THE SAS® SYSTEM FOR PCs: SOME SPEED SURPRISES

Richard H. Browne, Texas Scottish Rite Hospital For Crippled Children

I. INTRODUCTION

Amongst PC SAS® users, no topic generates more intense interest than how to get the most speed from your PC. The reason is simple. While a PC provides us with many desirable features over a mainframe (e.g., instant access, autonomy, no additional cost for rerunning a program, etc.), PCs are just much slower than most mainframes. Also, when a boss is breathing down your neck for a result needed in thirty minutes, ANYTHING that could make that PC run faster would be considered a godsend.

Several SUGI speakers have addressed the issue of speed at past SUGI’s, some providing general programming tips suitable for mainframes or PC’s (1,2,3) and some specifically for PC’s (4,5,6,7). All of these have been excellent presentations and should be read by PC users. However, they have not addressed the effects of combinations of add-ons (e.g., EMS, disk caching, math co-processor) across a range of MHz values.

I would like to approach the problem from the point of view of a SAS/PC consultant who has been asked:

1. “How fast does SAS run on PC’s, ranging from a 4.77 MHz IBM/XT to a 25 MHz 80386 machine? How big a factor is MHz, really? Does a 50% increase in MHz increase SAS speed by 50%? Is a 20 MHz 80386 machine faster than a less-expensive 20 MHz 80286 machine? Does knowing the Norton Utilities® SI values of a computer tell you more than the MHz about the speed of SAS execution? How does SAS on a fast PC compare to a mainframe?”

2. “What would really help the execution time of SAS on my PC? Should I get EMS, disk caching, a math co­processor, etc.? Would anything hurt the performance?”

3. “What other programming tips can you tell me that would help speed things up?”

II. GOALS OF THIS WORKSHOP

The chief goal of this workshop is to help you to do benchmarking so that you can determine what will provide the best overall SAS performance on your present or future PC. These are no magic formulas, but having some idea of what to look for is usually better than relying on a salesman who thinks SAS is a Scandinavian airline.

The second goal was to establish some useful guidelines about microprocessor sizes (8088, 80286, and 80386), processor speeds (MHz), and add-ons (EMS, co-processor, disk caching, RAM disk) that would help you choose what machine and/or configuration is best for you. I did not seek out the minutiae of each machine (e.g., hard drive interleave factor, number of wait states, 256K or 1 Mbyte chips, etc.), since this is often more than the average user knows anyway. What most people want is a good ballpark figure on relative performance that is easily determined, rather than a value reliable to the third decimal place that takes a computer guru and several weeks to find out.

What I chose to do was correlate readily obtainable information on a machine (e.g., microprocessor size and MHz, and the Norton SI values) with performance on a SAS benchmark program. From those findings, you should be able to make some relatively informed decisions.

The third goal is to impart some SAS programming tips that may save you a lot of run time in the future. I have endeavored not to repeat tips presented at prior SUGI’s, but will not guarantee that it hasn’t been seen at SUGI before.

III. DEVELOPING A BENCHMARK PROGRAM

The game plan was to see how quickly SAS runs on a cross-section of PC’s configured in a variety of ways. What characteristics should a good benchmark program have?

1. It should use a representative cross-section of functions and activities that are common to most SAS applications. These include DATA set creation, manipulation, sorting, and computation.

2. Since SAS is so disk-intensive, the effect of heavy disk I/O needs to be fully assessed, with multiple files being used simultaneously.

3. Tests of specific capabilities should be clearly separated so that reliable comparisons can be made. Each PROC and DATA step should be designed to primarily assess one capability.

4. The benchmark should run long enough to allow for performance differences to be readily apparent. It should not run so long, though, that testing the benchmark in a variety of hardware/software configurations becomes a practical impossibility.

The benchmark program in this study is shown in Appendix 1. Does this benchmark program meet all of these criteria perfectly? No. The perfect benchmark program does not yet (and probably never will) exist. However, I believe this benchmark meets much of the criteria.

The first DATA step involves the creation of four data sets simultaneously, each consisting of about 4,000 records. This provides a test of disk I/O and file handling speed with four files. The CALL to RANUNI also involves computation.

The second DATA step involves the concatenation of four data sets. Again, we are accessing data from four data sets and creating a fifth. No computation is involved.

The next step is PROC SORT, which involves the creation of a temporary data set (unseen by us) and a final sorted version of...
DATA ALL that replaces the original DATA ALL. It is well-known that the SORT procedure is a heavy I/O user and has much internal overhead. Towey (7) has shown that EMS improves the speed of PROC SORT.

The last step (PROC SUMMARY) is primarily computational. The effect of a math co-processor is easily assessed with this step.

One advantage of this benchmark is that it is easily entered and run with all versions of SAS and requires no auxiliary files. Another advantage of this benchmark is that no output (other than the SAS Log) is produced. This eases housekeeping problems.

IV. RUNNING THE BENCHMARK

The chief goal was to run the benchmark on a broad cross-section of PC's, from an original IBM/XT to a high-end 80386 machine. We also wanted to run the benchmark on some minis and mainframes to see how close PC performance is coming to them. According to an article in Business Week (8), the following equivalences exist in speed:

<table>
<thead>
<tr>
<th>XT</th>
<th>PDP 11/70</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>VAX 11/786</td>
</tr>
<tr>
<td>8086</td>
<td>VAX 8600</td>
</tr>
<tr>
<td>80486</td>
<td>IBM 3090</td>
</tr>
</tbody>
</table>

A. Computers Used

To run on a variety of PC's, I had to enlist the help of several members of the Dallas Area SAS Users Group (DASUG). The PC's used (and their speeds) were:

- 8086: IBM/XT (4.77 MHz)
- 80286: IBM/AT (8 MHz)
- 80386: Compaq 386 (16 MHz)
- 80486: IBM PS/2 Model 80 (25 MHz).

The benchmark was also run on several mainframes and a mini. Those were:

- VAX 8350 (mini)
- IBM 3084
- IBM 3090-400
- IBM 3090-160E
- Amdahl 5870.

B. Add-On Options

Besides running the benchmark on a variety of PC's, we also wanted to see how performance varied in PC's with additional of hardware and software options. Those options were:

- EMS: EMS (Expanded Memory System) requires RAM memory above 1 Mbyte. EMS allows the SAS supervisor to be removed from the 640K, allowing more room for program components and data. However, swapping memory pages in and out of EMS exacts an overhead cost in time. EMS is enabled in CONFIG.SAS by use of the -EMS ALL option. For a more complete discussion of EMS, see pp 478-479 in the SAS Language Guide for Personal Computers.

  - Disk Caching: Disk caching cuts down on hard disk reads and writes, the slowest part of any PC. Caching is controlled by a software program that makes use of RAM memory above 1 Mbyte. This is not the same as -FILECACHE (SAS global option).

  - RAM Disk: Software program that sets aside a portion of RAM memory above 1 Mbyte that can be "formatted" into a pseudo hard disk. Can be used to speed handling of SASWORK files.

  - Co-processor: Chip that speeds math calculations. Can be disabled in CONFIG.SAS with $NONDP.

  - FILEBUFFERS: A SAS global option that is set through CONFIG.SAS when SAS is invoked. Similar in effect to Disk Caching, it works with the buffers set up through the BUFFERS feature in CONFIG.SYS. Default in PC-SAS is 5 filebuffers of 512 bytes each in conventional RAM memory. See pp 479-480 of the Language Guide for more details. Previous SUGI talk (7) indicated that increasing size would be beneficial.

The first three options require expanded (or extended) memory above 1 Mbyte. The co-processor is also a hardware item. Only the filebuffers can be manipulated on all computers since they use conventional memory.

Because most of the PC's belonged to other DASUG members, we were unable to look at every combination of these additions on all PC's, since some did not have the hardware. However, we were able to get a good estimates of the synergistic and antagonistic effects of these additions with what was available. The options that were available on those PC's were:

<table>
<thead>
<tr>
<th></th>
<th>EMS</th>
<th>Cache</th>
<th>Coproc</th>
<th>Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM/XT</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>IBM/AT</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Compaq 286</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CompuAdd 220</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Compaq 386</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>AMAX 386/4</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>NCI 386</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>PS/2 Model 80</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

C. Norton SI Values

There are two Norton SI values: Performance Index (PI) and Computation Index (CI). CI is a measure of processor speed alone, while PI combines the processor speed with the disk access speed. It would seem logical that PI would be the better indicator of computer performance for SAS since SAS is so disk-intensive. Since disk caching will affect the PI, the value of PI was
determined on each PC with and without disk caching, if disk caching was on the PC.

To run the benchmark on a PC, a batch file was prepared. That file ran the benchmark with each available add-on either turned on or off. For the IBM PS/2 Model 80, all were absent, so only one combination of add-ons was run, all in the Off configuration. Conversely, for the Compaq 286, there were 16 combinations of the add-ons, we also looked at the effect of having one combination of add-ons was run, all in the Off configuration. For each combination of the add-ons, we also looked at the effect of having no filebuffers, 5 of 512 bytes, 5 of 2048 bytes, and 5 of 8132 bytes.

V. BENCHMARK RESULTS

We need a definition of speed for SAS processing. In this report, I will divide the number of observations processed in a step by the number of seconds needed to do so, yielding an Observations Per Second (OPS) rate. The faster the machine, the higher we would expect the OPS rate to be in each step. In this benchmark, each step involves 16,000 observations. For the IBM/XT, speeds of 50 OPS are typically seen, while for an IBM PS/2 Model 80, the speed for concatenation was 457 OPS.

It is important to remember that the OPS values derived in this report pertain ONLY to the benchmark shown in Appendix 1. The OPS rating would stay roughly the same if you only added or deleted the number of observations processed in the benchmark used, but if the steps/procs were altered, the OPS ratings could change dramatically. Other programs will likely have different OPS rates, since the OPS rate depends on what functions are performed, how large the records are, etc. The OPS values shown here are for comparison purposes only, so that you can compare one machine to another on a common basis; the OPS values are NOT an absolute speed rating for these machines and add-ons!!

A. Repeatability

Each time a job is run, it is very likely to take a different amount of time, due to disk file movement and spacing. To estimate replication error, I used the differences in speeds between the different filebuffer sizes when filebuffers were turned OFF, all other settings being the same. The times for this should have all been the same, so their variability provided a good measure of reproducibility. Most differences were less than +/-3%, with a maximum of +/-6%.

Differences in speeds of less than +/-3% in the results below can be ascribed to uncontrollable error rather than true performance differences. Differences greater than +/-6% can be considered real differences in speed.

B. Filebuffer Sizes

The first question is whether having filebuffers turned ON has a beneficial effect on run time. We also want to see if it is beneficial to use filebuffers larger than the SAS-supplied default of 512 bytes (i.e., 2048 and 8132 bytes).

In tables not shown (available on request), having filebuffers turned ON increased the speed between 7% and 30% for the two DATA steps, with larger filebuffers helping by less than 5% in most instances. For PROC SUMMARY, filebuffers ON increased speed between 5% to 18%, with no apparent difference between the 3 filebuffer sizes.

PROC SORT, though, presented a much different picture. In most cases, speed increases between 15% to 40% were seen, with larger filebuffer sizes providing up to 5% increased speed compared to 512 bytes. On the Compaq 286 and the AMAX 386/4, though, there were also decreases in speed of 3% to 11% with filebuffers of 8132 bytes, compared to no filebuffers. In those few cases, EMS and a math coprocessor were in use, but that combination did not always result in deviant results. Why this happened is not clear.

These results suggest that we gain much from having filebuffers turned ON, but little from using a filebuffer size other than 512. For that reason, and to simplify the presentation, all of the remaining results will be based on having 5 filebuffers of 512 bytes turned on (the SAS-supplied default).

C. Value of Add-Ons

To assess the value of add-ons (math co-processor, EMS, disk caching, and RAM disk), I looked at the percent change in speed between having the item ON versus OFF. This type of comparison was repeated across all other available combinations of add-ons. While these add-ons act synergistically and antagonistically, the results are presented for each add-on separately to show the range of effects for each add-on. The joint effects on speed will be discussed in Section D below.

It is important to note that the results seen here used a broad variety of disk caching programs and EMS drivers. It is quite likely that a different set of programs and/or drivers would have yielded slightly different results. However, I believe that the results in this study still provide a useful set of ballpark figures for performance.

Math Co-Processor: The range of results is shown in Table 1. Clearly, the first DATA step and PROC SUMMARY, which make much use of computation, are either unaffected or show significant increases in speed (up to 41%). On the other hand, those steps which do not make use of the co-processor may show increases in run time of up to 30%. The conditions that caused this for SORT were not those that caused it for concatenation, so there is no clear pattern.

Having a math co-processor is not necessarily a benign add-on. If your work is primarily non-arithmetic.

EMS: Table 2 shows that having EMS turned ON causes very large decreases in speed for all but PROC SORT, with decreases ranging as high as 51%. Only PROC SORT benefits from EMS (maximum speed increase of 36%), but that benefit is not seen on every PC. The benefit of EMS to sorting may be real, but is not highly predictable.

If you do a lot of heavy sorting, you may want to consider having EMS, but you need to benchmark your computer to be sure it is beneficial.

Disk Caching: Table 3 shows that disk caching can be, at worst, of little value and, at best, of great value. Increases between 26% and 48% are seen, while no measurable decreases in speed are seen. If used, disk caching will also decrease the time it takes to bring up SAS on your PC by at least 25%. Disk caching software is quite diverse, though, and the results with your software could be better or worse than these results.
Disk caching is of no harm and potentially of great value in all phases of SAS.

RAM Disk: Table 4 shows that RAM disk was only tested on a Compaq 286, where the SASWORK files were placed on a RAM disk. Like disk caching, a RAM disk is, at worst, of little value and, at best, to increase speed by up to 23% for sorting.

RAM disk is of no harm and potentially of good value, especially for sorting. However, for very large data sets you will need a substantial investment in expanded or extended memory to hold all of the temporary files in SASWORK. You can run out of space on a RAM disk just like any other disk.

D. Comparison of PC's

Side By Side Comparisons: Most of us would like to know how the speeds of actual PC's compare to one another, on the average. As an aid in comparing PC's, I decided use the IBM/XT speed as the baseline for each step. For a given step, I computed the mean speed for each PC, using the non-EMS values. (Using the non-EMS values allowed each computer to be compared and removed some of the variability in a PC's speeds.) For each step, I divided the average speed for each PC by the XT's average speed. The average speeds and relative speeds are shown in Table 5. The relative speeds can be thought of as how much faster a PC is than an XT. These values may then be compared to each other by ratio to see how much faster one machine is than another.

For example, comparing a Compaq 286 (12.5 MHz) to a CompuAdd 220 (20.0 MHz), we see that the CompuAdd runs 49% faster (5.5/3.7 = 1.49) in the DATA step, 55% faster in concatenation, 69% faster in PROC SORT, and 61% faster in PROC SUMMARY. Since 20 MHz is 60% faster than 12.5 MHz, this is a plausible result. We again see that the Compaq 386 at 16 MHz is equal or faster than the 20 MHz 80286 and 80386 machines listed. The CompuAdd 220 (a 80286 machine) is also just as fast as some more expensive 80386 machines.

To see if the relationship between OPS and MHz can be modeled statistically, a plot (not shown) was made of speed versus MHz, which suggested an increasing, slightly non-linear relationship for each step. After limiting ourselves to the non-EMS results, a regression model of: Speed = a*(MHz)^b was run using PROC NLIN. The results were:

Create DATA sets: Speed = 10.046*(MHz)^1.052 (R^2= .87)
Concatenate sets: Speed = 7.835*(MHz)^1.216 (R^2= .84)
Sort DATA sets: Speed = 7.787*(MHz)^1.173 (R^2= .83)
PROC SUMMARY: Speed = 6.151*(MHz)^1.241 (R^2= .99)

This gives strong evidence that the effect of MHz on SAS speed is not a linear function of MHz and suggests a slightly greater than linear effect on speed from increasing the MHz. However, I would NOT recommend using these formulae to project speeds for 33 MHz and higher machines, since these formulae have shown as much as +/-10% error in prediction of actual OPS values, even with such high R^2 values.

Buying an incremental increase in MHz speed will likely (though not absolutely) deliver an equal or greater proportionate increase in speed for SAS jobs. For a given MHz, it is not apparent that an 80386 has a greater speed than an 80286 PC. However, the PC's at 12.5 and 16 MHz can be comparable to 20 MHz PC's.

Effect of Add-Ons On Speed:

In Figures 1 through 4, the speeds (in OPS) for each step are plotted showing the combinations of add-ons tested on each PC. Where EMS is OFF, the number plotted is from 0 to 3; where EMS is ON, the number plotted is 4 to 7.

These figures show that the 20 MHz PC's (letters E,F,G) are largely comparable, though E is an 80286 PC and the others are 80386 PC's. Also the Compaq 386 at 16 MHz is equal in performance to some of the 20 MHz PC's.

If we look at the graphs again, we see the relative effects of the add-ons. In creating four DATA sets (Figure 1), the deleterious effect of EMS by itself and in combination with other add-ons is obvious. The same is true for concatenation (Figure 2) and PROC SUMMARY (Figure 4). In Figure 3, we see some evidence of the relative value of EMS for sorting. However, the apparent speed increases are usually not great. The higher positions of symbols (1) and (3) on Figures 1 and 4 show the value of a co-processor, if EMS is Off and much computation is done.

It is not apparent that EMS is generally of great value, even in sorting. A co-processor is of real value, if EMS is not on.

Relationship of MHz to Norton SI Values: Looking at all unique combinations of MHz and PI in this study, a regression line was computed. The results was: PI = -4.15 + 1.00*MHz with an R^2 = 0.92. A graph of these (not shown) displays a very clear linear relationship, with the 16 MHz Compaq 386 having PI values slightly below the regression line; all others are right on the line. Looking at all unique combinations of MHz and CI in this study, a regression line was computed. The results was: CI = -8.6 + 1.47*MHz with an R^2 = 0.91. A graph of these (not shown) displays a linear relationship, with the 16 MHz Compaq 386 having a CI much below the regression line.

Norton PI and CI appear to be very closely associated with MHz. For that reason, one can compare performance using either MHz, PI, or CI. Because MHz is more readily known to the user, it is preferable to use MHz in performance analysis.

E. Mainframe Results

For the IBM and Amdahl mainframes, execution time was under 3 seconds for all four steps, so the speed would be greater than 16,000 OPS. For the VAX 6350, the speed varied from 642 OPS for creating the DATA sets to 312 OPS for PROC SORT. This is not greatly different from the 25 MHz PS/2 Model 80, especially for sorting.

However, CPU time is irrelevant if it takes several minutes to get a job slot on the mainframe or to have the output available for viewing. In one mainframe shop, the computer took only 2.15 seconds of CPU time, but took 6.3 minutes of wall clock time to complete. If you are in a shop like that, a 20 MHz PC may be able to outperform an IBM mainframe in terms of turnaround time.

Benchmark your mainframe's turnaround time and compare to the times for the PC's. You will then have a good idea what speed of PC would be a satisfactory replacement, if any.
VI. SOME PROGRAMMING TIPS

A. Use of BY Versus CLASS with PROC MEANS

With Version 6.03, you can use a CLASS statement with PROC MEANS, as opposed to having to sort and then use a BY statement. But how much faster is one than the other? Using BYTEST.SAS (see Appendix 2), you can compare the results for a variety of group sizes and number of groups.

PROC MEANS using the CLASS statement runs no faster with sorted than unsorted data. For fastest turnaround, if the data is already sorted, use the BY statement. If the data is unsorted, use the CLASS statement, rather than sort and then use the BY statement. If data is grouped, though not in sort-order, use BY (groupvar) UNSORTED, rather than CLASS. If you have created an output file of means, you can sort it BY (groupvar) much faster than the original data set.

B. Use of Raw Code, Rather Than PROC's

In older versions of PC/SAS, you could often write raw SAS code that would do a job faster than a PROC (e.g., compute the mean of the data with a SUM statement in a DATA step, as opposed to PROC MEANS). This is less true today, but if you want to get an order statistic (e.g., a median), raw code is much faster than PROC UNIVARIATE. Using the example in Appendix 3, the PROC SORT and DATA _NULL_ took 53 seconds on a 20 MHz machine; PROC UNIVARIATE took 2.13 minutes. It does not reduce run time to ask for only one statistic from PROC UNIVARIATE.

If you are getting order statistics on a large data set, consider using raw code, along with PROC SORT.

VII. CONCLUSIONS

Some speed surprises we have seen are:

1. For a given MHz rating, it is not apparent that an 80386 is superior to an 80286 PC.
2. One can not easily classify the performance of PC's on the basis of their stated MHz rating. It is possible that a PC with somewhat slower MHz may have speed equal to that for a higher MHz PC. In general, a certain percent increase in MHz will usually yield and equal or greater percent increase in OPS speed, though not absolutely. Each machine needs to be tested.
3. The MHz of the PC will tell you as much as the Norton SI values about potential SAS performance.
4. Every type of DATA step or PROC has notably different OPS rates. Therefore, any benchmarking needs to include several types of DATA steps and PROCs. If you change the benchmark program (other than add observations), the OPS ratings will obviously be different.
5. EMS is of little value except for sorting, and then not always. You need to test EMS on your PC and then decide if you should keep it.
6. A math co-processor may hurt performance, if the jobs do not include a lot of calculation.
7. Disk caching is the best bet if you can only get an extra 1 Mbyte of memory. If you can afford 4 Mbytes of extra memory, you may want to using RAM disks for the SASWORK files.
8. Lower end minis may be no faster than 25 MHz PC's. Major mainframes are 40 or more times faster than a 25 MHz PC, but the actual turnaround time can be the same or less. This may influence whether a switch to PC/SAS is justifiable.

ACKNOWLEDGEMENTS

I wish to acknowledge the invaluable assistance of the following people for the use of their computers: Joe Casey (Houston General Insurance), Joella Forester (Coopers & Lybrand), Clarence Jackson (City of Dallas), Thomas Perez (J.C. Penney Financial Services), Gail Ridgley (Halliburton), Robbie Severance (Burlington Northern Railroad), James Skidmore (J.C. Penney), Willy Waars (2W Systems), and Barry Wolfe (TEAMS, Inc.).

SAS is a registered trademark of SAS Institute Inc., Cary, NC, USA. IBM and AT are registered trademarks and PC/XT and XT are trademarks of International Business Machines Corporation. Norton Utilities is a trademark of Peter Norton Computing Incorporated. Compaq is a trademark of Compaq Computers of Houston, Texas.

FOR FURTHER INFORMATION

A printed copy of the original data base can be obtained from:

Richard H. Browne, Ph.D.
Research Department
Texas Scottish Rite Hospital
2222 Welborn
Dallas, Texas 75219

REFERENCES


APPENDIX 1

******************************************************************************
** PGM: BENCHRUN.SAS
** REV: 18-AUG-89
** FOR: BENCHMARKING PC-SAS
******************************************************************************
DATA A B C D; LENGTH GROUP X 3
SEED=12345678; DROP SEED;
00 1=1 TO 16000; DROP I;
CALL RANUNI(SEED,X);
GROUP=INT((I-1)/1000);
END;
RUN;
DATA ALL; SET A B C D; RUN;
PROC SORT DATA=ALL; BY X; RUN;
PROC SUMMARY DATA=ALL; CLASS GROUP; VAR X;
OUTPUT OUT=SUMMARY N=NVAL MEAN=AVG MIN=MINVAL MAX=MAXVAL;
RUN;

APPENDIX 2

******************************************************************************
** PGM: BYTEST.SAS
** REV: 19-DEC-89
** FOR: BENCHMARKING PC-SAS FOR VALUE OF BY V. CLASS
******************************************************************************
%MACRO BYTEST(NOBS,NGRP);
OPTIONS NONOTES NOSOURCE;
DATA ALL; LENGTH GROUP X 3
SEED=12345678; DROP SEED;
DO I=1 TO &NOBS;
   CALL RANUNI(SEED,X);
   GROUP=INT(&NGRP*X) +1; DROP X;
   OUTPUT;
END;
RUN;
PROC PRINTTO PRINT=NUL; RUN;
OPTIONS NOTES;
XPUT **-- USING CLASS OPTION ON UNSORTED DATA ---***;
PROC MEANS DATA=ALL; VAR I; CLASS GROUP; RUN;
XPUT --- SORT DATA SET;
PROC SORT DATA=ALL; BY GROUP; RUN;
XPUT ***** USING BY OPTION ON SORTED DATA ---***;
PROC MEANS DATA=ALL; VAR I; BY GROUP; RUN;
XPUT ***** USING CLASS OPTION ON SORTED DATA ---***;
PROC MEANS DATA=ALL; VAR I; CLASS GROUP; RUN;
OPTIONS SOURCE;

1067
### TABLE 1

Percent Change in Observations Per Second (OPS) With MATH COPROCESSOR

\[
\% = 100 \times \frac{(\text{OPS w/ COPROC}) - (\text{OPS w/o COPROC})}{(\text{OPS w/o COPROC})}
\]

Using 5 FILEBUFFERS of 512 Bytes

NEGATIVE % Implies That Using Math Coprocessor Takes LONGER

<table>
<thead>
<tr>
<th>COMPUTER</th>
<th>EMS</th>
<th>CACHING</th>
<th>RAM</th>
<th>DISK</th>
<th>CREATE</th>
<th>DATA</th>
<th>CONCATENATE</th>
<th>PROC</th>
<th>PROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>36.5</td>
<td></td>
<td>1.8</td>
<td>-9.6</td>
<td>21.7</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>40.6</td>
<td></td>
<td>2.1</td>
<td>-6.2</td>
<td>20.8</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>38.0</td>
<td></td>
<td>4.1</td>
<td>-11.3</td>
<td>21.6</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>41.5</td>
<td></td>
<td>2.1</td>
<td>-6.3</td>
<td>23.5</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>27.4</td>
<td></td>
<td>1.4</td>
<td>-16.9</td>
<td>-2.1</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>28.6</td>
<td></td>
<td>1.5</td>
<td>-21.7</td>
<td>-1.3</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>3.0</td>
<td></td>
<td>-26.6</td>
<td>0.0</td>
<td>19.7</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>2.3</td>
<td></td>
<td>-30.0</td>
<td>0.0</td>
<td>21.3</td>
</tr>
<tr>
<td>COMPUADD-286</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>16.1</td>
<td></td>
<td>2.0</td>
<td>-1.0</td>
<td>9.1</td>
</tr>
<tr>
<td>COMPUADD-286</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>15.0</td>
<td></td>
<td>2.2</td>
<td>0.0</td>
<td>11.1</td>
</tr>
<tr>
<td>COMPUADD-286</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>9.3</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>7.6</td>
</tr>
<tr>
<td>COMPUADD-286</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>9.9</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>7.5</td>
</tr>
</tbody>
</table>

### TABLE 2

Percent Change In Observations Per Second (OPS) With EMS

\[
\% = 100 \times \frac{(\text{OPS w/ EMS}) - (\text{OPS w/o EMS})}{(\text{OPS w/o EMS})}
\]

Using 5 FILEBUFFERS of 512 Bytes

NEGATIVE % Change Implies Using EMS Takes LONGER

<table>
<thead>
<tr>
<th>COMPUTER</th>
<th>MATH</th>
<th>COPROCESSOR</th>
<th>DISK</th>
<th>CACHING</th>
<th>RAM</th>
<th>ON?</th>
<th>DATA</th>
<th>ON?</th>
<th>CREATE</th>
<th>SETS</th>
<th>CONCATENATE</th>
<th>PROC</th>
<th>PROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM AT</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-23.2</td>
<td>N</td>
<td>-29.3</td>
<td>2.1</td>
<td>-17.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-24.5</td>
<td>N</td>
<td>-25.3</td>
<td>21.8</td>
<td>-6.2</td>
<td>18.4</td>
<td>-8.2</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-25.3</td>
<td>N</td>
<td>-27.8</td>
<td>10.9</td>
<td>-9.8</td>
<td>10.9</td>
<td>-9.8</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-8.0</td>
<td>N</td>
<td>0.7</td>
<td>21.8</td>
<td>9.6</td>
<td>-6.3</td>
<td>-9.9</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-8.4</td>
<td>N</td>
<td>2.1</td>
<td>9.6</td>
<td>-6.3</td>
<td>9.6</td>
<td>-9.9</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-29.5</td>
<td>N</td>
<td>-25.7</td>
<td>7.7</td>
<td>-26.2</td>
<td>-26.2</td>
<td>-26.2</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-31.7</td>
<td>N</td>
<td>-28.2</td>
<td>-7.4</td>
<td>-26.8</td>
<td>-26.8</td>
<td>-26.8</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-31.3</td>
<td>N</td>
<td>-29.0</td>
<td>36.4</td>
<td>-11.2</td>
<td>-11.2</td>
<td>-11.2</td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-33.8</td>
<td>N</td>
<td>-30.0</td>
<td>17.0</td>
<td>-11.5</td>
<td>-11.5</td>
<td>-11.5</td>
</tr>
<tr>
<td>COMPAQ 386</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-4.7</td>
<td>N</td>
<td>-8.8</td>
<td>-3.3</td>
<td>-2.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMAX 386/4</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-39.2</td>
<td>N</td>
<td>-47.8</td>
<td>-14.3</td>
<td>-33.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMAX 386/4</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-41.0</td>
<td>N</td>
<td>-51.8</td>
<td>-15.7</td>
<td>-34.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPUADD-286</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-31.5</td>
<td>N</td>
<td>-43.3</td>
<td>-12.8</td>
<td>-29.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPUADD-286</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-35.0</td>
<td>N</td>
<td>-46.3</td>
<td>-15.3</td>
<td>-30.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPUADD-286</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-35.5</td>
<td>N</td>
<td>-44.4</td>
<td>-12.0</td>
<td>-30.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPUADD-286</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-37.9</td>
<td>N</td>
<td>-47.5</td>
<td>-15.3</td>
<td>-32.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NCI 386</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>-32.4</td>
<td>N</td>
<td>-29.1</td>
<td>9.3</td>
<td>-23.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1068
### TABLE 3

Percent Change in Observations Per Second (OPS) With DISK CACHING

\[ \% = 100 \times \frac{(\text{OPS w/ CACHE}) - (\text{OPS w/o CACHE})}{(\text{OPS w/o CACHE})} \]

Using 5 FILEBUFFERS of 512 Bytes

NEGATIVE % Implies Using DISK CACHING Takes Longer

<table>
<thead>
<tr>
<th>COMPUTER</th>
<th>MATH</th>
<th>COPROCESSOR</th>
<th>RAM</th>
<th>CREATE</th>
<th>DISK</th>
<th>DATA</th>
<th>CONCATENATE</th>
<th>SORT</th>
<th>SUMMARY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ON?</td>
<td>ON?</td>
<td>ON?</td>
<td>ON?</td>
<td>ON?</td>
<td>ON?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>4.6</td>
<td>9.8</td>
<td>4.0</td>
<td>4.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>4.8</td>
<td>3.5</td>
<td>2.7</td>
<td>1.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>27.4</td>
<td>48.0</td>
<td>7.0</td>
<td>3.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>28.6</td>
<td>46.4</td>
<td>1.5</td>
<td>1.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>5.7</td>
<td>12.2</td>
<td>1.0</td>
<td>4.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>5.4</td>
<td>3.6</td>
<td>2.5</td>
<td>3.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>3.0</td>
<td>7.2</td>
<td>28.0</td>
<td>26.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>2.3</td>
<td>1.0</td>
<td>29.6</td>
<td>24.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMAX 386/4</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>5.5</td>
<td>12.6</td>
<td>1.7</td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMAX 386/4</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>2.3</td>
<td>4.1</td>
<td>0.0</td>
<td>1.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPUADD-286</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>5.4</td>
<td>7.4</td>
<td>2.0</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPUADD-286</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>0.0</td>
<td>1.7</td>
<td>-0.8</td>
<td>-1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPUADD-286</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>4.4</td>
<td>7.5</td>
<td>3.0</td>
<td>1.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPUADD-286</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>0.5</td>
<td>1.7</td>
<td>-0.8</td>
<td>-1.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### TABLE 4

Percent Change in Observations Per Second (OPS) When Using a RAM DISK

\[ \% = 100 \times \frac{(\text{OPS w/RAM DISK}) - (\text{OPS w/o RAM DISK})}{(\text{OPS w/o RAM DISK})} \]

Using 5 FILEBUFFERS of 512 Bytes

POSITIVE % Implies Using RAM Disk Takes LONGER

<table>
<thead>
<tr>
<th>COMPUTER</th>
<th>MATH</th>
<th>COPROCESSOR</th>
<th>DISK</th>
<th>CREATE</th>
<th>DATA</th>
<th>CONCATENATE</th>
<th>SORT</th>
<th>SUMMARY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ON?</td>
<td>ON?</td>
<td>ON?</td>
<td>ON?</td>
<td>ON?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>4.6</td>
<td>13.5</td>
<td>18.4</td>
<td>4.9</td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>3.4</td>
<td>9.8</td>
<td>10.9</td>
<td>3.1</td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>4.8</td>
<td>7.0</td>
<td>16.9</td>
<td>1.5</td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>4.4</td>
<td>8.6</td>
<td>5.2</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>7.7</td>
<td>13.8</td>
<td>21.6</td>
<td>4.2</td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>4.4</td>
<td>9.9</td>
<td>4.6</td>
<td>3.9</td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>7.5</td>
<td>5.0</td>
<td>23.4</td>
<td>3.1</td>
<td></td>
</tr>
<tr>
<td>COMPAQ/286</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>3.6</td>
<td>3.5</td>
<td>5.9</td>
<td>2.7</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE 5

Average Speed and Relative Performance of PC's to an IBM/XT, Without EMS

<table>
<thead>
<tr>
<th>MHz</th>
<th>Computer</th>
<th>Speed (OPS)</th>
<th>Relative Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>DATA</td>
<td>Concat</td>
</tr>
<tr>
<td>4.77</td>
<td>IBM/XT</td>
<td>39.4</td>
<td>31.5</td>
</tr>
<tr>
<td>8.0</td>
<td>IBM AT</td>
<td>60.2</td>
<td>85.2</td>
</tr>
<tr>
<td>12.5</td>
<td>Compaq 286</td>
<td>146.5</td>
<td>177.1</td>
</tr>
<tr>
<td>16.0</td>
<td>Compaq 386</td>
<td>261.4</td>
<td>258.9</td>
</tr>
<tr>
<td>20.0</td>
<td>CompaAdd/286</td>
<td>215.4</td>
<td>273.9</td>
</tr>
<tr>
<td>20.0</td>
<td>AMAX 386/4</td>
<td>202.9</td>
<td>265.0</td>
</tr>
<tr>
<td>20.0</td>
<td>NCI 386</td>
<td>285.7</td>
<td>254.0</td>
</tr>
<tr>
<td>25.0</td>
<td>PS/2 Mod 80</td>
<td>320.0</td>
<td>457.1</td>
</tr>
</tbody>
</table>

1069
**Figure 1**
Observations Per Second For Creating 4 DATA sets of 4,000 Obs Each
5 Filebuffers of 512 Bytes, No RAM Disk

LEGEND OF NUMBERS PLOTTED:

- 0 = NO EMS, NO CACHE, NO COPROC
- 1 = EMS, NO CACHE, NO COPROC
- 2 = NO EMS, NO CACHE, NO COPROC
- 3 = NO EMS, NO CACHE, COPROC
- 4 = EMS, NO CACHE, COPROC
- 5 = EMS, NO CACHE, COPROC
- 6 = EMS, CACHE, NO COPROC
- 7 = EMS, CACHE, COPROC

**Figure 2**
Observations Per Second For Concatenating 4 DATA sets of 4,000 Obs Each
5 Filebuffers of 512 Bytes, No RAM Disk

LEGEND OF NUMBERS PLOTTED:

- 0 = NO EMS, NO CACHE, NO COPROC
- 1 = EMS, NO CACHE, NO COPROC
- 2 = NO EMS, NO CACHE, NO COPROC
- 3 = NO EMS, NO CACHE, COPROC
- 4 = EMS, NO CACHE, COPROC
- 5 = EMS, NO CACHE, COPROC
- 6 = EMS, CACHE, NO COPROC
- 7 = EMS, CACHE, COPROC

---

**Legend of Computers (MHz)**

- A = IBM/XT (4.77)
- B = IBM AT (8)
- C = COMPAQ 386 (12.5)
- D = COMPAQ 386 (16)
- E = PH/2 MODEL 80 (25)
- F = ASCII 386/4 (20)
- G = NCI 386 (20)
- H = PH/2 MODEL 80 (25)
Observations Per Second For Sorting a DATA set of 16,000 Obs
5 Filebuffers of 512 Bytes, No RAM Disk

LEGEND OF NUMBERS PLOTTED:
0 = NO EMS, NO CACHE, NO COPROC
1 = NO EMS, NO CACHE, COPROC
2 = NO EMS, CACHES, COPROC
3 = EMS, NO CACHE, NO COPROC
4 = EMS, NO CACHE, COPROC
5 = EMS, CACHES, COPROC
7 = EMS, CACHES, COPROC

FIGURE 3

Observations Per Second For PROC SUMMARY on 16,000 Obs
5 Filebuffers of 512 Bytes, No RAM Disk

LEGEND OF NUMBERS PLOTTED:
0 = NO EMS, NO CACHE, NO COPROC
1 = NO EMS, NO CACHE, COPROC
2 = NO EMS, CACHES, COPROC
3 = EMS, NO CACHE, NO COPROC
4 = EMS, NO CACHE, COPROC
5 = EMS, CACHES, COPROC
7 = EMS, CACHES, COPROC

FIGURE 4

LEGEND OF COMPUTERS (KHZ)
A = IBM/XT (477)
B = IBM AT (8)
C = COMPAQ/386 (13.5)
D = COMPAQ 386 (16)
E = COMPAQ/286 (20)
F = P3/2 MODEL 80 (25)

LEGEND OF COMPUTERS (MHz)
A = IBM/XT (477)
B = IBM AT (8)
C = COMPAQ/386 (13.5)
D = COMPAQ 386 (16)
E = P3/2 MODEL 80 (25)