Comparison of VMSORT®, SyncSort™, PLSORT™, and SORTT under VM/CMS Using Release 5.18 of the SAS® System

Kirsty D. Nunez, Tulane University

Abstract

The SAS Institute supplies sort routine SORTT and sort interface routines for SyncSort™, VMSORT®, PLSORT™, and other sort utilities as part of Release 5.18 of the SAS® System for VM/CMS. As Tulane Computing Services licenses both SyncSort and VMSORT, it was necessary to decide which sort utility to use as the default with the SAS System under VM/CMS in Tulane’s environment. Using Release 5.18 of the SAS System installed under VM/CMS, four sort utilities were compared: VMSORT, SyncSort, PLSORT, and SORTT. Data sets containing five integer variables per case and 100, 1000, 5000, 10000, 50000, 100000, or 250000 records were sorted. The parameters examined include: Virtual time (VTIME), total time (TTIME), and Start I/O’s for the SAS job execution; CPU time and main memory used for PROC SORT; and CPU time and main memory used for PROC PRINT. The results for PROC SORT CPU time show that SORTT is slower than VMSORT, SyncSort, and PLSORT. There is no statistically significant difference between VMSORT, SyncSort, and PLSORT. The results for main memory used for PROC SORT show that VMSORT and PLSORT address memory as required up to a data set size of approximately 10000 records, and then they access approximately 1300 kilobytes of memory. SORTT and SyncSort routinely access 1300 kilobytes of memory regardless of sort size.

Introduction

The SAS Institute supplies the sort routine SORTT as part of Release 5.18 of the SAS System for VM/CMS. The Institute also supplies the standard E15 (input) and E35 (output) sort interface routines for SyncSort, VMSORT, PLSORT, and DFSORT. These sort products can be used with the VM/CMS SAS System by setting the SORTLIB and SORTPGM options appropriately during installation.

The installation Instructions for Release 5.18 of the SAS System for VM/CMS, state

"We can make some general recommendations about the relative merits of these sorts, but you should evaluate the sorts and decide which performs best in your environment. If your installation leases SyncSort, VMSORT, PLSORT, DFO SORT, or CASORT, we recommend that you use it with the SAS System for large sorts. Otherwise, we suggest SORTT, an in-memory sort routine that is efficient for data sets with fewer than 2000 observations."

As Tulane Computing Services licenses both SyncSort and VMSORT, and users typically sort data sets with greater than 2000 observations, it was necessary to decide which sort utility to use as the default with the SAS System under VM/CMS on Tulane’s IBM 3081 XA. To allow comparison and presentation of comparative results at SAS User’s Group International in April, 1989, Phase Linear loaned PLSORT to Tulane Computing Services for a 30-day period.

Versions numbers of the software products currently licensed by Tulane University are as follows: VM/SP HPO, Version 4.2; SAS software, Version 5.18; SyncSort, Version 6.1 D; VMSORT, Version 2.1; and PLSORT VMX, Release 8.1D.

Methodology

Using Release 5.18 of the SAS System for VM/CMS, four sort utilities were compared: VMSORT, SyncSort, PLSORT, and SORTT. As different sort utilities may use different algorithms to sort small, medium, and large data files, seven different data file sizes were used. Data sets containing five integer variables per case and 100, 1000, 5000, 10000, 50000, 100000, or 250000 records were sorted.

Data Set Size

<table>
<thead>
<tr>
<th>Records</th>
<th>Blocks</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>2</td>
<td>8192</td>
</tr>
<tr>
<td>1000</td>
<td>20</td>
<td>81920</td>
</tr>
<tr>
<td>5000</td>
<td>98</td>
<td>401408</td>
</tr>
<tr>
<td>10000</td>
<td>190</td>
<td>804816</td>
</tr>
<tr>
<td>50000</td>
<td>977</td>
<td>4001792</td>
</tr>
<tr>
<td>100000</td>
<td>1954</td>
<td>8003584</td>
</tr>
<tr>
<td>250000</td>
<td>4883</td>
<td>2000768</td>
</tr>
</tbody>
</table>

The data sets contained only numeric variables. Data sets were created using a random number generator written in IBM® VS FORTRAN (FORTRAN 77). Each data set had a logical record length of 80 and a fixed record format. Data sets were sorted using PROC SORT according to the values of five variables. No PROC SORT options were employed.

A VM/CMS account with a 1-cylinder 191 disk was created to conduct the study. Data files containing 100, 1000, 5000, 10000, 50000, and 100000 records were sorted in both a 50-cylinder temporary disk area and a 100-cylinder temporary disk area. Data files containing 250000 records were sorted in a 100-cylinder temporary disk only, as these were too big for a 50-cylinder temporary disk. For each data file size, one SAS job was run for each sort utility, so that the results of each sort would be independent. In order that memory usage for the previous SAS job not contaminate the results, it was necessary to log off and then log back on to VM/CMS between..."
each SAS job execution. A new data file was generated for each job execution. All jobs were executed in a two-megabyte virtual machine.

**SAS Code**

The SAS code used for all sort utilities for each of the seven data set sizes is presented below:

```sas
CMS FILEDEF ddname DISK FILE
   filename filetype filemode;
PROC OPTIONS;
DATA TEST;
   INFILE ddname;
   INPUT V W X Y Z;
PROC SORT;
   BY V W X Y Z;
PROC PRINT;
```

All sorts were performed using PROC SORT. PROC SORTIT, PROC SYNSORT, and PROC PLSORT were not examined in this study.

**Parameters Compared**

The parameters examined include: Virtual time (VTIME), total time (TTIME), and Start I/O's for the SAS job execution; CPU time and main memory used for PROC SORT; and CPU time and main memory used for PROC PRINT. The connect time ("wall clock time") for each job execution was not examined, as this is dependent on VM/CMS system load. The CPU time and main memory used for PROC PRINT were included in order to study any effect on SAS performance caused by the sort utility being tested.

**System Load**

In order to study the sort utilities under a typical system load, all SAS jobs were executed during the hours of 8:30 AM and 5 PM on workdays. SAS jobs using the competing sort utilities were alternated.

**Results**

**SAS Job Execution: Start I/Os**

The Start I/O data do not show differences between the four sort utilities. Graph I depicts Start I/O's for all data sets sorted in 50- and 100-cylinder temporary disk areas.

A main effects analysis of variance was performed on the Start I/O data using PROC GLM. The predictor variables were: sort utility (SORTIT, VMSORT, SyncSort, or PLSORT), data set size (100, 1000, 5000, 10000, 50000, 100000, or 250000), and size of temporary disk area (50 or 100 cylinders). The analysis revealed a significant effect of data set size $F(6,41) = 27.50, p = 0.0001$ and sort utility $F(3,41) = 23.04, p = 0.0001$. The effect of the size of the temporary disk area was not statistically significant. The main effects model, including sort utility, data set size, and size of temporary disk area, was also significant $F(10,41) = 18.92, p = 0.0001$.

**SAS Job Execution: Virtual Time and Total Time**

The total time and virtual time data show differences between the four sort utilities. Figures are not provided for virtual time and total time, as CPU time for PROC SORT more accurately reflects the effect of each sort utility. PROC SORT CPU time is a component of both virtual time and total time.

A main effects analysis of variance was performed on the total time data using PROC GLM. The predictor variables were: sort utility (SORTIT, VMSORT, SyncSort, or PLSORT), data set size (100, 1000, 5000, 10000, 50000, 100000, or 250000), and size of temporary disk area (50 or 100 cylinders). The analysis revealed a significant effect of data set size $F(6,41) = 250.05, p = 0.0001$ and sort utility $F(3,41) = 4.13, p = 0.0119$. The effect of the size of temporary disk area was not statistically significant. The main effects model, including sort utility, data set size, and size of temporary disk area, was also significant $F(10,41) = 151.33, p = 0.0001$.

A main effects analysis of variance was performed on the virtual time data using PROC GLM. The predictor variables were: sort utility (SORTIT, VMSORT, SyncSort, or PLSORT), data set size (100, 1000, 5000, 10000, 50000, 100000, or 250000), and size of temporary disk area (50 or 100 cylinders). The analysis revealed a significant effect of data set size $F(6,41) = 245.25, p = 0.0001$ and sort utility $F(3,41) = 4.16, p = 0.0116$. The effect of the size of the temporary disk area was not statistically significant. The main effects model, including sort utility, data set size, and size of temporary disk area, was also significant $F(10,41) = 148.40, p = 0.0001$.

**PROC SORT: CPU Time**

The PROC SORT results do show differences in the efficiency of the four sort utilities. The most significant differences appear in the use of CPU time. In both 50-cylinder and 100-cylinder temporary disk areas, SORTIT is not as efficient as VMSORT, PLSORT, and SyncSort for data sets of 1000 observations or more. In terms of CPU time used, PLSORT is the most efficient sort utility, followed by...
SyncSort and then VMSORT. As the size of the data set increases, the differences in CPU utilization efficiency increases.

Graph 2 depicts CPU time used for PROC SORT for all data set sizes in a 50-cylinder temporary disk. Graph 3 depicts CPU time used for PROC SORT for all data set sizes in a 100-cylinder temporary disk.

A main effects analysis of variance was performed on the CPU time data using PROC GLM. The predictor variables were sort utility (SORTT, VMSORT, SyncSort, or PLSORT), data set size (100, 1000, 5000, 10000, 50000, 100000, or 250000), and size of temporary disk area (50 or 100 cylinders). The analysis revealed a significant effect of sort utility $F(3,41) = 4.44, p = 0.0086$ and of data set size $F(6,41) = 10.91, p = 0.0001$. The effect of size of temporary disk area was not statistically significant. The main effects model, including sort utility, data set size, and size of temporary disk, was also significant $F(10,41) = 7.88, p = 0.0001$. The Newman-Keuls multiple comparisons procedure was employed to determine significant differences between the means for CPU time for each of the sort utilities. The procedure revealed that SORTT is significantly different from VMSORT, SyncSort, and PLSORT at $p = 0.05$. However, there was no significant difference between VMSORT, SyncSort, and PLSORT.

PROC SORT: Memory

The four sort utilities differ in their use of memory for PROC SORT. SORTT and SyncSort access approximately 1300 kilobytes of memory for every sort regardless of data set size. VMSORT and PLSORT access only the amount of memory necessary to complete the sort. As data set size increases, the amount of memory accessed increases. However, with a data set of 50000 records, both VMSORT and PLSORT begin to access approximately 1300 kilobytes of memory.

Graph 4 depicts memory in kilobytes used for PROC SORT for all data set sizes in a 50-cylinder temporary disk. Graph 5 depicts memory in kilobytes used for PROC SORT for all data set sizes in a 100-cylinder temporary disk.

A main effects analysis of variance was performed on the memory usage data using PROC GLM. The predictor variables were sort utility (SORTT, VMSORT, SyncSort, or PLSORT), data set size (100, 1000, 5000, 10000, 50000, 100000, or 250000), and size of temporary disk area (50 or 100 cylinders). The analysis revealed a significant effect of sort utility $F(3,41) = 16.02, p = 0.0001$ and of data set size $F(6,41) = 6.31, p = 0.0001$. The effect of size of temporary disk area was not statistically significant. The main effects model, including sort utility, data set size, and size of temporary disk area, was statistically significant $F(10,41) = 8.61, p = 0.0001$. The Newman-Keuls multiple comparisons procedure was employed to determine significant differences between the means for memory for each of the sort utilities. The procedure revealed that SORTT and SyncSort are significantly different from VMSORT and PLSORT at $p = 0.05$. There was no statistically significant difference between SORTT and SyncSort, nor between VMSORT and PLSORT.
The different sort utilities do not appear to affect the CPU or memory usage of PROC PRINT.

Outlying values were obtained for memory usage for SORTT when sorting a data set of 1000 records. The memory usage was approximately 800 kilobytes in both cases. When these aberrant values were discussed with the SAS Institute, they could give no reason why this occurred in Tulane's environment. However, they did execute several jobs using SORTT with 1000 records on their CMS operating system, and they were unable to replicate these values. They obtained memory values within the normal range for a two-megabyte virtual machine.

Discussion

SAS Job Execution

The Start I/O's data do not reflect differences in the four sort utilities. The total time and virtual time data do reflect differences in the four sort utilities. As PROC SORT CPU time is a component of both total time and virtual time, the differences are delineated below in a discussion of the PROC SORT CPU time results.

PROC SORT: CPU Time

SORTT is slower than VMSORT, SyncSort, and PLSORT. There is no statistical difference between VMSORT, SyncSort, and PLSORT. PLSORT uses less CPU time than SyncSort, which in turn uses less CPU time than VMSORT. There is an apparent trend in the graphs indicating that as sort size increases beyond 250000 records, the difference between VMSORT, SyncSort, and PLSORT could become significant.

As the Installation Instructions for VM/CMS SAS System, Release 5.18, indicate, SORTT is not efficient for data sets with more than 2000 observations. For data sets of less than 1000 observations, the graphs suggest that PLSORT and SyncSort use less CPU time than SORTT. VMSORT uses the most CPU time, however, the difference between SORTT and VMSORT for 1000 records does not appear to be appreciable.

PROC SORT: Memory

Apparently VMSORT and PLSORT address memory as required up to a data set size of approximately 10000 records (802816 bytes), and then they access approximately 1300 kilobytes of memory. SORTT and SyncSort routinely access 1300 kilobytes of memory regardless of sort size. For PLSORT, the drop in memory addressed for a 50000-records sort apparently reflects a change in algorithm used by the sort utility. At Tulane University, such memory differences for small sorts are not significant in terms of machine resources. However, at other sites such memory differences might be significant, and therefore sites doing small sorts might prefer to use VMSORT or PLSORT. SORTT does not appear to be as efficient as VMSORT or PLSORT.

The findings suggest that for sorts up to 100000 records, PLSORT uses less memory than the SORTT, SyncSort, and VMSORT. VMSORT follows PLSORT. SORTT and SyncSort are very similar in their memory usage. At a data set size of 250000 records, VMSORT uses the least memory. This suggests that for larger data files (20 megabytes or more) VMSORT might be the most efficient in terms of memory usage.

PROC PRINT

The data derived from PROC PRINT indicate that none of the sort utilities appear to affect overall SAS performance.

Concluding Comments

The cost to license each of the sort utilities was not examined in this study; however, this may be of importance to individuals making decisions about which sort utility to use in their environment.

The new SAS sort interfaces PROC SYNCSORT and PROC PLSORT were not examined in this study. However, both Syncsort and Phase Linear (PLSORT) purport to offer procedures that read and sort records more efficiently than PROCSORT.

It was decided at Tulane Computing Services that the differences between VMSORT, SyncSort, and PLSORT under a VM/CMS operating system were not significant enough to warrant changing from VMSORT to SyncSort or PLSORT.
Acknowledgments

I would like to thank my colleagues at Tulane Computing Services for their assistance in the preparation of this manuscript: Alice Conniff, Alison Hartman, and John Voigt.

The Author may be contacted at:

Kirsty D. Nunez
User Consultant
Tulane Computing Services
Tulane University
New Orleans, Louisiana 70118

BITNET: TACVKN@TCSVM
INTERNET: TACVKN@VM.TCS.TULANE.EDU