C. K. Chow

On Optimization of Storage Hierarchies

Abstract: A simple model of the storage hierarchies is formulated with the assumptions that the effect of the storage management strategy is characterized by the hit ratio function. The hit ratio function and the device technology-cost function are assumed to be representable by power functions (or piece-wise power functions). The optimization of this model is a geometric programming problem. An explicit formula for the minimum hierarchy access time is derived; the capacity and technology of each storage level are determined. The optimal number of storage levels in a hierarchy is shown to be directly proportional to the logarithm of the systems capacity with the constant of proportionality dependent upon the technology and hit ratio characteristics. The optimal cost ratio of adjacent storage levels is constant, as are the ratios of the device access times and storage capacities of the adjacent levels. An illustration of the effect of overhead cost and level-dependent cost, such as the cost per "box" and cost for managing memory faults is given and several generalizations are presented.

Introduction

The general trend in the development of large computer systems is toward increasing the use of storage hierarchies. A linear storage hierarchy model consists of n levels, $M_1, M_2, \cdots, M_n$, connected in cascade as shown in Fig. 1. The convention is that the higher is the level the lower is its index. Generally, the higher is the level, the faster is its speed, the higher is its cost per byte, and the smaller is its capacity. Information transfers are between adjacent levels and are entirely controlled by the activity in the first level M_1. The rules of operation are

1. Whenever a page is stored in level M_i, there is a copy of it in each of the lower levels, $M_{i+1}, \cdots, M_n$.
2. Whenever a referenced page is not found in M_1, a request for it is sent to the successive lower levels until it is found in the say M_i level.
3. Whenever M_i is full and a new page is to be brought in from M_{i+1}, a replacement policy, usually the Least Recently Used (LRU) policy, is invoked to select a page to be deleted from M_i (since there is already a copy in M_{i+1}, there is no need to move the displaced page into M_{i+1}) [1].

The principal advantage of this storage organization is that a program's Working Set accumulates rapidly in the fastest level M_1, thus, accesses are completed at nearly the speed of M_1, but the total cost of the storage system approaches that of the lowest level. A second advantage is that the mechanism can be readily implemented, requiring very little operating system intervention [1].

The most notable examples are the cache memory [2] on the IBM System 360 Model 85 and Model 195. These systems use three levels ($n = 3$); a seven-level system is illustrated in a book by Lorin [3].

Several papers [4-8] describe some of the techniques used for cost-performance evaluation of storage hierarchies. These papers are concerned with storage hierarchies of two or three levels. Typically, their algorithms evaluate, for a given hierarchy configuration (given number of storage levels, device characteristics, etc.), its cost-performance in terms of the total system cost per memory access for different capacities and page sizes at each level, and select the configuration with the lowest cost per access.

These studies and numerical results have prompted some fundamental questions: How is the system performance affected by the cost and the required capacity? What is the minimum hierarchy access time? How should the cost be allocated to each storage level? What are the optimum capacity and technology of each level? What is the optimum number of the levels in the hierarchy?

To provide some answers to these questions, numerical computations and simulations, important as they are, are not sufficient. They must be supplemented by analysis, and the functional relations among key system parameters are called for. Unfortunately, hierarchical storage systems are complex and difficult for mathematical analysis, and few theoretical results of general nature are available. If one starts with an all-inclusive model of the system, it is doubtful that analytic solutions can be

achieved. We therefore begin with bare essentials and formulate the problem in a mathematically tractable yet, hopefully practical, and meaningful way.

A simple model is formulated with the following assumptions: 1) The effect of the storage management strategy is characterized by the memory hit ratio function. 2) The device technology is specified by the device access time and the cost per unit of storage (byte or other unit). 3) The hit ratio function and the device technology-cost function are representable by power functions (or piece-wise power functions). Under these assumptions the optimization of such a system becomes a geometric programming problem.

An explicit formula for the minimum hierarchy access time [Eqs. (31) and (32)] is obtained as a function of the hierarchy cost and the required capacity. The capacity and device technology are determined for each storage level [Eqs. (43), (44), (47) and (48)]. The optimal cost ratio of adjacent levels is constant (39) and, for an optimal configuration the ratios between device access times and of adjacent levels, as well as the ratios between their capacities, are also constant. [Eqs. (45) and (46)].

The optimal number of storage levels in a hierarchy is shown to be directly proportional to the logarithm of the system's capacity, with the constant of proportionality dependent upon the technology and hit ratio characteristic powers [Eqs. (57) and (60)].

Model and assumptions

The hierarchical storage system under consideration is shown in Fig. 1 with the operational rules described in the Introduction. It is a linear hierarchy of n levels, M_1, $M_2, \cdots, M_n$. The two major factors that determine the performance of a storage hierarchy are the storage device characteristics of each level and the storage management strategy. Since copies of the information stored in the higher levels are found in all lower levels, the system storage capacity, namely the maximum amount of information that can be stored in the system, is equal to the capacity of the lowest level. In our analysis the following assumptions are made:

A1. Each storage level, M_i, is characterized by its access time t_i and capacity C_i.

A2. The storage management strategy is completely characterized by the success function or hit ratio H, which is a function of the storage capacity C.

A3. The device technologies are characterized by their cost function $b(t)$, which represents either purchase or rental, per storage unit (say byte) of the technology giving the access time t.

The cost function is always a monotonic decreasing function in access time; a sample curve is shown in Fig. 2.

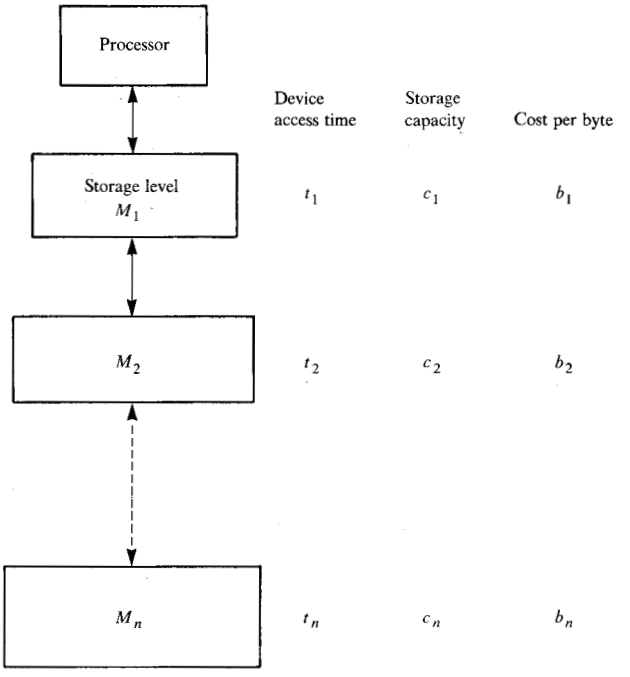

Figure 1 Schematic diagram of a linear storage hierarchy model.

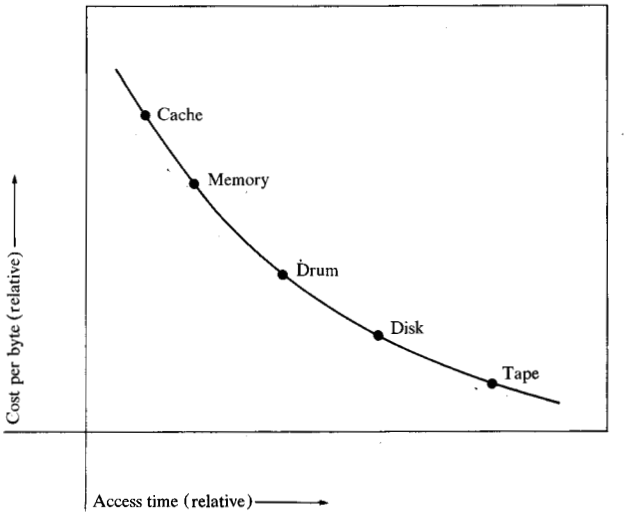

Figure 2 Relative cost-access time characteristic.

The access time includes device cycle time, transfer time, and waiting time.

The success function H generally depends not only on the storage capacity C, but also on the block size, the management algorithm, and other factors. For any reasonable choice of the algorithm, however, such as LRU, H is most sensitive to the capacity, which is the most

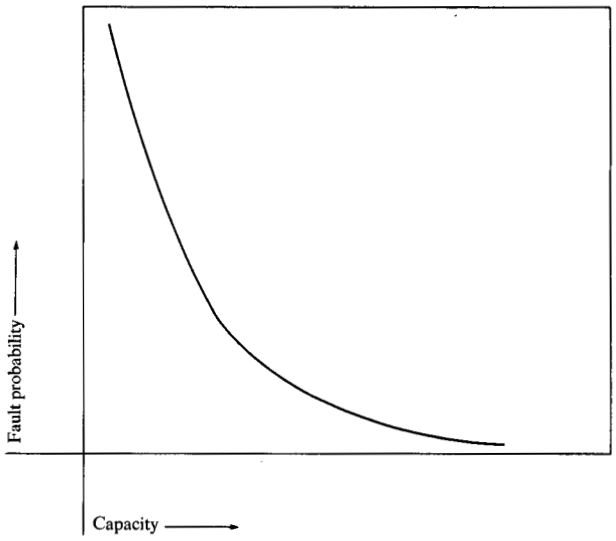

Figure 3 Qualitative representation of a miss ratio curve.

critical parameter [1]. Therefore, in the present analysis, the success function H is assumed to be a function of C only, and it is always monotone increasing in C.

The success function, $H(C)$, is the probability of finding the requested information for each memory reference to a memory of capacity C. Since a copy of all information in level i exists in every level greater than i, the probability of a hit at M_i and misses in the higher levels M_1, $M_2, \cdots, M_{i-1}$, is h_i, namely,

$$h_i = H(C_i) - H(C_{i-1}), \tag{1}$$

which is the relative number of successful accesses to level i, and is sometimes referred to as the access frequency. The miss ratio or fault probability is defined as

$$F(C) = 1 - H(C). \tag{2}$$

A sample of $F(C)$ is given in Fig. 3.

Let T_i be the effective hierarchy access time to the ith level M_i, i.e., T_i is the sum of individual access times of each level up to M_i, namely,

$$T_i = \sum_{j=1}^{i} t_j. \tag{3}$$

In our analysis, because of the particular choice of optimization criterion, Eq. (3) is equivalent to a more general form,

$$T_i = T_0 + K \sum_{j=1}^{i} t_j, \tag{4}$$

where T_0 and K are constants with $K > 0$. The effective or average hierarchy access time per each memory reference is, therefore,

$$T = \sum_{i=1}^{n} h_i T_i. \tag{5}$$

Substitution of Eqs. (1) and (3) into (5) gives

$$T = H(C_n)t_1 + \sum_{i=2}^{n} [H(C_n) - H(C_{i-1})]t_i. \tag{6}$$

It is a convention to define $H(C_n) = 1$, which amounts to assuming that all requested information is stored in the lowest level M_n (the largest store). For notational simplicity we define $H(C_0) = 0$. With these definitions $F(C_0) = 1$ and Eq. (6) becomes

$$T = t_1 + \sum_{i=2}^{n} [1 - H(C_{i-1})]t_i = \sum_{i=1}^{n} [1 - H(C_{i-1})]t_i$$

$$= \sum_{i=1}^{n} F(C_{i-1})t_i. \tag{7}$$

The total cost of the storage system is

$$S = \sum_{i=1}^{n} b(t_i)C_i, \tag{8}$$

which, in our analysis, is equivalent to a more general form of

$$COST = FCT(S), \tag{9}$$

where FCT is any monotonic function of S. The equivalences between Eqs. (3) and (4) and between (8) and (9) do not hold in Refs. [6, 7 and 8] because their measure of cost-performance is the total system cost per reference, namely, the product of S and T.

Optimization problem

Our criterion for optimization is to minimize the effective hierarchy access time T subject to the storage system cost and capacity constraints or, equivalently, to minimize the system cost subject to the constraints on T and the capacities. The problem is as follows:

Given:

- storage capacity C_n
- system cost S_0
- success function $H(C)$ or fault probability function $F(C)$
- technology cost function $b(t)$
- number of levels, n

Variables:

- $t_1, t_2, \cdots, t_n$
- $C_1, C_2, \cdots, C_{n-1}$

Minimize:

$$T = \sum_{i=1}^{n} F(C_{i-1})t_i. \tag{7}$$

Subject to the constraints:

$$S = \sum_{i=1}^{n} b(t_i)C_i \leq S_0; \tag{10}$$

$t_i > 0$ for $i = 1, 2, \cdots, n$ and (11)

$C_i > 0$ for $i = 1, 2, \cdots, n-1$. (12)

Since the functions $F(C)$ and $b(t)$ are generally not linear, the minimization is a problem in nonlinear programming. No general solution is known, so to obtain an explicit solution, we assume power functions for F and b. The number of levels, n, is considered to be, later on, an unknown variable, and its optimal value is determined by optimization techniques. First we derive some general relations by the method of Lagrange multipliers. Form the Lagrangian function

$$L = T + \lambda(S - S_0) \tag{13}$$

At the optimum values the partial derivatives of L with respect to all t_i's, C_i's and λ vanish. (For a more rigorous approach Kuhn-Tucker conditions [9] should be used.) As a result,

$$S = S_0, \tag{14a}$$

$$\lambda = -\frac{F'(C_i)t_{i+1}}{b(t_i)} \text{ for } i = 1, 2, \cdots, n-1, \tag{14b}$$

$$\lambda = -\frac{F(C_{i-1})}{C_i b'(t_i)} \text{ for } i = 1, 2, \cdots, n. \tag{14c}$$

Both derivatives $H'(C)$ and $b'(t)$ are negative. Eq. (14) is a system of $2n$ equations, solvable at least in principle, for the $2n$ variables $t_1, t_2, \cdots, t_n$; $C_1, C_2, \cdots, C_{n-1}$; and the multiplier λ. Some immediate observations can be made from Eq. (14):

1. The cost constraint is active (as expected).
2. The capacity or size C_i of the ith level M_i is directly proportional to the miss ratio of M_{i-1} and inversely proportional to the cost slope of the technology at M_i.
3. The access time, t_i, of M_i is proportional to the byte cost of M_{i-1} and inversely proportional to the slope of the miss ratio at M_{i-1}.
4. The cost of M_i is

   $$b(t_i)C_i = -F(C_{i-1})/\lambda \frac{d}{dt} \ln b(t) \mid_{t=t_i}.$$

   The higher is the miss ratio at M_{i-1}, the more money should be spent for M_i.
5. The average period of time for each memory reference when M_i is active, i.e., the contribution of M_i to the hierarchy access time T, is

   $$F(C_{i-1})t_i = -\lambda b(t_{i-1})/\frac{d}{dC} \ln F(C)|_{C=C_{i-1}}.$$

   The higher the cost of M_{i-1} per byte, the longer will be the period of activity for M_i.

Power functions

We assume that the fault probability $F(C)$ and technology cost function $b(t)$ are power functions; specifically:

A4. The fault probability is

$$F(C) = F_0 C^{-\alpha} \text{ for } F_0^{1/\alpha} \le C < C_n, \tag{15}$$

where F_0 and α are positive constants. Without loss of generality, we take the constant $F_0 = 1$, i.e.,

$$F(C) = C^{-\alpha}. \tag{15'}$$

This choice of F_0 amounts to using $F_0^{1/\alpha}$ as the unit for the storage capacity since Eq. (15) can be written as $F = (C/F_0^{1/\alpha})^{-\alpha}$. As seen, Eq. (7) does not contain $F(C_n)$; the required range for the validity of the power function is only up to and including C_{n-1}.

A5. The cost per unit capacity or byte is

$$b(t) = b_0 t^{-\beta} \text{ for } t > 0, \tag{16}$$

where b_0 and β are positive constants. Again, without loss of generality, we take $b_0 = 1$ (i.e., b_0 is the unit for cost):

$$b(t) = t^{-\beta}. \tag{16'}$$

These assumptions may appear restrictive; however, at the present level of analysis, and due to scarcity of data, they enable us to gain some insight and to determine the sensitivity of certain parameters in a storage hierarchy model. Furthermore, these functionals agree reasonably well with empirical data; Eq. (16) is used in [8] with $0.2 \le \beta \le 0.6$, and the power function is not too different from the empirical hit ratio data of Mattson [10]. Some empirical data show leveling off in miss ratio beyond certain capacities. This phenomenon can be accounted for by using different values of α for different levels (see later generalizations).

With these assumptions the expressions for the effective hierarchy access time and the cost constraint become

$$T = t_1 + \sum_{i=2}^{n} C_{i-1}^{-\alpha} t_i; \tag{17}$$

$$S/S_0 = \frac{1}{S_0} \sum_{i=1}^{n} t_i^{-\beta} C_i \le 1. \tag{18}$$

The system of equations (14) can now be solved explicity; however, the intermediate results are quite complex. Instead of solving (14) directly, we use a theory in geometric programming to obtain an expression for the minimum hierarchy access time without having first to solve for the C_i's and t_i's. With the positivity conditions of Eqs. (11) and (12), both the objective function

(17) and the constraint (18) are posynomials in the C_i's and t_i's. The optimization problem as formulated here is a standard geometric programming problem.

Constrained minimum

By the theory of geometric programming, the constrained minimum of the effective hierarchy access time is

$$T^* = \lambda^{\lambda} C_n^{\delta_{2n}} S_0^{-\lambda} \prod_{i=1}^{2n} \left(\frac{1}{\delta_i}\right)^{\delta_i}, \tag{19}$$

where

$$\lambda = \sum_{i=n+1}^{2n} \delta_i = \sum_{i=1}^{n} \delta_{n+i} \tag{20}$$

and the vector variable $\boldsymbol{\delta} = (\delta_1, \delta_2, \cdots, \delta_{2n})$ is subject to linear constraints, namely:

Positivity:

$$\delta_i \geq 0 \text{ for } i = 1, 2, \cdots, 2n. \tag{21}$$

Normalization:

$$\sum_{i=1}^{n} \delta_i = 1. \tag{22}$$

Orthogonality:

$$\boldsymbol{\delta}\mathbf{A} = 0, \tag{23}$$

where $\mathbf{A}$ is a $2n \times (2n-1)$ matrix of exponents obtained from Eqs. (17) and (18) as

$$\mathbf{A} = [a_{ij}] = \begin{array}{c|c|c} & n & n-1 \\ \hline n & \mathbf{A}_{11} & \mathbf{A}_{12} \\ \hline n & \mathbf{A}_{21} & \mathbf{A}_{22} \end{array} \tag{24}$$

$$= \begin{array}{c|c|c} & n & n-1 \\ \hline n & \mathbf{I}_n & \begin{matrix} 0 & & & & \\ -\alpha & 0 & & 0 & \\ & -\alpha & \ddots & & \\ & & \ddots & 0 & \\ & 0 & & -\alpha & 0 \end{matrix} \quad n-1 \\ \hline 1 & \multicolumn{2}{c}{0\ 0\ \bullet\ \bullet\ \bullet\ 0\ -\alpha} \\ \hline n & -\beta\mathbf{I}_n & \mathbf{I}_{n-1} \quad n-1 \\ \hline 1 & \multicolumn{2}{c}{0} \end{array}.$$

The submatrices $\mathbf{A}_{11}$ and $\mathbf{A}_{21}$ are $n \times n$, while $\mathbf{A}_{12}$ and $\mathbf{A}_{22}$ are $n \times (n-1)$. $\mathbf{A}_{11}$ and $\mathbf{A}_{21}$ come from Eq. (17) and $\mathbf{A}_{21}$ and $\mathbf{A}_{22}$ from (18). $\mathbf{A}_{11}$ is the $n \times n$ identity matrix $\mathbf{I}_n$; $\mathbf{A}_{21} = -\beta\mathbf{I}_n$; $\mathbf{A}_{12}$ has all zeros except the diagonal just below the main diagonal, where all elements are $-\alpha$; the top $n-1$ rows of $\mathbf{A}_{22}$ form an $(n-1) \times (n-1)$ identity matrix and the bottom row consists of all zeros.

With the definition of (24), Eq. (23) gives

$$\delta_i = \beta\delta_{n+i} \text{ for } i = 1, 2, \cdots, n; \tag{25}$$

$$\delta_{n+i} = \alpha\delta_{i+1} \text{ for } i = 1, 2, \cdots, n-1. \tag{26}$$

With the aid of Eqs. (22) and (25), (20) yields

$$\lambda = \beta^{-1} \sum_{i=1}^{n} \delta_i = \beta^{-1}. \tag{27}$$

The unique solution of the $2n$ equations of (22), (25) and (26) is obtained as

$$\delta_n = \begin{cases} \dfrac{1}{n} & \text{if } \alpha\beta = 1; \\ \dfrac{\alpha\beta - 1}{(\alpha\beta)^n - 1} & \text{if } \alpha\beta \neq 1; \end{cases} \tag{28}$$

$$\delta_i = \alpha\beta\delta_{i+1} = (\alpha\beta)^{n-i}\,\delta_n \text{ for } i = 1, 2, \cdots, n-1 \text{ and} \tag{29}$$

$$\delta_{n+i} = \beta^{-1}\delta_i = \beta^{-1}(\alpha\beta)^{n-i}\,\delta_n \text{ for } i = 1, 2, \cdots, n. \tag{30}$$

Note that λ and δ_i's are all positive.

By substituting Eqs. (28)-(30) into (19); and after some (tedious) algebraic manipulations, the minimum hierarchy access time is obtained as follows.

When $\alpha\beta = 1$,

$$T^* = S_0^{-1/\beta} \cdot C_n^{1/\beta n} \cdot n^{1+1/\beta} = S_0^{-\alpha} \cdot C_n^{\alpha/n} \cdot n^{1+\alpha}; \tag{31}$$

when $\alpha\beta \neq 1$,

$$T^* = S_0^{-1/\beta} \cdot C_n^{\delta_n/\beta} \cdot T_g(n, \alpha, \beta), \tag{32}$$

where

$$T_g(n, \alpha, \beta) = \left[\frac{(\alpha\beta)^{\gamma}}{\delta_n}\right]^{1+1/\beta} \tag{33}$$

and

$$\gamma = \sum_{i=1}^{n} i\delta_i - n. \tag{34}$$

The summation yields

$$\sum_{i=1}^{n} i\delta_i = \frac{\alpha\beta}{\alpha\beta - 1} - \frac{n}{(\alpha\beta)^n - 1}. \tag{34a}$$

Hence

$$\gamma = \frac{\alpha\beta}{\alpha\beta - 1} - \frac{n(\alpha\beta)^n}{(\alpha\beta)^n - 1}. \tag{34'}$$

Therefore

$$T_g(n, \alpha, \beta) = (\alpha\beta)^{\eta}/\delta_n^{1+1/\beta} \tag{33'}$$

where

$$\eta = \gamma(1+\beta^{-1}) = (1+\beta^{-1})\left(\frac{\alpha\beta}{\alpha\beta-1} - \frac{n(\alpha\beta)^n}{(\alpha\beta)^n-1}\right). \tag{34b}$$

The minimum value is a product of three factors; the system cost and the system capacity enter into only one (separate) factor each. As intuitively expected, the optimum access time decreases with increasing system cost and increases with increasing capacity. The logarithmic rate of decrease with respect to the cost is determined by the technology cost slope β, and the rate of increase with respect to the capacity is determined by both technology and the hit ratio characteristics as well as the number of levels. If the technology cost curve is steep (i.e., if β is large), the effect of the hierarchy cost in reducing the access time is lessened. The third factor depends only on α, β and n and is independent of S_0 and C_n.

Cost allocation and access time distribution

Here we examine the allocation of the total hierarchy cost to each storage level and the contribution of each level to the effective hierarchy access time. Returning to Eqs. (17) and (18), $t_i^{-\beta}C_i$ is the cost allocated to M_i and $c_{i-1}^{-\alpha}t_i$ is the activity time of M_i. The total cost S_0 and hierarchy access time are the respective sums of these terms over all levels.

The ratio between the cost of M_i and the total hierarchy cost, is a_i, i.e.,

$$a_i = t_i^{-\beta}\, C_i/S_0 \quad \text{for } i = 1, 2, \cdots, n. \tag{35}$$

The ratio between activity time of M_i and the effective hierarchy access time is d_i, i.e.,

$$d_i = C_{i-1}^{-\alpha}\, t_i/T, \quad \text{for } i = 1, 2, \cdots, n \tag{36}$$

with the convention that $C_0 = 1$.

At the optimizing point, it can be shown [12] that for $i = 1, 2, \cdots, n$,

$$a_i = t_i^{-\beta}C_i/S_0 = \delta_{n+i}/\lambda$$

and

$$d_i = C_{i-1}^{-\alpha}t_i/T^* = \delta_i, \tag{37}$$

where T^* is the optimal access time of the hierarchy. Substitution of (25)-(30) into (37) yields

$$a_i = d_i = \delta_i = (\alpha\beta)^{n-i}\,\delta_n = (\alpha\beta)^{n-i}\,\frac{\alpha\beta-1}{(\alpha\beta)^n-1}. \tag{38}$$

It also follows that

$$\frac{a_i}{a_{i+1}} = \frac{d_i}{d_{i+1}} = \alpha\beta. \tag{39}$$

In other words, the fractional cost allocations are equal to the relative activity time distributions; the more costly storage level contributes a larger fraction to the optimum hierarchy access time T^* since it is more active. The cost (or activity time) ratio of two adjacent storage levels is simply $\alpha\beta$. If $\alpha\beta < 1$, the greater cost should be allocated to the lower level (i.e., the slower and larger memory level). If $\alpha\beta = 1$, all levels are of equal cost. If $\alpha\beta > 1$, then the higher level (i.e., faster and smaller) should get the larger fraction of cost. For example, if $\alpha\beta = \frac{1}{2}$, then the costs of M_1, M_2, M_3 and M_4 should be in the ratio 1:2:4:8.

Determination of minimizing parameters

The constrained minimum of the hierarchy access time and the associated variables δ_i's are obtained without explicitly solving for the minimizing parameters, namely $C_1, C_2, \cdots, C_{n-1}$ and $t_1, t_2, \cdots, t_n$ which characterize all storage levels. We show that these minimizing parameters can be obtained by solving a set of $2n-1$ simultaneous *linear* equations. (They can also be obtained from Eqs. (14b) and (14c) derived from the Langangian function.)

In the system of equations as given by Eq. (37), the unknown variables are $C_1, C_2, \cdots, C_{n-1}, t_1, t_2, \cdots, t_n$, but all other parameters are known, being either given, namely S_0, C_n, α, β and n, or already solved in terms of the given parameters, namely T^*, λ and δ_i's. There are $2n-1$ independent equations since $\Sigma_i\, a_i - \Sigma_i\, d_i = 0$, and they are nonlinear. They can be transformed, however, into a set of linear equations in the logarithms of the unknown variables by taking logarithms of both sides of the equations as:

for $i = 1, 2, \cdots, n$

$$\ln t_i - \alpha \ln C_{i-1} = \ln(\delta_i T^*), \text{ and}$$

$$-\beta \ln t_i + \ln C_i = \ln(\beta\, \delta_{n+i}\, S_0), \tag{40}$$

which can be written in a matrix form as

$$\mathbf{Ap} = \mathbf{q} \tag{40'}$$

where

$$\mathbf{p} = \begin{bmatrix} \ln t_1 \\ \ln t_2 \\ \vdots \\ \ln t_n \\ \ln C_1 \\ \ln C_2 \\ \vdots \\ \ln C_{n-1} \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} \ln(\delta_1 T^*) \\ \ln(\delta_2 T^*) \\ \vdots \\ \ln(\delta_n T^*) \\ \ln(\beta\delta_{n+1}S_0) \\ \ln(\beta\delta_{n+2}S_0) \\ \vdots \\ \ln(\beta\delta_{2n-1}S_0) \\ \ln(\beta\delta_{2n}S_0) - \ln C_n \end{bmatrix}.$$

The matrix **A** is the same $2n \times (2n-1)$ matrix as given in Eq. (24).

To solve Eq. (40) for the minimizing parameters, a recurrence equation in $\ln C_i$ is first obtained by eliminating $\ln t_i$ from (40) as:

$$\ln C_i - \alpha\beta \ln C_{i-1} = m_i \qquad \text{for } i = 1, 2, \cdots, n, \tag{41}$$

where

$$m_i = \ln S_0 + \beta \ln T^* + (1+\beta)\ln \delta_i, \tag{42}$$

and δ_i is given by Eqs. (28) and (29) in terms of α, β and n. Once the C_i or $\ln C_i$ are determined, the t_i's are given directly by Eq. (40). The solution is obtained as follows.

When $\alpha\beta = 1$

$$C_i = C_n^{\,i/n} \tag{43}$$

and

$$t_i = (nC_i/S_0)^{\alpha} = (n/S_0)^{\alpha}\, C_n^{\,\alpha i/n}. \tag{44}$$

The capacities and access times are in a geometric progression. The ratios are constants which are dependent on the system capacity C_n:

$$C_{i+1}/C_i = C_n^{\,1/n}; \tag{45}$$

$$t_{i+1}/t_i = (C_{i+1}/C_i)^{\alpha} = C_n^{\,\alpha/n} \tag{46}$$

For illustration, when $\alpha = \beta = 1$, $n = 4$, and $C_n = 10^8$, the optimal capacities and access times of the four storage levels are respectively 10^2, 10^4, 10^6 and 10^8 bytes, and 10 ns, 1 μs, 100 μs and 10 ms. It is of interest to note that the constancy in the ratio of the access times of adjacent levels for the optimum configuration has been reported [8], and the constancy in the ratio of capacities has been empirically observed [11].

When $\alpha\beta \neq 1$

The solution of the recurrence equation (41) is

$$\ln C_i = (\alpha\beta)^{i-n} \ln C_n - \sum_{j=1}^{n-i} m_{i+j}/(\alpha\beta)^j, \tag{47}$$

and from Eq. (40) we have

$$\ln t_i = [\ln C_i - \ln(\delta_i S_0)]\beta^{-1} \tag{48}$$

or

$$t_i = (C_i/\delta_i S_0)^{1/\beta}. \tag{48'}$$

Substitution of Eqs. (32) and (42) into (47) with the aid of Eqs. (28), (29) and (33′) yields

$$\ln C_i = \left[\frac{1-(\alpha\beta)^i}{1-(\alpha\beta)^n}\left(1-\frac{n}{n_{\text{opt}}}\right)+\frac{i}{n_{\text{opt}}}\right]\ln C_n, \tag{47'}$$

where n_{opt} is given in Eq. (60). When $n = n_{\text{opt}}$, (47′) becomes

$$C_i = C_n^{\,i/n_{\text{opt}}}. \tag{43'}$$

Hence the capacity ratio are

$$C_{i+1}/C_i = C_n^{\,1/n_{\text{opt}}}. \tag{45'}$$

Furthermore (48) yields, for $n = n_{\text{opt}}$,

$$t_{i+1}/t_i = (\alpha\beta)^{1/\beta}\, C_n^{\,1/\beta n_{\text{opt}}}. \tag{46'}$$

Expected number of faults and CPU cost

Once the capacities of each storage level are determined, it is possible to evaluate the expected number of faults per reference. As given in Eq. (1), the probability for each reference to get a hit at M_i and $i-1$ faults at the all preceding levels, $M_1, M_2, \cdots, M_{i-1}$ is

$$h_i = H(C_i) - H(C_{i-1}) = F(C_{i-1}) - F(C_i). \tag{49}$$

The expected number of faults NF is then

$$NF = \sum_{i=1}^{n} (i-1)h_i. \tag{50}$$

With the assumption that the fault probability $F(C)$ is $F_0\, C^{-\alpha}$, i.e., Eq. (15), and with the convention that $H(C_n) = 1$, the expected number of faults becomes

$$NF = F_0 \sum_{i=1}^{n} (i-1)(C_{i-1}^{\,-\alpha} - C_i^{\,-\alpha}) = F_0 \sum_{i=1}^{n-1} C_i^{\,-\alpha}, \tag{51}$$

where the C_i's are given in Eq. (43) or (47′). The expression in terms of the basic parameters is quite simple for the case in which $\alpha\beta = 1$; substitution of (43) in (51) yields

$$NF = F_0 \sum_{i=1}^{n-1} C_n^{\,-\alpha i/n} = F_0 \frac{C_n^{\,-\alpha/n} - C_n^{\,-\alpha}}{1 - C_n^{\,-\alpha/n}}, \tag{52}$$

which increases with increasing n. For example, when $C_n = 10^6$, $n = 4$, $\alpha = \beta = 1$, and $F_0 = 4$, the expected number of faults per each reference is 0.12.

One could also compute the overhead cost of the CPU for managing the faults, which may differ for different levels. Let $\zeta(i)$ be the overhead cost for managing a fault at the i-th level. The expected overhead cost per each reference CMF is then

$$CMF = \sum_{i=2}^{n} \zeta(i-1)h_i. \tag{53}$$

If the cost is simply proportional to the index of the level, i.e., if

$$\zeta(i) = k\, i, \tag{54}$$

then (53) becomes

$$CMF = k \sum_{i=2}^{n} (i-1)h_i = k \times NF; \tag{55}$$

i.e., the average overhead cost for managing faults per each reference is directly proportional to the expected number of faults per reference.

Optimum number of memory levels

The constrained minimum of the storage hierarchy access time T^*, as given in Eq. (31) or (32) does not depend only on the system cost S_0, capacity C_n, technology and program work load characteristics α and β, but also on the number of the storage levels, n. In the analysis so far, the value of n is considered given as usually is the case in practice. We may, however, consider n as an additional unknown parameter to be determined. The optimum number of storage levels, is that value of n which minimizes the constrained minimum T^*, and can be derived by considering T^* as a function of n with the system parameters S_0, C_n, α and β as constants. The derivations are given for cases $\alpha\beta = 1$ and $\alpha\beta \neq 1$ and are achieved without solving for the minimizing parameters C_i and t_i.

$\alpha\beta = 1$

The derivative of T^*, using Eq. (31), with respect to n is:

$$\frac{dT^*}{dn} = S_0^{-\alpha} C_n^{\alpha/n} n^{\alpha} \left(1 + \alpha - \frac{\alpha}{n} \ln C_n\right), \tag{56}$$

which is negative for small n and positive for large n. Hence, T^* first decreases and then increases as n increases.

By setting the derivative to zero and solving for n, the minimizing value of n is obtained as

$$n_{\text{opt}} = \frac{\alpha}{1+\alpha} \ln C_n = \frac{\ln C_n}{1+\beta}. \tag{57}$$

The optimal number of storage levels is directly proportional to the logarithm of the storage system capacity. Since the number is an integer, the actual optimal value is either the integral part of the above expression or the smallest integer greater than that, and is determined by evaluating and comparing their respective values for T^*. Neglecting this quantization effect, we find the corresponding minimum access time to be

$$T_m^* = S_0^{-\alpha} (e\, n_{\text{opt}})^{1+\alpha}. \tag{58}$$

$\alpha\beta \neq 1$:

Equations (32) and (33) suggest that it would be simpler to take the derivative of $\ln T^*$ than of T^* itself. After some manipulation we find that

$$\frac{d}{dn} \ln T^* = \frac{(\alpha\beta)^n}{\beta[(\alpha\beta)^n - 1]^2} \ln \alpha\beta \times [(1+\beta)n \ln \alpha\beta - (\alpha\beta - 1)\ln C_n]. \tag{59}$$

It can be verified from the sign of the last expression that T^* first decreases and then increases as n increases from zero, and the unique minimizing value of n is

$$n_{opt} = \frac{\alpha\beta - 1}{(1+\beta)\ln \alpha\beta} \ln C_n, \tag{60}$$

which is again proportional to the logarithm of the capacity. As a matter of fact as $\alpha\beta \to 1$, (60) becomes

$$n_{opt} = \ln C_n / (1+\beta).$$

As a numerical example, when $\alpha = \beta = 1/2$, Eq. (60) yields $n_{\text{opt}} = 5$ (or 4), 6 (or 5), and 7 (or 6), respectively, as the optimum number of levels for the storage capacities of 10^6, 10^7 and 10^8 "bytes" [actual unit depends upon the normalization factors used in Eqs. (15) and (16)].

Level-dependent cost

The results obtained so far can readily be extended to include any additional cost that is dependent on the number of levels in the storage hierarchy, such as the so-called per box cost, constant bus cost, and the cost of managing faults as discussed in the preceding section. Let $g(n)$ denote such additional cost; $g(n)$ may depend on any given parameter such as α and β, but is independent of the variables C_i and t_i; $g(n)$ in general is monotone increasing in n.

The system cost now becomes

$$S = \sum_{i=1}^{n} b(t_i) C_i + g(n), \tag{61}$$

and the cost constraint becomes

$$\frac{1}{S_0 - g(n)} \sum_{i=1}^{n} t_i^{-\beta} C_i \leq 1. \tag{62}$$

Equation (62) has the same form as the original constraint, Eq. (18), with S_0 replaced by $S_0 - g(n)$ which is also independent of the variables. Therefore the formulas for the minimum access time T^* are still valid after replacing S_0 by $S_0 - g(n)$. More explicitly, T^* of Eqs. (31) and (32) becomes:

$$T^* = \begin{cases} [S_0 - g(n)]^{-\alpha} C_n^{\alpha/n} n^{1+\alpha} & \text{when } \alpha\beta = 1 \\ [S_0 - g(n)]^{-\alpha} C_n^{\delta n/\beta} T_g(n, \alpha, \beta) & \text{when } \alpha\beta \neq 1. \end{cases} \tag{63}$$

The inclusion of the level-dependent cost $g(n)$ does not change the values of λ and $\boldsymbol{\delta}$. Furthermore, it does not affect the cost allocation nor the access time distribution except that S_0 is replaced by $S_0 - g(n)$ in the definition of a_i in Eq. (35). The equations to determine the minimizing parameters remain valid, the only modification being the above mentioned replacement.

The additional cost $g(n)$, however, does have a significant effect on the optimum number of storage levels. Without $g(n)$, n_{opt} is independent of the system cost S_0; this is no longer true. Moreover, the inclusion of $g(n)$ tends to reduce the optimum number (as would be in-

tuitively expected). Consider first the case wherein $\alpha\beta = 1$ and for simplicity, take

$$g(n) = kn, \tag{64}$$

k being some constant.

$\alpha\beta = 1$

The derivative of ln T^* with respect to n is

$$\frac{d}{dn} \ln T^* = -\frac{kn^2 - [\alpha k \ln C_n + (1+\alpha)S_0]n + \alpha S_0 \ln C_n}{n^2(S_0 - kn)}. \tag{65}$$

By setting this expression to zero, we have a quadratic equation in n. The equation has two positive roots, one being less than, and the other greater than, S_0/k. Since $S_0 - kn$ must be positive, the smaller root is the unique minimizing point:

$$n_{\text{opt}} = \frac{1}{2}\left[\alpha \ln C_n + (1+\alpha)\frac{S_0}{k} - \frac{1}{k}\{[\alpha k \ln C_n + (1+\alpha)S_0]^2 - 4\alpha k S_0 \ln C_n\}^{\frac{1}{2}}\right], \tag{66}$$

which depends not only on ln C_n, but also on S_0, and can be shown to be always less than $\alpha \ln C_n/(1+\alpha)$, the optimum number of levels without including kn.

$\alpha\beta \neq 1$

The derivative is

$$\frac{d}{dn} \ln T^* = \frac{k}{\beta(S_0 - kn)} - \frac{(\alpha\beta)^n \ln \alpha\beta}{\beta[(\alpha\beta)^n - 1]^2} \times [(1+\beta)n \ln \alpha\beta - (\alpha\beta - 1)\ln C_n]. \tag{67}$$

The equation obtained by setting this expression to zero is transcendental in n and no closed-form solution exists. The optimum number of levels, however, can be readily computed and depends on the hierarchy capacity only as ln C_n. It can be seen that inclusion of level-dependent cost, such as per box or fault-managing overhead tends to reduce the optimum number of levels.

The results and formulas presented in this paper can also be used to include to some extent the effect of variable time delays such as address decoding time, bussing times which are dependent on the storage capacities, and access-dependent costs, such as bus cost, without any change in mathematics. All that is required is to change the values of the exponents α and β in the basic time and cost equations, (17) and (18).

Inclusion of these additional time delays and costs will increase the contribution of each storage level to the total hierarchy cost and access time, i.e., to the magnitude of each product term in Eqs. (17) and (18). A simple way to reflect this increase is to decrease the values of α and β, e.g., by replacing α and β by an effective $\alpha_e < \alpha$ and $\beta_e < \beta$, without introducing any additional term in the sum, and thus without requiring any new derivations. One of the effects of using smaller α and β is to reduce the value of the optimum number of memory levels in the hierarchy.

Some generalizations

The preceding formulation and results can be generalized to be applicable to the cases where the values of the characteristic parameters α and β are different for different storage levels.

• *Variable α and β*

Instead of assuming the same technology-cost and hit ratio functions for all storage levels, we can allow different characteristics (still assuming the power functional form) for different levels. Let α_i and β_i denote the corresponding powers for storage level M_i. The hierarchy access time and the cost constraint are now

$$T = t_1 + \sum_{i=2}^{n} C_{i-1}^{-\alpha_{i-1}} t_i \tag{68}$$

and

$$S = \sum_{i=1}^{n} t_i^{-\beta_i} C_i \leq S_0. \tag{69}$$

The phenomenon of leveling off of the miss ratio as previously mentioned can be accommodated by setting α_n equal to zero.

The expression of the constrained minimum of the hierarchy access time of Eq. (19) remains the same; however, the optimizing vector $\boldsymbol{\delta}$ is different. The orthogonality conditions become

$$\delta_i = \beta_i \delta_{n+i} = \alpha_i \beta_i \delta_{i+1} = \delta_n \prod_{j=0}^{n-i-1} \alpha_{i+j} \beta_{i+j} \quad \text{for } i = 1, 2, \cdots, n; \tag{70}$$

$$\alpha_{n+i} = \alpha_i \delta_{i+1} \quad \text{for } i = 1, 2, \cdots, n-1. \tag{71}$$

Hence, for $i = 1, 2, \cdots, n-1$,

$$\delta_i/\delta_{i+1} = \alpha_i \beta_i \text{ and } \delta_{n+i}/\delta_{n+i+1} = \alpha_i \beta_{i+1}. \tag{72}$$

The normalization condition, Eq. (22), yields

$$\delta_n = \left(1 + \sum_{i=0}^{n-1} \prod_{j=0}^{n-i-1} \alpha_{i+j} \beta_{i+j}\right)^{-1}. \tag{73}$$

Substitution of (73) into (70) and (71) gives the explicit expressions for all δ's in terms of the α_i's and β_i's. Furthermore,

$$\lambda = \sum_{i=1}^{n} \delta_{n+i} = \sum_{i=1}^{n} \frac{\delta_i}{\beta_i}. \tag{74}$$

Therefore, the minimum hierarchy access times, Eq. (19), becomes

$$T^* = \lambda^{\lambda} S_0^{-\lambda} C_n^{\delta_n/\beta_n} \prod_{i=1}^{2n} \delta_i^{-\delta_i}. \quad (75)$$

Notice that λ and the δ_i's are functions of n, α_i's, and β_i's but are independent of S_0 and C_n.

Similarly, Eq. (75) can be used to determine the optimum number of levels. An explicit closed form expression for n_{opt} does not seem likely; numerical computation is required. However, it is apparent that the effect of C_n enters only as $\ln C_n$.

• *Nonuniform coefficients:*

Consider the hierarchy access time and the cost constraint as

$$T = \sum_{i=1}^{n} r_i C_{i-1}^{-\alpha} t_i \quad (76)$$

and

$$S = \sum_{i=1}^{n} r_{n+i} t_i^{-\beta} C_i, \quad (77)$$

where the r_i's are some positive constants. When these constants are one, Eqs. (76) and (77) reduce to (17) and (18).

The constrained minimum of the hierarchy access time is now

$$T_r^* = \lambda^{\lambda} S_0^{-\lambda} C_n^{\delta_{2n}} \prod_{i=1}^{2n} \left(\frac{r_i}{\delta_i}\right)^{\delta_i}. \quad (78)$$

The orthogonality and normalization conditions remain the same as before [namely, Eqs. (22), (25) and (26)]. Therefore Λ and δ are the same as given by Eqs. (27)-(30). Substitution of these values in (78) renders

$$T_r^* = T^* \cdot \prod_{i=1}^{2n} (r_i)^{\delta_i}, \quad (79)$$

where T^* is the expression given by either (31) or (32). In other words the hierarchy access time is modified by a multiplicative factor.

• *Combination of previous cases*

The constrained minimum effective hierarchy access time is now given by Eq. (78) with λ and δ given by Eqs. (74), (70), (71) and (73). Again, the optimum number of memory level depends on the logarithm of the hierarchy capacity, rather than directly on the capacity, in all these cases.

Acknowledgments

The author is grateful to W. D. Frazer for suggesting the research, to T. Kaneko for assistance in verifying empirical data and checking some of the mathematical results, and to both for helpful discussions.

References

1. P. J. Denning, "Virtual Memory," *ACM Computing Surveys*, **2**, 153, (1970).
2. J. S. Liptay, "Structural Aspects of the System 360, Model 85, Part 2—The Cache," *IBM Syst J.* **7**, 15 (1968).
3. H. Lorin, *Parallelism in Hardware and Software*, Prentice Hall, Inc., Englewood Cliffs, New Jersey 1972, p. 444.
4. R. L. Mattson, J. Gecsei, D. R. Slutz and I. L. Traiger, "Evaluation Techniques for Storage Hierarchies," *IBM Syst J.* **9**, 78 (1970).
5. C. V. Ramamoorthy and K. M. Chandy, "Optimization of Memory Hierarchies in Multiprogramming Systems," *J. ACM.* **17**, 426 July 1970.
6. I. L. Traiger and D. R. Slutz, "One-Pass Technique for the Evaluation of Memory Hierarchies," Research Report RJ-892, IBM Research Laboratory, San Jose, Calif., July 23, 1971.
7. I. L. Traiger and R. L. Mattson, "The Evaluation and Selection of Technologies for Computer Storage Systems," *Magnetism and Magnetic Materials—1971, AIP Conference Proceedings* No. 5, Part 1, 1972, pp. 1-9.
8. Y. S. Lin and R. L. Mattson, "Cost-Performance Evaluation of Memory Hierarchies," IEEE Trans. Magnetics **MAG-8**, 390 (1972).
9. L. Cooper and D. Steinberg, *Introduction to Methods of Optimization* W. B. Saunders Company, Philadelphia, 1970.
10. R. L. Mattson, "Evaluation of Multilevel Memories," *IEEE Trans. Magnetics* **MAG-7**, 814 (1971).
11. Private communication, from Y. S. Lin of the IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y.
12. R. J. Duffin, E. L. Peterson, and C. Zener, *Geometric Programming*, John Wiley & Sons, Inc., New York, 1967.

Received December 18, 1973

The author is located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.