The Ninth International Symposium on Operations Research and Its Applications (ISORA'10)
Chengdu-Jiuzhaigou, China, August 19–23, 2010

Optimal Policies for a Generational Garbage Collector with Tenuring Threshold

Xu-Feng Zhao[1,2] Syouji Nakamura[3] Cun-Hua Qian[1]

[1]College of Management Science and Engineering, Nanjing University of Technology
Nanjing 210009, China
[2]Department of Business Administration, Aichi Institute of Technology
1247 Yachigusa, Yakusa-cho, Toyota 470-0392, Japan
[3]Department of Human Life and Information, Kinjo Gakuin University
1723 Omori 2-chome, Moriyama-ku, Nagoya 463-8521, Japan

Abstract It is an important problem to determine the tenuring threshold to meet the pause time goal for a generational garbage collector. From such viewpoint, this paper proposes two stochastic models based on the working schemes of a generational garbage collector: One is random minor collection which occurs at a nonhomogeneous Poisson process and the other is periodic minor collection which occurs at periodic times. Since the cost suffered for minor collection increases, as the amount of surviving objects accumulates, tenuring minor collection should be made at some tenuring threshold. Using the techniques of cumulative processes and reliability theory, expected cost rates with tenuring threshold are obtained, and optimal policies which minimize them are discussed analytically and computed numerically.

Keywords Garbage Collection; Tenuring Threshold; Minor Collection; Cumulative Process; Reliability

1 Introduction

The technique of *garbage collection* [1] is the automatic process of memory recycling in computer science community, in which objects no longer referenced by program are called *garbage* and should be thrown away. A *garbage collector* must determine which objects are garbage and make the heap space occupied by such garbage available again for subsequent new objects.

A garbage collection plays an important role of Java's security strategy, however, it adds a large overhead that can deteriorate program performance. In recent years, generational garbage collection is popular with programmers for the reason that it can be made more efficiently and fast. Based on the weak generational hypothesis which asserts that most objects are short-lived after their allocation, a *generational garbage collector* segregate objects by age into two or more regions of the heap called *generations* [2]. For instance, the garbage collector, which is used in Sun's HotSpot Java Virtual Machine, manages heap space for both young and old generations [3]: new objects space *Eden*, two equal survivor spaces for surviving objects *SS♯1* and *SS♯2*, and tenured objects space *Old* (*Tenured*), where Eden, SS♯1 and SS♯2 are for young generation, and Old (Tenured) is for

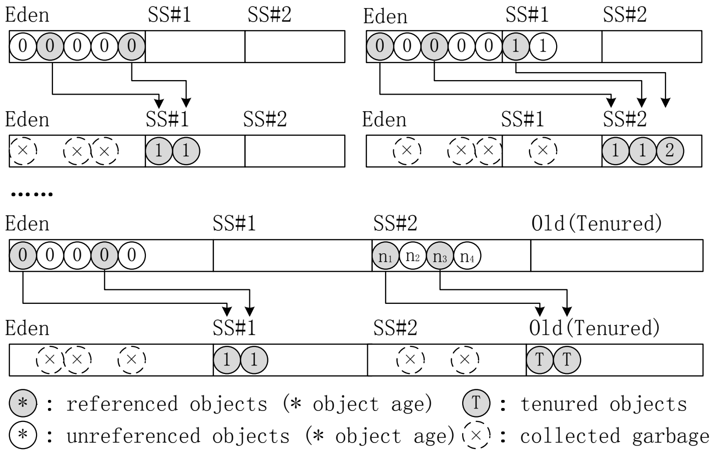

Figure 1: One cycle of minor collections

old one. Different generations can be collected at different frequencies, which means that young generation is collected more frequently than old one.

A generational garbage collector uses minor collection for young generation and major collection for multi-generation. Most generational garbage collectors are copying collectors, although it is possible to use mark-sweep collectors [4]. In this paper, we concentrate only on a generational garbage collector using copying collection. New objects are allocated in Eden. When Eden fills up, *minor collection* occurs and surviving objects are copied from Eden to survivor space. When Eden fills up again, all surviving objects from Eden and from the previously used survivor space are copied into the other survivor space. In this fashion, one survivor space always maintains surviving objects, while the other is empty. The minor collection copies surviving objects between survivor spaces until they become tenured, *i.e.*, *tenuring minor collection* occurs, and then, those objects are copied to old generation (Fig.1). Therefore, Old contains the tenured objects that are expected to be long-lived. When Old fills up, a *major collection* of the whole heap occurs, and surviving objects from Old are kept in Old, while objects from Eden and survivor space are kept in a survivor space. However, for every minor collection, the manner of *stop and copy* pauses all application threads to collect the garbage. The duration of time in which garbage collection has worked is called *pause time* [1], which is an important parameter for interactive systems, and depends largely upon the amount of surviving objects.

As an application of damage models, garbage collection models for database in a computer system [5] were studied, however, the theoretical points of garbage collection was not considered essentially. With regarding to garbage collection modeling, there have been very few research papers that studied analytically optimal policies for a generational garbage collector. Most problems were concerned with several ways to introduce garbage collection methods. This paper considers a pause time goal which is called time cost or cost for the simplicity. Our problem is to obtain an optimal tenuring threshold which minimizes the expected cost rate: If tenuring threshold is too low, objects that would have died in young generation are copied into old generation. This causes frequent tenuring minor collection, and the old generation is filled up with garbage too soon, resulting in a major collection with a long pause time. On the other hand, if the tenuring threshold is too high, although objects have enough time to die, the amount of surviving objects accumulates in the survivor space, and cost suffered for minor collection increases.

We formulate two stochastic models based on the working schemes of a generational garbage collector: random and periodic minor collection models. Using the techniques of cumulative processes [6] and reliability theory [7, 8], expected cost rates are obtained, and optimal policies which minimize them are discussed analytically and computed numerically.

2 Models and Optimal Policies

In this paper, two basic assumptions are given as follows: (i) Survivor rate α_k $(0 \leq \alpha_k < 1; k = 1,2,3,\cdots)$, where $1 > \alpha_1 > \alpha_2 > \cdots > \alpha_k > \cdots \geq 0$, means that new objects will survive $100\alpha_k$ percent at the kth minor collection. (ii) New objects can be tenured only if they survive at least one minor collection, because objects that survive two minor collections are much less than those survive just one minor collection [1], *i.e.*, increasing the number of minor collection beyond the two times is likely to reduce surviving objects slightly.

2.1 Random Minor Collection

For the second basic assumption, there are some objects $\alpha_1 G$ in a survivor space at time 0, which are surviving objects from last tenuring minor collection, where G is a constant. New objects are allocated in Eden. When new objects reach the threshold G in Eden, minor collection occurs, and the surviving objects $\alpha_1 G$ from Eden and $\alpha_2 G$ from survivor space are copied to the other space. When new objects reach the threshold G in Eden again, all surviving objects $\alpha_1 G$ from Eden and $\alpha_2 G + \alpha_3 G$ from previously used survivor space are copied into the other space. In the fashion above, tenuring minor collection is made at Nth $(N = 1,2,\cdots)$ minor collection. The surviving objects from Eden and from survivor space are copied to the survivor space and Old, respectively. After tenuring minor collection, another minor collection cycle begins.

It is assumed that the times between the new objects reach G are random variables, *i.e.*, minor collection occurs at a nonhomogeneous Poisson process with an intensity function $\lambda(t)$ and a mean-value function $R(t) \equiv \int_0^t \lambda(u)\mathrm{d}u$. Then, the probability that minor collections occur exactly j times in $(0,t]$ is

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

and the mean time to tenuring minor collection is

$$\int_0^\infty t H_{N-1}(t)\lambda(t)\mathrm{d}t = \sum_{j=0}^{N-1}\int_0^\infty H_j(t)\mathrm{d}t. \tag{1}$$

Then, the surviving objects at the kth minor collection is

$$\alpha_1 G + \alpha_2 G + \cdots + \alpha_{k+1} G = G \sum_{i=1}^{k+1} \alpha_i \quad (k = 1, 2, \cdots).$$

The following costs are introduced: Let $c_1 + c_2 x$ be the cost suffered for every minor collection and c_3 $(c_3 > c_1)$ be the cost suffered for tenuring minor collection, where x is the surviving objects that should be copied. Then, the expected cost at kth minor collection is

$$C_k \equiv c_1 + c_2 G \sum_{i=1}^{k+1} \alpha_i \quad (k = 1, 2, \cdots),$$

where $C_0 \equiv 0$. Therefore, the expected cost rate is

$$C_1(N) = \frac{\sum_{j=1}^{N-1} C_j + c_3}{\sum_{j=0}^{N-1} \int_0^\infty H_j(t)\mathrm{d}t} \quad (N = 1, 2, \cdots). \tag{2}$$

We seek an optimal number N_1^* that minimizes $C_1(N)$. From the inequality $C_1(N+1) - C_1(N) \geq 0$,

$$C_N \frac{\sum_{j=0}^{N-1} \int_0^\infty H_j(t)\mathrm{d}t}{\int_0^\infty H_N(t)\mathrm{d}t} - \sum_{j=1}^{N-1} C_j \geq c_3. \tag{3}$$

Denoting the left-hand side in (3) by $L(N)$,

$$L(N+1) - L(N) = \sum_{j=0}^{N} \int_0^\infty H_j(t)\mathrm{d}t \left(\frac{C_{N+1}}{\int_0^\infty H_{N+1}(t)\mathrm{d}t} - \frac{C_N}{\int_0^\infty H_N(t)\mathrm{d}t} \right).$$

From [7,p.97], if $\lambda(t)$ is increasing in t, then $\int_0^\infty H_j(t)\mathrm{d}t$ is decreasing in j, and converges to $1/\lambda(\infty)$ as $j \to \infty$, where $1/\lambda(\infty) = 0$ whenever $\lambda(\infty) = \infty$. Thus, because C_j is increasing strictly in j, if $\lambda(t)$ is increasing in t and $L(\infty) > c_3$, then there exists a finite and unique minimum N_1^* $(1 \leq N_1^* < \infty)$ which satisfies (3), and the expected cost rate is

$$\frac{C_{N_1^*-1}}{\int_0^\infty H_{N_1^*-1}(t)\mathrm{d}t} < C_1(N_1^*) \leq \frac{C_{N_1^*}}{\int_0^\infty H_{N_1^*}(t)\mathrm{d}t}. \tag{4}$$

2.2 Periodic Minor Collection

For the second basic assumption, there are some objects $\alpha_1 X_0$ in a survivor space at time 0, which are surviving objects from the last tenuring minor collection. New objects are allocated in Eden. It is assumed that minor collection occurs at time kT $(k = 1, 2, \cdots)$ for constant $T > 0$ and an amount X_k of new objects in Eden at kT has an identical distribution $G(x) \equiv \Pr\{X_k \leq x\}$. That is, when the first minor collection occurs, surviving objects $\alpha_1 X_1$ from Eden and $\alpha_2 X_0$ from survivor space are copied to the other space.

When the second minor collection occurs, surviving objects $\alpha_1 X_2$ from Eden and $\alpha_2 X_1 + \alpha_3 X_0$ from the previously used survivor space are copied into the other space. In the fashion above, tenuring minor collection is made at time NT $(N = 1, 2, \cdots)$. The other assumptions are the same as the random minor collection.

Then, the surviving objects at the kth minor collection is

$$\alpha_1 X_k + \alpha_2 X_{k-1} + \cdots + \alpha_{k+1} X_0 = \sum_{j=0}^{k} \alpha_{j+1} X_{k-j} \quad (k = 1, 2, \cdots).$$

Because X_0 and X_k $(k = 1, 2, \cdots)$ have an identical distribution $G(x)$, the distribution of the total surviving objects at the kth collection is

$$\Pr\left\{ \sum_{j=0}^{k} \alpha_{j+1} X_{k-j} \le x \right\} = G^{(k)}(x) \quad (k = 1, 2, \cdots),$$

and the expected cost of minor collection at time kT is

$$\widehat{C}_k = \int_0^\infty (c_1 + c_2 x) \mathrm{d}G^{(k)}(x) \quad (k = 1, 2, \cdots),$$

where $\widehat{C}_0 \equiv 0$. Therefore, the expected cost rate is

$$C_2(N) = \frac{c_3 - c_1 + c_2 \sum_{k=1}^{N-1} \int_0^\infty x \mathrm{d}G^{(k)}(x)}{NT} + \frac{c_1}{T} \quad (N = 1, 2, \cdots). \tag{5}$$

We seek an optimal number N_2^* analytically that minimizes $C_2(N)$. From the inequality $C_2(N+1) - C_2(N) \ge 0$,

$$\sum_{k=1}^{N-1} \int_0^\infty \left[G^{(k)}(x) - G^{(N)}(x) \right] \mathrm{d}x + \int_0^\infty \left[1 - G^{(N)}(x) \right] \mathrm{d}x \ge \frac{c_3 - c_1}{c_2}. \tag{6}$$

Denoting the left-hand side in (6) by $U(N)$,

$$U(N+1) - U(N) = (N+1) \int_0^\infty \left[G^{(N)}(x) - G^{(N+1)}(x) \right] \mathrm{d}x.$$

Because $G^{(j)}(x)$ is decreasing in j, $U(N)$ is increasing in N. If $U(\infty) > (c_3 - c_1)/c_2$, then there exists a finite and unique minimum N_2^* $(1 \le N_2^* < \infty)$ which satisfies (6), and the expected cost rate is

$$\frac{c_2 \int_0^\infty \left[1 - G^{(N_2^*-1)}(x) \right] \mathrm{d}x}{T} < C_2(N_2^*) - \frac{c_1}{T} \le \frac{c_2 \int_0^\infty \left[1 - G^{(N_2^*)}(x) \right] \mathrm{d}x}{T}. \tag{7}$$

3 Numerical Examples

For random minor collection, suppose that minor collection occurs in a Poisson process with rate λ, *i.e.*, $\lambda(t) \equiv \lambda$, $H_j(t) = [(\lambda t)^j / j!] \mathrm{e}^{-\lambda t}$. Then, from (3),

$$\sum_{j=0}^{N-1} (C_N - C_j) \ge c_3, \tag{8}$$

Table 1: Optimal N^* and $C(N^*)$ when $c_1 = 1$, $c_2G = c_2\mu = 10$ and $T = 1/\lambda = 1$.

α	c_3 = 20		40		60	
	N^*	$C(N^*)$	N^*	$C(N^*)$	N^*	$C(N^*)$
0.2	11	7.1343	22	8.3990	32	9.1519
0.4	6	10.9333	11	13.3595	17	14.8028
0.6	4	13.8750	8	17.4705	11	19.5847
0.8	3	16.2222	6	21.0333	9	23.8131

whose left-hand side is increasing in N. Thus, if $\sum_{j=0}^{\infty}(C_\infty - C_j) \geq c_3$, then $1 \leq N_1^* < \infty$. In this case, if $C_1 > c_3$, which means that the first minor collection cost is greater than tenuring minor collection cost, then $N_1^* = 1$.

For periodic minor collection, When X_i $(i = 0,1,2,\cdots)$ has a normal distribution $N(\mu_i, \sigma_i^2)$

$$\sum_{i=0}^{k} \alpha_{i+1} X_{k-i} \sim N\left(\sum_{i=0}^{k} \alpha_{i+1}\mu_{k-i}, \sum_{i=0}^{k} \alpha_{i+1}^2 \sigma_{k-i}^2\right).$$

Then, from (6),

$$\sum_{k=1}^{N}\sum_{i=0}^{N} \alpha_{i+1}\mu_{N-i} - \sum_{k=1}^{N-1}\sum_{i=0}^{k} \alpha_{i+1}\mu_{k-i} \geq \frac{c_3 - c_1}{c_2}. \tag{9}$$

In this case, a finite N_2^* $(1 \leq N_2^* < \infty)$ exists uniquely. If $c_1 + c_2(\alpha_1\mu_1 + \alpha_2\mu_0) > c_3$, then $N_2^* = 1$.

Suppose that $\alpha_k = \alpha/k$ $(0 < \alpha < 1; k = 1,2,\cdots)$. An optimal N_1^* $(1 \leq N_1^* < \infty)$ satisfies, from (8),

$$N + 1 - \sum_{j=1}^{N} \frac{1}{j+1} \geq \frac{c_3 - c_1}{\alpha G c_2}, \tag{10}$$

an optimal N_2^* $(1 \leq N_2^* < \infty)$ satisfies, from (9),

$$N\sum_{i=0}^{N} \frac{\mu_{N-i}}{i+1} - \sum_{k=1}^{N-1}\sum_{i=0}^{k} \frac{\mu_{k-i}}{i+1} \geq \frac{c_3 - c_1}{\alpha c_2}. \tag{11}$$

In particular, when $\mu_k \equiv \mu$ and $\sigma_k \equiv \sigma$, this agrees with (10) for $G = \mu$, *i.e.*, when $G = \mu$ and $T = 1/\lambda$, $C_1(N)$ and $C_2(N)$ become

$$C(N) \equiv \frac{C_1(N)}{\lambda} = \frac{C_2(N)}{\lambda} = \frac{(N-1)c_1 + c_2\mu\alpha\sum_{j=1}^{N-1}\sum_{i=1}^{j+1}(1/i) + c_3}{N}.$$

Tables 1 presents the optimal N^* and $C(N^*)$, for $c_3 = 20$, 40, 60 and $\alpha = 0.2$, 0.4, 0.6, 0.8 when $c_1 = 1$, $c_2G = c_2\mu = 10$, $T = 1/\lambda = 1$. These show that N^* is increasing with c_3 and decreasing with α, $C(N^*)$ is increasing with both c_3 and α.

4 Conclusions

We have solved the optimization problem theoretically when to make the tenuring minor collection for a generational garbage collector. Two stochastic models were considered, where the random and periodic minor collections occur at a nonhomogeneous Poisson process and at periodic times, respectively. Using the techniques of cumulative processes and reliability theory, the expected cost rates of each model were derived, and the optimal policies which minimize them were discussed analytically. Furthermore, optimal policies and their expected cost rates were computed and compared numerically. Useful discussions for these results were made. Such theoretical results would be applied to actual garbage collections by suitable modifications.

Acknowledgements

This project is supported by National Natural Science Foundation of China (70471017, 70801036); Humanities and Social Science Research Foundation of China (05JA630027); Grant-in-Aid for Scientific Research (C) of Japan Society for the Promotion of Science under Grant No. 22500897.

References

[1] Jones, R. and Lins, R. (1996) *Garbage Collection: Algorithms for Automatic Dynamic Memory Management*, John Wiley & Sons, Chichester.

[2] Ungar, D. (1984) *Generation scavenging: A non-disruptive high performance storage reclamation algorithm*, ACM Sigplan Notices 19(5), 157-167.

[3] Vengerov, D. (2009) *Modeling, analysis and throughput optimization of a generational garbage collector*, Technical Report, Sun Labs.

[4] Armstrong, J. and Virding, R. (1995) *One-pass real-time generational mark-sweep garbage collection*, Proceedings of International Workshop on Memory Management, Sweden.

[5] Satow, T., Yasui, K. and Nakagawa, T. (1996) *Optimal garbage collection policies for a database in a computer system*, RAIRO Operations Research 30(4): 359-372.

[6] Nakagawa, T. (2007) *Shock and Damage Models in Reliability Theory*, Springer, London.

[7] Nakagawa, T. (2005) *Maintenance Theory of Reliability*, Springer, London.

[8] Nakamura, S. and Nakagawa, T. (2010) *Stochastic Reliability Modeling, Optimization and Applications*, World Scientific, Singapore.