1981 Winter Simulation Conference Proceedings
T.I. Ören, C.M. Delfosse, C.M. Shub (Eds.)

PERFORMANCE ASPECTS OF DISK SPACE MANAGEMENT

by James O. Dyal and Michael K. Draughn
Sperry Univac Division of Sperry Corporation
PO Box 500
Blue Bell, PA 19424

ABSTRACT

Dynamic Disk Space Management is a set of strategies for allocating files to disk storage devices so as to reduce seek time, device contention, and storage fragmentation.

In this paper, the impact on system throughput and user response time is projected, via simulation model results, for a small general-purpose operating system which manages large capacity disk drives.

1. INTRODUCTION

Over the next few years, as small removeable disk packs are replaced by large capacity units, with more applications executing concurrently on-line, the penalties associated with inadequate disk space management rules become highly visible in degraded response time at the users' workstations, and also in low central site system throughput.

The user population for the OS/3 operating system (Univac 1980) numbers several thousand, and includes many mature batch-oriented sites which are hard-pressed to expand coverage to transaction-oriented applications within system budget constraints. Great importance attaches to ease-of-use, and the ability to provide compatible service upgrades with minimal prime-shift impact.

Accordingly, this paper is a status report on continuing studies of dynamic space management strategies which are candidates for early implementation in a general-purpose operating system oriented toward smaller users. The study procedure, for system configurations;

- Define an appropriate Benchmark Transaction
- Monitor performance with existing system
- Calibrate simulation model via monitored results
- Use model to project anticipated future benefits
- User-oriented cost-benefit analysis determines implementation priority

Topics currently being explored include disk space allocation rules which minimize fragmentation, initial allocations which provide better load balancing over a set of drives to reduce device contention, and automatic invocation of relocation utilities via adaptive control algorithms.
In this paper, the focus is on performance enhancements which can be achieved by reducing disk seek time. System-level performance gains are projected for two allocation strategies:

"Global reorganization" involves complete re-allocation of disk file storage to minimize seek time and device contention. For the benchmark at hand, system throughput can be increased 34%.

"Reference spike relocation" is used to fine-tune the storage system as reference patterns change over time. Under the test conditions, system throughput is improved 16% by re-locating the most active file partitions.

2. GLOBAL REORGANIZATION

2.1 Original Allocation

Table 1 (see Appendix) summarizes the files to be allocated and typical relative reference frequencies. In all there are 106 File Allocation Units (partitions or separate extents) which add up to 7121 disk tracks (102 megabytes, 509 cylinders).

Figure 1 is a graph of the cumulative file reference pattern for the original allocation. The file allocation units were copied from three smaller disk packs to one large pack. The partition extents were adjusted for the new track capacity and laid out in the same sequence as on the source packs. As can be inferred from Figure 1, each of the original packs had rather well-centered locality of reference i.e., they reduced seek time on the smaller device. However, when the files are copied onto the larger device, the three local reference peaks are spaced out, and ping-ponging will occur between the non-adjacent active tracks. The model computes an average seek time of 29 msec for this allocation, which is little better than the nominal hardware value (35 msec) for a rectangular distribution.

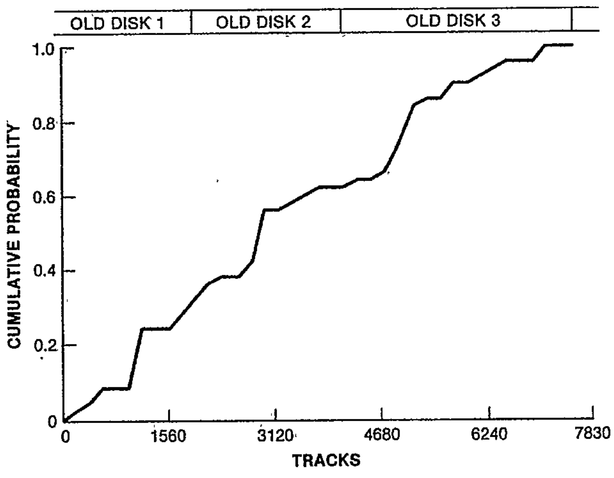


Figure 1: **Cumulative Track Reference Distribution Original Allocation. Seek Time: 29 msec.**

2.2 Re-Allocation Algorithm

The files are re-allocated, using a simplified form of Arora-Gallo optimization (Arora 1973). This proceeds in three steps:

1. Compute as sort key for each allocation entity, the ratio of reference probability to extent. The sort key is references/byte.
2. Sort the file allocation entities.
3. Place the allocation entities with highest keys in the most accessible memory. For disk allocation, this normally means working symmetrically from center cylinders outward to the two ends of the device. In the center will be found short files referenced very often. At the ends will be found large files infrequently referenced.

The new file reference pattern is given in Figure 2 as a cumulative probability distribution, and shows an average seek time of 14 msec. This represents a 15 msec improvement relative to the original allocation - almost a 50% reduction in average seek time.

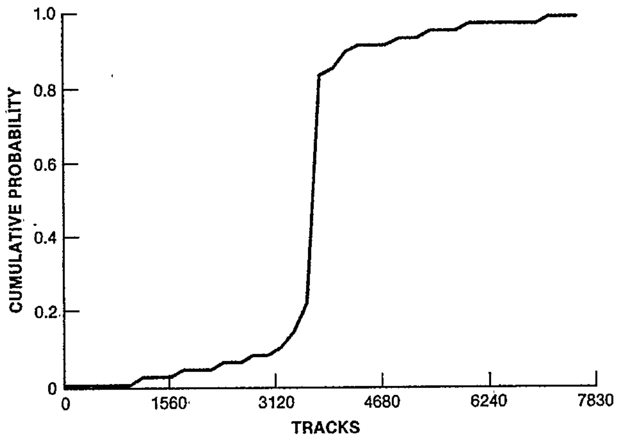


Figure 2: **Cumulative Track Reference Distribution after Global Reorganization. Seek Time: 14 msec.**

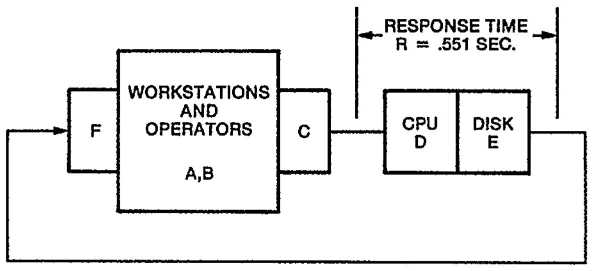
2.3 Modeling Considerations

Over and beyond those affects attributable to file allocation, the system can be throughput limited by many factors, such as processor speed, channel bandwidth, main memory size, etc. (Amdahl 1970). In this paper, we consider a medium-speed unit processor (500 KOPS), running a multi workstation triple-update benchmark, with enough main memory so there is no internal limit on the number of active threads, and one or two disk drives. Under these conditions we ask, how much can system throughput and response time be improved by better file allocation?

Figure 3 is a diagram of the system as modeled, together with single-thread performance parameters for the original file allocation in the single-disk configuration. The disk has nominal seek time of 35 msec and turns at 17.6 msec/revolution. Data transfers average 1.5 per seek, and there are 8.74 seeks per transaction. In the model, CPU time and disk service time for each transaction are pro-rated to the number of seeks issued, with mean and standard deviation adjusted to agree with monitored values.

From the ratio of disk service time to CPU pathtime (2.5/1), it is seen that this application is heavily I/O bound prior to file relocation. From the even larger ratio of operator time to response time (18/1), it is expected that many workstations will be required to load the system effectively.

The system executes in a closed-queue mode. After each workstation message is sent to the central site, the operator is idle until the answer has been displayed. As the number of workstations is increased, the number of active threads in the central site increases, and more I/O will be overlapped with CPU activity.



	Event	Time/Transaction Seconds
A	Operator Thinks	9.000
B	Operator Types	1.000
C	Message Transmit	.040
D	CPU Pathtime	.157
E	Disk Service Time	.394
F	Message Display	.127
	Total/Transaction	10.718
	Transactions/Minute (Single Thread)	5.6

Figure 3: **System Block Diagram**

2.4 System Performance

Figure 4 shows model predictions of system throughput versus response time for a single disk configuration supporting 1 to 60 workstations. The two curves in Figure 4 correspond to the two file reference patterns shown earlier (Figures 1 and 2).

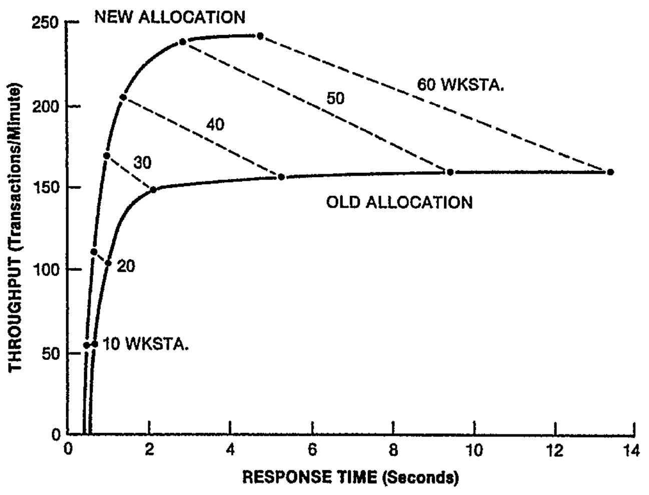


Figure 4: **Global Reorganization - One Disk**

The curve of system throughput versus response time has a fairly sharp bend, or "knee". Below the knee, the system is only lightly loaded, and a substantial increase in throughput can be achieved with only a small increase in response time. Above the knee, the converse is true; some critical component is approaching saturation loading, so only small increases in throughput are possible, and at the expense of a very large increase in response time.

For the original allocation, the knee occurs at about 150 transactions per minute, when disk usage approaches 100%. By reducing seek time, the performance knee is moved upward to 250 transactions per minute and twice as many workstations can be supported with only a modest increase in response time.

In Figure 4, instruction pathlength per transaction is independent of threading level. This is typical of table-driven dispatchers over some range of concurrency. However, it should be appreciated that this analysis is more oriented toward pair-wise comparison of two allocations at a given threading level, than absolute performance predictions over the full range of configurations presented.

2.5 Load Balancing Across Devices

In addition to reducing seek time, allocation strategies can be developed to minimize device contention by spreading the file allocations across available drives. Both effects are combined in Figure 5, which shows system throughput versus response time for a configuration with two disk drives.

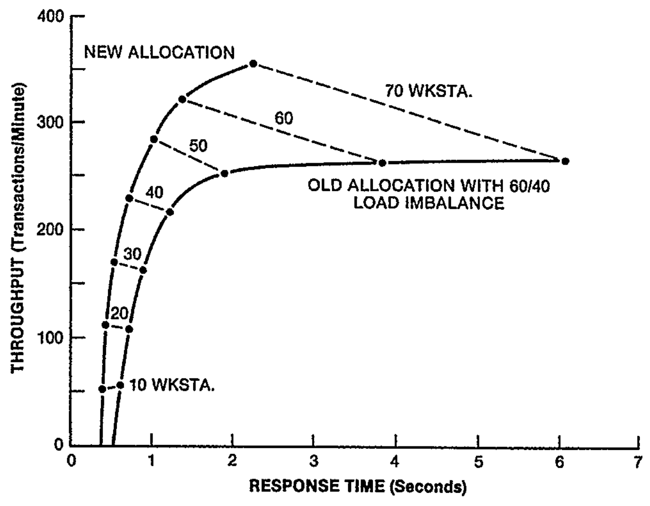


Figure 5: **Global Reorganization and Load Balancing - Two Disks**

Instead of preserving the original file sizes in Table 1, each allocation unit is made twice as big. The new total of 204 megabytes is then spread over the two disks by splitting each file into two equal partition extents; the first half of each file goes on device #1, and the second half goes on device #2, in the same vertical sequence by track number as in Figure 1. An overall reference imbalance of 60/40 between the two drives is assumed, and this, together with average seek time of 29 msec for each device, is input to the throughput model to generate the "old" runs in Figure 5.

For the "new" allocation in Figure 5, split partitions are first swapped sideways across drives as required to equalize overall reference probabilities between packs, then each disk is re-organized vertically to minimize seek time.

As shown in Figure 5, performance with two disks is much better under the combination of load balancing and seektime minimization than for the original allocation. Pairwise comparison for 70 workstations indicates throughput can be improved 34% (from 266 to 357 transactions per minute) while average response time at the workstation is reduced 63% (from 6.08 to 2.23 seconds per transaction).

Data plotted in Figures 4 and 5 is also listed in Tables 2 and 3.

3. REFERENCE SPIKE RELOCATION

Even though files may have been optimally located originally, performance can be expected to decline over time, as new files are opened, and old files are extended, unless some sort of corrective action is taken. A good initial allocation could be maintained by periodic global reorganization. Alternatively, at selected intervals, fine-tuning can be done by relocating currently popular allocation entities. Performance improvement under field conditions will depend on how sensitively a misplaced reference spike can be detected, the expected system benefits from relocating, and the cost of interchanging a more popular allocation unit with one less popular.

Figures 6 and 7 show a typical reference spike before and after relocation. The disk reference patterns are shown as cumulative probability of reference vertical and disk track numbers horizontal.

In Figure 6 there are two regions of above-average reference intensity. One is properly located in the center tracks. The other is at the left-hand end (the first 800 tracks, about 12 million bytes), perhaps because a newly allocated file has turned out to be unexpectedly popular. In Figure 7 the reference spike has been relocated just right of center, by swapping with a relatively less-popular file partition. Seek time is reduced by 5 msec (from 31 msec to 26 msec) by relocating 800*2 tracks.

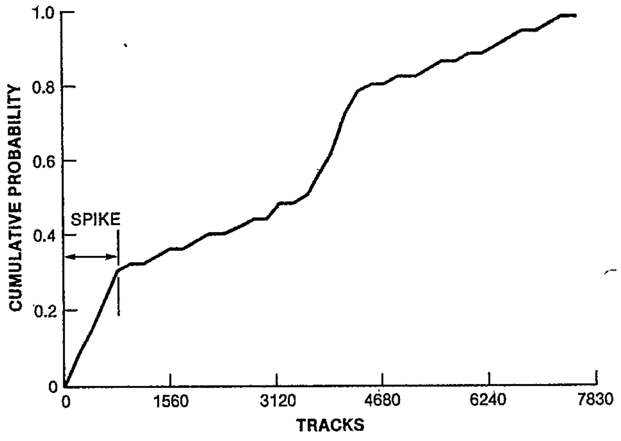


Figure 6: **Cumulative Track Reference Distribution Reference Spike before Relocation. Seek Time: 31 msec**

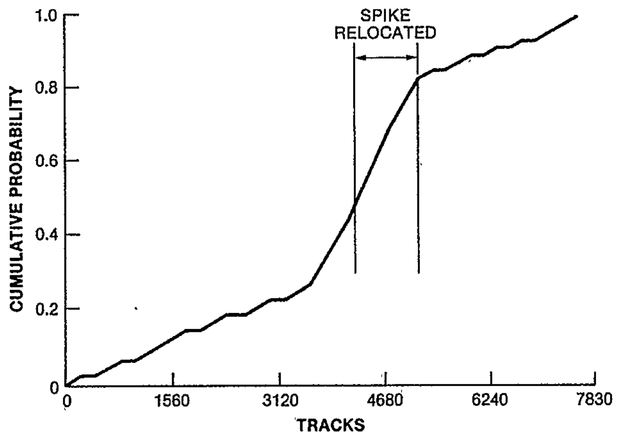


Figure 7: **Cumulative Track Reference Distribution Reference Spike after Allocation. Seek Time: 26 msec**

Figures 8 and 9 show system throughput versus response time, before and after reference spike relocation. Figure 8 is a single disk system. As can be seen, even the relatively small change of 5 msec in average seek time has a noticeable impact on system throughput for the larger configurations.

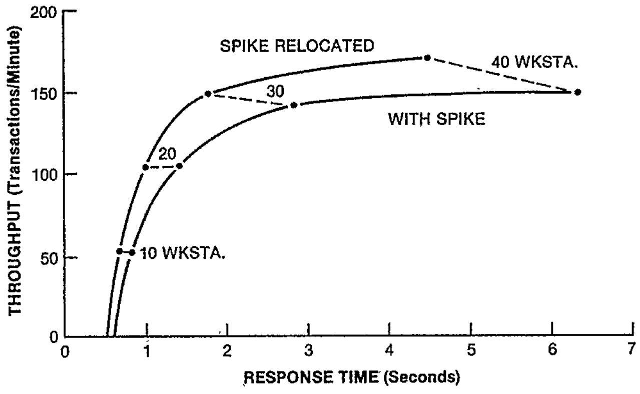


Figure 8: **Reference Spike Relocation - One Disk**

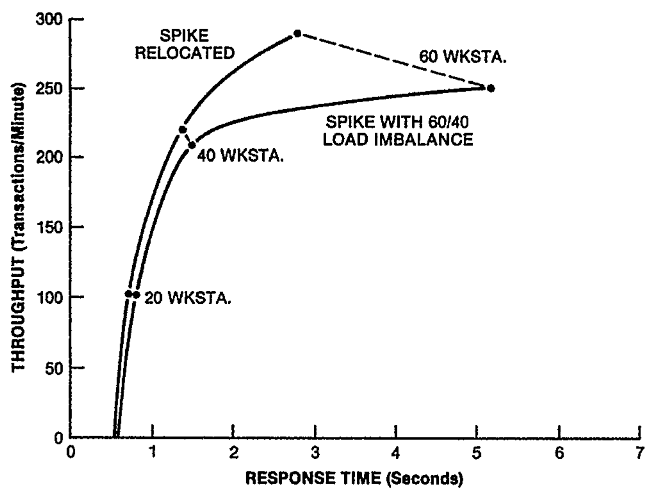


Figure 9: **Reference Spike Relocation and Load Balancing - Two Disks**

Figure 9 gives a similar comparison, only this time with two disks. In the model, the reference spikes (one on each device) are also relocated across devices to improve load balancing between drives. This leads to even higher throughput gains than was the case for a single disk. However, it should be noted that relocation to reduce seek time on each device is still advantageous in multi-disk systems, even without load balancing.

Tables 4 and 5 list the data plotted in Figures 8 and 9.

With a double-track main memory buffer (30 kilobytes), relocating the 800-track reference spike can be accomplished in only 2.4 minutes. For the 10 workstations, single disk configuration the system will henceforth be 6% faster than before relocation was invoked (149 versus 141 transactions/minute). After 45 minutes, the improved system throughput will compensate for the time spent in relocation.

It should be noted that detecting a reference spike (via state-of-art monitor techniques (Univac 1981)) presents no particular problem, in that a spike signal so small that it is difficult to detect, is probably not worth relocating.

The door is open, at least in principle, for a nearly-continuous adaptive control mechanism which automatically invokes a background batch file merge utility as indicated. A more conservative initial approach, would be to focus on activity during peak load conditions, say Tuesday from 9 to 11 A.M. Then based on last week's monitor analysis, start early on next Tuesday to prepare the disk subsystem for maximum peak load performance.

4. SYSTEM TRANSFER FUNCTIONS

Throughput vs Response Time curves, such as Figure 5, also include system load level as a parameter along each curve; lines of equal input loading, represented by the dashed lines in the figure, constitute a family of curves which cross the throughput curves at various angles. To separate effects and to explore various design trade-offs, it is convenient to construct three-axis System Transfer Functions, as shown in Figure 10.

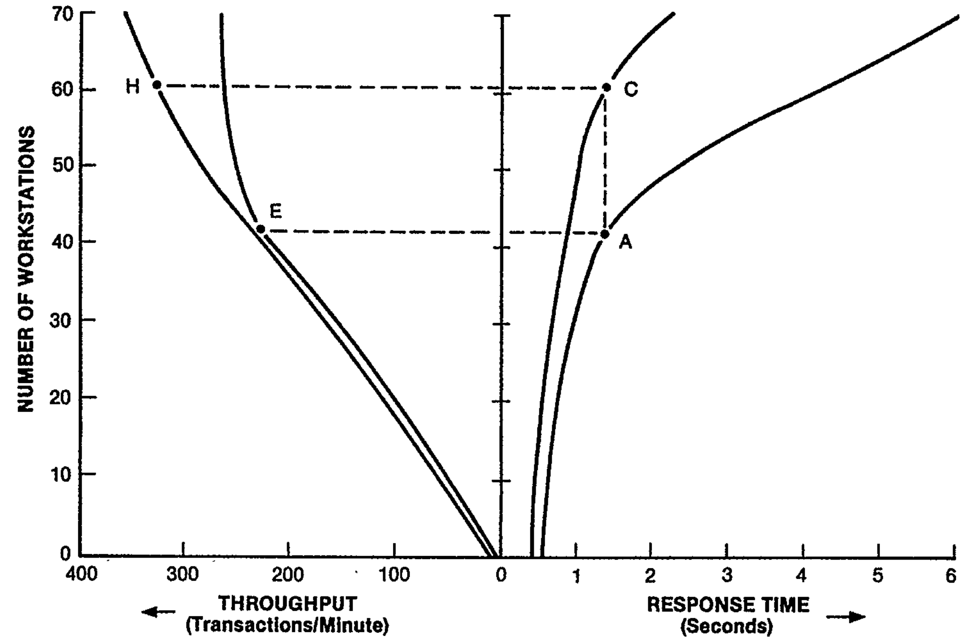


Figure 10: **System Transfer Function**
Same Response Time: (A, E) → (C, H)

Figure 10 displays throughput, response time and number of workstations for the two-disk system as in Figure 5, but with an additional axis. In Figure 10 the right-hand horizontal axis is response time as before, but throughput is also scaled horizontally to the left, and number of workstations is scaled vertically.

A given system state is now represented by a pair of values, one on the response curve, and another on the throughput curve, such as (A, E). "A" shows response time and "E" shows throughput for this configuration under these conditions.

Suppose it is desired to produce more throughput, but with no increase in response time, by re-locating the files. Starting at point A in Figure 10, move straight up to point C on the new allocation curve. From C go to H to find the new throughput. In this case, for no increase in response time, a very substantial gain in throughput is achieved (E to H) by re-allocating the files and increasing the number of workstations.

Alternatively, one can achieve reduced response time while holding throughput constant. In Figure 11, starting with the old file allocation in state (B, F), drop straight down from F to G (same throughput) then from G move across to D, to obtain response time with the new file allocation. Although throughput is the same, response time has been significantly reduced (from B to D), so that the same overall system workload in transactions per hour can be processed with fewer workstation operators.

Numerical values for these and other trade-offs are given in Table 6.

5. CONCLUSIONS

For workloads represented by the transaction benchmark, it is projected that substantial performance gains accrue from the proper utilization of dynamic space management techniques. For single disk systems, improvements in disk seek time translate directly into system performance improvement.

Configurations with two disks tend to be I/O bound if file storage has been laid out incorrectly. If the files have been badly placed (excessive seek time and lost of device contention) very large performance gains can be achieved via dynamic space management, both for global reorganization and reference spike relocation.

These file allocation strategies are especially relevant to conversion from small removeable disks to new high-density non-removeable media. Whenever two or more old packs are copied onto a larger unit, throughput and response time will suffer unless the combined files are correctly allocated.

Finally, it bears repeating that these performance gains are highly visible to the user, even without a performance monitor. More transactions/hour, at the central site, means more work is getting done, with expected improvements in the 10-30% range. Shorter average response time at the workstation improves operator morale and supports a cushion of capability for peak-load capacity, in the 20-60% range. Overall, there is a prospect for <u>improved productivity</u>, with no increase in current resources.

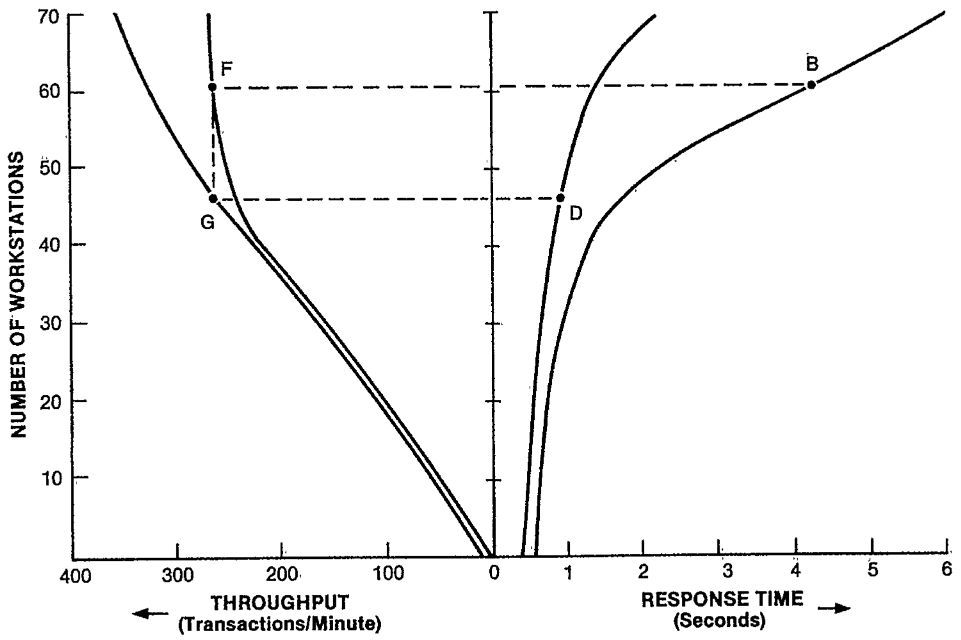


Figure 11: **System Transfer Function**
Same Throughput: (B, F) → (D, G)

REFERENCES

Amdahl, G.M. (1970), Storage and I/O Parameters and Systems Potential, Proceedings IEEE Computer Group Conference, June 1970, pp 371-372.

Arora, S.R., and A. Gallo (1973), Optimization of Static Loading and Sizing of Multi-Level Memory Systems, Journal ACM, April 1973, pp 307-319

UNIVAC (1980), Sperry Univac OS/3 Consolidated Data Management, UP-8825, CIDC, Sperry Corp., King of Prussia, Pa., 19406

UNIVAC (1981), Sperry Univac OS/3 System Activity Monitor, UP-8812, CIDC, Sperry Corp., King of Prussia, Pa., 19406.

APPENDIX

TABLE 1: FILE REFERENCE STATISTICS

File Category	Number of Allocation Entities*	Disk Tracks	Reference** (percent)
1. AA Files (System)			
A Segments	18	65	17.8
B Segments	30	2340	5.9
2. BB Files (System)			
A Segments	6	30	8.9
B Segments	19	400	9.4
3. CC Files (User)			
A Segments	7	53	13.8
B Segments	9	2085	8.8
4. DD Files (User)			
A Segments	7	53	20.7
B Segments	10	2095	14.7
5. TOTAL	106	7121	100.0

* Allocation entities are file subdivisions which appear as separate entities in the space allocation tables. B segments are composed of text or data records; A segments are usually (but not always) index lists. A file is composed of one or more allocation entities/segments. There are 38 files in this example.

** Reference frequencies have been altered to make odds more even across all files. The original benchmark referenced only a few files.

TABLE 2: GLOBAL REORGANIZATION
ONE-DISK CONFIGURATION

Old Allocation (A) Vs New (B)

	N	V	R	Usage	
Run No.	Wksta.	Trans/Min.	MSec.	CPU	Device
1A	1	5	541	.014	.036
1B	1	5	410	.014	.024
2A	10	56	655	.143	.360
2B	10	53	480	.137	.225
3A	20	105	935	.271	.676
3B	20	112	646	.288	.472
4A	30	149	2115	.383	.943
4B	30	170	978	.434	.719
5A	40	157	5305	.399	1.000
5B	40	207	1345	.539	.872
6A	50	160	9491	.405	1.000
6B	50	240	2779	.620	1.000
7A	60	162	13433	.411	1.000
7B	60	244	4763	.634	1.000

TABLE 3: GLOBAL REORGANIZATION
TWO-DISK CONFIGURATION

Old Allocation (A) Vs New (B)

	N	V	R	Usage	
Run No.	Wksta.	Tran/Min.	MSec.	CPU	Devices
10A	10	55	620	.140	.206,.158
10B	10	54	454	.139	.111,.125
11A	20	109	699	.283	.408,.317
11B	20	112	498	.292	.236,.265
12A	30	160	900	.410	.604,.476
12B	30	168	576	.433	.368,.410
13A	40	217	1302	.561	.828,.480
13B	40	227	821	.586	.929,.580
14A	50	254	1844	.643	.978,.788
14B	50	282	1092	.720	.661,.739
15A	60	262	3898	.664	1.000,.788
15B	60	323	1293	.821	.784,.854
16A	70	266	6083	.684	1.000,.795
16B	70	357	2229	.908	.864,.960

TABLE 4: REFERENCE SPIKE RELOCATION
ONE-DISK CONFIGURATION

A: Before Relocation
B: After Reloction

	N	V	R	Usage	
Run No.	Wksta.	Tran/Min.	MSec.	CPU	Devices
20A	1	5	568	.014	.038
20B	1	5	515	.014	.033
21A	10	55	804	.142	.375
21B	10	54	672	.140	.326
22A	20	106	1442	.271	.725
22B	20	104	987	.268	.620
23A	30	141	2866	.358	.957
23B	30	149	1782	.381	.892
24A	40	148	6380	.374	1.000
24B	40	169	4553	.432	1.000

TABLE 5: REFERENCE SPIKE RELOCATION
TWO-DISK CONFIGURATION

A: Before Relocation
B: After Relocation

	N	V	R	Usage	
Run No.	Wksta.	Tran/Min.	MSec.	CPU	Devices
30A	1	5	568	.014	.021,.016
30B	1	5	515	.014	.016,.018
31A	20	107	812	.274	.433,.335
31B	20	107	716	.278	.352,.316
32A	40	210	1503	.536	.860,.668
32B	40	221	1398	.561	.709,.758
33A	60	250	5243	.641	1.000,.781
33B	60	290	2805	.742	.888,.982

TABLE 6: COORDINATES FOR TRADE-OFF COMPARISONS
GLOBAL REORGANIZATION - TWO-DISK CONFIGURATION

New Allocation with Load Balancing Vs Original Allocation

	Wksta.	Throughput	Response Time	CPU Usage
Coordinates	N	Trans/Min.	Sec./Trans.	CPU
1. Original Allocation				
(A, E)	42	220	1.40	.560
(B, F)	61	266	4.38	.677
2. Same Response Time				
(A, E)	42	220	1.40	.560
(C, H)	61	330	1.40	.840
3. Same Throughput				
(B, F)	61	266	4.38	.677
(D, G)	47	266	.98	.677
4. Same No. of Workstations				
(B, F)	61	266	4.38	.677
(C, H)	61	330	1.40	.840