this paper was appeared in IEEE Transactions on Computers Vol. C-29, no. 7 (July 1980) pp. 611-617

OPTIMIZATION OF MEMORY HIERARCHIES IN MULTIPROGRAMMED COMPUTER SYSTEMS WITH FIXED COST CONSTRAINT

by

TR 78-7

Samuel T. Chanson and Prem S. Sinha
Department of Computer Science
University of British Columbia

ABSTRACT

This paper presents, using queuing theory and optimization techniques, a methodology for estimating the optimal capacities and speeds of the memory levels in a computer system memory hierarchy operating in the multiprogrammed environment. Optimality is with respect to mean system response time under a fixed cost constraint. It is assumed that the number of levels in the hierarchy as well as the capacity of the lowest level are known. The effect of the storage management strategy is characterized by the hit ratio function which, together with the device technology cost functions are assumed to be representable by power functions. It is shown that as the arrival rate of processes and/or the number of active processes in the system increase, the optimal solution deviates considerably from that under a uniprogrammed environment.

I. INTRODUCTION

The optimization of memory hierarchies is recognized as an important research area and has been attacked from several directions. Various solutions optimal under certain constraints have been obtained [1]-[5]. Ramamoorthy and Chandy [1] have obtained the size and type of each memory level by minimizing the average access time of an information block in a program for a given cost constraint. The concept is then extended to a general case in multiprogramming. The approach presupposes the knowledge of the frequency of access for each information block. MacDonald and Sigworth [3] have dealt with various combinations of optimization criteria such as fixed cost constraint, fixed and variable page size etc.. They too assume knowledge of the storage address sequence and have used its statistical properties extensively in their work. The objective function to be minimized is average access time or a function of it. Chow [2] has very nicely applied geometric programming to obtain not only the optimal size and speed of each memory level, but also the optimal number of levels for a given cost constraint. There it is assumed that storage management strategy is characterized by a hit-ratio function. Furthermore, the hit-ratio function and device technology cost function are taken as power functions of the capacity and access time respectively of each level of memory. Chow's analysis is restricted to uniprogrammed systems. Welch [4] gives a very simple and straightforward analysis of memory hierarchy for speed-cost trade-off with the assumption that the size and access probability of each level of memory are known and fixed. Rege [8] uses a simple two-server queuing network model to analyze the cost-performance trade-off by using different sizes and speeds at different memory levels. There is no optimization study.

All previous work in optimization use the mean memory access time as objective function for minimization and with the exception of [1], deal only with the uniprogrammed environment where only one process is active at any time and the processor simply hangs up when a request is made to any memory level. It is not clear that in a multiprogramming environment, where a process may be blocked while it is referencing information in certain memory level that minimizing the average memory access time is meaningful. In this paper we have combined performance evaluation techniques with optimization methods to extend the analysis of Chow [2] to cover multiprogrammed systems. Mean response time[1] is chosen as the objective function. With the number of memory levels fixed and the capacity of the lowest level known, we obtain from queuing theory an expression for mean system response time in terms of the capacity and speed of each memory level. The optimal expression of memory sizes and speeds are then obtained using the Lagrangian function under a cost constraint.

Notice that in the uniprogramming environment,

Average response time = c_1 * Average access time of the memory hierarchy + c_2

where c_1 = average number of accesses to the memory hierarchy per interaction

and c_2 = mean CPU time demand of the process per interaction

c_1 and c_2 are constants for a given process. Hence average response time and average memory hierarchy access time are equivalent objective functions in the uniprogramming environment.

---

[1] The term response time may be interpreted as request completion time in this paper.

II. SYSTEM DESCRIPTION, ASSUMPTIONS AND NOTATIONS

The memory hierarchy consists of N levels, M_1, M_2, ..., M_N, where N is known and fixed. Generally, the higher the level (i.e., the smaller the index) the smaller is the capacity, the faster its speed and the more expensive is its unit cost. It is assumed that information present in any level is also present in all subsequent lower levels. In the case of a uniprogrammed system, whenever the needed information is not found in the highest level M_1, a request is made to each of the lower level successively until it is found in a level M_i, $i = 2, \ldots, N$. The processor is held waiting all the time until the information is retrieved from M_i. As i increases, the time required to fetch the information goes up. When i exceeds a certain value, it becomes uneconomical to keep the processor idle while the information is being retrieved from M_i, particularly when there are other processes waiting for the processor. Thus in the case of multiprogramming, we have two types of memory - A and B. While the processor waits for access to type A memory, it does not do so for access to type B memory, but releases the current process and takes the next process ready to run if one exists. It is therefore possible for several requests to queue up at a type B memory level but there is at most one request at anyone time for a type A memory level. The model of such a system is shown in Figure 1 where n_1 and n_2 are the number of type A and type B memory levels respectively. X_i, $i = 1, 2, \ldots, N$ ($N = n_1 + n_2$ = a known constant) is the capacity of memory level M_i in the hierarchy. X_N is assumed to be known. In addition, in consistent with Chow's [1] terminology, we define the following:

> y_i, $i = 1, 2, \ldots, N$ is the mean transfer time of a unit of information from level M_i to M_{i-1} (this does not include the queue wait time for type B memory levels) and y_1 is simply the mean access time of the fastest memory.

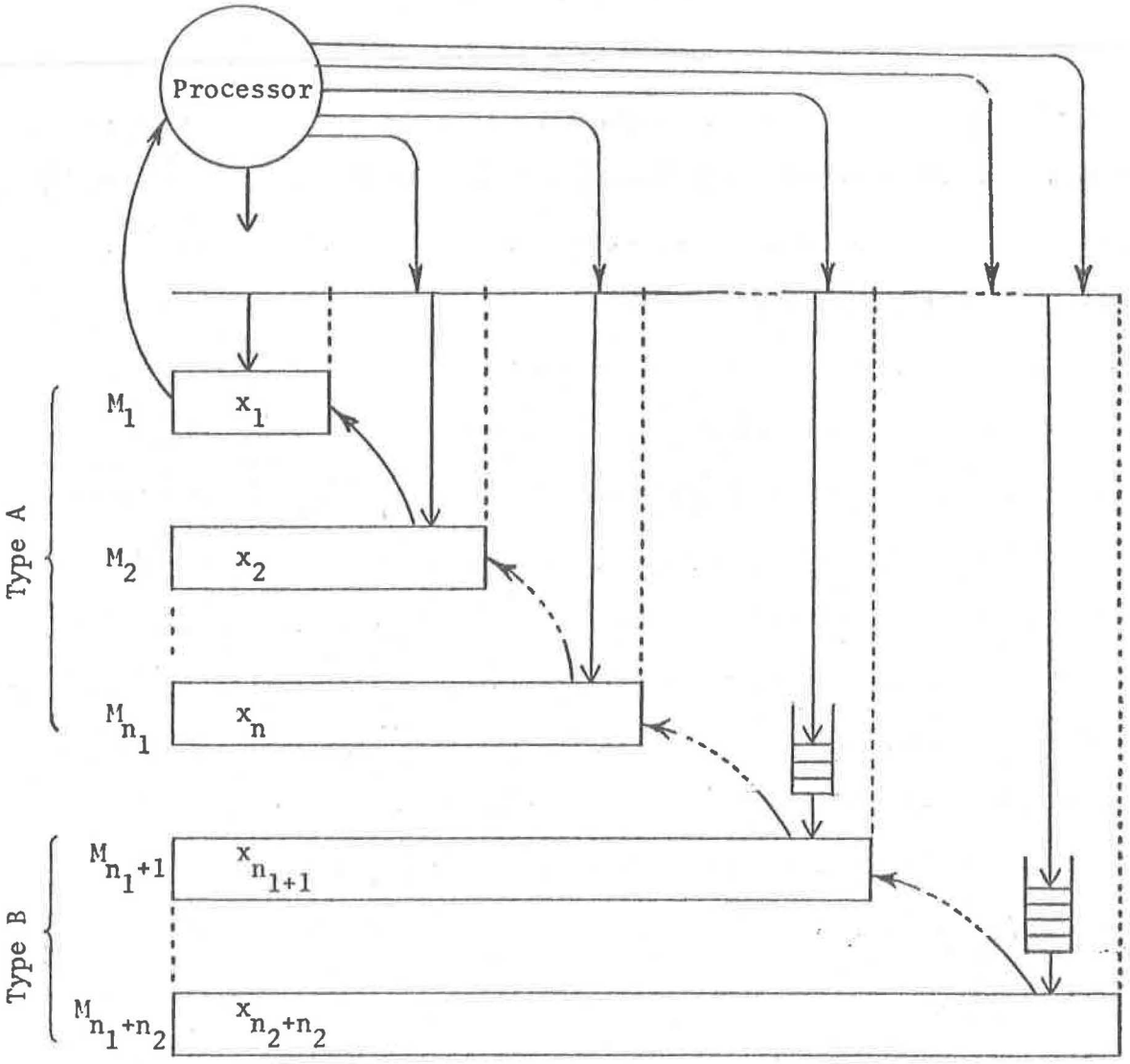

FIGURE 1: STORAGE HIERARCHY IN A MULTIPROGRAMMED SYSTEM

H(x) is the probability of finding the requested information in a memory level with capacity x.

The hit-ratio function p_i (i.e., the probability of successfully retrieving the needed information from level M_i) is therefore given by the difference in the probability of finding the information in M_i but not in M_{i-1}.

$$\text{i.e.,} \quad p_i = H(x_i) - H(x_{i-1}) \,, \qquad i = 1, 2, \ldots, N; \;\; H(x_0) = 0 \tag{1}$$

The missing ratio F(x) is simply 1 - H(x) and is assumed to be a power function of capacity x and positive constants K_1 and α, defined as

$$F(x) = K_1 \, x^{-\alpha} \tag{2}$$

The technology cost function (i.e., unit cost of a storage level with transfer time y to the next higher level) is assumed to take the form

$$b(y) = K_2 \, y^{-\beta} \tag{3}$$

where β and K_2 are positive constants. Without loss of generality, we take $K_1 = K_2 = 1$, i.e., equations (2) and (3) become

$$F(x) = x^{-\alpha} \tag{2'}$$

$$\text{and} \qquad b(y) = y^{-\beta} \tag{3'}$$

This means that $K_1^{\frac{1}{\alpha}}$ is the unit for storage capacity and K_2 is the unit for cost. Empirical data have shown that equations (2) and (3) are good approximations (see [2], [4]).

III. QUEUING MODEL

The queuing model of the system described in the previous section is shown in Figure 2.

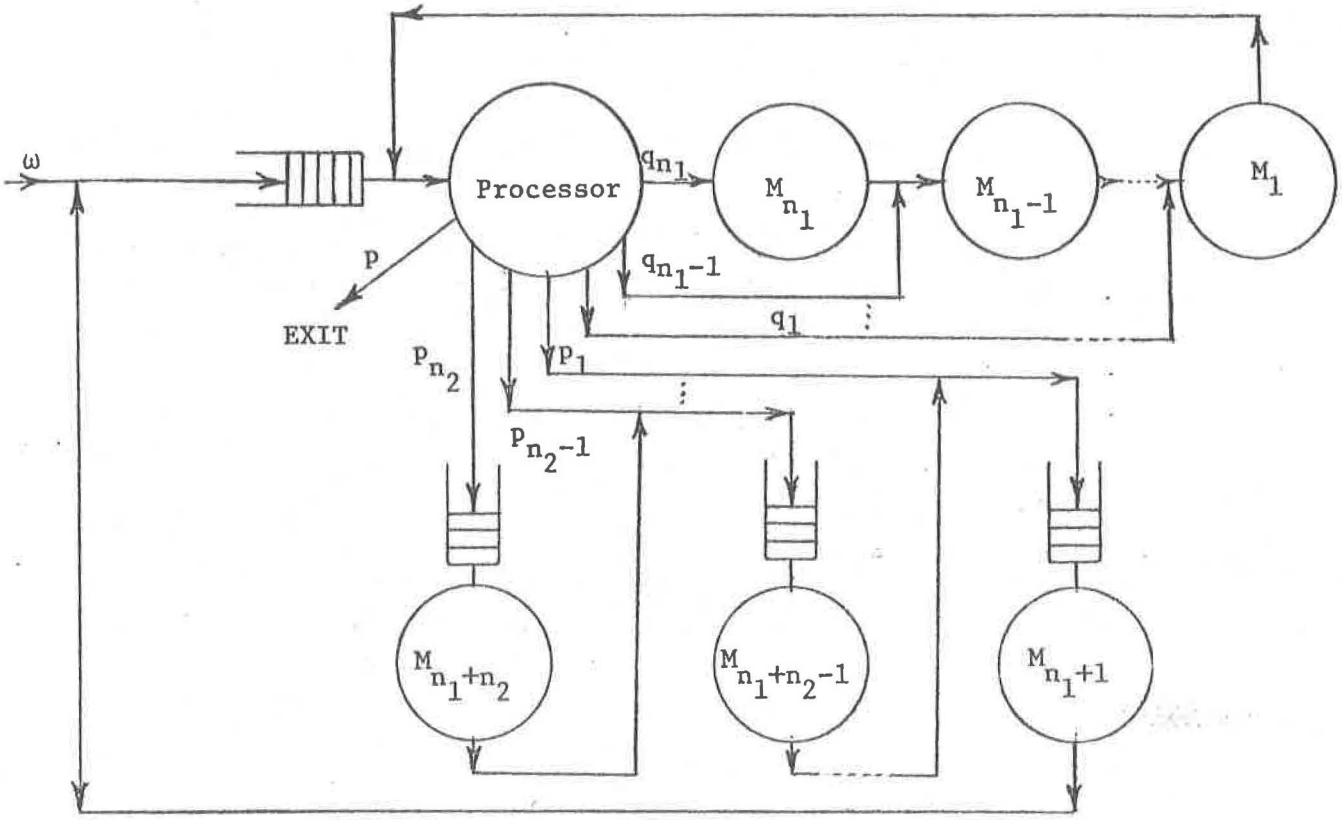

FIGURE 2: QUEUING NETWORK MODEL A OF SYSTEM

where ω is the mean rate of arrival of tasks (or requests in an interactive environment) to the system and is assumed to have an exponential distribution. $q_1, q_2, \ldots, q_{n_1}$ are the probabilities of referencing memory levels $M_1, M_2, \ldots, M_{n_1}$ respectively and $p_1, p_2, \ldots, p_{n_2}$ are the probabilities of referencing memory levels $M_{n_1+1}, \ldots, M_{n_1+n_2}$ respectively. The probability of exit (i.e., termination of task or completion of a request) is p.

Define

$$Q = \sum_{i=1}^{n_1} q_i \quad ,$$

$$P = \sum_{i=1}^{n_2} p_i + p \quad ,$$

then clearly, $P + Q = 1$

For type A memory, the mean effective hierarchy access time T_i to level $M_i (i \leq N_1)$ is the sum of the mean individual transfer time between two consecutive levels from M_1 up to M_i

$$\text{i.e.,} \qquad T_i = \sum_{j=1}^{i} y_j$$

In the case of type B memory the mean transfer times y_i's are taken as the inverse of the service rates of the memory levels. Furthermore, it is assumed that the service rates are exponentially distributed with mean $1/y_i$, $i = n_1 + 1, \ldots, n_1 + n_2$. The mean effective hierarchy access time of type B memory levels are not so easily obtained because of possible queuing of requests at these levels. Furthermore, since a process may be blocked while accessing these levels it is more reasonable to use mean response time (or mean request completion time) as the criteria of optimization. To do so, we first transform the model in Figure 2 to the model in Figure 3. It is easy

to show, using first principles in queuing theory that the two models are equivalent with respect to the objective function.

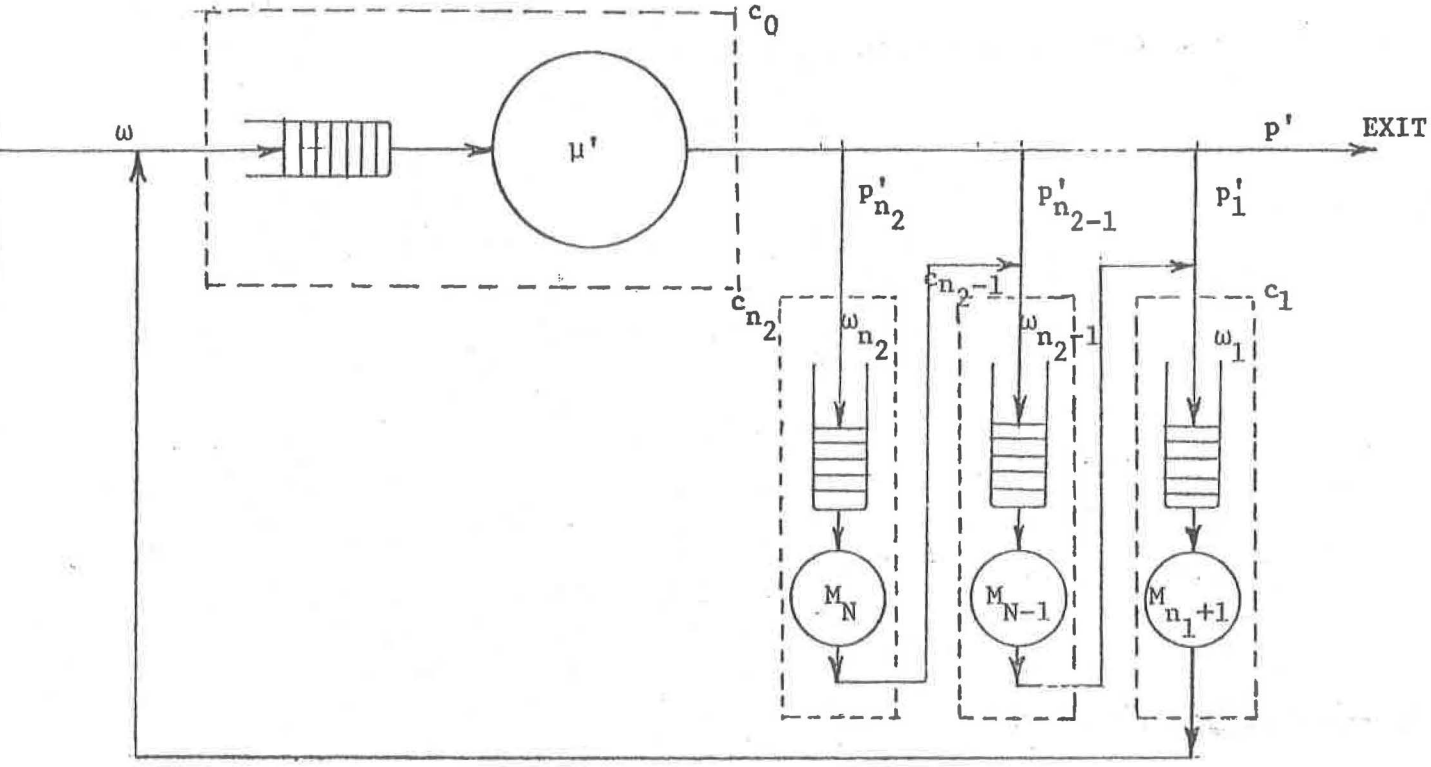

FIGURE 3: QUEUING NETWORK MODEL B OF SYSTEM

where $p_i' = p_i/P$ and the mean service time of centre c_0 is

$$\mu' = P\,K$$

$$\text{with} \quad 1/K = \sum_{i=1}^{n_1} q_i/\mu_i$$

$$\mu_i = 1/T_i \qquad i = 1, 2, \ldots, n_1 \tag{4}$$

Taking $\omega + \omega_1, \omega_1, \omega_2, \ldots, \omega_{n_2}$ to be the arrival rates of centres $c_0, c_1, \ldots, c_{n_2}$ respectively, the mean response time of this network is given by [6]

$$\bar{R} = \frac{1}{\omega}\left[\sum_{i=1}^{n_2} \frac{\omega_i\, y_{i+n_1}}{1 - \omega_i\, y_{i+n_1}} + \frac{(\omega + \omega_1)/\mu'}{1 - (\omega + \omega_1)/\mu'}\right] \tag{5}$$

$$= \frac{\sum_{i=1}^{n_1} x_{i-1}^{-\alpha}\, y_i}{1 - \sum_{i=1}^{n_1} \mu\, x_{i-1}^{-\alpha}\, y_i} + \sum_{i=n_1+1}^{n_1+n_2} \frac{x_i^{-\alpha}\, y_i}{1 - \mu\, x_i^{-\alpha}\, y_i} \tag{5'}$$

where $\mu = \omega/p$ (see Appendix 1 for details of derivation).

IV. OPTIMIZATION

Since the technology cost per unit information is given by

$$b(y) = y^{-\beta}$$

the system cost with storage sizes $x_1, x_2, \ldots, x_N$ for levels $M_1, M_2, \ldots, M_N$ with average transfer times $y_1, y_2, \ldots, y_N$ respectively, is given by

$$S = \sum_{i=1}^{N} x_i\, y_i^{-\beta} \tag{6}$$

Given N, x_N, and that the memory system cost is not to exceed S_0, the optimization problem becomes:

$$\text{Min} \quad \frac{\sum_{i=1}^{n_1} x_{i-1}^{-\alpha}\, y_i}{1 - \sum_{i=1}^{n_1} \mu\, x_{i-1}^{-\alpha}\, y_i} + \sum_{i=n_1+1}^{n_1+n_2} \frac{x_{i-1}^{-\alpha}\, y_i}{1 - \mu\, x_{i-1}^{-\alpha}\, y_i}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} x_i\, y_i^{-\beta} \le S_0 \quad ,$$

$$x_0 = 1 \; ; \; x_i > 0 \; ; \; y_i > 0 \qquad i = 1, 2, \ldots, N \qquad (7)$$

The problem (7) will have a solution only in the region where

$$\sum_{i=1}^{n_1} \mu\, x_{i-1}^{-\alpha}\, y_i < 1$$

$$\text{and} \quad \mu\, x_{i-1}^{-\alpha}\, y_i < 1 \qquad i = n_1+1, \ldots, n_1+n_2$$

This restriction meets one of the assumptions made while calculating equilibrium state probability, that the traffic intensity has to be strictly less than one [6].

Now by multiplying the objective function by μ (a constant) and adding $1 + n_2$ to it, the problem (7) reduces to

$$\text{Min} \quad \frac{1}{1 - \sum_{i=1}^{n_1} \mu\, x_{i-1}^{-\alpha}\, y_i} + \sum_{i=n_1}^{n_1+n_2} \frac{1}{1 - \mu\, x_{i-1}^{-\alpha}\, y_i}$$

$$\text{s.t.} \quad \frac{1}{S_0} \sum_{i=1}^{N} x_i\, y_i^{-\beta} \le 1 \qquad (8)$$

The natural constraints $x_i > 0$ and $y_i > 0$ can be ignored in the calculation by looking at a solution only in the positive region of x and y.

Introducing new variables r_0 and r_i's such that

$$r_0 \leq 1 - \sum_{i=1}^{n_1} \mu\, x_{i-1}^{-\alpha}\, y_i$$

and $$r_{i-n_1} \leq 1 - \mu\, x_{i-1}^{-\alpha}\, y_i \quad ; \quad i = n_1+1, \ldots, n_1+n_2$$

The problem (8) is equivalent to

$$\text{Min} \quad r_0^{-1} + \sum_{i=1}^{n_2} r_i^{-1}$$

$$\text{s.t.} \quad r_0 \leq 1 - \sum_{i=1}^{n_1} \mu\, x_{i-1}^{-\alpha}\, y_i$$

$$r_{i-n_1} \leq 1 - \mu\, x_{i-1}^{-\alpha}\, y_i \quad ; \quad i = n_1+1, \ldots, n_1+n_2$$

$$\text{and} \quad \frac{1}{S_0} \sum_{i=1}^{N} x_i\, y_i^{-\beta} < 1 \,. \tag{9}$$

We will use the standard Lagrange multiplier method together with geometric programming techniques to solve problem (9). The Lagrangian function for problem (9) is:

$$F(R,X,Y,\lambda) = r_0^{-1} + \sum_{i=1}^{n_2} r_i^{-1} + \lambda_0 \left(r_0 + \sum_{i=1}^{n_1} x_{i-1}^{-\alpha}\, y_i - 1\right)$$

$$+ \sum_{i=n_1+1}^{n_1+n_2} \lambda_{i-n_1} \left(r_{i-n_1} + \mu\, x_{i-1}^{-\alpha}\, y_i - 1\right)$$

$$+ \lambda' \left(\frac{1}{S_0} \sum_{i=1}^{N} x_i\, y_i^{-\beta} - 1\right) \tag{10}$$

Differentiating with respect to R, X, Y and λ respecting and equating to zero, we obtain,

$$\frac{\partial F}{\partial r_0} = (-1)\ r_0^{-1} + \lambda_0 r_0 = 0 \tag{11}$$

$$\frac{\partial F}{\partial r_i} = (-1)\ r_i^{-1} + \lambda_i r_i = 0 \quad ; \quad i = 1, 2, \ldots, n_2 \tag{12}$$

$$\frac{\partial F}{\partial x_i} = \begin{cases} (-\alpha)\ \mu\ \lambda_0\ x_i^{-\alpha}\ y_i + \dfrac{\lambda'}{S_0}\ x_{i-1}\ y_{i-1}^{-\beta} = 0 \quad ; \quad i = 1, 2, \ldots, n_1 & (13) \\ (-\alpha)\ \mu\ \lambda_{i-n_1}\ x_{i-1}^{-\alpha}\ y_i + \dfrac{\lambda'}{S_0}\ x_{i-1}\ y_{i-1}^{-\beta} = 0 \quad ; \quad i = n_1+1, \ldots, n_1+n_2 & (14) \end{cases}$$

$$\frac{\partial F}{\partial y_i} = \begin{cases} \mu\ \lambda_0\ x_{i-1}^{-\alpha}\ y_i + (-\beta)\ \dfrac{\lambda'}{S_0}\ x_i\ y_i^{-\beta} = 0 \quad ; \quad i = 1, 2, \ldots, n_1 & (15) \\ \mu\ \lambda_{i-n_1}\ x_{i-1}^{-\alpha}\ y_i + (-\beta)\ \dfrac{\lambda'}{S_0}\ x_i\ y_i^{-\beta} = 0 \quad ; \quad i = n_1+1, \ldots, n_1+n_2 & (16) \end{cases}$$

$$\frac{\partial F}{\partial \lambda_0} = r_0 + \sum_{i=1}^{n_1} \mu\ x_{i-1}^{-\alpha}\ y_i - 1 = 0 \tag{17}$$

$$\frac{\partial F}{\partial \lambda_{i-n_1}} = r_{i-n_1} + \mu\ x_{i-1}^{-\alpha}\ y_i - 1 = 0 \quad ; \quad i = n_1+1, \ldots, n_1+n_2 \tag{18}$$

$$\frac{\partial F}{\partial \lambda'} = \frac{1}{S_0} \sum_{1}^{N} x_i\ y_i^{-\beta} - 1 = 0 \tag{19}$$

Define $$f^0 = \lambda_0 \sum_{i=1}^{n_1} \mu\ x_{i-1}^{-\alpha}\ y_i + \sum_{n_1+1}^{n_1+n_2} \lambda_{i-n_1}\ \mu\ x_{i-1}^{-\alpha}\ y_i \tag{20}$$

$$\delta_0 = r_0^{-1} \tag{21}$$

$$\delta_i = r_i^{-1} \qquad i = 1, 2, \ldots, n_2 \tag{22}$$

$$\delta_{1i} = \begin{cases} \lambda_0\ \mu\ x_{i-1}^{-\alpha}\ y_i / f^0 & i = 1, 2, \ldots, n_1 \quad (23) \\ \lambda_{i-n_1}\ \mu\ x_{i-1}^{-\alpha}\ y_i / f^0 & i = n_1+1, \ldots, n_1+n_2 \quad (24) \end{cases}$$

$$\delta_{20} = \lambda_0 r_0 \tag{25}$$

$$\delta_{2i} = \lambda_i r_i \qquad i = 1, 2, \ldots, n_2 \tag{26}$$

$$\delta_{3i} = \frac{\lambda'}{S_0 f^0} x_i \, y_i^{-\beta} \qquad i = 1, 2, \ldots, n_1+n_2 \tag{27}$$

Now clearly,

$$\sum_{i=1}^{n_1+n_2} \delta_{1i} = 1 \qquad \text{(normality)} \tag{28}$$

$$(11) \Rightarrow (-1)\ \delta_0 + \delta_{20} = 0 \tag{29}$$

$$(12) \Rightarrow (-1)\ \delta_i + \delta_{2i} = 0 \ ; \ i = 1, 2, \ldots, n_2 \tag{30}$$

$$(13) \ \& \ (14) \Rightarrow (-\alpha)\ \delta_{1i} + \delta_{3i-1} = 0 \ ; \ i = 2, \ldots, n_1+n_2 \tag{31}$$

$$(15) \ \& \ (16) \Rightarrow \delta_{1i} + (-\beta)\ \delta_{3i} = 0 \ ; \ i = 1, 2, \ldots, n_1 + n_2 \tag{32}$$

Solving (28), (31) and (32) simultaneously, we obtain

$$\delta_{1i} = (\alpha\beta)^{N-i} \frac{\alpha\beta - 1}{(\alpha\beta)^N - 1}$$

$$\delta_{3i} = \frac{1}{\beta} \delta_{1i}$$

$$N = n_1 + n_2 \ ; \ i = 1, 2, \ldots, n_1 + n_2 \tag{33}$$

Now in order to obtain optimal values for x_i's and y_i's we first obtain values for f^0 and λ_i's.

Raising (23) and (24) to the power δ_{1i} and (27) to the power δ_{3i} for $i = 1, 2, \ldots, n_1 + n_2$ respectively, and then multiplying we obtain

$$(f^0)^{\sum_1^{n_1+n_2} \delta_{1i}} = f^0 = \lambda_0^{a_{n_1}} \prod_{i=1}^{n_2} \lambda_i^{b_i} \cdot c \tag{34}$$

where $a_{n_1} = \sum_{i=1}^{n_1} \delta_{1i}$

$b_i = \delta_{1n_1+i}$, $i = 1, 2, \ldots, n_2$

$c = \mu(\frac{1}{\beta S_0})^{1/\beta} \cdot x_N \cdot \prod_{i=1}^{n_1+n_2} (\frac{1}{\delta_{1i}})^{\delta_{1i}} (\frac{1}{\delta_{3i}})^{\delta_{3i}}$

$\frac{\lambda'}{f^0} = \frac{1}{\beta}$

from (17), (21), (23), (25) and (29)

$$\lambda_0 = (2f^0 a_{n_1} + \sqrt{4f^0 a_{n_1} + 1})/2 \tag{35}$$

and from (18), (22), (24), (26) and (30)

$$\lambda_i = (2f^0 b_i + 1 + \sqrt{4f^0 b_i + 1})/2 \tag{36}$$

Substituting (35) and (36) into (34)

$$f^{0*} = c \cdot (2f^0 a_{n_1} + 1 + \sqrt{4f^0 a_{n_1} + 1})^{a_{n_1}} \cdot \prod_{i=1}^{n_2} (2f^0 b_i + 1 + \sqrt{4f^0 b_i + 1})^{b_i}/2 \tag{37}$$

Substituting the value of f^{0*} in (35) and (36) we obtain values for the λ_i's.

From equations (23), (24), (27), (31), and (32), with some algebraic manipulation, we obtain the following expression for optimal x_i's and y_i's

$$\log x_i = \left[\frac{i}{1-a} - \frac{(1-a^i)N}{(1-a)(1-a^N)}\right] \cdot K + \frac{1-a^i}{(1-a^N)} \cdot \log x_N$$

$$+ \frac{1-a^i}{(1-a)} \cdot \frac{L - M}{(1-a^N)} \qquad i = 1, 2, \ldots, n_1-1 \qquad (38a)$$

$$\log x_i = \left[\frac{i}{1-a} - \frac{(1-a^i)N}{(1-a)(1-a^N)}\right] \cdot K + \frac{1-a^i}{(1-a^N)} \cdot \log x_N$$

$$+ \frac{1-a^i}{1-a}\left[\frac{L-M}{1-a^N}\right] + \sum_{j=0}^{i-n_1} h_j\, a^{i-j} \qquad i = n_1, \ldots, n_1+n_2-1 \qquad (38b)$$

where $a = \alpha\beta$

$K = -(\beta + 1) \log a$

$h_i = \beta(\log \lambda_i - \log \lambda_{i+1})$

$L = \sum_{i=0}^{n_2-1} h_i$

$M = \sum_{i=0}^{n_2-1} h_i\, a^{n_2-i}$

Also from (27) and (33), we have

$$y_i = (x_i/S_0 \cdot \delta_{1i})^{1/\beta} \qquad (39)$$

Hence knowing all x_i's, all y_i's can be found from equation (39). In the case of uniprogramming, $\mu = 0$, all λ_i's are equal and therefore $h_i = L = M = 0$. Equations (38a) and (38b) then reduce to

$$\log x_i = \left[\frac{i}{1-a} - \frac{(1-a^i)N}{(1-a)(1-a^N)}\right] \cdot K + \frac{1-a^i}{1-a^N} \log x_N \qquad i = 1, 2, \ldots, N \qquad (40)$$

This is the same result obtained by Chow [2]. The following example illustrates how the optimal solutions for a multiprogrammed environment approach the ones for a uniprogrammed environment as μ tends to zero.

Example

For the case $N = 4$, $n_1 = n_2 = 2$, and $\alpha = \beta = 1$, the expression for optimal x_i's are

$$x_1 = x_4^{1/4} \left(\frac{\lambda_0}{\lambda_1}\right)^{1/2}$$

$$x_2 = x_4^{1/2} \left(\frac{\lambda_0}{\lambda_1}\right)^{2}$$

$$x_3 = x_4^{3/4} \left(\frac{\lambda_0}{\lambda_1}\right)^{5/2}$$

Taking $x_4 = 10^8$ and $S_0 = 4 * 10^{10}$, (actual units depend upon the normalization factors used in Equations (2) and (3)) the values of x_i's are given in Table 1 for different values of μ.

Table 1

Optimal storage size for $N = 4$, $x_4 = 10^8$, $S_0 = 4*10^{10}$, $\alpha = \beta = 1$

μ	x_1	x_2	x_3
.100	100.08	10031	1003905
.1000	100.77	10313	1039305
10000	107.24	13224	1418069
0	100	10000	1000000

The values of x's for $\mu = 0$ are the ones obtained by Chow for a uniprogrammed system (the values are obtained using equation (40)). We observe that there is a significant difference when $\mu(= w/p)$ is large (i.e., when the arrival rate of requests is large and/or the exit probability of a process is small which implies a large number of active processes competing for limited systems resources simultaneously). Also from Equations (38a), (38b) and (39), the ratio between the sizes and speeds of successive memory levels of the same type are constant, as observed by Chow and reported in [9].

CONCLUSION

We have presented a methodology for estimating the optimal capacities and speeds of the memory levels in a memory hierarchy operating in the multiprogrammed environment. Optimality is with respect to mean system response time under a fixed cost constraint. Mean response time is the single performance parameter most important to users of interactive systems which are invariably multiprogrammed. It is therefore more meaningful in such an environment to use mean response time rather than mean memory access time as the criterion for optimization. Queuing theory allows us to analyze multiprogrammed systems and global performance indices such as mean response time while optimization techniques enable us to compute the optimal values. This model can be further refined by relaxing certain assumptions made in this paper at the expense of mathematical tractability. For example by assuming α to be constant for all jobs (or requests), we are assuming that they have similar memory access characteristics. This assumption can be removed by taking α_{ij} as the hit-ratio constant for the i^{th} job in the j^{th} memory level.

Since an explicit closed form solution for f^0 does not seem to exist, we are unable to obtain optimal values for n_1 and n_2 as has been done by Chow [2] for uniprogrammed systems.

ACKNOWLEDGEMENTS

This work was supported in part by the National Research Council of Canada under Grant No. A3554 and by UBC Summer Session Research Scholarship 4904.

REFERENCE

[1] C. V. Ramamoorthy and K. M. Chandy, "Optimization of memory hierarchies in multiprogramming system." J. Assoc. Comput. Mach., Vol. 17, pp. 426-445, July, 1970.

[2] C. K. Chow, "On optimization of storage hierarchies." IBM J. Res. Develop., Vol. 18, pp. 194-203, May, 1974.

[3] J. E. MacDonald and K. L. Sigworth, "Storage hierarchy optimization procedure." IBM J. Res. Develop., Vol. 19, pp. 133-140, March, 1975.

[4] T. A. Welch, "Memory hierarchy configuration analysis." IEEE Trans. on computers, Vol. C-27, No. 3, pp. 408-413, May, 1978.

[5] R. L. Maltson, "Evaluation of multilevel memories." IEEE Trans. Magnetics, Vol. MAG-7, pp. 814-819, December, 1971.

[6] D. Ferrari, Computer system performance evaluation. Prentice-Hall, Englewood Cliffs, New Jersey, 1978.

[7] R. J. Duffin, E. L. Peterson and C. Zener, Geometric Programming. John Wiley & Sons Inc., New York, 1967.

[8] S. L. Rege, "Cost, performance and size trade-off for different levels in memory hierarchy." Computer, Vol. 19, pp. 43-51, April, 1976.

[9] Y. S. Lin and R. L. Mattson, "Cost-performance evaluation of memory hierarchies." IEEE Trans. Magnetics MAG-8, pp. 390-392, Sept. 1972.

Appendix 1

The arrival rates for service centres are

$$\omega_{n_2} = (\omega + \omega_1)\, p'_{n_2} ,$$

$$\omega_{n_{2-1}} = (\omega + \omega_1)\, p'_{n_{2-1}} + \omega_{n_2}$$

$$= (\omega + \omega_1)(p'_{n_{2-1}} + p'_{n_2}) ,$$

$$\vdots$$

$$\omega_i = (\omega + \omega_1) \sum_{j=i}^{n_2} p'_j , \qquad \text{(A1.1)}$$

$$\vdots$$

$$\Rightarrow \quad \omega_1 = (\omega + \omega_1) \sum_{j=1}^{n_2} p'_j$$

Let $P' = \sum_{j=1}^{n_2} p'_j$ and

$$Q' = 1 - P' ,$$

then $\omega_1 = \frac{P'}{Q'}\, \omega$.

Now $$\omega + \omega_1 = \omega + \frac{P'}{Q'}\, \omega = \frac{\omega}{Q'} \qquad \text{(A1.2)}$$

$\therefore$ from (A1.I) $\omega_i = \frac{\omega}{Q'} \left[\sum_{j=i}^{n_2} p'_j \right]$,

or $$\omega_i = \frac{\omega}{P} \sum_{j=i}^{n_2} p_j . \qquad \text{(A1.3)}$$

because $Q' = 1 - P'$

$$= 1 - \sum_{1}^{n_2} p'_j$$

$$= 1 - \frac{1}{P} \left(\sum_{1}^{n_2} p_j \right)$$

$$= 1 - \frac{1}{P} [P - p]$$

$$= \frac{p}{P}$$

From Equation (1)

$$p_j = H(x_U) - H(x_{U-1}) \qquad U = j + n_1 ,$$

$$\therefore \quad \sum_{j=1}^{n_2} p_j = \sum_{j=1}^{n_1+n_2} [H(x_j) - H(x_{j-1})]$$

(taking $H(x_N) = 1$)

$$= 1 - H(x_{i-1})$$

$$= F(x_{i-1})$$

$$= x_{i-1}^{-\alpha}$$

substituting in (A1.3) we get

$$\omega_i = \frac{\omega}{p} x_{i-1}^{-\alpha}$$

$$= \mu \, x_{i-1}^{-\alpha} \qquad \text{(A1.4)}$$

where $\mu = \frac{\omega}{p}$

Also $$(\omega_1 + \omega)/\mu' = \frac{\omega}{Q'\mu'}$$

$$= \frac{\omega}{Q'} \left[\frac{1}{P} \sum_{i=1}^{n_1} q_i \, T_i\right]$$

$$= \frac{\omega}{p} \sum_{i=1}^{n_1} q_i \sum_{j=1}^{i} y_i$$

substituting $$q_i = H(x_i) - H(x_{i-1})$$

and $$F_i(x_i) = x_i^{-\alpha} = 1 - H(x_i) \ ,$$

we have $$(\omega_1 + \omega)/\mu' = \frac{\omega}{p} \sum_{1}^{n_1} x_{i-1}^{-\alpha} \, y_i$$

$$= \mu \sum_{1}^{n_1} x_{i-1}^{-\alpha} \, y_i \qquad \text{(A1.5)}$$

Substituting (A1.4) and (A1.5) into Equation (5) we obtain Equation (5').