DATE: April 13, 1984.

TO: Prime Personnel.

FROM: Ron Miller.

SUBJECT: TIMETRACE, Users Guide.

REFERENCE: Microsecond timer.

KEYWORDS: CPU Time, Tracing, Timing, Performance.

ABSTRACT

The TIMETRACE Tool (TTRACE) is a set of routines which allow a user to collect routine traces and timings of programs written in SPL, PL1G, PL1, F77, Pascal, CC or CBL. Your routines do NOT have to be re-compiled, only re-LOADED with the TIMETRACE library of metering routines. When the program is executed, these routines create a data file which identifies your routines as they are executed and their current virtual times. A data reduction program reads this file and produces a report detailing the routine activity.

DESCRIPTION

The TIMETRACE Tool (TTRACE) is a set of routines which allow a user to collect routine traces and timings of programs written in SPL, PL1G, PL1, F77, Pascal, CC or CBL. Your routines do NOT have to be re-compiled, only re-LOADED with the TIMETRACE library of metering routines. When the program is executed, these routines create a data file which identifies your routines as they are executed and their current virtual times. A data reduction program reads this file and produces a report detailing the routine activity.

TTRACE PACKAGE

The TIMETRACE package consists of the following files:

TTLIB.BIN
TTMAP.CPL
MAKEMAP.SAVE
TTRACE.SEG

To use the package:

Step 1: Re-link your program using SEG. TIMETRACE is not supported for the LOAD or BIND formats. Before linking the system or language libraries with the LIBRARY command, LINK the file TTLIB.BIN from either TOOLBOX*>TIMETRACE* or TOOLBOX*>LIBRARY*.

Step 2: After linking, run the CPL procedure called TTMAP from TOOLBOX*>TIMETRACE*, with the name of the SEG file as an argument. If the argument is not provided, it will be prompted for. This procedure collects the names of all the routines in program with the help of the MAKEMAP.SAVE program and sorts the file by ECB address. At the end of the procedure, the name of this new map file will be printed. Remember this name as you will need to supply it to the data reduction routine. This name will be the name of your SEG file with the suffix '.TTMAP'.

Step 3: Execute your program. A file named TIMETRACE_ will be created in the current directory. If it exists, it will be overwritten.

Step 4: Execute tha data reduction program TTRACE.SEG from TOOLBOX*>TIMETRACE*. An example of an execution (with your software being named "TIME") is :

```
SEG TOOLBOX*>TIMETRACE*>TTRACE
Enter special map file name         : TIME.TTMAP
Enter TIMETRACE_ (default) file name: <CR>
Enter report output file name       : TIME.TTRACE
Enter optional report description line:
This is an example.
Enter name of procedure to start timing with-
(default is main procedure): <CR>
Do you want a trace (y, [n])? <CR>
Reading map...     Map read in...
OK,
```

The response to the first question is the name of the file created in step 2. If a carriage return is given for the second prompt, the filename TIMETRACE_ is assumed. This is the usual case. The next prompt is for the name of the output file which contains the report. The next prompt is for a line of text which will be printed on the top of each page of the report. Intervening blanks are allowed. If the header is not desired, then simply carriage return. The next prompt is for the name of the procedure at which to start collecting timing information. If a carriage return is given, then the main program is assumed. This is useful for collecting information about a phase of a program. The last prompt asks whether a trace of subroutine calls should be included in the output file. Entering a carriage return selects no trace. A trace is useful in understanding the calling behavior of the program.

Step 5: Analyze the output. A portion of a sample report:

Module Name	Number of Calls	EXCLUSIVE TIME Per Call (usecs)	EXCLUSIVE TIME Percent of Total	INCLUSIVE TIME Per Call (usecs)	INCLUSIVE TIME Percent of Total
TIME	1	250641	93.04	269398	100.00
>A	201	8	0.63	8	0.63
>B	1	17059	6.33	17059	6.33
>C	1	10	0.00	10	0.00

The module TIME is an external subroutine and the '>' symbols precede the name of internal subroutines. The first column is the number of times the subroutine was invoked. The next four columns give the timing information in microseconds.

At the time of writing, the 9950, 850, 7750, 550-II and 250-II machines, the timing information will have a resolution of one microsecond; for all other V-mode models, the resolution is 1024 microseconds.

The exclusive time is the time spent in that particular subroutine alone. This may also include time spent in PRIMOS and un-metered library routines. The inclusive time includes the time spent in all the subroutines this one called. The

data reduction program is careful to exclude recursive times. For each of the two times, the average time per call and the percent of total time are given.

If you answered YES to the question "Do you want a trace", your output file (TIME.TTRACE in the example above) will also contain a trace of routine entrypoint names in the order they were executed.

In the above example, the trace would look like :

```
TIME
.A
.B
.C
```

Where the period prior to the routines A, B and C indicate that these are internal (in this case second level) routines.

NOTES

One needs to keep in mind that several PRIMOS related overheads may be included in you timing information. One example of this is the CPU time charged to you for processing a page fault. This of course implies that the timing information is not only MODEL dependent, but also CONFIGURATION (i.e. amount of memory) and LOAD dependent. It is therefore advised you use TIMETRACE in a controlled environment.

The data reduction program works by simulating the run-time stack and may occasionally have trouble when many non-local goto's are done, especially from inside on-units. An error message and a dump of the stack will be printed if this occurs.