PROJECT MAC

February 20, 1973

Computer Systems Research Division

Request for Comments No. 7

S.M. THESIS PROPOSAL: PERFORMANCE EVALUATION OF MOVEABLE-HEAD DISK SUBSYSTEMS

by Lee J. Scheffler

Abstract: Many activities in the design and performance optimization of general purpose computer systems require consideration of the behavior of their secondary storage disk subsystems. It is preferable in these activities to deal with a single or small set of parameters that summarize the relevant aspects of disk subsystem behavior, than to deal with the myriad detailed disk subsystem internal specifications. Models and analysis are needed to accurately predict this overall behavior of a disk subsystem, given its internal specifications and measures of how it is used by the computer systems.

Introduction

Rotating movable-head magnetic disk subsystems* are widely used in medium- and large-scale general purpose computer systems for on-line** storage of programs and data. There are many phases in the design process of such computer systems when the designers would prefer to deal with some overall measure of the behavior of their disk subsystems, such as access time†, rather than with the multitude of internal disk subsystem specifications. For example, Sekino [12] has developed models which, given the mean page fetch time†† of a time-shared paged virtual memory computer system's secondary memory subsystem (such as disk), along with several other parameters, can be used to predict several aspects of the overall behavior of the computer system. These models have provision only for dealing with disk subsystem mean page fetch time, and cannot make use of any other disk subsystem parameter.

The objective of this thesis is to provide mechanisms to predict the values of a particular disk subsystem overall behavior parameter, access time, as a function of disk subsystem configuration specifications and the pattern of disk subsystem use by the computer system. Such mechanisms consist of:

- a set of parameters to characterize the load, or pattern of use of a disk subsystem by a computer system;
- a set of parameters to characterize the configuration of a disk subsystem;
- a set of parameters to characterize the performance (access time) of a disk subsystem with a specific configuration when presented with a specific load;
- a definition of a disk subsystem interface, at which the above parameters are defined;

---

* The intuitive notion of a disk subsystem as a set of disk drives together with means for controlling and accessing them will suffice for this introductory discussion.

** accessible without operator intervention

† time between the arrival of an access (read or write) request at the disk subsystem interface and the completion of reading or writing for that request

†† time to fetch (read) a single page (fixed-size block) from secondary memory into primary memory

- a model, or set of models, to characterize the operations inside a disk subsystem, and to serve as analysis vehicles;

- analyses to provide expressions or computational algorithms relating disk subsystem performance parameters to disk subsystem configuration and load parameters; and

- metering methods to obtain values for disk subsystem load parameters to be used in actual performance predictions based on these analyses, and to obtain values for disk subsystem performance parameters to verify the results of these analyses.

For analytic simplicity, the disk subsystem configurations to be considered will be constrained as follows:

(1) All data channels in the disk subsystem can be used to send instructions to, receive status information from, and transmit data to and from, all disk units in the subsystem.

(2) No favoritism of using particular channels to access particular disk units is allowed.

(3) All disk units in the subsystem have the same mechanical and electronic specifications: rotational speed, head positioner mechanism characteristics, storage capacity, and data organization.

(4) All data channels have the same data transmission capacity.

These four constraints allow disk units and data channels to be viewed as identical and independent servers in various phases of the model(s) and analyses where such an assumption is crucial to mathematical tractibility.

It is believed that most of the disk subsystems commonly found in today's computer systems are included by these constraints. Disk subsystems composed of several independent partitions, each consisting of a channel and several disk units, can be made to fit into these constraints if each partition is viewed as a separate independent disk subsystem. More complex interconnection possibilities, such as disk subsystems with several disk units accessible only via one channel, several units accessible only via another channel, and several common units accessible over either channel, are specifically excluded from this thesis work. Such subsystems are not very common, and the problems posed by such interconnections are separate issues from the main focus of this thesis.

A further constraint on the disk subsystem to be considered is that the disk subsystem have complete control over the physical placement of data or program portions on disk units and locations within disk units. The

implication of this constraint is that a disk subsystem designer or vendor can, by appropriately designing the disk subsystem space allocation algorithm, attempt to match the logical accessing characteristics of the load supplied by the computer system to the physical accessing characteristics of his equipment to improve performance. For simplicity, only allocation algorithms having the following two properties will be considered:

(5) Allocation of data or program portions across disk units is such that there is an equal probability of any randomly chosen access request being directed to any particular disk unit.

(6) The allocation pattern of data or program portions across the cylinders of a disk is the same for all disk units.

These constraints may be relaxed or augmented during the course of the research if it is found that they are either too severe to allow models and analyses to be widely applicable to existing disk subsystems, or that they are insufficient to allow the solution of performance prediction problems without a great deal of complexity.

Note that no explicit consideration is given to the specific nature of the computer systems in which a disk subsystem is to be used. This is intentional. It is desired that all relevant information about the structure of a computer system be embodied in the values of disk subsystem load parameters. It is the purpose of the metering methods to be developed to map the relevant details of computer system operation into these load parameter values.

The following sections expand upon these ideas.

Previous Work

Although much quantitative work has been done concerning movable-head disks, very little of it deals with disk subsystems. Detailed attention has been given to the opimization of read-write head movement across the surface of a single disk [1,5,8,13,20]. In subsystems with many disks, the scheduling of control information and data flow over the disk subsystem channels is the more serious problem. Nevertheless, this and other work done on related but different rotating storage devices (movable-head disks with independent positioning mechanisms for each surface [7], and sectored drums [4,5]) provide a wealth of terminology and analysis methods on which to draw.

Several analytic and simulation models have been developed for predicting

disk subsystem throughput (number of access requests completed per unit time) [10,21]. Disk subsystem throughput, although interesting in certain applications, is not a complete characterization of a disk subsystem's behavior. Two vastly different disk subsystems, one with a very short mean access time, and one with a very long mean access time, can both exhibit the same throughput (although mean queue lengths in each will differ). Yet it is doubtful that both of these disk subsystems would fulfill any arbitrary computer system's secondary storage needs identically well. The assumptions made in the name of simplicity prevent these models from being sufficiently general to be widely applicable. And the validity of the use of simulations, based on restrictive assumptions, in predicting disk subsystem behavior, or in verifying analytic predictions, is questionnable.

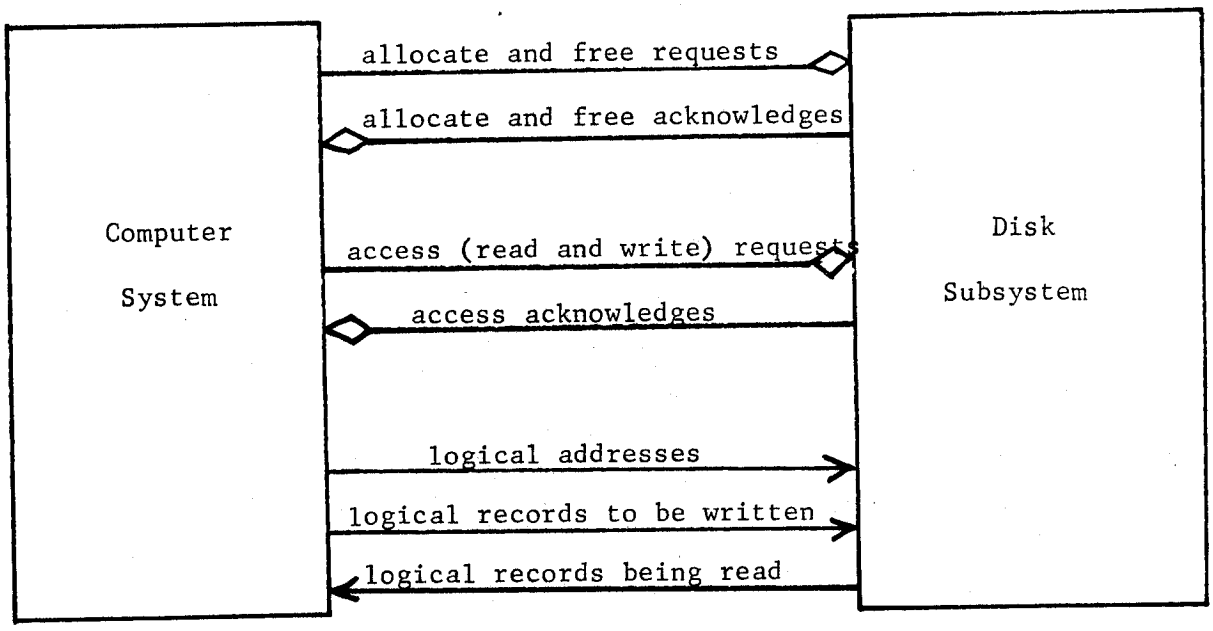
Model Interfaces

A clear definition of the interfaces between disk subsystem and computer system is necessary, as it is at these interfaces that load and performance parameters take on meaning. Figure 1 depicts such an interface drawn on functional grounds, and the elements of communication between computer system and disk subsystem across the interface. Communication consists of requests, acknowledges, and data. Requests are of two types: allocation (allocate and free) and access (read and write). Allocate requests request the assignment of a logical record* of program or data to a physical record* on some disk unit. Free requests destroy such an assignment and make the physical record available for later allocation**. Access requests request the movement of a logical record between computer system main memory and the physical disk record to which it is assigned. A read (or fetch) request moves a logical record into main memory, without destroying the copy on disk. A write (or store) request moves a logical record from main memory to disk, freeing the physical main memory space for other logical records, and overwriting the old logical record copy on disk. Acknowledges indicate the completion of the requested operations.

---

* A "logical record" would correspond to a "page" in a paged virtual memory environment, or a "record" in a non-virtual file system. A "physical record" is a contiguous set of disk sectors (basic disk storage units) that can hold exactly one logical record.

** There are two ways in which allocate and free requests can be used: only on the creation and deletion of a logical record, or on every write and read of a logical record. In this thesis, the former will be assumed, that once a logical record is assigned to a particular disk record, the assignment is permanent until the logical record is deleted.




indicates control information flow

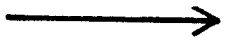

indicates data flow

Figure 1

Interfaces and elements of communication
between computer system and disk subsystem

Data is also of two types: logical addresses and logical records. By virtue of the interface definition, the computer system knows nothing about physical disk subsystem addresses, and communicates entirely via logical addresses. (A logical address might correspond to a [segment name, page number] pair in a segmented and paged memory system, or [file name, record number] in a file-oriented system.) Data flows between computer system and disk subsystem in single logical records, transmitted one at a time.

Parameters

The purpose of disk subsystem configuration parameters is to allow the various disk subsystem configuration alternatives to be specified in ways that can be directly taken into account in performance analyses. Such parameters describe numbers of disk units, numbers of channels, rotational speed and storage capacity of disk units, organization of physical disk records, and characteristics of read/write head positioning mechanisms.

The load placed on a disk subsystem is the pattern of arrival of access requests at the disk subsystem interface. Allocation requests need not be considered part of this load, because they do not require the use of any disk channel or disk unit, and therefore do not interfere with the service of access requests. Load parameters describe relationships between successive access requests: logical addresses, methods of access (read or write), request priorities (order of request service dictated by the computer system), record sizes, and times since the last access requests of various types. Real computer systems provide many simplifications on the many possible relationships. In a paged memory system, all records transferred are the same size. In a highly multiprogrammed computer system*, one expects a very low correlation between the logical addresses of successive access requests. Priority schemes for disk access requests rarely exceed two priority levels: a high level for access requests where speed is important (such as demand page fetches in a demand paged virtual memory), and a low level for other access requests. And the times between arrivals of requests of different priorities are less important to queueing analyses than times between successive requests of the same priority.

---

* many programs multiplexing main memory and processors at the same time

The primary performance parameters to be considered are access time distributions for each of the various types of access requests (reads, writes, high and low priority requests). Such distributions will find use in system design and analysis problems of various kinds: higher-level performance analysis (overall system response time and throughput), disk subsystem configuration choice, and system and disk subsystem tuning.

Equations describing these distributions in terms of configuration and load parameters are clearly the most desirable means of evaluating these parameters. In view of the complexities involved, the obtaining of such equations is more a hope than a goal of this research. Lacking these, computational algorithms for evaluating these distributions will satisfy the system design and analysis information needs. As this research is clearly aimed at analytic study of disk subsystem behavior, simulation is not viewed as an acceptable technique for evaluating access time distributions. The reasons for this include the expense of simulation, the large amount of non-insightful work involved in setting up simulation models, and the difficulty of changing such models once they are constructed.

Models and Analyses

Queueing models are the most natural type of model to deal with disk subsystem operation. It is desirable to construct a single model that is sufficiently general that most of the disk subsystems included in constraints 1-6 above can be viewed as specific cases of the model. One promising approach involves viewing a disk subsystem model as being composed of a hierarchy of lower-level models, as depicted in figure 2. Scheduling of disk requests is the outermost level. It is here that issues of priority among requests are handled. With an appropriate feedback model, pre-emption of requests already part way through Disk Unit Services might be represented. Once a request is scheduled, barring pre-emption, it first requires the read-write head assembly on the appropriate disk unit to seek for the desired cylinder, with a short burst of channel usage to start the seek operation. It is here that effects of seek-time minimization techniques such as those in [1,5,8,13,20] are accounted for. Once the heads are positioned, data must be rotated under the heads and read (or written). There is room at this level for latency-minimizing strategies such as reading an entire track as soon as possible and reconstructing

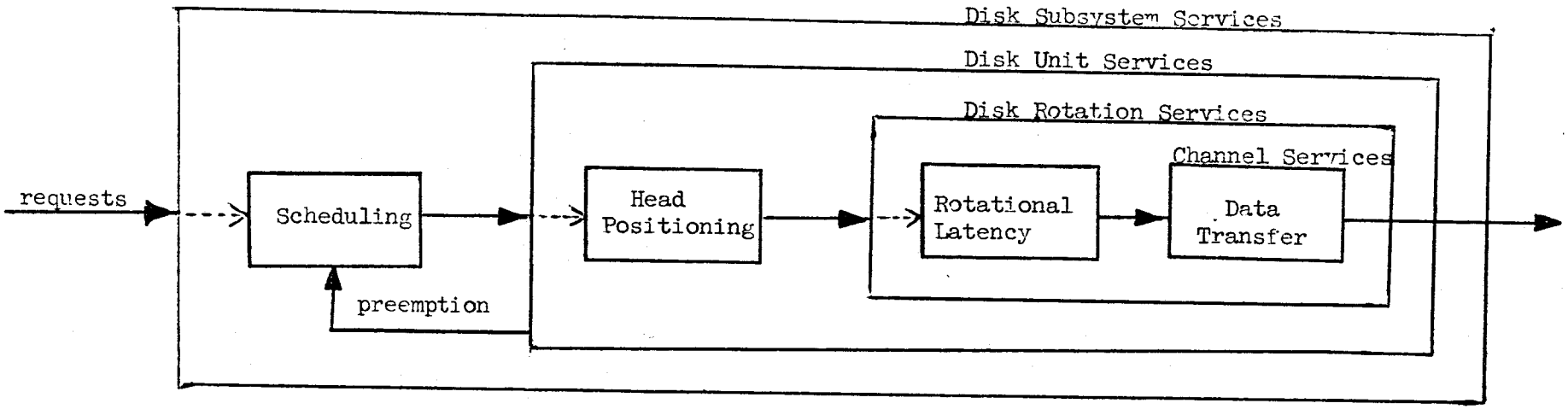


Figure 2

A disk subsystem model as a hierarchy of lower level models

the desired record from the track of data [8], and transmitting data over a channel for the request for the disk unit where data will arrive under the heads soonest [15]. Or, this model can be further broken down into a rotational latency model, and a data transmission model where differences in transmission times for different request sizes can beexplicitly taken into account.

This model is still in very rough form, but it seems to offer hope of breaking down the complexity of operations inside a disk subsystem into understandable pieces. This nest of queueing models leads to an analysis approach where the waiting time in one queueing system (queueing time plus service time) becomes an additive element of the service time of the next outer queueing system, and the arrival discipline of one queueing system is the queue departure discipline of the next outer queueing system. This leads to a system of equations which can probably be solved analytically or numerically for the desired access times (passage times through the outermost model).

It is an unfortunate property of the model of figure 2 that most request arrival and service disciplines of the various models will not be of a character easily dealt with analytically. Thus, established queueing formulas for waiting time distributions of M/M/1, M/G/1, and G/M/1 queueing systems will be of little use in analyzing this model. Analysis will have to be on a more basic level. The two directions which seem fruitful are the development of iterative procedures for evaluating G/G/1 queueing system distributions, and the use of semi-Markov models with non-exponential holding times to represent queue arrivals and departures.

Metering

For these models and analysis methods to be of any practical value in already existing computer systems, there must exist means to quantify the actual load that a computer system presents to a disk subsystem. Such values become inputs to formulas and computational algorithms to produce access time distributions. Metering methods are needed to obtain needed information without perturbing the characteristics being measured. For example, one can obtain request inter-arrival time histograms in systems with a program-readable real-time clock [11]. If a simple curve can be fit to such a histogram, there is some hope of obtaining fairly accurate analytic solutions. Alternatively, and more likely, such data can become direct input to numerical algorithms to

produce numerical access time predictions.

Metering methods are also needed to measure actual disk subsystem access times to validate the above analyses. Complete distributions or histograms may not be needed here, as long as those aspects of the access time distributions generally considered important (mean, variance, 90-percentile point) are verified.

Course of Research

The flow graph of figure 3 depcits the course I expect this research to follow. The parallel paths at various points are meant to indicate work that will go on in parallel.

A preliminary literature survey was completed last year as a part of my work on the DSU-170 (IBM 2314) disk subsystem Device Interface Module (DIM) for the Multics time-sharing system. My experience with the implementation of an overlapped-seeks disk accessing strategy [18] on the DSU-170 disk subsystem, and in constructing the analyses behind the design decisions for the DSU-181 and DSU-190 disk subsystem interfaces to Multics [17] have given me a solid grounding on the capabilities and features of state-of-the-art disk subsystems.

Using the disk subsystem interface definition and load parameters suggested here, I am currently metering the load placed on the DSU-170 (movable-head) and DSU-270 (fixed head) disk subsystems by the Multics system. Specific parameters being measured include request inter-arrival times, correlations between arrivals of read, write, and high and low priority requests, and correlations between successive requests for the same disk unit. Preliminary results indicate a very strong exponential character in the inter-arrival times between successive high priority requests.

I have begun a detailed study of the previous work on disk performance analysis, and on computer system performance analysis; the former for consistency in terminology, and the latter for applicability of this research to overall computer system performance evaluation.

I have evolved an access time model for disk subsystems with a single channel, and am currently pursuing numerical methods for evaluating the desired access time distributions[19].

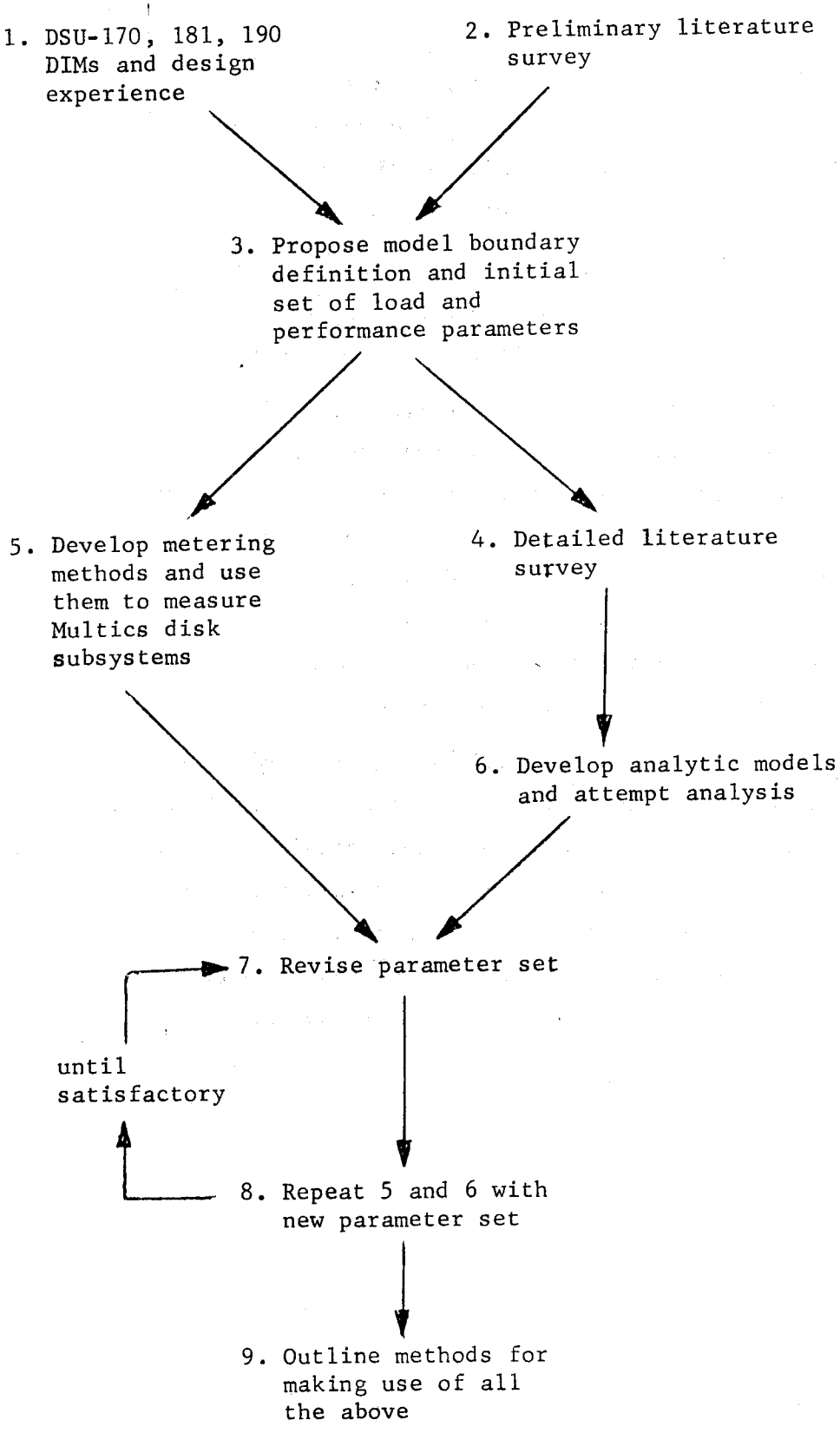


Figure 3

Projected course of thesis research

Subjects I have taken on probaility and queueing theory, several references on probability theory for computer systems [2,3,6,9], the many good probability texts around, and the several probability experts on campus should help me over the rough spots of analysis.

I anticipate that a first pass through analysis and metering will be necessary to place things in their proper perspective, and so expect the process of developing models and analyses to take several iterations.

Finally, since the whole purpose of this work is to make decisions by designers and customers of computer systems easier, a discussion of the practical application of these analyses is very much in order as the final phase of the thesis. Optimistically, programs might be developed which, given adequate parameterization of a disk subsystem (from the manufacturer) and of the load a computer system will place on it (from the computer system design specifications, metering, or performance models of other activities), would be used to predict actual access times.

It is hoped that the end result of this thesis research will be a useful compact theory of performance evaluation of movable-head disk subsystems, including models, analyses, metering methods, computational algorithms, and significant results about common disk subsystem configurations. Models and analyses will be verified as far as possible using metering implemented on Multics for the many possible disk subsystem configurations represented by the DSU-170, DSU-181, and DSU-190 disk subsystems.

BIBLIOGRAPHY AND REFERENCES

1. Abate, J., Dubner, H., and Weinberg, S. B., "Queueing Analysis of the IBM 2314 Disk Storage Facility", JACM, volume 15, number 4, October 1968, pp577-589.

2. Chang, W., "Queues with Feedback for Time-Sharing Computer System Analysis", ORSA Journal, volume 16, number 3, 1968, pp613-627.

3. _____, "Single-Server Queueing Processes in Computing Systems", IBM Systems Journal, volume 9, number 1, 1970.

4. Coffman, E. G., "Analysis of a Drum Input/Output Queue Under Scheduled Operation in a Paged Computer System", JACM, volume 16, number 1, January 1969.

5. Denning, P. J., "Effects of Scheduling on File Memory Operations", Proc. AFIPS, 1967 Spring Joint Computer Conference, volume 30, pp9-21.

6. _____, "Queueing Models for File Memory Operations", S.M. thesis, TR-21, MIT Project MAC, October 1965.

7. Fife, D. W., and Smith, J. L., "Transmission Capacity of Disk Storage Systems with Concurrent Armo Positioning", IEEE Transactions on Electronic Computers, August 1965, pp575-582.

8. Frank, H., "Analysis and Optimization of Disk Storage Devices for Time-Sharing Systems", JACM, volume 16, number 4, October 1969, pp602-620.

9. Kleinrock, L. and Coffman, E. G., "Feedback Queueing Models for Time-Shared Systems", JACM, volume 15, number 4, October 1968, pp549-576.

10. Rice, D., "An Analytical Model for Computer System Performance Analysis", Ph.D. thesis, University of Florida, 1971.

11. Scheffler, L. J., "Performance Measures for Movable-Head Disk Subsystems", working paper.

12. Sekino, Akira, "Performance Evaluation of Multiprogrammed Time-Shared Computer Systems", Ph.D. thesis, TR-103, MIT Project MAC, September 1972.

13. Teorey, T. J., and Pinkerton, T. B., "A Comparative Analysis of Disk Scheduling Policies", Proc. Third Symposium of Operating Systems Principles, October 1971.

14. Drake, A. W., Larson, R. C., Folk, J. F., Tien, J. M., "Lecture Notes and Problems for 6.536, Probabilistic Models in Systems Engineering and Operations Research", MIT Department of Electrical Engineering, 1971.

15. "DSS-190 Removable Media Disk Subsystem", Engineering Product Specification, Honeywell Information Systems Inc., 1971.

16. No reference available yet.

17. Scheffler, L. J., "Crude Estimates of Multics Follow-On Disk Subsystem Performance", Multics Performance Log number 67, MIT Project MAC, June 1972.

18. Scheffler, L. J., "Revision, Clarification, Pre-Analysis and Simulation of Proposed DSU-170 DIM Modifications", MIT Project MAC internal memo, Dec. 1971.

BIBLIOGRAPHY AND REFERENCES (cont)

19. Scheffler, L. J., "A Response-Time Model for Movable-Head Disk Subsystems with Parallel Seeking and Serial Data Transmission Over a Single Channel", working paper.

20. Lynch, W. C., "Do Disk Arms Move?", Performance Evaluation Review, ACM SIGME, volume 1, number 4, December 1972.

21. Stone, David L., and Turner, R., "Disk Throughput Estimation", Proceedings of the ACM Annual Conference, August 1972.