Translation from Transactions of IPSJ

Optimal Checkpointing Policies Using the Checkpointing Density

SATOSHI FUKUMOTO*, NAOTO KAIO** and SHUNJI OSAKI***

In a computer system, in particular, in a database system, constitutions and recovery techniques of files play an important role in the operation of the system. This paper discusses checkpointing policies for rollback-recovery which is one of the most general file recovery techniques. When files on a main memory are lost in a failure, we reprocess transactions from the latest checkpoint instead of the starting point of the system operation. Checkpoints are prespecified time points at which the information of the files is collected in a stable secondary storage. It is important to decide an effective checkpointing policy. If we execute checkpointing frequently, we must incur a great cost for the collection of the information, and conversely, if we execute rare checkpointing, we must incur a great cost for the recovery action after the system failure. Thus, checkpoints should be decided by considering the trade-off between the above two costs. In this paper, we discuss the checkpoint time sequence which minimizes the approximate total expected cost per unit time in the steady-state. The analysis shows the derivation of the total cost as the functional of the checkpointing density and the optimal checkpoints. We further present numerical examples assuming a failure-time distribution to be a Weibull distribution, and show how our analytical results are of great use.

1. Introduction

It is important in a computer system to avoid the system down and to restore rapidly the system to the normal operation even if the system down takes place. So far, many fault tolerant techniques for a computer system have been studied [1]. In particular, constitutions and recovery techniques of files on a database system play an important part, since reliability and performance of the system are considerably serious by loss of the files containing much information and/or by introduction of incorrect information. Several techniques to database systems are applied; error correcting methods improving the reliability of data, dual file form having redundancy by duplicated identical files, and so on. On the other hand, if files on a main memory and a secondary storage are lost or hurt by some system failures, several file recovery techniques have been introduced in order to restore the contents of the files to the consistent states just before the failure. In this paper, we consider rollback-recovery which is one of the typical file recovery techniques [2–4].

In a database system, in general, whenever a transaction modifies the files in the main memory, the redundant information about the transaction processing is collected in a stable secondary storage (magnetic tape or disks) which is called a journal (or a log). This information enables us to remove the effects of the transactions from the files, and to reprocess the transactions, where these operations are called UNDO and REDO operations, respectively. On the occasion of a failure, we can restore the files to the states just before the failure, by loading the main memory with a back-up copy of the files which had been taken at a normal operation time in advance, and adding the information of the transaction processing in the journal to the back-up copy. The operation that recovers the files to the states of the previous time when the back-up copy had been taken, is called rollback. If the back-up copy is in the oldest state, the rollback operation must trace back to the start of the system operation, and the file recovery by the journal may be a great deal. We therefore collect the information of the files on the main memory in a stable secondary storage at pre-specified time points before the system failure. We call this operation 'checkpointing', and the time points 'checkpoints'. Checkpoints must be decided by considering the total system performance, cost, and so on. If we execute checkpointing frequently, the total time and cost for checkpointing are large, while those for rollback-recovery are small. Conversely, if we execute rare checkpointing, the time and cost for rollback-recovery are large, while those for checkpointing are small.

This is a translation of the paper that appeared originally in Japanese in Transactions of IPSJ, Vol. 31, No. 6 (1990), pp. 887–893.

*Graduate School of Engineering, Hiroshima University, Higashi-Hiroshima-shi, 724, Japan.
**Department of Management Science, Hiroshima Shudo University, Hiroshima-shi, 731-31, Japan.
***Cluster II (Electrical Engineering) Faculty of Engineering, Hiroshima University, Higashi-Hiroshima-shi, 724, Japan.

Problems of deciding checkpoints have been discussed under the assumption that system failures occur as a Poisson process [5–8]. Such earlier contributions yield constant checkpoint intervals which maximize the system availability, in other word, minimize the total time for rollback-recovery and checkpointing. The problem has been also discussed by an algorithm which optimizes an execution of a task sequence containing checkpoints [9].

In this paper, we propose a new model in which one cycle is defined as the interval from the start of the system operation to the restart on the recovery completion after a failure. Checkpointing policies are discussed considering the trade-off between the cost for checkpointing and the cost for rollback-recovery. Estimating these overhead costs is basically equivalent to estimating the above total loss time, since the cost for the operations is proportional to the time for the operations in general [5]. However, it is of great convenience for our analysis to estimate the cost without treating the times for checkpointing and rollback-recovery. We therefore introduce the approximate expected cost per unit time in the steady-state as the criterion of our model. Assuming that a time point of the system restart is a regeneration point, the approximate expected cost per unit time for one cycle can be substituted for the above cost in the steady-state. We derive the cost for one cycle from a checkpointing cost function of the checkpoint interval and a rollback-recovery cost function of the interval from the latest checkpoint and the failure.

In our model, the optimal checkpoints are derived as a time sequence in which the checkpoint interval changes with time, since we assume that a failure-time distribution is an arbitrary failure-time distribution, that is, a failure rate of the system changes with time. Another model has been presented in the reference [11], which similarly yields the optimal checkpoint sequence under the assumption of an arbitrary failure-time distribution. Taking checkpoints on condition that the failure probability between successive checkpoints are always constant p, the model determines p which minimizes the total loss time.

In the analysis, we derive the approximate expected cost in the steady-state as the functional of the checkpointing density; the approximate number of checkpoints per unit time, introducing cost functions and a failure-time distribution. The optimal checkpointing density is obtained by minimizing the cost. The above cost and the optimal checkpointing density are replaced by new ones introducing concrete cost functions. Finally, we show numerical examples for our analyses under the assumption that the failure-time distribution is a Weibull distribution.

2. Model and Assumptions

1. The system operation is started at time 0 ($=t_0$). The planning horizon is infinite.

2. The kth checkpointing is instantaneously executed at the checkpoint $t_k (k=1, 2, 3, \cdots)$. A system failure is never induced by the checkpointing.

3. The detection of the system failure and rollback-recovery are executed instantaneously. The recovery action is always complete, and the system is restarted immediately.

4. One cycle is defined as the interval from the start of the system operation to the restart on the recovery completion after the failure.

5. We introduce the following quantities:
   - $F(t)$, $\bar{F}(t)$, $f(t)$, $r(t)$, $E[T]$: The cumulative distribution function, the reliability function, the probability density function, the failure rate, the mean of the failure time for the system, respectively. Note that $\bar{F}(t)=1-F(t)$ and $r(t)=f(t)/\bar{F}(t)$.
   - $n(t)$: The checkpointing density at time t; a smooth function which denotes the number of checkpoints per unit time at time t.
   - t_k: The time at which the system executes checkpointing for each cycle; $k=1, 2, 3, \cdots$.
   - $\{t_1^*, t_2^*, t_3^*, \cdots\}$: The optimal checkpoint sequence minimizing the approximate expected cost per unit time in the steady-state.
   - $C(n(t))$: The approximate expected cost per unit time in the steady-state.
   - $L_c(t_k-t_{k-1})$: The cost for checkpointing at checkpoint $t_k; k=1, 2, 3, \cdots$.
   - $L_r(t-t_N)$: The cost for rollback-recovery in case the system failure occurs at time t and the latest checkpoint is t_N.

3. Analysis

Let us derive the optimal checkpoint sequence which minimizes the approximate total expected cost per unit time in the steady-state from the above assumptions.

First, the approximate expected cost for checkpointing per one cycle, $S_1(n(t))$, is given by

$$S_1(n(t))=\int_0^\infty \int_0^t L_c(n(\tau)^{-1})n(\tau)d\tau dF(t)$$
$$=\int_0^\infty L_c(n(t)^{-1})n(t)\bar{F}(t)dt. \qquad (1)$$

We next derive the approximate expected cost for rollback-recovery per one cycle $S_2(n(t))$. If $t-t_N=\tau$, the cost $L_r(\tau)$ can be approximated by applying Taylor's expansion:

$$L_r(\tau)=L_r\left(\frac{\tau}{2}\right)+\frac{\tau}{2}L_r'\left(\frac{\tau}{2}\right)+\cdots$$
$$\simeq L_r\left(\frac{\tau}{2}\right)+\frac{\tau}{2}L_r'\left(\frac{\tau}{2}\right).$$

In case the system failure occurs at time t, the cost for rollback-recovery is approximately given by

$$\frac{\int_0^{n(t)^{-1}} L_r(\tau)d\tau}{n(t)^{-1}}$$
$$=n(t)\int_0^{n(t)^{-1}}\left[L_r\left(\frac{\tau}{2}\right)+\frac{\tau}{2}L_r'\left(\frac{\tau}{2}\right)\right]d\tau$$
$$=n(t)\int_0^{n(t)^{-1}} L_r\left(\frac{\tau}{2}\right)d\tau+n(t)\frac{\tau}{2}\cdot 2L_r\left(\frac{\tau}{2}\right)\Big|_0^{n(t)^{-1}}$$
$$-n(t)\int_0^{n(t)^{-1}} L_r\left(\frac{\tau}{2}\right)d\tau$$
$$=L_r\left(\frac{1}{2}n(t)^{-1}\right).$$

Thus, we can obtain $S_2(n(t))$:

$$S_2(n(t))=\int_0^{\infty} L_r\left(\frac{1}{2}n(t)^{-1}\right)dF(t). \tag{2}$$

From equations (1) and (2), we have

$$C(n(t))=\frac{S_1(n(t))+S_2(n(t))}{E[T]}=\frac{\int_0^{\infty} L_c(n(t)^{-1})n(t)\bar{F}(t)dt+\int_0^{\infty} L_r\left(\frac{1}{2}n(t)^{-1}\right)dF(t)}{E[T]}, \tag{3}$$

where the approximate expected cost per unit time in the steady-state coincides with the cost for a cycle [10].

We obtain the checkpointing density $n(t)$ minimizing the functional $C(n(t))$. This is a problem of calculus of variations in which $n(t)$ is an unknown function. Euler's equation implies

$$L_c(n(t)^{-1})-n(t)^{-1}L_c'(n(t)^{-1})$$
$$-\frac{1}{2}n(t)^{-2}L_r'\left(\frac{1}{2}n(t)^{-1}\right)r(t)=0. \tag{4}$$

Applying the concrete cost functions $L_c(x)$ and $L_r(x)$, and solving equation (4) yield the checkpointing density $n(t)$.

In general, if we use the checkpointing density $n(t)$, the checkpoint sequence $\{t_1, t_2, t_3, \cdots\}$ satisfies;

$$k=\int_0^{t_k} n(t)dt, \quad k=1, 2, 3, \cdots. \tag{5}$$

Substituting $n(t)$ obtained above into equation (5) enables us to derive the optimal checkpoint sequence $\{t_1^*, t_2^*, t_3^*, \cdots\}$.

Let us introduce the concrete cost functions to obtain the checkpoint density $n(t)$ based on the above analytical results. We assume that

$$L_c(x)=c_c+k_c a_c\frac{\lambda_r}{\mu_s}x \tag{6}$$

and

$$L_r(x)=c_r+k_r\frac{\lambda_r}{\mu_s}x, \tag{7}$$

respectively, where each parameter is defined as follows:

- λ_r: The arrival rate [4] of an update transaction which is reprocessed in rollback-recovery.
- μ_s: The processing (service) rate [4] for transactions, where $1/\mu_s \leq 1/\lambda_r$, i.e., $\lambda_r/\mu_s \leq 1$.
- a_c: The ratio of the overhead for checkpointing to the overhead for reprocessing of the update transactions which have been processed between two successive checkpoints.
- c_c: The cost attendant on checkpointing.
- K_c: The cost for checkpointing per unit time.
- c_r: The cost attendant on rollback-recovery.
- k_r: The cost for rollback-recovery per unit time.

If $\rho=\frac{\lambda_r}{\mu_s}$, $K_c=k_c a_c\frac{\lambda_r}{\mu_s}=k_c a_c\rho$, and $K_r=k_r\frac{\lambda_r}{\mu_s}=k_r\rho$, the cost functions are given by

$$L_c(x)=c_c+K_c x \tag{8}$$

and

$$L_r(x)=c_r+K_r x. \tag{9}$$

From equation (3), the approximate expected cost per unit time in the steady-state is given by

$$C(n(t))=\int_0^{\infty}(c_c+K_c n(t)^{-1})n(t)\bar{F}(t)dt+\int_0^{\infty}\left(c_r+\frac{K_r}{2}n(t)^{-1}\right)dF(t)/E[T] \tag{10}$$

$$=\frac{\int_0^{\infty}(c_c n(t)+K_c)\bar{F}(t)dt+\int_0^{\infty}\left(c_r+\frac{K_r}{2}n(t)^{-1}\right)f(t)dt}{E[T]}. \tag{11}$$

We further obtain Euler's equation from equation (4):

$$c_c + K_c n(t)^{-1} - n(t)^{-1}K_c - \frac{1}{2}n(t)^{-2}K_r r(t) = 0,$$

that is,

$$c_c - \frac{1}{2}n(t)^{-2}K_r r(t) = 0. \qquad (12)$$

Solving equation (12) with respect to $n(t)$ yields:

$$n(t) = \sqrt{\frac{K_r}{2c_c}}\, r(t). \qquad (13)$$

Introducing a new constant

$$K_{rc} = \frac{K_r}{2c_c} = \frac{k_r}{2c_c}\rho = \frac{k_r}{2c_c}\cdot\frac{\lambda_r}{\mu_s}, \qquad (14)$$

we have the following $n(t)$:

$$n(t) = \sqrt{K_{rc} r(t)} = [K_{rc} r(t)]^{1/2}. \qquad (15)$$

4. Numerical Examples

Let us show numerical examples by assuming the failure time distribution to be a Weibull distribution:

$$F(t) = 1 - e^{-(\lambda t)^m}, \quad (\lambda > 0, \quad m > 0)$$

where λ and m are called the scale and shape parameters, respectively. The Weibull distribution is able to give a reasonable description of several failure modes by varying the parameters, and is applied to describe a fatigue failure, an electronic element failure, and so on. It is sufficiently useful for our model in which the failure rate changes with time. We have $\bar{F}(t) = e^{-(\lambda t)^m}$, $f(t) = m\lambda^m t^{m-1} e^{-(\lambda t)^m}$, $r(t) = m\lambda^m t^{m-1}$, $E[T] = \left(\frac{1}{\lambda}\right)\Gamma\left(1 + \frac{1}{m}\right)$, where $\Gamma(k) = \int_0^\infty e^{-x}x^{k-1}dx$ (gamma function).

From equation (15), the checkpointing density is given by:

$$n(t) = [K_{rc} m\lambda^m t^{m-1}]^{1/2}. \qquad (16)$$

We can see from the failure rate properties of the Weibull distribution that the checkpoint interval increases with time for $0 < m < 1$ and decreases for $1 < m$. In case of $m = 1$, $F(t)$ is an exponential distribution and the checkpoint interval is always constant.

Substituting $n(t)$ from equation (16) into equation (15) yields the approximate expected cost per unit time in the steady-state:

$$C(n(t)) = \frac{2c_c \int_0^\infty n(t)\bar{F}(t)dt + c_r}{E[T]} + K_c. \qquad (17)$$

Table 1 shows the optimal checkpoint sequence $\{t_1^*, t_2^*, t_3^*, \cdots\}$ obtained from equations (5) and (16), and the checkpoint sequence $\{t_1^{**}, t_2^{**}, t_3^{**}, \cdots\}$ assuming

Table 1 Optimal checkpointing time sequence by the checkpointing density vs. the sequence by the constant time interval. ($F(t) = 1 - \exp[-(\lambda t)^m]$, $m=0.5$, $E[T]=500$, $c_c=10$, $c_r=10$, $k_c=1$ $k_r=1$, $a_c=0.1$, $1/\lambda_r=5$, $1/\mu_r=2$,)

k	t_k^*	$F(t_k^*)$	t_k^{**}	$F(t_k^{**})$
1	92	0.4557	158	0.5485
2	233	0.6192	316	0.6752
3	400	0.7178	474	0.7478
4	587	0.7840	632	0.7962
5	791	0.8311	791	0.8311
6	1008	0.8658	949	0.8574
7	1238	0.8920	1107	0.8780
⋮	⋮	⋮	⋮	⋮

Table 2 Optimal checkpointing time sequence by the checkpointing density vs. the sequence by the constant time interval. ($F(t) = 1 - \exp[-(\lambda t)^m]$, $m=1$, $E[T]=500$, $c_c=10$, $c_r=10$, $k_c=1$ $k_r=1$, $a_c=0.1$, $1/\lambda_r=5$, $1/\mu_r=2$,)

k	t_k^*	$F(t_k^*)$	t_k^{**}	$F(t_k^{**})$
1	158	0.2711	158	0.2711
2	316	0.4687	316	0.4687
3	474	0.6127	474	0.6127
4	632	0.7177	632	0.7177
5	791	0.7943	791	0.7943
6	949	0.8500	949	0.8500
7	1107	0.8907	1107	0.8907
⋮	⋮	⋮	⋮	⋮

Table 3 Optimal checkpointing time sequence by the checkpointing density vs. the sequence by the constant time interval. ($F(t) = 1 - \exp[-(\lambda t)^m]$, $m=2$, $E[T]=500$, $c_c=10$, $c_r=10$, $k_c=1$ $k_r=1$, $a_c=0.1$, $1/\lambda_r=5$, $1/\mu_r=2$,)

k	t_k^*	$F(t_k^*)$	t_k^{**}	$F(t_k^{**})$
1	262	0.1935	158	0.0755
2	415	0.4183	316	0.2696
3	544	0.6056	474	0.5068
4	659	0.7447	632	0.7154
5	765	0.8409	791	0.8596
6	864	0.9041	949	0.9408
7	957	0.9438	1107	0.9787
⋮	⋮	⋮	⋮	⋮

Table 4 The expected costs and gains ($(C_2 - C_1)/C_2[\%]$) by the checkpointing density vs. by the constant time interval. ($F(t) = 1 - \exp[-(\lambda t)^m]$, $E[T]=500$, $c_c=10$, $c_r=10$, $k_c=1$ $k_r=1$, $a_c=0.1$, $1/\lambda_r=5$, $1/\mu_r=2$,)

	C_1	C_2	$(C_2 - C_1)/C_2[\%]$
$m=0.5$	0.1721	0.1942	11.36
$m=2$	0.1764	0.1867	5.513

that the checkpointing is executed periodically, where $m=0.4$, $E[T]=500$, $c_c=10$, $c_r=10$, $k_c=1$, $k_r=1$, $a_c=0.1$, $1/\lambda_r=5$, and $1/\mu_r=2$. Figure 1 illustrates the relation between the checkpoint sequence and the checkpoint density $n(t)$. Table 2 and Figure 2 show the results for $m=1$, where all parameters are the same as

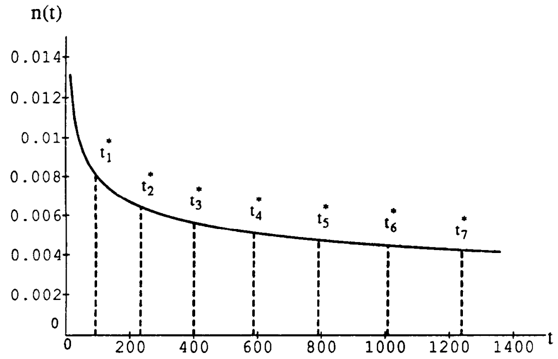

Fig. 1 The illustration for the checkpointing density and checkpointing time sequence. ($F(t)=1-\exp[-(\lambda t)^m]$, $m=0.5$, $E[T]=500$, $c_c=10$, $c_r=10$, $k_c=1$ $k_r=1$, $a_c=0.1$, $1/\lambda_r=5$, $1/\mu_r=2$,)

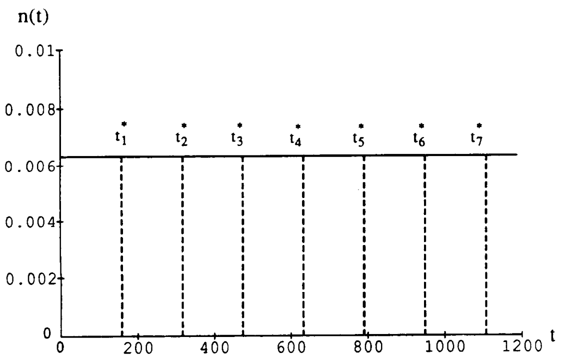

Fig. 2 The illustration for the checkpointing density and checkpointing time sequence. ($F(t)=1-\exp[-(\lambda t)^m]$, $m=1$, $E[T]=500$, $c_c=10$, $c_r=10$, $k_c=1$ $k_r=1$, $a_c=0.1$, $1/\lambda_r=5$, $1/\mu_r=2$,)

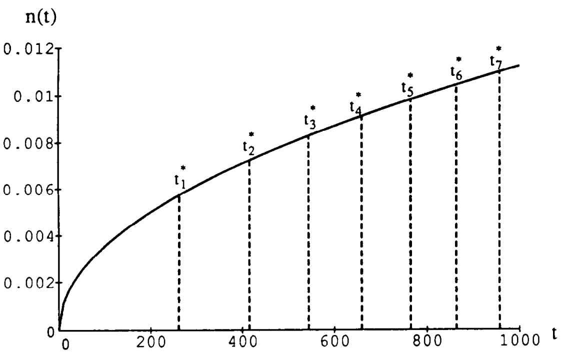

Fig. 3 The illustration for the checkpointing density and checkpointing time sequence. ($F(t)=1-\exp[-(\lambda t)^m]$, $m=2$, $E[T]=500$, $c_c=10$, $c_r=10$, $k_c=1$ $k_r=1$, $a_c=0.1$, $1/\lambda_r=5$, $1/\mu_r=2$,)

in Table 1 except the shape parameter m. Table 3 and Figure 3 similarly show the results for $m=2$. In case of $m=0.5$, the checkpoint intervals are short at the early stage of the system operation, and increase with time. Conversely, in case of $m=2$, checkpoint intervals are long at the early stage of the system operation, and decrease with time.

We next discuss comparisons between the approximate expected cost by the optimal checkpoint sequence and the one by the periodical checkpoints, for $m=0.5$ and $m=2$. Let C_1 and C_2 denote the cost by the optimal checkpoint sequence and the cost by the periodical checkpoints, respectively. Table 4 shows the gain of C_1 to C_2, $((C_2-C_1)/C_2)\times 100[\%]$, for $m=0.5$ and $m=2$. We can see from these results that the checkpointing policy by the optimal checkpoint sequence is better than the other in either case.

5. Conclusion

This paper has discussed checkpointing policies for rollback-recovery which is a file recovery technique in a computer system. We have shown the derivation of the checkpoint sequence which minimizes our criterion; the approximate expected cost per unit time in the steady-state. The checkpointing density which denotes the approximate checkpointing rate has been introduced by assuming that the checkpoint interval should change with the failure rate of the system. One cycle has been defined as the interval from the start of the system operation to the restart on the recovery completion after the failure. The approximate expected cost per unit time in the steady-state has been derived as the functional of the checkpoint density from the costs for checkpointing and rollback-recovery per one cycle. We have obtained the checkpointing density which minimizes the above cost. We have further obtained the above cost and the optimal checkpointing density by assuming concrete cost functions. Finally, numerical examples for the results obtained have been shown, in case the distribution function of the failure time is assumed to be a Weibull distribution.

Our results derived by using the checkpointing density are analytical results. Thus, applying a failure-time distribution and cost functions, we can relatively easily calculate the optimal checkpoint sequence. In particular, our analytical results are of great use for various kinds of distributions, since we have assumed an arbitrary failure-time distribution.

Rollback-recovery with checkpoints is in general of great use for a file recovery technique in a database system. A system failure does not always occur with a constant failure rate if we consider the times in the early stage of practical system operations and so on. In such situations, our results yield an effective checkpoint policy, since the policy by the optimal checkpoint sequence is better than the one by the periodical checkpoints as discussed by the numerical examples above.

References

1. MUKAIDONO, M. (ed.) *Introduction to Highly Reliable Techniques for Computer Systems*, Nippon Kikaku Kyokai (1988) (In Japanese).
2. JASPER, D. P. A Discussion of Checkpoint/Restart, *Software Age* (1969), 9–14.
3. HAERDER, T. and REUTER, A. Principles of Transaction-Oriented Database Recovery, *Comput. Surv.*, **15**, 4 (1983), 287–317.
4. CHANDY, K. M. and RAMAMOORTHY, C. V. Rollback and Recovery Strategies for Computer Programs, *IEEE Trans. Comput.*, **C-21**, 6 (1972), 546–556.
5. YOUNG, J. W. A First Order Approximation to the Optimum Checkpoint Interval, *Comm. ACM*, **17**, 9 (1974), 530–531.
6. CHANDY, K. M., BROWNE, J. C., DISSLY, C. W. and UHRIG, W. R. Analytic Models for Rollback and Recovery Strategies in Data Base Systems, *IEEE Trans. Softw. Eng.*, **SE-1**, 1 (1975), 100–110.
7. CHANDY, K. M. A Survey of Analytic Models of Rollback and Recovery Strategies, *Computer*, **8**, 5 (1975), 40–47.
8. GELENBE, E. On the Optimum Checkpoint Interval, *J. ACM*, **26**, 2 (1979), 259–270.
9. TOUEG, S. and BABAOĞLU, Ö. On the Optimum Checkpoint Selection Problem, *SIAM J. Comput.*, **13**, 3 (1984), 630–649.
10. ROSS, S. M. *Applied Probability Models with Optimization Applications*, Holden-Day, San Francisco (1970).
11. KAIO, N. and OSAKI, S. A Note on Optimum Checkpointing Policies, *Microelectron. Reliab.*, **25**, 3 (1985), 451–453.