SLAC COMPUTATION GROUP
Stanford, California

CGTM No. 186
June 1977

AN OS/VS2 WORKING SET MONITOR

Leonard Shustek
Computation Research Group
Stanford Linear Accelerator Center
P.O. Box 4349
Stanford, California 94305

ADDENDUM TO CGTM No. 186

WORMON

AN OS/VS2 WORKING SET MONITOR

The following options can now be specified as parameters to WORMON:

ELAPTIME

This causes the time scale to use absolute elapsed wall-clock time rather than any pseudo-clock. The three time-scale choices are thus:

CPUTIME only the cputime used by the measured program

REALTIME cputime used by the measured program plus voluntary wait time.

ELAPTIME elapsed time = cputime used by the measured program,
+ cputime used by WORMON,
+ voluntary wait time,
+ involuntary wait time

VSPAGING

This causes WORMON to monitor the paging algorithm that VS uses, rather than imposing the more deterministic working-set algorithm. The resulting graph will display virtual memory allocation and real memory residency as functions of time. The graphs will vary considerably depending on the activity in the system at the time, but when both VSPAGING and ELAPTIME are specified, the true real-memory utilization by the job will be measured.

L. Shustek
18 July 1977

****
* WORMON *
****

VS WORKING SET MONITOR

One of the difficulties with a global page allocation strategy as used by OS/VS2 is that the paging behavior of a job depends not only on the characteristics of the job, but also on the activity in the system at the time the job is run. This makes it difficult to estimate the real memory requirements of a job in a way that can be used to improve program behavior.

A useful system-independent measure of a program's real memory requirement is the size of the "working set", which is intended to be the number of real memory pages that the program needs to run efficiently at any point in its execution. Formally, the size of the working set at time T is defined as the number of different pages that the program has referenced in the time interval from T-TAU to T. The parameter TAU is called the working-set window size, and basically controls how much past history is to be used to compute the current working set. Any pages that are 'older' than TAU are considered to be unnecessary for the program's execution and hence not part of the working set. The working set algorithm has been much discussed in the literature [1-4].

WORMON is a system that watches the execution of an arbitrary program running under OS/VS2R1 at SLAC and tracks the size of its working set. The output is a plot of the working set size versus time, where time can be defined as either the cpu time used by the job, or a combination of both cpu and voluntary wait time. In addition to the working set size, the plot also shows the virtual memory allocation and paging I/O activity as a function of time. A separate histogram is also produced which shows how many times each virtual page was referenced.

WORMON works by attaching the load module to be measured as a subtask and periodically examining it as it runs, much like PROGLOOK [5]. At every tick of a clock it has established, WORMON looks at the pages that are currently in real memory for the job. If it sees a page that was not there before, it records it as a new member of the working

set. At the same time it locks the page into real memory (via an OS macro), resets the hardware reference bit which records whether the page has been accessed, and records the current time. For pages which were in the working set already, it checks to see whether the reference bit is still cleared and if the page has been unreferenced for longer than TAU. If so, it removes it from the working set and unlocks the page so that VS can eventually transfer it to disk if it continues to remain unreferenced. Whenever the working set changes, a record is written on a disk file for later use by a plotting program.

The records written to disk contain two other fields which are later plotted. One is the number of paging I/O operations which were required since the last record. This is not the same as the number of changes of the working set because pages which left the working set but were not modified by the program need not be rewritten to the paging file. WORMON uses the "change" bit maintained by the hardware to determine if a page has been changed. The other field in the disk file record is the number of virtual memory pages that are allocated by the program at the current time. This will vary as the program gets and releases buffers or dynamically links to other programs.

A WORMON run consists of two job steps. The first step executes the program to be measured and creates a temporary file with records describing the paging activity. The second step reads the file and plots the results using TOP DRAWER [6] and the Unified Graphics system [7].

Various parameters can be specified on the EXEC card for the WORMON measurement program in the first step. Parameters are free-form, separated by commas and/or blanks. A number (represented by nnn in the following description) consists of one to five decimal digits.

PGM=xxxx (Default: PGM=MAIN)

This specifies the name of the load module which WORMON is to measure. If the program is not in one of the standard system libraries, the STEPLIB DD card should point to the library which contains the program.

CPUTIME or REALTIME (Default: REALTIME)

This controls what WORMON is to use as a time scale for determining when a page has been in the working set too long. REALTIME means that both the CPU time used by the program and its voluntary wait time should be considered. The resulting graph reflects the fact that the working set will decrease

while the program is waiting for I/O or some other event. CPUTIME means that only the CPU time, and not voluntary wait time, is to be used for the timeline. With either option, the CPU time used by WORMON itself and any wait time caused by other jobs in the system are not included in the timeline.

The disadvantage of REALTIME is that the amount of time that the program waits for I/O will vary somewhat from run to run and so the amount by which the working set decreases will change. For most programs this effect is negligible, so REALTIME is recommended because it is a more realistic model of the true characteristics of the program as seen by the system. For programs which have a lot of voluntary wait time (such as most of the realtime network systems) specifying REALTIME is important so that the plot correctly reflects the release of real memory during idle periods. In some cases it may be useful to specify CPUTIME in order to minimize differences between runs of WORMON so that the effect of program changes can be more easily seen.

TAU=nnn (Default: TAU=200)

This specifies the working set window size in milliseconds. Making TAU larger has the effect of smoothing the data with a low-pass filter and raising the average working set size. Making TAU smaller will make the graph more 'spikey' and lower the average working set size. A reasonable range for TAU is between 50 and 1000.

DELTA=nnn (Default: DELTA=20)

This specifies the time in milliseconds between updating the working set. This should be at least a factor of 5 or so smaller than TAU so that the resolution in determining the working set is reasonable. Making this too small will cause excessive use of CPU time for the measurement itself, and also create an excessive number of records to be generated for the plotting program. If the measured program is going to execute for a very long time (where time is defined by the CPUTIME/REALTIME option) this number should be increased.

TICK=nnn (Default: TICK=18)

This specifies the real (elapsed, wall clock) time in milliseconds between awakenings of WORMON to see what the measured job is up to. It is sufficient to make this just a bit smaller than DELTA in most cases. It doesn't make much sense to make TICK smaller than about 5 since the system overhead in processing the interrupt caused by the timer tick is at least several milliseconds.

If any parameters are to be passed to the program which is being measured, they should be added to the end of the parameters to WORMON following a slash (/).

Examples of WORMON parameters:

```
PARM='PGM=SNOBOL'

PARM='PGM=REALTIME,TAU=400,DELTA=50,TICK=45/WET'

PARM='/FOOBAR'
```

JCL FOR RUNNING WORMON

The JCL for running WORMON to produce plots on the versatec printer is shown below. You may want to modify it to:

(1) Add parameters for WORMON or the measured program

(2) Add to the STEPLIB concatentation to specify a load module library which contains the program

(3) Add DD cards for the executing program where indicated

```
//         JOB
//*MAIN    TYPE=VS2
//WORMON   EXEC  PGM=WORMON,PARM=''  <--PARMS TO WORMON GO HERE
//*
//*  *** WORMON ***    WORKING SET MONITOR
//*
//STEPLIB    DD  DSN=WYL.CG.LJS.LOADMODS,DISP=SHR
//*              <-- ADD STEPLIB CONCATENATIONS HERE -->
//WORPRINT   DD  SYSOUT=A     --- OUTPUT FROM WORMON ---
//WORDATA    DD  UNIT=SYSDA,DISP=(,PASS),DSN=&&WSDATA,
//           SPACE=(TRK,(40,20),RLSE)
//SYSUDUMP   DD  SYSOUT=A
//*
//* --> DDCARDS FOR THE EXECUTING PROGRAM GO HERE <--
//*
//PLOT  EXEC LOADGO,GOPRM='V12FF    ',  (SELECT VERSATEC)
//*
//*  *** PLOT STEP ***
//*
//    LKEDPRM='SIZE=400K',
//    LKEDLB1='WYL.CG.RBC.LOADMODS',
//    LKEDLB2='WYL.CG.RCB.UGFTNLIB',
//    GOSL1='WYL.CG.RCB.UGRUNLIB',
//    GORGN=450K
//GO.SYSLIN   DD  DSN=WYL.CG.LJS.LOADMODS(WORMONP),DISP=SHR
//GO.FT01F001  DD  DSN=&&WSDATA,DISP=(OLD,DELETE)
//GO.UGDEVICE  DD  SYSOUT=P,
//             DCB=(RECFM=FBA,LRECL=134,BLKSIZE=1340)
```

If you wish to produce plot output which can be listed directly on a Tektronix 4013 terminal, use the following JCL instead (remember to change the name of the plot dataset):

```
//         JOB
//*MAIN    TYPE=VS2
//WORMON   EXEC  PGM=WORMON,PARM=''  <--PARMS TO WORMON GO HERE
//*
//*  *** WORMON ***    WORKING SET MONITOR
//*
//STEPLIB    DD  DSN=WYL.CG.LJS.LOADMODS,DISP=SHR
//*              <-- ADD STEPLIB CONCATENATIONS HERE -->
//WORPRINT   DD  SYSOUT=A     --- OUTPUT FROM WORMON ---
//WORDATA    DD  UNIT=SYSDA,DISP=(,PASS),DSN=&&WSDATA,
//           SPACE=(TRK,(40,20),RLSE)
//SYSUDUMP   DD  SYSOUT=A
//*
//* --> DDCARDS FOR THE EXECUTING PROGRAM GO HERE <--
//*
//PLOT   EXEC LOADGO,GOPRM='',  (SELECT 4013 PLOTS BY DEFAULT)
//*
//*  *** PLOT STEP ***
//*
//    LKEDPRM='SIZE=400K',
//    LKEDLB1='WYL.CG.RBC.LOADMODS',
//    LKEDLB2='WYL.CG.RCB.UGPTMLIB',
//    GOSL1='WYL.CG.RCB.UGRUNLIB',
//    GORGN=450K
//GO.SYSLIN  DD  DSN=WYL.CG.LJS.LOADMODS(WORMONP),DISP=SHR
//GO.FT01F001  DD  DSN=&&WSDATA,DISP=(OLD,DELETE)
//GO.UGDEVICE  DD  DSN=WYL.GG.UUU.P, <-- CHANGE TO YOUR ACCOUNT
//    DISP=(NEW,KEEP),
//    VOL=SER=SCR001,UNIT=DISK,SPACE=(TRK,(10,5,3)),
//    DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600)
```

NOTES

• The sample JCL above can be found in WYL.CG.LJS.LIB#WORMON on cat.

• WORPRINT is a print file which will contain messages generated by WORMON as it measures the program. The file characteristics are FBA/133/1330.

• WORDATA is the file which is transmitted to the plotting program. The file characteristics are FB/16/1600.

• Other graphics devices supported by the Unified Graphics system can be used simply by specifying the name of the device in the GOPRM for the plot step and changing the UGDEVICE ddcard appropriately. (See the TOP DRAWER [6] or Unified Graphics [7] manuals for details.)

• If the program you wish to measure is not already a load module, you may wish to add job steps at the beginning of the job to create the load module. For instance, to measure a FORTRAN program you can add

```
//  EXEC  FORTHCL
//FORT.SYSIN  DD  *
    -- your program --
```

after the JOB card, which creates a temporary load module containing your program. That load module can be made available to WORMON by adding

```
//  DD  DSN=&&GOSET,DISP=(OLD,DELETE)
```

to the STEPLIB concatenation in the WORMON step.

REFERENCES

[1] P. J. Denning, "The Working Set Model for Program Behavior", Comm. Assoc. Comput. Mach., vol 11, pp. 323-333, May 1968.

[2] P. J. Denning and S. C. Schwartz, "Properties of the Working Set Model", Comm. Assoc. Comput. Mach., vol 15, pp. 191-198, March 1972.

[3] W. J. Doherty, "Scheduling TSS/360 for Responsiveness", Proc. AFIPS 1970, FJCC

[4] J. Rodriguez-Rosell, "Experimental Data on How Program Behavior Affects the Choice of Scheduler Parameters", Proc. Third Symp. on Oper. Syst. Prin., October 1971.

[5] R. Johnson, T. Johnston, "PROGLOOK Users Guide", SLAC Computing Services User Note #33, Rev. 5, May 1976.

[6] Roger B. Chaffee, "TOP DRAWER", SLAC Computation Research Group Technical Memo CGTM #178, September 1976

[7] Robert C. Beach, "THE SLAC UNIFIED GRAPHICS SYSTEM", SLAC Computation Research Group Technical Memo CGTM #170, February 1976

Sample WORMON Output

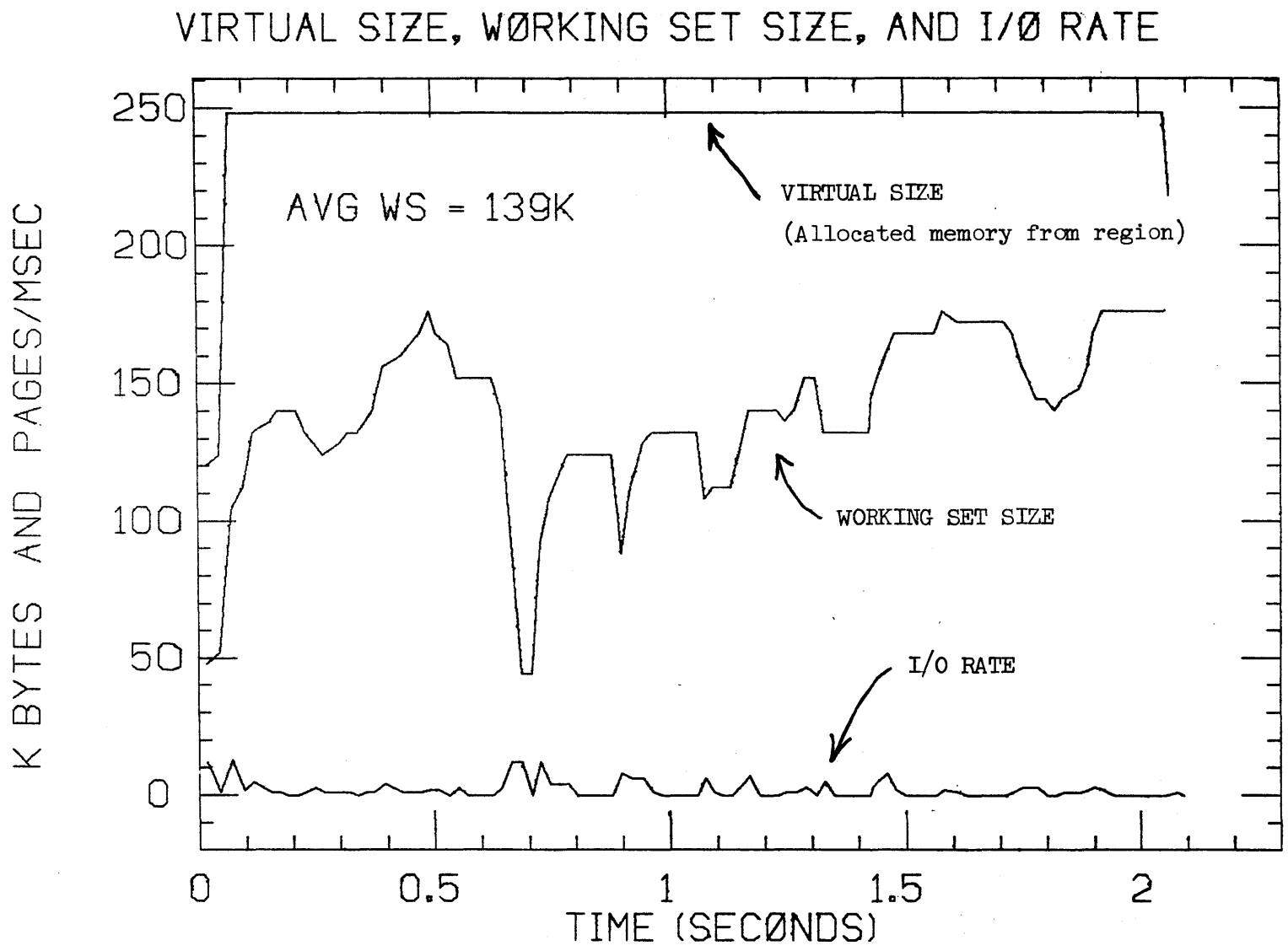
(A) Measurement Summary

```
***** WORMON *****  WORKING SET MONITOR
  6:3Ø P.M.   TUESDAY      JUNE 28, 1977
   OPTIONS SPECIFIED: PGM=FORTRANH

   MODULE "FORTRANH" WILL BE ATTACHED
   TIME SCALE WILL INCLUDE ONLY CPU TIME
---> TASK HAS TERMINATED WITH RETURN CODE ØØØØØ4

MEASUREMENT SUMMARY:
              2ØØ MSEC WORKING SET WINDOW SIZE
               2Ø MSEC SAMPLING INTERVAL
               18 MSEC REALTIME INTERRUPT INTERVAL
             32Ø K BYTE REGION WAS MONITORED
           7,76Ø PAGES WERE EXAMINED
           3,556 CALLS TO IEAPTRV WERE REQUIRED
              Ø PAGES DISAPPEARED THROUGH FREEMAINS
              43 CHANGED PAGES WERE REMOVED FROM THE WORKING SET
              91 WORKING SET INCREASES
              46 WORKING SET DECREASES
              91 PAGE FIXES WERE DONE
               Ø PAGE FIXES HAD TO BE WAITED FOR
           2,417 REFERENCE BIT RESETS
              46 PAGE FREES WERE DONE
             361 TIMER TICKS WERE REQUESTED
              97 DELTA INTERVALS WERE SAMPLED
               Ø DELTA INTERVALS WERE LOST
              23 DUPLICATE RECORDS WERE SUPPRESSED
           7,721 MSEC ELAPSED TIME
             178 MSEC CPU TIME USED BY WORMON
              79 MSEC VOLUNTARY WAIT TIME
           1,946 MSEC CPU TIME USED BY THE MEASURED PROGRAM

***** CIAO *****
```



(B) Working Set Size

(C) Virtual Page Utilization