

"Cross-Architectural Comparison of Processor Speed Using the SAS® System"

**O.V. Hanger
Nielsen Media Research**

INTRODUCTION

The current state of computer technology is encouraging the movement to a client/server type of architecture. For those who have traditionally worked in the mainframe world, the management of this "downsizing", "rightsizing" or transporting of their workload to a distributed environment can pose some challenging questions in the area of capacity planning. With Version 6 of the SAS System, the performance/capacity analyst has the potential to build SAS programs that can be run on a variety of commercial computing platforms. This paper presents one case study conducted under MVS®, VM®, OS/2® and UNIX®. The code developed and used for this benchmarking analysis is presented and some of the issues to consider when conducting this type of activity are outlined. Initial analysis results are shown.

MIPS AND BENCHMARKING

In the mainframe computer world, MIPS (millions instructions per second) are the general standard for identifying the power (speed) of the processor. Knowing the MIPS rating of a processor's engine(s) is useful for expressing a computer's capacity (mips per engine x number of engines) and normalizing CPU time for chargeback (the 'meaning' of a CPU second changes when going from one processor to another with a different speed). There are a variety of professional organizations and journals that publish MIPS ratings for many different types of processors. With the ever increasing complexity of computer systems in recent years (multiprocessors, partitioning, vector processors, reduced instruction sets, ect) the reliability of the industry "standard" MIPS numbers has begun less dependable. This is truly noticable when making comparisons across architectures (or even across vendors within the same architecture). In a recent review of some industry ratings, it was observed that a specific 486 PC was rated at 28 mips. Knowing that an existing inhouse mainframe running VM with 400+

concurrent users generally requires about 25 mainframe MIPS, it was doubted that the same workload could be supported on the 486 PC (and still have some capacity left over).

There has also been a rise in new metrics for evaluating processing power. These include TPC-A, TPC-B, Dhrystones, SPECmarks, Khornerstone, VUPs and more. All of these ratings address the issue of measuring processor performance in different ways and are intended to overcome the problems with MIPS. In many ways, though, they have clouded the issue even more. Each metric tends to be designed to effectively evaluate either specific types of processors or specific types of workloads on any processor. Unfortunately, new architectures and processors continue to be made available on the marketplace and there is no source that can provide all metrics for all platforms. Basically, processor speed or performance is truly a relative thing. It is relative to your workload, the performance that you have come to expect and the importance that computer systems play in your business. It is relative to the needs of each specific computing site.

Benchmarking is an activity that is conducted to evaluate the performance potential of a workload or application in one environment as compared to another computer environment (in a geological survey, a 'benchmark' is a known elevation from which unknown elevations can be determined). In its purist (and possibly most expensive) of forms, benchmarking uses a true workload and evaluates performance for a variety of resources (CPU, memory, I/O, ect) and service objectives. But, with the relative nature of performance measures, it is possible to create simple (and least costly) SAS programs that can provide CPU time (in seconds) for making cross-architectural comparisons of processor speed. In this way, a computing site can establish their own 'relative indicator of processing power' factor.

METHODOLOGY AND ASSUMPTIONS

At this case study site, processor benchmarking has been a standard procedure for evaluating CPU "speed" for every processor upgrade. The procedures and programs that have evolved over the 6 years that this task has been applied for the mainframe environment have proved to be effective and invaluable. The suite of programs are mostly written in SAS with a few containing code that can be run under both MVS and VM. With each processor change, the programs are run before and after the change. The sums of CPU seconds across programs are compared and a relative percent change is calculated. This percent change is applied to the unit value of the "pre" processor to get the "per engine" unit value of the "post" processor. The capacity of the processor is this unit value times the number of engines available to that specific operating system. These unit values are still referred to as MIPS (all relative to the MIP rating of the first "pre" processor ever used in benchmarking at this site).

The immediate need at this site was to make some initial capacity determinations for moving a workload from the mainframe environment to a UNIX based client/server platform. In addition, an evaluation of a couple of OS/2 platforms would be performed (mostly out of curiosity). A site specific MIP rating of 19.2 (per engine) had already been established for the IBM 3090-500J running (via PR/SM) both MVS/ESA and VM/HPO. This value is used as the benchmark for determining the relative processing power on the new distributed computer platforms: IBM PS/2 65SX running OS/2, COMPAQ Deskpro 486/33i running OS/2 and a SUN SparcServer 670 running SunOS. Each of these platforms had a Version 6 release of the SAS System installed and available (6.07 for MVS and VM, 6.08 for theSUN and 6.06 for OS/2). It is recognized that there can be significant differences in SAS System performance across release levels and that any interpretation of results must be so tempered.

For a variety of reasons, existing benchmarking programs could not be used for this case study. Three new one-step programs were written for performing this cross-architectural comparison of processor speed. The CPU time that is used for evaluating performance is obtained from the SAS log (as reported by the data step). On PC based

environments (such as OS/2) the CPU time reported is really elapsed time. This means that I/O time is also included in this value. Because of this limitation, the benchmarks were written to do minimum I/O processing. No datasets are created and the only I/O done is the initial load of the SAS System, the writing to the SAS log and a few messages to the terminal screen. All benchmarking program runs were done in the non-interactive mode of the SAS System. Both the low I/O content and the singular mode of SAS execution provide a means of controlling for problem aspects of the benchmarking activity. But they also limit the extent that interpretations can be made from the results.

THE PROGRAMS

The first program in this suite of routines is almost identical to one that had been historically used at this case study site. Among all of those routines, it was also the most reliable (it tended to best approximate the final result obtained from all routines combined). It consists of 2 nested do-loops with the outer loop sending a message to the terminal and the inner loop performing several simple arithmetic expressions. This program was run 10 times with the 'to-value' in the inner do-loop being varied to elongate the CPU time that is required to complete the step.

```
* BENCHMARKING ROUTINE 1 ;
FILENAME TERM TERMINAL;
DATA _NULL_ ; FILE TERM;
DO J=1 TO 10;
    PUT 'Starting outer iteration number -> ' J ;
    X=0;
    DO I=1 TO 500000 ;
        X=I+(1/I); X=X+I; X=X-I;
        X=X*I; X=X/I;
    END;
END;
STOP;
RUN;
```

With the second benchmarking routine, a number of SAS operations and functions are performed across two arrays of variables. The size of these arrays (number of elements/variables) was varied across ten different executions of the program by changing the value of the 'numvars' macro variable via the %LET macro statement in order to elongate CPU time required by the program.

```
* BENCHMARKING ROUTINE 2 ;
filename term terminal;
%let numvars=5 ;
data _null_ ; file term; date=mdy(12,7,52);
  array x {*} x1-x&numvars;
  array c {*} $ 1 c1-c&numvars;
  DO I=1 TO 20;
    put 'Starting outer iteration '
      ' number ' i z2. ' out of 20.';
    bigdate=put(date+i,worddate32.)||
      put(date+i,weekdate18.)||
      put(date+i,weekdate32.)||
      put(date+i,worddate18.)||
      put(date+i,worddate32.)||
      put(date+i,weekdate18.)||
      put(date+i,weekdate32.)||
      put(date+i,worddate18.) ;
    do j=1 to dim(x);
      x{j}=j+sqrt(j)-cos(i)+sin(j)-log(i+j);
      x{j}=x{j}*i/date;
      c{j}=substr(bigdate,i,1);
      x{j}=index(bigdate,put(i,3.));
    end;
  end;
run;
```

The final program in this suite of routines is a variation of the first program. A greater variety of functions, both character and numeric, are performed. This program was run 10 times on each test platform with the 'to-' value' in the 'DO I=' loop being varied to elongate CPU time across executions of the program.

```
* BENCHMARKING ROUTINE NUMBER 3 ;
FILENAME TERM TERMINAL;
DATA _NULL_ ; FILE TERM;
  DATE=MDY(1,25,82); TIME=HMS(8,0,0);
  WORD=
  'ABCDEFGHJKLMNOPQRSTUVWXYZ';
  DO J=1 TO 26 ;
    W=SUBSTR(WORD,J,1);
    PUT 'Starting outer iteration -> ' W ;
    DO I=1 TO 300;
      D=I/J; X=I+1+D; Y=(D-X)/3;
      DTM=DHMS(DATE+J,
        HOUR(TIME+I),
        MINUTE(TIME+I),
        SECOND(TIME+I));
      DOW=PUT(DATEPART(DTM),
        WEEKDATE3.);
      NEW=TRANSLATE(WORD,W,' ');
      LEN=LENGTH
        (COMPRESS(NEW,' '));
```

```
DO K=1 TO 10;
  IF MOD(K,2)=0 THEN Y=Y+1;
  ELSE Y=1;
  IF I+J>=18 THEN Z=1;
  ELSE Z=Y-X+2;
  V=MAX(I,J,D,X,Y,Z);
  U=MEAN(I,J,D,X,Y,Z);
  T=SUM(I,J,D,X,Y,Z);
  RPT=' '||REPEAT(W,16);
  SUB=SUBSTR(LEFT(RPT),5,8);
  DO L=1 TO 10;
    X=ROUND((Y+1-Z)/2);
    X=INT((Y-1+Z)/3);
  END;
  A=Z*X; B=A-Z; C=A+B;
END;
X=X+I; X=X-I; X=X*I; X=X/I;
END;
END;
RUN;
```

BENCHMARKING RESULTS

Each of the three benchmarking programs were run on all five computing platforms as described above and the CPU seconds reported in the SAS log was noted for each run. Three datasets were created (one for each benchmark routine) and the data was entered via the FSEDIT procedure. One observation in these datasets contains data from all test platforms for a specific value of 'extents' (value of the varied iteration/variable count) for that related benchmarking program. The variable names for each test platform are o2i (IBM PS/2), o2c (COMPAQ DeskPro), sos (Sun SparcServer), vme (VM/HPO) and mve (MVS/ESA). A print of each dataset follows:

Benchmark Test 1					
extents	o2i	o2c	sos	vme	mve
10	6.46	1.96	0.09	0.04	0.04
100	7.16	2.06	0.10	0.04	0.05
500	9.33	2.27	0.13	0.05	0.06
1000	10.97	2.62	0.17	0.07	0.08
5000	30.21	4.91	0.40	0.19	0.20
10000	52.71	7.82	0.70	0.34	0.35
50000	239.25	30.87	3.31	1.52	1.54
100000	490.43	59.81	6.24	3.01	3.05
500000	2328.28	290.85	30.50	14.76	14.99
1000000	4649.53	579.63	60.80	29.61	30.00

Benchmark Test 2					
extents	o2i	o2c	sos	vmh	mve
5	11.82	3.18	0.29	0.12	0.13
10	12.72	3.31	0.31	0.13	0.14
50	18.55	4.41	0.48	0.22	0.23
100	25.78	5.75	0.65	0.34	0.34
500	83.62	16.87	2.12	1.27	1.24
1000	156.13	30.39	4.02	2.43	2.36
5000	739.28	140.83	18.39	11.94	11.72
10000	1477.46	284.21	37.48	23.90	23.75
15000	2225.55	431.01	58.16	35.77	36.41
20000	3012.22	581.50	77.28	47.91	48.50

Benchmark Test 3					
extents	o2i	o2c	sos	vmh	mve
5	44.68	8.22	0.93	0.31	0.31
10	77.80	13.32	1.55	0.50	0.49
15	110.42	18.41	2.19	0.70	0.68
20	143.03	23.46	2.92	0.89	0.86
25	175.65	28.37	3.43	1.08	1.05
50	338.46	53.86	6.54	2.04	1.97
75	501.28	79.19	9.61	3.01	2.90
100	664.21	104.53	12.91	4.00	3.81
200	1315.10	205.80	25.51	7.92	7.51
300	1966.39	307.16	37.94	11.79	11.19

The following SAS program was used to analyze the benchmarking data. Note that the program determines relative performance attained (rpa) for the PS/2 (o2irpa), COMPAQ (o2crpa) and the SUN (sosrpa) using the average of VM and MVS as the base from which to extrapolate (both are running on the same mainframe which has already been evaluated to be 19.2 MIPS per engine).

```
* BENCHMARKING ANALYSIS 1;
%let base=19.2 ;
%macro datcalc(ds);
data benchmark&ds; set benchmarks.bentest&ds;
  format o2irpa o2crpa sosrpa 6.2 ;
  btot=(vmh+mve)/2; drop btot;
  o2irpa=(btot*&base)/o2i;
  o2crpa=(btot*&base)/o2c;
  sosrpa=(btot*&base)/sos;
proc print data=benchmark&ds noobs;
  title1 "Benchmark Test &ds";
run;
%mend datcalc;
%datcalc(1) %datcalc(2) %datcalc(3)
data summary; set benchmark1(in=b1)
  benchmark2(in=b2) benchmark3(in=b3) ;
  if b1 then routine='Benchmark1';
  else if b2 then routine='Benchmark2';
  else if b3 then routine='Benchmark3';
```

```
proc means data=summary noprint;
  var o2i o2c sos vmh mve ;
  output out=summary
  sum=o2i o2c sos vmh mve ;
  class routine;
data summary; set summary;
  drop _type_ _freq_ btot;
  format _numeric_ 6. ;
  format o2irpa o2crpa sosrpa 6.2 ;
  btot=(vmh+mve)/2;
  o2irpa=(btot*&base)/o2i;
  o2crpa=(btot*&base)/o2c;
  sosrpa=(btot*&base)/sos;
  if routine=' ' then routine='Total';
proc sort data=summary; by routine;
proc print data=summary noobs;
  title1 'Summary of Benchmark Tests';
run;
```

The summary data (edited from the final PROC PRINT in the above program) is as follows:

Summary of All Benchmark Tests			
Routine	o2irpa	o2crpa	sosrpa
Benchmark1	0.12	0.98	9.37
Benchmark2	0.31	1.59	11.99
Benchmark3	0.11	0.72	5.84
Total:	0.19	1.19	9.76

CONCLUSION

The transportability of programs written for Version 6 of the SAS System, provides a useful opportunity for analysts to evaluate capacity issues for moving workloads to a client/server type of environment. This paper presents a case study example of how the SAS System was applied in one instance to help with this type of analysis. The methodology, assumptions and code for this type of benchmarking activity was reviewed. Though it is recognize that the results have limitations in their interpretation, the results that are obtained provide a good first step towards understanding a new computing environment.

SAS is a registered trademark of SAS Institute Inc., Cary, NC, USA.

MVS/ESA, VM/HPO and OS/2 are registered trademarks of the IBM Corporation.

UNIX is a registered trademark of AT&T.