

1975

## **Estimating Response Time for Auxiliary Memory Configurations with Multiple Movable-Head Disk Modules**

K. Omahen

Report Number:  
75-158

---

Omahen, K., "Estimating Response Time for Auxiliary Memory Configurations with Multiple Movable-Head Disk Modules" (1975). *Department of Computer Science Technical Reports*. Paper 105.  
<https://docs.lib.purdue.edu/cstech/105>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.  
Please contact [epubs@purdue.edu](mailto:epubs@purdue.edu) for additional information.

Estimating Response Time For Auxiliary Memory  
Configurations With Multiple Movable-Head Disk Modules

K. Omahen  
Computer Science Department  
Purdue University  
West Lafayette, IN 47907

CSD-TR 158

Keywords: Response-Time Estimation, Movable-Head Disk Models, Multi-  
Module Configurations, Channel Models

### Abstract

The hardware architecture for a large data base application often involves the use of movable-head disk modules for auxiliary memory. This paper considers design calculations for such systems and is divided into two main sections:

- (1) A survey is given of the literature dealing with queueing models for multiple-module movable head disk configurations. Results for these models allow a system designer to estimate the performance of a specified auxiliary memory configuration; for example, the average file response time is one result typically given. References are also provided to papers related to single movable-head disks which describe techniques for estimating seek time distributions, queueing models for channel operations, etc. which are useful for the multiple-module case.
- (2) A simple method is provided for estimating the average response time for a multi-module configuration of movable-head disk units attached to a single block multiplexer channel. The technique is a synthesis of a method described by Seaman, Lind, and Wilson for analyzing a similar configuration having a selector channel and variations of a method for treating a block multiplexer channel described by Fuller and Baskett. Seaman et al. find the mean response time by viewing the operation of each disk module as an M/G/1 queueing system using the FCFS discipline, where the service time is the sum of the seek time, channel waiting time, and channel service time (rotational delay plus data transfer time). The channel operation in turn is analyzed using the "machine-interference" model (i.e., finite-Poisson-source, single-server system with exponential service time distribution). Fuller and Baskett treat the operation of a channel with rotational position sensing by means of a queueing model with Poisson arrivals (infinite-source) and service process consisting of (a) two exponential stages corresponding to rotational delay and data transmission time, respectively, where the first stage has variable service

(2) rate which is a function of the number of requests at the channel, or (b) one exponential stage with variable service rate dependent on number of requests at the channel system. The proposed technique involves the use of the method of Seaman et al., but replaces the machine-interference model for channel operation with either of two finite-source queueing models similar to those of Fuller and Baskett.

## Introduction

The architecture of the auxiliary memory subsystem plays a critical role in determining the overall performance of a large data base system. For systems of this type, economic considerations often require that movable-head disk units be employed because these devices offer lower cost per bit of secondary storage than that for drums and fixed-head disks yet are capable of achieving reasonable performance levels. This paper is concerned with certain design calculations of interest to a systems analyst, and the material covered is very relevant to the problem of hardware architecture for large data bases. The body of this paper is divided into two main sections as follows:

- A survey is given of the queueing models for auxiliary memory subsystems which have previously appeared in the literature.
- An approximate method is presented for estimating the mean response time for a multi-module auxiliary memory subsystem with movable-head disks and a single block multiplexer channel which employs rotational position sensing.

The objective of this paper is to examine tools available for estimating file response time and to consider ways in which these tools may aid in the design of large data base systems.

## Survey of Queueing Models for Auxiliary Memory Subsystems

The basic equipment configuration to be treated in this paper consists of a single channel attached (through a controller and interface devices) to a number of identical movable-head disk modules as shown in Figure 1.

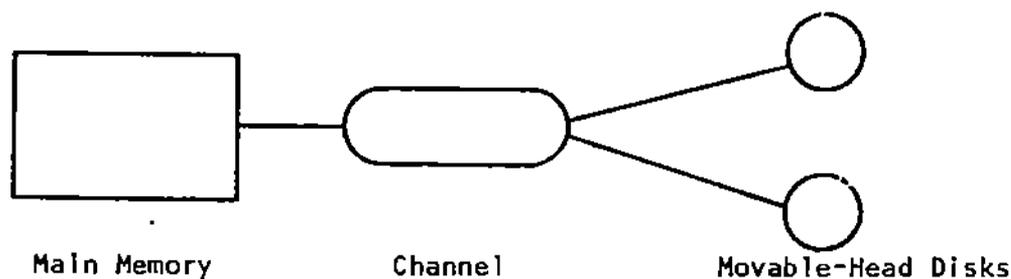


Figure 1. Auxiliary Memory Subsystem To Be Examined.

It is the variations of this basic model which will be the main concern of this survey, but other relevant papers will be mentioned which treat the case of a single movable-head disk module with dedicated channel/controller combination.

### Multiple Movable-Head Disk Modules With Single Channel

Seaman, Lind & Wilson [1] present a method for estimating the response time for an auxiliary memory unit whose configuration consists of  $m$  identical movable-head disk modules which share a selector (non-multiplexed) channel. The authors analyze the performance of the auxiliary memory subsystem for the situation in which the file system residing on the unit has a) direct access organization and b) indexed sequential organization. Case-a will be examined in some detail because an understanding of this model is essential when reading the second section of the paper.

For a file system involving only direct access organizations, records are retrieved and stored directly using a given or generated address. Below is a timing diagram for a typical file access:

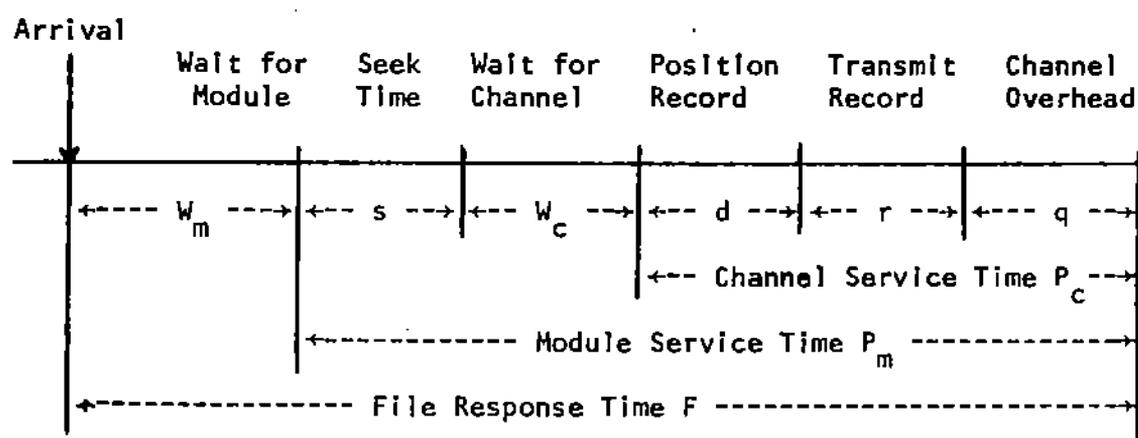


Figure 2. Typical Timing Diagram.

The model for the operation of the auxiliary memory unit takes the following form: One module queue is associated with each of the  $m$  identical devices, and requests arrive at each module queue as an independent Poisson stream with mean rate  $\lambda/m$  (i.e., the overall input rate is  $\lambda$  with requests uniformly distributed among the  $m$  modules). Each module queue is unlimited, and requests within each of these queues are serviced in FCFS order. The form of

the module processing time implies that a request occupies the module from the instant when the seek is initiated until the moment when the channel is released after the completion of data transmission for the request. Once the seek operation has been completed, a request must wait for the channel to become available.

The operation of the channel is modeled as a finite-source queueing system commonly called the "machine-interference model". Each module is treated as a finite source of requests for channel service; events appear to arrive from a module at (Poisson) rate  $w$  when there is no request from that module already waiting for channel service, or at rate zero otherwise. This means that there will be at most one request from each module which will require channel service. The authors assume that the channel service time is exponentially distributed and that the FCFS discipline is used for the channel queue. It is also assumed that seeks may be initiated without the channel being available and that no requests are generated or lost within the memory system. Before sketching out the analysis of the model, the notation used to represent important variables will be given;

$\lambda$  = Poisson input rate for file requests

$m$  = number of modules

$s$  = seek time for request

$d$  = rotational delay for request

$r$  = data transmission time for records

$q$  = channel overhead

$P_c$  = channel service time =  $d + r + q$

$W_c$  = channel waiting time

$P_m$  = module service time =  $s + W_c + P_c$

$\rho_m$  = module utilization =  $(\lambda/m) * E[P_m]$

$\rho_c$  = channel utilization =  $\lambda * E[P_c]$

The method of analysis used by the authors is designed to obtain the expected response time  $E[F]$  for the auxiliary memory unit, and the "given" information for the problem consists of values for the input rate  $\lambda$ , the number of modules  $m$ , and the mean and variance of the distribution for variables  $s$ ,  $d$ ,  $r$ , and  $q$ . Determining the mean and variance for the seek time  $s$  may, in particular, require a good deal of effort (more mention of the problem will be made later in this survey).

The expected response time  $E[F]$  may be determined by using the Pollaczek-Khintchine result for the M/G/1 queueing system (i.e., Poisson arrivals, General service time distribution, single server) which appears below:

$$E[F] = \frac{E[P_m]}{1-\rho_m} \left[ 1 - \frac{\rho_m}{2} \left( 1 - \frac{\text{VAR}[P_m]}{(E[P_m])^2} \right) \right]$$

Before the above result may be applied, it will be necessary to first determine the mean and variance for the module service time  $P_m$  before the above result may be applied. Using the given information, values may be calculated for the channel utilization  $\rho_c$  and for the mean and variance of the channel service time  $P_c$  as shown below:

$$E[P_c] = E[d] + E[r] + E[q];$$

$$\text{VAR}[P_c] = \text{VAR}[d] + \text{VAR}[r] + \text{VAR}[q];$$

$$\rho_c = \lambda * E[P_c] .$$

The mean and variance for the module service time may now be expressed as

$$E[P_m] = E[s] + E[W_c] + E[P_c];$$

$$\text{VAR}[P_m] = \text{VAR}[s] + \text{VAR}[W_c] + \text{VAR}[P_c].$$

The next problem in analyzing the model is to find the mean and variance for the wait in channel  $W_c$ ; once this is done, the response time for the file system can be calculated. The operation of the channel is viewed as a finite-source queueing system with single server and exponential service time distribution. There are  $m$  sources, each corresponding to a disk module, and a variable  $w$  is defined as follows:

$w$  = mean arrival rate for requests for channel service from a module when no request from that module is already in the channel queueing system.

Note that  $w$  is not equal to  $\lambda/m$ ; a value for  $w$  may be found by noting that results for this finite-source system give the channel utilization as a function of  $w$ . Since the value for the channel utilization can be found

using only the given information for the problem, a means is available for solving for the value of  $w$ . The equation below may be solved for  $w$  by using tables for Poisson terms or by an iterative process:

$$\rho_c = \frac{S_{m-1}(z)}{S_m(z)} \quad \text{where } z = w/E[P_c] \text{ and } S_m(z) = \sum_{n=0}^{m-1} \frac{\exp(-z)z^n}{n!}$$

The authors note that the expected wait in channel  $W_c$  is given by (using Little's Equation [2]) the relation  $E[W_c] = L_c/\lambda$  where  $L_c$  is the expected channel queue length. Using a known result for the expected queue length for the finite source system and dividing by  $\lambda$ , the expected wait in queue is found to be

$$E[W_c] = \frac{m}{\lambda} - E[P_c] - \frac{1}{w}$$

The variance for the channel wait is estimated to be the channel queue length variance divided by the input rate; that is,

$$\text{VAR}[W_c] = \frac{1}{\lambda} [(1+z-\rho_c)E[P_c] - (1-\rho_c)(2+z)(\frac{m}{\lambda} - \frac{1}{w})] \text{ where } z=w/E[P_c].$$

Having found the mean and variance for the channel wait, the Pollaczek-Khintchine result may be used to obtain the expected flow time. The method just described has the disadvantage that it will be a nuisance to determine the value of  $w$ ; in order to circumvent this problem for hand calculations, reference [3] (cf. p. 40) provides a table to aid in finding this value. The authors suggest another alternative which is to model the channel operation as a M/G/1 queueing system when there is a large number of modules; this enables the Pollaczek-Khintchine result to be used to estimate the channel flow time. The variance for the channel flow time is approximated as the square of the mean channel flow time (i.e., it is assumed to be exponentially distributed).

Seaman et al. also extend the analysis to deal with the case of indexed-sequential file organizations. It should be stressed that the authors assume that a selector channel is present whose operation is such that, upon the channel being allocated to a module, the rotational delay until the read/write heads reach the start of the record is uniformly distributed between zero and the time for one revolution of the device.

Abate, Dubner, & Weinberg [4] perform a queueing analysis of the IBM 2314 Disk Storage Facility; in this the memory configuration is comprised of a single selector channel, one controller, and eight movable-head disk units. Although the model used to analyze this configuration is somewhat less realistic than the one used by Seaman, Lind, & Wilson, the method of analysis used by Abate *et al.* can obtain more powerful results. Below is a timing diagram for a typical file access; note that the module service time for a request consists only of the time for the seek operation.

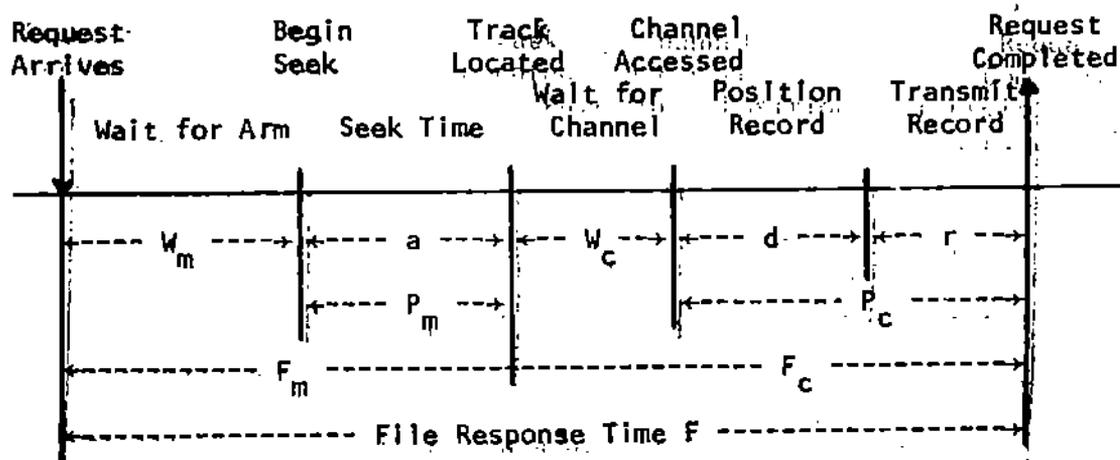


Figure 3. Timing Diagram - Abate, Dubner, and Weinberg.

Requests for file accesses are taken to arrive as a Poisson stream with mean rate  $\lambda$ , and requests are assumed to be uniformly distributed among the eight modules. There is a single independent queue for each device, and requests within each module queue are served in FCFS order. The operation of each module is modeled as a  $M/G/1$  system with arrival rate  $\lambda/8$  and with processing time described by the seek time distribution for requests.

The channel operation is also modeled as a  $M/G/1$  system, but with arrival rate  $\lambda$  and processing time given by the distribution for the sum of the rotational delay and the data transmission time. The overall operation of the auxiliary memory for an specified request is viewed as two independent queues (one module queue and the channel queue) in series. This assumption that the queues are independent is not true, of course, and the arrival stream at the channel queue will not be Poisson; nevertheless, the model should give reasonable results provided that the module utilization is fairly low since the operation of the module queue will be reasonably accurate under these circumstances.

Abate et al. first find the Laplace transforms for the density functions of the seek time, rotational delay, and data transmission time. The memory system operation is treated as two independent M/G/1 queues in series, and a variation of the Pollaczek-Khintchine formula is used to obtain the Laplace transform of the cumulative distribution function for the flow time within each of the queues. These results are then combined by noting that, because the response time is the sum of the flow times through the module queue and the channel queue, the Laplace transform of the density function for the response time equals the product of the density function transforms for the two flow times (note also that if  $\mathcal{F}(s)$  is the Laplace transform for the cumulative distribution, the transform of the density function is  $s\mathcal{F}(s)$ ). A numerical transform inversion technique is then used to find the distribution for the response time; it is the use of transform inversion techniques that is an important feature of this paper.

Gorenstein [5] extends the work of Wang and Ghosh [6] in order to apply known results for the M/G/1 queueing system to the case of multiple movable-head disk modules attached to a block multiplexer channel with rotational position sensing (RPS). Before describing his model, some characteristics of the block multiplexer channel will be discussed. A block multiplexer channel is designed to reduce the rotational latency (see variable  $d$  in Figure 2) involved in servicing data transfers for modules which have completed seeks. In the case of a selector channel, the channel is allocated to one of the active modules and remains busy for both the rotational delay  $d$  and data transfer time  $r$ . Tracks on devices used with a block multiplexer channel are divided into a large number of sector positions, and the channel programs can give commands to "seek" a specified sector positions (assuming access-arm seek completed). The channel is not busy during the sector seek and the module selected for data transfer is the one which is the first to respond with a "sector-seek completed" message from the controller. This rotational position sensing causes a reduction in the average rotational latency per request (and hence in the average channel waiting time) but increases the variance associated with the channel waiting time.

Figure 4 describes the view of the file system operation adopted by Gorenstein; results for the M/G/1 system are again applied to each module. The contribution made by this paper is related to the calculation of the

mean and variance associated with the channel waiting time (including rotational delay) plus record transmission time for a particular request. Gorenstein assumes that the devices are used for paging and that each transfer involves one sector of information; furthermore, the traffic is equally distributed among modules, and there is a uniform distribution of sector references. The cases of both synchronized disks (where the devices are kept in step so that sector start positions pass under heads at the same instants for all modules) and unsynchronized disks; assume Figure 4 is for the case of synchronized disks.

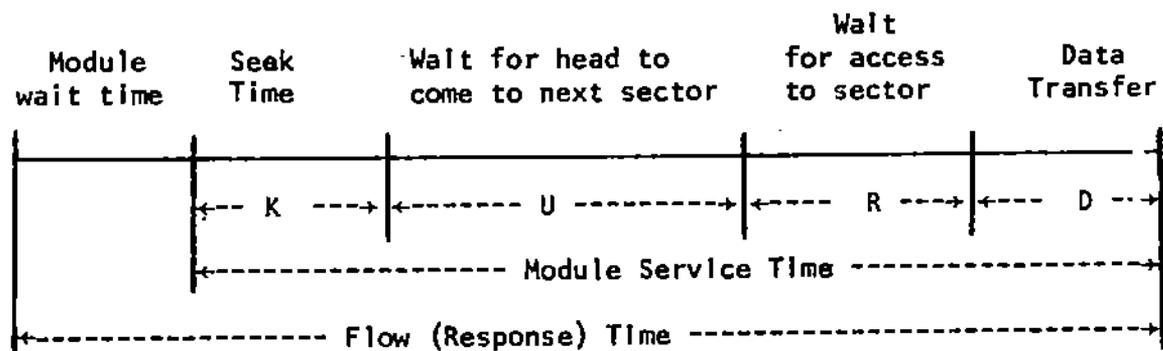


Figure 4. Timing Diagram - Gorenstein.

While it is possible to have overlapped seeks, there can be only one data transfer taking place at any given time, and the random variable  $R$  takes into account this interference between modules. The first two moments for random variable  $R$  are found by making use of results obtained in [6]; these results deal with synchronized devices under fully loaded conditions. Random variable  $U$  is taken to be uniformly distributed between zero and a sector traversal time, and random variable  $D$  is taken to equal a sector traversal time. Gorenstein also treats the more complicated case in which the disks are not synchronized; the effect on the model is that random variable  $U$  is deleted and the channel waiting time is calculated as a combined entity. For each of the two cases, the Pollaczek-Khintchine result is used to find the expected flow time, and the variance of the response time is also obtained from known results for the  $M/G/1$  system.

#### Single Movable-Head Disk Module With Dedicated Channel/Controller

The multiple-module models just described assume that the FCFS discipline is used within each module queue. Other scheduling disciplines may be used

which are concerned with the scheduling the movement of the access arm; the motivation is that it is possible to reduce the average seek time (usually the dominant part of the module service time) and thereby improve file response. The analyses dealing with the scheduling of access arm movement have invariably been for a single module; in the material which follows important algorithms and papers will be briefly covered.

The SSTF (shortest-seek-time-first) algorithm was proposed by Denning [7]; this rule assumes that there is a queue associated with each cylinder or access-arm position and always chooses to next process the cylinder queue which would incur the shortest seek time (relative to the current arm position) of the nonempty queues for the device. Once a cylinder queue gains service, requests are processed in FCFS order until the queue is empty. Denning uses an approximate analysis to obtain an estimate for the average seek time under this rule.

The SCAN rule is a policy in which the access arm is swept back and forth between the innermost and outermost cylinders, stopping at any cylinder having a nonempty queue. This algorithm may be considered as applying the SSTF rule in one direction only (i.e., the access arm is moved in a particular direction and to the nearest nonempty cylinder queue in that direction), where a direction change occurs when either the last track is reached or when there are no requests ahead of the current head position in the given direction. The average seek time was estimated by Denning [7], and this rule was also included in the simulation studies of Teorey and Pinkerton [8] under the name "LOOK" algorithm. Gottlieb and MacEwen [9] determined the average response time under this algorithm by extending the results of Harris [10] for a single-server queue having Poisson arrivals and state-dependent stochastic service rates. Gottlieb and MacEwen suggest handling the multi-module case by adding a constant (representing channel waiting time) to the seek time parameter in the representation for average response time for the single-module case. Coffman, Klimko, and Ryan [11] analyzed an idealized disk model using the SCAN policy and obtained representations for average response time conditioned on cylinder position.

The FSCAN algorithm was suggested by Coffman, Klimko and Ryan [11] as a modification to the SCAN policy. At a decision point, the entire queue of requests is considered, and these requests are serviced in a scan whose

direction is determined by the minimum distance from the outermost and innermost cylinder addresses of the requests to be serviced (that is, the access arm is moved to the nearer of the extreme cylinder addresses before the next scan begins). Requests arriving during a given scan are required to wait until the next scan before being serviced. Coffman *et al.* again analyze an idealized disk model and obtain the average response time conditioned on cylinder position for the single module case.

The CSCAN policy is a rule suggested by Seaman, Lind, and Wilson [1] in which the cylinder queues are serviced in scans which are in only one direction (e.g., from outer to inner cylinder positions). At the end of a scan, the access arm is moved to the outer cylinder position before commencing the next scan. Coffman and Turnbull [12] analyzed this policy for an idealized disk model and obtained the average response time conditioned on cylinder position.

The NSCAN algorithm was first suggested by Frank [13], who made the observation that the SSTF policy only optimized the next seek operation to be executed by the device, not the next  $n$  seeks. Frank observed that the problem of finding the minimum of the next  $n$  seeks is equivalent to the classical graph theory problem of determining the shortest Hamilton path (nodes correspond to access arm positions, and weights on arcs correspond to time for moving access arm between cylinder positions). Coffman *et al.* [11] gave the NSCAN to an algorithm which operates as FSCAN but with a maximum of  $N$  requests served in a crossing.

Weingarten [14] suggested several algorithms which are noteworthy because they involve periodic scanning of the cylinder queues so as to insure that requests to these queues have very predictable response times. The policies described in [14] are somewhat analogous to the above rules in that the cylinder queues are examined in some predefined order.

Some of the results concerning the relative performance of these algorithms will next be presented. All of these algorithms tend to result in shorter average seek times than the FCFS rule. The SSTF algorithm has lower average response time than the other policies but has the disadvantage that the access arm tends to remain in one place on the disk under heavy loading conditions (for this reason, the variance of the response time tends to be large, and the inner and outer cylinder positions receive poorer service than the central cylinders). The SCAN policy and FSCAN policy give better

service to the central cylinders than to the inner and outer cylinders, and the SCAN policy consistently gives smaller average response times than the FSCAN policy (see Coffman *et al.* [11]). The CSCAN does not discriminate against the inner and outer cylinder positions but has higher average response times than the SCAN policy due to the wasted time involved with resetting the access arm at the beginning of a scan. The CSCAN algorithm, however, does tend to exhibit lower variance in the response times.

#### Estimating Seek Times for Movable-Head Devices

The paper by Frank [13] covers techniques for estimating seek time distributions, rotational delay distributions and data transmission time distributions; many examples are given which demonstrate methods for handling the peculiarities of different devices and variations in operational aspects of a device due to characteristics of the files residing in auxiliary memory. The more recent paper by Waters [15] is also recommended because it summarizes many of the results obtained to date concerning seek time estimation.

#### Auxiliary Memory Architecture

The paper by Buzen [16] is an excellent survey of different approaches used in I/O Subsystem Architecture. Brown, Elbsen, and Thorn [17] offers insights in the factors which motivated the introduction of the block multiplexer channel by IBM and discusses a number of other alternatives which were considered. A description of a simulator for an auxiliary memory subsystem employing the IBM block multiplexer channel is given by Boutross, King, and Rutledge [18]; this paper gives quite a bit of information about this channel type.

#### Queueing Models for Channel Operation

The selector (non-multiplexed) channel was included in the models covered in references [1], [2], and [3]. Queueing models for this type of channel assume that the service time consists of a rotational delay (uniformly distributed between zero and the time for one revolution) plus the data transmission time (this may in fact involve searching operations performed by the channel). Poisson arrivals are usually assumed; reference [1] employed a finite-source model with exponential service times (the machine-interference

model), but reference [3] used an infinite-source model with general service times (i.e., the M/G/1 system). In general, one expects to find the finite-source models to be more accurate for the problem at hand, but for large numbers of modules the infinite-source model is preferred for computational convenience.

The block multiplexer channel or a channel employing rotational position sensing has been examined by a number of different authors. The paper by Abate and Dubner [19] was the first to perform an approximate analysis of a device employing rotational position sensing; the authors estimated channel waiting time using intuitive arguments. Fuller and Baskett [20] analyzed two approximate models for this type of channel and compared the performance of these to that of Abate and Dubner [19]. As mentioned earlier, Gorenstein [4] analyzed this channel under paging operation using the results of Wang and Ghosh [5]; data transfers were taken to be a sector traversal time (i.e., one page of information). In contrast, references [19] and [20] allow for arbitrary data transmission time.

#### Estimating Response Time for Multiple Disks With Block Multiplexer Channel

In this section a method will be presented for estimating the expected response time for an auxiliary memory configuration involving a number of movable-head disk modules attached to a single block multiplexer channel. The results are intended to deal with the FCFS discipline being used at each module, and the primary concern is for the case of randomly referenced direct-access files.

The survey of the earlier section gave a detailed description of the method used by Seaman, Lind, and Wilson [1] for handling a similar configuration having a selector channel. Recall that the channel model in this paper was a finite-source single-server queuing system. The method proposed here is to use the method of Seaman *et al.* exactly as described earlier, but to replace the channel model with either of two finite-source analogs of models suggested by Fuller and Baskett [20].

Fuller and Baskett [20] proposed two different infinite-source queuing models which will be briefly described. The channel operation may be viewed as alternating intervals involving rotational latency (waiting for the return of the next "sector-seek completed" response) and data transfer time. If

the sector starting position for requests are uniformly distributed (and the number of sectors large), the rotational latency when there are  $N$  active requests has approximately the distribution of the minimum of  $N$  random variables which are uniformly distributed between zero and the time for disk revolution. Define:

$T_r$  = time for one revolution of the device,

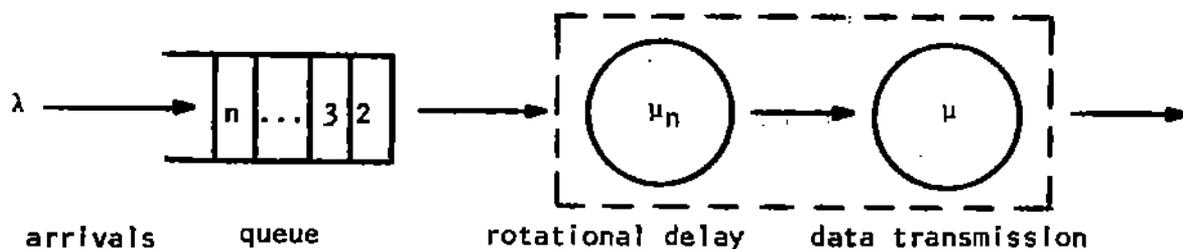
$D_n$  = rotational delay between data transfers when  $n$  requests are at the server (channel).

It may be easily shown that the distribution of  $D_n$  is such that

$$E[D_n] = T_r / (n+1).$$

If the channel service time is again denoted by  $P_c$ , the two models may be easily described. Both assume Poisson arrivals (infinite-source) and a single-server; the models differ in their representation of the service process. Case-1 assumes that the service consists of two exponential states, the first stage having the rotational delay as a function of the number of requests in system and the second representing data transfer time. Case-2 includes only a single exponential stage but again with service rate as a function of the number of active requests. These two models are illustrated in Figure 5.

#### Case-1: Two Exponential Service Stages



$$(1/\mu_n) = T_r / (n+1); \quad (1/\mu) = E[r] = \text{avg. data transmission time.}$$

#### Case-2: One Exponential Service Stage



$$(1/\mu'_n) = E[r] + T_r / (n+1) = (1/\mu) + (1/\mu_n); \quad n \geq 1.$$

Figure 5. Fuller and Baskett Channel Models.

The models for channel operation proposed in this paper are the finite-source equivalents of the above. That is, each Poisson source is taken to represent a particular disk module so that there will be at most one request from each module waiting for channel service. Notation corresponding to reference [1] will be employed for consistency:

$m$  = number of Poisson sources (i.e., number of modules)

$w$  = (Poisson) rate at which requests are generated by each source when active.

**Model-1: Two Exponential Service Stages**

$\mu_n$  = Stage-1 exponential service rate (corresponding to rotational delay) when  $n$  requests are present,

$\mu$  = Stage-2 exponential service rate (corresponding to data transmission time).

**Model-2: One Exponential Service Stage**

$\mu_n^1$  = Exponential service rate when  $n$  requests present.

Parameters  $\mu_n$ ,  $\mu$ , and  $\mu_n^1$  are chosen as in Figure 5, but defined only for  $1 \leq n \leq m$ .

Because the arrival and service processes are continuous Markov processes, the usual technique of solving for state probabilities using the steady-state balance equations may be used. The expected number in system, overall arrival rate, and expected flow time then follow in straightforward fashion.

**Model-1: Finite-Source With Two Exponential Service Stages**

The steady-state probabilities will be represented by the following notation:

$\pi_0$  = Pr[0 requests at server],

$\pi_{i,j}$  = Pr[ $i$  requests for service in system and request in stage- $j$ ],  
 $i = 1, 2, \dots, m; j = 1, 2$ .

It must be kept in mind that, when there are  $k$  requests at the service system, there are  $(m - k)$  active Poisson sources. New requests are serviced only when the previous request has completed both stages of processing. The steady-state balance equations are easily found to be:

$$\pi_0(mw) = \pi_{1,2}\mu$$

$$\pi_{1,1}((m-1)w + \mu_1) = \pi_0(mw) + \pi_{2,2}\mu$$

$$\pi_{1,2}((m-1)w + \mu) = \pi_{1,1}\mu_1$$

For  $2 \leq k \leq m-1$ , we have:

$$\pi_{k,1}((m-k)w + \mu_k) = \pi_{k-1,1}(m-k+1)w + \pi_{k+1,2}\mu,$$

$$\pi_{k,2}((m-k)w + \mu) = \pi_{k,1}\mu_k + \pi_{k-1,2}(m-k+1)w,$$

and

$$\pi_{m,1}\mu_m = \pi_{m-1,2}w,$$

$$\pi_{m,2} = \pi_{m,1}\mu_m + \pi_{m-1,2}w.$$

By elementary manipulations, it is found that the following relations hold:

$$\pi_{m,1} = \pi_{m-1,1}(w/\mu_m)$$

$$\pi_{m,2} = (w/\mu)(\pi_{m-1,1} + \pi_{m-1,2})$$

For  $2 \leq k \leq m-1$ , we have:

$$\pi_{m-k,1} = ((m-k+1)(w/\mu_{m-k}))((1 + kw/\mu)\pi_{m-k-1,1} + (kw/\mu)\pi_{m-k-1,2}),$$

$$\pi_{m-k,2} = ((m-k+1)(w/\mu))(\pi_{m-k-1,1} + \pi_{m-k-1,2}),$$

and

$$\pi_{1,1} = (((m-1)w + \mu)/\mu_1)(mw/\mu)\pi_0,$$

$$\pi_{1,2} = (mw/\mu)\pi_0.$$

It should be clear from the above equations that, by using successive substitutions, all of the  $\pi_{i,j}$  state probabilities can be represented in the form  $\pi_{i,j} = c_{i,j} \pi_0$ . Since the sum of the state probabilities must equal one, we can solve for  $\pi_0$  (and thereby obtain the remaining state probabilities) by means of the relation:

$$\pi_0 = 1 / \left\{ 1 + \sum_{i=1}^m \sum_{j=1}^2 c_{i,j} \right\},$$

The expected number in system,  $L_c$ , is given by

$$L_c = \sum_{i=1}^m \sum_{j=1}^2 \pi_{i,j} \cdot i,$$

and the overall arrival rate of jobs to the system,  $\lambda$ , is given by

$$\lambda = \sum_{i=1}^m \sum_{j=1}^2 \pi_{i,j}^{*(m-1)w} + \pi_0^{(mw)} = w \cdot (m - L_c).$$

Little's Equation [2] gives the expected channel flow time as

$$\begin{aligned} E[F_c] &= E[W_c] + E[P_c], \quad (\text{see notation of Seaman et al. [1]}), \\ &= L_c / \lambda. \end{aligned}$$

The necessary algebraic manipulations have been omitted here, but in practice the actual calculations cause no difficulties. We again note that, given the overall input rate  $\lambda$ , the value of  $w$  must be determined through the use of an iterative computational technique.

#### Model-2: Finite-Source With Single Exponential Service Stage

The analysis proceeds in exactly the same manner but is much simpler for this case; define the following steady-state probabilities:

$$\pi_i^1 = \text{Pr}[i \text{ requests in system}]; \quad i = 0, 1, \dots, m.$$

As for the previous model, the steady-state balance equations are found to be:

$$\begin{aligned} \pi_1^1 &= \pi_0^1 (w / \mu_1^1), \\ \pi_2^1 &= \pi_1^1 (w / \mu_2^1) = \pi_0^1 (w^2 / \mu_1^1 \mu_2^1), \\ &\vdots \\ \pi_m^1 &= \pi_{m-1}^1 (w / \mu_m^1) = \pi_0^1 (w^m / \mu_1^1 \mu_2^1 \dots \mu_m^1). \end{aligned}$$

It follows that

$$\pi_0^1 = 1 / \left\{ 1 + \sum_{j=1}^m \left( w^j / \prod_{i=1}^j \mu_i^1 \right) \right\}.$$

Again denoting the expected number in the channel system by  $L_c$ , we find

$$L_c = \sum_{j=1}^m \pi_j^{1*} j.$$

The overall arrival rate  $\lambda$  is found to be

$$\lambda = \sum_{j=0}^m \pi_j^{1*} (m-j)w = w(m - L_c),$$

and Little's Equation [2] again gives the expected channel flow time as

$$E[F_c] = E[W_c] + E[P_c] = L_c/\lambda .$$

Having described the computational technique for finding the mean channel flow time using either of two models, the following approximation is suggested for estimating the variance of the channel flow time:

$$\text{VAR}[F_c] = \text{VAR}[W_c] + \text{VAR}[P_c] \approx (E[F_c])^2 .$$

That is, we assume the flow time to be exponentially distributed; this is usually a conservative approximation (i.e., it usually overestimates the variance). The rotational delay for a channel with RPS has been observed by several authors [17, 20] to be approximately exponentially distributed; this is to be expected since the minimum of  $n$  identically distributed random variables (i.e., the time until the sector starting address for the request at each module passes under the heads) approaches the exponential with increasing  $n$ . Fuller and Baskett [20] found that their models had a tendency to underestimate the channel flow times when drum units were attached to the channel at higher input rates. In their simulation studies, a device had a tendency to monopolize the channel (in the sense that successive requests serviced by the channel often were associated with the same device) more than predicted by the model; a similar effect was observed by Brown et al. [17] in simulation studies. It may be argued that this should be a less serious problem when movable-head disks are attached because the same module cannot immediately generate another channel request unless the seek time is zero.

#### Sample Calculation

In order to illustrate the use of this technique, the mean response time will be estimated using Model-2 for the following case:

The time for a revolution of the device,  $T_r$ , will be taken to be the basic time unit. There are 8 movable-head disks attached to a single block multiplexer channel, and FCFS scheduling is used at each module. The average data transmission time is equal to one-half the time for a revolution of the device, the average seek time takes 1.03 revolutions, and the variance in the seek time equals 0.28. File requests

are uniformly distributed over the file modules and that on the average 1.0 file requests arrive to the auxiliary memory subsystem during a revolution of the device.

The important quantities for this problem are therefore given by:

$$m = 8 \text{ (no. of modules),}$$

$$E[r] = \frac{1}{2} \text{ (mean data transmission time in units of } T_r),$$

$$E[s] = 1.03 \text{ (avg. seek time)}$$

$$\begin{aligned} \text{VAR}[s] &= 0.28 \text{ (variance in seek time)} \\ &= 1.0 \text{ arrivals/rev.} \end{aligned}$$

Using the value for  $E[r]$ , the service rates are found to be specified by the following:

$$(1/\mu_n) = (1/(n+1)) + (1/2) = (n+3)/(2(n+1)) \quad \text{for } n = 1, 2, \dots, 8$$

We proceed in the analysis by solving for the value of  $w$  through use of the relation

$$\lambda = w(m - L_c).$$

The value of  $w$  corresponding to the given overall input rate  $\lambda$  may be found by initially assuming that  $w_0 = 1/m$  and converging upon the desired value of  $w$  using the relation

$$w_i \leftarrow \lambda / (m - L_c) \quad \text{where } L_c \text{ denotes the expected number in system calculated using } w_{i-1}.$$

When successive values of  $w$  agree to the desired no. of significant digits, the algorithm terminates and the final value of  $L_c$  is found. For this sample problem we obtain:

$$w = 0.166,$$

$$L_c = 1.98,$$

and it follows that the mean channel flow time is

$$E[F_c] = L_c / \lambda = 1.98.$$

The variance of the channel flow time is then approximated as

$$\text{VAR}[F_c] = \text{VAR}[W_c] + \text{VAR}[P_c] = (E[F_c])^2 = 3.92.$$

With the above results and the given information, it is then found that

$$\begin{aligned} E[P_m] &= E[s] + (E[W_c] + E[P_c]) = E[s] + E[F_c], \\ &= 1.03 + 1.98 = 3.01; \end{aligned}$$

$$\begin{aligned}\text{VAR}[P_m] &= \text{VAR}[s] + (\text{VAR}[W_c] + \text{VAR}[P_c]) = \text{VAR}[s] + \text{VAR}[F_c], \\ &= 0.28 + 3.92 = 4.2 \quad ;\end{aligned}$$

$$\rho_m = (\lambda/m)E[P_m] = 0.376.$$

Evaluating the Pollaczek-Khintchine equation, the mean response time is found to be

$$E[F] = 4.34 \quad \text{i.e., 4.34 revolutions of the device.}$$

### Summary

This paper has provided a guide to literature containing information of value for obtaining estimates of auxiliary memory performance. The main concern has been with auxiliary memory subsystems containing multiple movable-head disk modules attached to a single channel by means of a controller and interface devices. The author believes that most of the relevant references for the above case have been included in this survey.

A method was presented for treating multiple movable-head disks attached to a block multiplexer channel or channel employing rotational position sensing. The described method is a synthesis of the technique described by Seaman, Lind, and Wilson [1] for handling a similar configuration with a selector channel and the technique of Fuller and Baskett [20] for treating a block multiplexer channel. Two finite-source queueing models for channel operation, comparable to those proposed by Fuller and Baskett, are presented and analyzed.

The survey made evident that a number of useful techniques exist for the stated problem, some of which contain limitations which may diminish their usefulness in certain instances. The author believes that one area deserving further study involves the use of the rules for scheduling access arm movement for the case of multiple disk modules attached to the same channel. The analysis of such a situation is likely to be difficult, and simulation may be the best tool for studying the problem.

## BIBLIOGRAPHY

1. Seaman, P. H., Lind, R. A., and Wilson, T. L. An analysis of auxiliary-storage activity. IBM Sys. J. 5,3 (1966), 158-170.
2. Little, J. D. C. A proof of the queueing formula  $L = \lambda W$ . Operations Research 9 (1961), 383-387.
3. IBM Staff. Analysis of some queueing models in real-time systems. IBM Form GF20-0007-1, IBM Data Processing Div., White Plains, NY, 1971.
4. Abate, J., Dubner, H., and Weinberg, S. Queueing analysis of the IBM-2314 disk storage facility. J. ACM 15, 4 (1968), 577-589.
5. Gorenstein, S. Queueing models for paging storage devices. IBM Research Rpt. RC 3636, Dec. 1., 1971, 26 pages.
6. Wang, C. P., and Ghosh, S. P. Analysis of modeling of a multimodule disk storage system with sector scheduling. IBM Research Rpt. RJ 596, Aug. 4, 1969.
7. Denning, P. J. Effects of scheduling on file memory operations. Proc. AFIPS SJCC 1967, Vol. 30, 9-22.
8. Teorey, T. J., and Pinkerton, T. B. A comparative analysis of disk scheduling policies. C. ACM 15, 3 (1972), 177-184.
9. Gottlieb, C. C., and MacEwen, G. H. Performance of movable-head disk storage devices. J. ACM 20, 4 (1973), 604-623.
10. Harris, C. M. Queues with state dependent stochastic service rates. Operations Research 15, 1 (1967), 117-130.
11. Coffman, E. G., Klimko, L. A., and Ryan, B. Analysis of scanning policies for reducing disk seek times. SIAM J. on Computing 1, 3 (1972), 269-279.
12. Coffman, E. G., and Turnbull, C. J. M. A note on the relative performance of two disk scanning policies. Info. Proc. Letters 2, 1 (1973), 15-17.
13. Frank, H. Analysis and optimization of disk storage devices for time-sharing systems. J. ACM 16, 4 (1969), 602-620.
14. Weingarten, A. The analytical design of real-time disk systems. Proc. IFIP Congress 1968, Hardware 1, Booklet D, 131-137.
15. Waters, S. J. Estimating magnetic disc seeks. The Computer J. 18, 1 (1975), 12-17.

16. Buzen, J. P. I/O subsystem architecture. Unpublished paper, 35 pages.
17. Brown, D. T., Elbsen, R. L., and Thorn, C. A. Channel and direct access device architecture. IBM Sys. J. 11, 3 (1972), 186-199.
18. Boutross, A. J., King, W. F., III, Rutledge, J. D. Discrete simulation model of direct access storage devices, IBM Research Rpt. RC 3698, Jan. 20, 1972, 13 pages.
19. Abate, J., and Dubner, H. Optimizing the performance of a drum-like storage. IEEE Trans. Comp. C-18, 11 (1969), 992-997.
20. Fuller, S. H., and Baskett, F. An analysis of drum storage units. J. ACM 22, 1 (Jan. 1975), 83-105.