Host when it enters into the network, by using public key method.

Bidyut Gupta Shahram Rahimi and Ziping Liu [14] had presented a non-intrusive coordinated snapshot algorithm suitable for mobile environments. The merits make the proposed algorithm appropriate for mobile distributed computing systems are following merits: (a) The proposed algorithm does not take any temporary snapshot and hence the overhead of converting temporary snapshot to permanent snapshot is eliminated.(b) The proposed algorithm does not exploit mutable snapshots. Hence the overhead of converting them to permanent ones is eliminated. (c) Their algorithm allow any process to take zero snapshot. It uses very few control messages and participating processes are interrupted less number of times[14]. Algorithm Non-intrusive produces a CGS(consistent global state) of the system. In first two steps of algorithm for the initiator process P_i identifies all application messages received from different processes that might become orphan if it takes a snapshot by looking at its dependency vector. The initiator then sends primary snapshot requests to all those processes that have sent at least one message to it asking them to take their respective snapshot. Consider the pseudo code for any process P_j. Process P_j makes certain that all processes from which it has received messages also take snapshot so that there are refusal of orphan messages that it has received In second else if block of the pseudo code process P_j first takes its snapshot if needed then processes the piggybacked application message[14] .Hence such a messages cannot be orphan .Hence the algorithm generates a consistent global state of the system.

Lai and Yang's[17] global snapshot algorithm for non-FIFO systems is based on two observations on the role of a marker in a FIFO system. The first observation is that a marker ensures that condition **C2** is satisfied for LS_i and LS_j when the snapshots are recorded at processes p_i and p_j respectively. The Lai-Yang algorithm accomplish this role of a marker in a non-FIFO system by using a coloring scheme on computation messages that works as follows :Every process is initially white and turns red while taking a snapshot. The corresponding of the "marker sending rule" is executed when a process turns red. Every message sent by a white (red) process is colored white (red). Thus a white (red) message is a message that was sent before (after) the sender of that message recorded its confined snapshot. Every white process takes its snapshot at its ease, but no later than the moment it receives a red message.

Thus, when a white process receives a red message, it records its local snap-shot before processing the message. This ensures that no message sent by a process after recording its local snapshot is processed by the destination process before the destination records its confined snapshot. Thus, an explicit marker message is not required in this algorithm and the "marker" is piggybacked on computation messages using a coloring scheme.

The second observation is that the marker informs process p_j of the value of [send (m_{ij}) send (m_{ij}) $\in LS_j$} so that the state of the channel C_{ij} can be computed as transit (LS_i, LS_j). The Lai-Yang algorithm fulfills this role of the marker in the following way: Every white process records a history of all white messages sent or received by it along each channel. When a process twist red, it sends these histories along with its snapshot to the initiator process that collects the global snapshot. The originator process evaluates transit (LSi, LSj) to figure out the state of a channel Cij as given below:

SCij=white messages sent by pi on Cij – white messages received by pj on Cij ={mij} send (mij) ∈ LSi} – {mij} rec (mij) ∈ LSj}Condition C2 holds because a red message is not included in the snapshot of the recipient process and a channel state is the difference of two sets of white messages. Condition C1 holds because a white message mij is included in the snapshot of process Pj if Pj receives mij before taking its snapshot. Otherwise, Mij is included in the state of channel

3. CONCLUSION

Different approaches of snapshot are reviewed, few with their merits & demerits are described in table 1. It is reasonable to say that stable storage latency is the major source of overhead, as the latency of network communication decreases, network communication overhead becomes a minor source of overhead and It has been found that selective process coordinated snapshot is a suitable approach to introduce the concept of fault tolerance in mobile distributed system transparently At last we conclude that for a consistent global Snapshot, the algorithms has the following enviable features:

1. The time taken by snapshot algorithms should be selective during failure free run.
2. There should be minimum Domino effect or Rollback propagations.
3. Selective rollback should be possible.
4. Resources requirement for snapshot should be selective.
5 Recovery should be fast in event of failure .Availability of consistent global state in stable storage expedite recovery.

Table 1. Analysis of few more Snapshot Algorithms

Ref. No year of publication	Technique used	Advantages & disadvantages
[11],2001	A blocking coordinated snapshot scheme , issue related to mobility management addressed	Solution to how mobility can be handled in blocking coordinated scheme.
[12], 2001	Concept of mutable snapshot is used, neither temporary or Permanent but can be converted into permanent when request for taking snapshot comes	Elimination of avalanche effect.
[13] , 2007	When a process has good probability of receiving snapshot request , then induced snapshot is taken before processing message, if probability is not good then process buffers messages till it takes snapshot or receive commit message	1. Tentative minimum set of processes calculated and made available in beginning to all the MSSs so as to reduce blocking time. 2.Reduced number of useless snapshots 3.Selective messages are delayed at receiver end so to allow process to send message during its blocking period.
[14],2006	Proposed a single phase non blocking coordinated snapshot approach for mobile computing environment	Disadvantages: It does consider the case of failure during the snapshot operation which may result in inconsistent state of the processes.
[26],1996	The protocol uses local timer which is able to store recoverable consistent states of the application without having to exchange messages	Advantages: 1.The first global state is used to recover the permanent and then to recover transient as well as soft failures. 2. Provide QOS to the current network.
[16], 1998	This algorithm is based under the chandy & lamport's assumption that one consistent global snapshot is obtained for a set of concurrent snapshot initiations	Advantages: 1. Total number of snapshots are minimized 2. Reuse of snapshot in a consistent global snapshot 3. Optimal algorithm

4. REFERENCES

[1] Kanmani P. " Fault Tolerance Using Token Ring Checkpointing In Dmcs " , 2014

[2] S. Kalaiselvi & V Rajaraman " A Survey Of Checkpointing Algorithms For Parallel & Distributed Computers " sadhana October 2000, Volume 25, Issue 5, pp 489-510

[3] Sneha & Ramtek " An Optimal Checkpointing Interval , A Novel Checkpointing Approach For Mobile Consumer Devices , IJARCSSE ,Volume 4, Issue 3, March 2014 ISSN: 2277 128X, 2014.

[4] Sarmistha Neogy " Wtmr – A New Fault Tolerance Technique For Wireless & Mobile Computing Systems "Future Trends of Distributed Computing Systems, 2007. ftdcs '07. 11th IEEE international workshop pp 130 – 137,ISSN 1071-0483

[5] Kyne-Sup BYUN, Sung_Hwa LIM, Jai-Hoon KIM," Two-Tier Checkpointing Algorithm Using MSS in Wireless Networks", IEICE Trans. Communications, Vol E86-B, No. 7, pp. 2136-2142, July 2003.

[6] L.Kumar , M.Misra,R.C Joshi " Low Overhead Optimal Checkpi For Mobile Distributed System " IEEE International Conference On Data Engineering pp 686-888 , 2003 .

[7] Anil Kumar , Mukesh Kumar , Parveen Kumar " Minimum Process Synchronous Checkpointing In Mobile Distributed Systems " IJCA, International Journal of Computer Applications (0975 – 8887), Volume 17– No.4, March 2011

[8] Parveen Kumar & Preeti Gupta ":A Mini Process Global State Detection Scheme For Mobile Distributed System. International Journal of Computer Applications (0975 – 8887) ,Volume 3 – No.10, July 2010

[9] Hélary j. M., mostefaoui a. And raynal m., "Communication-Induced Determination of Consistent Snapshots," Proceedings of the 28th International Symposium on Fault-Tolerant Computing, pp. 208-217, June 1998.

[10] Wang Y. and Fuchs, W.K., "Lazy Checkpoint Coordination for Bounding Rollback Propagation," Proc. 12th Symp. Reliable Distributed Systems, pp. 78-85, Oct. 1993.

[11] Suparna Biswas and Sarmistha Neogy : A Low Overhead Checkpointing Scheme For Mobile Computing Systems", int. Conf. Advances computing and communications , IEEE 2007 , pp: 700-705

[12] Cao G. and Singhal M., "Mutable Checkpoints: A New Checkpointing Approach for Mobile Computing systems," IEEE Transaction On Parallel and Distributed Systems, vol. 12, no. 2, pp. 157-172, February 2001.

[13] Cao G. and Singhal M., "Checkpointing With Mutable Checkpoints", Theoretical Computer Science, 290(2003), pp. 1127-1148.

[14] Jayanta Datta "A Fast and Efficient Non-Blocking Coordinated Movement-Based Check pointing Approach for Distributed Systems" International Journal Of Computational Engineering Research / ISSN: 2250–3005, IJCER | Jan-Feb 2012 | Vol. 2 | Issue No.1 | 136-142

[15] Y. Manable, "A Distributed Consistent Global Checkpoint Algorithm with Minimum Number of Checkpoints", Proceedings of 13th International Conference on Information Networking (ICOIN'98), pp. 549-555, January 1998

[16] B. Gupta, S. Rahimi, Z. Lui, "A New High Performance Checkpointing Approach for Mobile Computing Systems",(IJCSNS), Vol. 6, No. 5, pp. 95-104, May 2006.

[17] H. Lai and T.H. Yang," On Distributed Snapshots", Information Processing Letters, vol. 25, pp. 153-158, 1987.

[18] Acharya A. and Badrinath B. R., "Checkpointing Distributed Applications on Mobile Computers," Proceedings of the 3rd International Conference on Parallel and Distributed Information Systems, pp. 73-80, September 1994.

[19] Acharya A., "Structuring Distributed Algorithms And Services For Networks With Mobile Hosts", Ph.D. Thesis, Rutgers University, 1995.

[20] Adnan Agbaria, William H. Sanders, " Distributed Snapshots for Mobile Computing Systems", Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications (Percom'04), pp. 1-10, 2004.

[21] Badrinath B. R, Acharya A., T. Imielinski "Structuring Distributed Algorithms for Mobile Hosts", Proc. 14th Int. Conf. Distributed Computing Systems, June 1994.

[22] Higaki H. and Takizawa M. " Checkpointing Recovery Protocol For Mobile Checkpointing " IEEE 9th International conference on database expert system applications , 25-28 Aug 1998, pp 520 – 525,

[23] Rao, S., & Naidu, M.M.,"A New, Efficient Coordinated Checkpointing Protocol Combined with Selective Sender-Based Message Logging", International Conference on Computer Systems and Applications.IEEE/ACS, 2008.

[24] Tony T-Y. Juang and S. Venkaesan "Jaug-Venkatesan "CRASH RECOVERY WITH LITTLE OVERHEAD" IEEE, Distributed Computing Systems, 1991., 11th international conference,pp 454 – 461, **isbn** 0-8186-2144-3

[25] Basu "A Mobility Based Metric for Clustering in Mobile Ad Hoc 1Proc. IEEE ICDCS 2001 Workshop on Wireless Networks and Mobile Computing, Phoenix, AZ, April 2001.

[26] Neves & Fuchs " Adaptive Recovery For Mobile Environments " in proceedings of the IEEE High-Assurance Systems Engineering Workshop, October 1996 , pp 134 – 141.

[27] Sung-Hwa Lim "Power-Aware Optimal Checkpoint Intervals For Mobile Consumer Devices" **in:**Consumer Electronics, IEEE Transactions on (Volume:57 , Issue: