Computers Math. Applic. Vol. 24, No. 1/2, pp. 63–70, 1992
Printed in Great Britain.

0097-4943/92 $5.00 + 0.00

A STUDY OF CHECKPOINT GENERATIONS FOR A DATABASE RECOVERY MECHANISM

S. FUKUMOTO
Department of Industrial and Systems Engineering, Hiroshima University
Higashi-Hiroshima 724, Japan
N. KAIO
Department of Management Science, Hiroshima Shudo University
Hiroshima 731-31, Japan
S. OSAKI
Department of Industrial and Systems Engineering, Hiroshima University
Higashi-Hiroshima 724, Japan

Abstract—This paper discusses checkpoint generations for a database recovery mechanism. The density of checkpoint generations is analytically derived from minimizing the expected total overhead to completion of a phase, and this density yields the optimum sequence of checkpoint generations measured in unit of update pages. We further present the numerical examples for the results obtained and show that the sequence gives effective checkpoint generations.

1. INTRODUCTION

Fault tolerant techniques play an important role in the operation of a computer system with high reliability. In particular, recovery mechanisms are indispensable for reconstructing the states of the computation after the system failure. A database system is a typical example of what seriously needs such recovery mechanisms. There are excellent surveys for database recovery in [1,2]. This paper discusses checkpoint generations for a recovery mechanism on large applications of database systems.

When a system failure makes update information in the database buffer lost, the recovery action consists of two operations. One is "UNDO operation" which rolls back the effects of all incomplete transactions from the database, and the other is "REDO operation" which reflects the results of all complete transactions in the database (see [2]). In general, we execute REDO operation from the latest checkpoint instead of the starting point of the system operation. Generating a checkpoint implies that the update information in the buffer is collected in a stable secondary storage. It is important to decide the effective checkpoint generations. If we generate checkpoints frequently, we must incur large overhead for checkpoint generations, and conversely, if we generate few checkpoints, we must incur large overhead for recovery actions after the system failures. We should, therefore, decide checkpoint generations considering the trade-off between the two overheads above.

Several studies of deciding checkpoint generations have been discussed, which are the components of general recovery mechanisms including a database recovery. Young [3] derived the optimum checkpoint interval for the computation restart after the system failure. Chandy *et al.* [4] and Gelenbe [5] discussed evaluation models for database recovery and the generalized forms of the optimum checkpoint interval maximizing the system availability or the overhead during the normal operation. In these previous works, the failure rate of the system is assumed to be constant. The authors proposed a model for evaluating the database recovery action in the case where the failure rate of the system changes with time [6]. While these efforts yield the optimum checkpoint interval measured in units of time, some models deal with the checkpoint interval measured in another quantity to describe the recovery mechanisms more reasonably. Reuter [7]

considered the models to evaluate the transaction throughput as a performance measure for the database recovery mechanisms of the taxonomy in [2], where the checkpoint interval is measured in units of block transfers. Toueg and Babaoğlu [8] derived an algorithm which minimizes expected execution times of tasks placing checkpoints between two consecutive tasks with very general assumptions. Koren *et al.* [9] also discussed the model which minimizes the average time per instruction as a function of the number of instruction retries and the checkpoint interval measured in the number of the instructions, assuming the constant failure rate.

In this paper, we propose a new model to determine the checkpoint generations for the database recovery. We consider that the transaction arrival rate and the failure rate of the system vary with time. The algorithm above derived by Toueg and Babaoğlu [8] seems to give a reasonable description of such situations. However, the dynamic programming algorithm, which yields the optimum sequence of checkpoints, is not suitable for large applications since the number of the transactions is expected to reach a great deal between the successive checkpoint generations. One of the primary interests in our model is that the transaction arrival rate, i.e., the load of the system changes with time in a shape of a cycle (e.g., a day) as an illustration of Figure 1. In this case, we can see that the constant checkpoint interval measured in units of time is not pertinent, since the failure rate of the system and the overhead for the recovery action obviously seems to vary with the load of the system. Taking account of these situations the third model exhibited by Chandy *et al.* [4] yields the problem of finding the shortest route of the graph whose nodes correspond to the beginning of intervals divided into from a cycle.

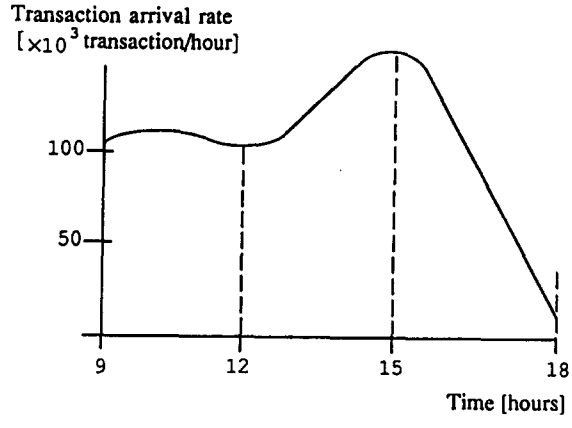

Figure 1. A sample function of transaction arrival rate for a cycle.

We derive an analytically efficient result by means of a simpler model. Occurrence of the failure and checkpoint generations are estimated by units of update pages in the database buffer instead of time. We further regard the cumulative update of pages as a continuous quantity. Assume that the failure rate of the system (as a function of the cumulative update of pages) is dependent on the transaction arrival rate at which the corresponding page is updated, and the failure mode of a cycle is described as consisting of phases, e.g., as shown in Figure 2. The optimum checkpoint generations are derived as the sequence measured in the cumulative update minimizing the expected total overhead to completion of a phase, where the checkpoint interval changes with the failure rate of the system.

In the following section, we define our new model introducing a density of checkpoint generations and several assumptions. Section 3 discusses the analysis of the model. The expected total overhead to completion of a phase is derived. We obtain the density of checkpoint generations minimizing the total overhead, which yields the optimum sequence of checkpoint generations. Moreover, the above total overhead and density are replaced by new forms assuming concrete overhead functions. We next show the results in case where the cumulative update to the system failure obeys a Weibull distribution. Section 4 gives numerical examples for our analyses under the assumption that the failure rate is described as the shape of phases in Figure 2.

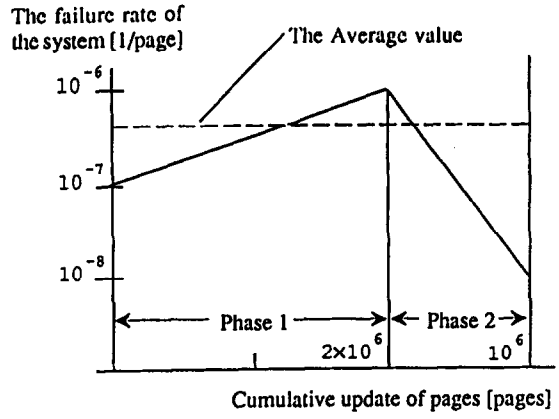

Figure 2. A shape of the failure rate for a cycle.

2. MODEL AND ASSUMPTIONS

In our model, all the pages modified by complete transactions remaining in the buffer are reflected to the secondary storage by generating a checkpoint. While the cumulative update of pages in the buffer is a discrete quantity obviously, we can regard it as continuous, since we consider a great number of update pages such as thousands or tens of thousands of pages. A phase we deal with completes when the cumulative update reaches to N, with the s^{th} checkpoint generation. Let $\{n_1, n_2, \ldots, n_{s-1}, n_s(=N)\}$ be the sequence of checkpoint generations, where each checkpoint is generated sequentially up to the cumulative update from the beginning of a phase to $n_k (k = 1, 2, \ldots, s)$. Note that these checkpoint generations are executed independently of real time lost by recovery actions, since the generations are managed by unit of update pages instead of time.

We introduce a density of checkpoint generations, $g(n)$, when the cumulative update is n, which is a smooth function and denotes the number of checkpoint generations per unit update. If we use the density $g(n)$, the above sequence satisfies:

$$k = \int_0^{n_k} g(n)\, dn, \qquad (k = 1, 2, \ldots, s-1). \tag{1}$$

We assume that the cumulative update of pages to the system failure X obeys the cumulative distribution function $F(n)$. If the reliability function $\bar{F}(n) = 1 - F(n)$ and the probability density $f(n) = \frac{dF(n)}{dn}$, the failure rate of the system is defined by $\gamma(n) = f(n)/\bar{F}(n)$. For all the failures occurred in the checkpoint interval $(n_{k-1}, n_k]$, $(k = 1, 2, \ldots, s; n_0 = 0)$, we make recovery actions from the state of k^{th} checkpoint generation to the consistent states which had been constructed just before those failures. We consider that checkpoint generations and recovery actions never cause the system failure and never change the failure rate of the system.

The expected total overhead to completion of a phase, $L(N, g(n))$, consists of the overhead for checkpoint generations to completion of a phase and the expected overhead for recovery actions to completion of a phase. In order to derive these overheads, we introduce the overhead for the k^{th} checkpoint generation, $H_c(n_k - n_{k-1})$, and the overhead for a recovery action, $H_r(n - n_l)$, in case $X = n$ and the latest checkpoint generation is the l^{th} one.

3. ANALYSIS

3.1. General Analysis

Let us derive the optimum sequence $\{n_1^*, n_2^*, \ldots, n_{s-1}^*, n_s\}$ which minimizes the expected total overhead to completion of a phase from the assumptions above.

First, the overhead for checkpoint generations to completion of a phase is obtained as follows by using the density of checkpoint generations:

$$\sum_{k=1}^{s} H_c(n_k - n_{k-1}) = \int_0^N H_c(g(n)^{-1})\, g(n)\, dn. \tag{2}$$

We next derive the expected overhead for recovery actions to completion of a phase. If $X = n$, the overhead for recovery actions between two successive checkpoint generations is approximately given by

$$\begin{aligned}\int_{n_{k-1}}^{n_k} H_r(n - n_{k-1})\,\gamma(n)\,dn &\simeq \int_{n_{k-1}}^{n_k} H_r\left(\frac{n_k - n_{k-1}}{2}\right)\gamma(n)\,dn \\ &= H_r\left(\frac{n_k - n_{k-1}}{2}\right)\int_{n_{k-1}}^{n_k}\gamma(n)\,dn, \qquad (k = 1, 2, \ldots, s),\end{aligned} \tag{3}$$

where we consider the overhead for a recovery action to be equal to the overhead after a system failure in the middle of the checkpoint interval, in average, similarly to [7]. This approximation can be expected to be a good one, since we are estimating the mean value of the total overhead. Thus, we can obtain the expected overhead for recovery actions to completion of a phase:

$$\begin{aligned}\sum_{k=1}^{s}\int_{n_{k-1}}^{n_k} H_r(n - n_{k-1})\,\gamma(n)\,dn &\simeq \sum_{k=1}^{s} H_r\left(\frac{n_k - n_{k-1}}{2}\right)\int_{n_{k-1}}^{n_k}\gamma(n)\,dn \\ &= \int_0^N H_r\left(\frac{1}{2}\,g(n)^{-1}\right)\gamma(n)\,dn.\end{aligned} \tag{4}$$

From Equations (2) and (4), we have the expected total overhead to completion of a phase:

$$L(N, g(n)) = \int_0^N \left[H_c(g(n)^{-1})\,g(n) + H_r\left(\frac{1}{2}\,g(n)^{-1}\right)\gamma(n)\right]dn. \tag{5}$$

We obtain the density of checkpoint generations, $g(n)$, minimizing the functional $L(N, g(n))$. This is a problem of calculus of variations in which $g(n)$ is the unknown function. Euler's equation implies

$$H_c(g(n)^{-1}) - g(n)^{-1}H_c'(g(n)^{-1}) - \frac{1}{2}\,g(n)^{-2}H_r'\left(\frac{1}{2}\,g(n)^{-1}\right)\gamma(n) = 0. \tag{6}$$

Applying concrete overhead functions $H_c(\cdot)$ and $H_r(\cdot)$, and solving Equation (6) yield the density $g(n)$. Substituting $g(n)$ into Equation (1) enables us to derive the optimum sequence $\{n_1^*, n_2^*, \ldots, n_{s-1}^*, n_s\}$.

3.2. Overhead Functions

Let us introduce concrete overhead functions to obtain the density $g(n)$ based on the analytical results above. In large applications of database systems, we can assume the overhead function for a checkpoint generation to be the simplest form:

$$H_c(x) = h_c, \tag{7}$$

that is, the overhead for a checkpoint generation is always constant and independent of the checkpoint interval (see [2,6]). We further assume the overhead function for a recovery action:

$$H_r(x) = h_r x + h_u, \tag{8}$$

where h_u is the constant overhead for UNDO operation and h_r is the overhead for REDO operation per unit update of pages corresponding to the forms of [4–6]. From Equation (5), the expected total overhead to completion of a phase is given by

$$L(N, g(n)) = \int_0^N \left[h_c g(n) + \left(\frac{h_r}{2g(n)} + h_u\right)\gamma(n)\right]dn. \tag{9}$$

We further obtain Euler's equation from Equation (6):

$$h_c - \frac{h_r}{2}\,g(n)^{-2}\gamma(n) = 0. \tag{10}$$

Solving Equation (10) with respect to $g(n)$ yields:

$$g(n) = \sqrt{\frac{h_r}{2h_c}\gamma(n)}. \tag{11}$$

3.3. A Case of Weibull Distribution

We next discuss a case where the cumulative update of pages to the system failure obeys the Weibull distribution:

$$F(n) = 1 - e^{-(\eta n)^m}, \qquad (\eta > 0, m > 0). \tag{12}$$

The Weibull distribution is able to give a reasonable description of several failure modes, in which the failure rates change with the time variables, by varying the parameters. The parameters η and m are called the scale and shape parameters, respectively. We have $\bar{F}(n) = e^{-(\eta n)^m}$, $f(n) = m\eta^m n^{m-1} e^{-(\eta n)^m}$ and $\gamma(n) = m\eta^m n^{m-1}$.

From Equation (11), the density of checkpoint generations is given by

$$g(n) = \sqrt{\frac{h_r m\eta^m n^{m-1}}{2h_c}}. \tag{13}$$

Moreover, substituting $g(n)$ from Equation (13) into Equation (9) yields the expected total overhead to completion of a phase:

$$L(N, g(n)) = \frac{2\sqrt{2h_c h_r m\eta^m}}{m+1} N^{\frac{m+1}{2}} + h_u(\eta N)^m. \tag{14}$$

From Equations (1) and (13), we can explicitly obtain the optimum sequence as follows:

$$n_k^* = (m+1)^{\frac{2}{m+1}} \left(\frac{h_c}{2h_r m\eta^m}\right)^{\frac{1}{m+1}} k^{\frac{2}{m+1}}, \qquad (k = 1, 2, \ldots, s-1). \tag{15}$$

We can see that the interval between checkpoint generations increases with the cumulative update for $0 < m < 1$ and decreases for $1 < m$. In particular, in case of $m = 1$, $F(n)$ is an exponential distribution. We have the constant intervals between checkpoint generations:

$$n_k^* - n_{k-1}^* = \sqrt{\frac{2h_c}{h_r \eta}}, \qquad (k = 1, 2, \ldots, s-1), \tag{16}$$

which coincides with the formula obtained by Young [3] when we regard n as the time variable and $h_r = 1$.

4. NUMERICAL EXAMPLES

Let us numerically compute the sequence of checkpoint generations by assuming the phases as shown in Figure 2. If the failure rate $\gamma(n)$ is described as the function of the first degree, i.e., $\gamma(n) = vn + w$, the optimum sequence of checkpoint generations is given by

$$n_k^* = \frac{1}{v}\left[\frac{3v}{2}\sqrt{\frac{2h_c}{h_r}} \cdot k + w^{\frac{3}{2}}\right]^{\frac{2}{3}} - \frac{w}{v}, \qquad (k = 1, 2, \ldots, s-1), \tag{17}$$

from Equations (1) and (11). We further have the expected total overhead from Equation (9) as follows:

$$L(N, g(n)) = \frac{2\sqrt{2h_c h_r}}{3v}\left\{(vN + w)^{\frac{3}{2}} - w^{\frac{3}{2}}\right\} + h_u\left(\frac{v}{2}N^2 + wN\right). \tag{18}$$

Let $\{n_1', n_2', \ldots, n_{s-1}', n_s\}$ be the sequence of checkpoint generations in case where the constant failure rate η_c, that is the average value of the failure rate of Figure 2, is used instead of the

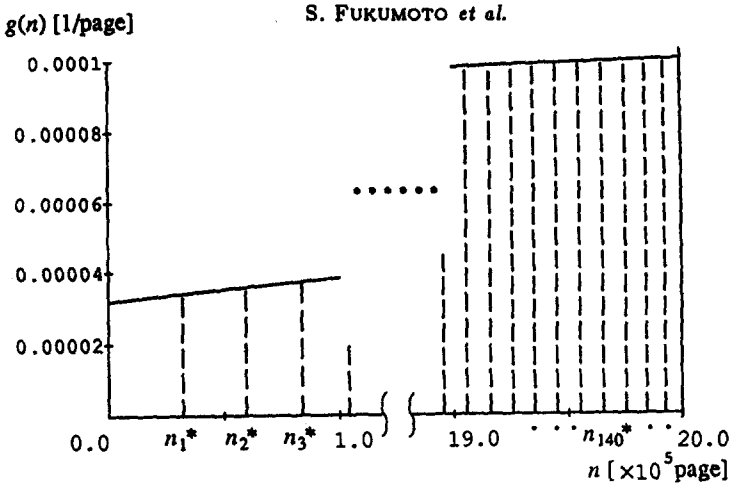

Figure 3. The illustration for the density and the sequence of checkpoint generations for the phase 1.

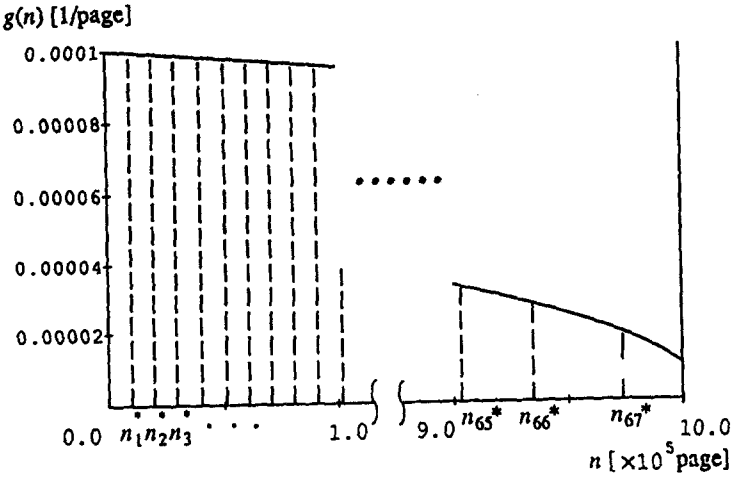

Figure 4. The illustration for the density and the sequence of checkpoint generations for the phase 2.

failure rate of the phase 1 or the phase 2 to obtain the density $g(n)$. Table 1 shows the optimum sequence $\{n_1^*, n_2^*, \ldots, n_{s-1}^*, n_s\}$ for the phase 1, and the sequence $\{n_1', n_2', \ldots, n_{s-1}', n_s\}$, where $v = (10^{-6} - 10^{-7})/(2 \times 10^6)$, $w = 10^{-7}$, $N = 2 \times 10^6$, $h_c = 5$[sec], $h_r = 0.1$[sec] and $h_u = 5$[sec]. Figure 3 illustrates the relation between the sequence and the density of checkpoint generations $g(n)$. Table 2 and Figure 4 similarly show the results for the phase 2, where $v = (10^{-8} - 10^{-6})/10^6$, $w = 10^{-6}$, $N = 10^6$ and the other parameters are the same as in Table 1. Note that the checkpoint interval is decreasing with the cumulative update in case of the phase 1, since the failure rate is increasing. Conversely, the interval is increasing with the cumulative update in case of the phase 2, since the failure rate is decreasing.

We next discuss comparisons between the expected total overhead by the optimum sequence $\{n_1^*, n_2^*, \ldots, n_{s-1}^*, n_s\}$ and the one by the sequence $\{n_1', n_2', \ldots, n_{s-1}', n_s\}$ assuming the failure rate is described as the phase 1 or the phase 2. Let L_p denote the expected total overhead by the optimum sequence which is obtained by Equation (18). Furthermore, let L_c denote the expected total overhead by the sequence $\{n_1', n_2', \ldots, n_{s-1}', n_s\}$. We can obtain L_c from Equation (5) in which $g(n)$ is derived by the constant failure rate above although $\gamma(n)$ is the failure rate of the phase 1 or the phase 2. Table 3 shows the gain of L_p to L_c, $((L_c - L_p)/L_c) \times 100$[%], for the phase 1 and the phase 2, where all parameters are the same as in Tables 1 and 2, and the average value of the failure rate is calculated as $\eta_c = 5.35 \times 10^{-7}$. It is evident that checkpoint generations by the optimum sequence is more effective than the other in either case. This fact implies that the sequence of checkpoint generations, varying its interval with the failure rate of the system, gives a reasonable strategy of the database recovery mechanism.

Table 1. The sequences of checkpoint generations for the phase 1.

k	n_k^* [$\times 10^4$ pages]	n_k'
1	3.05	1.36
2	5.94	2.73
3	8.68	4.10
4	11.31	5.46
.	.	.
.	.	.
.	.	.
140	196.52	191.40
141	197.52	192.77
142	198.53	194.13
143	199.53	195.50
144	200.00	200.00

Table 2. The sequences of checkpoint generations for the phase 2.

k	n_k^* [$\times 10^4$ pages]	n_k'
1	1.00	1.36
2	2.01	2.73
3	3.02	4.10
4	4.04	5.46
.	.	.
.	.	.
.	.	.
64	87.37	87.49
65	90.25	88.86
66	93.59	90.23
67	98.03	91.60
68	100.00	100.00

Table 3. The expected total overheads to completion of phases.

	L_p [sec]	L_c [sec]	$((L_c - L_p)/L_c) \times 100$
Phase 1	1440	1488	3.3
Phase 2	675	713	5.6

5. CONCLUSION

In this paper, we have discussed checkpoint generations for a database recovery mechanism. The expected total overhead to completion of a phase has been presented. The density of checkpoint generations has been analytically derived minimizing the total overhead, which yields the optimum sequence of checkpoint generations measured in unit of update pages. Finally, numerical examples for the results have been given in case where the failure rate of a phase is described as a linear shape.

The results presented in this paper are the analytical ones. Applying the appropriate failure rate and the parameters enable us to calculate the optimum sequence relatively easily. We can see that the sequence obtained is of great use for various kinds of failure modes and gives reasonable strategy for checkpoint generations as discussed by the numerical examples above.

REFERENCES

1. J.M. Verhofstadt, Recovery techniques for database systems, *ACM Comput. Surv.* **10**, 167–196 (1978).
2. T. Haerder and A. Reuter, Principles of transaction-oriented database recovery, *ACM Comput. Surv.* **15**, 287–317 (1983).
3. J.W. Young, A first order approximation to the optimum checkpoint interval, *Comm. ACM* **17**, 530–531 (1974).
4. 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. Softw. Eng.* **SE-1**, 100–110 (1975).
5. E. Gelenbe, On the optimum checkpoint interval, *J. ACM* **26**, 259–270 (1979).
6. S. Fukumoto, N. Kaio and S. Osaki, Evaluation for a database recovery action with periodical checkpoint generations, *Trans. IEICE of Japan* **E-74**, 2076–2082 (1991).
7. A. Reuter, Performance analysis of recovery techniques, *ACM TODS* **9**, 526–559 (1984).
8. S. Toueg and Ö. Babaoğlu, On the optimum checkpoint selection problem, *SIAM J. Comput.* **13**, 630–649 (1984).
9. I. Koren, Z. Koren and S.Y.H. Su, Analysis of a class of recovery procedures, *IEEE Trans. Comput.* **C-35**, 703–712 (1986).