



ELSEVIER

European Journal of Operational Research 123 (2000) 206–225

EUROPEAN
JOURNAL
OF OPERATIONAL
RESEARCH

www.elsevier.com/locate/orms

Theory and Methodology

An archiving model for a hierarchical information storage environment

Kamran Moinzadeh ^{a,*}, Emre Berk ^b

^a *School of Business, Box 353200, University of Washington, Seattle, WA 98195, USA*

^b *Faculty of Business Administration, Bilkent University, Ankara, Turkey*

Received 2 March 1998; accepted 15 December 1998

Abstract

We consider an archiving model for a database consisting of secondary and tertiary storage devices in which the query rate for a record declines as it ages. We propose a 'dynamic' archiving policy based on the number of records and the age of the records in the secondary device. We analyze the cases when the number of new records inserted in the system over time are either constant or follow a Poisson process. For both scenarios, we characterize the properties of the policy parameters and provide optimization results when the objective is to minimize the average record retrieval times. Furthermore, we propose a simple heuristic method for obtaining near-optimal policies in large databases when the record query rate declines exponentially with time. The effectiveness of the heuristic is tested via a numerical experiment. Finally, we examine the behavior of performance measures such as the average record retrieval time and the hit rate as system parameters are varied. © 2000 Elsevier Science B.V. All rights reserved.

1. Introduction

The rapid increase of data requirements has made data management one of the greatest concerns of information system managers (Brancheau and Wetherbe, 1977; Dickson and Nechis, 1984; Niederman et al., 1991; Szajna, 1994). According to industry analysts, the average Fortune 1000 company now manages over one terabyte of data, and, by the end of this century, it will manage over one petabyte (LSC, 1995).¹ Data requirements between enterprises differ greatly. Some require the storage of very large files, some have data that are dynamic, others have data that are only read, never updated. Following Zipf's law, some data are accessed frequently after its creation, some rarely, and some never (Considine and Myers,

* Corresponding author. Tel.: +1 206 543 1932; fax: +1 206 685 9392.

¹ A terabyte is one million megabytes, and a petabyte is one billion megabytes. As a point of reference, storing one terabyte on 9-track tape requires 6666 reels, at an estimated cost of over \$100,000. Although tape prices have recently declined sharply, maintenance costs remain high.

1977). It is estimated that only 15–25% of local area network (LAN) data are accessed or modified within 90 days, and that the remaining 75–85% of data are static (LSC, 1995). Furthermore, it has been observed in various settings that the intensity of access to a data set declines throughout the lifetime of the data set [e.g., financial records in Gravina (1978) text editor data sets in Smith (1981) images or voice recordings as objects in Harding et al. (1990)]. This is due to the fact that the information value of stored data diminishes with time. As the record ages, it is less valuable to users and is accessed less frequently; and at some age it may become obsolete altogether and can be justifiably deleted permanently. The retention regulations for data are industry-specific. For instance, in banking, financial records are required by law to be maintained accessible for seven years in the US (10 years in UK); in aircraft industry, the documentation is kept for the lifetime of an airplane, say 50 years. This diverse nature of access requirements of stored data has long been recognized and exploited through storage hierarchies.

A storage hierarchy is usually defined in terms of access speed, capacity and cost. Short term data that are accessed most frequently is stored on the secondary device such as magnetic disk. As the data ages, it is migrated nearline to less costly and more abundant tertiary storage. It may be first moved to erasable optical media (jukebox), where access takes a few seconds and, then to magnetic tape (cartridge library), where it can take several minutes to retrieve the data when needed. The oldest data can be stored offline, where retrieval may take days. Organizations can set up three or four tiers of migration and storage, depending on their needs and resources.

The static design issues as selection of storage medium and the operational trade offs between speed, capacity and cost ensuing from assignment and reorganization of files in multiple media have been studied by several authors (e.g., Gecsei and Lukes, 1974; Lum et al., 1975; Cohen et al., 1989; Han and Diehr, 1991; Klastorin et al., 1993). The migration management of files along a hierarchy of different storage media, called hierarchical storage management (HSM), has also received some attention (e.g., Smith, 1981; Lawrie et al., 1982).

HSM technology has existed since the 1970s on mainframes; users who have operated terminals in a MVS environment, for example, were using HSM technology transparently (Considine and Myers, 1977). Under HSM, data are transferred from secondary to tertiary archival storage medium according to user defined migration criteria. Among the used criteria are predetermined lifespans and access intensity based rules. With predetermined lifespans, a file is moved away from the secondary medium whenever it reaches a certain age regardless of the secondary medium (disk) capacity. With access intensity based rules, eligible files for archiving are selected by means of a ‘desirability’ index. The system administrator may tag certain executable files as permanently desirable so that they are never archived. The remaining files are classified according to their future usage, that is, the future access intensity. The files that are to be most frequently accessed are tagged ‘active’ and those that are to be less frequently accessed are tagged ‘inactive’. Inactive files are eligible for archiving. The archiving is performed either periodically or as triggered by a high threshold or watermark of disk capacity. The most commonly used rule for determining the intensity with which a file is to be accessed in the future is the least-recently-used (LRU) rule [e.g., Considine and Myers (1977), Smith (1981) and Lawrie et al. (1982), on mainframes, and Nance (1995), on commercial migration software for LANs]. According to the LRU rule, the file that has been accessed most recently is deemed most frequently used and the one that has not been accessed for the longest time is deemed least frequently used. Thus, the LRU rule assumes that the interreference distribution for files is stationary, that is, the access intensity of a file does not depend on the age of the file. Although this assumption may hold for some data, it is a shortcoming for others. More sophisticated rules of file migration based on the entire history of file usage are already becoming available in commercial software [(e.g., Disk Historian for PCs in Brown (1994)].

In this paper, we consider a database to which new records are added over time, and in which the information value of individual records is decreasing as the records age. Examples of such databases are online full text libraries of newspapers and journals (e.g., LEXIS/NEXIS), databases of stock and foreign exchange quotes (e.g., Teletex in Tanton, 1979), meteorological data repositories (e.g., NCAR in Than-

hardt and Harano, 1988), customer account information archives for banks (e.g., Gravina, 1978) and databases of patient medical records for hospitals. In these databases, the most recently created record has the highest information content but the aged records may also be accessed for historical analysis. Given the large sizes of such databases, one may want to exploit the cost advantages of multiple storage media with HSM; however, current migration rules used in HSM are not appropriate for databases with aging records. The main reason is that HSM treats a database as one large file and ignores the information value of aging individual records within a database (Ryan, 1994). For this purpose, we propose a dynamic archiving policy class that considers both the disk capacity usage and the true access intensity of individual records measured as a function of their ages.

The remainder of the paper is organized as follows: In Section 2, we present the basic model and propose an archiving policy to minimize the average access times. In Sections 2 and 3, we analyze two special cases of the basic model from a theoretical perspective. In Section 3, we consider the case when new records are added to the database periodically (at regular intervals) and develop the expressions for operating characteristics of the system. In Section 4, we analyze the case when new records arrive randomly. In Section 5, we provide a numerical study to investigate the behavior of the archiving model under the proposed policy and discuss some of the practical aspects on its implementation. Section 6 summarizes our findings and mentions possible extensions.

2. The model and assumptions

We consider a hierarchical information storage system (as depicted in Fig. 1) consisting of a secondary (e.g., magnetic disk) and a tertiary storage device (e.g., optical juke box). New records (i.e., volumes of journals, criminal or medical records, etc.) are added (inserted) to the system over time. We assume that as records age, they become less desirable by the users of the system and, thus, experience less inquiry. As discussed before, this behavior is experienced frequently in many situations. For an example, the reader is referred to Gravina (1978). Let $\theta(x)$ be a continuous and differentiable function which denotes the instantaneous arrival rate of queries for a record that is aged x since its insertion in the system.² We note that $\partial\theta(x)/\partial x < 0$, and that the overall number of queries over the life time of a record, $\Gamma = \int_0^\infty \theta(x) dx$, is finite. Furthermore, let $\Theta(t)$ denotes the expected cumulative number of inquiries for a record up to age t ; that is

$$\Theta(t) = \int_0^t \theta(x) dx. \quad (1)$$

New records are first added to the secondary device. As the number of records in the secondary device grows, their access (retrieval) times may deteriorate. Therefore, in order to achieve a better average access time for inquiries, older records which have a lower average query rate are transferred to the tertiary storage device according to an archiving policy described later. Let τ_n denote the average access (retrieval) time of a record from the secondary device with a total of n records. We assume τ_n is a non-decreasing, concave function in n (Knuth, 1973; Sahni and Horowitz, 1990). Furthermore, the average access (retrieval) time of a record from the tertiary storage device is assumed to be constant and independent of the number of the records in the tertiary storage device and is denoted by $\bar{\tau}$. We note that we have assumed that the access times are independent of the record size. This assumption is reasonable in most situations since the size of records which are commonly accessed in blocks is usually significantly smaller than a block (see Ullman, 1988, pp. 296). In cases when the record size is larger than a block, the assumption is still rea-

² A summary of the key notations appears in Appendix B.

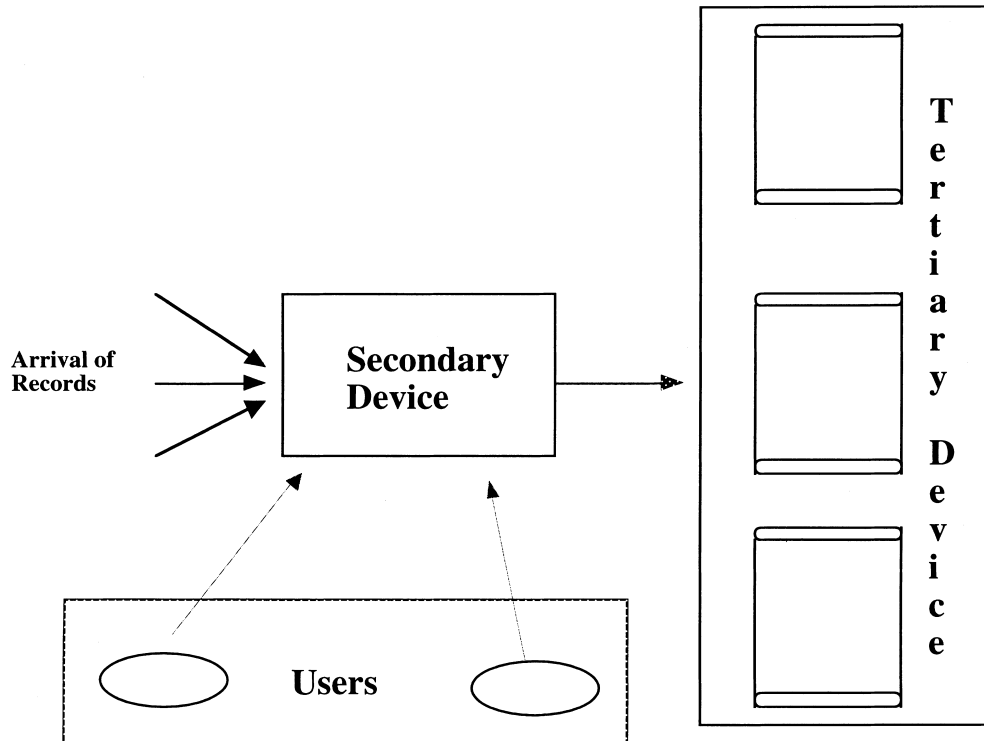


Fig. 1. The schematic of the hierarchical storage environment.

sonable if all records are homogeneous. Without loss of generality, we assume that both secondary and tertiary devices have ample capacity. As we shall see later, this assumption can be easily relaxed for the secondary device. Furthermore, the assumption is quite reasonable for the tertiary device as the storage medium for tertiary devices (i.e., magnetic or optical disk) is relatively inexpensive and can be added to the system as needed. Finally, we assume that transfer times of records from secondary to the tertiary device (the archiving operation) are negligible, transfers are performed in real time and that records transferred to the tertiary storage device will remain there permanently (i.e., once archived, they will not be moved back to the secondary device).

We first approach archiving decisions for such hierarchically stored databases purely from a modeling perspective and propose a general operating policy. Our objective is to lay out a theoretical framework so that the trade off structures of the optimization problems may be exhibited in Sections 3 and 4. This theoretical study of the properties of the optimization problems will help us address the practical implementation issues for large scale systems, and we shall examine in Section 5 an effective yet simple heuristic for realistic databases.

The form of the optimal archival policy in the setup described above is an open question and depends on the operational objective(s) of the organization or the system manager; however, one may conjecture that it would involve the number records as well as the ages and the size of the all the records in the secondary device. We propose the following ‘dynamic’ archiving policy class which captures the essence of the first two elements (the number of records and the age of the records in the secondary device) of the conjecture.

Archive Policy: *When there are n records in the secondary device, a record is archived to the tertiary storage device if its age is greater than or equal to T_n .*

Define the state of the system as $x_n(t) = \{x_1(t), \dots, x_n(t)\}$, where n is the number of records in the secondary device at time t and $x_i(t)$ denotes the age of the i th record at time t . We assume that $0 \leq x_1 \leq x_2 \leq \dots \leq x_n$. Note that $\mathbf{T} = \{T_n; n > 0\}$ is the policy vector. It is practical to assume that the archiving trigger times are a non-increasing function of the number of records in the secondary device; that is $T_i \geq T_{i+1}$. In other words, with more records in the secondary device, the archiving decisions are made at least as early as when there are fewer records in the secondary device. Clearly, $0 \leq x_1(t) \leq \dots \leq x_n(t) \leq T_n$ for $n \geq 1$. Since the system is monitored continuously, therefore, archiving decisions apply only to the oldest record in the secondary device.

We believe that the above policy is most suitable in situations when the secondary objective of the system is to minimize the average access (retrieval) times for a record. Several common archival policies employed in practice are special cases of the archive policy described above. For instance, by setting all the elements of the policy vector, \mathbf{T} , equal, a single age archiving policy can be achieved; that is, records are transferred and stored in the tertiary storage device when they reach the same pre-specified age. Furthermore, by setting T_i to infinity for $i \leq M_L$ and $T_i = 0$ for $i \geq M_U$, and choosing finite non-zero values for T_i , $M_L \leq i \leq M_U$, one can achieve, in steady state, an archiving policy with a minimum and a maximum number of records, M_L and M_U , respectively, in the secondary device. Also, setting $M_L = M_U - 1$ simply results in a policy where the number of records in the secondary device stays at a constant level M_L at all times. Finally, we note that the assumption on secondary device having infinite capacity can be relaxed by setting $T_i = 0$ for $i > M_C$, where M_C is the maximum number of records allowed in the secondary device.

In the next two sections, we develop and analyze two special cases of such systems, one with constant inter-arrival times of records and the other with random (Poisson) arrivals of the records.

3. The case with periodic arrivals of the records

In this section, we analyze the above model when the inter-arrival time of the records is constant, $1/\mu$; that is, a record is added (inserted) every $1/\mu$ time units. Such arrival patterns of records occur when the data stored becomes available periodically such as newspapers, journals, daily stock market reports.

Let $p_j(x_1, \dots, x_j)$ denote the steady state probability density of the system being in state \mathbf{x}_j where \mathbf{x}_j is the vector containing the ages of the j records in the secondary device, as defined before. Then, for any given policy \mathbf{T} which follows the properties of the policy class described above ($T_i \geq T_{i+1}$ for $i \geq 1$), let n be the *smallest* value of the size of the secondary device such that $T_n \leq n/\mu$. This implies that $T_1 \geq T_2 \geq \dots \geq T_{(n-1)} > (n-1)/\mu$ and $T_i \leq T_n \leq n/\mu$ for $i > n$. Since arrivals of the records occur every $1/\mu$ time units, then after the first n arrivals of records, the secondary device will grow to n records with ages which are $1/\mu$ units apart from each other at all times. The secondary device will have n records for $[T_n - (n-1)/\mu]^+$ after the arrival of a record where $[x]^+ = \max(0, x)$. At this time, the oldest record reaches an age T_n and, therefore, is transferred to the tertiary device leaving the secondary device with $(n-1)$ records until the next arrival of a record (for $1/\mu - [T_n - (n-1)/\mu]^+$ time units) as depicted in Fig. 2. Note that, at the time when a new record is about to arrive, the age of the oldest record in the secondary device will be $(n-1)/\mu$ and the age of the youngest record in the archive will be n/μ . Thus, it can be observed that the process \mathbf{x}_n is ergodic and:

$$P_n(\xi, \xi + 1/\mu, \xi + 2/\mu, \dots, \xi + (n-1)/\mu) = \mu \quad \text{for } 0 \leq \xi \leq [T_n - (n-1)/\mu]^+, \quad (2)$$

$$P_{n-1}(\xi, \xi + 1/\mu, \xi + 2/\mu, \dots, \xi + (n-2)/\mu) = \mu \quad \text{for } [T_n - (n-1)/\mu]^+ \leq \xi \leq 1/\mu, \quad (3)$$

and $p_j(x) = 0$, otherwise.

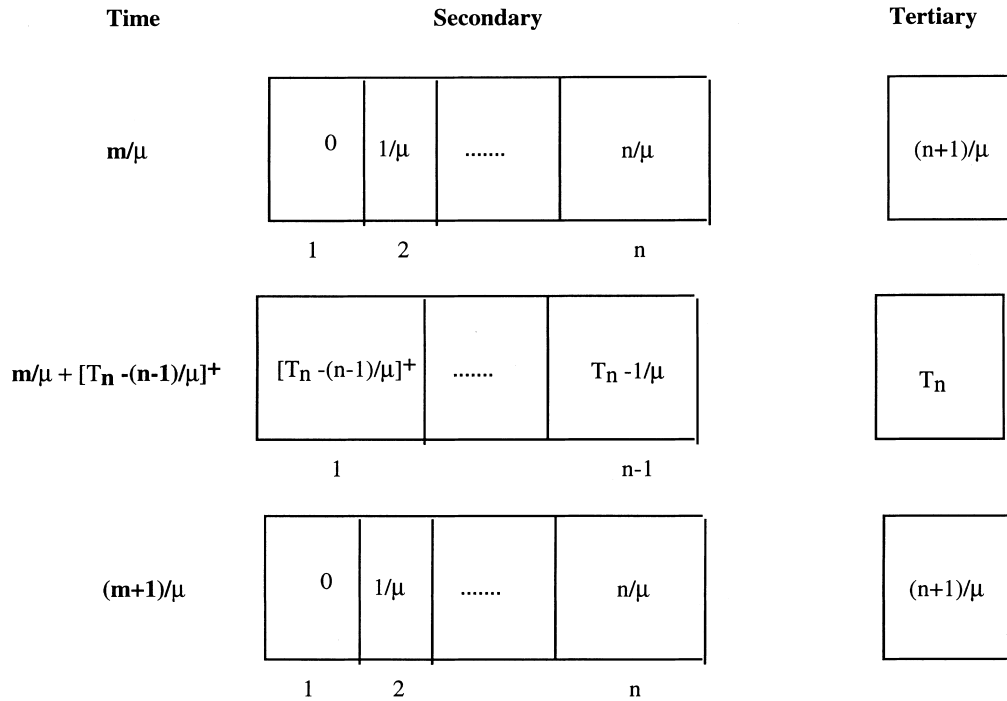


Fig. 2. Depiction of the evolution of the system over time when inter-arrival times are constant.

Furthermore, the steady state probability of having j records in the secondary system can be obtained from Eqs. (2) and (3) as

$$P_j = \begin{cases} [T_n - (n-1)/\mu]^+ \mu & j = n, \\ [1/\mu - [T_n - (n-1)/\mu]^+] \mu & j = n-1, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Let L denote the number of queries residing in the system and R be the record retrieval time. Following from the above discussion, it is shown in Appendix A that the average number of queries residing in the system at any point in time, $E(L)$, is

$$E(L) = \mu \bar{\tau} \Gamma - \mu (\bar{\tau} - \tau_{n-1}) \Theta([n-1]/\mu) - \mu (\bar{\tau} - \tau_n) \int_0^{(T_n - (n-1)/\mu)^+} \theta([n-1]/\mu + \xi) d\xi \\ + \mu (\tau_n - \tau_{n-1}) \sum_{i=0}^{n-2} \int_0^{(T_n - (n-1)/\mu)^+} \theta(i/\mu + \xi) d\xi. \quad (5)$$

One can easily show that the average query rate for records in the system is equal to $\mu \Gamma$. This can be observed by setting the access times (t_{n-1} , τ_n and $\bar{\tau}$) equal to one in Eq. (5).³ Thus, from Little's law (Stidham, 1972), the average record retrieval time for the system, $E(R)$, can be written as

³ The derivation of the average query rate to the system is similar to that of $E(L)$ in the Appendix without the access times in the expression.

$$E(R) = E(L)/(\Gamma\mu). \quad (6)$$

Finally, using Eq. (A.1), the hit rate, γ , defined as the fraction of queries accessed from the secondary device can be written as

$$\begin{aligned} \gamma &= \left\{ \int_0^{(T_n - (n-1)/\mu)^+} \theta([n-1]/\mu + \xi) \, d\xi + \sum_{i=0}^{n-2} \int_0^{1/\mu} \theta(i/\mu + \xi) \, d\xi \right\} / \Gamma \\ &= \Theta((n-1)/\mu + [T_n - (n-1)/\mu]^+) / \Gamma. \end{aligned} \quad (7)$$

From Eq. (5), we first note that the average retrieval time for a record in the system is only a function of a single trigger time, T_n , with n being the smallest value of the size of the secondary device such that $T_n \leq n/\mu$, as discussed above. We also note that, due to the integration limits, for a given value of n , $E(R)$ takes on the same value for all $T_n \leq (n-1)/\mu$. Thus, to find n^* and T_n^* which minimize the average retrieval time for a record, $E(R)$, we only need to consider values of $T_n \in [(n-1)/\mu, n/\mu]$.

Lemma 1. *Let*

$$\alpha(n) = (\bar{\tau} - \tau_n) / \{(n-1)(\tau_n - \tau_{n-1})\} \quad \text{for } n > 1.$$

Then $\alpha(n)$ is non-increasing in n when τ_n is non-decreasing and concave in n .

Proof. The proof follows from examining $\alpha(n+1) - \alpha(n)$.

Lemma 2. *Let n_U be the smallest value of n for which $\alpha(n) \leq 1$. Then, $n^* \in [1, 2, \dots, n_U]$.*

Proof. We write

$$T_n = (n-1)/\mu + t_n, \quad 0 \leq t_n \leq 1/\mu. \quad (8)$$

Define,

$$\psi(n, t_n) = E(R)\Gamma. \quad (9)$$

Now, for $n = 1$,

$$\partial\psi/\partial t_1 = -(\bar{\tau} - \tau_1)\theta(t_1) < 0.$$

Next, for $n > 1$,

$$\partial\psi/\partial t_n = -(\bar{\tau} - \tau_n)\theta([n-1]/\mu + t_n) + (\tau_n - \tau_{n-1}) \sum_{i=0}^{n-2} \theta(i/\mu + t_n).$$

Since $\theta(\cdot)$ is a decreasing function, we know that

$$\partial\psi/\partial t_n > -(\bar{\tau} - \tau_n)\theta([n-1]/\mu + t_n) + (\tau_n - \tau_{n-1})(n-1)\theta([n-1]/\mu + t_n),$$

simplifying,

$$\partial\psi/\partial t_n > -(\tau_n - \tau_{n-1})(n-1)[\alpha(n) - 1]\theta([n-1]/\mu + t_n).$$

Since the RHS of the above inequality is positive when $\alpha(n) \leq 1$, $\partial\psi/\partial t_n > 0$ for $n \geq n_U$. Noting that $\alpha(n)$ is non-increasing in n , $\partial\psi/\partial t_n$ if attains the value of zero exactly (in accordance with the first order opti-

mal condition), it should happen at $n^* \in (1, n_U - 1]$; otherwise, $(\partial\psi/\partial t_n)$ changes sign but never equals zero exactly), ψ reaches its minimum at $n^* \in [1, n_U - 1]$. \square

Corollary 1. *When the query rate for a record decays exponentially,⁴ that is, $\theta(x) = a \exp(-bx)$ for $x \geq 0$ with b and a both being positive, then $T_n^* = n^*/\mu$ where n^* is the largest value of n satisfies:*

$$\exp(b(n-1)/\mu) \leq [(\bar{\tau} - \tau_n)/(\tau_n - \tau_{n-1})][1 - \exp(-b/\mu)] + 1. \quad (10)$$

Proof. For $n > 1$, let

$$\begin{aligned} \phi(n) &= \psi(n, 1/\mu), \\ \Delta\phi(n) &= \phi(n) - \phi(n-1), \\ \Delta^2\phi(n) &= \Delta\phi(n+1) - \Delta\phi(n). \end{aligned}$$

Then, from Eqs. (5) and (6), we have

$$\begin{aligned} \phi(n) &= \bar{\tau}\Gamma - (\bar{\tau} - \tau_n)\Theta(n/\mu), \\ \Delta\phi(n) &= (\tau_n - \tau_{n-1}) \left\{ -[(\bar{\tau} - \tau_n)/\tau_n - \tau_{n-1}] \int_{(n-1)/\mu}^{n/\mu} \theta(x) \, dx + \Theta([n-1]/\mu) \right\} \\ &= [a(\tau_n - \tau_{n-1})/b] \left\{ 1 - \{[(\bar{\tau} - \tau_n)/(\tau_n - \tau_{n-1})][1 - \exp(-b/\mu)] + 1\} \exp(-b[n-1]/\mu) \right\}. \end{aligned}$$

Now, with some effort, we can write

$$\partial\psi/\partial t_n = b\Delta\phi(n) \exp(-bt_n)/[1 - \exp(-b/\mu)]. \quad (11)$$

Note, that Eq. (11) is positive (negative) when $\Delta\phi(n)$ is positive (negative) for any value of t_n . Therefore, $T_N^* = n^*/\mu$ where n^* is found by satisfying the first order condition of optimality for $\phi(n)$. That is, $\Delta\phi(n) \geq 0$ and $\Delta\phi(n+1) > 0$, which implies Eq. (10). \square

We should note that whether the extremum is a global minimum is specific to the functional form of τ_n . In the case when τ_n is linear in n , for instance, it can easily be shown that the extremum is a global minimum.

4. The case with random arrivals of records

In this section, we analyze the model when the inter-arrival time of the records is random. Specifically, we assume that the inter-arrival time of the records is exponentially distributed with a mean $1/\mu$; or alternatively, the arrivals of new records follow a Poisson process with a mean rate of μ . Such an arrival pattern occurs when a record is added randomly, possibly by many users (or sources) in real time, such as order invoices, new bank accounts, police or medical records.

⁴ The exponential family is usually a good fit for describing the query rates for records. For example, extracting the query data from Fig. 1 in Gravina (1978), we were successfully able to fit the retrieval times for both the ‘on-line’ and ‘overall’ requests to the exponential family.

As before, $p_n(t, x_1, \dots, x_n)$ let denote the probability density of the system being in state \mathbf{x}_n at time t . We now derive the system of partial differential equations and their boundary conditions which describe the state of the system. Our approach is similar to the one employed by Cox (1955), Gnedenko and Kovalenko (1968), Schmidt and Nahmias (1985) and Moinzadeh (1989).

The state of the system at time t , $\mathbf{x}_n(t)$, can be viewed as the position of a particle in the region $0 \leq x_1 \leq \dots \leq x_n \leq T_n$. The motion of the particle is discontinuous when a new record is inserted. Such instances (i.e., $x_1 = 0$) constitute the boundary points. Thus, the partial differential equations governing the state of the system and their boundary conditions can be written as follows:

Case 1: $n \geq 1$, $x_1 > 0$ and $x_n < T_{n+1}$. Then

$$\begin{aligned} p_n(t+h, x_1+h, \dots, x_n+h) \\ = (1-\mu h)p_n(t, x_1, \dots, x_n) + (1-\mu h) \int_{T_{n+1}-h}^{T_{n+1}} p_{n+1}(t, x_1, \dots, x_n, \zeta) d\zeta + o(h). \end{aligned}$$

This follows since the state \mathbf{x}_n can then be reached at time $t+h$ either if there were n records in the secondary device at time t , or if there were $n+1$ records in the secondary device at time t and the oldest record was archived to the tertiary device after having reached the age of T_{n+1} during the interval $(t, t+h)$. All other transitions have probability $o(h)$. Adding and subtracting terms (see Moinzadeh, 1989), employing the integral mean value theorem, dividing both sides by h , and letting $h \rightarrow 0$, at steady state we obtain

$$\sum_{i=1}^n \frac{\partial}{\partial x_i} p_n(x_1, \dots, x_n) = -\mu p_n(x_1, \dots, x_n) + p_{n+1}(x_1, \dots, x_n, T_{n+1}),$$

where $p_n(x_1, \dots, x_n)$ denotes the steady state probability density of \mathbf{x}_n .

Case 2: $n \geq 1$, $x_1 > 0$ and $T_{n+1}x_n < T_n$. This case is similar to the one above except that, since $T_{n+1} \leq x_n < T_n$, transitions from states with $(n+1)$ records are not allowed. Then

$$p_n(t+h, x_1+h, \dots, x_n+h) = (1-\mu h)p_n(t, x_1, \dots, x_n) + o(h)$$

and at steady state, we obtain

$$\sum_{i=1}^n \frac{\partial}{\partial x_i} p_n(x_1, \dots, x_n) = -\mu p_n(x_1, \dots, x_n).$$

Case 3: $n = 0$. In this case, all records are in the tertiary storage device and the secondary device is empty. This state can be reached only if there were no records in the secondary device at t and no new record was added in $(t, t+h)$ or there was one record in the secondary device at t and its age reached T_1 during $(t, t+h)$ and was, therefore, archived to the tertiary device. Hence,

$$p_0(t+h, \cdot) = (1-\mu h)p_0(t, \cdot) + (1-\mu h) \int_{T_1-h}^{T_1} p_1(t, \zeta) d\zeta + o(h).$$

Once again, at steady state we have

$$\mu p_0(\cdot) = p_1(T_1).$$

Next we consider the boundary conditions for the above system of partial differential equations. As noted before, that the boundary conditions are found by considering the discontinuities in the

motion of the state of the system caused by an insertion (arrival) of a new record and are derived as follows.⁵

For, $n \geq 1$ and $x_{n-1} < T_n$, the insertion (arrival) of a new record introduces a record with an age of zero in the secondary system. Therefore, the transitions to state $(0, x_1, \dots, x_{n-1})$ occur *either* if there are $(n-1)$ records in the secondary device and an insertion (arrival) of a new record occurs, or if there are n records in the secondary device and the age of the oldest record on-line has exceeded T_{n+1} when a new record is inserted (has arrived). In such situations, the insertion (arrival) of the new record will bring the state of the system to $(n+1)$ records, causing the oldest record in the secondary device to be archived which leaves the secondary device with n records. These transitions occur in an infinitesimal time.

Thus, we have

$$p_n(0, x_1, \dots, x_{n-1}) = \mu p_{n-1}(x_1, \dots, x_{n-1}) + \mu \int_{(T_{n+1} \vee x_{n-1})}^{T_n} p_n(x_1, \dots, x_{n-1}, \zeta) d\zeta,$$

where $(x \vee y) = \max(x, y)$.

4.1. Operating characteristics of the system

It can be verified that a solution to the above system of partial differential equations and their boundary conditions is:

$$p_0(\cdot) = \exp(-\mu T_1), \quad (12a)$$

$$p_n(x_1, \dots, x_n) = \begin{cases} \mu^n \exp(-\mu T_{n+1}), & x_n \leq T_{n+1} \\ \mu^n \exp(-\mu x_n), & T_{n+1} < x_n \leq T_n \end{cases} \quad \text{for } n \geq 1. \quad (12b)$$

The steady state probability of having n ($n \geq 1$) records in the secondary device, P_n , is given by

$$P_n = \int_0^{T_{n+1}} \int_{x_1}^{T_{n+1}} \cdots \int_{x_{n-1}}^{T_{n+1}} \mu^n \exp(-\mu T_{n+1}) dx_n \cdots dx_2 dx_1 \\ + \int_{T_{n+1}}^{T_n} \int_0^{x_n} \cdots \int_{x_{n-2}}^{x_n} \mu^n \exp(-\mu x_n) dx_{n-1} \cdots dx_2 dx_1 dx_n.$$

Upon simplification, we get

$$P_n = Q(n, \mu T_n) - Q(n+1, \mu T_{n+1}), \quad (13)$$

where

$$q(i, \mu t) = (\mu t)^i \exp(-\mu t) / i!,$$

⁵ For a detailed discussion on the derivation of the boundary conditions, see Moinzadeh (1989).

$$Q(y, \mu t) = \sum_{i=y}^{\infty} q(i, \mu t).$$

It is shown in Appendix A that the expected number of queries residing in the system, $E(L)$, is given by

$$\begin{aligned} E(L) = & \mu \bar{\tau} \Gamma - \mu(\bar{\tau} - \tau_1) \Theta(T_1) + \mu \sum_{n=2}^{\infty} (\tau_n - \tau_{n-1}) \Theta(T_n) Q(n-1, \mu T_n) \\ & + \mu \sum_{n=1}^{\infty} (\bar{\tau} - \tau_n) \int_{y=T_{n+1}}^{T_n} \theta(y) Q(n, \mu y) dy. \end{aligned} \quad (14)$$

In a similar fashion as in Section 3, one can easily show that the average query rate for records in the system is equal to $\mu \Gamma$ by setting the access times to unity in Eq. (14). From Little's law (Stidham, 1972), the average record retrieval time for the system, $E(R)$, can be obtained as

$$E(R) = E(L) / (\Gamma \mu). \quad (15)$$

Finally, from Eq. (A.6), the hit rate, γ , defined as the fraction of queries accessed from the secondary device can be written as

$$\gamma = \left\{ \Theta(T_1) - \sum_{n=1}^{\infty} \int_{T_{n+1}}^{T_n} \theta(y) Q(n, \mu y) dy \right\} / \Gamma. \quad (16)$$

In what follows, we present some properties of the policy parameters which minimize the average retrieval times for a record, $E(R)$.

Proposition 1. *The optimal policy parameters, T_n^* , which minimize the average record retrieval time are only a function of the arrival rate of queries and the average access times at the secondary and tertiary device. Specifically:*

- (i) T_n^* is set to infinity.
- (ii) Let n_U be the smallest value of n for which $\alpha(n) \leq 1$. Then, $T_n^* = 0$ for $n \geq n_U$. For $1 < n < n_U$, $\alpha(n) > 1$ and $T_n^* > 0$.

Proof. To find the policy vector, \mathbf{T}^* , which minimizes the average record retrieval times, we examine the derivative of $E(R)$ w.r.t. T_n :

- (i) Follows from

$$\partial E(R) / \partial T_1 = -(\bar{\tau} - \tau_1) q(0, \mu T_1) / \Gamma < 0 \quad \text{for } T_1 > 0. \quad (17)$$

- (ii) With some effort, we can write

$$\partial E(R) / \partial T_n = \mu(\tau_n - \tau_{n-1}) q(n-2, \mu T_n) f(T_n) / \Gamma \quad \text{for } n > 1, \quad (18)$$

where

$$f(T_n) = \Theta(T_n) - \alpha(n) T_n \theta(T_n). \quad (19)$$

We note that the optimality conditions are determined only by $f(T_n)$. Now, $\theta(\cdot)$ since is decreasing, we can show

$$\Theta(T_n) \geq T_n \theta(T_n). \quad (20)$$

From Eq. (20), we note that $\partial E(R)/\partial T_n \geq 0$ for all values of n such that $\alpha(n) \leq 1$ which implies that $T_n^* = 0$. Since $\alpha(n)$ is non-increasing in n , (ii) follows. \square

Lemma 3. *When the query rate for a record decays exponentially, that is, $\theta(x) = a \exp(-bx)$ for $x \geq 0$ with a and b both being positive, then for $1 < n < n_U$, T_n^* is obtained by solving*

$$[bT_n\alpha(n) + 1] = \exp(bT_n). \quad (21)$$

Proof. From Eq. (19), the first order condition of optimality reduces to Eq. (21). It can be verified that, the solution to Eq. (21) is a maximum when $T_n \leq [\alpha(n) - 1]/[b\alpha(n)]$ and a minimum, otherwise. Furthermore, $f(0) = 0$ and $\partial f/\partial T_n$ is negative at $T_n = 0$. Therefore, the solution to Eq. (20) is greater than $[\alpha(n) - 1]/[b\alpha(n)]$ and, thus, is a minimum. \square

Collorary 2. *The approximate solution to Eq. (21) is*

$$T_n^* = 2[\alpha(n) - 1]/b. \quad (22)$$

Proof. Using the Taylor's expansion, we have

$$\exp(bT_n) \cong 1 + bT_n + (bT_n)^2/2. \quad (23)$$

Inserting Eq. (23) in Eq. (21) we get Eq. (22).

5. Numerical results and practical considerations

In this section, we build on the optimization results of the previous sections, and address the practical considerations for implementing the proposed archiving policy and investigate the impact of the parameters of the operating environment on the average retrieval time and hit rate performance of the information system.

One can obtain the archiving policy parameters which minimize $E(R)$ using the results in the previous sections; that is, a distinct trigger time T_n^* can theoretically be determined for every possible value of the size of the secondary device, n . However, for large databases, it may be computationally tedious to search for and then implement all of the elements of the policy vector. Thus, one may suffice with a small number of blocks of distinct trigger times (or, distinct trigger levels) rather than the entire policy vector. Furthermore, users or system administrators may find it cumbersome to implement such a large policy vector for realistic databases of hundreds of thousands of records. Instead, they may choose to operate with a small number distinct trigger levels similar to watermarks. Hence, it is of both practical and theoretical interest that we examine the sensitivity of the archiving policy to the number of distinct trigger levels and investigate robust heuristic alternatives for realistic databases.

In the following, we provide an efficient heuristic archiving policy for the case when record arrivals are random. The proposed heuristic utilizes the optimization results obtained earlier and approaches the best policy in the limit. The heuristic policy is determined as follows: In accordance with Proposition 1, we set $T_1 = \infty$ and $T_n = 0$ for $n \geq n_U$. A heuristic solution with a single trigger level is, then, obtained by letting $T_n = T_1$ for $1 \leq n < n_U$. Under this operating regime, the secondary device holds n_U records at all times at steady state. The heuristic solutions with more than one trigger level are obtained using a *fractile* rule based on the query request distribution over the lifetime of a record. For example, in order to get a heuristic solution with two trigger levels, we first compute the median age of a record (i.e., the age at which 50% of

the overall query requests for the record have been made), say, t . Next, using Lemma 3, we obtain the corresponding secondary device size for this trigger time, say, m . Then, the two level heuristic solution is: $T_n = \infty$ for $1 \leq n < m$, $T_n = t$ for $m \leq n < n_U$ and $T_n = 0$ for $n \geq n_U$. Note that, in this case, the median (or, the 50th fractile) age divides the cumulative query distribution into two equal parts. In order to get a heuristic solution with *three* trigger levels, we partition the cumulative query distribution into *four* equal parts; thus, we obtain the 25th, 50th and 75th fractile ages. Then, we use Lemma 3 to find the corresponding secondary device sizes, and determine the blocks of trigger times in a similar fashion. A heuristic solution with *four* trigger levels is found by partitioning the query distribution into *eight* equal parts (at the 12.5th, 25th, 37.5th, 50th, 62.5th, 75th, and 87.5th fractiles); a heuristic solution with *five* trigger levels is found by partitioning the query distribution into *sixteen* equal parts; so on and so forth. Clearly, this *fractile* heuristic asymptotically approaches the best policy.

The average retrieval time of a record in a database is strongly affected by the organization of the database. The methods of organizing large files vary from trees to hash-coding to linear lists; we refer the reader to Severance (1974) for an introductory discussion of their respective properties and merits. The search times then vary from $o(N)$ for a linear list to $o(\log(N))$ for balanced trees (Sahni and Horowitz, 1990; Severance, 1974). In the presence of frequent insertion and deletion operations (as would be the case for a dynamically updated database with new record arrivals), balanced tree structures are known to be difficult to maintain and retrieval times, in general, suffer. Thus, for the purpose of the numerical illustration which follows, we decided to use a linear relationship between the access times and the number of records in the secondary device.

In order to examine the sensitivity of the archiving policy with respect to the number of trigger levels and the effects of the operating environment on the performance measures, we conducted a numerical study for the case when record arrivals are random using the above proposed fractile heuristic. We considered three values of record arrival rate ($\mu = 100, 500$ and 1000) and six values of query decay rate ($b = 0.01, 0.05, 0.1, 0.2, 1.0$ and 2.0). In our numerical study, we used a linear relationship to describe the access time for a record in the secondary device as a function of the number records in the secondary device,⁶ $\tau_n = c n$, and considered two rates of access times ($c = 0.001$ and 0.00001). The access time for the tertiary device was fixed, $\bar{\tau} = 1$.

First, we consider the sensitivity of the archiving policy. The typical behavior of the average retrieval times, $E(R)$, with respect to the number of distinct trigger levels is illustrated in Fig. 3. We observe that $E(R)$ rapidly converges to its minimum value as the number of trigger levels increases. The convergence is faster for larger record arrival rates, μ and fastest for very small and very large query decay rates, b . Therefore, the heuristic is most attractive precisely for large databases in which the value content of records stays high for long periods of time and a large number of new records are added over time. The efficiency of the heuristic from a computational perspective is also evident in that a close to optimal result is obtained, for example, by just computing 16 distinct trigger times as opposed to the entire policy vector with 50,000 elements for the case when $\mu = 500$, $b = 0.01$ and $c = 0.00001$. Similar observations also hold for the effects of the number of trigger levels on hit rates, γ .

It should be pointed out that the rapid convergence of the heuristic is due to the effective selection of the policy parameters by employing the optimization results of Section 4.1. The above observations do not necessarily imply the insensitivity of $E(R)$ or γ to policy parameters since an arbitrary choice of (T_n, n) pairs results in highly suboptimal results.

Next, we consider the impact of the parameters of the operating environment on the performance of the information system. The typical behavior of the average retrieval time, $E(R)$ is illustrated in Fig. 4. $E(R)$

⁶ We like to emphasize that our analysis holds for a rather general form of τ_n as a function of n (i.e., step function, among others) as we only require that τ_n to be a non-decreasing, concave function in n .

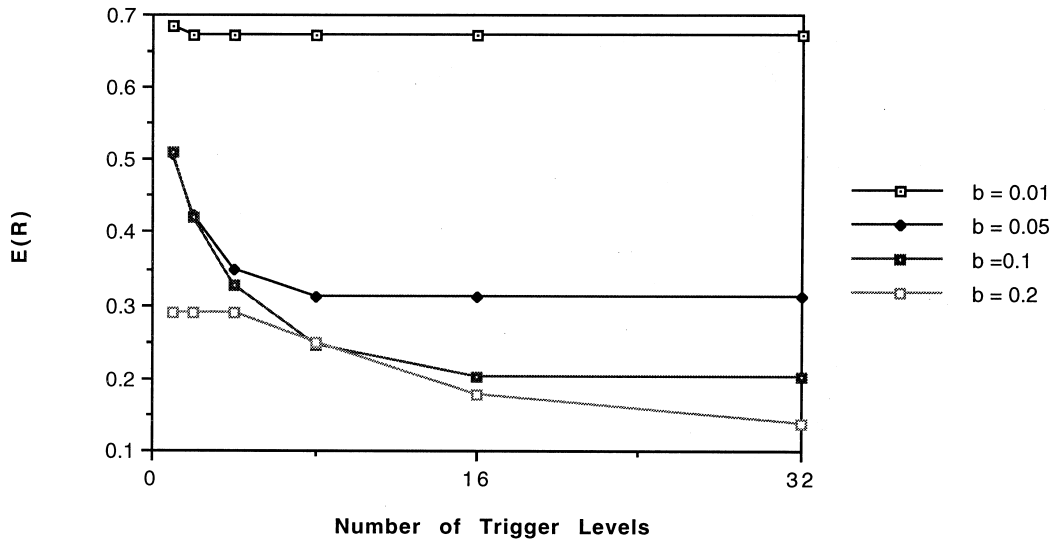


Fig. 3. The average retrieval times as the number of trigger levels are varied ($\mu = 500$ and $c = 0.00001$).

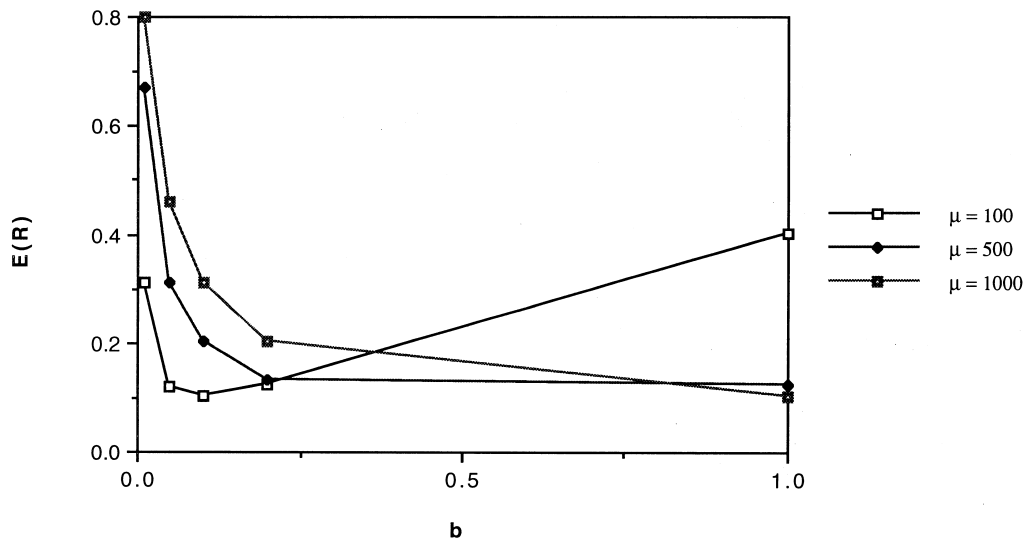


Fig. 4. Behavior of the average retrieval times, $E(R)$, versus different query decay rate, b ($c = 0.00001$).

decreases as the query request decay rate increases for small values of b , but it starts increasing for larger values of b . The curve gets flatter to the right as the record arrival rate, μ , increases. For small values of b , $E(R)$ is observed to be increasing in μ . However, for larger values of b , the reverse holds. Such behaviors can be explained as follows: As shown in Proposition 1, the trigger values minimize $E(R)$, T_n^* ($n = 1, 2, \dots$), are independent of the average growth rate of the system, μ . Thus, the number of records residing in the secondary, $E(L)$, and the average query rate for records in the system, $\mu\Gamma$, which determine the average retrieval times, $E(R)$ in Eq. (15), are increasing in μ and decreasing in b , respectively. Therefore, depending on the value of μ and b , $E(R)$ can be decreasing or increasing in μ and b .

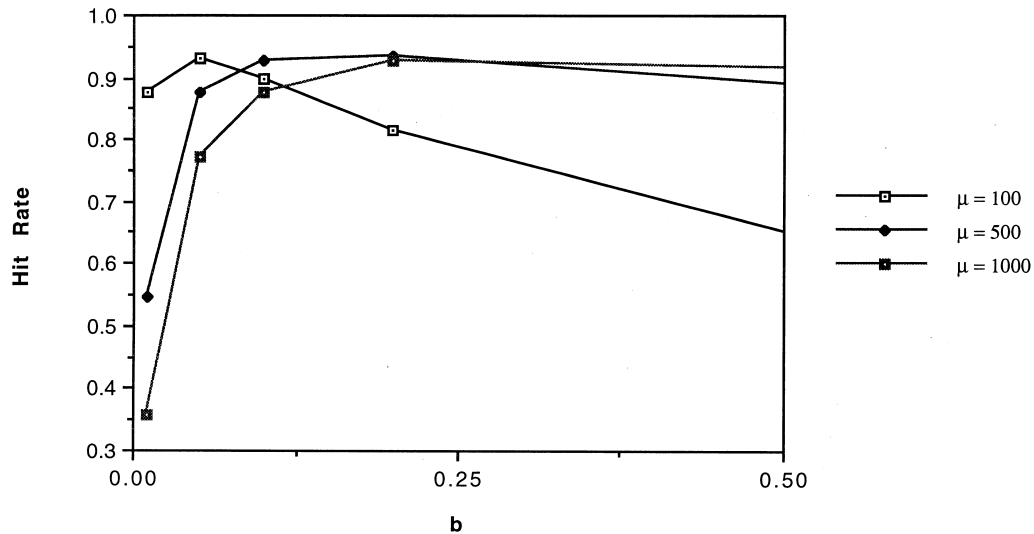


Fig. 5. Behavior of the hit rate, γ , versus the query decay rate, b ($c = 0.00001$).

Furthermore, we observed that average retrieval time is also increasing in the access time rate, c , as expected.

Fig. 5 displays the typical behavior of the corresponding hit rates, γ . The hit rate is increasing in b for small values of b , and decreasing in b , otherwise. We observe that the slope of decrease is more pronounced for small values of record arrival rate, μ . Also, the hit rate is larger for smaller values of μ when b is small. For larger values of b , the hit rate is larger for larger values of μ . These observations can be explained from the same argument given in explaining Fig. 4. Once again, we observed that the hit rate decreases as c increases, as expected.

6. Summary

In this paper, we considered an archiving model for a database maintained in secondary and tertiary storage devices. The information value of records is assumed to be decreasing in time, resulting in a lower query request intensity for a record as it ages. We proposed a 'dynamic' archiving policy based on the number of records and the age of the records in the secondary device. Under this policy, we developed two special models with constant and random inter-arrival arrival times of records. Within the theoretical framework of these models, we obtained the optimization results for minimizing the average retrieval time. In a numerical study, we tested the effectiveness of a *fractile* heuristic utilizing optimization results and examined the impact of the parameters of the operating environment on system performance.

A number of extensions of our work are possible for future research. It would be interesting to consider the case when there are multiple classes of records, either in size or other category, and to incorporate priority schemes based on class type into our archiving policy. Another important extension would be to allow for fixed costs of deletion and insertion operations which would necessitate moving records in batches. The archiving policy would have to incorporate batching decisions, as well. Finally, an interesting avenue for future research can be a simulation study of the performance of the proposed policy described in this paper in environments where one or more of the assumptions made in the paper is violated.

Acknowledgements

The authors wish to thank D. Dey and the two anonymous reviewers for their helpful comments. Funding for the first author was provided by Burlington Northern/Burlington Resources foundation.

Appendix A

Derivation of Eq. (5):

First,

$$E(L) = \mu \left\{ \tau_n \sum_{i=0}^{n-1} \int_0^{(T_n-(n-1)/\mu)^+} \theta(i/\mu + \xi) d\xi + \bar{\tau} \sum_{i=n}^{\infty} \int_0^{(T_n-(n-1)/\mu)^+} \theta(i/\mu + \xi) d\xi \right\} \\ + \mu \left\{ \tau_{n-1} \sum_{i=0}^{n-2} \int_{T_n-(n-1)/\mu}^{1/\mu} \theta(i/\mu + \xi) d\xi + \bar{\tau} \sum_{i=n-1}^{\infty} \int_{T_n-(n-1)/\mu}^{1/\mu} \theta(i/\mu + \xi) d\xi \right\}. \quad (\text{A.1})$$

Combining terms and rearranging, we get

$$E(L) = \mu \tau_{n-1} \Theta([n-1]/\mu) + \mu(\tau_n - \tau_{n-1}) \sum_{i=0}^{n-2} \int_0^{(T_n-(n-1)/\mu)^+} \theta(i/\mu + \xi) d\xi \\ + \mu \tau_n \int_0^{(T_n-(n-1)/\mu)^+} \theta([n-1]/\mu + \xi) d\xi + \mu \bar{\tau} \{ \Gamma - \Theta([n-1]/\mu) \} \\ - \mu \bar{\tau} \int_0^{(T_n-(n-1)/\mu)^+} \theta([n-1]/\mu + \xi) d\xi \\ = \mu \bar{\tau} \Gamma - \mu(\bar{\tau} - \tau_{n-1}) \Theta([n-1]/\mu) - \mu(\bar{\tau} - \tau_n) \int_0^{(T_n-(n-1)/\mu)^+} \theta([n-1]/\mu + \xi) d\xi \\ + \mu(\tau_n - \tau_{n-1}) \sum_{i=0}^{n-2} \int_0^{(T_n-(n-1)/\mu)^+} \theta(i/\mu + \xi) d\xi.$$

Derivation of the Eq. (14):

Let:

$E(L_P)$ = average number of queries residing in the system which have to be accessed from the secondary device,

$E(L_S)$ = average number of queries residing in the system which have to be accessed from the tertiary device.

Then, the average number of queries residing in the system, $E(L)$ can be written as

$$E(L) = E(L_P) + E(L_S).$$

We now find expressions for $E(L_P)$ and $E(L_S)$ using Eq. (13). First,

$$\begin{aligned}
 E(L_P) &= \sum_{n=1}^{\infty} \tau_n \left\{ \int_{x_n=0}^{T_{n+1}} \int_{x_1=0}^{x_n} \cdots \int_{x_{n-1}=x_{n-2}}^{x_n} \left[\sum_{j=1}^n \theta(x_j) \right] \mu^n \exp(-\mu T_{n+1}) dx_{n-1} \cdots dx_1 dx_n \right. \\
 &\quad \left. + \int_{x_n=T_{n+1}}^{T_n} \int_{x_1=0}^{x_n} \cdots \int_{x_{n-1}=x_{n-2}}^{x_n} \left[\sum_{j=1}^n \theta(x_j) \right] \mu^n \exp(-\mu x_n) dx_{n-1} \cdots dx_1 dx_n \right\} \\
 &= \sum_{n=1}^{\infty} \tau_n \left\{ \sum_{j=1}^n \int_{x_j=0}^{T_{n+1}} \theta(x_j) \mu \frac{[\mu x_j]^{j-1} [\mu(T_{n+1} - x_j)]^{n-j}}{(j-1)!(n-j)!} \exp(-\mu T_{n+1}) dx_j \right. \\
 &\quad + \sum_{j=1}^{n-1} \int_{x_n=T_{n+1}}^{T_n} \int_{x_j=0}^{x_n} \theta(x_j) \mu^2 \frac{[\mu x_j]^{j-1} [\mu(x_n - x_j)]^{n-j-1}}{(j-1)!(n-j-1)!} \exp(-\mu x_n) dx_j dx_n \\
 &\quad \left. + \int_{x_n=T_{n+1}}^{T_n} \theta(x_n) \mu \frac{[\mu x_n]^{n-1}}{(n-1)!} \exp(-\mu x_n) dx_n \right\}. \tag{A.2}
 \end{aligned}$$

Using the property,

$$(a+b)^n = \sum_{i=0}^n a^i b^{n-i}, \tag{A.3}$$

we can write Eq. (A.2) as

$$\begin{aligned}
 E(L_P) &= \sum_{n=1}^{\infty} \tau_n \left\{ \Theta(T_{n+1}) \mu q(n-1, \mu T_{n+1}) + \int_{T_{n+1}}^{T_n} \Theta(y) \mu^2 q(n-2, \mu y) dy \right. \\
 &\quad \left. + \int_{T_{n+1}}^{T_n} \theta(y) \mu q(n-1, \mu y) dy \right\}. \tag{A.4}
 \end{aligned}$$

Now using (Hadley and Whitin, 1963, Appendix 3),

$$\begin{aligned}
 \int_{T_{n+1}}^{T_n} \Theta(y) \mu^2 q(n-2, \mu y) dy &= \Theta(T_{n+1}) \int_{T_{n+1}}^{T_n} \mu^2 q(n-2, \mu y) dy + \int_{T_{n+1}}^{T_n} \int_x^{T_n} \theta(y) \mu^2 q(n-2, \mu y) dy \\
 &= \mu \Theta(T_{n+1}) [Q(n-1, \mu T_n) - Q(n-1, \mu T_{n+1})] + \mu [\Theta(T_n) - \Theta(T_{n+1})] Q(n-1, \mu T_n) \\
 &\quad - \mu \int_{T_{n+1}}^{T_n} \theta(x) Q(n-1, \mu x) dx. \tag{A.5}
 \end{aligned}$$

Thus, upon simplification, Eq. (A.4) will be reduced to

$$\begin{aligned}
 E(L_P) &= \sum_{n=1}^{\infty} \tau_n \left\{ \Theta(T_n) \mu Q(n-1, \mu T_n) - \Theta(T_{n+1}) \mu Q(n, \mu T_{n+1}) - \int_{T_{n+1}}^{T_n} \theta(y) \mu Q(n, \mu y) dy \right\} \\
 &= \mu \tau_1 \Theta(T_1) + \mu \sum_{n=2}^{\infty} (\tau_n - \tau_{n-1}) \Theta(T_n) Q(n-1, \mu T_n) - \mu \sum_{n=1}^{\infty} \tau_n \int_{T_{n+1}}^{T_n} \theta(y) Q(n, \mu y) dy. \quad (\text{A.6})
 \end{aligned}$$

Next, to find $E(L_S)$, first note that since the insertion process of records to the system (arrivals) is a Poisson process, the probability of having a record which has an age in the interval $(y, y + dy)$ is equal to μdy . We can now express $E(L_S)$ as

$$\begin{aligned}
 E(L_S) &= \bar{\tau} \sum_{n=0}^{\infty} \left\{ \left(\int_{y=T_{n+1}}^{\infty} \mu \theta(y) dy \right) \int_{x_n=0}^{T_{n+1}} \int_{x_1=0}^{x_n} \cdots \int_{x_{n-1}=x_{n-2}}^{x_n} \mu^n \exp(-\mu T_{n+1}) dx_{n-1} \cdots dx_1 dx_n \right\} \\
 &\quad + \bar{\tau} \sum_{n=1}^{\infty} \left\{ \int_{x_n=T_{n+1}}^{T_n} \left(\int_{y=x_n}^{\infty} \mu \theta(y) dy \right) \int_{x_1=0}^{x_n} \cdots \int_{x_{n-1}=x_{n-2}}^{x_n} \mu^n \exp(-\mu x_n) dx_{n-1} \cdots dx_1 dx_n \right\},
 \end{aligned}$$

where y is the age of the record in the archive.

Simplifying in the same fashion as before, we get

$$E(L_S) = \bar{\tau} \sum_{n=0}^{\infty} \left\{ q(n, \mu T_{n+1}) \int_{y=T_{n+1}}^{\infty} \mu \theta(y) dy \right\} + \bar{\tau} \mu \sum_{n=1}^{\infty} \int_{x=T_{n+1}}^{T_n} \left\{ \int_{y=x}^{\infty} \mu \theta(y) dy \right\} q(n-1, \mu x) dx. \quad (\text{A.7})$$

Changing, the order of integrals in the second term of Eq. (A.7), we obtain

$$\begin{aligned}
 E(L_S) &= \bar{\tau} \sum_{n=0}^{\infty} \left\{ q(n, \mu T_{n+1}) \int_{y=T_{n+1}}^{\infty} \mu \theta(y) dy \right\} \\
 &\quad + \bar{\tau} \sum_{n=1}^{\infty} \int_{y=T_{n+1}}^{T_n} \left\{ \int_{x=T_{n+1}}^y \mu \frac{(\mu x)^{n-1}}{(n-1)!} \exp(-\mu x) dx \right\} \mu \theta(y) dy \\
 &\quad + \bar{\tau} \sum_{n=1}^{\infty} \left\{ \int_{y=T_n}^{\infty} \mu \theta(y) dy \right\} \left\{ \int_{x=T_{n+1}}^{T_n} \mu \frac{(\mu x)^{n-1}}{(n-1)!} \exp(-\mu x) dx \right\}.
 \end{aligned}$$

Hence,

$$\begin{aligned}
E(L_S) = & \bar{\tau} \sum_{n=0}^{\infty} \left\{ q(n, \mu T_{n+1}) \int_{y=T_{n+1}}^{\infty} \mu \theta(y) \, dy \right\} \\
& + \mu \bar{\tau} \sum_{n=1}^{\infty} \int_{T_{n+1}}^{T_n} \theta(y) [Q(n, \mu y) - Q(n, \mu T_{n+1})] \, dy \\
& + \mu \bar{\tau} \sum_{n=1}^{\infty} \left\{ [Q(n, \mu T_n) - Q(n, \mu T_{n+1})] \int_{T_n}^{\infty} \theta(y) \, dy \right\},
\end{aligned}$$

which, upon further simplification, reduces to

$$E(L_S) = \mu \bar{\tau} [\Gamma - \Theta(T_1)] + \mu \bar{\tau} \sum_{n=1}^{\infty} \int_{y=T_{n+1}}^{T_n} \theta(y) Q(n, \mu y) \, dy. \quad (\text{A.8})$$

Combining Eqs. (A.6) and (A.8), we get

$$\begin{aligned}
E(L) = & \mu \bar{\tau} \Gamma - \mu (\bar{\tau} - \tau_1) \Theta(T_1) + \mu \sum_{n=2}^{\infty} (\tau_n - \tau_{n-1}) \Theta(T_n) Q(n-1, \mu T_n) \\
& + \mu \sum_{n=1}^{\infty} (\bar{\tau} - \tau_n) \int_{y=T_{n+1}}^{T_n} \theta(y) Q(n, \mu y) \, dy,
\end{aligned}$$

which is Eq. (14).

Appendix B

Summary of key notations

$\theta(x)$	Instantaneous arrival rate of queries for a record that is aged x since its insertion in the system.
τ_n	Average access (retrieval) time of a record from the secondary device with a total of n records.
$\bar{\tau}$	Average access (retrieval) time of a record from the tertiary storage device.
$1/\mu$	Average inter-arrival time of the records to the system.
x_n	Age of the n th record in the secondary device.
$\mathbf{x}_n = \{x_1, \dots, x_n\}$	Number and the ages of the records in the secondary device.
$\mathbf{T} = \{T_n; n > 0\}$	Archival policy.
$p_n(x_1, \dots, x_n)$	Steady state probability density of the system being in state \mathbf{x}_n .
P_n	Steady state probability of having $n(n \geq 1)$ records in the secondary device.
L	Number of queries residing in the system.
R	System's record retrieval time.
γ	Fraction of queries accessed from the secondary device (i.e., hit rate).
$E(\cdot)$	Expected value of a random variable.

References

- Brancheau, J.C., Wetherbe, J.C., 1977. Key issues in information systems management. *MIS Quarterly* 1 (1), 23–44.
- Brown, D., 1994. Disk historian turns obsolete hard disk files into memories. *InfoWorld* 16 (42), 136.
- Cohen, E.I., King, G.M., Brady, J.T., 1989. Storage hierarchies. *IBM Systems Journal* 28 (1), 62–76.
- Considine, J.P., Myers, J.J., 1977. MARC: MVS archival storage and recovery program. *IBM Systems Journal* 4, 378–397.
- Cox, D.R., 1955. The analysis of non-Markovian stochastic processes by the inclusion of supplementary variables. *Proceedings of the Cambridge Philosophical Society* 51, 441–443.
- Dickson, G.W., Nechis, M., 1984. Key information systems issues for the 1980s. *MIS Quarterly* 8 (3), 135–159.
- Gecsei, J., Lukes, J.A., 1974. A model for the evaluation of storage hierarchies. *IBM Systems Journal* 13 (2), 163–178.
- Gnedenko, B., Kovalenko, I.N., 1968. *Introduction to Queueing Theory*, Israel Program for Scientific Translations, Jerusalem.
- Gravina, C.M., 1978. National Westminster Bank mass storage archiving. *IBM Systems Journal* 17 (4), 344–358.
- Hadley, G., Whitin, T.M., 1963. *Analysis of Inventory Systems*, Prentice Hall, Englewood Cliffs, NJ.
- Han, B., Diehr, G., 1991. An algorithm for storage device selection and file assignment. *European Journal of Operational Research* 61, 326–344.
- Harding, W.B., Clark, C.M., Gallo, C.L., Tang, H., 1990. Object storage hierarchy Management. *IBM Systems Journal* 29 (3), 384–397.
- Klastorin, T.D., Moinszadeh, K., Diehr, G., Han, B., 1993. Optimal file management in a hybrid storage system. *European Journal of Operational Research* 64, 370–383.
- Knuth, D., 1973. *The Art of Computer Programming: Sorting and Searching*, Addison-Wesley, Reading, Mass.
- Lawrie, D.H., Randal, J.M., Barton, R.R., 1982. Experiments with automatic file migration. *Computer* 15, 45–55.
- Large Storage Configurations, 1995. Inc., Storage Server Market Overview.
- Lum, V.Y., Senko, M.E., Wang, C.P., Ling, H., 1975. A cost oriented algorithm for data set allocation in storage hierarchies. *Communication of the ACM* 18 (6), 318–322.
- Moinszadeh, K., 1989. Operating characteristics of the (S-1, S) inventory system with partial backorders and constant resupply times. *Management Science* 4, 472–477.
- Nance, B., 1995. Network storage economizers. *Byte* 20 (3), 137–142.
- Niederman, F., Brancheau, J.C., Wetherbe, J.C., 1991. Information systems management issues for the 1990s. *MIS Quarterly* 15 (1), 15–25.
- Ryan, A.J., 1994. Kick the hard disk habit. *Datamation* 40 (23), 62–65.
- Sahni, E., Horowitz, S., 1990. *Data Structures in Pascal*, Third Edition, Computer Science Press, NY.
- Schmidt, C.P., Nahmias, S., 1985. (S-1, S) policies for perishable inventory. *Management Science* 31, 719–728.
- Severance, D.G., 1974. Identifier search mechanisms: A survey and generalized model. *Computer Surveys* 6 (3), 175–194.
- Smith, A.J., 1981. Long term file migration: Development and evaluation of algorithms. *Communications of the ACM* 24 (8), 521–532.
- Stidham, S., 1972. $L = \lambda W$: A discounted analogue and a new proof. *Operations Research* 18, 1115–1126.
- Szajna, B., 1994. How much is information systems research addressing key practitioner concerns? *Database* 25 (2), 49–59.
- Tanton, N.E., 1979. Teletex – evaluation and potential. *IEEE Transactions on Consumer Electronics* 25, 246–250.
- Thanhardt, E., Harano, G., 1988. File Migration in the NCAR Mass Storage System, Digest of Papers. Ninth IEEE Symposium on Mass Storage Systems. *Storage Systems: Perspectives*, pp. 114–121.
- Ullman, J.D., 1988. *Data Base and Knowledge-Base Systems: Classical Database Systems*, Computer Science Press, Rockville, MD.