Volume 6, Issue 5, May 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Parameterised Comparison of Checkpointing Algorithms for Mobile Distributed Systems

Vijaya Kapoor
Shri Venketeshwar University
Gajrula, Uttar Pradesh, India

Dr. Parveen Kumar
NIMS University
Jaipur, Rajasthan, India

Abstract— Most of the man made systems are prone to faults. It is difficult to predict whether the system will continue to perform throughput or till prescribed time. Fault tolerance is the property that enables a system to continue properly, in case of sudden, abrupt disconnection in the event of failure of some of its components. Localization, data aggregation, reliability, fault tolerance, scalability & security, energy consumption are the various challenges of the Mobile distributed system. Most of the checkpointing strategies used the mobile service stations to help deal with checkpoints. The operating cost of wireless message transmission is still high. It is again difficult due to disconnections. Hence, to design new checkpoints methods supporting the unique feature totally for mobile networks is the necessity of the hour. This paper gives a summary of checkpointing strategies for mobile networks which are categories on the basis of QOS of wireless networks, based on mobile agents, considering the mobility of MHS and transmission of checkpoints.

Keywords— QOS Quality of service, MH,MSS,MDCS.

I. INTRODUCTION

In centralized algorithms, there is one node which always initiates checkpointing procedure & coordinate the participating nodes, while in the distributed system checkpointing algorithm, nodes are given autonomy in initiating checkpointing by allowing any node in the system to initiate complete checkpointing or selective checkpointing. Distributed system is comprised of a shared heterogeneous computing & data resources are distributed across the network boundaries. Due to geographically distributed resources, heterogeneous in nature, owned by different individuals or organizational own policies with varying loads & availability causes the Resource Management system, QOS should be obtained even in the existence of resource faults a complex task. Thus, detection of errors & their effective rectification should be the criteria for fault tolerant service. In distributed the algorithm for distributed systems are more difficult to design & debug as compared to centralized systems as there is an absence of common global clock & no shared memory. All the traditional two phase checkpointing is an expensive fault tolerance method for various reasons, as it may involve unnecessary node rollbacks and it requires a large number of control messages. As a result it affects the bandwidth and power negatively as compared to the traditional distributed environment; mobile networks are typically slow with low bandwidth & throughput. As mobile nodes are more effective to failure than static nodes and failure in the network may lead to wastage of resources. When a recovery operation is performed, a failed node only connects itself hence mobile failure node affect the recovery in mobile computing environment. There are certain memory limitations; therefore, some garbage collection mechanism is necessary. As weak wireless links exist between Mobile nodes & mobile host proxy, only essential write events need to be transferred over wireless links.

Other requirements include conserve energy need for planned connections, optimization of recovery cost to reduce the loss of data, reduction in cost of traversal of data, in existence of orphan messages, reduction in coordination overhead by sending only the snapshot request messages to only dependent MHs in the current snapshot interval, time taken by snapshot need to be minimum during a failure free run, rollback recovery need to be fast in existence of selective rollbacks, number of useless snapshots need to be less. Failure of system is neither threatening nor safety critical. To make a network sustainable & error free, fault tolerance is an important challenge which should be solved. Unfortunately, research does not consider fault tolerance as important. Only limited work has been done in past for fault tolerance & recovery by the Researchers. There are few critical events like patient monitoring system which may lead to many losses, including life, cost & data involved. Backward error recovery is usually better approach than forward recovery approach in which recovery means moving the faulty state to an already known error free state. For a system to make it fault tolerant, few popular approaches are dealt which involves installation of additional nodes in the environment for checking & repairing nodes that had failed to do the assigned work. Some of the checkpointing examples include incremental checkpointing & memory exclusion checkpointing, copy-on writing checkpointing & forced checkpointing. During checkpointing & rollback Technique system will be in a blocked state or frozen state.

Definitions

A. Dependency Vector (dv): A Matrix of N fields at every process. dvi[i] is initialized to one and incremented on a checkpoint. It gives current interval index (CII) or the checkpoint index (CI) of the next checkpoint.In the algorithm proposed by Briatico, Ciuffoletti, and Simoncini (BCS) [5], every process maintains and propagates an index sn that is similar to a logical clock [20]. The proposed algorithm requires only the sequence number (SN) to be piggybacked along with the message instead of the sequence number and EQi (equivalence vector). Thus the Enhanced index-based checkpointing algorithm decreases the number of checkpoints with less complexity as compared to the existing index-based checkpointing algorithms.

System Model

The algorithms use the common system model in which a mobile computing system consists of a set of mobile hosts (MHs) and mobile support stations (MSSs). The static MSS provides various services to support the MHs and a region covered by an MSS is called a cell. A wireless communication link is established between a MH and an MSS; and a high speed wired communication link is assumed between any two MSSs. The wireless links support FIFO communication in both directions between a MSS and the MHs in the cell. A distributed computation is performed by a set of MHs or MSSs in the network.

Fault Tolerant strategies Checkpointing Strategies based on QOS of the wireless networks

Reconfiguration means restore age of the resources to some operational state by eliminating faulty component from a resource, reconfiguration process is: 1. Fist step is to detect faults before any recovery procedure is initiated. 2. Fault location means where a fault has occurred. 3. Fault containment means to prevent the foot from propagating throughout the entire system 4. Fault recovery: is the process of regaining the resource from failure.

Checkpointing Strategies based on QOS of the wireless networks

When distributed systems users submit their jobs to the scheduler in which end user wants their jobs to be executed by specifying end user jobs on the best available resources by time optimization as a result of the task is submitted to user upon successful completion of the task. If a fault occurs at a distributed system resource, the job is scheduled on another resource, which consumes more time, hence the failing to user QOS requirement as the task is rescheduled. Such a distributed system environment has major drawbacks. If a fault occurs, the solution for this can be achieved, by maintaining the log of information of the distributed system resource hence checkpointing is used effectively. Some authors used the concept of soft/hard checkpoints for maintaining QOS. Transmit soft checkpoints to its MSS by wireless channels. The value of MAXSOFT can be fine tuned with the QOS adaptively.

Checkpointing strategies based on Mobile agents:

In Case of handoff procedure and Mobile agent will take charge of transmitting the checkpoints & message logs from MSS to MH. In the method, mobile agents can execute at one host & then dispatched to another. At the time of recovery, Related Checkpoints & message logs are recovered by the Mobile agents.

Checkpointing strategies based on Mobility of MHs:

Kumar et. al. [6] used the concept of Movement patterns intracell, intracell or combination . The drawback of this method is that due to the long time of recovery , in case of multiple handoffs. As most of the MHs are moving in a particular movement pattern. The movement patterns are intercell, intracell and combination of both.On the basis of movement pattern decision whether to:a) Take a snapshot periodically, exploiting snapshot interval if movement pattern is intracell.b) Take snapshot when handoff value increased by threshold value, if movement pattern is intercell or In combination .If there is no movement of MHs, the handoff will not occur, then initiate snapshot at an interval of K unit time ,here the snapshot initiation value is snapshot interval.If there is mobility of MHs , handoff will occur , increment h_count variable till its value exceeds the handoff threshold value. Here snapshot initiation value is handoff threshold value.

Types of checkpointing from upper to lower hierarchy

1. User level checkpointing.
2. Application checkpointing.
3. Uncoordinated/ Coordinated/Incremental checkpointing/etc.
4. Low level checkpointing.

Assumptions for a good checkpointing algorithm :

1. Number of disk contentions should be reduced.
2. Output commit latency should be minimized.
3. Garbage collection should be simplified and.
4. Strongly consistent global state should be guaranteed.

II. LITERATURE REVIEW

Some authors have introduced a weighted checkpointing approach(WCA) for the mobile distributed computing system (MDCS) that radically reduces checkpointing overheads on mobile hosts. There are various parameters on which a Checkpoint protocols will be needed during recovery depends such as the fault rate ,message arrival rate, number of messages sent by a process before it takes a checkpoint and the communication pattern. The protocol used these parameters to calculate the weight of a checkpoint that decides whether to take or skip the checkpoint

Some authors have introduced a weighted checkpointing approach (WCA) for the mobile distributed The other used a Distance and frequency based scheme [58] which allow movement of checkpoint and message logs to a nearby MSS when either distance between MH and MSS on which latest checkpoint is saved exceeds a threshold value, or when Number of handcuffs that are a number of MSS carrying message logs of a MH exceeds a threshold respectively. These Schemes keep the recovery information of MH in a certain range. Sarmistha Neogy proposed A WirelessTMR-(WTMR) Checkpointing technique is proposed that uses checkpointing technique to add fault tolerance to a fault tolerant TMR node in wireless system.

In Suparna Biswas[22] protocol there exists Pinitiator, which coordinates with all the processes to take a consistent global checkpoint. Pinitiator is responsible for invoking the checkpoint operation periodically.

The concept of active interval of initiator is introduced in which sending of control messages, preparation of checkpoints & taking checkpoint messages to all other processes. The active interval of any process includes, the time lapses between two events by receiving " prepare checkpoint " & take a checkpoint message by any process. T is the max transmission delay incurred by any message to reach the destination. To survive the transmission delay of the control message the active interval value should be 3ti. e t>3t. The protocol also enables the logging of the message within an active interval, value should be 3ti.e t>3t . The protocol also enables the logging of the message within an active interval, then process execution is continued. By this protocol will handle the lost messages [2]. Initialisation of two counters MRC, MSC by zero at the starting of active interval. Every process maintains two counters namely Message Received Count (MRC) and Message Sent Count (MSC). These counters are initialized to zero at the start of the 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, Processing at the initiator :

1. The initiator sends 'take checkpoint' signal to other processes. 2. It takes the checkpoint 3. Exits from the active interval.
2. Processing of rest of the process. : 1. Take checkpoint 2. Exits from the respective active intervals.

Two situations arise First is if failure occurs, after all processes exited from their respective active intervals, the application rolls back to the latest consistent global state, namely G. Second, if failure occurs before one of the processes exits from their respective active intervals, the application will roll back to the previous global state namely G-1.

In Cao& Singhal[8] **,** the processes need not block during checkpointing by using a checkpoint sequence number to identify orphan messages , however these algorithms assumes that a distinguish initiator decides when to take a checkpoint . in order to keep the csn updated , any time a process takes a checkpoint , it has to notify to all the processes in the system.

In Koo& Toueg[37] , in first phase an initiating process pi takes a tentative checkpoint & request all other processes to take tentative checkpoints . each process informs pi whether it succeeded in taking a tentative checkpoint , A process says "no" to a request if it fails to take a tentative checkpoint , if pi learns that all the processes have successfully taken tentative checkpoints , pi decides that all tentative checkpoints should be made permanent otherwise pi decides that all the tentative checkpoints should be discarded. In the second phase, either all or none of the processes advance the checkpoint by taking permanent checkpoints. The algorithm requires that after a process has taken a tentative checkpoint, it cannot send messages related to the underlying computation.

According to Elnozahy[50], the Stable storage access is now the major source of overhead in checkpointing or message logging systems. Communication overhead is much lower in comparison. Such changes favor coordinated checkpointing schemes over message logging or uncoordinated checkpointing systems, as they require less access to stable storage and are simpler to implement. The case for message logging has become the ability to interact with the outside world, instead of reducing the overhead of multi-process coordination [Elnozahy and Zwaenepoel 1994]. Message logging systems can implement efficient protocols for committing output and logging input that are not possible in checkpoint only systems.

In Rachit Garg[43], initiator process collects the dependency vectors of all processes & compute the tentative minimum set. To balance the checkpointing overhead & loss of computation on recovery, we design a hybrid checkpointing algorithm for mobile distributed systems, where an all process checkpointing is taken after executing minimum process algorithm for 15 numbers of times. In the proposed algorithm, in first phase, the relevant MHS are required to take soft checkpoint only. A soft checkpoint are stored on the disk of the MH and is similar to mutable checkpoints, if any process fails to take its checkpoint in coordination with others, then all relevant processes need to abort their soft checkpoints only.

In Parveen Kumar[35] , MSSin sends a request to all MSSs to send the direct dependency vectors of the processes in their cell. All ddv vectors are at MSSs and thus no initial checkpoint message or response travels on wireless channels. The initiator MSS sends checkpoint request to all processes. On receiving the checkpoint request a process takes the tentative checkpoint if it has not taken the checkpoint during current initiation. After checkpoints commencement a process updates its CIs. A process informs about the checkpoint or its inability to take checkpoint informs to its local

MSS. When MSS learns that all processes have taken the tentative checkpoint or failed, it send the response to initiator MSS. Finally MSS sends commit or abort to all MSSs.

A. *Analysis based on Scheme, Phase, Type of checkpoints, Channel allocation, Concurrent execution, useless checkpoint requests, piggybacks.*

Analysis Parimeter	**Cao& singhal[8]**	**Koo& toueg[37]**	**Elnozahy[50] et al.**	**Rachit garg[43] et al.**	**Parveen kumar[35] algorithm**
scheme	Coordinated,minprocess, initially blocking then updated to non blocking	coordinated checkpointing Minimum-Process but Blocking Algorithms	Message logging with coordinated checkpointing, All Process but Non-Blocking Algo.	Coordinated, hybrid non blocking	Coordinated, hybrid, non blocking
Phase	two	two	none	Three	three
Type of checkpoints	Mutable checkpoints	permanent and tentative	none	Mutable , permanent and tentative	Adhoc, tentative and permanent New concept of proxy
Channel allocation	--	FIFO	All process blocking algorithm can be differentiates on the basic of FIFO and non-FIFO channels.	Sequence of messages without using explicit sequence number hence FIFO	FIFO as messages are exchanged through reliable channels, whose transmission delays are finite but arbitrary. Communication between the MHs and the PMSS and between PMSSs and MSS are assumed to be lossless and FIFO
Concurrent execution	allowed	single process initiation But proposed methods for handling concurrent initiations of snapshot collection	--	Concurrent invocations of the algorithm do not occur	No. as our goal is to min checkpointing effort defeated
Useless checkpoint requests	possible	--	--	No	--
Piggybacks	csni [j] and a data structure MR.	integer	integer	no need to piggyback these data structures onto checkpoint requests.(MR)only In the proposed protocol, tminset and tnp_minset are piggybacked onto checkpoint requests.	Integer k bit CI instead of CSN the value of k can be fined tuned. If min process algo is executed for 7 Number of times, then 3 bit CI required.

B. Analysis based on broadcasted on the static network, lead to inconsistencies, Message complexity, Number of useless checkpoints, Avg. number of checkpoints, Avg. Blocking Time, expensive.

Analysis Parameters	**Cao& Singhal[8]**	**Koo& Toueg[37]**	**Elnozahy[50] et al.**	**Rachit Garg [43]et al.**	**Parveen Kumar [35] algorithm**
broadcasted on the static network	Only commit request is broadcasted.	--	--	Exact minimum set along with commit request	--
lead to inconsistencies	May lead to inconsistencies, due to concurrent executions	--	several advantages, including faster recovery, improved failure free performance, bounded recovery time, simplified garbage collection and reduced complexity.	--	--
Message complexity	2Nmin *Cpp +Cbst+ Nucr* Cpp	3*Nmin *Cpp * Ndep	2Cbst + N *Cpp	Reduced compared to cao	$2*C_{wl}+3*C_{bst}+4*N_{min}*C_{wl}+2*N_{min}*C_{st}$ Slightly increased
No# of useless checkpoints	Presents –hence avg. no# of useless chkpt are higher as transitive dependencies are captured by direct dependencies	---	--	Same as cao .huge MR[] data structure is also attached with snapshot request	Present but reduced at the cost of negligible blocking
Avg. no# of checkpoints	$N_{min} + N_{mut}$	N_{min}	N	Nmin +Nind	$(m. N_{min}+N)/(m+1)$
Avg. Blocking Time	0	$N_{min} *T_{ch}= N_{min}*(T_{msg} +t_{data}+T_{disk})$ The blocking time of the Koo-Toueg [11] protocol is highest,	0	0	0 as negligible blocking
Expensive	Costly due to non blocking, as non Blocking are more costly than blocking	Expensive as two phase are more costly due to 1. unnecessary rollback. 2. Require large no. of messages.	Expensive than others however, revealed that coordinated algorithms are better than independent algorithms. The cost of coordination is much lower when compared with the cost of maintaining multiple checkpoints and logging messages.	Slightly efficient	efficient and suitable for tolerating mobile distributed systems against faults

C. Analysis Based on Domino effect, Orphan Messages, Dependencies handling, Deterministic/non deterministic, Checkpoint tree formation, Storage, Distributed/ Centralised, handoff, disconnection, and failure cost/ handling disconnections, Bandwidth/low overhead of wireless channels, Low memory overhead on /m/hs.

Analysis Parameters	**Cao& Singhal[8]**	**Koo& Toueg[37]**	**Elnozahy[50] et al.**	**Rachit Garg[43] et al.**	**Parveen Kumar[35] algorithm**
Domino effect	Avoid domino effect	Prevention of domino effect	Possibility	No	No

Orphan Messages	No	No	No as message logging guaranteed that upon recovery , no process is an orphan	No	Orphan message free
Dependencies handling	No guaranteed to get Z-dependencies information in time since the computation is in progress	--	--	Z-dependencies TAKEN CARE	Z-dependencies TAKEN CARE
Deterministic/non deterministic	Non Deterministic	--	Determinant information	Non Deterministic	Non Deterministic as we consider that the processes which are running in the distributed mobile are non deterministic
Checkpoint tree formation	Suffers from the formation of checkpoint tree	Large checkpointing tree h1: height of the checkpointing tree in Koo-Toueg algorithm [9].	--	--	Height of chkpt tree is lowas compared to cao& singhal as most of the transitive dependencies are captured during normal processing
Storage requirement	Small	Minimal stable storage as each process stores at most two checkpoints in stable storage.	Minimum As minimizes the stable storage requirement hence it seems better than asynchronous checkpointing	--	To avoid any waste of bandwidth or CPU consumption, the algorithm is loop Free. Reducing storage overhead.
Distributed/ Centralised	Distributed	Distributed	Centralized	Distributed	Distributed The algorithm is distributed in nature. There is no centralized controlling node
Handoff, disconnection, and failure cost/ handling disconnections	--	--	--	Yes as disconnection does not lead to wait state	Yes, In our approach all the log information is stored in PMSS. When a MH moves within the same cell, no log information is transferred, as PMSS handles all the data of the cell. in such way handoff, disconnection, and failure can be reduced & Fewer checkpoints are transferred through wireless link
Bandwidth/low overhead of wireless channels	Affects negatively as large amount of data is transferred to MSS in two phase algorithm	--	--	--	Better utilization

III. CONCLUSIONS

As a new kind of the Distributed network, Mobile networks have been widely applied to different kinds of application situations, but the MHs in the mobile environment are much liable to fail. From the analysis it has been concluded that 1. Limited storage space will necessitate that storage spaces occupied by the checkpoints should be cleaned up regularly. 2. The message generated by the checkpoints should be reduced as wireless bandwidths of mobile networks are limited. 3. Location consistency of MHs should be considered along with checkpoint method. Network failure, resource overloading or no availability of required software components are the several reasons for failure in execution. Thus, in fault tolerant system, there must be a provision to identify & rectify the failures & support reliable execution in the presence of failure. To conclude a survey on the basis of the various fault tolerance techniques & mechanisms, a comparative chart on various parameters is made.

REFERENCES

[1] Wikipedia Ipv6

[2] Ruchi Tuli & Parveen Kumar "The Design and Performance of a Snapshot Scheme for Mobile Ad Hoc Networks "Chapter Springer "Advances in Parallel Distributed Computing" Volume 203 of the series Communications in Computer and Information Science pp 204-212, 2011

[3] Suparna Biswas and sarmisthat neogy " . Snapshot and Recovery Using Node Mobility among Clusters in Mobile Ad Hoc Network" springer "Volume 176 of the series Advances in Intelligent Systems and Computing pp 447-456, 2010

[4] Yogita khatri "Distance-based Asynchronous recovery approach in mobile computing environment "International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.3, May 2012

[5] P.K. Suri and menu satiza "An efficient snapshot protocol for mobile distributed systems ,ijlrst vol 1 , issue 2 : page no 109-114 (2012)

[6] Vijaya kapoor & Parveen kumar ,"An Efficient checkpointing protocol for Mobile distributed environment " IJRE journal

[7] Parveen Kumar, Rachit Garg (2010) " Soft-Snapshot Based Coordinated Snapshot Protocol for Mobile Distributed Systems", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 5, May 2010

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

[9] Lalit Kumar Awasthi, Kumar Parveen. 2007 "A Synchoronous checkpointing protocol for mobile distributed systems :probabilistic approach". Int J. Information and Computer Security, Vol.1, No.3 .pp 298-314

[10] Pradeep kumar & parveen kumar " MHP based synchronous snapshot approach for mobile distributed systems.", International Journal of Innovations in Engineering and Technology (IJIET), Vol. 3 Issue 4 April 2014, ISSN: 2319 – 1058

[11] Deepika, Gaurav Garg, Parveen Kumar "Three Phase Non-blocking Snapshoting Algorithm for Mobile Distributed System" , Volume 5, Issue 5, May 2015 ISSN: 2277 128X "International Journal of Advanced Research in Computer Science and Software Engineering"

[12] Suparna biswas & sarmistha Neogy " A mobility-based checkpointing protocol for mobile computing system", IJCSIT vol 2, no. 1 , Feb 2010.

[13] Mukesh Singhal, Niranjan G.Shivaratri, Niranjan Shivaratri "Advanced concepts in operating systems" McGraw-Hill, Inc. New York, NY, USA ©1994

[14] Kanmani P "Fault tolerance using token ring checkpointing in distributed mobile computing systems" shodhganga 2014

[15] Wood., W.G., "A Decentralized recovery control protocol", IEEE Symposium on Fault Tolerant Computing, 1981.

[16] Storm R., and Temini, S., "Optimistic recovery in distributed systems",ACM Trans. Computer Systems, Aug, 1985, pp. 204-226.

[17] Rachit Garg & Parveen kumar" Low overhead checkpointing protocols for mobile distributed systems: A Comparative STUDY" International Journal of Engineering Science and Technology , Vol. 2(7), 2010, 3267-3276

[18] Bidyut Gupta, Shahram rahimi and Ziping liu., "A New high performance checkpointing approach for mobile computing system", CSNS International Journal of Computer Science and Network Security, VOL 6, N05B May 2006

[19] Cao G. and Singhal M., "On the Impossibility of min-process nonblocking checkpointing and an efficient checkpointing algorithm for mobile computing systems", Proceedings of International Conferenceon Parallel Processing, August 1998, pp. 37-44.

[20] R.C. Gass, B. Gupta, "An Efficient checkpointing scheme for mobile computing systems", European Simulation Symposium, Oct 2001 (18-20), pp.1-6

[21] Lalit Kumar, Parveen Kumar and R K Chauhan., "Pitfalls in minimumprocess coordinated checkpointing protocols for mobile

[22] Suparna biswas & sarmistha Neogy " A mobility-based checkpointing protocol for mobile computing system", IJCSIT vol 2, no. 1 , Feb 2010.

[23] Mukesh Singhal, Niranjan G.Shivaratri, Niranjan Shivaratri "Advanced concepts in operating systems" McGraw-Hill, Inc. New York, NY, USA ©1994

[24] Kanmani P "Fault tolerance using token ring checkpointing in distributed mobile computing systems" shodhganga 2014

[25] Wood., W.G., "A Decentralized recovery control protocol", IEEE Symposium on Fault Tolerant Computing, 1981.

[26] Storm R., and Temini, S., "Optimistic recovery in distributed systems",ACM Trans. Computer Systems, Aug, 1985, pp. 204-226.

[27] Rachit Garg & Parveen kumar" Low overhead checkpointing protocols for mobile distributed systems: A Comparative STUDY" International Journal of Engineering Science and Technology , Vol. 2(7), 2010, 3267-3276

[28] Bidyut Gupta, Shahram rahimi and Ziping liu., "A New high performance checkpointing approach for mobile computing system", CSNS International Journal of Computer Science and Network Security, VOL 6, N05B May 2006

[29] Cao G. and Singhal M., "On the Impossibility of min-process nonblocking checkpointing and an efficient checkpointing algorithm for mobile computing systems", Proceedings of International Conferenceon Parallel Processing, August 1998, pp. 37-44.

[30] R.C. Gass, B. Gupta, "An Efficient checkpointing scheme for mobile computing systems", European Simulation Symposium, Oct 2001 (18-20), pp.1-6

[31] Lalit Kumar, and R K Chauhan., "Pitfalls in minimum process coordinated checkpointing protocols for mobile distributed", ACCST Journal of Research, Volume III, No. 1, 2005 pp. 51-56.

[32] Lalit Kumar and Parveen Kumar., "A Synchronous checkpointing protocol for mobile distributed systems: a probabilistic approach", International Journal of Information and Computer Security, Vol. 3 No. 1, 2007, pp 298- 314 .

[33] Rachit Garg, Praveen Kumar " A Review of checkpointing based fault tolerance techniques in mobile distributed systems " , International Journal of Computer Applications, 2010 ,Number 2 - Article 2

[34] Prof. S. M. Tidke, Rucha Ravindra Galgali "Predicting resource allocation in distributed environment by using online predictive approach a review" International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 12, December 2013

[35] Parveen kumar "A Minimum process global state detection scheme for mobile distributed systems " IJEST, vol. 2(7),2010, 2853-2858

[36] J.L.kim & T. park "An Efficient Protocol for Checkpointing Recovery in Distributed Systems", IEEE Transactions on Parallel and Distributed Systems, Vol -4, Aug.1993, Page 955-960

[37] Richard Koo , Sam Toueg , "Checkpointing and rollback-recovery for disitributed systems (1987) Published in, Software Engineering, IEEE Transactions on (Volume:SE-13 , Issue: 1)

[38] Acharya A., "Structuring distributed algorithms and services for networks with mobile hosts", Ph.D. Thesis, Rutgers University, 1995.

[39] Parveen Kumar, Poonam Gahlan " A Low overhead minimum process coordinated checkpointing algorithm for mobile distributed systems" IJCA , volume3, number1

[40] R. Prakash and M. Singhal. "Low-cost checkpointing and failure recovery in mobile computing systems". IEEE Trans. on Parallel and Distributed System, pages 1035-1048,Oct. 1996.

[41] Cao G. and Singhal M., "On Coordinated checkpointing in Distributed Systems", IEEE Transactions on Parallel and Distributed Systems, vol. 9, no.12, pp. 1213-1225, Dec 1998.

[42] L. Kumar, M. Misra, R.C. Joshi, "Checkpointing in distributed computing systems" Book Chapter "Concurrency in Dependable Computing", pp. 273-92, 2002.

[43] Parveen Kumar, Rachit Garg ,"Soft checkpointing based hybrid synchronous checkpointing protocol for mobile distributed systems", International Journal of Distributed Systems and Technologies", 2(1), 1-13, January-March 2011.

[44] Ruchi Tuli, Parveen Kumar, "The Design and performance of a checkpointing scheme for mobile ad hoc networks", Springer-Verlag CCIS 203, pp 204-212, 2011.

[45] Pradhan, D.K., Krishna, P., and Vaidya, N.H."Recovery in mobile environments: design and trade-off analysis" In Proceedings of the 26th International Symposium on Fault-Tolerant Computing, (Sendai, Japan, June 1996), IEEE, pp. 16–25.

[46] Nuno Neves and W. Kent Fuchs. "Adaptive recovery for mobile environments", in Proc.IEEE High-Assurance Systems Engineering Workshop,October 21-22,1996,pp.134

[47] Ruchi Tuli , Parveen Kumar " Analysis of recent checkpointing techniques for mobile computing systems". IJCSES, vol. 2, no. 3 , August 2011.

[48] Vijaya Kapoor and Parveen Kumar. "A Comparative study on snapshot protocols for mobile distributed systems" International Journal of Computer Applications 2014 Volume 106 - Number 3 Year of Publication: 2014

[49] Lalit Kumar Awasthi, Kumar Parveen. 2007 "A Synchoronous checkpointing protocol for mobile distributed systems :probabilistic approach". Int J. Information and Computer Security, Vol.1, No.3 .pp 298-314

[50] Elnozahy E.N., Alvisi L., Wang Y.M. and Johnson D.B., "A Survey of rollback-recovery protocols in message-passing systems," ACM Computing Surveys, vol. 34, no. 3, pp. 375-408, 2002.

[51] Taesoon Park, Namyoon Woo and Heon Y. Yeom, "An Efficient Recovery Scheme for Fault Tolerant Mobile Computing Systems", FGCS- 19, 2003 Kumar, P.," A Low-cost hybrid coordinated checkpointing protocol for mobile distributed systems", Mobile Information Systems pp 13-32, Vol. 4, No. 1. ,2007.

[52] Ch.D.V. Subba Rao, A New, Efficient Coordinated Checkpointing Protocol Combined with Selective Sender-Based Message Logging

[53] Chandy K. M. and Lamport L., "Distributed snapshots: determining global state of distributed systems," ACM Transaction on Computing Systems, vol. 3, No. 1, pp. 63-75, February 1985.

[54] Pushpendra Singh, Gilbert Cabillic, "A Checkpointing algorithm for mobile computing environment", LNCS, No. 2775, pp 65-74, 2003.

[55] S Kalaiselvi And V Rajaramana ,"A Survey of checkpointing algorithms for parallel and distributed computers" SaÅdhanaÅ, Vol. 25, Part 5, October 2000, pp. [35] Guohong Cao, Mukesh Singhal "On Coordinated checkpointing in distributed systems" IEEE transactions on parallel and Distributed Systems , Volume 9 Issue 12, December 1998 Page 1213-1225

