

Simon Griffiths, Witwatersrand University Computer Centre

## ABSTRACT

The VM/CMS SAS system can be installed as modules or as saved segments. Use of segments results in better system performance. This paper deals with the advantages and disadvantages of the segment- and module-orientated versions. An analysis of four SAS test programs running with modules and segments is made using (1) CPU time, (2) virtual storage utilisation and (3) the number of start I/Os, as measures of system performance. All three measures indicate that the use of segments is more efficient than modules. Some disadvantages of using SAS software from segments are also discussed.

users can be saved in shared segments. When any virtual machine loads the saved system, it shares the same set of pages in real storage as other virtual machines using the system. This results in savings of both real and external storage. Discontiguous saved segments have an added advantage because the segments can be attached to, and detached from, a virtual machine. Programs which are not continuously required can be shared and loaded when needed. The stipulations for discontiguous saved segments is that the code must be re-entrant and the segments must be loaded and saved at an address beyond the normal end of a virtual machine. System programs and user programs that are not shared may be termed modules.

## INTRODUCTION

At computer sites running the IBM VM/CMS operating system, the SAS® system can be installed in two ways.

1. The simplest procedure is to unload the SAS system files from tape and store them as modules on a CMS mini-disk. This procedure, though, is inefficient in the use of system resources and in job execution times.
2. The second way is more complex and involves defining the SAS modules as discontiguous saved segments. Saved segments are, however, a more efficient means of running the SAS system.

There are pros and cons to both methods, and so the decision on which one to adopt will depend on their relative merits. To the author's knowledge, SAS Institute has not as yet published bench mark figures showing the performance differences between module- and segment-types of installations. Nevertheless, the Institute has released some figures to indicate the difference and also recommends that saved segments be installed. A short article on the subject appeared in the Spring 1984 issue of SAS Communications® (SAS Institute, 1984). The present paper contains a quantitative analysis of the performance of module- and segment-orientated SAS systems.

## MODULES AND SEGMENTS IN THE VM/CMS OPERATING SYSTEM

The fundamental concept of the VM/CMS operating system is that of the virtual machine. Users signing on to the system are given the functional equivalent of their own computer. Each virtual machine is allocated an amount of virtual memory which can be addressed by user programs and commands. System code that is shared by all

## THE SAS SYSTEM IN MODULES AND SEGMENTS

In CMS, the SAS files may be installed as modules on a service machine. When a user runs a SAS program, the necessary files are loaded into the user's own virtual memory. Loading the modules for each SAS step into memory results in an increased amount of I/O, and generally longer execution times. Moreover, if there are many users running SAS programs simultaneously, each user receives an individual copy of the required SAS modules in virtual memory. Naturally, swapping modules in and out of memory consumes system resources, and on a heavily loaded system, the use of modules may slow down response times.

If saved segments are used, only a single copy of the SAS code is necessary. Several programs running concurrently may share the same segment of SAS code rather than using separate copies of the same module. Overall usage of virtual memory can thereby be reduced resulting in improved performance. In a multi-SAS user environment the overhead of managing segments is less than the overhead of module management.

## ANALYSIS OF THE SAS SYSTEM USING MODULES AND SEGMENTS

The 82.3 version of CMS/SAS runs on the Witwatersrand University's IBM 3083 running under VM/SP HPO 3.2. As there are over 300 SAS users at the University, it was considered a worthwhile exercise to compare the performance of SAS programs that run from modules versus those using segments. When the SAS system was installed, a system file, PROFILE SAS, was loaded which contains code to indicate whether segments or modules are used. To compare the efficiency of SAS programs accessing modules or segments, the PROFILE SAS was copied to a virtual machine and the code modified depending on whether segment or module performance was being tested.

There are three main variables by which the efficiency of SAS jobs can be measured.

1. The CPU time for each step (DATA or PROC) in a SAS program.
2. The amount of virtual storage used in each step.
3. The amount of I/O.

Obtaining values for the first two variables is very easy since they are reported in the SAS log at the completion of every step. The I/O rate cannot be obtained while running a SAS program, but an approximate value can be found by displaying the total number of start I/Os (the CPU program instruction to begin I/O operations) immediately before and after running the program. Subtracting one value from the other gives a reasonable indication of the I/O rate during that program.

The programs used in the evaluation process were the test programs supplied for the SAS base product, for SAS/ETS™ and for checking SAS/GRAPH® output on a Tektronix 4010 screen. These programs are, respectively, TESTBASE SAS, TESTETS SAS and GTEK4010 SAS. The test programs were run when the 3083 was moderately heavily loaded with the CPU usage at 100 percent, and approximately sixty users signed on, of which about six were running SAS programs.

#### CPU TIMES

The CPU times for programs using segments was definitely lower than those running from modules (Table 1). Although the differences between each pair of steps may appear small (Figs. 1-3), the cumulative effect produces much greater differences. As a result, the longer the program, the more obvious is the advantage of using segments. For example, the difference for the GTEK4010 program, with seventeen steps, was only 0.14 seconds, whereas the TESTETS programs, with sixty-five steps, was 1.89 seconds faster in segments.

When studying individual steps, procedures NLIN and X11 used the greatest amount of CPU time, as well as several SAS/GRAPH procedures (G3D, GMAP, GCONTOUR, GCHART, GPLOT). A data step in the TESTETS program also took a long time to execute, possibly because it contained multiple ARRAY statements and a large DO loop. To select those steps that ran quicker in either segments or modules the cutoff value was arbitrarily set as a CPU time difference of 0.05 seconds. PROC NLIN and a PROC PLOT with the overlay option were identified by the cutoff criterion as notably faster when running under segments. Similarly, PROCs MATRIX, MEANS, FREQ, REG in the SAS base software performed better in segments. In the TESTETS program, the systems modelling procedures (SYSREG, SIMLIN, SYSNLIN) were definitely speedier when

executing from segments, as were some of the time series procedures AUTOREG and STATESPACE, and the powerful report-generator PROC COMPUTAB. PROC X11 did not rank as significantly better, neither did any of the SAS/GRAPH procedures meet the criterion. Except for the DATA step mentioned above, the times of DATA steps and PROC PRINTs also deserve attention. There is almost no difference in CPU time between DATA steps using segments and those using modules, perhaps due to negligible differences in compilation time. A completely different picture arises with PROC PRINT which is very efficient when under segments. Often, PROC PRINT is 50-100 percent faster if segments are installed.

Reasons for the time differences, or lack of difference, in some procedures cannot be positively identified because the source code of the PROCs could be studied. It may be, however, that those routines exhibiting the greatest difference are those which are the most complex of the SAS procedures. Unfortunately, there are no test programs to analyse the performance of SAS/FSP® procedures. Nevertheless, experience suggests that PROC FSEDIT runs slower if modules are used.

#### VIRTUAL STORAGE

The performance improvement of SAS programs in segments as opposed to modules becomes clear when analysing the amount of virtual storage required. Not only are there significant differences between the cumulative values, but also between the individual steps (Table 2). On average, as well as in total, running the SAS test programs with modules requires 50-65 percent more storage than with segments. The improvement was most obvious when running SAS/GRAPH programs because the graphics procedures tend to consume more storage than others. Before segments were installed, SAS/GRAPH jobs would routinely 'abend' in a 1M virtual machine due to lack of storage. Since segments have been installed, this problem has almost disappeared. Apart from SAS/GRAPH, the procedures which used most virtual storage were PROCs SORT, GLM, EDITOR, MORTGAGE, SIMLIN and GMAP. SORT and GLM are known make extensive use of memory, which is one reason why PROC ANOVA is recommended for analysis of variance rather than GLM under certain circumstances.

#### I/O

Analysis of the I/O rates repeat the results from the comparison of virtual storage utilisation. The large value of start I/Os associated with the use of modules shows how the effect of swapping modules in and out of storage markedly increases the I/O rate (Table 3). A percentage increase of 127 and 144 percent, respectively, was re-

corded when modules were being used instead of segments in the TESTBASE and TESTETS programs. Whereas only a fourteen percent improvement was measured for the SAS/GRAPH test program. Since the graphics procedures require larger amounts of virtual storage than other procedures, the lower percentage increase may indicate that movement between real and external storage is a major factor in the performance of SAS/GRAPH procedures.

#### DISADVANTAGES OF SAS SEGMENTS

So far in the discussion, the emphasis has been on the advantages of shared segments over modules. There are, however, disadvantages to installing SAS segments. Three major problems have been encountered.

1. A large amount of system disk space is required to store the segments of the 82.3 version of VM/CMS SAS. This posed some problems for the Computer Centre, which has limited disk storage space, when trying to allocate space between SAS segments and system programs.
2. When saving segments, the load address must not be too high or real storage is wasted on segment table entries that may never be used. Conversely, too low an address may overlay the size of the virtual machine or other segments that are attached at the same time. Thus, there is a trade-off between the maximum virtual machine size and efficient use of real storage. At Witwatersrand University, the load address of SAS segments is just above 4M, but occasionally users with large programs submit SAS jobs to a 10M service machine, and so lose the use of the segments. It is these jobs that would benefit most from more efficient execution of SAS procedures.
3. Two SAS procedures (PROCs G3D and FORMS) contain errors, probably non-re-entrant code. As a result, these two procedures sometimes do not run if segments are used.

#### CONCLUSION

The recommendation by SAS Institute that VM/CMS SAS be installed in discontinuous saved segments to improve efficiency, rather than in modules, appears to have been substantiated by the analysis presented in this paper. Although there are some disadvantages associated with segments, the advantages seem to far outweigh the problems. At VM/CMS computer centres where many users may be running SAS jobs at the same time, use of segments will help to maintain adequate response times, whereas modules will lead to deteriorating response times as more people come to learn and use the SAS system.

#### ACKNOWLEDGEMENTS

I would like to thank the Director and my colleagues at the Witwatersrand University Computer Centre for assistance and advice during the preparation of this paper, and the University Cartography Unit for preparing the figures.

#### REFERENCE

SAS Institute, 1984: Saved Segments and the VM/CMS SAS System, SAS Communications, 9(4), p.20-22.

SAS, SAS/FSP, SAS/GRAPH and SAS Communications are registered trademarks of SAS Institute Inc., Cary, NC, U.S.A.

SAS/ETS is a trademark of SAS Institute Inc.

Author's address: Simon Griffiths: Computer Centre, Witwatersrand University, 1 Jan Smuts Avenue, Johannesburg, 2001 South Africa.

TABLE 1

COMPARISON OF CPU TIMES BETWEEN SEGMENTS AND MODULES  
(in seconds).

PROGRAM	TOTAL TIME		NO. OF STEPS QUICKER		
	Segments	Modules	Segments	Modules	Neither
TESTBASE	10.93	12.35	40	0	24
TESTETS	14.76	16.65	45	4	16
GTEK4010	12.10	12.24	8	1	8

FIGURE 1 TESTBASE CPU TIME COMPARISON

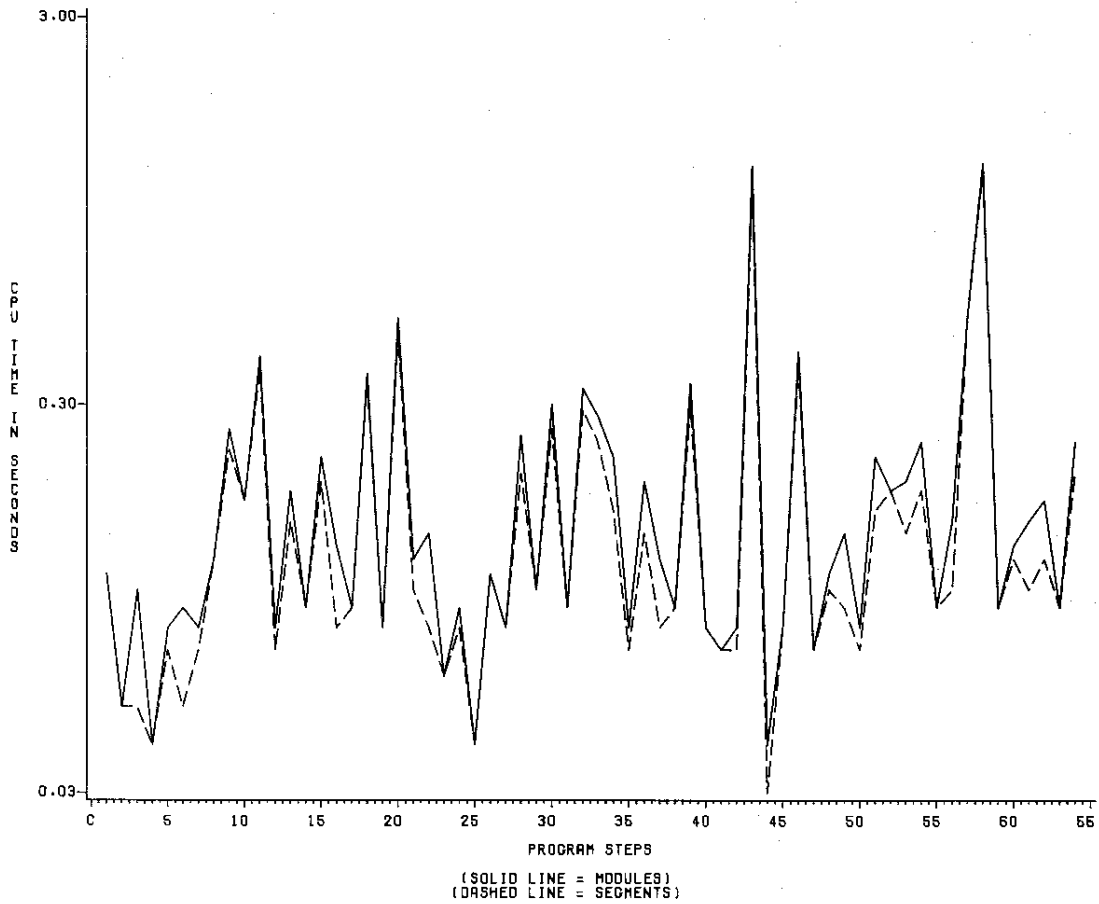


FIGURE 2 TESTETS CPU TIME COMPARISON

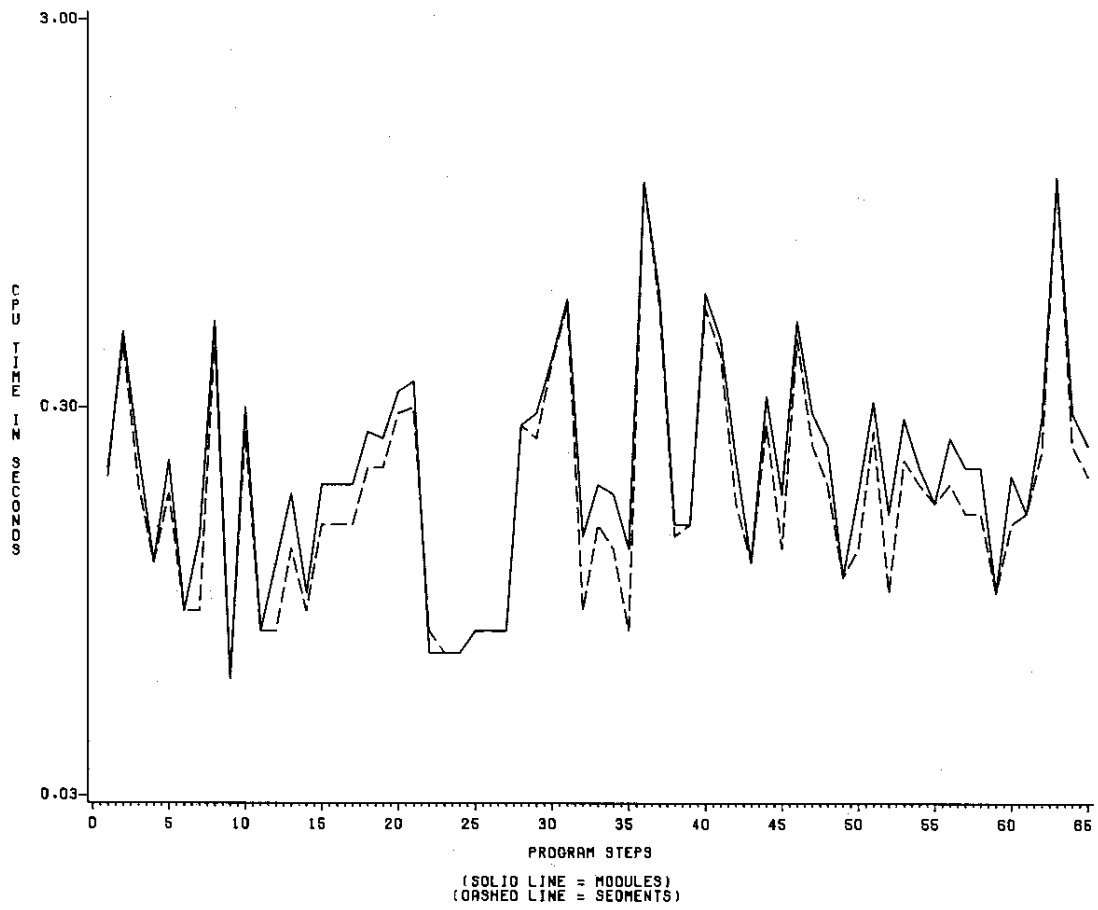


FIGURE 3 GTEK4010 CPU TIME COMPARISON

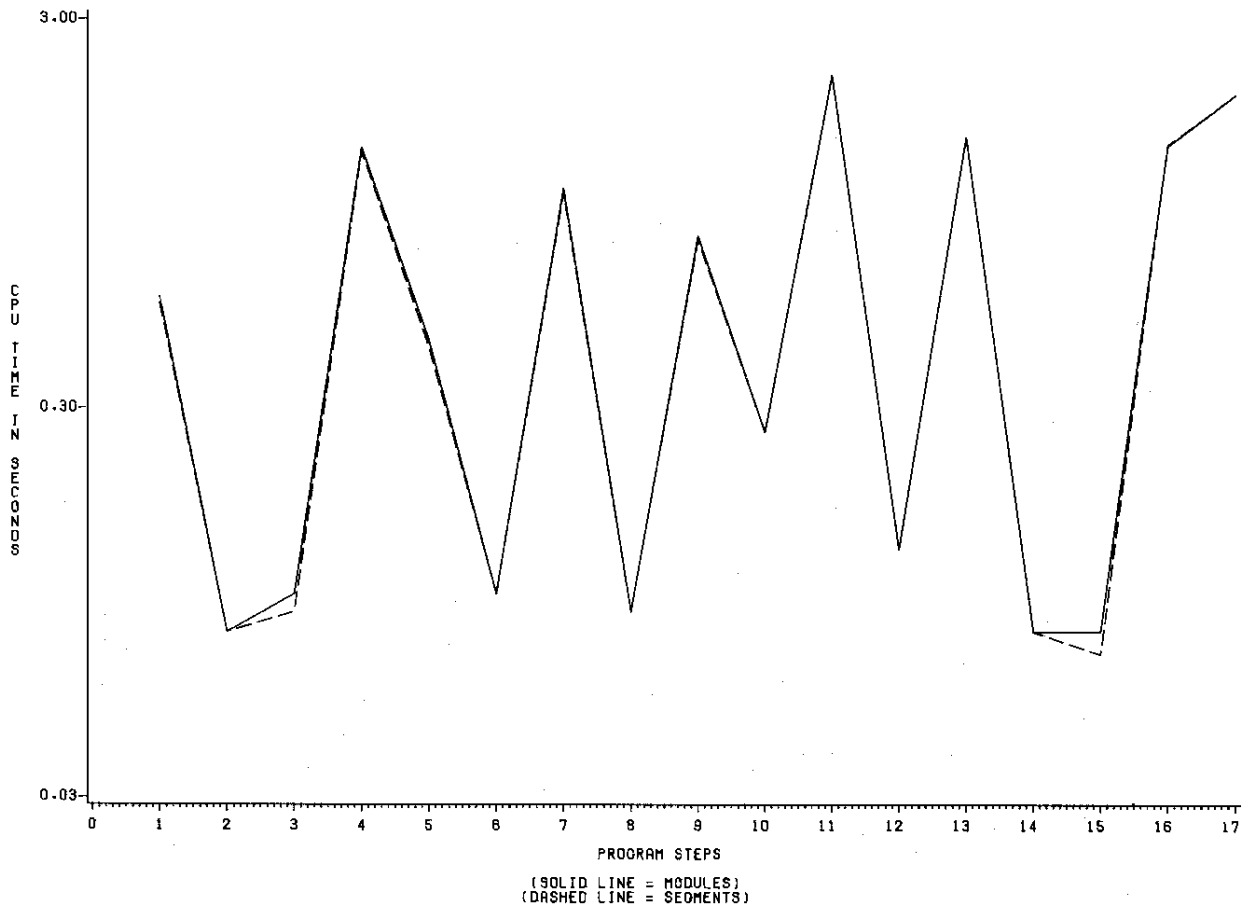


TABLE 2

COMPARISON OF VIRTUAL STORAGE BETWEEN SEGMENTS AND MODULES (in kbytes).

PROGRAM	MEDIAN STORAGE PER STEP		CUMULATIVE STORAGE	
	Segments	Modules	Segments	Modules
TESTBASE	576	896	39200	60618
TESTETS	576	896	39926	61686
GTEK4010	704	1024	11406	16822

TABLE 3

COMPARISON OF I/O RATES BETWEEN SEGMENTS AND MODULES

PROGRAM	START I/O DIFFERENCE	
	Segments	Modules
TESTBASE	1816	4117
TESTETS	2133	5205
GTEK4010	1895	2152