



INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume3, Issue3)

Available online at www.ijariit.com

Review on checkpoint optimization in WSN

Gurbhej Singh

Universal Institute of Engg. & Tech. IKG PTU Jalandhar
gurbhejtusshar07@gmail.com

Er. Aayushi Chadha

Universal Institute of Engg. & Tech. IKG PTU Jalandhar
aayushichadha.237@gmail.com

Abstract: A shared heterogeneous computing is comprised of distributed system & data resources are distributed across the network boundaries. Owing to geographically distributed resources, which is heterogeneous in nature, owned by different individuals or organizational that owns policies with varying loads & the Resource Management system caused by the availability, even obtaining the QOS in the resource faults existence a complete which overhead of checkpoint by clustering based checkpoint and optimize the number of nodes in cluster by Flower pollination algorithm and use centrality base trust case depend on connectivity and importance of node using game theory approaches.

Keywords: Check Point, QOS, and Network.

I. INTRODUCTION

In centralized algorithms, checkpointing procedure always initiated by one node & the participating nodes is being coordinated, while checkpointing algorithm in the distributed system, nodes are given autonomy in initiating check-pointing by allowing any node in the system to initiate complete check-pointing or selective checkpointing.

A shared heterogeneous computing is comprised of distributed system & data resources are distributed across the network boundaries. Owing to geographically distributed resources, which is heterogeneous in nature, owned by different individuals or organizational that owns policies with varying loads & the Resource Management system caused by the availability, even obtaining the QOS in the resource faults existence a complex task. Thus, errors detection & the criteria for fault tolerant service should be their effective rectification. The algorithm for distributed systems in distributed is more difficult to design & debug in comparison to the centralized systems because of an absence of common global clock & no shared memory. All the traditional two-phase check-pointing for various reasons is an expensive fault tolerance method, 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 a 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 is 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 the 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 to approach than forwarding 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 check-pointing examples include incremental checkpointing & memory exclusion check-pointing, copy-on writing check-pointing & forced check-pointing. During checkpointing & rollback Technique system will be in a blocked state or frozen state.

Types of check-pointing from upper to lower hierarchy

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

Assumptions for a good checkpointing algorithm:

1. A number of disk contentions should be reduced.
2. Output commit latency should be minimized.
3. Garbage collection should be simplified and.
4. The strongly consistent global state should be guaranteed.
- 5.

II. LITERATURE REVIEW

Pulak Khamrui et al [1] In this paper they have designed a new a trusted node based checkpointing algorithm which is very effective to avoid temporary, tentative, mutual or soft checkpoints. This scheme reduces the number of processes involved in the checkpointing scheme. Due to the absence of temporary, tentative, mutable or soft checkpoints, the number of the control message is very low. Moreover, due to less number of processes participating in this checkpointing scheme, the cost is also less compared to other checkpointing algorithms.

Rachit Garg et al [2] In this paper, they propose a non-blocking coordinated checkpointing algorithm for mobile computing systems, which requires only a minimum number of processes to take permanent checkpoints. They reduce the message complexity as compared to the Cao-Singhal algorithm while keeping the number of useless checkpoints unchanged. And also address the related issues like failures during checkpointing, disconnections, concurrent initiations of the algorithm and maintaining exact dependencies among processes. Finally, the paper presents an optimization technique, which significantly reduces the number of useless checkpoints at the cost of a minor increase in the message complexity. In coordinated checkpointing, if a single process fails to take its tentative checkpoint; all the checkpoint effort is aborted. They try to reduce this effort by taking soft checkpoints in the first phase at Mobile Hosts.

Bidyut Gupta et al [3] In this paper, they presented a single phase non-blocking coordinated checkpointing algorithm suitable for mobile computing environments. The distinct advantages that make the proposed algorithm suitable for distributed mobile computing systems are the following. It produces a consistent set of checkpoints, without the overhead of taking temporary checkpoints; the algorithm makes sure that only minimum number of processes are required to take checkpoints in any execution of the checkpointing algorithm; it uses very few control messages and the participating processes are interrupted less number of times. Performance analysis shows that the proposed approach outperforms some existing important related works.

Guohong Cao et al [4] In this paper, the concept of the mutable checkpoint is introduced, which is neither a tentative checkpoint nor a permanent checkpoint, to design efficient checkpointing algorithms for mobile computing systems. Mutable Checkpoints can be saved anywhere, e.g., the main memory or local disk of MHs. In this way, taking a mutable checkpoint avoids the overhead of transferring large amounts of data to the stable storage at MSSs over the wireless network. And presented techniques to minimize the number of mutable checkpoints. Simulation results show that the overhead of taking mutable checkpoints is negligible. Based on mutable checkpoints, our non-blocking algorithm avoids the avalanche effect and forces only a minimum number of processes to take their checkpoints to the stable storage.

Praveen Kumar et al [5] In this paper, we propose a non-intrusive minimum process synchronous checkpointing protocol for mobile distributed systems, where only the minimum number of tentative checkpoints is taken. And also optimize and the number of useless forced (mutable) checkpoints and message overheads.

Ruchi Tuli et al [6] In this paper, studied a very fundamental problem, the fault tolerance problem, in a MANET environment and propose a minimum process coordinated checkpointing scheme. Since potential problems of this new environment are insufficient to power and limited storage capacity, the proposed scheme tries to reduce the amount of information saved for recovery. The MANET structure used in our algorithm is hierarchical based. The scheme is based on Cluster Based Routing Protocol (CBRP) which belongs to a class of Hierarchical Reactive routing protocols. The protocol proposed by us is not - blocking coordinated checkpointing algorithm suitable for ad hoc environments. It produces a consistent set of checkpoints; the algorithm makes sure that only minimum number of nodes in the cluster are required to take checkpoints; it uses very few control messages. Performance analysis shows that the algorithm outperforms the existing related works and is a novel idea in the field. Firstly, describing an organization of the cluster. Then proposed a minimum process coordinated checkpointing scheme for cluster based ad hoc routing protocols.

Suparna Biswas et al [7] Proposed work is a mobility based checkpointing and trust based rollback recovery algorithm to provide fault tolerance in Mobile Ad hoc Network (MANET). Here each mobile host maintains a count of number of clusters a mobile host traverses through, during a single checkpoint interval. A mobile host increments 'cluster-change-count' by 1, each time it leaves a cluster and joins another. Each mobile host saves a checkpoint independently if its 'cluster-change-count' exceeds a

predefined threshold. This measure is important because each mobile host leaves its last checkpoint and logs at different clusters that it has visited earlier. If the mobile host fails, the time to search and collect its last checkpoint and logs gets added to the recovery time of the mobile host. In MANET, retrieval of checkpoint and logs has to be done through a number of intermediate mobile hosts because each mobile host has short area coverage, hence direct communication among distant mobile hosts is not possible. Now if any of these mobile hosts fail, depending on the nature of the failure, the checkpoint or log may be lost or forwarding of them to the failed host may be delayed causing unsuccessful or delayed recovery of the failed host respectively. This can be avoided if it is ensured that the checkpoints and logs are forwarded only through trusted nodes. Trust model proposed here computes trust value of a mobile host based on four factors: failure rate, availability in the network, unused energy, and recommendations from neighbour mobile hosts. Simulation results show that proposed algorithm achieves low recovery cost and high recovery probability of failed mobile hosts.

A.Acharya et al [8] The integration of mobile/portable computing devices within existing data networks can be expected to spawn distributed applications that execute on mobile hosts (MHs). For reliability, it is vital that the global state of such applications be checkpointed from time to time. A global checkpoint consists of a set of local checkpoints, one per participant. This paper first identifies the problems in recording a consistent global state of mobile distributed applications. The location of an MH within the static network varies with time and therefore, an MH will first need to be located ("searched") in order to obtain its local checkpoint. Moreover, MHs often (voluntarily) disconnect from the network; a disconnected MH is not reachable from the rest of the network. This means that a (disconnected) MH may not be available to provide its local checkpoint. Lastly, an MH is not equipped with stable storage; disk space at an MH is not considered stable due to the vulnerability of MHs to loss, theft and physical damage. Therefore, an alternative stable repository is required to save local checkpoints of MHs. This paper presents a checkpointing algorithm for MHs that satisfies these constraints.

Lalit Kumar Awasthi et al [9] Coordinated checkpointing is a method that minimises the number of processes to the checkpoint for an initiation. It may require blocking of processes, extra synchronisation messages or useless checkpoints. They propose a minimum process coordinated checkpointing algorithm where the number of useless checkpoints and blocking are reduced using a probabilistic approach that computes an interacting set of processes on checkpoint initiation. A process checkpoints if the probability that it will get a checkpoint request in current initiation is high. A few processes may be blocked but they can continue their normal computation and may send messages. We also modified methodology to maintain exact dependencies.

Dilbag Singh et al [10] This paper presents a methodology for providing high availability to the demands of cloud's clients. To attain this objective, failover stratagems for cloud computing using integrated checkpointing algorithms are purposed in this paper. Purposed strategy integrates checkpointing feature with load balancing algorithms and also make multilevel checkpoint to decrease checkpointing overheads. For the implementation of purpose failover strategies, a cloud simulation environment is developed, which has the ability to provide high availability to clients in case of failure/recovery of service nodes. The primary objective of this research work is to improve the checkpoint efficiency and prevent checkpointing from becoming the bottleneck of cloud data centers. In order to find an efficient checkpoint interval, checkpointing overheads has also considered in this paper. By varying rerun time of checkpoints comparison tables are made which can be used to find optimal checkpoint interval. The purposed failover strategy will work on the application layer and provide high availability for Platform as a Service (PaaS) feature of cloud computing.

Author Name	Year	Technology Used	Description
Pulak Khamrui et. al.	2015	A trusted node based checkpointing scheme	In this paper they have designed a new a trusted node based checkpointing algorithm which is very effective to avoid temporary, tentative, mutual or soft checkpoints. This scheme reduces the number of processes involved in the checkpointing scheme. Due to the absence of temporary, tentative, mutable or soft checkpoints, the number of the control message is very low.
Rachit Garg et. al.	2010	A Non-blocking Coordinated Checkpointing Algorithm	In this paper, they propose a non-blocking coordinated checkpointing algorithm for mobile computing systems, which requires only a minimum number of processes to take permanent checkpoints. They reduce the message complexity as compared to the Cao-Singhal algorithm while keeping the number of useless checkpoints unchanged. And also address the related issues like failures during checkpointing, disconnections, concurrent initiations of the algorithm and maintaining exact dependencies among processes.
Bidyut Gupta et. al.	2006	A new high performance checkpointing approach	In this paper, they presented a single phase non-blocking coordinated checkpointing algorithm suitable for mobile computing environments. The distinct advantages that make the proposed algorithm suitable for distributed mobile computing systems are the following. It produces a consistent set of

			checkpoints, without the overhead of taking temporary checkpoints; the algorithm makes sure that only minimum number of processes are required to take checkpoints in any execution of the checkpointing algorithm; it uses very few control messages and the participating processes are interrupted less number of times.
Guohong Cao et. al.	2001	checkpointing approach	In this paper, the concept of the mutable checkpoint is introduced, which is neither a tentative checkpoint nor a permanent checkpoint, to design efficient checkpointing algorithms for mobile computing systems. Mutable Checkpoints can be saved anywhere, e.g., the main memory or local disk of MHs. In this way, taking a mutable checkpoint avoids the overhead of transferring large amounts of data to the stable storage at MSSs over the wireless network
Praveen Kumar et al.	2005	synchronous checkpointing protocol	In this paper, we propose a non-intrusive minimum process synchronous checkpointing protocol for mobile distributed systems, where only the minimum number of tentative checkpoints is taken. And also optimize and the number of useless forced (mutable) checkpoints and message overheads.
Ruchi Tuli et. al.	2011	Minimum Process Coordinated Checkpointing Scheme	In this paper, studied a very fundamental problem, the fault tolerance problem, in a MANET environment and propose a minimum process coordinated checkpointing scheme. Since potential problems of this new environment are insufficient to power and limited storage capacity, the proposed scheme tries to reduce the amount of information saved for recovery. The MANET structure used in our algorithm is hierarchical based. The scheme is based on Cluster Based Routing Protocol (CBRP) which belongs to a class of Hierarchical Reactive routing protocols.
Suparna Biswas et.al.	2012	Mobility based Checkpointing and trust based recovery	Proposed work is a mobility based checkpointing and trust based rollback recovery algorithm to provide fault tolerance in Mobile Ad hoc Network (MANET). Here each mobile host maintains a count of a number of clusters a mobile host traverses through, during a single checkpoint interval. A mobile host increments 'cluster-change-count' by 1, each time it leaves a cluster and joins another. Each mobile host saves a checkpoint independently if its 'cluster-change-count' exceeds a predefined threshold. This measure is important because each mobile host leaves its last checkpoint and logs at different clusters that it has visited earlier.
A.Acharya et. al.	1994	Checkpointing distributed applications	The integration of mobile/portable computing devices within existing data networks can be expected to spawn distributed applications that execute on mobile hosts (MHs). For reliability, it is vital that the global state of such applications be checkpointed from time to time. A global checkpoint consists of a set of local checkpoints, one per participant. This paper first identifies the problems in recording a consistent global state of mobile distributed applications
Lalit Kumar Awasthi et. al.	2007	A synchronous checkpointing protocol	Coordinated checkpointing is a method that minimises the number of processes to the checkpoint for an initiation. It may require blocking of processes, extra synchronisation messages or useless checkpoints. They propose a minimum process coordinated checkpointing algorithm where the number of useless checkpoints and blocking are reduced using a probabilistic approach that computes an interacting set of processes on checkpoint initiation. A process checkpoints if the probability that it will get a checkpoint request in current initiation is high. A few processes may be blocked but they can continue their normal computation and may send messages. We also modified methodology to maintain exact dependencies.

Dilbag Singh et. al.	2012	integrated multilevel checkpointing algorithms	This paper presents a methodology for providing high availability to the demands of cloud's clients. To attain this objective, failover stratagem for cloud computing using integrated checkpointing algorithms are purposed in this paper. Purposed strategy integrates checkpointing feature with load balancing algorithms and also make multilevel checkpoint to decrease checkpointing overheads. For the implementation of purpose failover strategies, a cloud simulation environment is developed, which has the ability to provide high availability to clients in case of failure/recovery of service nodes.
----------------------	------	--	--

REFERENCES

- [1] Khamrui, Pulak, and Koushik Majumder. "A trusted node based checkpointing scheme for mobile ad-hoc networks (MANETs)." *Electronics and Communication Systems (ICECS), 2015 2nd International Conference on.* IEEE, 2015.
- [2] Garg, Rachit, and Praveen Kumar. "A Nonblocking Coordinated Checkpointing Algorithm for Mobile Computing Systems." *IJCSI International Journal of Computer Science Issues* 7.3 (2010): 41-46.
- [3] Gupta, Bidyut, Shahram Rahimi, and Ziping Liu. "A new high-performance checkpointing approach for mobile computing systems." *IJCSNS International Journal of Computer Science and Network Security* 6.5B (2006): 95-104.
- [4] G. Cao, and M. Singhal, "Mutable Checkpoints: a new checkpointing approach for mobile computing systems," IEEE Transactions on Parallel and Distributed Systems, vol. 12, Issue 2, pp. 157-172, Feb 2001.
- [5] P. Kumar, L. Kumar, R.K. Chauhan, and V.K. Gupta, "A non-intrusive minimum process synchronous checkpointing protocol for mobile distributed systems", ICPWC 2005, IEEE International Conference on Personal Wireless Communications, pp. 491-495, Jan 2005.
- [6] R. Tuli & P. Kumar, "Minimum Process Coordinated Checkpointing Scheme for Ad-Hoc Networks", International Journal on AdHoc Networking Systems (IJANS), Vol. 1, No. 2, pp. 51-63, October 2011.
- [7] S. Biswas, S. Neogy and P. Dey, "Mobility-based Checkpointing and trust based recovery in MANET", International Journal of Wireless & Mobile Networks (IJWMN), Vol. 4, No. 4, pp. 53-69, August 2012.
- [8] Acharya, Arup, and B. R. Badrinath. "Checkpointing distributed applications on mobile computers." *Parallel and Distributed Information Systems, 1994. Proceedings of the Third International Conference on.* IEEE, 1994.
- [9] Awasthi, Lalit Kumar, and Prashant Kumar. "Asynchronous checkpointing protocol for mobile distributed systems: a probabilistic approach." *International Journal of Information and Computer Security* 1.3 (2007): 298-314.
- [10] Singh, Dilbag, Jaswinder Singh, and Amit Chhabra. "Evaluating overheads of integrated multilevel checkpointing algorithms in cloud computing environment." *International Journal of Computer Network and Information Security* 4.5 (2012): 29.