Checkpointing for the Reliability of Real-Time Systems with On-Line Fault Detection

Sang-Moon Ryu and Dong-Jo Park

Korea Advanced Institute of Science and Technology
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea
smryu@kaist.ac.kr, djpark@ee.kaist.ac.kr

Abstract. The checkpointing problem in real-time systems equipped with on-line fault detection mechanisms is dealt with from a reliability point of view. The reliability analysis is performed with the assumption that transient faults occur in accordance with a Poisson process and are detected immediately by the detection mechanisms. And the best equidistant checkpointing strategy that maximizes the reliability of the system against transient faults is derived.

1 Introduction

Transient faults in semiconductor devices are becoming more significant because of increased density, low supply voltage, fast switching signals and so on [1]. Checkpointing is a well known technique to overcome transient faults in computer systems. It means periodically saving the state of a task in a safe storage place. When the manifestation of a transient fault is detected, the state of the affected task will be restored to the state stored at the latest checkpoint. This process is called rollback-recovery. The specific points at which checkpointing is performed are called checkpoints and the length of the time between two successive checkpoints is said to be a checkpoint interval.

Many papers dealt with the problems of checkpointing in real-time systems from various points of view [2–6]. In this paper, the reliability problem of equidistant checkpointing, which relies on the use of a constant checkpoint interval, in a single task real-time system under transient faults is explored. The transient faults are assumed to occur according to a Poisson process and be detected by on-line detection mechanisms [7, 14–16] with no latency. The reliability of the system with equidistant checkpointing is analyzed and the best checkpointing strategy is derived, which achieves the maximum probability of successful task completion with given parameters such as task execution time, available slack time, checkpointing and recovery overheads.

The following assumptions were made in this work, which were used in other literature:

– Transient faults occur according to a Poisson process with rate λ, which is common in many papers dealing with transient faults [2–5, 8, 10–13].

This work was supported by SaTReC of KAIST.

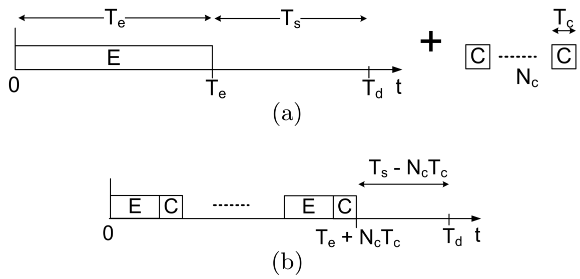

Fig. 1. The task (a) before inserting checkpoints (b) after inserting checkpoints.

– The effect of a transient fault disappears during the corresponding recovery operation.
– The manifestation of transient faults is perfectly detected by the detection mechanisms [3–5, 10]. If necessary, the fault detection coverage [14] may be taken into account after the reliability model is obtained.
– Checkpointing is possible anywhere in the task with a constant checkpoint interval [5]. In practice, it might be difficult to accomplish. But the result of analysis with equidistant checkpointing can give an insight into how checkpoints should be inserted to improve the reliability of a system.
– Checkpointing and recovery overheads are same and remain constant [5].

2 Reliability Analysis

If N_c checkpoints are inserted uniformly into the task whose worst case execution time, relative deadline, slack time, and checkpointing/recovery overhead are T_e, T_d, T_s, and T_c, respectively, as shown in Fig. 1(a), the task is divided into N_c subsections creating N_c time-slots as illustrated in Fig. 1(b). Each time-slot is composed of a part of normal execution and a checkpointing operation, and its length is $\frac{T_e}{N_c} + T_c$. Due to the checkpointing overheads, the available slack time of the task is reduced from T_s to $T_s - N_cT_c$.

With on-line fault detection mechanisms, the recovery process can be performed immediately after a fault occurrence as shown in Fig. 2. To tolerate transient faults that may occur during either checkpointing or recovery operation, at least two secure storage places should be provided and used for checkpointing alternately. The storage place where the earlier state was saved should be used for the current checkpointing operation and the other one where the more recent state was saved should be reserved for a recovery operation in case of a fault occurrence during the current checkpointing operation.

Transient faults can be classified into two types: One type is those which may occur during normal task execution or during checkpointing operations (fault type 'E' and 'C' in Fig. 2), and the other type is those which may occur during recovery operations (fault type 'R' in Fig. 2). Transient faults of the

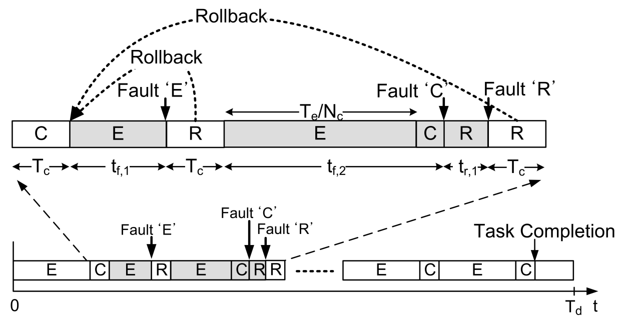

Fig. 2. Checkpointing with on-line fault detection.

former type will be counted to N_f and transient faults of the latter type will be counted to N_r. Then the sum of N_f and N_r is the total number of transient faults that may occur in $[0,\ T_d]$. As shown in Fig. 2, let the time elapsed since the last checkpointing when a fault occurs during normal execution or checkpointing operations be denoted by $t_{f,i}$, $i = 1, 2, \cdots, N_f$, and the time elapsed since the beginning of a recovery operation when a fault occurs during the recovery operation be denoted by $t_{r,l}$, $l = 1, 2, \cdots, N_r$ and define T_t as $\frac{T_e}{N_c} + T_c$. Then the time intervals $t_{f,i}$ and $t_{r,l}$ can be thought of as independent identically distributed (i.i.d.) random variables with exponential distributions on intervals $[0,\ T_t]$ and $[0,\ T_c]$, respectively.

Now we derive the probability of task completion in the presence of transient faults. Since theoretically the number of faults of type 'E' or 'C' in interval $[0,\ T_d]$ ranges from 0 to ∞, N_f ranges from 0 to ∞ as well. A fault of type 'R' can be brought about only after a fault of type 'E' or 'C' occurs. Therefore N_r can range from 0 to N_f. The probability, $P(N_c)$, that the task completes its execution in the presence of transient faults is

$$\begin{aligned} P(N_c) &= Pr\{\text{success with no fault}\} \\ &\quad + \sum_{N_f=1}^{\infty} \sum_{N_r=0}^{N_f} Pr\{\text{success with } N_f \text{ and } N_r \text{ faults}\} . \end{aligned}$$

Since we assumed that faults occur in accordance with a Poisson process, the probability of successful task completion with no fault is

$$Pr\{\text{success with no fault}\} = e^{-\lambda(T_e + N_c T_c)} . \tag{1}$$

The probability of successful task completion with N_f and N_r faults is the product of three probabilities: the probability that no fault occurs in N_c time-slots for task completion, the probability that N_f and N_r faults occur before the task completes its execution, and the probability that the time lost by these faults is small enough for the task to meet its deadline.

$$Pr\{\text{success with } N_f \text{ and } N_r \text{ faults}\} = Pr\{\text{no fault in } N_c \text{ time-slots}\}$$

$$\cdot Pr\{\text{lost time by } N_f \text{ and } N_r \text{ faults is small}\}$$
$$\cdot Pr\{N_f \text{ and } N_r \text{ faults occur}\}\,. \tag{2}$$

The first probability in the right side of (2) is the same as the probability in (1). In the following, we derive the second and the third probabilities in the right side of (2).

In order for the task to meet its deadline, N_c checkpointing operations, N_f recovery operations should be done within the slack time T_s in spite of the loss in time caused by the faults. If we define T_{ns} as $T_s - N_c T_c - N_f T_c$ and S_{fr} as $\sum_{i=1}^{N_f} t_{f,i} + \sum_{l=1}^{N_r} t_{r,l}$, the probability that the lost time by these faults is small enough for the task to meet its deadline can be expressed as

$$Pr\{\text{lost time by } N_f \text{ and } N_r \text{ faults is small}\} = P(S_{fr} \leq T_{ns})\,. \tag{3}$$

In order to get the probability in (3), we have to find out the probability density function (pdf) of S_{fr}. It is already known that the sum of exponential random variables has the Erlang distribution [17]. If T_i's, $i = 1, 2, \cdots, n$, denote i.i.d. exponential random variables, and S_n denotes the sum of these exponential random variables, i.e., $S_n = T_1 + T_2 + \cdots + T_n$, the corresponding pdf's are, respectively,

$$f_{T_i}(x) = \lambda e^{-\lambda x},\ x \geq 0 \tag{4}$$

and

$$f_{S_n}(x) = \frac{(\lambda x)^{n-1}}{(n-1)!}\lambda e^{-\lambda x},\ x \geq 0\,. \tag{5}$$

From the definition of the characteristic function of a random variable [17], the characteristic functions of T_i and S_n are, respectively,

$$\Phi_{T_i}(\omega) = \frac{\lambda}{\lambda - j\omega} \tag{6}$$

and

$$\Phi_{S_n}(\omega) = \left(\frac{\lambda}{\lambda - j\omega}\right)^n\,. \tag{7}$$

The characteristic function of S_{fr} can be derived from those of $t_{f,i}$ and $t_{r,l}$. Since the time intervals $t_{f,i}$ and $t_{r,l}$ are random variables with exponential distribution on intervals $[0,\ T_t]$ and $[0,\ T_c]$, their pdf's are, respectively,

$$f_{t_{f,i}}(x) = \begin{cases} \frac{\lambda e^{-\lambda x}}{1-e^{-\lambda T_t}}, & \text{when } 0<x<T_t \\ 0, & \text{otherwise} \end{cases}$$

and

$$f_{t_{r,l}}(x) = \begin{cases} \frac{\lambda e^{-\lambda x}}{1-e^{-\lambda T_c}}, & \text{when } 0<x<T_c \\ 0, & \text{otherwise}\,. \end{cases}$$

Then the characteristic functions of $t_{f,i}$ and $t_{r,l}$ are, respectively,

$$\Phi_{t_{f,i}}(\omega) = \frac{\lambda}{\lambda - j\omega} \frac{1 - e^{-\lambda T_t} e^{jT_t\omega}}{1 - e^{-\lambda T_t}}$$

and

$$\Phi_{t_{r,l}}(\omega) = \frac{\lambda}{\lambda - j\omega} \frac{1 - e^{-\lambda T_c} e^{jT_c\omega}}{1 - e^{-\lambda T_c}} .$$

Since the characteristic function of the sum of two random variables is the product of the characteristic functions of these random variables, the characteristic function of S_{fr} is

$$\Phi_{S_{fr}}(\omega) = \left(\Phi_{t_f}(\omega)\right)^{N_f} \left(\Phi_{t_r}(\omega)\right)^{N_r} ,$$

which is

$$\begin{aligned}
\Phi_{S_{fr}}(\omega) &= \left(\frac{\lambda}{\lambda - j\omega}\right)^{N_f+N_r} \frac{1}{(1-e^{-\lambda T_t})^{N_f}} \frac{1}{(1-e^{-\lambda T_c})^{N_r}} \\
&\quad \cdot \sum_{i=0}^{N_f} \binom{N_f}{i} \left(-e^{-\lambda T_t}\right)^i e^{jT_t\omega i} \cdot \sum_{l=0}^{N_r} \binom{N_r}{l} \left(-e^{-\lambda T_c}\right)^l e^{jT_c\omega l} \\
&= \left(\frac{\lambda}{\lambda - j\omega}\right)^{N_f+N_r} \frac{1}{(1-e^{-\lambda T_t})^{N_f}(1-e^{-\lambda T_c})^{N_r}} \\
&\quad \cdot \sum_{i=0}^{N_f} \left[\binom{N_f}{i} \left(-e^{-\lambda T_t}\right)^i \sum_{l=0}^{N_r} \binom{N_r}{l} \left(-e^{-\lambda T_c}\right)^l e^{j(iT_t + lT_c)\omega} \right] .
\end{aligned}$$

Then the pdf of S_{fr} can be derived by using the relationship among (4), (5), (6) and (7) as

$$\begin{aligned}
f_{S_{fr}}(x) &= \frac{1}{(1-e^{-\lambda T_t})^{N_f}(1-e^{-\lambda T_c})^{N_r}} \\
&\quad \cdot \sum_{i=0}^{N_f} \left[\binom{N_f}{i} \left(-e^{-\lambda T_t}\right)^i \sum_{l=0}^{N_r} \binom{N_r}{l} \left(-e^{-\lambda T_c}\right)^l f(x - iT_t - lT_c) \right] ,
\end{aligned}$$

where

$$f(x) = \frac{(\lambda x)^{N_f+N_r-1}}{(N_f+N_r-1)!} \lambda e^{-\lambda x}, \; x \geq 0 .$$

Now that the pdf of S_{fr} is obtained, the probability in (3) can be calculated.

Finally, the third probability in the right side of (2) is derived, which can be decomposed as

$$\begin{aligned}
Pr\{N_f \text{ and } N_r \text{ faults occur}\} &= Pr\{N_f \text{ faults occur}\} \\
&\quad \cdot Pr\{N_r \text{ faults occur} \,|\, N_f \text{ faults occur}\} .
\end{aligned}$$

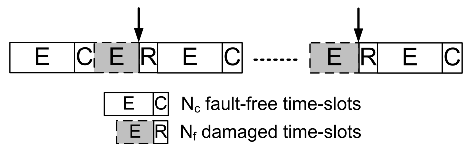

Fig. 3. Possible cases for N_f faults.

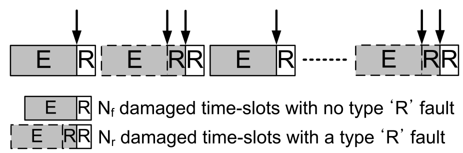

Fig. 4. Possible cases for N_r faults.

The probability that a fault occurs and damages a time-slot of length T_t is $1 - e^{-\lambda T_t}$. Among N_c fault-free time-slots, the N_f faults of type 'E' or 'C' can occur in $\binom{N_c+N_f-1}{N_f}$ ways as illustrated in Fig. 3. Therefore the probability that N_f faults occur before the task completion is

$$Pr\{N_f \text{ faults occur}\} = \binom{N_c + N_f - 1}{N_f} \left(1 - e^{-\lambda T_t}\right)^{N_f} .$$

The N_f faults of type 'E' or 'C' would bring about N_f times of recovery operations. Among the N_f recovery operations, N_r operations are damaged by the N_r faults of type 'R'. The probability that N_r recovery operations are damaged is $(1 - e^{-\lambda T_c})^{N_r}$, and the probability that $N_f - N_r$ recovery operations are performed normally is $e^{-\lambda(N_f - N_r)T_c}$. Since the N_r faults may occur among the N_f damaged time-slots in $\binom{N_f}{N_r}$ ways as shown in Fig. 4, the probability that N_r faults of type 'R' occur given that the N_f faults of type 'E' or 'C' have occurred is

$$Pr\{N_r \text{ faults occur} \mid N_f \text{ faults occur}\} = \binom{N_f}{N_r} \left(1 - e^{-\lambda T_c}\right)^{N_r} e^{-\lambda(N_f - N_r)T_c} .$$

Consequently the probability, $P(N_c)$, that the task completes its execution even in the presence of transient faults is

$$P(N_c) = e^{-\lambda(T_e + N_c T_c)} \left(1 + \sum_{N_f=1}^{\infty} \sum_{N_r=0}^{N_f} P_{fr}(N_c,\, N_f,\, N_r)\right) ,$$

where

$$P_{fr}(N_c,\, N_f,\, N_r) = \binom{N_c + N_f - 1}{N_f} \binom{N_f}{N_r} e^{-\lambda(N_f - N_r)T_c}$$

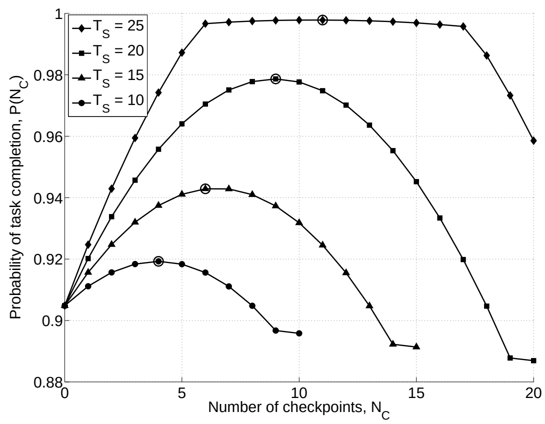

Fig. 5. $P(N_c)$ when $\lambda = 0.001$, $T_e = 100$ and $T_c = 1$.

$$\cdot \sum_{i=0}^{N_f} \left[\binom{N_f}{i} \left(-e^{-\lambda T_t} \right)^i \sum_{l=0}^{N_r} \binom{N_r}{l} \left(-e^{-\lambda T_c} \right)^l \int_0^{T_{ns}} f(x - iT_t - lT_c)\, dx \right]$$

and

$$f(x) = \frac{(\lambda x)^{N_f + N_r - 1}}{(N_f + N_r - 1)!} \lambda e^{-\lambda x}, \;\; x \geq 0\,.$$

3 Numerical Examples

Figures 5 and 6 show the graphs of $P(N_c)$ with different values of T_s and λ, respectively. In theses figures, only the terms up to $N_f = 5$ were calculated for each value of $P(N_c)$. The probability of task completion in $[0,\ T_d]$ increases as the available slack time increases. And excessive checkpointing can result in adverse effect. This is because transient faults may occur during checkpointing operations, and more checkpoints than necessary may expose the task to more transient faults and cause the net slack time to decrease leading to the lack of extra time for recovery operations in case of fault occurrences.

4 Best Checkpointing Strategy

From Figs. 5 and 6, it is apparent that there exists the best number of checkpoints which maximizes the probability of successful task completion for the given parameters, such as execution time, checkpointing overhead and available slack time. In practice, the value of λT_d is much smaller than 1. Therefore the

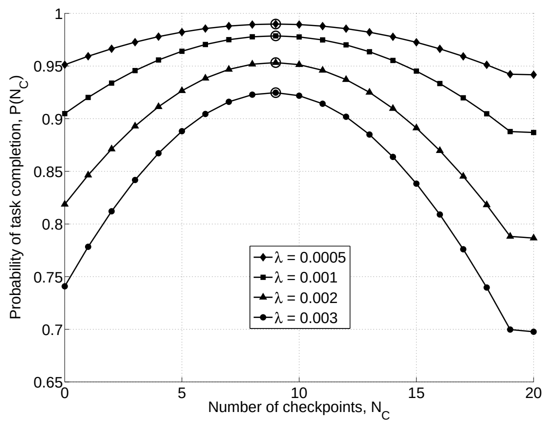

Fig. 6. $P(N_c)$ when $T_e = 100$, $T_c = 1$ and $T_s = 20$.

value of $P(N_c)$ is mainly dominated by the term resulting from $N_f = 1$ and $N_r = 0$, and can be approximated as

$$P(N_c) \approx e^{-\lambda(T_e+T_cN_c)} \left[1 + N_c \left(1 - e^{-\lambda(T_s-T_c-T_cN_c)}\right) e^{-\lambda T_c}\right] .$$

Since it is reasonable that the total checkpointing overhead, N_cT_c, is less than the execution time of the task, we have $e^{-\lambda T_c N_c} \approx 1 - \lambda T_c N_c + \frac{1}{2}(\lambda T_c N_c)^2$. And the probability $P(N_c)$ can be approximated once more as

$$P(N_c) \approx e^{-\lambda T_e} \left[1 + (b - a - bc)N_c + (\frac{a^2}{2} - ab)N_c^2 + \frac{a^2b}{2}N_c^3\right] , \tag{8}$$

where $a = \lambda T_c$, $b = e^{-\lambda T_c}$ and $c = e^{-\lambda(T_s-T_c))}$.

Then, by temporarily assuming N_c to be continuous and by solving the equation which results from taking the derivative of the right side of (8) with respect to N_c and letting it be zero, we can obtain two candidates for the value of N_c that maximizes $P(N_c)$:

$$\left\lfloor \frac{(2b-a) - \sqrt{(2b-a)^2 - 6b(b-a-bc)}}{3ab} \right\rfloor$$

and

$$\left\lceil \frac{(2b-a) - \sqrt{(2b-a)^2 - 6b(b-a-bc)}}{3ab} \right\rceil .$$

The best number of checkpoints is one of the above candidates which leads to the larger value of $P(N_c)$. In Figs. 5 and 6, the points corresponding to the best number of checkpoints for each case are marked by a circle.

5 Conclusion

In this paper, we considered the best equidistant checkpointing strategy for real-time systems from a reliability point of view with the assumption that transient faults are detected with no latency by on-line detection mechanisms. The reliability analysis shows that the reliability of the system can be improved as much as expected by providing the required slack time with the tasks of the system, and the best number of checkpoints hardly depends on the occurrence rate of transient faults.

References

1. E. Dupont, M. Nicolaidis and P. Rohr, "Embedded Robustness IPs for Transient-Error-Free ICs," IEEE Design & Test of Computers, vol. 19, pp. 56–70, May–Jun. 2002.
2. K. G. Shin, T.-H. Lin and Y.-H. Lee, "Optimal Checkpointing of Real-Time Tasks," IEEE Trans. Computers, vol. C-36, no. 11, pp. 1328–1341, Nov. 1987.
3. S. Punnekkat, A. Burns and R. Davis, "Analysis of Checkpointing for Real-Time Systems," Real-Time Systems, vol. 20, no. 1, pp. 83–102, Jan. 2001.
4. Y. Zhang and K. Chakrabarty, "Energy-Aware Adaptive Checkpointing in Embedded Real-Time Systems," Proc. Design, Automation and Test in Europe Conference and Exhibition, pp. 918–923, Messe Munich, Germany, 2003.
5. S. W. Kwak, B. K. Kim and B. J. Choi, "An Optimal Checkpointing-Strategy for Real-Time Control Systems under Transient Faults," IEEE Trans. Reliability, vol. 50, no. 3, pp. 293–301, Sep. 2001.
6. A. Ranganathan and S. J. Upadhyaya, "Simulation Analysis of a Dynamic Checkpointing Strategy for Real-Time Systems," Proc. 27th Annual Simulation Symp., pp. 181–187, Apr. 1994.
7. D. P. Siewiorek, Reliable Computer Systems: Design and Evaluation, A K Peters, 1998.
8. A. Duda, "The Effects of Checkpointing on Program Execution Time," Information Processing Letters, vol. 16, pp. 221–229, Jun. 1983.
9. P. L'ecuyer and J. Malenfant, "Computing Optimal Checkpointing Strategies for Rollback and Recovery Systems," IEEE Trans. Computers, vol. 37, no. 4, pp. 491–496, Apr. 1988.
10. V. Grassi, L. Donatiello and S. Tucci, "On the Optimal Checkpointing of Critical Tasks and Transaction-Oriented Systems," IEEE Trans. Software Engineering, vol. 18, no. 1, pp. 72–77, Jan. 1992.
11. A. Ziv and J. Bruck, "An On-Line Algorithm for Checkpoint Placement," IEEE Trans. Computers, vol. 46, no. 9, pp. 976–985, Sep. 1997.
12. R. Geist, R. Reynolds and J. Westall, "Selection of a Checkpoint Interval in a Critical-Task Environment," IEEE Trans. Reliability, vol. 37, no. 4, pp. 395–400, Nov. 1988.
13. A. M. Saleh and J. H. Patel, "Transient-Fault Analysis for Retry Techniques," IEEE Trans. Reliability, vol. 37, no. 3, pp. 323–330, Aug. 1988.
14. B. W. Johnson, Design and Analysis of Fault-Tolerant Digital Systems, Addison-Wesley, 1989.
15. J. Sosnowski, "Transient Fault Tolerance in Digital Systems," IEEE Micro, vol. 14, no. 1, pp. 24–35, Feb. 1994.

16. M. Pflanz and H. T. Vierhaus, “Online Check and Recovery Techniques for Dependable Embedded Processors,” IEEE Mirco, vol. 21, no. 5, pp. 24–40, Sep.–Oct. 2001.
17. A. Leon-Garcia, Probability and Random Processes for Electrical Engineering, Addison Wesley, 1994.