The Average Time Until Bucket Overflow

ROBERT B. COOPER AND MARTIN K. SOLOMON
Florida Atlantic University

It is common for file structures to be divided into equal-length partitions, called buckets, into which records arrive for insertion and from which records are physically deleted. We give a simple algorithm which permits calculation of the average time until overflow for a bucket of capacity n records, assuming that record insertions and deletions can be modeled as a stochastic process in the usual manner of queueing theory. We present some numerical examples, from which we make some general observations about the relationships among insertion and deletion rates, bucket capacity, initial fill, and average time until overflow. In particular, we observe that it makes sense to define the *stable point* as the product of the arrival rate and the average residence time of the records; then a bucket tends to fill up to its stable point quickly, in an amount of time almost independent of the stable point, but the average time until overflow increases rapidly with the difference between the bucket capacity and the stable point.

Categories and Subject Descriptors: C.4 [**Computer Systems Organization**]: Performance of Systems—*modeling techniques*; D.4.8 [**Operating Systems**]: Performance—*queueing theory*; H.2.2 [**Database Management**]: Physical Design—*access methods*; H.2.7 [**Database Management**]: Database Administration—*logging and recovery*

General Terms: Performance

Additional Key Words and Phrases: File structures, bucket overflow

1. INTRODUCTION

It is common for file structures to be divided into equal-length partitions, called *buckets*, into which records are inserted and from which they are deleted. In this paper we study the average time it takes for a bucket to overflow (i.e., the average time until, for the first time, a record arrives for insertion into a bucket when the bucket is full). Thus, this paper provides a measure of the time it takes for the structure of certain such files to start degenerating.

We briefly describe our mathematical model. In this model records *arrive* for insertion into a bucket according to a renewal process; that is, the interarrival times are independent, identically distributed, nonnegative random variables, with an otherwise arbitrary distribution. If an arriving record finds the bucket filled to capacity, the record *overflows*; otherwise, it is inserted into the bucket, where it remains for a duration of time, called the *residence time* (or, in queueing theory parlance, the *service time*), at the expiration of which the record is

Authors' address: Dept. of Computer and Information Systems, Florida Atlantic University, Boca Raton, FL 33431.

physically deleted from the bucket. (Hence, when a record is deleted, the slot in which it was held becomes available to hold another record. The case in which a logical deletion does not result in such a physical deletion is mathematically trivial: The time until first overflow from a bucket which initially has room for k additional records is simply the time until the $(k + 1)$th record arrives for insertion.) We assume that the residence times are identical, exponentially distributed random variables, independent of each other and of the arrival process. We calculate, for a bucket which initially contains i records and has capacity n (i.e., has room to hold simultaneously a maximum of n records), the average length of time until the first overflow occurs (see the end of Section 2 for immediate generalizations). Although our model can handle any renewal process as a description of the interarrival times of the records, we restrict our numerical calculations to a particular class of interarrival-time distributions whose effect is to allow records to arrive in *batches*, with the number of records in a batch having a geometric distribution with mean value $1/p$, and with the batch interarrival times having an Erlangian (or gamma) distribution of arbitrary order ν. (See [10], Section 6.4 for a discussion of the Erlangian distribution in a similar context.) In particular, when $\nu = 1$ the batches arrive according to a Poisson process, which is the most common and least presumptive assumption; and when $p = 1$, each batch contains exactly one record. Most of our numerical results will be restricted to this case ($\nu = 1, p = 1$). Also, we will consider some cases where $\nu > 1$, which corresponds to lower variability than the Poisson process (when $\nu = \infty$, the interarrival times are constant), and where $p < 1$, which corresponds to records arriving in batches.

To place our study in context, we note that a similar model was used by Larson [6], who calculated the distribution of the number of records in an indexed sequential bucket at any time t (which does not give any direct information about the length of time until an overflow occurs), under the assumptions that the records arrive one-at-a-time according to a Poisson process (the most important special case in our model), and, as in our model, stay in the bucket for residence times that are exponentially distributed. (Although it is not pointed out in [6], Larson did not need the exponential residence-time assumption; however, this assumption cannot be relaxed in our calculation.)

Heyman [3] considers the same question as we do, but his model is different from ours. Specifically, the essential difference (there are others) is that in Heyman's model (which is a discrete-parameter random walk model) the rate at which deletions occur is constant, independent of the number of records in the bucket when the transaction (deletion) occurs. In contrast, in our model (which is a continuous-parameter queueing model), the (instantaneous) deletion rate is proportional to the number of records in the bucket. (Thus, for example, in our model a transaction that occurs when there are 20 records in the bucket is about twice as likely to be a deletion as it would be if there were only 10 records in the bucket; in Heyman's model these probabilities are the same.) Consequently, numbers derived from our model are incommensurate with those derived from Heyman's.

Also, Heyman gives a formula [3, p. 617] for the approximate calculation of the expected value of the number $N(t)$ of overflows to occur by (what is equivalent to) time t. He shows that $E(N(t))$ consists (approximately) of a "delay" followed by a linear dependence on t; that is, in Heyman's notation, $E(N(t)) \doteq 0$ when $t < \eta_1$ and $E(N(t)) \doteq (t - \eta_1)\mu$ when $t \geq \eta_1$, where η_1 is (equivalent to) the mean time until first overflow from a bucket of capacity n, and μ is the difference between the probability that an arbitrary transaction is an insertion and the probability that it is a deletion. By the elementary renewal theorem, this formula can also be used in our model if we replace Heyman's μ by the reciprocal of the mean time between overflows (i.e., $1/E(T_n)$, where $E(T_n)$ is calculated according to our algorithm from Eq. (4)) and replace η_1 by the mean time for the bucket to fill to capacity (i.e., $E(T(n-1, i))$, which is calculated in Eq. (5)).

Finally, Heyman gives a formula [3, p. 617] for the optimal value T^* of the time at which the database should be reorganized, as a function of the cost of reorganization versus the cost of processing a transaction when the bucket is full. Again, Heyman's results hold for our model if we make the same substitutions for Heyman's η_1 and μ. Hence, Heyman's analysis and ours are compatible; the only difference is that our model for the calculation of the fundamental quantity, the average time until the bucket first overflows, is different (arguably, more general and realistic). For further discussion on motivation and background for this study, the reader is referred to Heyman [3], to Larson [6], and the references cited by these authors.

In Section 2 of this paper we describe the model in more detail, give its analytic solution, and discuss its applicability. In Section 3 we present some numerical results and some conclusions suggested by these results.

As a practical matter, then, our results provide a database administrator with information that helps in making rational decisions concerning the choice of values for reorganization time, bucket capacity, and initial fill. Of course, these decisions will be influenced by other considerations as well, such as the fact that block size may tightly constrain bucket size. This latter consideration can complicate the treatment of bucket capacity as an adjustable parameter since, in order to increase bucket capacity, one might have to adopt record compression, leading to variable-length records.

For the reader whose only interest is the algorithm by which we calculate the expected value of the time $T(n, i)$ to overflow a bucket of capacity n records whose initial fill is i records, we remark that the required expected value $E(T(n, i))$ is given in Section 2 by Eq. (10), complemented by Eqs. (8) and (9), where the term $\gamma_0(s)$ that initializes the recurrence (9) is the Laplace–Stieltjes transform of the distribution function of the interarrival times of the records. The particular choice for $\gamma_0(s)$ that we have used for the numerical examples of Section 3 is given by Eq. (16), where λ^{-1} is the mean interarrival time of the batches, measured in units of mean residence time. In the case of ordinary one-at-a-time Poisson arrivals of records, the algorithm simplifies considerably, so that Eqs. (8) and (9) can be replaced by Eqs. (17) and (18), with (16) given by $\gamma_0(s) = \lambda/(\lambda + s)$.

2. THE MATHEMATICAL MODEL

We assume (i) that records arrive for insertion according to a renewal process (that is, the interarrival times are independent, identically distributed, nonnegative random variables) with distribution function, say, $G_0(t)$; and (ii) that the residence times are identical, exponentially distributed random variables, independent of each other and of the arrival process. Let $G_j(t)$ be the distribution function of the duration of time separating the first instant at which the bucket contains j ($j = 0, 1, 2, \ldots, n$) records and the arrival epoch of a new record whose insertion would, for the first time, cause the bucket to contain $j + 1$ records. (If the bucket has capacity n, then $G_n(t)$ is the distribution function of the duration of time separating the first instant at which the bucket becomes full and the first instant at which a record overflows the bucket.) Let $\gamma_j(s)$ be the Laplace–Stieltjes transform of $G_j(t)$; that is,

$$\gamma_j(s) = \int_0^\infty e^{-st}\, dG_j(t) \qquad (j = 0, 1, 2, \ldots, n). \tag{1}$$

Then, it follows that

$$\gamma_j(s) = \frac{\gamma_{j-1}(s+\mu)}{1 - \gamma_{j-1}(s) + \gamma_{j-1}(s+\mu)} \qquad (j = 1, 2, \ldots, n), \tag{2}$$

where μ is the reciprocal of the mean residence time. (For a discussion of queueing theory and related topics, such as Laplace–Stieltjes transforms, see Cooper [2].)

The recurrence (2) was first found by Palm [7] (see also [2, p. 207]) in the context of a queueing model with exponential service times, in which the servers are numbered 1, 2, ..., and each customer is served by the lowest-numbered server that is idle when the customer arrives (in our context, record slots in a bucket can be thought of as servers, and the residence time of a record in a slot as the service time). Then Eq. (2) gives the Laplace–Stieltjes transform $\gamma_j(s)$ of the distribution function $G_j(t)$ of the interoverflow times from the jth server in terms of the corresponding transform for the $(j-1)$st server. It follows from the assumption of exponential service times that $G_j(t)$ is also the distribution function of the duration of time between any instant at which the jth server is seized and the next instant at which an overflow occurs from that server.

For our application, let us define the random variable $T(n, i)$ to be the elapsed time until first overflow from a bucket of capacity n records whose initial fill is i records, and T_j to be the elapsed time until a bucket that for the first time contains j records will next contain $j + 1$ records (or overflow, if $j = n$). Then, clearly, T_j has the distribution function $G_j(t)$, with Laplace–Stieltjes transform $\gamma_j(s)$, and

$$T(n, i) = \sum_{j=i}^{n} T_j. \tag{3}$$

Since the expected value of T_j is given by

$$E(T_j) = -\gamma_j'(0) \qquad (j = 0, 1, 2, \ldots, n), \tag{4}$$

it follows from (3) that the expected value of the time until first overflow from a bucket of capacity n whose initial fill is i is

$$E(T(n, i)) = -\sum_{j=i}^{n} \gamma_j'(0). \tag{5}$$

The derivatives on the right-hand side of (5) can be calculated in a straightforward manner from (2); the result is

$$\gamma_j'(0) = \frac{\gamma_{j-1}'(0)}{\gamma_{j-1}(\mu)} \qquad (j = 1, 2, \ldots, n). \tag{6}$$

We can, without loss of generality, take the mean residence time μ^{-1} to be the basic time unit, that is, $\mu = 1$. If we set

$$m_j = -\gamma_j'(0) \qquad (j = 0, 1, 2, \ldots, n), \tag{7}$$

then our computational problem reduces to calculating each value m_j from the previous values m_{j-1} and $\gamma_{j-1}(1)$; this is done using the recurrence relations (6) and (2), which become, in the present notation,

$$m_j = \frac{m_{j-1}}{\gamma_{j-1}(1)} \qquad (j = 1, 2, \ldots, n;\ m_0 = -\gamma_0'(0)), \tag{8}$$

and

$$\gamma_j(k) = \frac{\gamma_{j-1}(k+1)}{1 - \gamma_{j-1}(k) + \gamma_{j-1}(k+1)} \qquad (j, k = 1, 2, \ldots). \tag{9}$$

(Note that m_0 equals the mean interarrival time of the records; when records arrive in a batch of size b, say, then m_0 includes the $b - 1$ "interarrival" intervals of length zero.) If we rewrite (5) in the present notation,

$$E(T(n, i)) = \sum_{j=i}^{n} m_j \qquad (i = 0, 1, 2, \ldots, n), \tag{10}$$

then we can conclude that, for a bucket of any capacity n and any initial loading i, the mean time $E(T(n, i))$ until overflow first occurs is calculated in a straightforward manner from (8), (9), and (10); and the results are valid for any arrival process that is a renewal process characterized by interarrival times with any specified distribution function $G_0(t)$, as long as the residence times are exponentially distributed.

Similarly, since the random variables $T_0, T_1, T_2, \ldots$ are mutually independent, the variance of the sum (3) is given by

$$V(T(n, i)) = \sum_{j=i}^{n} v_j, \tag{11}$$

where

$$v_j = \gamma_j''(0) - (\gamma_j'(0))^2. \tag{12}$$

The second derivative $\gamma_j''(0)$ on the right-hand side of (12) can be calculated from the recurrence

$$\gamma_j''(0) = \frac{\gamma_{j-1}''(0)}{\gamma_{j-1}(\mu)} + \frac{2\gamma_{j-1}'(0)}{\gamma_{j-1}^2(\mu)} (\gamma_{j-1}'(0) - \gamma_{j-1}'(\mu)), \tag{13}$$

which follows by successive differentiation of (2). (Again, without loss of generality, we can take $\mu = 1$ and use (8) and (9).) Formulas (11) through (13) can be used to approximate, by application of the central limit theorem, the distribution (as opposed to only the mean value) of the time until overflow.

For the purposes of our application, we consider a family of input processes characterized by the distribution function

$$G_0(t) = 1 - p + pF_\nu(t), \tag{14}$$

where $0 < p \leq 1$, and $F_\nu(t)$ is a gamma distribution of order ν. (See, for example, [2, p. 64].) Since $F_\nu(0) = 0$, it follows from (14) that when $p = 1$ arrivals occur one-at-a-time with the times between arrivals having the distribution function $F_\nu(t)$. When $p < 1$ there is a positive probability $G_0(0) = 1 - p > 0$ that an interarrival time is zero; with the probability $(1 - p)^{m-1}p$ there will be exactly $m - 1$ ($m = 1, 2, \ldots$) successive interarrival times, each of which has duration zero. We can interpret this to mean that exactly m records arrived simultaneously. That is, our model allows records to arrive in batches, where the number of records in a batch has the geometric distribution with mean value $1/p$; then $F_\nu(t)$ is the distribution function of the time intervals separating the arrival epochs of the batches.

If we let λ be the rate at which the arrival epochs occur (counting the multiple simultaneous arrival instants of the records that comprise a batch as a single arrival epoch), then the Laplace–Stieltjes transform $\phi(s)$ of the gamma distribution function $F_\nu(t)$ which describes the time intervals between the arrival epochs of the batches is

$$\phi(s) = \left(\frac{\nu\lambda}{\nu\lambda + s}\right)^{\nu}. \tag{15}$$

Then, it follows from (14), (15), and (1) that

$$\gamma_0(s) = 1 - p + p\left(\frac{\nu\lambda}{\nu\lambda + s}\right)^{\nu}. \tag{16}$$

For the numerical examples presented in this paper we use (16), with several different choices for the parameter values p, ν, and λ as the description of the arrival process (which, since it provides the starting point for the recurrences (8) and (9), allows calculation of the mean time until overflow, given by (10)).

When ν is a positive integer, the gamma distribution function $F_\nu(t)$ is that of a sum of ν independent, identically distributed, exponential random variables, and is commonly referred to as the Erlangian distribution function of order ν. In particular, the Erlangian distribution function of order $\nu = 1$ is simply the exponential, which corresponds to the case in which the arrival epochs of the batches follow a Poisson process. When ν increases, the variance of the interarrival times decreases, approaching 0 as ν approaches ∞. When ν decreases, this variance increases, approaching ∞ as ν approaches 0. Thus, (16) provides a model for a class of input processes, ranging from very high to very low variance of batch interarrival times, with the batch size having a geometric distribution.

In particular, when $p = 1$ and $\nu = 1$, (16) describes the case that is mathematically simplest and probably the most realistic, namely, records arriving one-at-a-time according to a Poisson process. It can be shown that, for this important case, Eqs. (8) and (9) yield (after considerable but straightforward algebra) the simple formula

$$m_j = \frac{1}{\lambda B(j, \lambda)} \qquad (j = 0, 1, 2, \ldots, n), \tag{17}$$

where $B(j, \lambda)$ is the well-known Erlang B formula (see [2, p. 80]). The right-hand side of (17) is easily calculated from the recurrence

$$B(j, \lambda) = \frac{\lambda B(j-1, \lambda)}{j + \lambda B(j-1, \lambda)} \qquad (j = 1, 2, \ldots; B(0, \lambda) = 1) \tag{18}$$

(see [2, p. 82]). (The result (17) could have been anticipated on intuitive grounds: $\lambda B(j, \lambda)$ is the rate at which customers overflow the jth server; its reciprocal is the mean time between such overflows.)

At this point, we digress briefly to comment on Larson's model [6]. Larson's basic calculation is the probability $P_i(x, t)$ (given explicitly by Larson's Eq. (3)) that a bucket (of infinite capacity) which initially contains i records will, at time t, contain x records. Knowledge of $P_i(x, t)$ gives no direct information about how long it took for the number of records in the bucket to reach the value x, because the bucket might have contained *any* number of records prior to time t; in other words, $P_i(x, t)$ contains no direct information about how long it would take for a bucket of finite capacity to overflow. (However, it can be used, of course, to obtain such results as the time when the average number of records in an infinite bucket is any specified value.) Larson's model is similar to ours in that it assumes Poisson arrivals of records (which, in our model, is the special case where $p = 1$ and $\nu = 1$ in Eq. (16)) and exponential residence times. Interestingly, although our model cannot be generalized (at least by us) to handle residence times with an arbitrary distribution, Larson's can (as long as the arrival process is restricted to be Poisson). The general version of Larson's Eq. (3) is

$$P_i(x, t) = \sum_{k=0}^{\min(i,x)} P(i, k; t) P_0(x - k, t), \tag{19}$$

where $P(i, k; t)$ is the probability that k of the original i records are still in the bucket at time t. This latter probability is (clearly) given by the binomial distribution

$$P(i, k; t) = \binom{i}{k} (1 - H(t))^k (H(t))^{i-k}, \tag{20}$$

where $H(t)$ is the residence-time distribution function. $P_0(x - k, t)$ is the probability that exactly $x - k$ records, none of which was present initially, are in the bucket at time t; it is given by the Poisson distribution

$$P_0(j, t) = \frac{(\lambda t p(t))^j}{j!} e^{-\lambda t p(t)} \qquad (j = 0, 1, 2, \ldots), \tag{21}$$

where

$$p(t) = 1 - H(t) + \int_0^t \frac{y}{t}\, dH(y). \tag{22}$$

(A derivation of Eqs. (21) and (22) is given in [2, pp. 86–87, 119].)

We conclude this section by commenting on the applicability of our model. For purposes of computation, the Palm model assumes that we always place an incoming record into the lowest-numbered slot available at that time. Since we are interested in bucket overflow, our use of the Palm model obviously does not presume any particular scheme for record placement within a bucket, as long as a record will overflow if and only if there is no room in the home bucket. Thus, for example, our model obviously applies to the case in which the records are maintained in sort-order within a bucket.

Similarly, by treating a chain of buckets as a single bucket, our model can be used to study the time until a kth overflow bucket is created (where an overflow bucket need not be the same size as a prime bucket), as long as a kth overflow bucket is created when and only when a new record arrives and there is no room for the record in any of the previous $k - 1$ buckets on the chain. Thus our model directly yields the average time until a kth overflow bucket is created for hash structures, as in INGRES [9], as well as hash structures which maintain the collision chains as sorted lists. However, our model is not directly applicable to open addressing collision resolution schemes [5], since overflows from preceding buckets cause "arrivals" not accounted for in our mathematical model.

Finally, we consider the applicability of our model to indexed sequential files. The application of our model presumes the ability to compute or estimate λ, the arrival rate to a bucket. In the structures discussed thus far, the arrival rate to a bucket can be taken as a fraction R of the arrival rate to the file, where R is the same for each bucket (namely, the reciprocal of the number of buckets in the file). In the case of indexed sequential files, however, R is a random variable whose value is determined by the keys of the records in the file at load time. If all the buckets are statistically identical, the expected value $E(R)$ is still the same for each bucket, but in any particular case the value of R will, in general, differ from its expected value. If the number of records in the file at load time is large enough, the resulting value of R will be close to its expected value, and no substantial numerical error will be introduced by treating indexed sequential files the same as hash files. However, if one is not willing to assume that the initial number of records in the file is "large enough," the problem can still be overcome, but at the cost of doing some numerical integration. In particular, let $F_R(x)$ be the distribution function of R. Then

$$E(T(n, i)) = \int_0^1 E(T(n, i) \mid R = x)\, dF_R(x),$$

where $E(T(n, i) \mid R = x)$ denotes the expected time to overflow of a bucket whose arrival rate λ is the fraction x of the arrival rate (assumed known) to the whole file.

In order to evaluate this integral, however, one needs an expression for $F_R(x)$. This question has been addressed by Keehn and Lacy [4] and Batory [1]. (The reader should note that the expression for $F_R(x)$ given by Keehn and Lacy [4, p. 196, Eq. (6)] differs from that given by Batory [1, p. 80, first equation], presumably due to different assumptions concerning the last bucket in the file.)

3. NUMERICAL EXAMPLES

In this section we present some tables and curves calculated according to the formulas given in the preceding section, and we discuss some interpretations and observations suggested by the calculations.

Table I shows the values of $\bar{T}(n) = E(T(n, 0))$, the expected value of the time $T(n, 0)$ until overflow of a bucket of capacity n that is initially empty, as a function of the arrival rate λ, for different values of these parameters, under the assumption that the records arrive one-at-a-time ($p = 1$) according to a Poisson process ($\nu = 1$) (and, as always in our model, residence times are exponentially distributed with mean 1). Some data are presented in graphical form in Figure 1. From these tables and graphs one can calculate $E(T(n, i))$, the expected value of the time $T(n, i)$ until overflow for a bucket whose initial fill is i records, by using the formula

$$E(T(n, i)) = \bar{T}(n) - \bar{T}(i - 1) \qquad (i = 0, 1, 2, \ldots, n;\ \bar{T}(-1) = 0). \quad (23)$$

In Table II we compute the mean overflow times from an initially empty bucket of capacity n for the case in which the records arrive in batches and the arrival times of the batches follow a Poisson process. In this example we have taken the mean batch sizes $p^{-1} = 1, 2, 4, 5, 10, 20, 100$; in each case λ has been chosen so that the overall arrival rate of the records (not the batches) is 10 records per unit time (i.e., $\lambda = 10/p^{-1}$ in Eq. (16)). The first column ($p = 1$) is the same as that of Table I for $\lambda = 10$; the entries for increasing values of p^{-1} show how the mean time to overflow decreases as the mean batch size increases, all other things being equal.

In Table III we compute the mean overflow times from an initially empty bucket of capacity n for the case in which the batch size is always one ($p = 1$) and the interarrival times have the Erlangian distribution of order $\nu = 1, 2, 5, 10, 20, 50, \infty$; again, we have chosen to keep the arrival rate of the records at 10 records per unit time (i.e., $\lambda = 10$ in Eq. (16)). (We have chosen not to include any calculations for the case $0 < \nu < 1$, which models an input process that has higher variability than the Poisson process, because the same effect is obtained by increasing p^{-1}, the mean batch size.) Again, the first column ($\nu = 1$) is the same as that of Table I for $\lambda = 10$; the entries for increasing values of ν show how the mean time to overflow increases as the variance of the interarrival times decreases. (When $\nu = \infty$, the Laplace–Stieltjes transform (16) becomes $\gamma_0(s) = e^{-s/\lambda}$, which corresponds to constant interarrival times of length λ^{-1}.)

We remarked earlier that we could assume $\mu = 1$ with no loss of generality. To see this, observe that for fixed n and i the mean time until overflow is a function of λ and τ, say $f(\lambda, \tau)$, where τ is the mean residence time, and, in general, $\tau = \mu^{-1}$. Then, since the time scale is arbitrary, it follows that $f(\lambda, \tau) = \tau f(\lambda\tau, 1)$.

Hence it follows from the data in Table I that for an arrival rate of $\lambda = 4$ records per day and a mean residence time of $\tau = 2.5$ days, say, the mean time until overflow from a bucket of capacity 15 whose initial fill is 10 is $2.5(\bar{T}(15) - \bar{T}(9)) \approx 19.0$ days.

Larson [6] defines a *stable file* as one in which "both insertions and deletions occur, but the number of records in the file remains more or less unchanged." More precisely, Larson's stable file is equivalent to an infinite-capacity bucket for which $i = \lambda$; that is, the bucket initially contains exactly $\lambda\tau$ ($\tau = 1$) records, which is the limiting value as $t \to \infty$ of the expected number of records in an infinite-capacity bucket to which records arrive one-at-a-time according to a Poisson process with arrival rate λ and mean residence time $\mu^{-1} = 1$. Let us call this long-run expected value the *stable point*. (Note that the stable point is *not* the long-run expected value of the number of records in a bucket, unless the records that overflow are queued and put into the bucket when space becomes available. For a bucket of finite capacity n for which overflow records are forever lost, the long-run mean number of records in the bucket is $a(1 - B(n, a))$, where $a = \lambda\tau$ and $B(n, a)$ is given by Eq. (18).) From our calculations we make the interesting observation that the mean time required for an empty bucket to reach its stable point (as long as its capacity exceeds the stable point) is a relatively small number which grows very slowly with λ. For example, according to Table I, when $5 \leq \lambda \leq 30$, the mean time for an initially empty bucket to reach its stable point is about 2.5 residence times.

We now use our data to illustrate the relative effect on mean time until overflow of changing bucket capacity and free space within a bucket when the bucket's capacity exceeds its stable point. According to Table I, if records arrive at the rate of 10 per day for insertion into a bucket of capacity 20, and the average residence time of a record is 1 day, then the expected time until overflow is $\bar{T}(20) \approx 115.9$ days if the bucket is initially empty; however, if the bucket initially contains 19 records, then the mean time until overflow is $\bar{T}(20) - \bar{T}(18) \approx 115.9 - 35.7 = 80.2$ days.

Observe that the mean time to overflow increases very quickly with n when $n > \lambda$. For example, if the capacity of an initially empty bucket (with arrival rate $\lambda = 10$) were decreased from $n = 20$ to $n = 10$ (which would decrease the bucket arrival rate to $\lambda = 5$, assuming the file size remains fixed), the mean time until overflow would decrease from 115.9 days to 24.1 days.

From the above we see that a decrease in the amount of initially available space in a bucket that corresponds to a decrease in bucket size from 20 records to 10 records yields a greater decrease in mean overflow time (even accounting for a corresponding increase in the number of buckets) than would occur if the 20-record bucket were instead initially loaded with 19 records. It is clear that in general bucket capacity is more important than initial fill in determining time to overflow (however, see [8] for the negative effect a bucket capacity of $n > 1$ can have on hash-structured files which are subject to certain physical storage assumptions).

Finally, we mention how the central limit theorem can be applied to estimate the distribution (as opposed to only the mean) of the time until bucket overflow.

Table I.

λ \ n	5	10	15	20	25	30	35	40	45	50
0	0.20	0.10	0.07	0.05	0.04	0.03	0.03	0.03	0.02	0.02
1	0.44	0.21	0.14	0.10	0.08	0.07	0.06	0.05	0.04	0.04
2	0.74	0.33	0.21	0.16	0.12	0.10	0.09	0.08	0.07	0.06
3	1.11	0.47	0.30	0.22	0.17	0.14	0.12	0.10	0.09	0.08
4	1.62	0.62	0.38	0.28	0.22	0.18	0.15	0.13	0.12	0.10
5	2.32	0.80	0.48	0.34	0.27	0.22	0.18	0.16	0.14	0.13
6	3.36	1.01	0.59	0.41	0.32	0.26	0.22	0.19	0.17	0.15
7	5.02	1.25	0.70	0.49	0.37	0.30	0.25	0.22	0.19	0.17
8	7.87	1.55	0.83	0.57	0.43	0.35	0.29	0.25	0.22	0.20
9	13.21	1.91	0.97	0.65	0.49	0.39	0.33	0.28	0.25	0.22
10	24.09	2.38	1.14	0.75	0.56	0.44	0.37	0.32	0.28	0.24
11	48.23	2.99	1.32	0.85	0.62	0.49	0.41	0.35	0.30	0.27
12	106.35	3.83	1.54	0.96	0.70	0.55	0.45	0.38	0.33	0.30
13	257.66	5.01	1.79	1.08	0.77	0.60	0.50	0.42	0.37	0.32
14	681.53	6.77	2.09	1.21	0.86	0.66	0.54	0.46	0.40	0.35
15	1953.34	9.51	2.46	1.37	0.95	0.73	0.59	0.50	0.43	0.38
16	6023.32	14.00	2.92	1.54	1.05	0.79	0.64	0.54	0.46	0.41
17	19861.49	21.72	3.51	1.73	1.15	0.87	0.70	0.58	0.50	0.44
18	69679.08	35.72	4.29	1.96	1.27	0.94	0.75	0.63	0.54	0.47
19	258986.13	62.42	5.33	2.22	1.40	1.02	0.81	0.67	0.57	0.50
20	1016214.55	115.92	6.80	2.54	1.54	1.11	0.87	0.72	0.61	0.53
21	4196574.07	228.38	8.91	2.92	1.70	1.21	0.94	0.77	0.65	0.57
22	0.1819E+08	475.89	12.08	3.39	1.88	1.31	1.01	0.82	0.69	0.60
23	0.8256E+08	1045.25	17.00	3.98	2.09	1.42	1.08	0.88	0.74	0.64

24	0.3915E+09	2411.81	24.94	4.73	2.33	1.54	1.16	0.94	0.78	0.68
25	0.1936E+10	5828.32	38.25	5.73	2.60	1.68	1.25	1.00	0.83	0.71
26	0.9970E+10	14711.35	61.37	7.07	2.93	1.83	1.34	1.06	0.88	0.75
27	0.5335E+11	38695.63	103.07	8.94	3.33	2.00	1.44	1.13	0.93	0.80
28	0.2963E+12	105851.70	180.97	11.60	3.81	2.19	1.55	1.20	0.99	0.84
29	0.1705E+13	300604.42	331.64	15.51	4.41	2.41	1.67	1.28	1.04	0.88
30	0.1016E+14	884862.68	633.06	21.42	5.17	2.66	1.80	1.36	1.11	0.93
31	0.6257E+14	2696063.38	1256.04	30.63	6.16	2.96	1.94	1.45	1.17	0.98
32	0.3980E+15	8491905.71	2585.15	45.42	7.45	3.30	2.10	1.55	1.24	1.03
33	0.2612E+16	0.2762E+08	5509.25	69.88	9.21	3.72	2.28	1.66	1.31	1.09
34	0.1767E+17	0.9265E+08	12137.29	111.50	11.63	4.22	2.48	1.77	1.39	1.14
35	0.1231E+18	0.3203E+09	27602.76	184.40	15.07	4.84	2.71	1.90	1.47	1.20
36	0.8818E+18	0.1140E+10	64719.97	315.66	20.05	5.62	2.98	2.03	1.55	1.26
37	0.6497E+19	0.4171E+10	156275.81	558.54	27.47	6.61	3.29	2.18	1.65	1.33
38	0.4917E+20	0.1569E+11	388217.34	1020.06	38.79	7.90	3.66	2.35	1.75	1.40
39	0.3820E+21	0.6062E+11	991265.38	1920.08	56.48	9.61	4.09	2.54	1.86	1.48
40	0.3045E+22	0.2403E+12	2599393.58	3720.17	84.84	11.92	4.62	2.76	1.98	1.56
41	0.2488E+23	0.9772E+12	6994944.03	7410.40	131.37	15.12	5.26	3.01	2.11	1.64
42	0.2083E+24	0.4072E+13	0.1930E+08	15159.94	209.60	19.62	6.07	3.29	2.26	1.73
43	0.1786E+25	0.1738E+14	0.5458E+08	31821.49	344.18	26.11	7.08	3.62	2.42	1.83
44	0.1567E+26	0.7593E+14	0.1581E+09	68476.95	581.08	35.67	8.39	4.00	2.60	1.94
45	0.1406E+27	0.3394E+15	0.4686E+09	150951.79	1007.55	50.04	10.09	4.46	2.80	2.06
46	0.1290E+28	0.1551E+16	0.1421E+10	340643.99	1792.29	72.10	12.37	5.02	3.03	2.18
47	0.1209E+29	0.7248E+16	0.4404E+10	786420.68	3267.65	106.69	15.44	5.70	3.29	2.32
48	0.1158E+30	0.3459E+17	0.1395E+11	1856284.81	6100.37	162.08	19.70	6.53	3.59	2.48
49	0.1132E+31	0.1688E+18	0.4514E+11	4477451.97	11652.55	252.58	25.68	7.58	3.94	2.65
50	0.1130E+32	0.8385E+18	0.1491E+12	0.1103E+08	22756.95	403.45	34.25	8.92	4.35	2.84

Mean time $\overline{T}(n)$ until overflow from an initially empty bucket, as a function of arrival rate λ and bucket capacity n.
Assumptions: Records arrive one-at-a-time according to a Poisson process; residence times are exponentially distributed, with mean value unity.

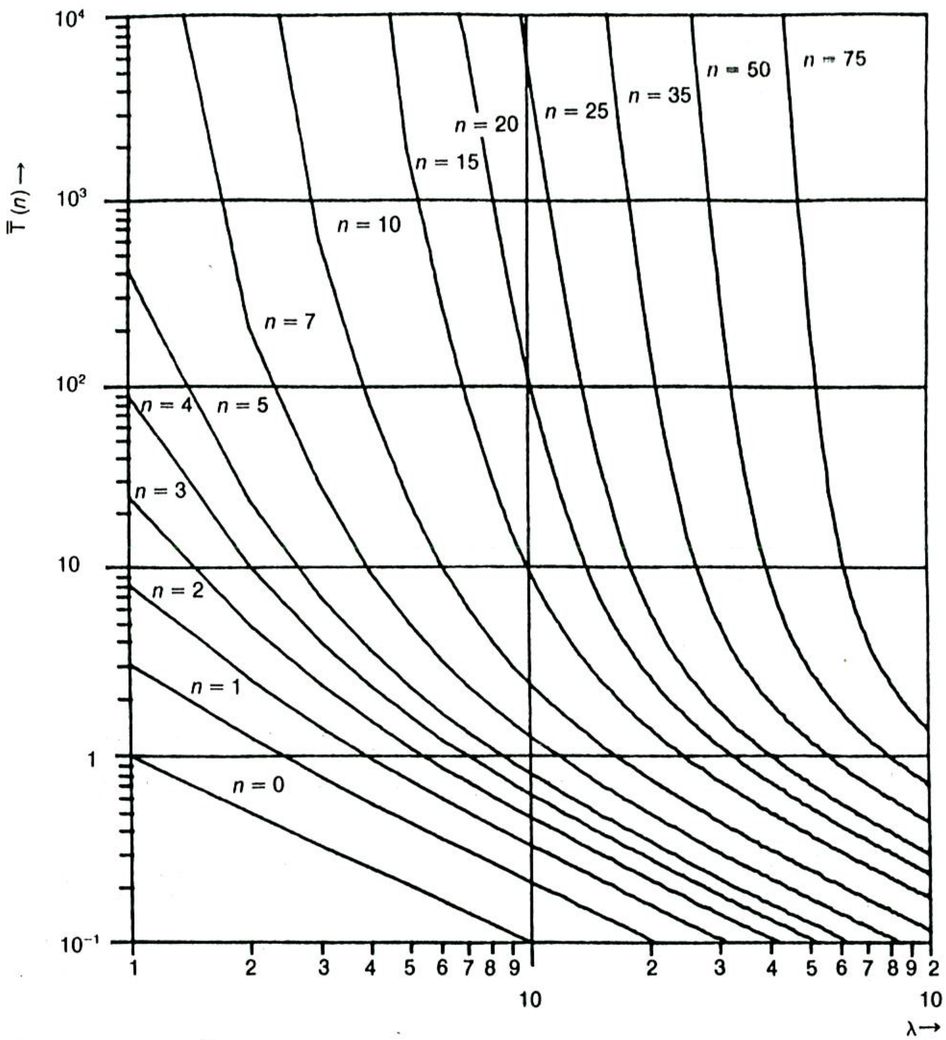

Fig. 1. Mean time $\bar{T}(n)$ until overflow from an initially empty bucket of capacity n, as a function of arrival rate λ.

Assumptions: Records arrive one-at-a-time according to a Poisson process; residence times are exponentially distributed, with mean value unity.

Table II.

n \ p^{-1}	1	2	4	5	10	20	100
0	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
1	0.2100	0.2091	0.2077	0.2071	0.2053	0.2034	0.2009
2	0.3320	0.3288	0.3240	0.3222	0.3162	0.3105	0.3028
3	0.4686	0.4610	0.4502	0.4461	0.4331	0.4213	0.4055
4	0.6232	0.6079	0.5873	0.5799	0.5565	0.5360	0.5093
5	0.8006	0.7725	0.7371	0.7247	0.6869	0.6547	0.6139
6	1.0070	0.9585	0.9011	0.8819	0.8248	0.7777	0.7196
7	1.2514	1.1705	1.0816	1.0530	0.9707	0.9051	0.8262
8	1.5470	1.4145	1.2808	1.2398	1.1252	1.0371	0.9338
9	1.9130	1.6983	1.5017	1.4442	1.2891	1.1738	1.0424
10	2.3790	2.0320	1.7476	1.6687	1.4629	1.3156	1.1519
11	2.9917	2.4293	2.0224	1.9158	1.6475	1.4626	1.2625
12	3.8268	2.9082	2.3310	2.1887	1.8437	1.6151	1.3741
13	5.0125	3.4930	2.6788	2.4909	2.0524	1.7732	1.4868
14	6.7725	4.2169	3.0727	2.8266	2.2746	1.9372	1.6005
15	9.5124	5.1256	3.5205	3.2005	2.5114	2.1074	1.7152
16	13.9964	6.2825	4.0319	3.6183	2.7639	2.2840	1.8310
17	21.7190	7.7763	4.6183	4.0864	3.0333	2.4674	1.9478
18	35.7199	9.7327	5.2937	4.6123	3.3211	2.6577	2.0658
19	62.4214	12.3307	6.0748	5.2049	3.6288	2.8554	2.1848
20	115.9245	15.8281	6.9817	5.8745	3.9578	3.0607	2.3049
21	228.3811	20.5987	8.0389	6.6331	4.3102	3.2739	2.4262
22	475.8855	27.1895	9.2763	7.4947	4.6876	3.4955	2.5485
23	1045.2457	36.4070	10.7299	8.4760	5.0923	3.7258	2.6720
24	2411.8101	49.4493	12.4439	9.5963	5.5264	3.9651	2.7967
25	5828.3210	68.1099	14.4722	10.8785	5.9925	4.2139	2.9224

Mean time $\bar{T}(n)$ until overflow from an initially empty bucket when arrivals occur in batches, as a function of mean batch size p^{-1} and bucket capacity n.
Assumptions: Batch arrival epochs follow a Poisson process; overall arrival rate of records is 10 records per unit time; batch sizes are geometrically distributed; residence times are exponentially distributed, with mean value unity.

Now, since the time until a bucket overflows is given by the sum (3) of independent random variables, it follows that for sufficiently large $n - i$,

$$P\{T(n, i) \le t\} \approx \Phi\left(\frac{t - E(T(n, i))}{\sqrt{V(T(n, i))}}\right), \tag{24}$$

where $\Phi(\cdot)$ is the standard normal distribution function. For example, consider a bucket of capacity $n = 20$ and initial fill $i = 0$, and suppose that records arrive one-at-a-time ($p = 1$) according to a Poisson process ($\nu = 1$) with rate $\lambda = 10$ records per unit time. Then, from Table I, $E(T(20, 0)) \approx 115.9$ and from Eqs. (11) to (13), $V(T(20, 0)) \approx 12{,}818$. (The large value of the variance is typical of the characteristically high variability of the time to overflow.) Hence, from (24), the probability that overflow will occur sooner than half the expected time is

$$P\{T(20, 0) \le \tfrac{1}{2}E(T(20, 0))\} \approx \Phi(-.51) = 30\%.$$

Table III.

n \ ν	1	2	5	10	20	50	∞
0	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
1	0.2100	0.2102	0.2104	0.2105	0.2105	0.2105	0.2105
2	0.3320	0.3329	0.3335	0.3337	0.3338	0.3338	0.3339
3	0.4686	0.4708	0.4722	0.4727	0.4730	0.4731	0.4732
4	0.6232	0.6278	0.6308	0.6319	0.6324	0.6327	0.6329
5	0.8006	0.8092	0.8150	0.8171	0.8181	0.8188	0.8192
6	1.0070	1.0226	1.0334	1.0373	1.0393	1.0405	1.0413
7	1.2514	1.2791	1.2987	1.3059	1.3097	1.3120	1.3135
8	1.5470	1.5954	1.6312	1.6447	1.6518	1.6561	1.6591
9	1.9130	1.9980	2.0641	2.0897	2.1034	2.1119	2.1178
10	2.3790	2.5301	2.6550	2.7057	2.7332	2.7505	2.7625
11	2.9917	3.2656	3.5104	3.6152	3.6736	3.7110	3.7369
12	3.8268	4.3374	4.8394	5.0693	5.2019	5.2882	5.3489
13	5.0125	5.9968	7.0833	7.6242	7.9492	8.1662	8.3214
14	6.7725	8.7453	11.2463	12.6234	13.4940	14.0937	14.5316
15	9.5124	13.6430	19.8051	23.6263	26.1947	28.0313	29.4073
16	13.9964	23.0646	39.3988	51.0143	59.4048	65.6781	70.5224
17	21.7190	42.6581	89.4137	128.2237	158.6820	182.6618	201.8437
18	35.7199	86.6895	231.6003	374.3805	497.4499	600.2177	685.8107
19	62.4214	193.4351	680.5982	1259.3467	1813.1113	2307.1349	2737.7738
20	115.9245	471.9422	2250.3969	4834.8284	7608.7270	10271.7860	12714.1277
21	228.3811	1251.8782	8306.6470	21013.9582	36470.0622	52554.6233	68157.2181
22	475.8855	3589.8844	34006.4436	0.1027E+06	0.1984E+06	0.3072E+06	0.4193E+06
23	1045.2457	11072.7925	0.1536E+06	0.5623E+06	0.1220E+07	0.2042E+07	0.2948E+07
24	2411.8101	36580.8114	0.7623E+06	0.3431E+07	0.8440E+07	0.1537E+08	0.2360E+08
25	5828.3210	0.1290E+06	0.4141E+07	0.2326E+08	0.6551E+08	0.1308E+09	0.2144E+09

Mean time $\bar{T}(n)$ until overflow from an initially empty bucket, as a function of the order ν of the Erlangian distribution of interarrival times and bucket capacity n.

Assumptions: Arrivals occur one-at-a-time; overall arrival rate of records is 10 records per unit time; residence times are exponentially distributed, with mean value unity.

We remark that Eq. (24) is only an approximation with unknown accuracy; however, its application suggests that the actual time to overflow in a particular instance is not unlikely to differ appreciably from its expected value.

4. SUMMARY

In Section 2 we derived, assuming exponential residence times and a variety of input processes, the average time until a bucket overflows by applying Eqs. (16), (8), (9), (10) in that order. These results yield the average time until an overflow chain contains any specified number of buckets for any external chaining scheme in which a new bucket is added to a chain if and only if a record arrives which belongs on the chain, but there is no room in the already existing buckets on the chain.

In Section 3, using the central limit theorem, we have shown how to approximate the distribution of time until first overflow. In order to continue our method to obtain the *exact* distribution of overflow time, we would have to invert a Laplace–Stieltjes transform—often a very difficult mathematical problem, but possibly solvable numerically or with the aid of an algebraic manipulation system.

Given an arrival rate λ and an average residence time τ, we defined the stable point as the expected number of records in an infinite-capacity bucket (to which records arrive one-at-a-time according to a Poisson process) with that arrival rate and average residence time, namely $\lambda\tau$ records. From the data in Section 3, it appears that the mean time required for an empty bucket to reach its stable point (as long as its capacity exceeds the stable point) is a relatively small number that grows very slowly with λ.

The data in Section 3 also suggest that for any bucket with capacity which exceeds its stable point, a slight decrease in the load factor ratio i/n obtained by increasing the bucket capacity n can cause a drastic increase in mean time until overflow, while the same decrease in the load factor ratio obtained by keeping the bucket capacity fixed, but initially loading less records, can have a much less pronounced effect on mean time until overflow.

In summary, then, our analysis provides information, both quantitative and qualitative, that can, in conjunction with consideration of other design criteria and constraints, help the database administrator in making rational choices for values of reorganization time, bucket capacity, and initial fill.

ACKNOWLEDGMENT

We are indebted to Abby Solomon for her typing and editorial assistance, and to Robert C. Gray and Samuel C. Hughes for assistance in preparing the graphs.

REFERENCES

1. BATORY, D.S. Optimal file designs and reorganization points. *ACM Trans. Database Syst. 7*, 1 (Mar. 1982), 60–81.
2. COOPER, R.B. *Introduction to Queueing Theory.* 2nd ed., Elsevier, New York, 1981.
3. HEYMAN, D.P. Mathematical models of database degradation. *ACM Trans. Database Syst. 7*, 4 (Dec. 1982), 615–631.
4. KEEHN, D.G., AND LACY, J.O. VSAM data set design parameters. *IBM Syst. J. 13*, 3 (1974), 186–212.

5. KNUTH, D.E. *The Art of Computer Programming, Vol. III: Sorting and Searching.* Addison-Wesley, Reading, Mass., 1973.
6. LARSON, P.-Å. Analysis of index-sequential files with overflow chaining. *ACM Trans. Database Syst. 6*, 4 (Dec. 1981), 671–680.
7. PALM, C. Intensitätsschwankungen im Fernsprechverkehr. *Ericsson Technics 44* (1943), 1–189.
8. QUITTNER, P., CSÓKA, S., HALÁSZ, S., KOTSIS, D., AND VÁRNAI, K. Comparison of synonym handling and bucket organization methods. *Commun. ACM 24*, 9 (Sept. 1981), 579–583.
9. STONEBRAKER, M.R., WONG, E., AND KREPS, P. The design and implementation of INGRES. *ACM Trans. Database Syst. 1*, 3 (Sept. 1976), 189–222.
10. WIEDERHOLD, G. *Database Design.* McGraw-Hill, New York, 1977.