STAN-CS-73-351

SU-SEL-73-010

Performance of An I/O Channel With Multiple Paging Drums

(Digest Edition)

by

Samuel H. Fuller

August 1972

Technical Report No. 27

This work was conducted while the author was a Hertz Foundation Graduate Fellow at Stanford University, Stanford, Calif., and was partially supported by the Joint Services Electronics Program, U.S. Army, U.S. Navy, and U.S. Air Force under Contract N-00014-67-A-0112-0044. Computer time was provided by the Atomic Energy Commission under Contract AT-(0403) 515.

DIGITAL SYSTEMS LABORATORY

STANFORD ELECTRONICS LABORATORIES

STANFORD UNIVERSITY · STANFORD, CALIFORNIA

STAN-CS-73-351 SEL 73-010

PERFORMANCE OF AN I/O CHANNEL WITH MULTIPLE PAGING DRUMS

(Digest edition)

by

Samuel H. Fuller

August 1972

Technical Report no. 27

DIGITAL SYSTEMS LABORATORY

Dept. of Electrical Engineering Dept. of Computer Science

Stanford University

Stanford, California

This work was conducted while the author was a Hertz Foundation Graduate Fellow at Stanford University, Stanford, Calif., and was partially supported by the Joint Services Electronics Program, U.S. Army, U. S. Navy and U.S. Air Force under Contract N-00014-67-A-0112-0044. Computer time was provided by the Atomic Energy Commission under Contract AT-(04-3)515.

PERFORMANCE OF AN I/O CHANNEL WITH MULTIPLE PAGING DRUMS

ABSTRACT

For rotating storage units, a paging drum organization is known to offer substantially better response time to I/O requests than is a more conventional (file) organization [Abate and Dubner, 1969; Fuller and Baskett, 1972]. When several, asynchronous paging drums are attached to a single I/O channel, however, much of the gain in response time due to the paging organization is lost; this article investigates the reasons for this loss in performance.

A model of an I/O channel with multiple paging drums is presented and we embed into the model a Markov chain that closely approximates the behavior of the I/O channel. The analysis then leads to the moment generating function of sector queue size and the Laplace-Stieltjes transform of the waiting time. A significant observation is that the expected waiting time for an I/O request to a drum can be divided into two terms: one independent of the load of I/O requests to the drum and another that monotonically increases with increasing load. Moreover, the load varying term of the waiting time is nearly proportional to $(2 - 1/k)$ where k is the number of drums connected to the I/O channel. The validity of the Markov chain approximation is examined in several cases by a comparison of the analytic results to the actual performance of an I/O channel with several paging drums.

DISCUSSION

The performance of computer systems has become increasingly dependent on secondary storage in the past years as the relative performance gap has widened between central processors on the one hand, and secondary storage on the other. This article focuses attention on the performance of one major form of secondary storage, the drum-like storage unit. Examples of drum-like stores include fixed-head disks, semiconductor storage units built from large shift registers, magnetic bubble shift registers, and delay lines, as well as storage units that actually contain physically rotating drums as shown in Fig. 1.1. The purpose of this article is to investigate exactly how the performance of several drums attached to the same I/O channel, as shown in Fig.1.2, compares to the simpler case of a single drum per I/O channel.

An attractive organization for a drum storage unit is as a *shortest-latency-time-first* (SLTF), *paging drum*, often just called a *paging drum*; Fig. 1.1 is an illustration of a paging drum. The drum rotates at a constant angular velocity and information is read or written onto the surface of the drum as the surface passes under the fixed read-write heads. That fraction of the drum's surface that passes under a particular read-write head is called a *track*, and a block of information, or a *record*, is recorded serially onto a track. In a paging drum, the angular co-ordinate of the drum is partitioned into a number of equal size intervals called *sectors*. Furthermore, we impose the restriction that records be stored at the intersection of a track and a sector. In other words, we require that records start at sector boundaries and all records are the same size, the length of a sector. The scheduling policy used by a paging drum is , as we already mentioned, shortest-latency-time-first.

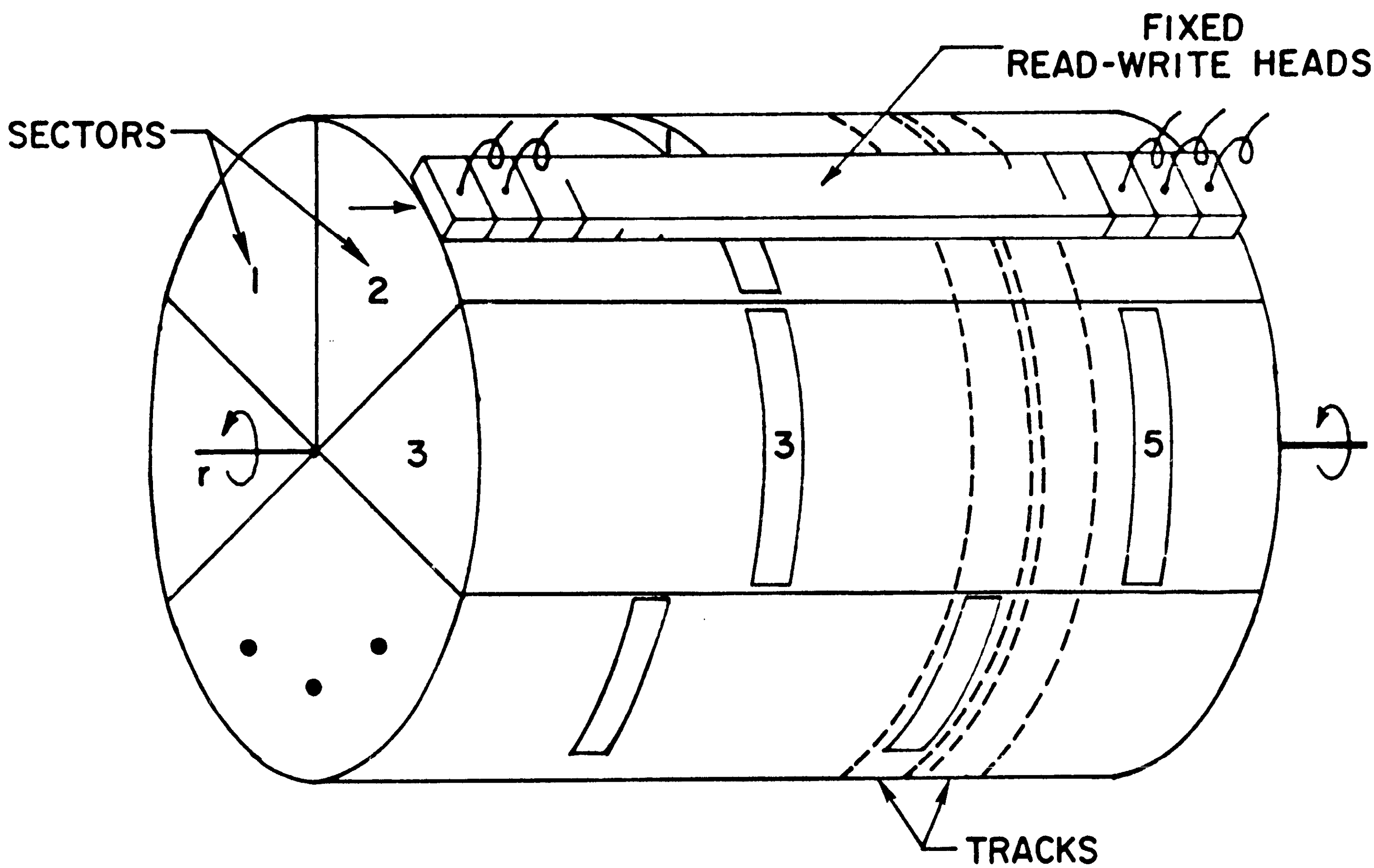

Figure 1.1. A drum storage unit organized as a paging drum.

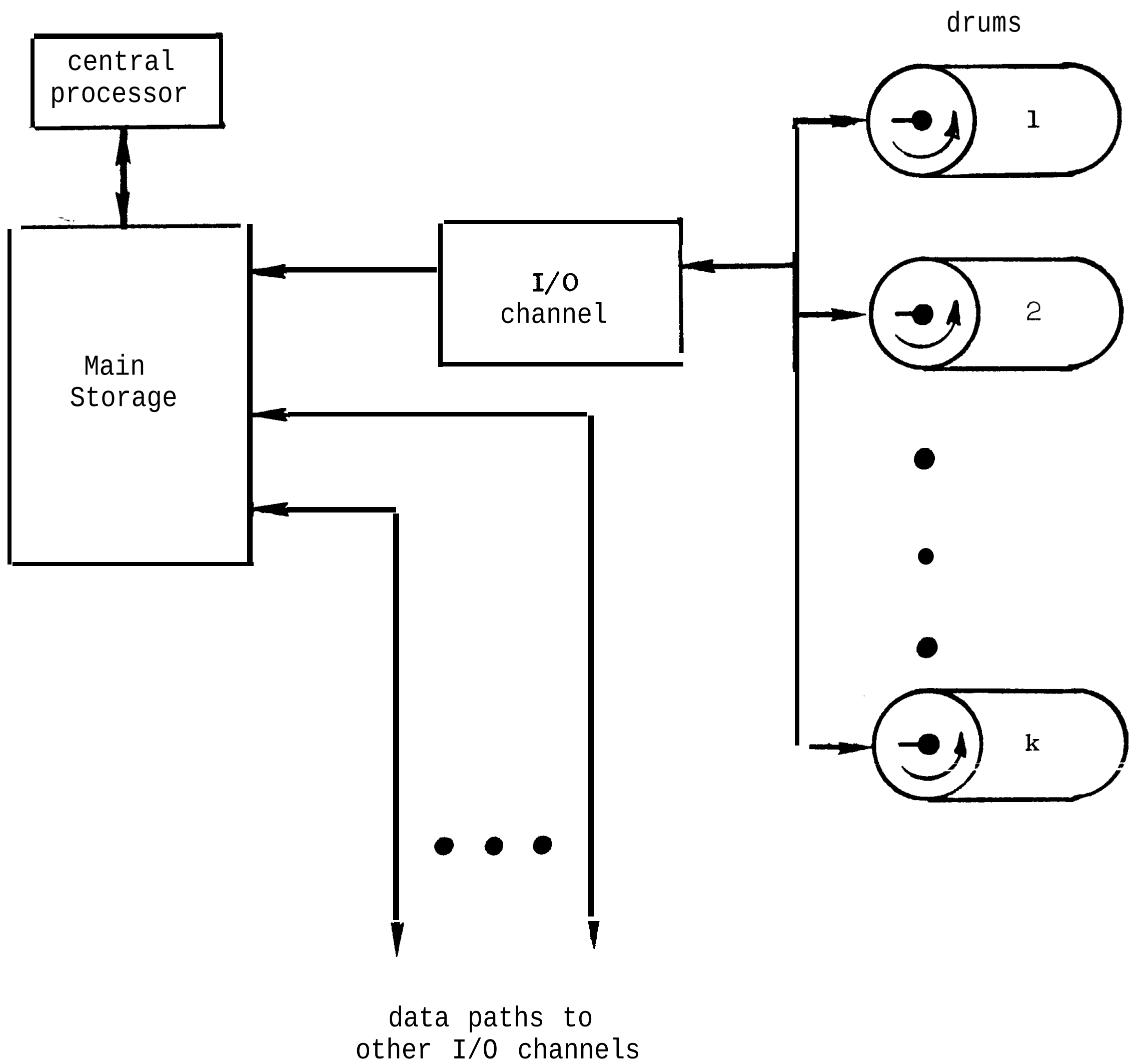

Figure 1.2. Data paths between drums, I/O channel, and main storage.

This simply means that when a new sector arrives at the read-write heads, if there is an outstanding I/O request on the sector it will begin service immediately. If more than one I/O request requires service at a particular sector, a first-in-first-out (FIFO) policy is used. In Fig. 1.1, if requests to read records 2, 3, and 5 in sector 3 arrived in the order 3, 5, 2, then the paging drum will begin servicing record 3 as soon as it finishes processing record 1 in sector 2.

There are several important reasons for using a paging drum organization rather than some more general, less constrained organization. First, fixed size records considerably ease the task of managing space on the surface of the drum. If variable size records are allowed, the same problems of (external) memory fragmentation that have received so much attention in main storage [cf. Knuth, 1968; Randell, 1969] also occur in the drum store.

Skinner [1967] and Coffman [1969] have analyzed mathematical models that accurately depict a single, SLTF, paging drum. An inspection of their results show it is possible to divide the response time of I/O requests into two terms: a term independent of the load of I/O requests placed upon the drum, denoted W_{LI}, and a term that monotonically increases with increasing load, denoted W_{LV}. The general form of the expected I/O waiting time, $\overline{W}$, a single paging drum is

$$\overline{W} = W_{LI} + W_{LV} , \tag{1.1}$$

the specific composition of W_{LI} and W_{LV} is discussed in detail later in this article. For an SLTF file (nonpaging) drum it has been shown that the expected I/O waiting time is [Fuller and Baskett, 1972]

$$\overline{W} > W_{LI} + 2W_{LV} , \tag{1.2}$$

where W_{LI} and W_{LV} are the same expressions shown in Eq. (1.1). A comparison of the above two equations gives an indication of the substantial gains in performance offered by a paging drum over a file drum. However, economic and hardware constraints often require that more than one drum must be attached to a single I/O channel, as shown in Fig. 1.2. When an I/O channel must manage several drums, much of the gain in response time achieved by using a paging drum organization is lost. The reason for this loss is that each drum on the I/O channel is driven by a separate, asynchronous motor. The records on one drum that are carefully- stored in non-overlapping sector positions now overlap by some arbitrary, and slowly varying, amount with all the other records on the remaining drums attached to the I/O channel.

In the analysis that follows we will see that the expected I/O waiting time of an I/O channel with k paging drums is approximately

$$\overline{W} = W_{LI} + (2 - \frac{1}{k})W_{LV} .$$

Note that the load varying term increases by $\frac{1}{2}$ with the addition of a second drum to the I/O channel and doubles in the limit as $k \to \infty$.

Should an individual decide, after seeing how much is lost with several drums on a single channel, that he would like to regain the lost performance there are several possible solutions. The most obvious one is to replace all the drums on the I/O channel by a single drum large enough to hold all the information once contained on the several smaller drums. This solution unfortunately runs counter to efforts to maintain modularity and reliability in the I/O structure of the computer system. A more attractive solution would be to provide sufficient hardware in the drum controllers so that all the paging drums on the same I/O channel can remain in synchronization and hence reply to I/O requests with the responsiveness of a single drum.

REFERENCES

Abate, J. and H. Dubner (1969) Optimizing the performance of a drum-like storage. IEEE Trans. on Computers C-18 (Nov. 1969), 992-996.

Coffman, E. G. (1969) Analysis of a drum input/output queue under scheduling operation in a paged computer system. J. ACM 16, 1 (Jan. 1969), 73-90.

Fuller, S. H. and F. Baskett (1972) An analysis of drum storage units. Technical Report no. 26, Digital Systems Laboratory, Stanford University, Stanford, California, (June 1972).

Knuth, D. E. (1968) The Art of Computer Programming, volume 1: Fundamental Algorithms. Addison-Wesley Co., Reading, Mass. (1968).

Randell, B. (1969) A note on storage fragmentation and program segmentation. C. ACM 12, 7 (July 1969), 365-369, 392.

Skinner, C. E. (1967) Priority queueing systems with server-walking time. Operations Research 15, 2 (1967), 278-285.