Performance of the IBM 3850 Mass Storage System (MSS) is analyzed with a view toward workload planning. Simple analytical models are discussed. The notion of staging capacity of the MSS is defined and analyzed. The main result is a set of staging capacity curves that define the processing ability of the MSS to stage and destage data to support concurrent execution of the user programs.

Capacity analysis of the Mass Storage System

by P. N. Misra

The IBM 3850 Mass Storage System (MSS) provides for economical and efficient storage of large volumes of data under system control. The data are stored on magnetic tapes in data cartridges and, when needed by the system, are transferred to direct access storage devices (DASDs) in a process called staging. The DASDs used to hold these data are called staging volumes and are a component of the MSS. When no longer needed, the new or changed data are transferred back (or destaged) to their data cartridges. The contents of the staging volumes thus change dynamically.

Once staged, the data are treated as any other data on a DASD to which the system has access and have a comparable access time. The execution of a user program is delayed, however, if access is required to data not yet staged. It may take 10 to 20 seconds to stage a cylinder of data, not counting any contention-related waits in the MSS. An individual user, especially if in interactive mode, will, therefore, be concerned about the MSS response time, which depends upon its current load from other users. From the point of view of an installation, there is a broader issue of the MSS usage: selection of data sets to be stored in the MSS, and workload planning *vis-à-vis* the processing ability of the MSS to stage/destage data to meet the requirements of concurrent users. The main objective of this paper is to analyze this processing ability. In particular, we define the notion of staging capacity of the MSS, and analyze it through a simple analytical model.

Central to the working of the MSS is the concept of a virtual device: The MSS presents to a processor the appearance of having more direct access devices than are actually present. The number of such virtual devices is limited only by the device addressing scheme—64 per channel interface—regardless of how many staging drives are attached. The system treats each pair of logically associated data cartridges as an IBM 3330-1 volume, referred to as a mass storage volume or virtual volume. The user programs request mounting/demounting of these volumes and access to their data as before. The latter requests are "translated" by the MSS in much the same way such translation occurs in the context of virtual storage. If already staged, the data are obtained from their actual location on the staging drives; if not already there, a cylinder fault is said to occur, and the data are staged and subsequently made available to the user program.

Insofar as the implementation of the notion of virtual DASD is concerned, almost any data set can reside in the MSS. The advisability of storing a data set in the MSS, however, depends upon factors such as organization of the data, access method used, data referencing pattern of the user programs, and the size of the data set relative to the staging space available.

Consider an environment with a number of interactive users and batch jobs, with their data sets in the MSS, executing their programs concurrently. These programs will require their data sets to be staged and destaged. This activity may be initiated in direct response to the user-issued command, e.g., mount/demount volumes, and open/close data sets. It could also be initiated by the MSS as a part of its housekeeping: by a cylinder fault in recognition of the fact that a user I/O operation cannot be completed because the required data do not exist on the staging drives, or by a least-recently-used (LRU) space allocation algorithm in recognition of the fact that the amount of allocatable space on the staging drives has fallen below that prescribed by the installation, and some of the active data sets must be destaged. The pattern of these staging/destaging activities characterizes the usage of the MSS at an installation. While the MSS executes its tasks in accordance with a priority structure, it does not recognize user priority from a host processor and services requests for staging basically on a first-come-first-served basis. As such, to guarantee one user a certain response is to guarantee all users the same response.

The need for definition and analysis of a notion of MSS capacity arose at the Shuttle Data Processing Complex (SDPC) of the NASA/Johnson Space Center as follows. The SDPC uses an MSS as an online storage device for data sets of users engaged in software development and other supporting functions for the Space Shuttle operations. It was proposed that the MSS be used during Space

Shuttle missions as the primary on-line storage device for telemetry data streams from the Orbiter and its payloads. This application basically entails two functions: retention of real-time data in mass storage volumes, and recall from the mass storage volumes of the previously retained data to be processed by investigators and flight controllers. Insofar as the software development and the other supporting work were to continue during the missions, these users were also to be allowed access to their data sets, perhaps in a restricted mode. Therefore, this question arose: Can the MSS meet the requirements of the real-time application? If so, how much more load can it accommodate from the other users, and how can this load be controlled to assure the users a certain performance?

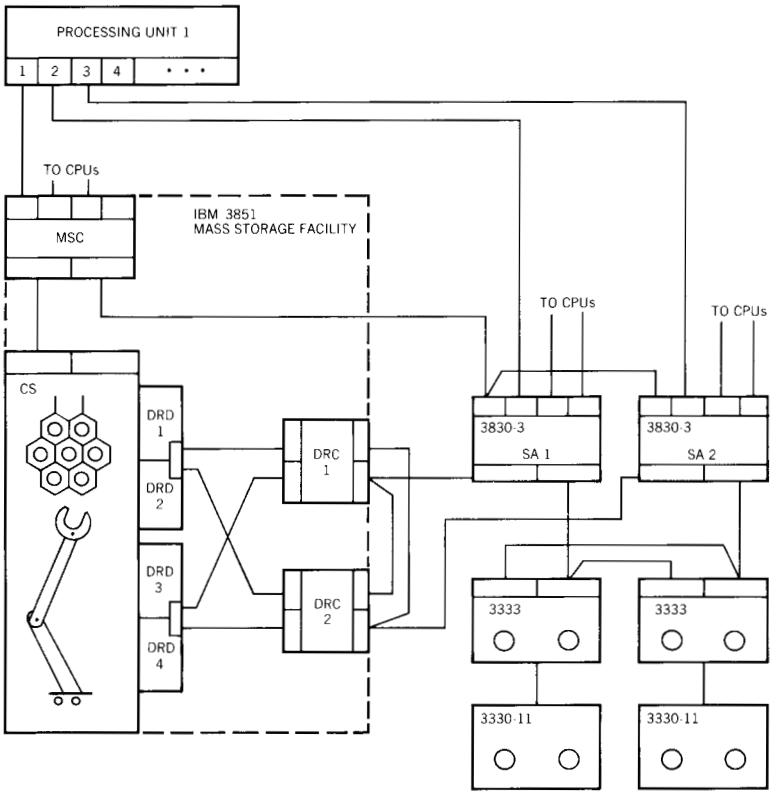
In this paper, we first briefly describe the main components of the MSS and their functions. This description is followed by a discussion of the MSS executing a request to stage data. Next we discuss development of a simple performance model and present the so-called staging capacity curves derived from it. Finally, the validation, uses, and generalizations of the model are discussed.

MSS configuration

The main components of the MSS are mass storage control (MSC), cartridge store (CS), data recording devices (DRD), data recording controls (DRC), staging adapters (SA), and staging drives. We describe their functions only briefly here, and in slightly greater detail in a later section. A comprehensive discussion is found in Reference 1.

The mass storage control provides an overall control of the MSS functions. Its microcode routines, structured like an operating system, build job lists to accomplish tasks requested by the host processors and dispatch commands to the cartridge store and the staging adapter, both controlled by microcode, to perform these tasks. The data cartridges are kept within the cartridge store, which also controls their movement by an accessor arm between their cells and the data recording devices. The latter hold and manipulate the cartridges, position the tape, and perform read/write operations on it. The data recording control regulates the transfer of data from or to the cartridge loaded in a data recording device. This includes formatting and error detection/correction. The staging adapter serves as a control unit for the data recording controls during the data staging/destaging process. A data recording control-staging adapter pair used to transmit data between a data recording device and a staging DASD is referred to as a staging path. The staging adapter also serves as a control unit for the staging DASDs for I/O operations to host processors. Additionally, it maintains several tables giving the status of the virtual volumes and space allocated to them on the staging drives.

Figure 1 An MSS configuration (Model B2)



In this paper, we will consider a B2 model of the MSS configured as shown in Figure 1. The basic notions discussed, however, carry over easily to the other models and configurations, which may differ in the number of staging adapters, data recording controls and data recording devices, size of the cartridge library, and number of staging drives. A B2 model contains four data recording devices and two data recording controls. Our MSS configuration consists of two staging adapters (IBM 3830-3 Storage Controls) and eight IBM 3333/3330-11 staging drives. The staging adapters are string-switched to the two 3330-11 Disk Storage facilities, each consisting of four drives. The control and data paths are set up redundantly as shown. Each of the data recording devices is connected to both data recording controls, each of which is connected to both staging adapters. Each data recording device can thus use either of the two staging paths. Similarly, the data on any staging drive can be accessed via either channel-control unit (staging adapter) pair.

In the following discussion, we occasionally use staging in a generic sense to include destaging.

Figure 2 Staging operation: Events and Trace measurements

Request to stage data received by the MSC
T1—Request processing completed by the MSC
Wait for a DRD to be available
T2—DRD available
Wait for accessor arm to move cartridge
Move cartridge
Load cartridge
Wait for staging path and label buffer availability
T3—Read and verify cartridge label
Wait for staging path for command to seek
T4—Begin seek
Unwind tape to the 'right' stripe
Wait for and share a staging path to stage data
T5—Data staging completed
Rewind tape
Unload cartridge

Staging process

We describe the staging process in the context of the MSS executing two host-issued commands: Mount Volume and Acquire-with-Stage. The discussion, though tedious, is essential to understanding model development. A summary is given in Figure 2.

When a user program running in a host processor requests a virtual volume, the OS/VS (Operating System/Virtual Storage) selects a virtual unit address and sends a Mount Volume command to the mass storage control. In response, the mass storage control microcode routines build job lists and control blocks required for the execution of the command, check and update several tables and directories containing information on the virtual volumes currently mounted and the space assigned to them on the staging drives, allocate space on a staging drive to stage cylinder 0 (VOLID—Volume Identification, and VTOC—Volume Table of Contents), and build a schedule queue block (SQB) containing the addresses of the data cartridge, data recording device, and staging adapter to be used in the execution of this stage request. At this point, the mass storage control sends the device end status to the host processor, and passes control to the stage scheduler which controls the staging activity as specified in the SQB. The stage scheduler sends a Move Cartridge command to the cartridge store. When the cartridge is loaded, the data recording device sends a Load Complete interrupt to the staging adapter. The staging adapter communicates the interrupt to the mass storage control, which then sends a Read command to the staging adapter to read the cartridge label. The staging adapter transfers the cartridge label to the mass storage control, which compares it to that in the SQB. Having verified the cartridge label,

the mass storage control sends a Move Data command to the staging adapter to stage cylinder 0 of the data cartridge in the space assigned on a staging drive. The device end status from the staging adapter to the mass storage control marks completion of the staging operation. The staging adapter commands the data recording device to rewind and unload the cartridge. The mass storage control updates the required tables and directories and commands the cartridge store to return the cartridge to its home cell. To the host, the operation is tantamount to having a disk pack mounted on the assigned drive.

An Acquire-with-Stage order may be issued by the host processor when a user opens an existing data set on a virtual volume. We discuss the MSS response to illustrate the mechanics of data transfer associated with a multicylinder staging and to bring out salient features of the staging adapter. The execution begins much the same way as before. We pick up the trail where the cartridge has been loaded in a data recording device and its label has been verified. The mass storage control sends a Move Data command to the staging adapter for the first page (eight cylinders) of data. The staging adapter determines the staging drive and its cylinder number for a DASD seek and the cartridge stripe number for a data recording device seek, issues a Position Cartridge command to the data recording device, and disconnects. Upon completion of the seek, the data recording device attempts to reconnect to the staging adapter. The staging adapter interleaves control commands to the data recording devices (such as seek, rewind, and unload) with data transfer associated with staging. In this fashion, up to four data recording devices can operate simultaneously with a staging adapter, but only one of these can be staging data at a time. This feature, known as data recording device overlap, was introduced in Release 7 of the MSS microcode. In the previous releases, the staging adapter was actually "busy" while the seek, rewind, and unload were completed by a data recording device.

The staging adapter breaks up a Move Data command into one-cylinder units. At the completion of each cylinder transfer, it initiates cartridge advance in this data recording device to the beginning of the next cylinder, clears pending interrupts from the other data recording devices, and issues non-Move Data commands. It then continues with the current staging if the cartridge advance is complete. If not, or if the current data recording device has completed staging an eight-cylinder page unit, the staging adapter switches to another data recording device with a pending Move Data command. The data recording devices ready to stage data thus time-share the staging paths.

The data are organized on the tape in diagonal stripes in DASD format images, 67 stripes to a 3330-cylinder image. The data

recording device incrementally traverses these stripes, and transmits data via a data recording control to a 32K-byte speed-matching buffer, called a data buffer store (DBS), in the staging adapter. The automatic data transfer (ADT) circuits transmit data between the data recording control and DBS; the staging adapter microcode transmits data between the DBS and DASDs. This scheme allows the staging adapter to transfer data between the data recording control and DBS while simultaneously executing other microcode routines, e.g., transferring data between DBS and a DASD, or between the host and a DASD.

During a staging operation, ADT circuits read eight stripes of data into the DBS. When three stripes have been read, the DASD selection routine selects a staging drive and sends a seek command. At the end of the seek, the staging adapter transfers data between the DBS and DASD and is "busy" to all interfaces during this operation. Up to four tracks are written during each DASD access, at the end of which the staging adapter deselects the drive and allows host processors access to the data on the staging drives. The ADT circuits continue to transfer data between the data recording control and DBS, and the process is repeated until all cylinders specified in the Move Data command have been staged.

The I/O request from a user program to an open data set is handled as follows. The request goes through the access method and the I/O supervisor as before. When the channel program reaches the staging adapter, this device translates the virtual unit address and the cylinder number. The data being staged in increments of cylinders, the track number, and record number need no translation. The I/O operation is now completed in the usual manner. A request for data not yet staged results in a cylinder fault and is delayed until the data can be staged, giving the host processor the appearance of an unusually long seek.

Figure 2 gives a summary of events associated with a cartridge move and data staging. T1 through T5, the so-called T times, are recorded for each transaction by the MSS Trace, a monitor in the mass storage control. These measurements provide a basis for analysis of the workload and the MSS response at an installation. We have used the Trace data to estimate service times associated with the various steps in staging needed by the analytical model, and for model validation.

Performance modeling

The nature and extent of staging and destaging activities at an installation depend entirely upon the interaction of the user programs and their data sets, size of the staging space, and the

installation-specified thresholds on the staging space utilization. As such, a breakdown of staging activity by its sources (e.g., Mount Volumes, Acquires, cylinder faults, LRU destaging) is important in understanding the usage of the MSS. Insofar as our interest is in determining the throughput achievable, we model workload as consisting of a stream of requests for staging, regardless of how they came about or whether they represent good usage of the MSS. We characterize such a job stream by the average number of cylinders to be staged per request and assume that there are always requests waiting to be processed. The rate at which each such job stream is processed defines the throughput achievable, or capacity, of MSS. In this section, we develop a simple analytical model with the objective of estimating this staging capacity.

A request for staging occupies a data recording device during the entire length of the operation, from allocation until the tape is rewound and the cartridge unloaded. Having been allocated a data recording device, the job waits for and receives service from the accessor arm for cartridge moves. Beyond that, it waits for availability of the staging path and a label buffer in the mass storage control for label verification and, again, for the staging path to initiate seek. Once ready to stage data, the data recording device starts sharing a staging path with the other data recording devices in the same step. A data recording control sees a sustained load only during actual data transfer, one cylinder at a time, between a data recording device and the DBS. The staging adapter is busy on staging only intermittently during the process, interleaving I/O operations to host processors with emptying out the contents of the DBS onto a staging drive. We assume the transmittal of commands and the cartridge label read/update to be instantaneous. But how long may a data recording device have to wait to initiate these steps? The answer: at most, the time it takes to stage a cylinder of data.

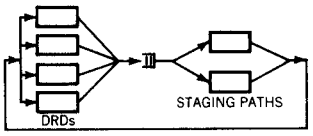
For the purpose of analyzing the rate at which a waiting job stream is processed, we can regroup the steps as shown in Figure 2 and think of a staging operation as consisting of the following operations. The "busy" components during the execution of each step are shown below in parentheses.

1. Rewind tape, update label, and unload cartridge used in the previous request (data recording device)
2. Restore the cartridge to its cell, move the cartridge needed by the current job to the data recording device (data recording device, accessor arm)
3. Load cartridge in data recording device, verify label, and seek on tape (data recording device)
4. Stage data (data recording device, data recording control, staging adapter)

From the users' viewpoint, the above grouping of steps makes sense in that a request is completed when the required data have been staged up and are accessible to the host processor. The real reason for this regrouping will become clear later. The first three steps, taken together, will be referred to subsequently as the cartridge preparation step.

It is interesting to observe the similarities in the working of the MSS and those of a multiprogrammed computer system. In the latter, a job waits until it can enter main storage and, once there, starts sharing the CPU and I/O devices with the other jobs in the system. In the MSS, a request for staging waits until it acquires a data recording device, and beyond that, starts sharing the other system resources: accessor arm, data recording control, staging adapter, and staging drives. There is an important difference, though. A job, while in main storage, is either waiting for or receiving service from the CPU and I/O devices. A data recording device, however, actually services a request by itself in several of the steps in a staging operation: load, seek, rewind, and unload. During the remaining steps, viz., cartridge moves and data transfer, the data recording device remains occupied while the job waits for and receives service from the accessor arm or a staging path. A closer analogy, then, is with a time-sharing system with data recording devices thought of as user terminals and staging paths as CPUs. The cartridge preparation time, defined as the time taken to complete the cartridge preparation step (namely, cartridge moves by accessor arm, load, seek, rewind, and unload), is analogous to the user think time, and the data staging time to the system service time.

Figure 3 A finite source queuing model of the MSS



We dispense with the notion of a waiting job stream. Instead, each request for staging is thought of as being generated at a data recording device during cartridge preparation time and as being executed by a staging path while the data recording device stays blocked as in a TSO (Time Sharing Option) transaction. The cycle is then repeated. This procedure gives us the familiar machine repairman model (or finite source queuing model) with processor-sharing service discipline, as in Scherr's analysis of time-sharing systems.[2] We have four machines (data recording devices) to be maintained by two repairmen (staging paths). The information on the staging job stream is translated to the model in terms of the average up time (cartridge preparation time) and the average service time (data staging time). We will not present an analysis of this model here. An interested reader is referred to Kobayashi.[3] The model, shown schematically in Figure 3, effectively gives the rate at which the MSS can process a staging job stream.

The transaction-to-transaction variability in cartridge preparation times comes from the length of the tape to be rewound, the length of accessor arm movement, and the seek distance (location of

data on the tape). It also depends upon waiting times for the accessor arm to move the cartridges, for the staging path and label buffer to verify the cartridge label, and for the staging path to initiate seek. The first two depend upon cartridge traffic, and the last one on data traffic (staging and host I/O activity). The staging time is determined by the number of cylinders to be transferred and by the data rate of the staging path. Staging and host I/O activities contend for the staging adapters, which control data transfer associated with both, and for the staging drives. The data staging rate thus depends upon the extent of host I/O activity. We resort to experimental estimation of this dependence for a case of interest to us.

Estimation of service times

The steps 1 to 4 of staging, as given previously, represent a cleaner breakdown of the staging operation than that implied by the MSS Trace measurements (Figure 2). Direct estimation of step-by-step service times in this breakdown on the basis of the Trace data, however, is not possible. For example, in periods during which the MSS has requests waiting to be served, the amount of time (T3 − T2) for a request comprises a wait for the accessor arm, two cartridge moves, a cartridge load, and verification (Step 2 and part of 3 of our model). Similarly, (T5 − T4) accounts for seek, wait for staging path, and data transfer (remaining part of Step 3 and Step 4). The time elapsed between T5 for one job and T2 for the next one allocated to the same data recording device accounts for rewind, label update, and unload (Step 1).

Figure 4 Average time taken to move, load, and verify a cartridge

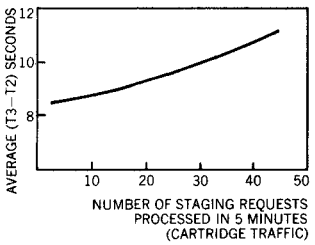


Figure 4 gives the average values of (T3 − T2) plotted against cartridge traffic rates. The Trace measurements used correspond to "busy" periods of the MSS during which there were always requests waiting to be served. Given the cartridges at a facility, their arrangement in storage cells, and relative frequencies of usage, an average value can be associated with the cartridge move time. Insofar as cartridge load time is a constant, the increase in average (T3 − T2) is attributable entirely to waits for the accessor arm and the label buffer. Figure 4 suggests that for the observed cartridge traffic rates the utilization of the accessor arm is not high.

Figure 5 Average seek and data staging times for one-cylinder stagings

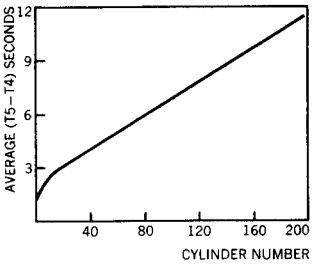


Figure 5 shows average values of (T5 − T4) for various one-cylinder stagings from different locations along the length of the tape. These measurements correspond to "easy" periods of the MSS with no contention for the staging paths. Given the data staging rate, we can estimate the seek/rewind times. If we assume that the locations of user data sets that are to be staged are uniformly distributed along the length of the tape, and disregard

Figure 6 Average data staging rate per staging path at various levels of host I/O (Block size: 10 800 bytes)

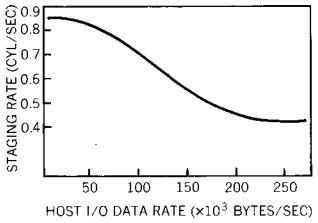


Figure 7 Staging capacity of the MSS: Throughput achievable in five minutes under "light" host I/O as predicted by the analytical model

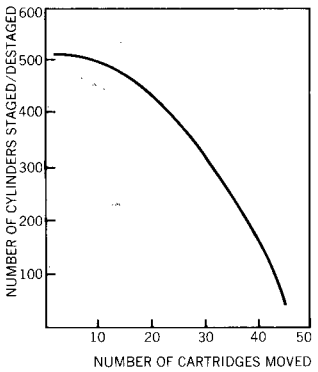
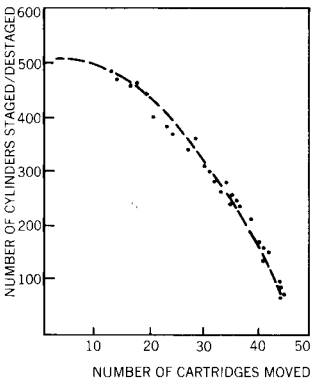


Figure 8 Staging capacity of the MSS (model validation): Highest throughputs observed in five-minute intervals compared with those predicted by the analytical model



the nonlinearity at the left end of the curve in Figure 5, the average seek/rewind time is approximately 6.7 seconds minus the time taken to stage a cylinder of data. Actually, the first cylinder (VTOC) is staged more frequently than the last one on a cartridge, introducing asymmetry in the distribution of seek/rewind times.

The average value of (T4 − T3), the wait for a staging path for command to seek, is invariably small for the range of staging and host I/O activity of interest to us and is disregarded. Figure 6 is a plot of average data staging rates for various host I/O rates for an operational environment of interest to us: retention in mass storage volumes of real-time telemetry streams with predefined block size (10 800 bytes) and EXCP rates. The staging data rates have been estimated from the Trace measurements gathered in experiments in which staging requests were executed under various sustained I/O rates. The staging data rate is seen to be 0.85 cylinders per second under "light" host I/O activity, and drops off to roughly half that as contention for the staging adapters and DASD arms increases under heavy I/O activity.

The average cartridge preparation time can now be estimated from the earlier plots of the Trace data. At our installation it ranges between 23.5 and 26 seconds, depending upon the cartridge traffic rate.

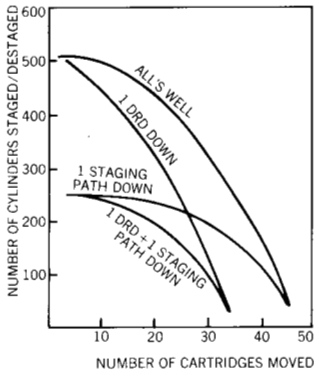
We are now ready to analyze the throughput achievable on the MSS for different job streams on the basis of our model. We assume the number of cylinders to be staged for requests in a job stream, and, hence, their service time, to be exponentially distributed. We make a similar assumption regarding the cartridge preparation time. In reality, neither assumption is expected to hold. For one thing, the number of cylinders to be transferred is a whole number. Also, unless there is a special design to the arrangement of cartridges in storage cells and of data sets on the tapes, the cartridge preparation time is more likely to be uniformly distributed. We proceed with the analysis based on our assumptions, however, anticipating the usual robustness of results on averages to provide us with a useful approximation to the true behavior of the MSS.

MSS capacity

The machine repairman model gives, among other things, the rates at which the different job streams can be processed and the corresponding utilization of the staging paths. Insofar as the cartridge preparation time depends upon the cartridge traffic (i.e., transaction processing rate), the analysis is done iteratively. The results on the throughput, plotted in Figure 7, correspond to

"light" host I/O activity (staging rate: 0.85 cylinders per second). The curve gives a trade-off between cartridge moves and cylinder stagings, and the limit of each, that the MSS can deliver. For example, the MSS can execute on the average up to about 44.5 one-cylinder stagings in five minutes. Alternately, it can deliver up to 31 cartridge moves with an average of 10 cylinders staged per cartridge move.

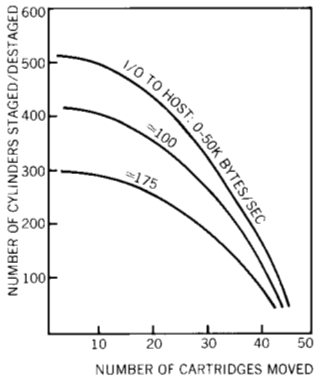
Figure 9 Staging capacity of the MSS (model predictions): Throughput achievable in five minutes in degraded modes under "light" host I/O



The choice of five minutes is somewhat arbitrary, one reason being to avoid fractions. The main reason for the choice, however, is our interest in verification of these results. We do this by analyzing the available MSS Trace data for throughputs (cartridges moved and cylinders staged) observed in various time intervals, and comparing the highest throughputs observed with those predicted by the model. Obviously, the choice of time interval depends upon the length of time over which saturation of MSS can be sustained. The highest throughputs observed in five-minute intervals at our facility are shown in Figure 8. When the simplicity of the model is considered, the agreement between these and the model predictions is remarkable.

Hardware duplication and redundancy in control and data paths usually permit the MSS to function, albeit in degraded mode, when components fail. Insofar as our proposed real-time application is to use the MSS over a period of several days at a time, it is important to understand the nature and extent of this degradation. The effect of losses of the data recording devices and/or a staging path is easily analyzed by our model by making the appropriate changes in the number of machines and/or repairmen. The corresponding losses in capacity of the MSS to process requests for staging are shown in Figure 9. Interference in staging due to host I/O activity is similarly analyzed: Determine data staging rate corresponding to the level of host I/O activity of interest, and determine the data staging time for the job stream. Figure 10 shows the staging capacity curves for three levels of host I/O activity.

Figure 10 Staging capacity of the MSS (model predictions): Highest throughputs achievable in a five-minute interval at various host I/O data rates (Block size: 10 800 bytes)



The proposed model also provides a basis for assessing the effect of changes in design and usage of the MSS. A case in point in regard to the former is the analysis of gain in staging capacity realized from the feature of data recording device overlap in Release 7 of the MSS microcode, referred to earlier. In regard to the latter, consider the gain to be realized from arranging data sets on cartridges so that the more frequently used data sets will be closer to the beginning of the tape. The savings are in the seek and rewind times and, hence, in what we have called the cartridge preparation time. Whether the effort is worthwhile is easily analyzed.

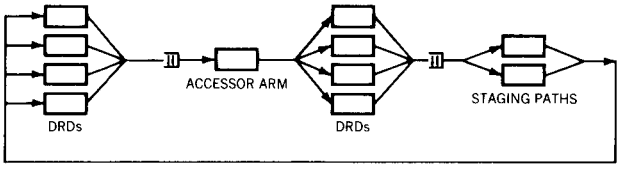
Workload planning

The principal motivation for our analysis came from the need for workload planning. Given estimates for host I/O rates and staging/destaging activity, the first concern is with the ability of the MSS to keep up with the load. The issue is easily settled by comparing the workload with the staging capacity to ensure that the backlog does not continually increase. Given that the MSS can deliver the required throughput, the next question is: How well? The question is particularly important in an interactive environment. In our model, we dispensed with the arrival mechanism of the staging requests and, hence, the response time issue. There is no basic difficulty. The situation can be analyzed through an open network model (say, a mixed central-server model).[3] We outline a simpler approach below.

Suppose we are given the average interarrival time of requests and the average number of cylinders to be staged per request. We can model the MSS as an open four-server queuing system, the servers being the data recording devices. There is a slight twist, though: We do not have *a priori* knowledge of data recording device occupancy time per request. The occupancy time is made up of the cartridge preparation time, which we assume is known, and the data staging time, which depends upon the number of cylinders to be staged and the degree of contention for the staging paths. Recall that a staging path is time-shared by data recording devices that are concurrently in the data staging step. In our configuration, up to two of these devices may be sharing a path to stage data (up to four if a staging path were off line). Consider the case where only very few cylinders are to be staged per request. The staging path utilization will clearly be low, and we can assume that there is no contention for (and sharing of) the staging paths. We can determine the data recording device occupancy time directly and analyze the queuing system modeled as, say, M/M/4 to estimate the required performance indices.[3] But as the data traffic on the staging paths increases, the staging times stretch because of time sharing of these paths.

In a general case, we solve the problem by defining a stretch factor α such that the staging time is α times that in an environment with no contention for the staging paths. We assume a value for α, compute data recording device occupancy time, and analyze the queuing system to determine the probabilities that 0, 1, 2, 3, or 4 of the data recording devices are occupied. From these we can compute the probabilities that 1, 2, 3, or 4 such devices are staging data concurrently. Knowing the stretch factor for each case (namely, 1, 1.5, 1.67, and 2, respectively), we can compute the average stretch factor for this job stream. If the latter value is sufficiently close to that assumed at the beginning, the problem is solved. If not, iterate the procedure.

Figure 11 A more general model of the MSS



We have used these models in capacity and response time analysis for workload planning. Briefly stated, we found that the MSS will meet the requirements of the proposed real-time application. Actually, it was concluded that with certain mild restrictions in the form of operational procedures, the data sets of some of the current and planned applications can coexist in the MSS with the real-time data.

Model generalization

The model of the MSS discussed above is perhaps the simplest we could have built to obtain meaningful results. We outline here a generalization to account explicitly for the accessor arm as a server. We managed to avoid this earlier on the basis of empirical measurements that effectively gave the accessor arm responses at various cartridge traffic rates of interest. If these measurements had not been available, or if we were analyzing a larger MSS unit (more data recording devices) capable of higher cartridge traffic rates than those for which we had measurements, we would have had to include the accessor arm as another service station in the model to analyze waiting times for cartridge movement. Actually, the inclusion of an additional service station in the finite source queuing model discussed earlier does not pose any special difficulty. The resulting cyclic queuing system is shown in Figure 11. Four jobs circulate through the four service stations. A transaction begins at a data recording device with the first cartridge preparation step consisting of tape rewind and cartridge unload. It is followed by service at the accessor arm for cartridge moves. Next comes cartridge preparation step 2, consisting of load and seek at the data recording device and, finally, service at a staging path. For analysis of this simple closed queuing network, the reader is referred to Kobayashi.[3]

As noted earlier, the proposed model is easily adapted to analyze another MSS configuration. The changes required for differences in the number of data recording devices and staging paths are straightforward. The cartridge preparation time and staging data

rate, both determined empirically here, are installation-dependent and may differ slightly depending upon the average length of the accessor arm movement, data block size, host I/O data rate, and the configuration of staging drives.

Concluding remarks

We have examined the process of staging/destaging of data with a view toward establishing certain performance characteristics of the MSS. The main issue we address is: What is the maximum throughput achievable on the MSS? The analysis is based on a simple queuing model, the validity of which has been established by comparing its predictions with the measurements gathered in our operational environment. The results have been found of value in monitoring and tuning MSS usage, in evaluating alternative design changes, and in workload planning.

ACKNOWLEDGMENTS

The author is grateful to J. Garner of the IBM Washington Systems Center, M. Coome of IBM Canada, S. Wheeler of IBM Austin, and G. Dodson of IBM Tucson for their review of an earlier version of this paper. He is also indebted to his Houston colleagues, A. Aldrich, D. Brewer, K. Cowden, F. Hertel, H. Hulen, B. Hunter, J. Nylund, E. Pape, and J. Poje for their support.

This work was supported by NASA/Johnson Space Center under contract NAS 9-14350.

CITED REFERENCES

1. *IBM 3850 Mass Storage System (MSS) Principles of Operation: Theory*, GA32-0035-0, IBM Corporation (1978); available through IBM branch offices.
2. A. L. Scherr, *An Analysis of Time-Shared Computer Systems*, MIT Press, Cambridge, MA (1967).
3. H. Kobayashi, *Modeling and Analysis: An Introduction to System Performance Evaluation Methodology*, Addison-Wesley Publishing Co., Reading, MA (1978).

The author is with the IBM Federal Systems Division, 1322 Space Park Drive, Houston, TX 77058.

Reprint Order No. G321-5153.