International Journal of Electronics, Communication & Soft Computing Science and Engineering
ISSN: 2277-9477, Volume 2, Issue 9

Exploring Various Aspects of Remote Checkpoint Strategy for Mobile Devices

Ms. S. R. Kaware   Prof. P.L. Ramteke

Abstract- **Mobile computing systems often suffer from high failure rates that are transient and independent in nature. To add reliability and high availability to such distributed systems, checkpoint and rollback scheme is a useful fault-tolerance method for mobile devices in wireless environments. It may possible to reduce the loss of computation of a process in the presence of failures by periodically saving the process's state on stable storage as a checkpoint and rolling back to the latest checkpoint when a failure has occurred. However, a mobile consumer device is not considered to have sufficiently large and stable storage to store its checkpoint data. Therefore, a remote checkpoint technique is preferred in wireless technique is preferred in wireless environments in which the checkpoint data of a mobile device is kept in a remote checkpoint server instead of the mobile device. Since battery power is one of the most critical resources for mobile devices, it is important to identify optimal checkpoint intervals that minimize power consumption. The proposed method is used to recover the mobile devices from failures and minimizes power consumption in wireless remote checkpoint environments by finding the optimal checkpoint interval.**

Keywords- **Mobile Consumer Device, Fault-tolerance, Remote checkpoint, Wireless Networks.**

I. INTRODUCTION

A mobile computing system is a distributed system where some of the nodes are mobile computers. It consists of a fixed node called Mobile Support Station (MSS) and a number of Mobile Hosts (MHs). A cell is a geographical area around a MSS in which it can support a MH. A mobile Host can change its geographical position freely from one cell to another or even to an area covered by no cell. All the communication from one Mobile Host to another Mobile Host goes through MSS. MSS has both types of links—wired and wireless links. A MSS communicate with Mobile Host by wireless links, while with other MSSs by wired links. In ubiquitous environments, mobile consumer electronic devices are getting smarter and smaller to provide interactive intelligent services to users by collaborating with internet services. For example, ubiquitous well-being services can be provided by employing various wearable biological sensing devices, which collect user's heartbeat, body heat, and so on, and transmit the collected information to the user's smart phone or internet health-care servers through wireless communications [1].

Although computer hardware is more reliable in this decade compared to previous decades, smart mobile consumer devices such as wearable smart sensors (e.g., smart pendants [1]), computational RFIDs, or smart phones tend to fail for a variety of reasons. The high complexity of emerging software architectures increases the occurrence of transient faults during the computation. Memory allocation faults can easily occur on a mobile device that usually has a small volume of memory, because small memory tends to be fully occupied by memory leakage bugs quicker than large memory does. External damage or battery drain also easily occurs on mobile devices. Therefore, an effective fault tolerant strategy should be employed in mobile consumer devices. However, employing hardware-based fault-tolerant schemes (e.g., hardware replications) to mobile devices is not realistic [2], though they are quite simple and effective. Therefore, software-based fault-tolerance techniques that do not require additional hardware resources are desirable to mobile devices. A checkpoint and rollback recovery technique is an efficient software-based fault tolerance strategy that does not require additional hardware resources [3]. We can reduce the loss of computation of a process by periodically saving the process's state on stable storage as a checkpoint and rollback to the most recent checkpoint when a failure has occurred [4], [5].

In wireless environments, a mobile consumer device is not considered to have sufficiently large and stable storage to store its checkpoint data, because mobile devices have a lack of stable storage, small memory, limited battery capacity, and are prone to external damage. Therefore, it is better to save the checkpoint data of a mobile device into the stable storage of a remote server through a wireless interface and download the last checkpoint from the server when the device has failed [6], [7], [8].

However, energy expenditure is a big concern for mobile devices nowadays, because battery power of mobile devices is now one of the most crucial resources. The proposed method is used for determining an approximation to the optimal checkpoint interval that minimizes the energy expenditure in wireless remote checkpoint environments.

II. LITERATURE REVIEW

Employing a checkpoint and rollback scheme on a mobile device is necessary for fault tolerance, because spatial fault-tolerance schemes such as hardware replications cannot be applied [2]. Checkpoint and rollback recovery techniques provide software-based tolerance against transient and intermittent faults to the device. By restarting from the last checkpoint instead of the beginning of the execution for a process, the loss of useful computations caused by failure can be reduced [5]. The length of the checkpoint interval (i.e., how frequently checkpoints should be established during useful computation) has great impact on the performance of the system. Several studies have presented approximate solutions to determine optimal checkpoint intervals to minimize the expected execution time, which includes the time required for useful computation as well as the time wasted by failures and overhead for checkpoint and rollback.

Many research papers have proposed approximated methods to find out the optimal checkpoint interval that minimizes the expected execution time. Young has proposed a first order approximation to compute the optimal checkpoint interval according to the failure rate and the checkpoint overhead [3] for traditional checkpoint systems. Daly proposed a perturbation solution providing a higher order approximation [10] than Young's approximation [3]. Zhan et Al. presented a heuristic method for remote checkpoint systems which dynamically adjusts the checkpoint frequency according to variety of failures when the failure rate is assumed to change over time in wireless networks [2]. Men et al. proposed a remote checkpoint method to obtain the appropriate checkpoint interval by considering not only the failure rate and checkpoint overhead but also the handoff rate of a mobile device in cellular networks when the mobile device frequently moves across the cellular areas [9]. George et al. proposed an aperiodic remote checkpoint scheme in which a mobile device takes checkpoints only when its handoff rate exceeds a predefined threshold value instead of taking checkpoints periodically [11].

These studies have been focused on reducing the expected execution time, but power consumption is also an important performance metric in mobile devices. Power consumption is not a linear function of time precisely because of the establishment of checkpoints. Otherwise, these studies would be sufficient to optimize for power consumption. Most research on remote checkpoint strategies has focused on reducing message overhead between mobile hosts and the mobile support server [12, 13, 14], but determining optimal checkpoint intervals is also important for the time and power efficiency. However, previous studies on optimal checkpoint intervals have been focused on reducing the expected execution time but power consumption is also an important performance metric in mobile devices [3, 4, 10].

As a previous work, an approximated method to the energy optimal checkpoint interval to minimize the expected energy expenditure for a mobile device has been proposed [15]. However, previous work did not consider the wireless link error, while wireless link errors occur more frequently than local faults do on a mobile device, generally. Therefore, a method to determine the optimal checkpoint interval should be developed, that compensates not only for local faults on the mobile device (e.g., memory faults) but also wireless link errors.

III. ANALYSIS OF PROBLEM

The high complexity of emerging software architectures increases the occurrence of transient faults during the computation. Memory allocation faults can easily occur on a mobile device that usually has a small volume of memory, because small memory tends to be fully occupied by memory leakage bugs quicker than large memory does. External damage or battery drain also easily occurs on mobile devices. Unlike a workstation in a fixed network, a Mobile Host engaged in a client-server application in a wireless network can easily fail because of limited resources.

Mobile devices usually lack sufficient and stable storage to store checkpoint data. Remote checkpointing requires wireless transmissions that greatly increase the power consumption of mobile devices. It is clear that much more power is required to transmit or receive data through wireless interfaces than to access local memory. Battery power is one of the most critical resources for mobile devices, and it is therefore important to minimize power consumption. Therefore, it is necessary to identify optimal points (i.e., optimal checkpoint intervals) that minimize power consumption by considering the following two aspects- Failure rate of the mobile device and Power for storing checkpoint data. It is highly desired to employ a checkpoint and rollback scheme on a mobile consumer device for fault tolerance, because spatial fault-tolerant schemes such as hardware replications cannot be used.

IV. Remote Checkpoint Strategy

When a mobile device fails, its local state is corrupted and all the useful computation done before the failure occurred is lost [4]. To relive this loss of computation, the mobile device periodically saves the current state of its running process as checkpoints, and recovers from a failure by a rollback to the most recent checkpoint. A checkpoint is a copy of current process's state, which is stored on stable storage. An example of the checkpoint and rollback process is shown in Figure 1. The proposed system uses Non-blocking Checkpoint Co-ordination protocol.

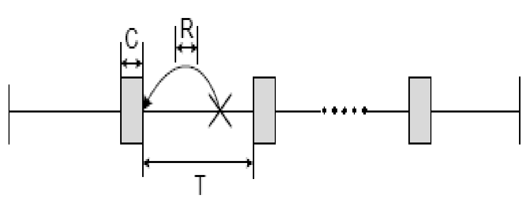

Figure 1. Rollback and checkpoint scheme

A checkpoint is said to have been established, if the execution of the process can start over from this checkpoint by a rollback and recovery in the presence of failure [4]. Establishing a checkpoint incurs time overhead denoted by C, and rolling back to the last checkpoint when a failure has occurred requires time overhead denoted by R. It is assume that C and R are constants for simplicity, though a more elaborate model may assume C and R to be some function of time. The execution of a process can be considered as a series of intervals, each of which begins immediately after a checkpoint is established and ends after the next checkpoint is successively established. It is assumed that every interval between consecutive checkpoints for a process has the same time length T of useful computation as presented in figure 1. At least, time T+C is required to complete one checkpoint interval is even if any failures are not occurred.

For simplicity, it is assumed that a failure occurs once a fault has occurred. In cases of failure, the mobile device is assumed to lose current states of running processes in the memory. The mobile device may fail during normal, checkpoint, or recovery operations. Failure requires rollback to the last established checkpoint. When a mobile device fails during execution, its local processes running in memory are corrupted, and all of the useful computation executed before the failure is lost [4]. It is assumed that wireless communication, as well as local memory access, is required during checkpoint establishment (C) or rollback recovery (R), whereas only local memory access is required during normal local computation (i.e., useful computation).

A mobile device is able to access the Internet by connecting to a mobile support server through the wireless interface. System can employ a simple remote check pointing mechanism for mobile devices as presented in [15], in which an application process of the mobile device periodically stores checkpoint data on the mobile support server, and loads the last checkpoint from the server when failures occur. Actually, in more practical models, the remote checkpoint server may not also be the mobile support server but may be another dedicated remote internet server.

It has been one of the main objectives of the research on checkpoint and rollback schemes to reduce the total overhead which consists of the loss of computation and the checkpoint overhead and to minimize energy expenditure of a mobile device in a wireless remote checkpoint environment.

V. CONCLUSION

Designing a fault tolerant system is always difficult. Fault tolerance using checkpoints in a Mobile Computing System imposes more challenges because of some unique characteristics of mobile hosts. Proposed checkpointing strategy is a complete one in comparison with other relevant works. It determines the optimal checkpoint intervals that minimize power consumption in wireless remote checkpoint environments. Unike others, MHs moving within a cell is checkpointed exclusively. checkpointing can be completed without any delay resulting enhanced fault tolerance in the proposed scheme. Hence, checkpointing scheme is stronger from the point of view of failure recovery.

REFERENCES

[1]. H.-N. Lee, Sung-Hwa Lim, J.-H. Kim, "UMONS: Ubiquitous Monitoring System in Smart Space," IEEE Transactions on Consumer Electronics, vol. 55, no. 3, pp.1056-1064, Aug. 2009.

[2]. Z. Zhan, Z. De-chaeng, C. Yi-wei, and Y. Xian-zong, " The Checkpoint Interval Optimization of Kernel-level Rollback Recovery based on the Embedded Mobile Computing System," IEEE International Conference on Computer and Information Technology Workshops, Australia, July 2008.

[3]. J. W. Young, "A first order approximation to the optimum checkpoint interval," Communications on the ACM, vol. 17, pp. 530-531, Sept. 1974.

[4]. N. H. Vaidya, On checkpoint latency, in \Proceedings of Pacific Rim International Symposium on Fault-Tolerant Systems," (1995), 60-65.

[5]. K. M. Chandy, J. C. Browne, C. W. Dissly and W. R. Uhrig, Analytic models for rollback and recovery strategies in data base systems, IEEE Trans. Software Eng., 1 (1975), 100-0110.

[6] K.-S. Byun, Sung-Hwa Lim and J.-H. Kim, "Two-Tier Checkpointing Algorithm Using MSS in Wireless Network," IEICE transactions on Communications, vol. E86-B, no. 7, pp. 1-7, July 2003.

[7] G.-H. Li, H. Wang, "A Novel Min-process Checkpointing scheme for Mobile Computing Systems," Journal of Systems Architecture, vol. 51, pp. 45-61, 2005.

[8] I.-R. Chen, B. Gu, S. E. George, and S.-T. Cheng, "On Failure Recoverability of Client-Server Applications in Mobile Wireless Environments," IEEE Transactions on Reliability, vol. 54, no. 1, pp. 115-122, 2005.

[9] C. Men, and Z. Xu, "Performance Analysis of Rollback Recovery Schemes for the Mobile Computing Environment," International Conference on Internet Computing in Science and Engineering, Aug. 2008.

[10]. J. T. Daly, A higher order estimate of the optimum checkpoint interval for restart dumps, Future Generation Computer Systems, 22 (2006), 303-312.

[11] S. E. George, I.-R. Chen, Y. Jin, "Movement-based checkpointing and logging for recovery in mobile computing systems," ACM International Workshop on Data Engineering for Wireless and Mobile Access, U.S.A., June 2006

[12]. G.-H. Li and H. Wang, A novel min-process checkpointing scheme for mobile computing systems, J. Systems Architecture, 51 (2005), 45-61.

[13]. S. Biswas and S. Neogy, A low overhead checkpointing scheme for mobile computing systems, in Proceedings of the International Conference on Advanced Computing and Communications," IEEE, (2007), 700-705.

[14]. P. Kumar, P. Gupta and A. K. Solanki, Dealing with frequent aborts in minimum-process coordinated checkpointing algorithm for mobile distributed systems, Int. J. Computer Applications, 3 (2010), 7-12.

[15]. S.-H. Lim, S. W. Lee, B.-H. Lee, S. Lee and H. W. Lee, Energy-aware checkpoint intervals in error-prone mobile networks, in proceedings of QTNA, 2011 (2011), 128-133.

AUTHOR'S PROFILE

Ms. S. R. Kaware
ME II Year
Computer science and Engineering
HVPM College of Engineering & Technology,
Amravati, India.
snehakaware@gmail.com

Prof. P.L. Ramteke
HOD (IT)
HVPM College of Engineering & Technology,
Amravati, India.
pl_ramteke@rediffmail.com