REPORT R-501, JANUARY, 1971

UILU-ENG 71-2204

CSL COORDINATED SCIENCE LABORATORY

ON THE RESPONSE TIME OF ON-LINE RETRIEVAL SYSTEMS

J. T. CORDARO, JR.
R. T. CHIEN

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

UILU-ENG 71-2204

ON THE RESPONSE TIME OF ON-LINE RETRIEVAL SYSTEMS

by

J. T. Cordaro, Jr. and R. T. Chien

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under Contract DAAB-07-67-C-0199, in part by the International Business Machines Corporation, in part by the Office of Education under Contract C-1-7-071213-4557, and in part by the National Science Foundation under Grant GK-2339.

ON THE RESPONSE TIME OF ON-LINE RETRIEVAL SYSTEMS*

Summary

In this report the results of an investigation of on line retrieval systems are presented. The report is divided into two chapters. The first chapter is an analysis of some queuing models. The result is a formula relating search time, response time, and the number of terminals when requests are processed in batches.

In the next chapter response times of linear and inverted files are evaluated. One at a time and batch processing are compared.

The principal support for this work was a postdoctoral fellowship awarded to Dr. J. T. Cordaro, Jr.

*This work was sponsored by the International Business Machines Corporation; auxiliary support was provided by the Joint Services Electronics Program under Contract DAAB-07-67-C-0199.

A QUEUING MODEL FOR INFORMATION RETRIEVAL

J. T. Cordaro, Jr.

Introduction

On line information retrieval systems are in wide use. A typical system consists of a file and processor connected by appropriate equipment to several remote terminals. Users at the terminals request and receive file data. It is of interest to know the response time, that is, the time a user has to wait for a reply. A knowledge of the relation between the response time, the number of terminals, and some file parameters should be useful during the modification of an existing retrieval system or the design of a new one. For example one might need to estimate the effect on response time of adding new terminals to an operating system. A fundamental distinction between the analysis of single and multiple terminal systems is the problem of queuing and the resulting increase in response time. With a multiple terminal system the response time depends not only on hardware, programming, and file organization as in the single terminal case, but also on scheduling among the terminals and the number of terminals.

In this report a simple retrieval system model is examined and the average response time is estimated. With the usual time-shared installation, a system is used for many types of jobs in addition to retrieval. In this situation, with different inputs large variations in response time can be expected and knowledge of the average response time is of little value. On the other hand, suppose a system is used exclusively for retrieval of data from one file. Here the same type of job is repeated many times. In this case the response time should not show large statistical variations and the average response time should be a good indication of system performance.

A possibility for response time reduction arises in retrieval only systems. Each request requires for its service a file access. Usually in order to read a portion of the file some mechanical motion is required. For example with a disc file, the read heads have to move to the correct track and the disc has to rotate into position. If the number of terminals is large, then it may be anticipated that several requests arriving in the same time interval require access to nearby file records. When this happens file access time is saved by batching the requests and organizing the file search to minimize the access time for the whole batch. The amount of savings that can be obtained depends of course on the type of mechanical file. In this report the file is modeled as a random delay. Detailed file characteristics do not enter the analysis.

In the first section below the basic retrieval process is discussed. Then a model for one at a time processing is examined and this model is modified to include batch processing. Finally, as an application of the results, the response time is found for a system with a disc file.

Retrieval Process

When a retrieval system is in operation requests travel from the terminals into the processor and data travels from the file to the terminals. The details of these operations are very intricate and of course difficult to model exactly. However, by omitting a number of details a model of the gross system behavior can be obtained.

The two basic phases of a retrieval cycle for a system with only one terminal are the search and the evaluation. In the search, a request received from a terminal is converted to file addresses, the file is read, and a number of data items are sent to the user. Then the next phase, the evaluation begins. During this period the user looks at the data, decides if they are what he wanted, and either formulates a new request or gives the terminal to another user. The completion time for either of these phases is a function of many factors and varies from one request and user to the next. For modeling purposes the two completion times may be thought of as random variables. Associated with the phases are two distribution functions F(x) and B(x) where

$$F(x) = \text{Prob}\,[\text{search time} \leq x]$$

and

$$B(x) = \text{Prob}\,[\text{evaluation time} \leq x]$$

Approximations to these functions are available either from data on real systems or from mathematical considerations of particular files and I/O equipment.

Notice that if the system has only one terminal, then the search time and the response time are identical and F(x) contains all the response time information. However, if the system has more than one terminal a request may arrive at the machine during a time interval when the processor and file are working on another request. When this happens the arriving request has to queue for processor time. The response time is then the sum of the queuing time and search time. The queuing time depends on the number of requests waiting ahead of the arriving request and the manner in which requests are chosen for processing. Requests from one group of terminals may have priority over other requests, and request may be processed singly or in batches. Thus in order to find the response time of a multiple terminal system it is necessary to make some explicit assumptions about queuing. In the next section a model for a system with one at a time processing is examined. Batch processing is taken up in a later section. In all that follows the goal is to find relations for the average response time in terms of the number of terminals and the distribution functions $F(x)$ and $B(x)$ of search time and evaluation time.

One at a Time Processing

In the first model requests are processed one at a time. Suppose the system has m terminals. When a request arrives at the machine, processing begins immediately unless another request is being processed. In this case the new request joins a queue. Requests are processed in order of their arrival. The processing times are independent random variables distributed according to the function $F(x)$. When the work on a request is completed, file data are returned to the corresponding terminal. The users at the terminals act independently and their evaluation times are distributed according to the function $B(x)$.

This model is well known in the queuing theory literature where it is applied to the servicing of machines. In special cases corresponding to different assumptions on $F(x)$ and $B(x)$ several authors have obtained expressions for the response time [1]. The most general results of which the author is aware are due to Takács [2]. He assumes that the evaluation time has the exponential distribution, that is, that $B(x) = 1-e^{-\mu x}$ where $\mu > 0$. He puts no conditions on $F(x)$ except of course that the search time is non-negative. Under these assumptions the average response time ER is given by

$$ER = m\alpha - \frac{1}{\mu}(1 - P_{m-1}). \qquad (1)$$

where

$$\alpha = \int_o^\infty x dF(x) \qquad (2)$$

$$\frac{1}{\mu} = \int_o^\infty x dB(x)$$

are the average search and evaluation times, and P_{m-1} is the probability that at the end of a search phase the queue is empty. The quantity P_{m-1} is given by

$$P_{m-1} = \left[1 + \sum_{j=1}^{m-1} \binom{m-1}{j} \frac{1}{C_j}\right]^{-1} \qquad (3)$$

where

$$C_j = \frac{f(\mu)}{1-f(\mu)} \frac{f(2\mu)}{1-f(2\mu)} \cdots \frac{f_{(j\mu)}}{1-f(_{j\mu})} \qquad (4)$$

and

$$f(x) = \int_o^\infty \bar{e}^{sx} dF(x) \qquad (5)$$

is the Laplace transform of F(x).

The expression (1) for ER is rather simple except for the factor P_{m-1}. In order to evaluate P_{m-1} a detailed knowledge of F(x) is required. However, it is possible to upper-bound P_{m-1} in a simple way. In retrieval applications it is reasonable to assume that for some $z > 0$, $F(z) = 0$. This just says that the search time cannot be less than some time z. For this case

$$f(r\mu) = \int_o^\infty e^{-r\mu x} dF(x) = \int_z^\infty e^{-r\mu x} \qquad (6)$$

$$\leq e^{-r\mu z}$$

It follows that

$$C_j \leq \prod_{i=1}^{j} \frac{e^{-i\mu z}}{1-e^{-i\mu z}} = \prod_{i=1}^{j} \frac{1}{e^{i\mu z}-1} \tag{7}$$

and that

$$P_{m-1} \leq \left[1 + \sum_{j=1}^{m-1} \binom{m-1}{j} \prod_{i=1}^{j} (e^{i\mu z}-1) \right]^{-1} = k_m(z\mu) \tag{8}$$

From (8) it can be seen that for fixed μ and z, P_{m-1} goes to zero rapidly as m increases. The bound on inequality (8) depends only on m and the product $z\mu$. Some values of $k_m(z\mu)$ have been computed and are listed below.

$z\mu$ \ m	8	10	16	20	40	50	80
.173	$9.47 \cdot 10^{-2}$	$1.67 \cdot 10^{-2}$					
.079			$5.37 \cdot 10^{-2}$	$3.21 \cdot 10^{-3}$			
.046					$2.41 \cdot 10^{-7}$	$1.67 \cdot 10^{-14}$	$1.7 \cdot 10^{-37}$
.025							$1.7 \cdot 10^{-14}$

Table 1. Values of $k_m(z\mu)$.

For systems with $z\mu$ and m large enough that $k_m(z\mu) \approx 0$, the average response time can be written simply as

$$ER \approx m\alpha - \frac{1}{\mu} . \tag{9}$$

Thus in this case the average response time depends on only the average search and evaluation times. Detailed knowledge of the search time distribution is not required. Higher moments of the response time distribution can be calculated from the results in [2]. For example, it can be shown that if the search time is constant and $k_m(z\mu) \approx 0$, then the variance of the response time is

$$\text{Var } R \approx \frac{1}{\mu^2} - \frac{\alpha}{\mu}\frac{1}{e^{\alpha\mu}-1} \quad . \qquad (10)$$

This is a rather surprising result as it is independent of m. The variance for more general cases can be computed of course but the results are not as simple.

A Modification

In this section the model considered in the last section is modified. The resulting model is used later in the analysis of batch processing.

The processor in the model of the last section has alternating busy and idle periods. An idle period begins whenever a search ends and the queue is empty. The probability distribution of the length of an idle period is $1-e^{-m\mu x}$. The expected length is $\frac{1}{m\mu}$ which, as one would hope from physical reasoning, decreases with increasing m. Also, using some Markov chain results, it can be argued that on the average, idle periods begin at the end of $\frac{1}{P_{m-1}}$ search periods. As table 1 shows this is not very often. However because of the assumed form of the evaluation time distribution, $B(x) = 1-e^{-\mu x}$, very long idle times have non-zero probability. Thus even though idle periods occur rarely and are short when m is large they cannot be neglected.

For the work that follows it is necessary to remove the idle periods. Notice that the lengths of idle and busy periods are independent random variables. The idle period distribution can be changed without changing the distribution of the busy period. Consider the following modification of the model. Suppose the evaluation time is distributed as $1-e^{-\mu x}$ as long as the processor is busy. But suppose that whenever the processor finds an empty queue, one of the terminals immediately puts in a request. The busy period distribution remains the same but the idle period distribution in the new model is concentrated at zero. That is, the idle periods are eliminated. For values of m and $z\mu$ such that $P_{m-1} \approx 0$ and $\frac{1}{m\mu}$ is small there should be little difference in the operation of the two models. A consideration of reference [2] shows that the arguments there carry over to the modified model and that the average response time is given as before by equation (9). Some insight into the workings of the proof can be gained by considering Q_k, the probability that at an arbitrary time k requests are in queue. The average response time is a continuous function of Q_k, $k = 1,\ldots,m-1$. Let T_k be the average time during a long run that k requests are waiting. Then since the average length of a busy period is α/P_{m-1}, it follows that

$$Q_k \approx \frac{T_k}{\frac{1}{m\mu} + \frac{\alpha}{P_{m-1}}} \tag{11}$$

for the unmodified model. Modifying the idle period changes (11) to

$$Q_k \approx \frac{T_k}{\frac{\alpha}{P_{m-1}}} \tag{12}$$

Eqs. (11) and (12) indicate that when $P_{m-1} >> \frac{1}{m\mu\alpha}$ the two models have approximately the same average response time.

One way to have a physical system approximate the modified model would be to let the processor notify the terminals when it is free. With the incentive of zero queuing time some user could be expected to submit a request quickly.

Batch Processing

In this section an expression is found for the average response time of a system that processes requests in batches instead of one at a time. As mentioned earlier, batch processing should improve the response time when several requests are in queue at the end of a search phase. From Eq.(1) it follows that the average queuing time is

$$(m-1)\alpha - \frac{1}{\mu}(1-P_{m-1}). \qquad (13)$$

Thus for large m most of the response time is due to queuing.

A straightforward way to analyze the batch processing scheme would be to use the unmodified model above and require that requests be processed in batches of k at a time. Unfortunately the resulting equations are very difficult to handle. Another way would be to use one of the batch arrival batch service models from queuing theory. This would have the disadvantage that the evaluation time distribution would not be exponential and the results could not be directly compared with the one at a time processing model.

An approximation to the average response time can be obtained using the modified model. Suppose a retrieval system has m terminals and that requests are processed in batches of k in an average time per batch of α_k. If at the end of a search phase the queue has at least k requests, then each of the other terminals is in an evaluation phase with length distributed as $1-e^{-\mu x}$. But if a search phase ends and only $\ell < k$ requests are in queue, the processor immediately notifies $k-\ell$ terminals which then submit requests. In either case the next search phase begins immediately after the preceding one and there are no idle periods. This is done at the expense of a slight change in the evaluation time distribution. As can be seen this model is just the batch processing analogue to the modified one at a time model above.

One more step will be taken in the evolution of a final model for batch processing. Suppose that instead of one queue there are k queues in the machine. One queue corresponds to a group of $\frac{m}{k}$ terminals. If m/k is not an integer then let $(\frac{m}{k})^*$ be the smallest integer greater than m/k. Then the terminals can be divided so that some of the k queues have $(\frac{m}{k})^*$ corresponding terminals and some $(\frac{m}{k})^*-1$ terminals. At the beginning of a search phase one request is taken from the head of each of the k queues. If at the end of the search each queue has at least one request then the next search phase begins and the evaluation time for each terminal is exponentially distributed. If one or more queues is empty then the processor notifies the corresponding terminal group, a request arrives immediately, and the search begins. Batches of requests are processed in an average time α_k. In this model each group of $\frac{m}{k}$ terminals behaves exactly as the modified one at a time model. Thus the average response time for the system ER_k is given by

$$ER_k = \frac{m}{k}\alpha_k - \frac{1}{\mu}\left(1-P_{\frac{m}{k}-1}\right) \qquad (14)$$

If m/k is not an integer then it can be replaced by $(m/k)^*$ in (14) yielding an upper bound to ER_k.

This final model appears to be quite different in operation from the first model discussed at the beginning of the section. Two things can be said in justifying its use. First the final model is easy to analyze since it is actually just a combination of k simpler systems. Second, when $P_{\frac{m}{k}-1} \approx 0$ the k queues will hardly ever be empty at the end of a service period; it will rarely be necessary to disturb the exponential evaluation distribution; and as a result the operation of the two models should be very similar.

Under the condition that $P_{\frac{m}{k}-1} \approx 0$ the average response time is

$$ER_k \approx \frac{m}{k} \alpha_k - \frac{1}{\mu} . \qquad (15)$$

This may be compared with Eq.(1). It is possible now to evaluate the effect of batching. Whenever $\alpha_k < k\alpha$ batch processing will improve the average response time. If $\alpha_k \approx k\alpha$ then queuing time is traded for an equal amount of processing time. No decrease in response time results. From these observations it may be concluded that retrieval systems with many terminals should be studied on the basis of curves of α_k vs. k. If α_k increases at a rate less than linearly with k then batching will decrease the response time. In the design of systems, consideration should be given to methods of file search and organization that produce the best behavior in α_k.

An Example of Batching

In this section an example that illustrates the potential of batch processing is discussed. One would expect a multiple terminal system to have a very large file. If the file were on a disc unit then it would occupy perhaps several independent disc modules. The dynamic analysis of this type of file is a difficult problem in itself. Here a simpler problem is considered. Suppose the file is on one disc module. The time for the search phase of a single request is the sum of: (1) an initial CPU use time during which the record locations corresponding to the request are determined; and (2) a file access time during which the actual records are obtained from the disc. The average CPU use time will be denoted by c. No attempt will be made here to improve the CPU time by batching so the average CPU time for a batch of k requests is just kc.

Suppose that a request corresponds to n record locations. To access each record the read heads must be positioned at the proper track and sector. The total time to look up n records is

$$\sum_{i=1}^{n} [T_i + S_i + R_i] \tag{16}$$

where T_i and S_i are the track and sector seek times and R_i the read time for the i^{th} record. The average file access time for n records A_n is then

$$A_n = n[t + s + r] \tag{17}$$

where t, s and r are the average values of T, S and R. The quantities t, s and r depend on the physical characteristics of the file. If the number of records per track is d and the disc rotation time is b then as a reasonable approximation

$$s + r = \frac{b}{2} + \frac{b}{d} = b\left(\frac{d+2}{2d}\right). \qquad (18)$$

Note that s and r do not depend on n the number of records accessed. The track seek time t does depend on n. Because of this dependence on n, t will be written as t(n). As n increases the number of record locations increases and the distance between records to be accessed decreases. To compute t(n) assume that the locations of records to be accessed are distributed randomly in the file. Then the probability that the number of tracks between any two locations or between a location and the first track is at least a, $P_n(a)$ is given by

$$P_n(a) = \frac{(M-a)u}{Mu} \times \ldots \times \frac{(M-a)u-n+1}{Mu-n+1}, \quad n \geq 2, \quad 1 \leq a \leq M-1 \qquad (19)$$

where M is the number of cylinders and u is the number of records per cylinder. Eq.(19) can be approximated by

$$P_n(a) \approx \left(1 - \frac{a}{M}\right)^n \exp - \frac{n(n-1)}{2(M-a)u}, \quad n \geq 2, \quad 1 \leq a \leq M-1. \qquad (20)$$

The total track seek time t(n) depends of course on the order in which the records are accessed. Here it will be assumed that before each search phase the read heads are positioned over either the first or last disc track. The heads can then sweep all the way across the disc stopping whenever necessary. With this access procedure, t(n) is just the average time for the read heads to move between two adjacent locations. Given Eq.(19) and the relation between tracks transversed and time for a particular file, the quantity t(n) can be computed.

As an example consider the IBM 2302 disc unit. For the 2302, a head movement within blocks of 10 tracks requires 50 ms, and roughly a movement of 11-50 tracks requires 120 ms and a movement of 51 or more tracks requires 180 ms. The average track seek time for n records t(n) is given by

$$t(n) = 50[P_n(1)-P_n(11)]+120[P_n(51)]+180\ P_n(51). \tag{20}$$

Some values of t(n) are listed in Table 2 below for a file with 247 cylinders and 1035 records per cylinder.

n	1	5	10	15	20	25	30	35	40
t(n)	160	123	98	84	75	68	62	57	54

Table 2. Track seek time, ms. vs. number of record locations.

The table shows as expected that when the number of locations increases, the average track seek time decreases. The limiting value of t(n) in this example is 50 ms.

If the possibility of more than one location in the same cylinder is neglected then simpler results can be obtained. In this case

$$P_n(a) = (1 - \frac{a}{M})^n \tag{21}$$

and the average distance between locations is just $\frac{1}{n+1}$. This is a useful result when the seek time vs. number of tracks travelled is approximately linear. Then the average seek time is directly proportional to the average distance between locations so that

$$t(n) = \frac{c}{n+1} \qquad (22)$$

where the constant c is the product of the number of tracks and the number of ms per track traveled.

Using these results for t(n), the average search time for a batch of k requests α_k can be computed. Suppose each request requires that an average of ℓ locations be accessed. Then α_k is given by

$$\alpha_k = kc + k\ell[t(k\ell) + b(\frac{d+2}{2d})]. \qquad (23)$$

This equation is of the form

$$\alpha_k = k\ell[t(k\ell)] + k[\text{constant term}]. \qquad (24)$$

The point worth noting here is that the first term in (24) increases less than linearly while the second term increases linearly with k. As a result the batching described here can be used to reduce the track seek time but not the processing, rotational delay, and read times.

Conclusion

Using a number of approximations some simple relations have been found for the average response time of multiple terminal retrieval systems. From these relations it was shown that batch processing can improve the response time.

References

[1] Cox, D. R. and W. C. Smith, Queues, Methren 1961.

[2] Takács, L., "On a Stochastic Process Concerning Some Waiting Time Problems," Theory of Probability and Its Applications, Vol. 11, No. 1, 1957.

[3] Feller, W., An Introduction to Probability Theory and Its Applications, Vols. I and II, 1968, 1966.

DESIGN CONSIDERATIONS OF ON-LINE DOCUMENT RETRIEVAL SYSTEMS

J. T. Cordaro, Jr. and R. T. Chien*

I. Introduction

Currently a great deal of interest is being expressed in multiple terminal on-line document retrieval systems. A number of retrieval schemes exist, and it is desirable to have a set of objective criteria with which to compare various systems. For example systems might be compared on the basis of storage requirement, precision, recall, and response time. In this paper we consider only the response time. We find simple relations between the average response time and file parameters for systems using linear and inverted files. In addition we show that if either type of system has many terminals then the use of batch processing can decrease the average response time. Throughout the paper we do not hesitate to use several simplifying assumptions and approximations.

In the first section we discuss some queuing results and then in the last two sections apply these results to a response time analysis of inverted and linear files.

II. Queuing

The basic system structure considered here consists of a computer with m remote terminals. In a terminal-machine interaction, a user submits a request in the form of a Boolean function of index terms and after a waiting

---

*Dr. Chien's work was supported in part by the Office of Education under Grant C-1-7-071213-4557 and in part by the National Science Foundation under Grant GK-2339.

period the machine displays at time T_F the first of a list of documents satisfying the request and at time T_L the last. We will refer to T_F and T_L as the first and last response times. ET_F and ET_L are the expected values of T_F and T_L. Initially we will assume that requests are processed one at a time in order of arrival. When a request is made it is processed immediately unless the computer is busy. In this case the request joins a queue. The average search time α is the time from when a request leaves the queue and begins to be processed until the last requested document is displayed. The average search time obviously depends on detailed system characteristics and file organization. It, as well as ET_F, will be dealt with in Sections III and IV. The purpose of this section is to make some rather general statements about ET_L for a given α.

Suppose that the m terminals operate independently and that the average terminal use time, that is, the time from when a terminal receives the last document satisfying its request until it submits another, is β. The following heuristic argument relates ET_L, α and β. Let W be the average time a request spends in queue. Then the probability that at an arbitrary time a terminal is waiting for a response is $\frac{\alpha + W}{\alpha + \beta + W}$. The fraction of terminals in this condition is $\frac{\alpha + W}{\alpha + \beta + W} m$. Thus when a request arrives it has to wait a time

$$W = \frac{\alpha + W}{\alpha + \beta + W} m \alpha. \qquad (1)$$

Solving for W we find that when both m and $\frac{\alpha m}{\beta}$ are large enough

$$W \approx (m - 1) \alpha - \beta \qquad (2)$$

so that

$$ET_L \approx m \alpha - \beta \qquad (3)$$

A rigorous treatment of this problem has been given by Takács [1] who has shown that if the terminal use time has the exponential distribution with mean β, then

$$ET_L = m\,\alpha - \beta(1 - P_{m-1}) \qquad (4)$$

where the factor $1 - P_{m-1}$ is the probability that at the end of a search the queue is not empty. If the minimum possible value of the search time is z, where of course $z \leq \alpha$, then

$$P_{m-1} \leq \left[1 + \sum_{j=1}^{m-1} \binom{m-1}{j} \prod_{k=1}^{j} (e^{kz/\beta} - 1)\right]^{-1} \qquad (5)$$

As Eq.(5) shows, P_{m-1} converges to zero as the number of terminal m increases. For the values of m, z and β in our applications below, we can verify that $P_{m-1} \approx 0$. Thus we can use Eq.(3) as the relation between ET_L, α and β. When the system has many terminals, the final response time is much longer than the search time. This fact is due to queuing. Let Q be the number of requests waiting for service at an arbitrary time. Then from Eq.(3) we may conclude that

$$E\,Q \approx (m-1) - \beta/\alpha \qquad (6)$$

which shows that long queues are common whenever m is large compared with β/α.

The fact that many requests are waiting with one at a time service leads to a consideration of batch service. Let α_k be the average search time when requests are serviced in batches of k. It is reasonable to expect that with batching we can organize the search efficiently and find that the average time to service a batch of k requests is less than the time to service k requests one at a time. Supposing for now that this is true, that is, that $\alpha_k < k\alpha$, we must ask if batching provides any improvement in response time. To answer this question we consider a model like the one at a time model except that now requests are serviced k at a time in an average time α_k. If at the end of a search only p requests are in queue, the system waits until k-p additional requests arrive and then resumes operation. Arguing as before we find that when m and α_k are large enough that $P_{\frac{m}{k}-1} \approx 0$ we have

$$ET_L \approx \frac{m\,\alpha_k}{k} - \beta . \qquad (7)$$

This shows that batch service is better than one at a time service if α_k increases less than linearly with k. In the next two sections we examine the behavior of α_k as a function of k for two types of retrieval.

III. Inverted File

The inverted file search consists of four basic operations.

1. Determine the locations of the inverted file records corresponding to all terms appearing in a given Boolean request.
2. Access the records.
3. Compare the records and identify the document locations satisfying the search request.
4. Access and display the bibliographic data at all locations found in part 3.

As can be seen the search has two processing and file reading phases. We will assume that either a random addressing or a core-based indexing scheme is used to relate terms to inverted file records. When this is done the time required for the first processing phase, part 1 is negligible compared with that for the rest of the search. The second processing phase, part 3 must be examined in more detail. A request is a Boolean function of its terms, e.g., $(A \vee B \vee C) \wedge (D \vee E \vee F) \wedge (G \vee H \vee I)$. During processing "AND" and "OR" operations are performed on the inverted file records corresponding to the terms in a request. An inverted file record is an ordered list of document location numbers. By writing simple algorithms and counting loops we can estimate the time for an "OR" of two lists, $\{a_1,\ldots,a_x\}$ and $\{b_1,\ldots,b_y\}$, to be $(x+y)(6u)$ and that for an "AND" to be $(x+y)(4u)$ where u is the memory cycle time. Details of the algorithms can be found in [7]. Let f be the average number of documents per term. For the request in the example above we can estimate the total processing time γ to be

$$\gamma = 3[(6u)(2f + 3f)] + (4u)[6f + f] = 118\ fu \qquad (8)$$

where we assume that any two lists have very few common elements. In the following we will assume that requests have enough uniformity that Eq.(8) is a reasonable processing time estimate for any request.

The remaining search time consists of reading the inverted and document files. The actual files could be placed in a number of different hardware configurations. We will limit the discussion to the case where the two files are on separate disc units. This seems to be a reasonable configuration in view of the anticipated file size. Each disc unit has several disc modules and each module has an independent read mechanism. We will also suppose that the two disc units are on separate channels. During the search, a file reading period begins either at step 2 or at step 4. In either case the disc controller is presented with a matrix (ℓ_{ij}) of record locations where ℓ_{ij} is the location of the i^{th} record to be read from the j^{th} module. Let M be the number of tracks per module. We will assume that the same number n of records are read from each module and that the n locations for a particular module are evenly spaced $\frac{M}{n+1}$ tracks apart. Thus the distance between the first track and the first location or between any two adjacent locations is $\frac{M}{n+1}$. These assumptions simplify the analysis but still provide insight into the physical problem. Now suppose there are S disc modules and that nS records are to be read. The following procedure is used. Each read arm starts from one edge of its file and sweeps across to the opposite edge, stopping for record seek and read operations every $(\frac{M}{n+1})$ tracks. Let T(j) be the time for an arm to move j tracks. The S read arms move simultaneously to the first S locations in time $T(\frac{M}{n+1})$. Now this is followed by a waiting period while the correct records rotate into position for reading. For the inverted file we expect the records to be long and perhaps occupy a whole track. If we put several markers around each track then we can begin reading when the first marker appears. The record can be assembled into its correct order in the processor from a knowledge of the markers. If we have enough markers then the waiting time to the first one is negligible and the rotational delay the same as the read time which is the disc rotation time R_o. Thus after the access arms move to the first S locations, one arm, say A, reads in time R_o while the others wait for the channel. After reading, arm A moves to the next location and the next arm begins to read. The process continues in this way. If $T(\frac{M}{n+1}) \geq (S-1)R_o$, then when arm A finishes its second seek operation, the channel will be free. The same applies to the other arms. In this case the average time to read all the inverted file locations is

$$T_n = nT(\frac{M}{n+1}) + (S-1+n)R_o.$$

Now if $T(\frac{M}{n+1}) < (S-1)R_o$ then arm A finishes its second seek operation before the other arms have finished reading. In this case all but the first of the track seek times are overlapped by rotational delays and

$$T_n = T(\frac{M}{n+1}) + nSR_o \tag{10}$$

These two cases can be written together as

$$T_n = T(\frac{M}{n+1}) + nSR_o + (n-1)[T_n-(S-1)R_o]^+ \tag{11}$$

where

$$[x]^+ = 0 \quad \text{if} \quad x < 0$$
$$= x \quad \text{if} \quad x \geq 0$$

The first term is the initial track seek time. The second term is the total rotational delay. For an inverted file with the marking scheme outlined above nSR_o is the total read time. The third term is the remaining track seek time after accounting for the overlapping of track seeks with rotational delays.

The situation is essentially the same with the document file. Here we would expect to have several records per track. The average rotational delay is $(\frac{1}{2}+f)R_o = R'$ where f is the fraction of a track occupied by a record. The average time to read all the document file locations is given by Eq.(11) with R' in place of R_o. The rotational delay in document file reading could be decreased by using sector addressing as suggested by Wang and Ghosh [3]. However in the example below, at least, it is the reading time for the inverted file and not the document file that is the real limitation in search time reduction.

The dependence of Eq.(11) on the batch size k can be made explicit by setting kD = nS where for the inverted file the factor D is the average number of terms per request and for the document file D is the average number of documents satisfying a request. The average lookup time for a batch of k, L(k) is then

$$L(k) = T\left(\frac{M}{\frac{kD}{S}+1}\right) + kDR + \left(\frac{kD}{S} - 1\right)\left[T\left(\frac{M}{\frac{kD}{S}+1}\right) - \left(S - 1\right)R\right]^{+} \qquad (12)$$

where R is R_o or $(\frac{1}{2}+f)R_o$ depending on the file. The utility of batching is in the fact that only the second term of L(k) increases linearly with k. With batching the part of file lookup time due to track seeks can be made negligible compared with the rotational delay time.

When the inverted and document files are on separate channels parts of the total search can be overlapped. Parts 1-3 of the search can be done for one batch while part 4 is being done for the other batch. Let $L_I(k)$ and $L_D(k)$ be the inverted and document file look-up times. Then the time for parts 1-3 of the search is

$$L_I(k) + k\gamma \qquad (13)$$

where γ is the average processing time estimated above. This figure is conservative since some processing can begin before all the records are accessed. The average time for an overlapped search α_k is then

$$\alpha_k = L_D(k) + \left[L_I(k) + k\gamma - L_D(k)\right]^{+} \qquad (14)$$

since parts 1-3 of a search must be completed before part 4 can begin.

In a strict sense we cannot use α_k in the queuing Eq.(7). This is because there may be times when the queue is empty and there is no batch to overlap with the batch currently being serviced. However when $P_{\frac{m}{k}-1} \approx 0$ the fraction of time that the queue is empty is negligible and we will use Eq.(19) to describe the overlapped search. We have then that

$$ET_L \approx \frac{m}{k}\{L_D(k)+[L_I(k)+k\gamma-L_D(k)]^+\} -\beta \tag{15}$$

Rather than compute ET_F directly we can notice that it is lower bounded by the queuing time and upper bounded by ET_L so that

$$ET_L - \alpha_k \leq ET_F \leq ET_L. \tag{16}$$

These bounds are tight when m is large.

<u>Example</u> Consider a collection of $8 \cdot 10^5$ documents with 10^4 terms and an average of 15 terms per document. This gives an average of 1200 documents per term. We will suppose that the average number of documents satisfying a request is 16.

The inverted and document files are stored on two separate IBM 2314 disc units. These units have 8 modules each. A module has M = 200 cylinders with a rotational time R_o = 25 ms. Allowing 150 bytes for a document file entry and 3600 bytes for an inverted file entry the document file will fit on 8 modules with 25 records per track and the inverted file on 2 modules with one record per track.

Using these numbers and the published table for 2314 seek times we can evaluate Eq.(15). The results are shown in Fig. 1 for M = 1000, 500, and 100 terminals and β = 20 sec. It can be seen that batching is useful up to the point where the ET_L curves begin to level off. This is the point where the linear factors in α_k, i.e., read time and processing time, begin to dominate.

IV. Linear File

In a linear file search each request must be compared with the terms of each document. To evaluate the potential of linear files in an on-line environment, we will calculate first α_k, the time required for processing all documents against k queries. It can be verified that with modern tape units file data can be read from tapes into core faster than they can be processed. We may imagine that file data are continually flowing into core at the same average rate as processing. Consequently we will assume that the file portion to be examined is always in core when it is needed. This being the case it is seen that

$$\alpha_k = Ns_k \tag{17}$$

where N is the number of documents in the entire file, and s_k is the time required for processing k requests at once against the terms of one document. Let h be the average number of terms per request. We will use a two level search in which the first step is to compare all the kh terms of the k requests with the d terms of a document. If there are any matches an exact comparison is made at the second level. If we view the first step as an "AND" of the kh request terms and d document terms, then the processing time is (4u)(kh+d) as in Section III. This estimate can be improved. When one or both the lists are short compared with the list of all terms the chances are good that there will be little overlap between lists. Taking this into account we show in the appendix that a better first stage processing time estimate is

$$4u\ (kh\ \frac{d}{d+1} + d\ \frac{kh}{kh+1}) \tag{18}$$

To evaluate the time for the second stage processing we suppose that terms in the k requests are chosen independently, each with probability kh/N_T where N_T is the total number of terms in the collection. The average number of matches is then dkh/N_T. By writing an algorithm we estimate the time to process a single query against one document to be 4uhd. Thus the average time for second stage processing is $4uh^2kd/N_T$ and we have that

$$s_k = 4u(kh\frac{d}{d+1} + d\frac{kh}{kh+1}) + 4uh^2kd/N_T \tag{19}$$

But N_T is usually large enough that the second term can be neglected so Eqs.(17) and (19) yield

$$\alpha_k \approx 4uN(kh\ \frac{d}{d+1} + d\ \frac{kh}{kh+1}) \tag{20}$$

From Eqs.(7) and (20) we see that for batches of k the average last response time is given by

$$ET_L \approx 4mNu(h\ \frac{d}{d+1} + d\ \frac{h}{kh+1})\ -\beta \tag{21}$$

It is worth noting that batching is useful in reducing ET_L only up to the point where the second term in parentheses in Eq.(21) becomes negligible in comparison with the first. Beyond this point batching reduces queuing time but increases processing time resulting in no net effect on ET_L.

However the documents satisfying a search request can be read out as soon as they are identified. For a particular request the average time to the first document is $\frac{\alpha_k}{b}$ where b is the number of documents satisfying a request. Thus we have

$$ET_F = ET_L - \alpha_k + \frac{\alpha_k}{b}$$
$$= \alpha_k(\frac{m}{k} + \frac{1}{b} - 1) \tag{22}$$

As noted earlier we cannot use Eq.(7) for large values of k. But for large k and hence large α_k, we can argue that β should be small and Eq.(7) holds for $\beta = 0$. This fact is used to obtain Eq.(22). The best value of k in Eq.(22) is k = m which gives

$$ET_F = \frac{\alpha_M}{b} = \frac{4Nu}{b}[mh \frac{d}{d+1} + d \frac{mh}{mh+1}] \tag{23}$$

Increasing the batch size up to the maximum value of m makes an almost negligible decrease in ET_L but does improve ET_F. Our conclusion is that file data should flow continually into core at a rate which allows each document to be processed against m requests. This being the case there is no queuing for machine time. A request enters processing immediately and produces a first response $\alpha_{M/b}$ time units later.

Example Consider a system with $m = 100$, $d = 15$, $h = 10$, $N = 8\cdot10^5$, $u = 0.5\cdot10^{-6}$, $b = 16$. The graph of Fig. 1 shows ET_L and ET_F as a function of batch size. The graph of Fig. 2 shows ET_L as a function of batch size and the number of terminals.

V. Concluding Remarks

The response time for on-line document retrieval systems has been investigated. It is shown that for systems with a large number of terminals the response time is approximately linear with m, the number of terminals. Two file organizations have been evaluated. It is found that if traffic is not too heavy the inverted file seems adequate. The linear file is rather slow in comparison. The general conclusion here is that conventional techniques for document retrieval are not adequate for on-line systems when the number of terminals is very large. For such systems to be functional one needs to develop new and original file organization and search techniques.

We wish to note that when a parallel processor such as the ILLIAC IV is available the efficiency of the linear file system can improve by a factor at least equal to the number of PU's available. For sixty-four parallel PU's the system can handle about one hundred times the load. The additional improvement is obtained because of savings in execution time.

References

[1] Senko, M. E. File Organization and Management Information Systems. Cuadra, C.A. (Ed.) Annual Review of Information Science and Technology. Vol. 5, 1969. Brittannica, Chicago, pp. 111-143.

[2] Takács, L. Introduction to the Theory of Queues. Oxford University Press, Oxford, 1962, Chapter 5.

[3] Wang, C. P. and Ghosh, S. P. Analysis and Modeling of a Multimodule Disc Storage System with Sector Scheduling. Report, IBM San Jose Research Laboratory.

[4] Abade, J., Dubner, H., and Weinberg, S. B. Queuing Analysis of the IBM 2314 Disc Storage Facility. Journal ACM, 15, 4, (October, 1968), pp. 557-589.

[5] Feller, W. An Introduction to Probability and Its Applications. Vol II, John Wiley and Sons, New York, 1966, pp. 20-23.

[6] Barnes, G. H., Brown, R.M., Kato, M., Kuck, D.J., Slotnick, D.L., and Stokes, R.A. The ILLIAC IV Computer. IEEE Trans. Computers, C-17, (August, 1968), pp. 746-757.

[7] Cordaro, J.T., and Chien, R.T. Design Considerations of On-Line Document Retrieval Systems. Report R-456, Coordinated Science Laboratory, February, 1970, pp. 5-8.

Appendix

Theorem: Suppose we are given two sequences of points $a_1, a_2, --- a_x$ and $b_1, b_2, --- b_y$ obtained by ordering x and y points each uniformly distributed between 0 and M. Then if the sequences are matched in an "AND" operation the average number of steps is given by

$$x(\frac{y}{y+1}) + y(\frac{x}{x+1})$$

Proof of Theorem

From the probability that $a_x \leq t$ the density function for a_x is found to be (see for example Ref.[5], p.21)

$$(\frac{t}{M})^{x-1}(\frac{x}{M}).$$

Similarly the density for b_y is

$$(\frac{s}{M})^{t-1}(\frac{y}{M})$$

The number of processing steps when $t > s$ is $y + (x-1)\frac{s}{t}$. It is $x + (y-1)\frac{t}{s}$ when $s > t$. The expected number of processing steps is, therefore, given by

$$\theta = \int_o^M (\frac{t}{M})^{x-1}(\frac{x}{M})[\int_o^t (\frac{s}{M})^{y-1}(\frac{y}{M})(y + \frac{(x-1)s}{t})ds$$

$$+ \int_t^M (\frac{s}{M})^{y-1}(\frac{y}{M})(x + \frac{(y-1)t}{s})ds]dt$$

$$= y(\frac{x}{x+1}) + x(\frac{y}{y+1})$$

The ranges for integration are illustrated in Figure 3.

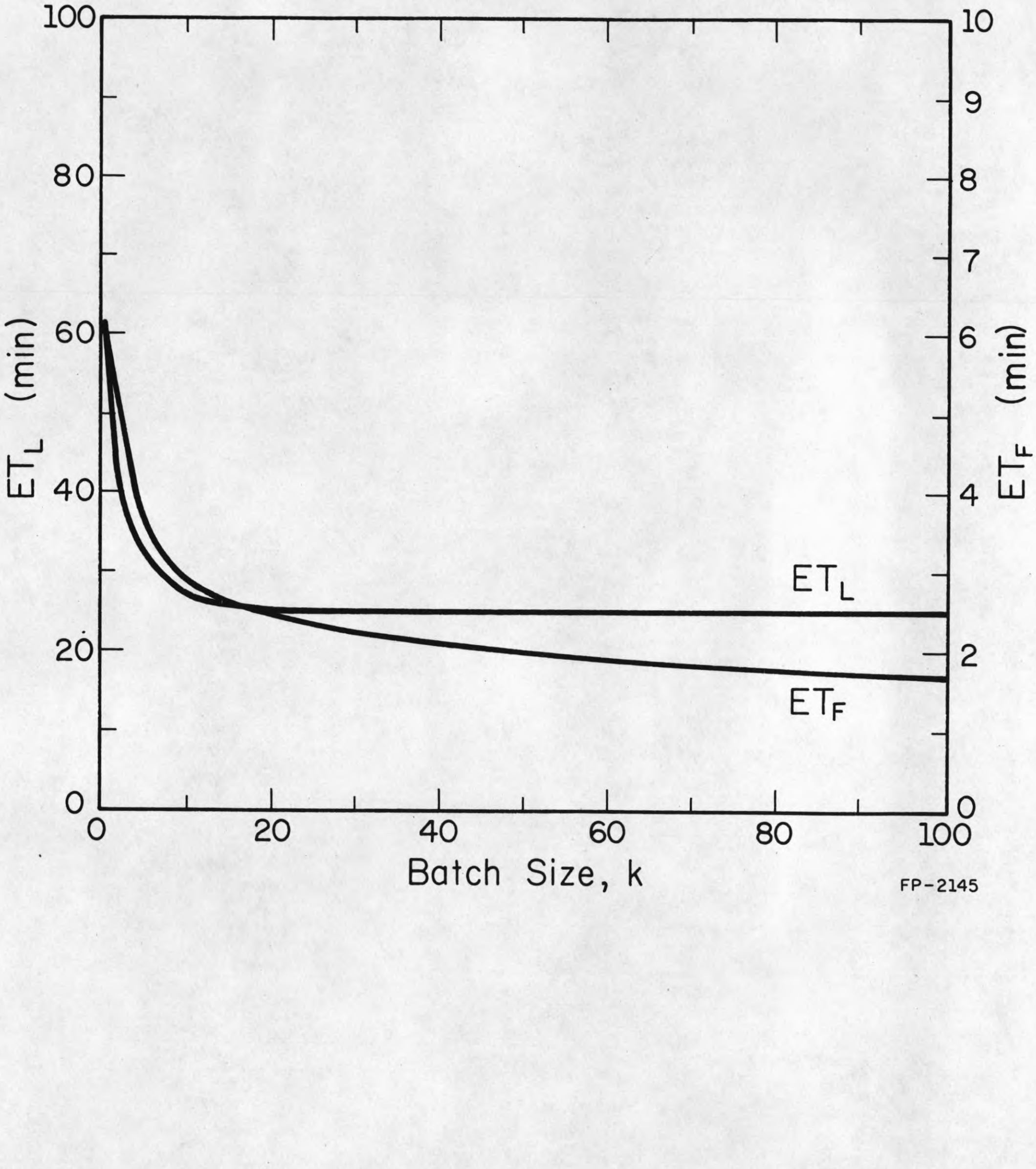
100
80
60
40
20
0
10
9
8
7
6
4
2
0
ET_L (min)
ET_F (min)
ET_L
ET_F
0
20
40
60
80
100
Batch Size, k
FP-2145

Figure 1.

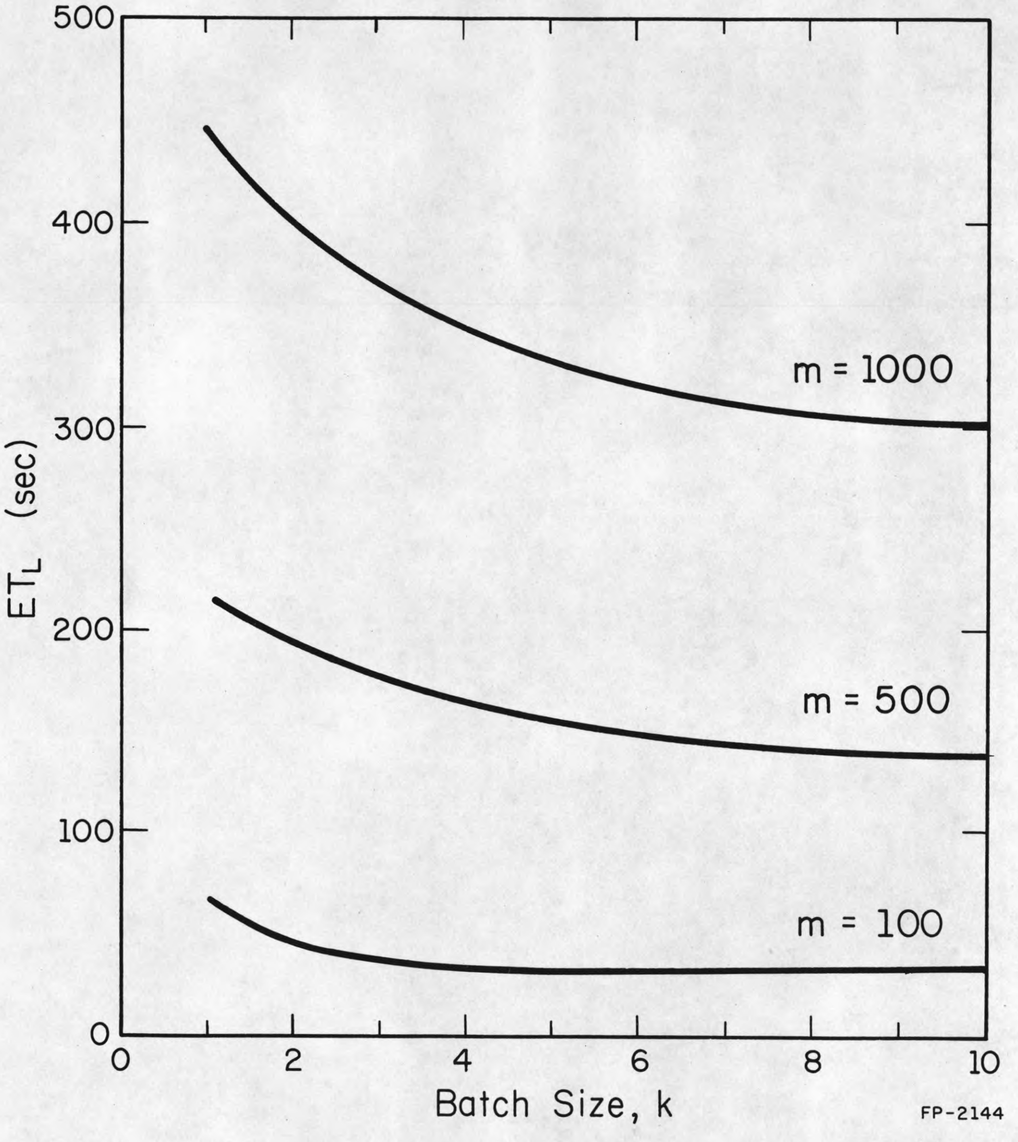
500
400
300
200
100
0
0
2
4
6
8
10
ET_L (sec)
m = 1000
m = 500
m = 100
Batch Size, k
FP-2144

Figure 2.

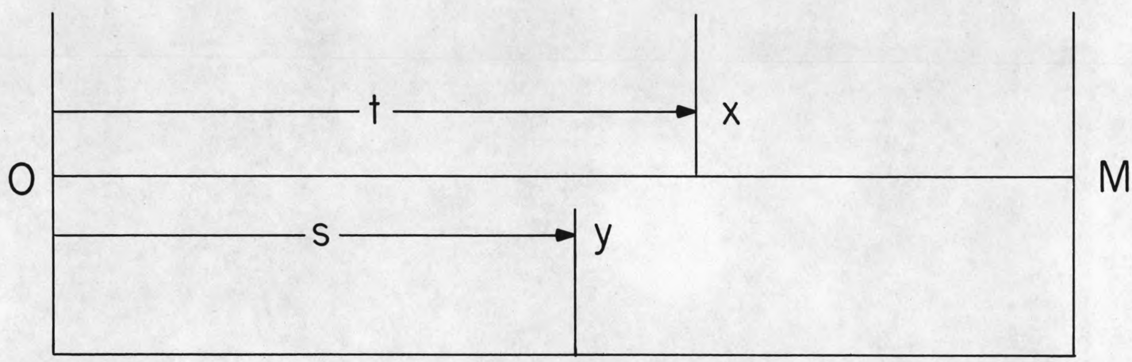

Figure 3.

Distribution List as of 1 November 1970

ESD (ESTI)
L. G. Hanscom Field
Bedford, Mass. 01731 2 copies

Naval Air Systems Command
AIR 03
Washington, D. C. 20360 2 copies

Commanding General
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703
Attn: AMSEL-RD-PB (2 copies)
(Miss F. Morris)

Naval Electronic Systems Command
ELEX 03, Room 2046 Munitions Building
Department of the Navy
Washington, D. C. 20360 2 copies

Director
Naval Research Laboratory
Washington, D. C. 20390
Washinton, D. C. 2027 6 copies

Commander
U.S. Naval Ordnance Laboratory
Attn: Librarian
White Oak, Md. 20910 2 copies

Dr. L. A. Wood, Director
Electronic and Solid State Sciences
Air Force Office of Scientific Research
1400 Wilson Boulevard
Arlington, Virginia 22209 5 copies

Director, Electronic Programs
Attn: Code 427
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217 3 copies

Defense Documentation Center
Attn: DDC-TCA
Cameron Station
Alexandria, Virginia 222314 50 copies

Commanding General
Attn: STEWS-RE-L, Technical Library
White Sands Missile Range
New Mexico 88002 (2 copies)

Commander
Naval Electronics Laboratory Center
Attn: Library
San Diego, Calif. 92152 2 copies

Dr. L. M. Hollingsworth
AFCRL (CRN)
L. G. Hanscom Field
Bedford, Massachusetts 01731

Division of Engineering and Applied Physics
210 Pierce Hall
Harvard University
Cambridge, Massachusetts 02138

Director
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Miss R. Joyce Harman
Project MAC, Room 810
545 Technology Square
Cambridge, Mass. 02139

Professor R. H. Rediker
Electrical Engineering, Prof.
Mass. Institute of Technology
Building 13-3050
Cambridge, Mass. 02139

Raytheon Company
Research Division Library
28 Seyon Street
Waltham, Massachusetts 02154

Sylvania Electronic Systems
Applied Research Laboratory
Attn: Documents Librarian
40 Sylvan Road
Waltham, Mass. 02154

Commanding Officer
Army Materials & Mechanics Res. Center
Attn: Dr. H. Priest
Watertown Arsenal
Watertown, Massachusetts 02172

MIT Lincoln Laboratory
Attn: Library A-082
P.O. Box 73
Lexington, Mass. 02173

Commanding Officer
Office of Naval Research Branch Office
495 Summer Street
Boston, Massachusetts 02210

Commanding Officer (Code 2064)
U.S. Naval Underwater Sound Laboratory
Fort Trumbull
New London, Connecticut 06320

Dept. of Eng. & Applied Science
Yale University
New Haven, Conn. 06520

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-CT-A
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-CT-D
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-CT-I
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-CT-L (Dr. W. S. McAfee)
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-CT-O
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-CT-R
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703
Attn: AMSEL-CT-S

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-DL
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-CC-DD
Fort Monmouth, N. J. 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-KL-D
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-KL-E
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-KL-I
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-KL-SM
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-KL-SM
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-KL-S
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-KL-T
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-NL-A
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-NL-C
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-NL-D (Dr. H. Bennett)
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-NL-P
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-SC
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-VL-D
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-VL-F
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-WL-D
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-XL-DT
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-XL-D
Fort Monmouth, New Jersey 07703

Mr. Norman J. Field, AMCPM-AA-PM
Program Management Division
Project AACOMS
USAECOM, Bldg-2525
Fort Monmouth, New Jersey 07703

Project Manager
Common Positioning & Navigation Systems
Attn: Harold H. Bahr (AMCPM-NS-TM),
Bldg. 439
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703

Dr. H. K. Ziegler, Chief Scientist
Army Member TAC/JSEP (AMSEL-SC)
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703

Mr. I. A. Balton, AMSEL-XL-D
Executive Secretary, TAC/JSEP
U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL-XL-G (Dr. S. Kronenberg)
Fort Monmouth, New Jersey 07703

Commanding General
U.S. Army Electronics Command
Attn: AMSEL -XL-H (Dr. R. G. Buser)
Fort Monmouth, New Jersey 07703

U.S. Army Munitions Command
Attn: Science & Technology
Info. Br., Bldg. 59
Picatinny Arsenal, SMUPA-RT-S
Dover, New Jersey 07801

European Office of Aerospace Research
Technical Information Office
Box 14, FPO New York 09510

Director
Columbia Radiation Laboratory
Columbia University
538 West 120th St.
New York, New York 10027

New York University
Engineering Library
Bronx, New York 10453

Mr. Jerome Fox, Research Coordinator
Polytechnic Institute of Brooklyn
333 Jay St.
Brooklyn, New York 11201

Airborne Instruments Labroratory
Deerpark, New York 11729

Dr. W. R. Lepage, Chairman
Syracuse University
Dept. Of Electrical Engineering
Syracuse, New York 13210

Rome Air Development Center
Attn: Documents Library (EMTLD)
Griffiss Air Force Base, New York 13440

Mr. H. E. Webb (EMBIS)
Rome Air Development Center
Griffiss Air Force Base, New York 13440

Professor James A Cadzow
Department of Electrical Engineering
State University of New York at Buffalo
Buffalo, New York 14214

Dr. A. G. Jordan
Head of Dept. of Electrical Engineering
Carnegie-Mellon University
Pittsburgh, Penn. 15213

Hunt Library
Carnegie-Mellon University
Schenley Park
Pittsburgh, Penn.

Lehigh University
Department of Electrical Engineering
Bethelehem, Penn. 18015

Commander (ADL)
Naval Air Development Center
Attn: NADC Library
Johnsville, Warminister, Penn. 18974

Technical Director (SMUFA-A2000-107-1)
Frankford Arsenal
Philadelphia, Penn. 19137

Mr. M. Zane Thornton, Chief, Network
Engineering, Communications and
Operations Branch, Lister Hill
National Center/Biomedical Communications
8600 Rockville Pike
Bethesda, Maryland 20014

U.S. Post Office Department
Library - Room 6012
12th & Pennsylvania Ave. N.W.
Washington, D.C. 20260

Technical Library
DDR&E
Room 3C-122, The Pentagon
Washington, D.C. 20301

Director for Materials Sciences
Advanced Research Projects Agency
Department of Defense
Washington, D.C. 20301

Assistant Director, (Research)
Office of Director of Defense Research
& Engineering
Pentagon, Rm. 3C128
Washington, D.C. 20301

Chief, R & D Division (340)
Defense Communications Agency
Washington, D.C. 20305

Commanding General
U. S. Army Material Command
Attn: AMCRD-TP
Washington, D.C. 20315

Hq. USAF (AFRDD)
The Pentagon
Washington, D.C. 20330

Hq. USAF (AFRDDG)
The Pentagon
Washington, D.C. 20330

Hq. USAF (AFRDSD)
The Pentagon
Washington, D.C. 20330

AFSC (SCTSE)
Andrews Air Force Base, Maryland 20331

Dr. I. R. Mirman
Hq. AFSC (SGGP)
Andrews AFB, Maryland 20331

Naval Ship Systems Command
Ship 031
Washington, D.C. 20360

Naval Ship Systems Command
Ship 035
Washington, D.C. 20360

U.S. Naval Oceanographic Office
Attn: M. Rogofsky, Librarian (Code 640)
Washington, D.C. 20390

Commander
U.S. Naval Security Group Command
Attn: G43
3801 Nebraska Avenue
Washington, D.C. 20390

Director
Naval Research Laboratory
Washington, D.C. 20390
Attn: Dr. A. Brodizinsky, Sup. Elec. Div.

Director
Naval Research Laboratory
Washington, D.C. 20390
Attn: Maury Center Library (Code 8050)

Director
Naval Research Laboratory
Washington, D.C. 20390
Attn: Dr. W. C. Hall, Code 7000

Director
Naval Research Laboratory
Attn: Library, Code 2029 (ONRL)
Washington, D.C. 20390

Dr. G. M. R. Winkler
Director, Time Service Division
U.S. Naval Observatory
Washington, D.C. 20390

Colonel E. P. Gaines, Jr.
ACDA/FO
1901 Pennsylvania Ave. N.W.
Washington, D.C. 20451

Commanding Officer
Harry Diamond Laboratories
Attn: Mr. Berthold Altman (AMXDO-TI)
Connecticut Ave. & Van Ness St., N.W.
Washington, D.C. 20438

Central Intelligence Agency
Attn: CRS/ADD Publications
Washington, D.C. 20505

Dr. H. Harrison, Code RRE
Chief, Electrophysics Branch
National Aeronautics & Space Admin.
Washington, D.C. 20546

The John Hopkins University
Applied Physics Laboratory
Attn: Document Librarian
8621 Georgia Avenue
Silver Spring, Maryland 20910

Commanding Officer (AMXRD-BAT)
U.S. Army Ballistics Research
Laboratory
Aberdeen Proving Ground
Aberdeen, Maryland 21005

Technical Director
U.S. Army Land Warfare Laboratory
Aberdeen Proving Ground
Aberdeen, Maryland 21005

Electromagnetic Compatibility
Analysis Center (ECAC)
Attn: ACOAT
North Severn
Annapolis, Maryland 21402

Commanding Officer
U.S. Army Engineer Topographic Labs
Attn: STINFLO Center
Fort Belvoir, Virginia 22060

Director (NV-D)
Night Vision Laboratory, USAECOM
Fort Belvoir, Va. 22060

U.S. Army Mobility Equipment Research
and Development Center
Attn: Technical Document Center,
Bldg. 315
Fort Belvoir, Virginia 22060

Dr. Alvin D. Schnitzler
Institute for Defense Analyses
Science and Technology Division
400 Army-Navy Drive
Arlington, Va. 22202

Director
Physical & Engineering Sciences Division
3045 Columbia Pike
Arlington, Va. 22204

Commanding General
U.S. Army Security Agency
Attn: IARD-T
Arlington Hall Station
Arlington, Virginia 22212

Dr. Joel Trimble, Code 437
Information Systems Branch
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217

Commanding General
USACDC Institute of Land Combat
Attn: Technical Library, Rm. 636
2461 Eisenhower Avenue
Alexandria, Virginia 22314

Director
USA Advanced Material Concepts Agency
2461 Eisenhower Avenue
Alexandria, Virginia 22314

VELA Seismological Center
300 North Washington St.
Alexandria, Virginia 22314

U.S. Naval Weapons Laboratory
Dahlgren, Virginia 22448

Research Laboratories for the Eng.
Sciences, School of Engineering and
Applied Science
University of Virginia
Charlottesville, Virginia 22903

Dr. Hermann Robl
Deputy Chief Scientist
U.S. Army Research Office (Durham)
Box CM, Duke Station
Durham, North Carolina 27706

Richard O. Ulsh (CRDARD-IP)
U.S. Army Research Office (Durham)
Box CM, Duke Station
Durham, North Carolina 27706

ADTC (ADBPS-12)
Eglin AFB, Florida 32542

Commanding Officer
Naval Training Device Center
Orlando, Florida 32813

Technical Library, AFETR
(ETV,MU-135)
Patrick AFB, Florida 32925

Commanding General
U.S. Army Missile Command
Attn: AMSMI-RR
Redstone Arsenal, Alabama 35809

Redstone Scientific Information Center
Attn: Chief, Document Section
U.S. Army Missile Command
Redstone Arsenal, Alabama 35809

AUL3T-9663
Maxwell AFB, Alabama 36112

Hq. AEDC (AETS)
Attn: Library/Documents
Arnold AFB, Tennessee 37389

Case Institute of Technology
Engineering Division
University Circle
Cleveland, Ohio 44106

NASA Lewis Research Center
Attn: Library
21000 Brookpark Road
Cleveland, Ohio 44135

Director
Air Force Avionics Laboratory
Wright-Patterson AFB, Ohio 45433

AFAL (AVT) Dr. H. V. Noble, Chief
Electronics Technology Division
Air Force Avionics Laboratory
Wright-Patterson AFB, Ohio 45433

Dr. Robert E. Fontana
Head, Dept. of Electrical Engineering
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

AFAL/TEA
Wright-Patterson Air Force Base, Ohio 45433

Department of Electrical Engineering
Clippinger Laboratory
Ohio University
Athens, Ohio 45701

Commanding Officer
Naval Avionics Facility
Indianapolis, Indiana 46241

Dr. John C. Hancock, Head
School of Electrical Engineering
Purdue University
Lafayette, Indiana 47907

Professor Joseph E. Rowe
Chairman, Department of Electrical
Engineering
The University of Michigan
Ann Arbor, Michigan 48104

Dr. G. J. Murphy
The Technological Institute
Northwestern University
Evanston, Ill. 60201

Commanding Officer
Office of Naval Research Branch Office
219 South Dearborn St.
Chicago, Illinois 60604

Illinois Institute of Technology
Dept. of Electrical Engineering
Chicago, Illinois 60616

Deputy for Res. and Eng. (AMSE-DRE)
U.S. Army Weapons Command
Rock Island Arsenal
Rock Island, Illinois 61201

Commandant
U.S. Army Command & General Staff
College
Attn: Acquisitions, Library Division
Fort Leavenworth, Kansas 66027

Department of Electrical Engineering
Rice University
Houston, Texas 77001

Dr. Billy Welch
USAFSAM (SMC)
Brooks AFB, Texas 78235

HQ AMD (AMR)
Brooks AFB, Texas 78235

USAFSAM (SMKOR)
Brooks AFB, Texas 78235

Mr. B. R. Locke
Technical Adviser, Requirements
USAF Security Service
Kelly Air Force Base, Texas 78241

Director
Electronics Research Center
The University of Texas at Austin
Eng.-Science Bldg. 110
Austin, Texas 78712

Department of Electrical Engineering
Texas Technological University
Lubbock, Texas 79409

Commandant
U.S. Army Air Defense School
Attn: Missile Sciences Div., C&S Dept.
P.O. Box 9390
Fort Bliss, Texas 79916

Director
Aerospace Mechanics Sciences
Frank J. Seiler Research Laboratory (OAR)
USAF Academy
Colorado Springs, Colorado 80840

Director of Faculty Research
Department of the Air Force
U.S. Air Force Academy
Colorado Springs, Colorado 80840

Major Richard J. Gowen
Tenure Associate Professor
Dept. of Electrical Engineering
U.S. Air Force Academy
Colorado Springs, Colorado 80840

Academy Library (DFSLB)
U.S. Air Force Academy
Colorado Springs, Colorado 80840

M. A. Rothenberg (STEPD-SC(S))
Scientific Director
Desert Test Center
Bldg. 100, Soldiers' Circle
Fort Douglas, Utah 84113

Utah State University
Dept. of Electrical Engineering
Logan, Utah 84321

School of Engineering Sciences
Arizona State University
Tempe, Arizona 85281

Commanding General
U.S. Army Strategic Communications
Command
Attn: SCC-CG-SAE
Fort Huachuca, Arizona 85613

The University of Arizona
Dept. of Electrical Engineering
Tucson, Arizona 85721

Capt. C. E. Baum
AFWL (WLRE)
Kirkland AFB, New Mexico 87117

Los Alamos Scientific Laboratory
Attn: Report Library
P.O. Box 1663
Los Alamos, New Mexico 87544

Commanding Officer
Atmospheric Sciences Laboratory
White Sands Missile Range, N.M. 88002

Commanding Officer
(AMSEL-BL-WS-R)
Atmospheric Sciences Laboratory
White Sands Missile Range,
New Mexico 88002

Chief, Missile Electronic Warfare
Tech. Area
(AMSEL-WL-M)
Electronic Warfare Laboratory, USAFCOM
White Sands Missile Range, N.M. 88002

Director
Electronic Sciences Lab.
University of Southern California
Los Angeles, Calif. 90007

Engineering & Mathematical Sciences
Library
University of California at Los Angeles
405 Hilgred Avenue
Los Angeles, Calif. 90024

Aerospace Corporation
P.O. Box 95085
Los Angeles, California 90045
Attn: Library Acquisitions Group

HQ SAMSO (SMTTA/Lt. Belate)
AF Unit Post Office
Los Angeles, Calif. 90045

Dr. Sheldon J. Wells
Electronic Properties Information Center
Mail Station E-175
Hughes Aircraft Company
Culver City, California 90230

Director, USAF PROJECT RAND
Via: Air Force Liaison Office
The RAND Corporation
Attn: Library D
1700 Main Street
Santa Monica, Calif. 90406

Deputy Director and Chief Scientist
Office of Naval Research Branch Office
1030 East Green Street
Pasadena, California 91101

Aeronautics Library
Graduate Aeronautical Laboratories
California Institute of Technology
1201 E. California Blvd.
Pasadena, California 91109

Professor Nicholas George
California Institute of Technology
Pasadena, California 91109

Commanding Officer
Naval Weapons Center
Corona Laboratories
Attn: Library
Corona, California 91720

Dr. F. R. Charvat
Union Carbide Corporation
Materials Systems Div., Crystal Products
Dept., 8888 Balboa Avenue
P.O. Box 23017
San Diego, Calif. 92123

Hollander Associates
P.O. Box 2276
Fullerton, California 92633

Commander, U.S. Naval Missile Center
(56322)
Point Mugu, California 93041

W. A. Eberspacher, Associate Head
Systems Integration Division
Code 5340A, Box 15
U.S. Naval Missile Center
Point Mugu, California 93041

Sciences-Engineering Library
University of California
Santa Barbara, California 93106

Commander (Code 753)
Naval Weapons Center
Attn: Technical Library
China Lake, California 93555

Library (Code 2124)
Technical Report Section
Naval Postgraduate School
Monterey, California 93940

Glen A. Myers (Code 52Mv)
Assoc. Professor of Elec. Engineering
Naval Postgraduate School
Monterey, California 93940

Dr. Leo Young
Stanford Research Institute
Menlo Park, Calif. 94025

Lenkurt Electric Co., Inc.
1105 County Road
San Carlos, California 94070
Attn: Mr. E. K. Peterson

Director
Microwave Laboratory
Stanford University
Stanford, California 94305

Director
Stanford Electronics Laboratories
Stanford University
Stanford, California 94305

Director, Electronics Research
Laboratory
University of California
Berkeley, California 94720

ADDENDA

Mr. A. D. Bedrosian, Room 26-131
U.S. Army Scientific Liaison Off.
Mass. Institute of Technology
77 Massachusetts Avenue
Cambridge, Mass. 02139

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
(Code AX)
Washington, D. C. 20380

Mr. Gordon D. Goldstein
Code 437
Information Systems Program
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217

Mr. H. T. Darracott (AMXAM-FT)
U.S. Army Advanced Materiel
Concepts Agency
2461 Eisenhower Avenue
Alexandria, Virginia 22314

Mr. Jules I. Wittebort, Chief
Electronics Branch
Manufacturing Technology Div.
AFAL/LTE
Wright-Patterson AFB, Ohio 45433

DELETE	REPLACE WITH
Col. E. P. Gaines, Jr. ACDA/FO 1901 Penna. Ave., N.W. Washington, D.C.,20451	Col. E. P. Gaines, Jr. ESD (MCD) L. G. Hanscom Field Bedford, Massachusetts 01730
VELA Seismological Cen. 300 N. Washington St. Alexandria, Va. 22314	VILA Seismological Center 312 Montgomery Street Alexandria, Virginia 22314
Commanding General U.S. Army Strategic Communications Comm. Attn: SCC-CG-SAE Fort Huachuca, Ariz. 85613	Commanding General U.S. Army Strategic Communications Command Attn: SCC-ATS (Mr. Peter B. Pichetto) Fort Huachuca, Arizona 85613

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Coordinated Science Laboratory University of Illinois Urbana, Illinois 61801	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

ON THE RESPONSE TIME OF ON-LINE RETRIEVAL SYSTEMS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

J. T. Cordaro, Jr. and R. T. Chien

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
January, 1971	33	10

8a. CONTRACT OR GRANT NO. DAAB-07-67-C-0199; IBM; NSF GK-2339; b. PROJECT NO. OE C-1-7-071213-4557 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) R-501 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* UILU-ENG 71-2204

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Joint Services Electronics Program through U.S. Army Electronics Command

13. ABSTRACT

In this report the results of an investigation of on-line retrieval systems are presented. The report is divided into two chapters. The first chapter is an analysis of some queuing models. The result is a formula relating search time, response time, and the number of terminals when requests are processed in batches. In the next chapter, response times of linear and inverted files are evaluated. One at a time and batch processing are compared.

DD FORM 1 NOV 65 1473

Security Classification

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Information Retrieval						
Queuing						
Time Sharing Systems						

Security Classification