D2

N78-32468

NASTRAN COMPUTER RESOURCE

MANAGEMENT FOR THE MATRIX DECOMPOSITION MODULES

Charles W. Bolz
Computer Sciences Corporation

SUMMARY

Detailed computer resource measurements of the NASTRAN matrix decomposition spill logic were made using a software input/output monitor. These measurements showed that, in general, job cost can be reduced by avoiding spill. The results indicated that job cost can be minimized by using dynamic memory management. A prototype memory management system is being implemented and evaluated for the CDC CYBER computer.

INTRODUCTION

The early large structural analysis programs were designed for second-generation computer systems that were severely core-limited, requiring structural programmers to develop ingenious strategies for using mass storage to extend the range of solvable problems. It was for such a computer that NASTRAN was initially developed, and the matrix decomposition code with its efficient spill logic was a singular achievement in numerical analysis software. As NASTRAN was implemented on third generation computers which allowed multiprogramming, such as the UNIVAC 1108 and the CDC 6000 series, it remained expedient to use as little central memory as possible in order to maximize overall system efficiency. However, present day computers such as the CDC CYBER 175 and the UNIVAC 1110 have very large, fast, low-cost semiconductor memories, and excessive mass storage usage can rapidly degrade overall system efficiency and increase job cost. It therefore becomes important for the user to select an optimum memory region size for his problem.

In order to accurately assess the effects of memory region size on I/Ø utilization and job cost, a software monitor was developed to measure I/Ø volumes by file on CDC CYBER computers. Spill volume statistics were accumulated for the SDCØMP and CDCØMP matrix decomposition modules using NASTRAN Level 17.0.0 on the CDC CYBER 175 under the NØS 1.2 operating system. These statistics were interpreted using job cost accounting relations typical of CDC and UNIVAC systems. The results suggested that a dynamic memory management system designed to avoid spill would be cost effective, and a prototype system is being implemented on the CDC CYBER.

SPILL

Matrices to be decomposed by NASTRAN are normally sparse banded matrices with relatively few terms away from the band. During the decomposition, it is desirable to have all the non-zero terms of a row, and all the non-zero terms of the triangular factor generated by reduction of that row, in main memory. If this is possible for each row, then the matrix need be read in from secondary storage only once during the decomposition, and the factorized matrix written out. If insufficient memory is allocated, however, intermediate results must be stored on spill files. Numerous passes through the spill files may be required to perform the decomposition.

The matrix decomposition and spill logic is described in detail in references 1 and 2.

JOB COST ACCOUNTING ALGORITHMS

The astute NASTRAN user interprets computer resource utilization guidelines in terms of job cost, as assessed by his installation accounting algorithm. Results presented in this study are interpreted in terms of two accounting algorithms: one used commonly at CDC installations, and the second at UNIVAC sites.

Many factors go into an accounting algorithm, but for NASTRAN execution only central memory used (CM), central processor-unit time (CPU), and mass storage input/output transfers (I/Ø) are important. In terms of these resources, the CDC accounting formula may be generalized as

$$\text{Cost} = (1 + C_1\ \text{CM})\ (C_2\ \text{CPU} + C_3\ \text{I/Ø})C_4$$

and the UNIVAC relation as

$$\text{Cost} = \text{CM}\ (C_2\ \text{CPU} + C_3\ \text{I/Ø})C_4$$

where C_2 and C_3 are functions of the CPU and mass transfer device speeds. The constant C_1 is set by the CDC NØS operating system at 0.007 per 512-word block. The dollar multiplier, C_4, is installation dependent, so all matrix decomposition costs presented in this study are normalized to the no-spill case.

IBM accounting formulas vary with installation and operating system, so the IBM user should interpret the results presented in terms of his particular system.

THE I/Ø MONITOR

The basic utility for this study is a software monitor which was originally developed by the author for analyzing the I/Ø usage of programs running under the CDC SCØPE operating system. The monitor decodes all I/Ø requests and records, by data block, the type of request and the number of physical records transferred between central memory and mass storage. The record is printed at the end of each module, as shown in Figure 1.

As adapted for the CDC version of NASTRAN, the I/Ø monitor is called from XIØRTNS, which is the interface between GINØ (General input/output) and the operating system (ref. 3). The monitor was validated by checking the total I/Ø volume printed out against accounting log (dayfile) statistics for each NASTRAN run. Since the monitor itself occupies only 350 words of CYBER 175 memory, and uses about 20 microseconds of central processor time per I/Ø request, it has negligible impact on the job environment.

THE COMPUTER RESOURCE UTILIZATION STUDY

Complex Decomposition

Two problems were chosen for study. The first is a complex eigenvalue analysis of a gas-filled, thin elastic cylinder (NASTRAN Demonstration Problem 7-2-1 of reference 4). This case requires decomposition of an order 390 complex matrix, and can be solved by NASTRAN in a reasonable memory region only by using the determinant method. This particular problem was the impetus of the present study. When it was run on an IBM S/360-95 under the Multiple Variable Tasking (MVT) operating system with a memory region of 410000 bytes, an I/Ø timeout resulted after twenty minutes I/Ø time. When the region size was increased to 500000 bytes, the I/Ø time was less than five minutes.

Computer resource requirements for this problem are shown in figure 2, and dramatically illustrate the effect of spill on resource utilization. As long as memory region size is small enough to require spill, I/Ø volume and CPU time are steep inverse functions of open core (scratch memory) size and job cost (as measured by the CDC accounting algorithm) is decreased by increasing core. But once sufficient open core is provided to avoid spill, CPU and I/Ø utilization remain constant, and job cost increases with increasing memory size.

Real Symmetric Decomposition

The second problem chosen is the static analysis of a long, narrow orthotropic plate, based on NASTRAN Demonstration Problem 1-4-1 of reference 4. This problem is useful for study because data can be readily generated for a broad range of grid sizes. Problem sizes ranging from 128 to 1100 active columns were studied. (For a given memory region, spill is closely related to the number of active columns.) These were produced by grids of from 300 to 2100 points, generating matrices of order 760 to 4990, respectively.

A problem size of 277 average active columns, generated by a grid of 660 points, resulting in a matrix of order 1660 was selected for detailed investigation. This problem has spill characteristics typical of large user problems commonly analyzed using NASTRAN. The grid is comparatively small; however for problems of similar spill behavior, CPU and I/Ø resource utilization are linearly proportional to matrix order for a constant memory region.

Results for this case are shown in figure 3 in non-dimensional form, normalized to the conditions at the open core size where spill is no longer required. The outstanding feature of figure 3 is the I/Ø required by spill. At an open core size of 50% of that required for in-core reduction, I/Ø volume is seven times that required for in-core reduction. The CPU time curve illustrates that, refined as the symmetric decomposition spill logic is, considerable computer time is used processing spill I/Ø. And the cost curve shows that the cost penalty incurred by using more open core is more than compensated for by the reduced I/Ø and CPU resource requirements.

To lend perspective, a cost curve was developed for a typical UNIVAC 1110 system, where cost is directly proportional to memory used, and I/Ø is relatively less expensive. This curve is not as dramatic as the CDC curve, but still shows the importance of increasing open core to minimize spill.

DYNAMIC MEMORY MANAGEMENT

When matrix decomposition dominates a NASTRAN problem, the foregoing discussion indicates that computer resource utilization can be minimized by requesting sufficient core to avoid spill, if possible. For typical problems, however, matrix decomposition is only part of the solution procedure. This is illustrated by the problem described in Table 1. The decomposition of the order 7000 matrix without spill would require a memory region of 160,000 decimal words on a CYBER 175, which is 30,000 words more than is available to a single program. But the decomposition step is only about 40% of the computational effort. Another 50% of the computation can be performed in 50,000 words core, and the remainder in 70,000.

This suggests that an ideal strategem to reduce computer costs would be to dynamically manage memory to give each module only the core it needs. Direct implementation of this idea would present a formidable task - 160 NASTRAN modules to be modified. However, the results presented in Table 1 indicate that most of these modules - input, sort, geometry processing, element matrix assembler and generator, etc; require a small memory region, and suggest the following three-phase memory management scheme.

(1) Execution of each module is attempted in a small memory region.

(2) Modules which can be expected to have large memory requirements compute and request the needed core.

(3) Any other module which runs out of core while executing has its memory region expanded to a predetermined intermediate size.

CDC IMPLEMENTATION OF DYNAMIC MEMORY MANAGEMENT

The dynamic memory management scheme described above is being implemented on the CDC CYBER 175 as follows.

(1) The user specifies to NASTRAN an initial and a nominal memory region size.

(2) Before invoking each module, the link driver (XSEM) routine calls a subroutine MEMMGR (memory manager) to reset the memory region to its initial value.

(3) The matrix decomposition routines call MEMMGR to obtain the open core needed to execute without spill. If insufficient memory exists, all available memory is obtained.

(4) Modules that run out of open core normally issue an error abort call to subroutine MESAGE. This call is intercepted by MEMMGR, the nominal memory region is assigned, and control returned to the calling module. (Note that this requires that the call to MESAGE be an in-line call).

This scheme is being tested using the cases of figures 1 and 2 and Table 1. The predicted cost savings are shown in figure 2. These cases indicate that dynamic memory management to avoid spill can reduce job costs significantly.

CONCLUSION

An input/outout monitor was developed for the CDC version of NASTRAN which allows detailed analysis of computer resource utilization of the matrix decomposition modules. This analysis shows that for typical accounting algorithms, job costs can be reduced by avoiding spill in the decomposition. Analysis of a typical problem indicates that dynamic memory management could further reduce overall job cost.

REFERENCES

1. The NASTRAN Programmer's Manual, NASA SP-222(03), July, 1976.

2. The NASTRAN Theoretical Manual, NASA SP-221(03), March, 1976.

3. Brown, W. K., and Schoellmann, W. F., "Study of the NASTRAN Input/Output Subsystems," Sixth NASTRAN Users' Colloquium, NASA CP-2018, October, 1977.

4. The NASTRAN Demonstration Problem Manual, NASA SP-224(03), July 1976.

TABLE 1

THERMAL STABILITY STUDY

Order of Matrix = 7215
Average Active Columns = 238
Maximum Active Columns = 505
Three Spill Groups

Operation	CPU, seconds	I/O, 10^3 PRU(1)	Memory Region (60 bit words)
Input Processing	49	16	52000
Geometry Processing	14	20	52000
Element Matrix Processing	140	81	52000
Constraint Elimination	183	25	52000
Decomposition	307	157	98000(2)
Static Solution Generation	70	81	66000
Totals	736	380	

(1) One PRU = Sixty-four 60 bit words

(2) The decomposition would require 160000 words without spill

TABLE 2

NASTRAN DYNAMIC MEMORY MANAGEMENT ON THE CYBER 175

EXPECTED RESOURCE UTILIZATION AND COST SAVINGS

Problem	Memory Region (10^3 word)	CPU (seconds)	I/Ø (10^3PRU)(1)	Cost Savings (Percent)
Demo Problem 7-2-1	52	45.3	12.4	
	74	242.8	49.2	2.5%
Demo Problem 1-4-1	52	49.6	34.7	
	94	66.3	26.9	12.7%
Thermal Stability Study	52	386	142	
	66	70	81	
	98(2)	307	157	14.3%

(1) One PRU = sixty-four 60 bit words

(2) The decomposition would require 160,000 words without spill

ORIGINAL PAGE IS
OF POOR QUALITY

105 RBMG2 END

CALL 21 FILE	PPCALLS	PRUS TRANSFERRED	AVERAGE DATA TRANSFER	READ/WRITES	OPEN/CLOSE	FILE POSITION
POOL	14	140	10	14	0	0
NTRAN	6	323	53	6	0	0
KFF	49	819	16	49	0	0
SCRATCH4	999	16355	16	998	0	1
SCRATCH1	5	38	9	4	0	1
	472	7496	15	472	0	0
KGGX	5793	94741	16	5786	0	7
SCRATCH3	6513	106537	16	6506	0	7
MOD. TOTAL	13851	226449	16			
GRAND SUM	15167	240131	16			

111 SSG1 END

CALL 22 FILE	PPCALLS	PRUS TRANSFERRED	AVERAGE DATA TRANSFER	READ/WRITES	OPEN/CLOSE	FILE POSITION
POOL	11	131	11	11	0	0
NT-AN	19	1154	60	19	0	0
KGGX	9	54	9	6	0	3
SLT	6	60	10	6	0	0
CASECC	4	40	10	4	0	0
SCRATCH1	5	38	9	4	0	1
SCRATCH3	5	38	9	4	0	1
BGPDT	4	54	13	4	0	0
SIL	2	20	10	2	0	0
PG	2	18	9	2	0	0
MOD. TOTAL	67	1607	23			
GRAND SUM	15234	249738	16			

MPYAD--NULL MATRIX PRODUCT

CALL 28 FILE	PPCALLS	PRUS TRANSFERRED	AVERAGE DATA TRANSFER	READ/WRITES	OPEN/CLOSE	FILE POSITION
POOL	4	40	10	4	0	0
NTRAN	8	898	112	8	0	0
KELM	3	18	9	2	0	1
KSS	3	18	9	2	0	1
SCRATCH6	3	18	9	2	0	1
PG	3	18	9	2	0	1
KDICT	3	18	9	2	0	1
KGGX	3	18	9	2	0	1
PS	3	18	9	2	0	1
ULV	3	18	9	2	0	1
YS	3	18	9	2	0	1
USET	3	18	9	2	0	1
KFS	3	18	9	2	0	1
MOD. TOTAL	49	1136	25			
GRAND SUM	16523	271898	16			

137 SDR2 END

CALL 29 FILE	PPCALLS	PRUS TRANSFERRED	AVERAGE DATA TRANSFER	READ/WRITES	OPEN/CLOSE	FILE POSITION
POOL	4	40	10	4	0	0
NTRAN	15	1152	76	15	0	0
KELM	3	18	9	2	0	1
KSS	3	18	9	2	0	1
SCRATCH6	3	18	9	2	0	1
PG	3	18	9	2	0	1
KDICT	7	52	10	5	0	2
KGGX	198	3191	16	196	0	2
PS	11	116	12	9	0	2
ULV	3	18	9	2	0	1
YS	9	84	12	7	0	2
USET	3	18	9	2	0	1
KFS	3	18	9	2	0	1
CASECC	8	80	10	8	0	0
MPT	2	20	10	2	0	0
EST	19	309	16	19	0	0
EQEXIN	7	91	13	7	0	0
KLL	2	20	10	2	0	0
UGV	8	108	13	8	0	0
MOD. TOTAL	311	5389	17			
GRAND SUM	16834	277287	16			

Figure 1. Output from the NASTRAN I/Ø Monitor (typical).

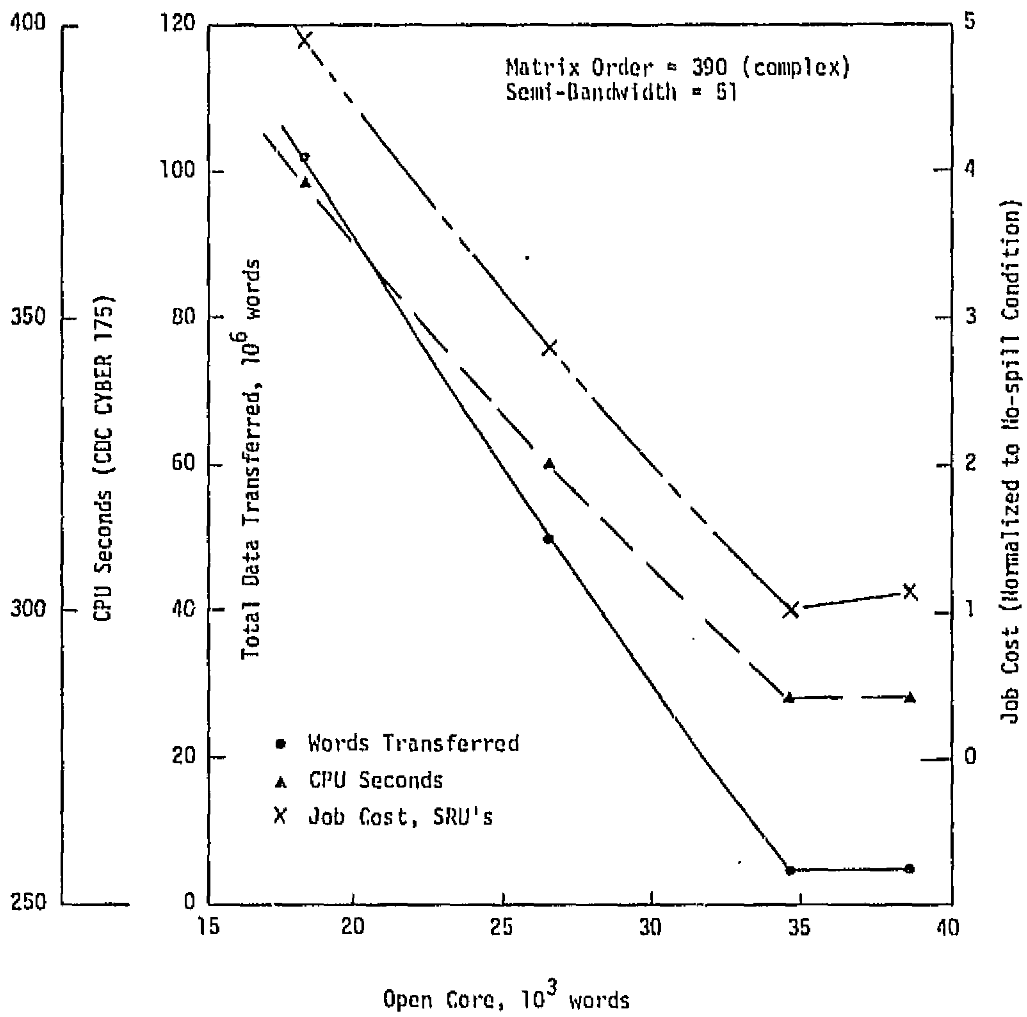

Figure 2. Resource Analysis vs. Open Core
Eigenvalue Analysis of a Thin Elastic Cylinder.

ORIGINAL PAGE IS
OF POOR QUALITY

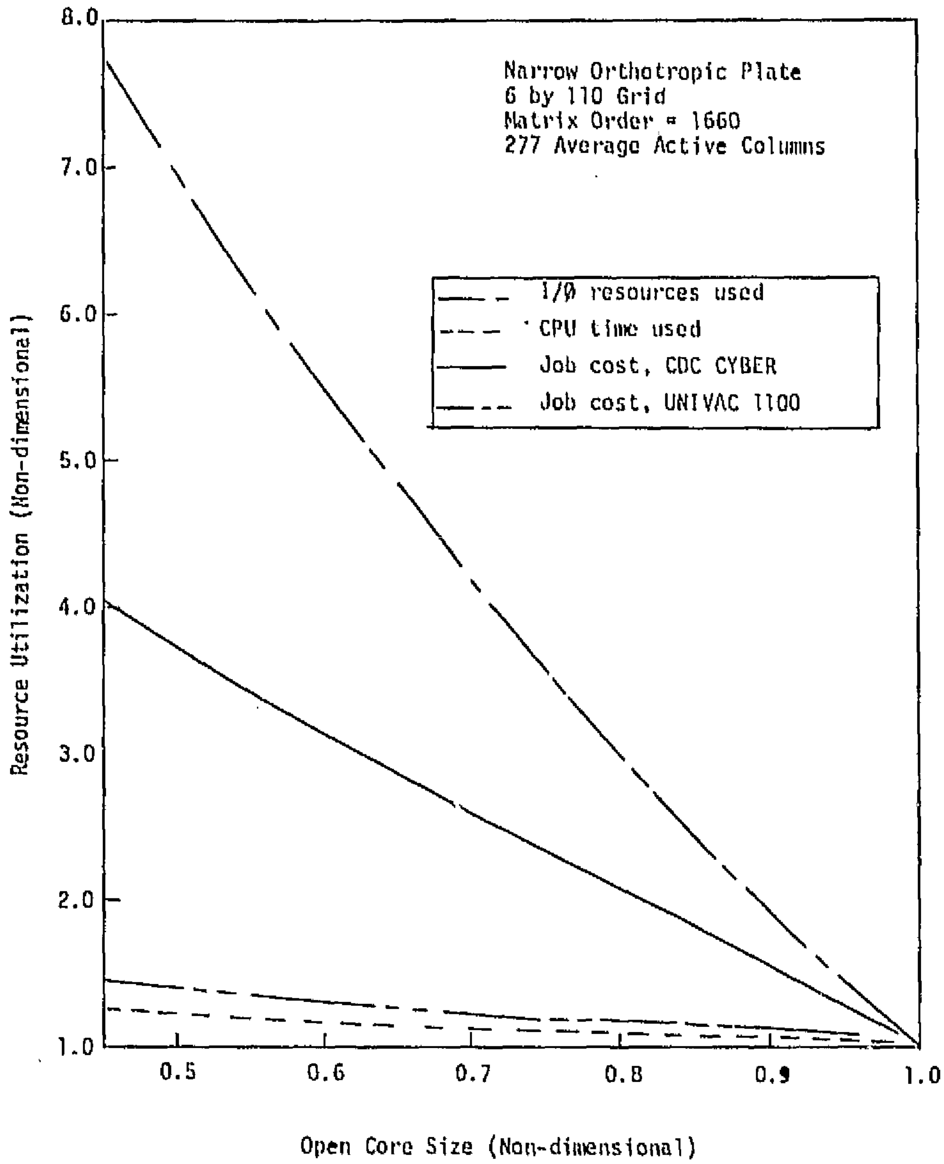

Figure 3. Resource Utilization for Symmetric Decomposition Normalized to No-spill Values.