IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 5, May 2016.
www.ijiset.com
ISSN 2348 – 7968

Comparison of Disk-Based and Memory-Based Checkpoint Schemes Clustering and Recovery Mechanism Using Check Pointing for Mobile Ad-Hoc Networks

Vikas Kumar[1], Dr. A. K. Solanki[2]

[1] Department of Computer Science and Engineering, Bhagwant University, Ajmer, Rajasthan, India

[2] Department of Computer Science and Engineering, BIET, Jhansi, Uttar Pradesh, India

Abstract

In this paper we are presenting a comparison of disk based and memory based check pointing schemes clustering and recovery mechanism using check pointing for mobile ad hoc network. A Mobile Ad hoc network is a group of mobile nodes with self organizing protocol that create a temporary network without any centralized management or infrastructure. Check pointing each process stores its data to memory of two different processors. It has faster memory accessing capability, low checkpoint overhead and faster restart to achieve better performance than disk-based checkpoint. In this paper we also focus on the comparison of disk based and memory based check pointing.

Keywords: *disk based Memory Based, Check Pointing Domain Effect, and Network.*

1. Introduction

A Mobile Ad hoc network consists of a group of mobile nodes that self-configure to form a temporary network without the aid of a preset infrastructure or centralized management. Such networks are characterized by: dynamic topologies, existence of bandwidth constrained, variable capacity links, and energy con-strained operations and highly prone to security threats. Due to all these features routing is a major issue in mo-bile ad hoc networks. Routing in a network is the process of selecting paths to send network traffic. Routing can take place either in a flat structure or in a hierarchical structure. In a flat structure, all nodes in the network are in the same hierarchy level and thus have the same role. Although this approach is efficient for small networks, it does not allow the scalability when the number of nodes in the network increases. In large networks, the flat routing structure produces excessive information flow which can saturate the network. Hierarchical routing protocols have been proposed to solve this problem among others. This approach consists of dividing the network into groups called clusters. This results in a network with hierarchical structure. Different routing schemes are used between clusters (inter-cluster) and within clusters (intra- cluster).

1.1 Clustering

Reducing the volume of inter-node communication by localizing data transmission within the formed clusters and decreasing the number of transmissions to the sink node.

1.2 Clustering Scheme in Mobile Ad hoc Network

A Mobile Ad hoc network (MANET) consists of a group of mobile nodes that self-configure to form a temporary network without the aid of a preset infrastructure or centralized management. Such networks are characterized by: dynamic topologies, existence of bandwidth constrained, variable capacity links, and energy con-strained operations and highly prone to security threats.

1.3 Recovery Line

Figure 1 indicates the bars of the checkpoint processes. Also the system's state indicates that process A_2 has sent the message D_3 but process A_0 has not yet received it. In such situation, if A_0 fails and rolls back to the state represented by the checkpoints G_{00}, then the system goes to an inconsistent global state because the state of A_2 indicates that it has sent D_3 to A_0 and the A_0 does not indicate that it has received D_3.

The consistency of the global system's state depends on how the recovery protocol deals with in-transit messages. If the rollback recovery protocol assumes that the message channels are reliable, then the global state in figure 1 is inconsistent and D_3 is a lost message. On the other hand, if the rollback recovery protocol assumes that the message channels are unreliable, this global state is consistent and D_3 is an in-transit message. The example of figure 2 also shows an inconsistent state because the state of process A_0

ISSN 2348 – 7968

considers that A_0 has received D_3 but the state of process A_2 does not considers that $A_{2\,]}$has sent D_3. In this case, D_3 is an orphan message.

If a set of checkpoints of the system, i.e., a system global state, satisfy the following restrictions, then it is a recovery line and the recovery protocol can use it as a recovery point.

- The set contains only one checkpoint for each process.
- For a given set, there is no send event succeeding the recovery point of a sender process P whose equivalent receive event in the destination process Q occurs before the recovery point of Q in the set (no orphan messages).

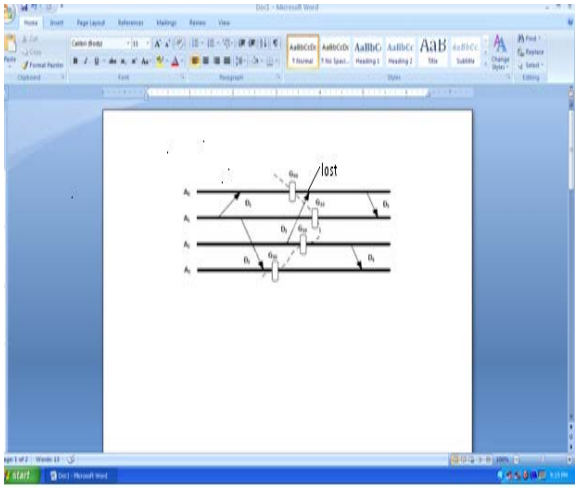

Fig.1. Inconsistency caused by a lost message.

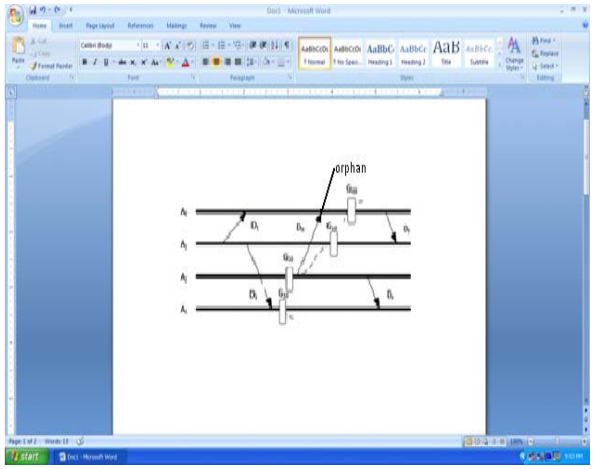

Fig.2. Inconsistency caused by orphan message

2. Phases of Check pointing

Check pointing has two phases:
• Saving a checkpoint
• Checkpoint recovery following the failure.
To save a checkpoint, the memory and system, necessary to recover from a failure is sent to storage. Checkpoint recovery involves restoring the system state and memory from the checkpoint and restarting the computation from the checkpoint stored.

3. Types of Check pointing

There are following types of check pointing:

- Disk based check pointing
- Disk less check pointing
- Double check pointing

3.1 Disk Based Check pointing

In checkpoint based methods, the state of the computation as a checkpoint is periodically saved to a stable storage, which is not subject to failures. When a failure occurs the computation is restarted from one of these previously saved states. According to the type of coordination between different processes while taking checkpoints, checkpoint-based methods can be broadly classified into three categories:
i) Uncoordinated check pointing or asynchronous check pointing
ii) Coordinated check pointing or synchronous check pointing
iii) Communication-induced or Quasi-Synchronous or Hybrid Check pointing

3.2 Diskless Check pointing

It is a technique for distributed system with memory and processor redundancy. It requires two extra processors for storing parity as well as standby. Process migration feature has ability to save a process image. The process can be resumed on the new node without having to kill the entire application and start it over again. It has memory or disk space .In order to restore the process image after a failure, a new processor has to be available to replace the crashed processor. This requires a pool of standby processors for multiple unexpected failures. The comparison between disk based and disk less check pointing for distributed and parallel system in certain parameter is described in table 1.

ISSN 2348 – 7968

Table 1: On Disk and Disk less check pointing for Distributed system

Parameter	***Disk Based***	***Diskless***
Latency time	High	Low
CPU Overhead	High	High
Memory Requirement	Low	High
Stable Storage Requirement	High	Low
Toleration of Wholesale Failure	Yes	No
Reliability	High	Low
Efficiency	Low	High
Addition Hardware	Not Required	Additional Processors
Portability	High	Low

3.3 Double Check pointing

Double check pointing targets on relatively small memory footprint on very large number of processors when handles fault at a time, each checkpoint data would be stored to two different locations to ensure the availability of one checkpoint. In case, one is lost, other can be used since two buddy processors have identical checkpoints. It can be stored either in the memory or local disk of two processors. These are double in-memory check pointing and double in-disk check pointing schemes. This scheme stores checkpoint in a distributed fashion to avoid the network bottleneck to the central server. The comparison between Disk-based and Memory-based Checkpoint in certain parameter is described in table 2.

3.3.1 Double In-memory Check pointing

In this check pointing each process stores its data to memory of two different processors. It has faster memory accessing capability, low checkpoint overhead and faster restart to achieve better performance than disk-based checkpoint. But it will increase the memory overhead and initiate check pointing at a time when the memory footprint is small in the application. This can be applied to many scientific and engineering applications such as molecular dynamics simulations that are iterative.

Table 2: Comparison of Disk-based and Memory-based Checkpoint Schemes

Fault Tolerant Protocols Shrink/Expand	***Double In Memory Yes***	***Double in Disk Yes***
Portability	Low	Low
Foolproof	NO	NO
Diskless	Yes	No, Local Disk
Halts job	No	No
Bottleneck	No	No
Require Backup Processors	Not Necessarily	Not Necessarily
Transparent Checkpoint	No	No
Synchronized Checkpoint	Yes	Yes
Automatic Restart	Yes	Yes

3.3.2 Double In-disk Check pointing

It is useful for applications with very big memory footprint where checkpoints are stored on local scratch disk instead of in processor memory. Due to the duplicate copies of checkpoints it doesn't rely on reliable storage. It incurs higher disk overhead in check pointing but does not suffer from the dramatic increase in memory usage as in the double in-memory check pointing. Taking advantage of distributed local disks, it avoids the bottleneck to the central fileserver.

4. Conclusions

In this paper we also focal point on the comparison of disk based and memory based check pointing. In this paper we are present a comparison of disk based and memory based check pointing schemes clustering and recovery mechanism using check pointing for mobile ad hoc network. In this check pointing each process stores its data to memory of two different processors. It has faster memory accessing capability, low checkpoint overhead and faster restart to achieve better performance than disk-based checkpoint. In checkpoint based methods, the state of the computation as a checkpoint is periodically saved to a stable storage, which is not subject to failures. When a failure occurs the computation is restarted from one of these previously saved states. It is a technique for distributed system with memory and processor redundancy. It requires two extra processors for storing parity as well as standby. Process migration feature has ability to save a process image. The process can be resumed on the new node without having to kill the entire application and start it over again.

References

[1] Guo-Hui Li, Hong-Ya Wang, A Novel min-process checkpointing scheme for mobile computing systems: Journal of system Architecture,51(1), 2005.
[2] Qiangfeng Jiang, Yi Luo and D. Manivannan, An Optimistic Checkpointing and message Logging approach for

ISSN 2348 – 7968

consistent global checkpoint Collection in distributed Systems: Journal of Parallel and Distributed Computing ,68(12) ,pp. 1575-1589, 2008.
[3] Bidyut Gupta, S.Rahimi and Z.Lui. A New High Performance Checkpointing Approach for Mobile Computing Systems: IJCSNS International Journal of Computer Science and Network Security,6(5B), 2006.
[4] Lalit Kumar Awasthi, A Synchoronous Checkpointing Protocol For Mobile Distributed Systems: Probabilistic Approach. Int J. Information and Computer Security, 1(3) , pp. 298-314, 2007.
[5] J. L. Kim and T. Park. An efficient protocol for checkpointing recovery in Distributed Systems: IEEE Transaction On Parallel and Distributed Systems,4(8),pp.955-960, 1993.
[6] D.J. Baker and A. Ephremides, The Architectural Organisation of a Mobile Radio Network via a Distributed algorithm: IEEE Trans. Commun., vol. 29, no. 11, pp 1694-1701, 1981.
[7] D.J. Baker, A. Ephremides and J.A. Flynn, The design and Simulation of a Mobile Radio Network with Distributed Control: IEEE J. sel. Areas Commun.., pp 226-237, 1984.
[8] B.Das, R. Sivakumar and V. Bharghavan, Routing in Ad-hoc networks using a Spine: Proc. Sixth International Conference, 1997.
[9] B.Das, R. Sivakumar and V. Bharghavan, Routing in Ad-hoc networks using Minimum connected Dominating Sets: Proc. IEEE International Conference, 1997.
[10] M.Gerla, G. Pei, and S.J. Lee, Wireless Mobile Ad-hoc Network Routing: Proc. IEEE/ACM FOCUS'99, 1999.
[11] Iman, S.; Adnan, A.; Mohamed, E. In-network fault tolerance in networked sensor systems: In Proceedings of the Workshop on DEPENDABILITY ISSUES in WIRELESs Ad Hoc Networks and Sensor Networks, Los Angeles, CA, USA, pp. 47-54, 2006.
[12] S.Monnet, C.Morin, R. Badrinath, Hybrid check pointing for parallel in cluster federation: In 4th IEEE/ACM International symposium on cluster computing and the Grid, Chicago, USA, pp 773-782, 2004.

Vikas Kumar is a research scholar of Dept. of Computer Science in Bhagwant University, Ajmer, Rajasthan. He has published a lot of National and International (Review and Research) paper in Computer Science.

Dr. A. K. Solanki working as a Professior in BIET Janshi (U.P). He Have passed M.Tech (CSE) and P.hd. He has published several no. of books on field of Computer Science and he has published several number of national and International paper in Computer Science.