Performance Enhancements in SAS® Procedures for Small and Large Applications

Leigh A. Ihnen, SAS Institute Inc.
Cary, NC

ABSTRACT

The term Release 6.07 will be used to indicate only the current Release 6.07 of the SAS System for MVS, CMS, and VMS. Release 6.07 continues the enablement of small and large applications through performance improvements at the procedure level. Solutions for applications too large to run in Release 5.18 are enabled by exploiting larger address spaces. At the same time enhancements to procedures allow small jobs to run faster. The continuing evolution of Version 6 of the SAS System aims to provide cost-effective solutions for a wide range of enterprise applications. With this release better utilization of host-dependent features such as IBM® XA and ESA architecture, vector facilities, and interfaces to other software allow applications to be designed on a larger scale.

INTRODUCTION

The performance and flexibility of software on both large and small problems is becoming increasingly more important. In the 1990's local economies will face competition from global markets. Ensuring profitability will require production of quality products and services that are price competitive. Increased worker productivity and improvements in the production and distribution processes are factors in improving price competitiveness. Effective use of market and internal information allows companies to meet the challenge of increased competitiou. Integration of data from human resource management, market research, technical specifications, logistics, and manufacturing control in electronic communications present software with a mixture of large and small problems.

As more companies focus on shifting the corporate culture to a market-driven approach, information becomes an increasingly important critical success factor. Current approaches to increasing profitability such as Total Quality Management (TQM) are inherently dependent upon turning raw data into useful information. Release 6.07 of the SAS System provides a flexible tool for building an Information Delivery System (IDS). Traditional approaches to empowerment of employees involves turnkey applications. Unfortunately with the shift to cross-functional teams, an application should allow mulitple views of raw data. Since product development and quality improvement are ongoing processes, the need for new ways of viewing raw data as well as the creation of new types of raw data make turnkey applications possible inhibitors to quick response to changing market directions. Developing interactive applications with the SAS/AF® SCL language provides an easily updatable front end, while SAS/TOOLKIT™ software allows user extensions to the analysis capabilities of the SAS System.

Release 6.07 continues the evolution of Version 6 of the SAS System. With previous releases of the SAS System Version 6, the

MultiVendor Architecture™ (MVA™) allows applications based on the SAS System to work the same way across different hardware platforms. Since Release 6.06 of the SAS System was a total rewrite of Release 5.18, a significant number of compatibility issues were introduced. Also, the recoding in C versus host-dependent assembler code for some functions allowed Release 6.06 to suffer significant performance degradation when compared to Release 5.18. For this reason a major emphasis was placed on improving performance in Release 6.07. At the same time extensions in the MVA design were added to better utilize host-dependent features that improve performance.

Version 6 of the SAS System significantly extends the interactive, data access, communications, analytic, and presentation capabilities of the SAS System. Release 6.07 enhances the new functionality of Version 6 of the SAS System by improving performance. Other papers are being presented detailing performance improvements in the DATA step and SAS System I/O. The focus of this paper is the most commonly used procedures that are used to build an Information Delivery System(IDS). **Table 1** at the end of this paper presents statistics on CPU seconds used and frequency of invocation from nine MVS sites running Release 5.18. These statistics can be collected on MVS systems for Releases 5.18, 6.06, and 6.07 of the SAS System using IBM's SMF records. For Version 6 collection instructions see the *SAS Companion for the MVS Environment, Version 6, First Edition.* If you wish to have your site's usage patterns considered when performance improvements for future releases are planned, please send the SMF records on a tape to Leigh Ihnen (see the end of this paper for the address).

The following list summarizes the top procedures by CPU usage in rank order from **Table 1.**

- COPY
- SORT
- LOGIST
- FREQ
- SUMMARY
- MEANS
- GLM
- PRINT
- TABULATE
- DISCRIM
- FORMAT
- APPEND
- UNIVARIATE
- REG
- DELETE
- GPLOT
- CANDISC

The most used procedures listed above and the DATA step consume 98% of the reported CPU time. The procedures fall into four general classes of types of problems. The classes can be defined as data management, data presentation, simple data reduction, and analytic data reduction.

Data management includes data maintenance and organization and is provided by the following procedures:

- COPY
- SORT
- FORMAT
- APPEND
- DELETE

Data presentation procedures consist of

- PRINT
- TABULATE
- GPLOT

Simple data reduction procedures that are widely used are

- FREQ
- SUMMARY
- MEANS
- UNIVARIATE

Analytic data reduction procedures require the user to choose a procedure relevant to the user's data model. The use of analytic data reduction procedures is extremely site dependent. Many different types of data models can be analyzed using the followiug procedures.

- LOGIST
- GLM
- DISCRIM
- REG
- CANDISC

Data Management Procedures

Data management procedures can be divided into data ordering procedures and data set maintence procedures. The SORT procedure and the FORMAT procedure allow users to order data within a SAS data set. PROC SORT physically reorders the observations in a SAS data set, while PROC FORMAT allows users to logically regroup data within a SAS variable.

PROC FORMAT style formats performance has been improved in certain cases where all values are of the same type. Some examples are

- one value numeric with FUZZ=0
- character value with length less than 16
- numeric range value with FUZZ=0

Unfortunately Version 6 PROC FORMAT style formats are slower than Release 5.18 formats. In Release 5.18 for MVS and CMS, PROC FORMAT style formats are executable binary search code. In a future release of the SAS System executable user defined formats will be built using the SAS portable code generator. This will provide a significant performance improvement for large format tables.

PROC SORT includes a new design change for storing and honoring the sorted by variable information in the SAS data set header. When SAS code is written, there are several situations that might lead to re-sorting sorted data.

- permanent SAS data sets that are not read only
- %INCLUDE of predefined SAS code
- use of the SAS macro facility
- use of the SCL language

In general, unless the SAS data set exists in a single user environment sorting will be required if by-groups are used.

Some SAS applications at SAS Institute have shown up to a 25% reduction in CPU time even before the SAS base engine was improved. Improvements in the base SAS I/O engine and the SAS Institute supplied sort combine to make PROC SORT significantly faster for both large and small data sets. The times present in the following tables were run on an IBM 3090-600S using SyncSort as the host sort. Results should be applicable to other hosts and host sorts.

Release 6.07 vs. Release 5.18				
* denote SAS sort used in Release 607				
Obs.	Var.		CPU Savings	%
100	10	*	0.09	90%
500	10	*	0.10	91%
1,000	10	*	0.09	75%
5,000	10	*	0.11	42%
10,000	10	*	0.14	32%
100,000	10		0.81	22%
1,000,000	10		9.61	26%
100	50	*	0.08	80%
500	50	*	0.09	75%
1,000	50	*	0.10	67%
5,000	50	*	0.15	41%
10,000	50	*	0.18	29%
50,000	50		0.69	23%
100,000	50		1.47	25%
100	100	*	0.09	75%
500	100	*	0.10	67%
1,000	100	*	0.10	56%
5,000	100	*	0.19	38%
10,000	100		0.22	25%
50,000	100		1.08	24%
100,000	100		2.02	23%
100	500	*	0.12	48%
500	500	*	0.17	46%
1,000	500	*	0.21	42%
5,000	500		0.44	23%
10,000	500		0.81	23%
50,000	500		2.83	17%
100	1000	*	0.09	20%
500	1000	*	0.19	27%
1,000	1000		0.19	20%
5,000	1000		1.07	27%
10,000	1000		2.36	31%

The variables are numeric with a length of 8. Times reported are for preallocated data sets for use by PROC SORT.

The next table consists of a comparison of Release 6.06 and Release 6.07. A comparision of possible improvements in PROC SORT on UNIX and PC systems can be formed using the entries marked with stars in the table above.

Release 6.07 vs. Release 5.18			
Obs.	Var.	CPU Savings	%
100	10	0.01	50%
500	10	0.02	50%
1,000	10	0.04	57%
5,000	10	0.17	53%
10,000	10	0.13	30%
100,000	10	1.04	27%
1,000,000	10	9.76	26%
100	50	0.01	33%
500	50	0.03	50%
1,000	50	0.10	67%
5,000	50	0.16	42%
10,000	50	0.43	49%
50,000	50	1.01	30%
100,000	50	2.21	33%
100	100	0.01	25%
500	100	0.03	38%
1,000	100	0.10	56%
5,000	100	0.29	48%
10,000	100	0.58	47%
50,000	100	1.76	34%
100,000	100	3.37	33%
100	500	0.02	13%
500	500	0.06	23%
1,000	500	0.15	34%
5,000	500	0.41	22%
10,000	500	0.88	24%
50,000	500	1.89	12%
100	1000	0.09	20%
500	1000	0.12	19%
1,000	1000	0.28	26%
5,000	1000	0.37	11%
10,000	1000	0.56	10%

PROC APPEND in Release 6.07 outperforms Release 5.18 when the number of variables is less than 200 and the number of observations is less than 1,000. Also on the positive side, APPEND's performance has been improved across the board since Release 6.06. See **Table 2** at the end of this paper.

PROC COPY's performance in Release 6.07 has improved relative to Release 6.06. Unfortunately, results are mixed when com-

paring Release 6.07 to Release 5.18. A rule of thumb seems to be if the number of observations exceeds the number of variables Release 6.07 will out perform Release 5.18. For some examples see **Table 3** at the end of this paper.

Analytic Procedures

Since the use of analytic procedures is site dependent, only a short summary of performance improvements are presented. In Release 5.18 there were several factors that inhibited using SAS software on large problems. The limitations of a small address space have been relieved in Release 6.06. Poor performance, when the number of variables and/or observations grow large, has in some cases been resolved in Release 6.07. Release 6.07 can provide significant CPU savings compared to Release 5.18 and can solve a much larger class of problems.

PROC LOGIST was a user-contributed procedure in Release 5.18. The Institute Release 6.07 version can provide CPU reductions of up to 77% on problems around 1 CPU second in Release 5.18. If the problem has 10 variables and 5,000 observations, the CPU time is reduced from 22 seconds to 3 seconds. The most dramatic improvements are reductions of 90% plus on Release 5.18 runs using more than 700 seconds. The IBM vector facility can provide a further 50% reduction in CPU time for large variable models.

PROC DISCRIM in Release 5.18 used large amounts of CPU time when solving a large variable and/or large observation problem. Improvements in Release 6.07 can reduce CPU utilization by 40% to 90% when building the discriminant function. For 50 variables, Release 5.18 uses 4.35 CPU seconds and Release 6.07 uses .47 CPU seconds. For 200 variables, Release 5.18 uses 64.66 CPU seconds and Release 6.07 uses 4.62 CPU seconds. When building a discriminant function on 50 or more variables, a vector facility can reduce CPU utilization by 10% to 40%. If a pre-built discriminant function is used, then classifying test observations in Release 6.07 uses 35% to 99% less CPU than Release 5.18. The largest improvement is observed for 200 variables with 20,000 observations to be classified. Release 5.18 did the classification in 1,311 seconds and Release 6.07 used 8.83 seconds. When classifying data, a vector facility can improve performance by 3% to 20% for large variable problems.

PROC GLM can solve a much larger class of problems in Version 6. Test results show that performance can be 20% to 70% faster than Release 5.18. Use of a vector facility can result in an additional 20% to 50% reduction in CPU utilization for problems with a large number of parameters.

PROC REG has the same general performance characteristics in Releases 5.18 and 6.07. Additional vector support has been incorporated in Release 6.07. With a vector facility, reductions in CPU utilization of 25% to 70% have been observed for a wide range of medium to large problems.

Simple Data Reduction Procedures

PROC MEANS examples were considered that produced output data sets in keeping with the IDS strategy. Release 6.07 shows a fairly consistent 20% to 25% performance improvement when compared to Release 5.18. The only exception is when the number of variables is greater than 70 and the number of observations is less than a 100. In these cases the creation of the output name space dominates the problem. A performance degradation of up to 50% has been seen. This problem is known and will be fixed in a later release of Version 6.

PROC SUMMARY in Version 6 has removed the restriction that any class must have fewer than 32,676 levels. Further work has been done to make large inter-

action problems more robust than in Release 6.06. As with PROC MEANS, fairly consistent improvements in performance of 20% to 35% exist in Release 6.07 for small to medium problems. If the output name space dominates, degradation of up to 40% has been seen. Once the problem grows large enough to require intermediate results to be stored to disk, predicting performance gains versus Release 5.18 becomes harder. Typically Release 6.07 is as fast as Release 5.18 and can show as much as a 40% improvement in CPU time. An example on a 1024 X 760 grid with 3 points per node and three variables runs in 1,744.64 seconds in Release 6.06 and 660.27 seconds in Release 6.07. The improvement in CPU utilization is 62%. This problem has 778,240 interaction levels.

PROC FREQ shows mixed performance results compared to Release 5.18. For small problems the table printing dominates. A problem with 800 variables and 100 observations uses 3.39 seconds in Release 6.07 and 2.40 seconds in Release 5.18. This is a 40% performance degradation. For extremely small problems the better loading and initialization of Release 6.07 procedures offsets the printing overhead. If the same example is run on 10,000 obsevations, Release 6.07 shows an 8% improvement relative to Release 5.18. In all cases Release 6.07 is faster than Release 6.06. When the number of interaction becomes large, Release 6.07 shows performance gains of 25% to 40%. A 10 X 10 X 10 X 2 X 2 table requires 174 seconds in Release 5.18 but only 102 second in Release 6.07.

Data Presentation Procedures

PROC PRINT has undergone significant performance changes in Release 6.07. CPU times have been reduced between 30% and 40% relative to Release 6.06. Performance relative to Release 5.18 depends on the type of data being printed. Integer values can print 25% to 40% faster than in Release 5.18. Similarly, character values can print 20% to 30% faster than in Release 5.18. The improvement/degradation of printing numbers with decimal points depends on the number of decimal places, number of observations, and number of variables. For small problems performance of Release 6.07 is even with Release 5.18. A problem with 30 variables and 600 observations in Release 6.07 is 80% slower than in Release 5.18. Release 6.07 executes in 1.75 seconds while Release 5.18 executes in .96 seconds.

PROC TABULATE for large problems is significantly faster in Release 6.07 than in Release 5.18. Programs contributed by one large SAS user show several TABULATE steps running 40% to 50% faster than in Release 5.18.

User-Supplied Teststream Results

Tables 4 and **5** show performance gains in Release 6.07 on user-supplied test programs. Multiple appearances indicate different job streams.

Leigh Ihnen
Manager, Numerical Architecture and Performance
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513

TABLE 1

Release 5.18 SMF DATA									
PROCEDURE CPU utilization in seconds and FREQuence of invocation									
PROC	# of Sites	CPU	Pcnt. CPU	Cum. Pcnt	PROC	# of Sites	Freq	Pcnt. Freq	Cum. Pcnt
SASDATA	9	2200525	66.89	66.8	SASDATA	9	1444927	50.60	50.6
COPY	9	284288	8.64	75.5	SORT	9	578848	20.27	70.8
SORT	9	200820	6.10	81.6	PRINT	9	145262	5.09	75.9
LOGIST	5	101635	3.09	84.7	APPEND	8	130011	4.55	80.5
FREQ	9	80670	2.45	87.1	DELETE	9	84019	2.94	83.4
SUMMARY	9	56237	1.71	88.8	FORMAT	9	58251	2.04	85.5
MEANS	9	49883	1.52	90.4	FREQ	9	54906	1.92	87.4
GLM	7	32997	1.00	91.4	REG	8	54822	1.92	89.3
MATRIX	4	30985	0.94	92.3	MEANS	9	42711	1.50	90.8
PRINT	9	30702	0.93	93.2	DATASET	9	34108	1.19	92.0
TABULAT	9	24935	0.76	94.0	PRINTTO	8	21409	0.75	92.7
SYSNLIN	1	20236	0.62	94.6	SUMMARY	9	21303	0.75	93.5
DISCRIM	4	18059	0.55	95.2	DISPLAY	7	19153	0.67	94.2
FORMAT	9	17709	0.54	95.7	UNIVARI	9	14013	0.49	94.6
TABLES	3	17544	0.53	96.2	PLOT	9	12887	0.45	95.1
APPEND	8	12919	0.39	96.6	GLM	7	12120	0.42	95.5
UNIVARI	9	11509	0.35	97.0	MATRIX	4	11602	0.41	95.9
REG	8	10529	0.32	97.3	SYSREG	1	11435	0.40	96.3
DELETE	9	7345	0.22	97.5	CONTENT	9	10213	0.36	96.7
GPLOT	7	6108	0.19	97.7	LP	2	8942	0.31	97.0
CANDISC	2	5156	0.16	97.9	GPLOT	7	8157	0.29	97.3
IML	2	4781	0.15	98.0	TABULAT	9	7951	0.28	97.6
CHART	9	4346	0.13	98.1	FSEDIT	7	6694	0.23	97.8
PHGLM	4	4216	0.13	98.3	TRANSPO	8	6001	0.21	98.0
SESUDAA	1	3483	0.11	98.4	LOGIST	5	4654	0.16	98.2
TRANSPO	8	3143	0.10	98.5	COPY	9	3581	0.13	98.3
GCONTOU	4	2985	0.09	98.6	CHART	9	3498	0.12	98.4
FSEDIT	7	2943	0.09	98.7	CORR	6	2569	0.09	98.5
LP	2	2876	0.09	98.7	SYSNLIN	1	2182	0.08	98.6
CONTENT	9	2396	0.07	98.8	FSBROWS	6	2148	0.08	98.7
G3D	4	2262	0.07	98.9	RELEASE	6	2014	0.07	98.7

Table 2

PROC APPEND Release 6.07 vs. Release 5.18				
Obs.	Var.	CPU Savings	%	6.06 %
10	20	0.05	71%	0%
100	20	0.05	71%	0%
1,000	20	0.02	22%	22%
10,000	20	-0.18	-50%	25%
10	60	0.04	57%	0%
100	60	0.04	50%	20%
1,000	60	0.01	10%	25%
10,000	60	-0.19	-40%	33%
10	120	0.04	50%	0%
100	120	0.02	25%	0%
1,000	120	0.01	8%	33%
10,000	120	-0.22	-34%	39%
10	400	-0.06	-55%	6%
10,000	400	-0.44	-30%	15%
10	1000	-0.57	-335%	1%
1,000	1000	-0.57	-106%	7%
10	4000	-9.20	-2000%	-2%
100	4000	-9.17	-1730%	-2%

Table 3

PROC COPY Release 6.07 vs. Release 5.18					
Members	Obs.	Var.	CPU Savings	%	6.06 %
5	10,000	10	0.30	41%	63%
10	100,000	10	5.02	39%	65%
10	100	10	0.04	33%	11%
10	5,000	100	0.62	30%	60%
5	5,000	50	0.19	29%	59%
5	1,000	50	0.04	21%	44%
5	10	100	0.02	17%	0%
10	1,000	1000	0.45	12%	11%
1000	100	10	3.25	10%	-22%
5	1,000	500	0.05	5%	27%
10	100	100	0.00	0%	17%
100	100	100	-0.20	-1%	3%
50	100	100	-0.19	-10%	17%
100	100	50	-0.20	-15%	4%
330	100	500	-4.10	-21%	9%
10	10	500	-0.14	-33%	3%
5	1,000	4000	-2.19	-39%	-12%
50	10	1000	-1.95	-57%	-2%
100	10	100	-1.11	-74%	-16%
50	10	4000	-15.47	-120%	-44%

Table 4

User teststream summary Release 6.07 vs. Release 5.18 CPU times					
Proc	Freq	518	607	Diff.	% Diff.
CHART	1	0.23	0.10	.13	56%
FREQ	9	10.40	6.18	4.22	40%
MEANS	11	12.04	8.15	3.89	32%
PRINT	7	0.33	0.35	-0.02	-6%
SORT	54	49.17	36.35	12.82	26%
TABULATE	16	39.09	21.69	17.40	44%
PRINT	5	1.08	0.85	0.23	21%
SORT	23	10.02	6.53	3.49	34%
FREQ	1	0.89	0.46	0.43	48%
PRINT	3	0.60	0.47	0.13	21%
PRINTTO	6	0.20	0.09	0.11	55%
SORT	7	8.97	6.25	2.72	30%
SUMMARY	1	6.00	0.32	5.68	94%

Table 5

User teststream individual steps Release 6.07 vs. Release 5.18 CPU times				
Proc	518	607(o)	Diff.	% Diff.
FREQ	3.13	1.78	1.35	43%
FREQ	03.13	1.79	1.34	42%
FREQ	2.98	1.74	1.24	41%
MEANS	4.37	2.83	1.54	35%
MEANS	6.92	4.82	2.10	30%
SORT	3.70	2.67	1.03	27%
SORT	7.47	5.42	2.05	27%
SORT	20.72	15.25	5.47	26%
SORT	3.88	2.87	1.01	26%
TABULATE	6.58	3.30	3.28	49%
TABULATE	10.12	5.69	4.43	43%
TABULATE	10.20	5.76	4.44	43%
TABULATE	10.21	5.77	4.44	43%
PRINT	0.82	0.69	0.13	15%
SORT	1.01	0.56	0.45	44%
SORT	0.96	0.56	0.40	41%
SORT	0.93	0.59	0.34	36%
SORT	0.84	0.51	0.33	39%
SORT	0.86	0.54	0.32	37%
SORT	0.71	0.50	0.21	29%
SORT	0.76	0.55	0.21	27%
SORT	0.26	0.10	0.16	61%
SORT	2.86	2.11	0.75	26%
SORT	2.47	1.74	0.73	29%
SORT	2.58	1.85	0.73	28%
SUMMARY	6.00	0.32	5.68	94%