Control and Cybernetics

vol. **22** (1993) No. 1/2

A computer memory sharing problem

by

Ryszard Antkiewicz
Tadeusz Nowicki

Military University of Technology
Warsaw
Poland

A multitask computer system with shared memory is described. The memory is considered to be limited. Initially it is shared among tasks in the form of separable memory areas. Each task generates random stream of data which fills memory area connected with it. When any memory area is filled with data the system must be reorganized. The problem formulated and solved is: how to share limited memory among tasks in order to maximize expected time to system reorganization.

1. Introduction

The present paper considers the problem of dividing common and limited resource among a number of users. Users consume their parts of the resource. When one of these parts is consumed completely the resource must be renewed and divided again. In most cases users are interested in maximizing the time periods between moments of resource division and renewal. This is connected with cost of such operations. Costs can be understood in a variety of ways. In some cases users must pay money, in other ones cost means time necessary to renewal and distribution of resource.

It is possible to give many examples of resource division problems encountered in real life. One of them comes from computer systems and consists in sharing of common and limited memory among a number of users.

A user can in this case be: a data set, a process or a program. Memory can mean main or external memory of a computer system.

Especially important is the problem of sharing of limited external memory among files accumulating data connected with fixed discipline. This could be for example represented by: library file, file containing scientific information, medical or statistical data, data of an insurance company, administration file with data about citizens, and so on. Accumulated data can be divided into different types and stored in different subfiles within the same file. Data arrival rates for particular subfiles can differ, and thus subfiles can become busy (filled) within different time. So, it is necessary to allocate more memory to the subfiles characterized by greater speed of data arrival, in order to maximize time until the subsequent reorganization of the file. It is especially important for very large files, whose reorganizations are very expensive.

Other examples of the problem considered are provided by an on–line system comprising n serial files Mendelson, Pliskin, Yechiali (1979), a partitioned data set consisting of several sequential subfiles Mendelson, Pliskin, Yechiali (1980), Mendelson, Pliskin, Yechiali (1979), and the overflow area of some index–sequential file organization scheme Mendelson, Pliskin, Yechiali (1979), Martin (1983).

It is assumed in each of problems mentioned above that shared memory is limited and it is only consumed, i.e. space once used is not available until the next reorganization.

The problem considered is how to best allocate a given total amount of storage space among the files or subfiles.

This problem was formulated in Mendelson, Pliskin, Yechiali (1980) and Mendelson, Pliskin, Yechiali (1979). In Mendelson, Pliskin, Yechiali (1979) the adopted criterion for allocation was probability that time to reorganization would be greater than fixed time. In Mendelson, Pliskin, Yechiali (1980) the criterion was the expected time to reorganization, and this criterion is used in our study. Therefore we consider a computer system with a database. The database has limited memory divided among a few files. The memory spaces connected with different files are disjoint. At random instants records come and are added to the files. Addition of each record to a file requires some amount of storage space. Whenever one file runs out of its space the database must be reorganized. We call such situations failure of a database.

We consider the following problem: how to allocate a given total amount of memory among files in order to maximize expected time to database failure.

2. The mathematical model

Let the database consist of N files. The volume of memory being allocated among files is C. We define memory allocation to files as a vector:

$$x = (x_1, x_2, x_3, \dots, x_n, \dots x_N), \tag{1}$$

where x_n is the volume of memory allocated to n–th file. The following constraints should be satisfied by the vector x:

$$x_n \geq 0, \quad n = \overline{1, N} \tag{2}$$

$$\sum_{n=1}^{N} x_n \leq C. \tag{3}$$

We asume that records come to n–th file at random instants of time and time periods between consecutive instants of record arrivals, $\tau_1^n, \tau_2^n, \ldots$ are independent identically distribuled (i.i.d.) random variables. We suppose also that they are continuous random variables with nondecreasing failure intensity function.

Let $N_n(t)$ for $n = \overline{1, N}$ be a stochastic process whose value at fixed t means the number of records which came to n–th file until time t. From assumptions about $\tau_1^n, \tau_2^n, \ldots$ we know Beichelt, Franken (1983) that processes $N_n(t)$ are renewal processes. Let $V_1^n, V_2^n, \ldots, V_i^n, \ldots$ be random variables whose values mean the volume of storage space needed for addition of records to the n–th file. We assume that $\{V_i^n\}_{i=1,2,\ldots}$ are i.i.d. and continuous.

We suppose that sequences $\{V_i^n\}$ and $\{\tau_j^n\}$ are stochastically independent and that they are also independent for different $n \in \{1, 2, \ldots, N\}$.

The number of records which have come to the n–th file before it runs out of its space is defined as follows:

$$M_n(x) = \max\{m \ : \ \sum_{i=0}^{m} V_i^n < x_n\}, \quad n = \overline{1, N}, \tag{4}$$

and its expected value as

$$E\{M_n(x)\} = H_n(x), \quad n = \overline{1, N}. \tag{4A}$$

The definition of random variables $\{V_i^n\}$ implies that $M_n(x)$ is a renewal process.

Let $T_n(x)$ be the time to failure of the n–th file. Then, for fixed x

$$T_n(x) = \min\left\{t \ : \ \sum_{i=0}^{N_n(t)} V_i^n > x_n\right\} = \sum_{i=0}^{M_n(x)+1} \tau_i^n, \quad n = \overline{1, N} \tag{5}$$

where $V_0^n \equiv 0$

Let $T(x)$ be the time until failure of database for fixed x. Then from (5) it follows that

$$T(x) = \min\{T_1(x), T_2(x), \ldots, T_N(x)\} \tag{6}$$

The problem of allocation of a given total volume of memory among files in order to maximize the expected time to reorganization can be formulated as follows:

$$\max_{x \in X} E\{T(x)\} \tag{7}$$

where

$$X = \{x \in R^N \ : \ \sum_{n=1}^{N} x_n \leq C, \ x_n \geq 0, \ n = \overline{1, N}\}.$$

It is difficult to solve the problem (7) because the formula for the function $E\{T(x)\}$ is unknown. We can find lower approximation for function $E\{T(x)\}$ and apply it as a criterion function.

We find the expected value μ^n of $T_n(x)$ in order to obtain lower approximation for function $E\{T(x)\}$. Then

$$\begin{aligned} \mu^n &= E\{T_n(x)\} = E\left\{\sum_{i=0}^{M_n(x)+1} \tau_i^n\right\} = E\{\tau_1^n\}\, E\{M_n(x)+1\} = \\ &= E\{\tau_1^n\}\,(H_n(x)+1), \quad n = \overline{1,N}. \end{aligned} \tag{8}$$

Let μ_t^n be defined as follows:

$$\mu_t^n = E\{T_n(x) - t \mid T_n(x) \geq t\}, \quad n = \overline{1,N}, \quad t \geq 0.$$

The following Lemma is proved in Antkiewicz, Nowicki (to apprear).

LEMMA 1 $\mu_t^n \leq \mu^n, \quad n = \overline{1,N}.$

It follows from Lemma 1 and Beichelt, Franken (1983)that distribution functions of random variables $T_n(x)$, $n = \overline{1,N}$, satisfy the following inequality:

$$G_n(t) = P\{T_n(x) \geq t\} \geq \begin{cases} 1 - \frac{t}{\mu^n} & \text{for } 0 \leq t \leq \mu^n, \\ 0 & \text{for } t > \mu^n. \end{cases} \tag{9}$$

It could be proved, using (9) and Tchebichev inequality Klimov (1986), that

$$E\{T(x)\} = E\{\min\{T_n(x),\ n = \overline{1,N}\}\} \geq \left(\frac{1}{2}\right)^N \cdot \mu^N \tag{10}$$

where

$$\mu^N = \min\{E\{T_n(x)\},\ n = \overline{1,N}\}.$$

Thus

$$\begin{aligned} E\{T(x)\} &\geq \left(\frac{1}{2}\right)^N \min\{E\{T_n(x)\},\ n = \overline{1,N}\} = \\ &= \left(\frac{1}{2}\right)^N \min\{(H_n(x)+1)\ :\ n = \overline{1,N}\} = D(x). \end{aligned} \tag{10A}$$

By applying the function $D(x)$, we replace the problem (7) by the following optimization problem:

$$\max_{x \in X} D(x) \tag{11}$$

where X is as in problem (7). The problem (11) can be formulated equivalently as:

$$\max_{y \in Y} y \tag{12}$$

where

$$Y = \left\{ (y, x) \in R^{N+1} : y \leq E\{\tau_1^n\} \cdot (H_n(x) + 1),\ x_n \geq 0,\ n = \overline{1, N}, \sum_{n=1}^{N} x_n \leq C \right\}.$$

3. Optimization of memory allocation

Problem (12) is a mathematical programming problem with linear criterion function and nonlinear constraints.

Each function $H_n(x)$ for $n = \overline{1, N}$ is nondecreasing, and therefore it is a quasi-convex function. Thus, we can use Kuhn–Tucker conditions for solving problem (12). The Lagrange function for above problem has the following formula Chudy (1980):

$$\begin{aligned} L(y, x, u) &= -y + \sum_{n=1}^{N} u_n (y - (H_n(x) + 1) \cdot \theta_n) + \\ &\quad - \sum_{n=1}^{N} u_{N+n} \cdot x_n + u_{2N+1} \left(\sum_{n=1}^{N} x_n - C \right), \end{aligned} \tag{13}$$

where $\theta_n = E\{\tau_1^n\}$, $n = \overline{1, N}$.

The Kuhn–Tucker conditions are as follows Chudy (1980):

$$\sum_{n=1}^{N} u_n = 1, \tag{14a}$$

$$u_{2N+1} = u_{N+n} + u_n \theta_n \cdot h_n(x), \quad n = \overline{1, N}, \tag{14b}$$

$$y \leq \theta_n \cdot (h_n(x)) + 1), \quad n = \overline{1, N}, \tag{14c}$$

$$\sum_{n=1}^{N} x_n \leq C, \ \ x_n \geq 0, \quad n = \overline{1, N}, \tag{14d}$$

$$\sum_{n=1}^{N} u_n (y - (H_n(x) + 1) \cdot \theta) - \sum_{n=1}^{N} u_{N+n} \cdot x_n + u_{2N+1} \left(\sum_{n=1}^{N} x_n - C \right) = 0, \tag{14e}$$

$$u_n \geq 0, \quad n = \overline{1, 2N+1} \tag{14f}$$

where

$$h_n(x) = \frac{d}{dx} H_n(x), \quad n = \overline{1, N}.$$

The solution of equations (14) is presented in Antkiewicz, Nowicki (to apprear). It is prepared for two main cases:
(a) functions $H_n(x)$ for $n = \overline{1, N}$ are increasing,
(b) functions $H_n(x)$ for $n = \overline{1, N}$ are nondecreasing.

In the case (a) we can find the solution using the following relations: from equation

$$\sum_{n=1}^{N} H_n^{-1} \left(\frac{y^*}{\theta_n} - 1 \right) = C \tag{15}$$

we find y^* and using formula

$$x_n^* = H_n^{-1} \left(\frac{y^*}{\theta_n} - 1 \right), \quad n = \overline{1, N} \tag{16}$$

we calculate x_n^* for $n = \overline{1, N}$.

In the case (b) situation is the same if solution of equation (15) exists. In the other case, equation (15) must be transformed and solved, but x_n are calculated in the same way as previously from (16).

It is easy to see that Jensen inequality Klimov (1986) implies upper approximation to the function $E\{T(x)\}$:

$$E\{T(x)\} \leq \min\{E\{T_n(x)\},\ n = \overline{1, N}\}. \tag{17}$$

From (17) and (10A) we have, that

$$\left(\frac{1}{2}\right) \min\{E\{T_n(x)\},\ n = \overline{1, N}\} \leq E\{T(x)\} \leq \min\{E\{T_n(x)\},\ n = \overline{1, N}\}$$

We can see that approximation error is less then

$$\left(1 - \left(\frac{1}{2}\right)\right) \min\{E\{T_n(x)\},\ n = \overline{1, N}\}.$$

4. Concluding remarks

We could not find the solution to problem (7) because of unknown criterion function formula. We proposed, therefore, a method for obtaining suboptimal solutions. It consists in formulation and solving of the same problem with the criterion function replaced by its lower approximation. The method for solving the new problem is simple and effective. Another method for solving problem (7) was proposed in Martin (1983). This method was also suboptimal and used the approximation of criterion function, but this approximation was proper only for large x_n, $n = \overline{1, N}$, and was not a lower or upper approximation.

Reference

ANTKIEWICZ R., NOWICKI T. (TO APPEAR) Optymalizacja wstępna podziału pamięci w pewnej klasie baz danych *Advances of Cybernetics*.

BEICHELT F., FRANKEN P. (1983) Zuverlassigkait und Instandhaltung. Matematische Metoden. VEB Verlag Technik, Berlin.

CHUDY M. (1980) Programowanie matematyczne. Część 2. Programowanie nieliniowe. WAT, Warszawa.

KLIMOV G. (1986) Probability Theory and Mathematical Statistics. Mir Publishers, Moscow.

KOPOCIŃSKI B. (1973) Zarys teorii odnowy i niezawodności. PWN, Warszawa.

MARTIN J. (1983) Organizacja baz danych. PWN, Warszawa.

MENDELSON H., PLISKIN J., YECHIALI U. (1980) A Stochastic Allocation Problem. Operations Research No.3

MENDELSON H., PLISKIN J., YECHIALI U. (1979) Optimal Storage Allocation for Serial Files. *Communications of the ACM*, **22**, No.2.

IN MEMORIAM PROFESSOR EDMUND LIPIŃSKI