RAIRO Operations Research
RAIRO Oper. Res. **36** (2002) 227-235
DOI: 10.1051/ro:2003004

OPTIMAL POLICIES FOR A DATABASE SYSTEM WITH TWO BACKUP SCHEMES

Cunhua Qian[1], Yu Pan[1] and Toshio Nakagawa[2]

Communicated by Naoto Kaio

Abstract. This paper considers two backup schemes for a database system: a database is updated at a nonhomogeneous Poisson process and an amount of updated files accumulates additively. To ensure the safety of data, full backups are performed at time NT or when the total updated files have exceeded a threshold level K, and between them, cumulative backups as one of incremental backups are made at periodic times iT $(i = 1, 2, \cdots, N-1)$. Using the theory of cumulative processes, the expected cost is obtained, and an optimal number N^* of cumulative backup and an optimal level K^* of updated files which minimize it are analytically discussed. It is shown as examples that optimal number and level are numerically computed when two costs of backup schemes are given.

Keywords. Database, full backup, cumulative backup, cumulative process, expected cost.

1. Introduction

A database in a computer system is frequently updated by adding or deleting data files, and is stored in floppy disks or other secondary media. Even high reliable computers might sometimes break down eventually by several errors due to noises, human errors and hardware faults. It would be possible to replace hardware and

Received June, 2002.

[1] College of Management Science and Engineering, Nanjing University of Technology, Nanjing 210009, China; e-mail: qch64317@hotmail.com, panyu@dns.njuct.edu.cn

[2] Department of Marketing and Information System, Aichi Institute of Technology, 1247 Yachigusa, Yakusa-cho, Toyota 470-0392, Japan; e-mail: nakagawa@mkt.aitech.ac.jp

© EDP Sciences 2003

software when they fail, but it would be impossible to do a database. One of important things for using computers is to backup data files regularly, *i.e.* to copy all files in a secondary medium. Fukumoto *et al.* [5] discussed optimal checkpoint generations for a database recovery mechanism. Qian *et al.* [9] considered the backup policy where a database fails according to a probability distribution and derived the optimal time of full backup.

Cumulative damage models in reliability theory, where a system suffers damage due to shocks and fails when the total amount of damage exceeds a failure level K, generate a cumulative process [2]. Some aspects of damage models from reliability viewpoints were discussed by Esary *et al.* [3]. It is of great interest that a system is replaced before failure as preventive maintenance. The replacement policies where a system is replaced before failure at time T [11], at shock N [7], or at damage Z [4, 6] were considered. Nakagawa and Kijima [8] applied the periodic replacement with minimal repair [1] at failure to cumulative damage models and obtained optimal values T^*, N^* and Z^* which minimize the expected cost. Satow *et al.* [10] applied the cumulative damage model to garbage collection policies for a database system.

In this paper, we apply the cumulative damage model to the backup of files for database media failures, by putting *shock* by *update* and *damage* by *updated files*: a database is updated at a nonhomogeneous Poisson process and an amount of updated files accumulates additively. To lessen the overhead of backup processing, cumulative backups with small overhead are adopted between full backups [11]. The mean time to full backup and the expected costs are derived, using the theory of cumulative processes. Further, an optimal number of cumulative backup and an optimal level of updated files which minimize the expected costs are analytically derived. Numerical examples are finally shown when two costs of backup schemes are given.

2. Model description

Backup frequencies of a database would usually depend on the factors such as its size and availability, and sometimes frequency in use and criticality of data. The simplest and most indispensable method to ensure the safety of data would be always to shut down a database, and to make the backup copies of all data, log and control files in other places, and to take them out immediately when some data in the original secondary media are corrupted. This is called *total backup*. But, such backup has to be made while a database is off-line and unavailable to its users, and would take more hours and costs as data become larger.

To overcome these disadvantages, *incremental backup* has been developed because only a small percentage of files changes in most applications between successive backups [9, 11]: incremental backup makes the copies of only files which have changed or are new since a prior backup. The resources required for such backup are proportional to the transactional activities which have taken place in a database, and not to its size. This can shorten backup times and can decrease the required resources, and would be more useful for larger databases. On the other hand, the type of backups, in which the copies of all files are made in a storage area and the attributes of archives while backing up only modified files are updated, is called *full backup*. Incremental backup cannot take the place of full backup, however, it can reduce the frequency of full backup which is required. For failures, a database can recover from these points by log files and restore a consistent state by the last full backup and incremental backups.

Cumulative backup has been well-known as one of incremental backups: it makes all copies of modified files since the last full backup, however, does not update the attributes of archives. The list of files is growing up each day until the next full backup which will clear all attributes of archives. We can restore all data by the last full backup and the last cumulative backup. From the above point of view, we can reduce the frequency of full backups by cumulative backups. The problem is to decide the interval of full backups.

3. Expected cost

Suppose that a database should be operating for an infinite time span. Cumulative backups are performed at periodic time iT $(i = 1, 2, \cdots)$, *e.g.* daily or weekly, and make the copies of only updated files which have changed or are new since the last full backup. But, because the time and resources required for cumulative backups are growing up every time, full backup is performed at iT, when the total updated files have exactly exceeded a threshold level K during the interval $((i-1)T, iT]$, or NT $(i = 1, 2, \cdots, N-1;\ N = 1, 2, \cdots)$, whichever occurs first, and makes the copies of all files. A database returns to an initial state by such full backups.

Taking the above considerations into account, we formulate the following stochastic model of the backup policy for a database system: suppose that a database is updated at a nonhomogeneous Poisson process with an intensity function $\lambda(t)$ and a mean-value function $R(t)$, *i.e.*, $R(t) \equiv \int_0^t \lambda(u)\mathrm{d}u$. Then, the probability that the j-th update occurs exactly during $(0, t]$ is

$$H_j(t) \equiv \frac{[R(t)]^j}{j!}\mathrm{e}^{-R(t)} \quad (j = 0, 1, 2, \cdots), \tag{1}$$

where $R(0) \equiv 0$ and $R(\infty) \equiv \infty$.

Further, let Y_j denote an amount of files, which changes or is new at the j-th update. It is assumed that each Y_j has an identical probability distribution $G(x) \equiv P_r\{Y_j \leq x\}$ $(j = 1, 2, \cdots)$. Then, the total amount of updated files $Z_j \equiv \sum_{i=1}^{j} Y_i$ up to the j-th update where $Z_0 \equiv 0$ has a distribution

$$\Pr\{Z_j \leq x\} \equiv G^{(j)}(x) \quad (j = 0, 1, 2, \cdots), \tag{2}$$

and $G^{(0)}(x) \equiv 1$ for $x \geq 0$, 0 for $x < 0$, where $G^{(j)}(x)$ $(j = 1, 2, \cdots)$ is the j-fold convolution of $G(x)$ with itself. Then, the probability that the total amount of updated files exceeds exactly a threshold level K at j-th update is $G^{(j-1)}(K) - G^{(j)}(K)$. Let $Z(t)$ be the total amount of updated files at time t. Then, the distribution of $Z(t)$ is [3]

$$\Pr\{Z(t) \le x\} = \sum_{j=0}^{\infty} H_j(t) G^{(j)}(x). \tag{3}$$

Since the probability that the total amount of updated files does not exceed a threshold level K at time iT is, from (3),

$$F_i(K) \equiv \sum_{j=0}^{\infty} H_j(iT) G^{(j)}(K) \quad (i = 1, 2, \cdots), \tag{4}$$

where $F_0(K) \equiv 1$, the probability that its total amount exceeds exactly a level K during $((i-1)T, iT]$ is $F_{i-1}(K) - F_i(K)$.

Suppose that full backup cost is c_1, and cumulative backup cost is $c_2 + c_0(x)$ when the total amount of updated files is x $(0 \le x < K)$. It is assumed that the function $c_0(x)$ is continuous and strictly increasing with $c_0(0) \equiv 0$ and $c_2 < c_1 \le c_2 + c_0(K)$. Then, from (4), the expected cost of cumulative backup, when it is performed at time iT, is

$$C_I(i, K) \equiv \sum_{j=0}^{\infty} H_j(iT) \int_0^K [c_2 + c_0(x)] \mathrm{d}G^{(j)}(x) / F_i(K) \quad (i = 1, 2, \cdots), \tag{5}$$

where $C_I(0, K) \equiv 0$. Further, the mean time to full backup is

$$\sum_{i=1}^{N-1} (iT)[F_{i-1}(K) - F_i(K)] + (NT) F_{N-1}(K) = T \sum_{i=0}^{N-1} F_i(K), \tag{6}$$

and the total expected cost to full backup is

$$\sum_{i=1}^{N-1} \left[c_1 + \sum_{j=0}^{i-1} C_I(j, K) \right] [F_{i-1}(K) - F_i(K)] + \left[c_1 + \sum_{j=0}^{N-1} C_I(j, K) \right] F_{N-1}(K)$$
$$= c_1 + \sum_{i=1}^{N-1} C_I(i, K) F_i(K). \tag{7}$$

Therefore, the expected cost per unit time in the steady-state is

$$C(K, N) \equiv \frac{c_1 + \sum_{i=1}^{N-1} C_I(i, K) F_i(K)}{T \sum_{i=0}^{N-1} F_i(K)}. \tag{8}$$

4. Optimal policy

We discuss optimal values K^* and N^* which minimize the expected cost $C(K,N)$ in (8). We have that $C(0,N) \equiv \lim_{K\to 0} C(K,N) = c_1/T$ for all N, $C(K,1) = c_1/T$ for any K and $C(\infty,\infty) \equiv \lim_{K\to\infty,N\to\infty} C(K,N) = [c_2 + c_0(\infty)]/T$, when $M(K) \equiv \sum_{j=1}^{\infty} G^{(j)}(K) < \infty$. Thus, there exists a positive pair (K^*,N^*) $(0 < K^*, N^* \leq \infty)$ which minimizes $C(K,N)$.

Differentiating $C(K,N)$ with respect to K and setting it equal to zero, we have

$$\sum_{i=0}^{N-1} [c_2 + c_0(K) - C_I(i,K)]F_i(K) = c_1 - c_2. \tag{9}$$

Forming the inequalities $C(K,N+1) \geq C(K,N)$ and $C(K,N) < C(K,N-1)$, we have

$$L(N,K) \geq c_1 - c_2 \text{ and } L(N-1,K) < c_1 - c_2, \tag{10}$$

where

$$L(N,K) \equiv \sum_{i=0}^{N-1} [C_I(N,K) - C_I(i,K)]F_i(K). \tag{11}$$

Noting that $C_I(N,K) < c_2 + c_0(K)$ from (5) scine the cost function $c_0(x)$ is continuous and strictly increasing, we have that there does not exist a positive pair (K^*,N^*) $(0 < K^*, N^* < \infty)$ which satisfies (9) and (10), simultaneously. Thus, if such a positive pair (K^*,N^*) exists which minimizes $C(K,N)$ in (8), then $K^* = \infty$ or $N^* = \infty$.

4.1. Optimal level

Consider an optimal level for full backup, *i.e.*, a database undergoes full backup at time iT $(i = 1,2,\cdots)$ only when the total updated files have exceeded exactly a level K during $((i-1)T, iT]$. Putting $N = \infty$ in (8), the expected cost is

$$\begin{aligned} C_1(K) &\equiv \lim_{N\to\infty} C(K,N) \\ &= \frac{c_1 + \sum_{i=1}^{\infty}[c_2 F_i(K) + \sum_{j=0}^{\infty} H_j(iT) \int_0^K c_0(x)\mathrm{d}G^{(j)}(x)]}{T\sum_{i=0}^{\infty} F_i(K)}. \end{aligned} \tag{12}$$

A necessary condition that an optimal K^* minimizes $C_1(K)$ is

$$\sum_{i=0}^{\infty}\sum_{j=0}^{\infty} H_j(iT) \int_0^K G^{(j)}(x)\mathrm{d}c_0(x) = c_1 - c_2. \tag{13}$$

In particular, suppose that $c_0(x) = a\,(1 - \mathrm{e}^{-sx})$ $(a, s > 0)$. Letting $Q(K)$ be the left-hand side of (13), we have

$$Q(K) = a\sum_{i=0}^{\infty}\left[\sum_{j=0}^{\infty} H_j(iT)\int_0^K \mathrm{e}^{-sx}\mathrm{d}G^{(j)}(x) - \mathrm{e}^{-sK}F_i(K)\right]. \tag{14}$$

It can be easily seen that $Q(K)$ is a strictly increasing function from 0 to $Q(\infty) = a\sum_{i=0}^{\infty}\mathrm{e}^{-R(iT)[1-g(s)]}$, where $g(s) \equiv \int_0^{\infty}\mathrm{e}^{-sx}\mathrm{d}G(x)$ denotes the Laplace–Stieltjes transform of $G(x)$.

Therefore, we have the following optimal policy:

(i) if $Q(\infty) > c_1 - c_2$ then there exists a finite and unique K^* $(0 < K^* < \infty)$ which minimizes $C_1(K)$, and it satisfies (13). In this case, the resulting expected cost is

$$C_1(K^*) = \frac{c_2 + a\left(1 - \mathrm{e}^{-sK^*}\right)}{T}; \tag{15}$$

(ii) if $Q(\infty) \le c_1 - c_2$ then $K^* = \infty$ and $C_1(\infty) = (c_2 + a)/T$.

Example 1. Suppose that a database system is updated according to a Poisson process with rate λ, *i.e.*, $\lambda(t) = \lambda$. Further, it is assumed that $G(x) = 1 - \mathrm{e}^{-\mu x}$, *i.e.*, $G^{(j)}(x) = 1 - \sum_{i=0}^{j-1}[(\mu x)^i/i!]\mathrm{e}^{-\mu x}$ and $M(K) = \mu K$. Then, equation (13) is simplified as

$$a\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\frac{(i\lambda T)^j}{j!}\mathrm{e}^{-i\lambda T}\sum_{m=j}^{\infty}\frac{[\mu^j(s+\mu)^{m-j} - \mu^m]K^m}{m!}\mathrm{e}^{-(s+\mu)K} = c_1 - c_2. \tag{16}$$

The left-hand side of (16) is a strictly increasing function of K from 0 to $a/\left(1 - \mathrm{e}^{-\frac{s\lambda T}{s+\mu}}\right)$. Noting that $c_1 \le c_2 + a$, there exists a finite and unique K^* $(0 < K^* < \infty)$ which satisfies (16).

Table 1 gives the optimal level K^* for full backup, and the resulting cost $C_1(K^*)T$ for $s = 2 \times 10^{-2}$, 2×10^{-3}, 2×10^{-4} and 2×10^{-5} when $c_1 = 3$, $c_2 = 1$, $a = 2$, $\mu = 1$ and $\lambda T = 100$.

TABLE 1. Optimal level K^* and the resulting cost $C_1(K^*)T$.

s	2×10^{-2}	2×10^{-3}	2×10^{-4}	2×10^{-5}
K^*	133	337	1064	3180
$C_1(K^*)T$	2.859	1.979	1.375	1.123

4.2. Optimal number

Next, consider an optimal number when a database undergoes full backup only at time NT $(N = 1, 2, \cdots)$. Putting that $K = \infty$ in (8), the expected cost is

$$\begin{aligned} C_2(N) &\equiv \lim_{K\to\infty} C(K,N) \\ &= \frac{c_2}{T} + \frac{c_1 - c_2 + \sum_{i=1}^{N-1}\sum_{j=0}^{\infty} H_j(iT)\int_0^\infty c_0(x)\mathrm{d}G^{(j)}(x)}{NT}. \end{aligned} \tag{17}$$

From the inequality $C_2(N+1) - C_2(N) \geq 0$, we have

$$\sum_{i=0}^{N-1}\sum_{j=0}^{\infty}[H_j(NT) - H_j(iT)]\int_0^\infty c_0(x)\mathrm{d}G^{(j)}(x) \geq c_1 - c_2. \tag{18}$$

In particular, suppose that $c_0(x) = a\,(1 - \mathrm{e}^{-sx})$ $(a, s > 0)$. Letting $L(N)$ be the left-hand side of (18), we have

$$L(N) = a\sum_{i=0}^{N-1}\left\{\mathrm{e}^{-R(iT)[1-g(s)]} - \mathrm{e}^{-R(NT)[1-g(s)]}\right\}, \tag{19}$$

which is strictly increasing to $L(\infty) = a\sum_{i=0}^{\infty}\mathrm{e}^{-R(iT)[1-g(s)]}$. Note that $L(\infty) = Q(\infty)$ in Section 4.1.

Therefore, we have the following optimal policy:

(iii) if $L(\infty) > c_1 - c_2$ then there exists a finite and unique minimum N^* $(1 \leq N^* < \infty)$ which satisfies (18);

(iv) if $L(\infty) \leq c_1 - c_2$ then $N^* = \infty$.

Example 2. Suppose that $\lambda(t) = \lambda$ and $G(x) = 1 - \mathrm{e}^{-\mu x}$. Then, an N^* is given by a finite and unique minimum such that

$$a\left(\frac{1 - \mathrm{e}^{-\frac{Ns\lambda T}{s+\mu}}}{1 - \mathrm{e}^{-\frac{s\lambda T}{s+\mu}}} - N\mathrm{e}^{-\frac{Ns\lambda T}{s+\mu}}\right) \geq c_1 - c_2. \tag{20}$$

Table 2 gives the optimal number N^* and the resulting cost $C_2(N^*)T$ for $s = 2 \times 10^{-2}$, 2×10^{-3}, 2×10^{-4} and 2×10^{-5} when $c_1 = 3$, $c_2 = 1$, $a = 2$, $\mu = 1$ and $\lambda T = 100$.

TABLE 2. Optimal number N^* and the resulting cost $C_2(N^*)T$.

s	2×10^{-2}	2×10^{-3}	2×10^{-4}	2×10^{-5}
N^*	2	4	11	32
$C_2(N^*)T$	2.859	1.980	1.375	1.123

Compared with Tables 1 and 2, two expected costs $C_1(K^*)$ and $C_2(N^*)$ are almost the same, *i.e.*, $C_1(K^*) \leq C_2(N^*)$. Thus, if two costs of backups are the same, we should adopt the level policy in Section 4.1 as full backup scheme. However, it would be generally easier to count the number of backups than to check the amount of updated files. From this point of view, the number policy would be better than the level policy. Therefore, how to select among two policies would depend on actual mechanism of a database system.

5. Conclusions

We have considered two schemes of cumulative and full backup for a database system, and have analytically discussed optimal backup policies which minimize the expected cost, using the theory of cumulative processes. It would be of interest that if two expected cost rates are almost the same, we should select the number policy rather than the level one as full backup scheme. These results would be applied to the backup of a database, by estimating the costs of two backups and the amount of updated files from actual data. However, backup schemes become very important and much complicated, as database systems have been largely used in most computer systems and information technologies have been greatly developed. These formulations and techniques used in this paper would be useful and helpful for analyzing such backup policies.

References

[1] R.E. Barlow and F. Proschan, *Mathematical Theory of Reliability.* John Wiley & Sons, New York (1965).
[2] D.R. Cox, *Renewal Theory.* Methuen, London (1962).
[3] J.D. Esary, A.W. Marshall and F. Proschan, Shock models and wear processes. *Ann. Probab.* **1** (1973) 627-649.
[4] R.M. Feldman, Optimal replacement with semi-Markov shock models. *J. Appl. Probab.* **13** (1976) 108-117.
[5] S. Fukumoto, N. Kaio and S. Osaki, A study of checkpoint generations for a database recovery mechanism. *Comput. Math. Appl.* **1/2** (1992) 63-68.
[6] T. Nakagawa, On a replacement problem of a cumulative damage model. *Oper. Res. Quarterly* **27** (1976) 895-900.
[7] T. Nakagawa, A summary of discrete replacement policies. *Eur. J. Oper. Res.* **17** (1984) 382-392.
[8] T. Nakagawa and M. Kijima, Replacement policies for a cumulative damage model with minimal repair at failure. *IEEE Trans. Reliability* **13** (1989) 581-584.

[9] C.H. Qian, S. Nakamura and T. Nakagawa, Cumulative damage model with two kinds of shocks and its application to the backup policy. *J. Oper. Res. Soc. Japan* **42** (1999) 501-511.
[10] T. Satow, K. Yasui and T. Nakagawa, Optimal garbage collection policies for a database in a computer system. *RAIRO: Oper. Res.* **4** (1996) 359-372.
[11] K. Suzuki and K. Nakajima, Storage management software. *Fujitsu* **46** (1995) 389-397.
[12] H.M. Taylor, Optimal replacement under additive damage and other failure models. *Naval Res. Logist. Quarterly* **22** (1975) 1-18.

To access this journal online:
www.edpsciences.org