UCRL-17893

cy 2

University of California
Ernest O. Lawrence Radiation Laboratory

SYSTEM EVALUATION ON THE CONTROL DATA 6600

David F. Stevens

October 28, 1967

TWO-WEEK LOAN COPY

This is a Library Circulating Copy which may be borrowed for two weeks. For a personal retention copy, call Tech. Info. Division, Ext. 5545

UCRL-17893 cy 2

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

For Proceedings of the International
Federation for Information Processing,
Edinburgh, Scotland, Aug. 5-10, 1968

UCRL-17893
Preprint

UNIVERSITY OF CALIFORNIA

Lawrence Radiation Laboratory
Berkeley, California

AEC Contract No. W-7405-eng-48

SYSTEM EVALUATION ON THE CONTROL DATA 6600

David F. Stevens

October 23, 1967

SYSTEM EVALUATION ON THE CONTROL DATA 6600*

DAVID F. STEVENS
Lawrence Radiation Laboratory, University of California
Berkeley, California, USA

Abstract: This paper describes a simple technique by which a peripheral processor can be used on a Control Data 6600 as a programmable hardware monitor for system evaluation with a minimum of interference. Two examples are given, and some suggestions for future investigations are offered.

As computing systems increase in size and complexity, it becomes both more important and more difficult to determine precisely what is happening within a computer. The two sorts of performance measurements which are readily available are not very useful; they are the micro-timing information provided by the manufacturer (.4 microseconds/floating add) and the macro-timing information provided by the user ("why does it take three days to get my job back?"). The relationship, if any, between them is obscured by the intricate bulk of the operating system; if it is a multi-programming or time-sharing system, the obscurity is compounded.

The tools available to the average installation for penetrating this maze are few and inadequate. Simulation is not particularly helpful: the information which is lacking is the very information necessary for the construction of an accurate model. Trace routines interfere excessively with the operation of the system, distorting storage requirements as well as relative timing information. Hardware monitors are not generally available, and though a wondrous future is foreseen for certain of them [1, 2], they have yet to demonstrate their capabilities in an operational environment; furthermore, they are certain to be too costly for permanent installation, and perhaps too cumbersome for temporary use. The peripheral processor of the Control Data 6000 series computers, however, provides some installations with an easily utilized, programmable hardware monitor for temporary use at no extra cost.

The peripheral processor (PPU) has access to all channels (and hence to all peripheral devices), all of central memory, the location counter of the central processor, and (through the exchange jump instruction [3]) the contents of all the operation registers. It has no interface problems, since it was an integral part of the original design of the computer. It can output data either directly onto any external device available to the system, or it can stream data into central memory for formatting and analysis by a Fortran program. When not in use as a hardware monitor it automatically reverts to its status as a "pool processor," available for use by the system. In only one respect is it less than satisfactory as a hardware monitor: it has no direct access to any other peripheral processor. (Communication is possible between two PPUs either across a channel or through central memory, but they must cooperate: no peripheral processor has a window into any other.)

Our use of a PPU as a hardware monitor began more than two years ago with the introduction of a PPU program (instead of a conventional trace routine) to produce a histogram of the behavior of the (central processor) location counter during long production jobs. The purpose was to facilitate the task of optimizing the production code; the program was quite successful and is still in use today. Of more importance than this success itself was the impetus it provided towards the development of other, more sophisticated, information-gathering routines.

Since the information to be collected by the histogramming program is pertinent to one particular program, it is reasonable that its PPU be "attached" to that program for accounting purposes throughout the histogramming run. A general-purpose monitor should be attached to no specific user program, however, for it is collecting information about total system performance. Furthermore, the design of the operating system, which restricts the number of simultaneously active jobs to seven, makes it desirable to avoid the introduction of a special system program to control such a monitoring routine. To eschew such control completely would be to require the monitoring routine to cease operation when the PPU memory is full, or else to output directly to an external device or to a central memory buffer. In the first alternative, one is hampered by the small size of the PPU (4096 12-bit words); in the others, by the introduction of interference which could distort the picture one is trying to form. The method actually adopted is felt to be a reasonable compromise, providing for a modicum of control with negligible interference.

The monitoring routine (MR) is initiated by a system job input through the normal job stream. This results in the MR PPU being attached to the system job. The MR's first act is to detach its PPU immediately from that job, and then to abort it (the system job), permitting access to the computer by more productive jobs. This effectively removes the MR PPU from the system, reducing the number of available "pool processors" from eight to seven. Periodically thereafter (the period being dependent upon the time required to fill the PPU's memory with

*Work done under the auspices of the U. S. Atomic Energy Commission.

data) another system job is initiated to accept the information collected by the MR, format it nicely, and print it. The formatting programs are quite simple; they are quick and small, requiring only 200-300 kws (kiloword seconds) and less than five central processor seconds per run. Since they need be run only once every two hours or so (the shortest period we have found necessary so far is 2 1/2 hours), these runs require less than 0.25% of the central computer during the monitoring period (there are 864 000 kws available to the users in two hours, and 7200 CP seconds). The loss of one PPU for the duration of the monitoring has entailed even less interference: our studies have shown that the seven pool processors which were available during monitoring runs were in use simultaneously only about 0.2% of the time. (Other installations attempting similar experiments should recognize that our experience is a function both of our work load and our operating system--which is not standard. They may experience much greater PPU loading, and hence much greater interference.)

To minimize operator error, the periodic resubmission of the system job to format and print the information collected by the MR is handled automatically by the operating system. Instead of disappearing from the system at its conclusion, the job is saved on the disk until the desired period has elapsed, when it reenters the input queue. (This automatic resubmission has been even more useful for the running of accounting and diagnostic maintenance routines than for the infrequent MR runs.)

Two MRs have been written to date. The first was EYE, which gathers information on central processor activity, central memory utilization, channel activity, PPU activity, control point disposition (the "control points" are the seven software constructs via which the users gain access to the system [4]), and tape usage. The second was SEE, which furnishes data on the utilization of the disks (percent of activity and whether activity was generated during compilation, user execution, or system I/O; the total number of disk requests; the number of requests requiring repositioning) and on a job profile, showing how the real time at a control point of the users' jobs was divided among active use of the central processor, waiting for I/O, and waiting for the CP, with separate breakdowns for compile and non-compile time.

Sample output from the formatting routines for those two jobs is included as figs. 1-3. Figures 1 and 2 show the first and last pages of the output from an EYE run; the intervening pages all look like the bottom portion of the first page. The CP utilization figures derive from a 4.096 ms sampling process; this rather odd interval was chosen for simplicity: the 6600 has an internal clock with one-microsecond ticks and a 12-bit field, hence 4096 microseconds. The line entries arise from a 15-second sampling process. This period is rather large, but PPU memory limitations forced a choice between a large period or frequent polling of the MR; the former seemed preferable. A brief explanation of each entry in figs. 1 and 2 is given below:

Figure 1:

Operating characteristics: The times are provided by the system real-time clock

Total count: The number of 4.096 ms samples

CP idle: Central processor not assigned to a user job

Non-compile: Central processor assigned to a user job not in compilation phase

Compile time: Central processor assigned to a user job during compilation phase

Jobs started: number of new user jobs arriving at control points during the sample period

Channels (incl. CP) active:

$$\frac{1}{n} \sum_{1}^{n} (A_i + CP_i),$$

where n = total count

A_i = number of I/O channels active at i^{th} sample time

CP_i = 0 if CP not active at i^{th} sample time

CP_i = 1 if CP active at i^{th} sample time

Number of active PPUs: Does not include EYE, the system monitor, or the PPU driving the operator's display console

Control points: There are seven altogether; USER+OPERATOR+EMPTY = 7

User: Number of control points active with user jobs

Operator: Number of control points active with operator-originated system I/O jobs

Empty: Number of inactive control points

Recall: Number of control points in "recall" status (i.e., waiting for the completion of an I/O operation)

Compiling: Number of control points at which compilation is in progress

Active channels: The second occurrences of 0-3 denote channels 10-13.

A = channel active; blank = channel inactive

Tapes assigned: The number of tape drives assigned to active jobs

Disk tracks assigned: The number of disk tracks assigned to jobs at user control points; does not include input or output queues.

Memory in use: Central memory actually occupied by the system and the active jobs; the entry is in octal; the maximum possible is 400 000.

Figure 2:

The first few entries are the same as for fig. 1.

Average and peak values for m line entries: m is the number of 15-second samples

Channels: Does not include central processor

Copies of RUN = number of compilations; RUN is the name of the compiler

RCL jobs = Jobs in recall

Individual channels

0 - Disk

1 - Communication link to the Control Data 6411 I/O Module
2 - Disk
3 - Unused
4 - Unused
5 - Disk pack and Data Cell Drive
6 - Disk pack
7 - Tape (4 drives; dual access with channel 11)
10 - Operator's console
11 - Tape (4 drives; dual access with channel 7)
12 - Tape (4 drives; dual access with channel 13)
13 - Tape (4 drives; dual access with channel 12)

Figure 3 shows the output from a SEE run. SEE also makes use of the hardware clock to establish a 4.09 ms sample time. The entries in this output are less mysterious than those for EYE, but still require some explanation.

Total samples: Σ_1^n Ni, where n = number of sample periods and Ni = number of user jobs active at i^{th} sample time.

Total compile samples: Σ_1^n Ci, where Ci = number of user jobs at i^{th} sample time which are in the compile phase.

Total non-compile samples: Σ_1^n Xi, where Xi = number of user jobs at i^{th} sample time which are not in the compile phase.

"Active" means the central processor is assigned to a user job

"Recall" means a job is waiting for the completion of an I/O operation

"Waiting" means a job is waiting to be assigned the central processor

ACT/RCL is the ratio of active samples to recall samples

Disk survey: The survey period is the same as for the compiler survey

OPERATING CHARACTERISTICS FROM 15.56.41 TO 17.15.13, 08/22/67.

```
 TOTAL COUNT=    1149868 (ONE COUNT EVERY 4.096 MILLISECONDS)
CP IDLE TIME=     391989 (34.081 PERCENT)
 NON-COMPILE=     677389 (58.910 PERCENT)
COMPILE TIME=      80591 ( 7.009 PERCENT)
JOBS STARTED=         75
```

2.61 CHANNELS (INCL CP) ACTIVE

NUMBER OF ACTIVE PPUS	------CONTROL POINTS------ USER	OPERATOR	EMPTY	RECALL	COMPILING	ACTIVE CHANNELS 0	1	2	3	4	5	6	7	0	1	2	3	TAPES ASSIGNED	DISK TRACKS ASSIGNED	MEMORY IN USE
2	3	1	3	1	1	A								A				0	190	331300
1	3	1	3	0	1									A				0	190	331300
2	3	1	3	1	2	A								A				0	191	331300
2	3	1	3	1	2	A								A				0	190	331300
1	3	1	3	1	2	A								A				0	190	331300
2	3	1	3	1	2	A								A				1	195	331300
2	3	1	3	1	2									A				1	195	331300
0	3	1	3	0	2									A				1	197	331300
1	3	1	3	0	2									A				1	197	331300
1	3	1	3	0	2									A				1	197	331300
1	3	1	3	0	2									A				1	198	331300
1	3	1	3	0	2									A				1	198	331300
1	3	1	3	0	2									A				1	198	331300
2	3	1	3	0	2	A								A				1	198	331300
2	3	1	3	1	2									A				1	198	331300
2	3	1	3	1	2									A				1	198	331300
3	3	1	3	1	2	A								A				1	198	332400
2	3	1	3	1	2									A				1	198	332400
2	3	1	3	1	2									A				1	198	332400
2	3	1	3	1	1									A				2	198	332400
3	3	1	3	1	1	A								A				2	200	332400
1	3	1	3	1	1									A				1	165	332400
2	3	1	3	1	1									A				2	165	332400
3	3	1	3	1	0	A								A				2	165	332400
1	3	1	3	1	0									A				2	165	332400
2	3	1	3	1	0									A				2	165	332400
3	3	1	3	1	0									A				2	168	332400
3	3	1	3	1	0									A				2	172	332400
2	3	1	3	1	0									A				1	163	332400
2	3	1	3	1	0									A				1	167	342400
4	3	1	3	1	0			A						A				1	170	342400
2	3	1	3	1	0									A				1	165	332400
2	3	1	3	1	0									A				1	166	332400
3	3	1	3	2	0	A								A				1	167	332400
2	3	1	3	2	1	A								A				2	166	332400
4	3	1	3	2	1	A								A				1	178	332400
4	3	1	3	2	1	A								A				1	182	332400
4	3	1	3	1	1	A								A	A			1	186	332400
3	3	1	3	0	1									A	A			1	189	332400
3	3	1	3	0	1									A				1	192	332400

Fig. 1. First page of output from EYE.

XBL 684-612

```
                    OPERATING CHARACTERISTICS FROM 15.56.41 TO 17.15.13, 08/22/67.
                                        (SUMMARY)

     TOTAL COUNT=   1145868 (ONE COUNT EVERY 4.096 MILLISECONDS)
    CP IDLE TIME=    391888 (34.081 PERCENT)
     NON-COMPILE=    677389 (58.910 PERCENT)
    COMPILE TIME=     80591 ( 7.009 PERCENT)
    JOBS STARTED=        76

                    2.61 CHANNELS (INCL CP) ACTIVE

AVERAGE AND PEAK VALUES FOR 314 LINE ENTRIES....AVERAGE                          PEAK
                                                 1.94 CHANNELS (EXCL CP) ACTIVE    4
                                                 3.03        PPUS ACTIVE           7
                                                 4.00         USER JOBS            6
                                                 1.00       OPERATOR JOBS          1
                                                 2.00    EMPTY CONTROL POINTS      4
                                                 1.69       JOBS IN RECALL         4
                                                  .72        COPIES OF RUN         3
                                                 2.26        TAPES ASSIGNED        6
                                               238.27    DISK TRACKS ASSIGNED    352
                                               343535      MEMORY ASSIGNED   377400

ACT.PPUS    USER JOBS    OP.JOBS    EMPTY CPS    RCL JOBS    COMPILES    CH. ACT.    TAPES    DISK TRACKS ASSIGNED  MEMORY IN USE
0    .64    0    0.      0    0.     0    2.23   0   10.51   0   48.09   0    0.     0   4.46     0-100      0.     000-040K   0.
1   6.69    1    0.      1100.00     1   34.08   1   32.48   1   34.08   1   28.66   1  27.07   100-200    13.06   040-100K   0.
2  23.89    2    6.69    2    0.     2   31.53   2   36.94   2   15.61   2   50.64   2  35.99   200-300    85.35   100-140K   0.
3  31.85    3   25.48    3    0.     3   25.48   3   17.83   3    2.23   3   19.11   3  11.46   300-400     1.59   140-200K   0.
4  28.98    4   31.52    4    0.     4    6.69   4    2.23   4    0.     4    1.59   4  13.06   400-500,    0.     200-240K  10.19
5   5.10    5   34.08    5    0.     5    0.     5    0.     5    0.     5    0.     5   7.64   500-600     0.     240-300K    .32
6   2.55    6    2.23    6    0.     6    0.     6    0.     6    0.     6    0.     6    .32   600-700     0.     300-340K  24.20
7    .32    7    0.      7    0.     7    0.     7    0.     7    0.    MORE  0.     7   0.       MORE       0.     340-400K  65.29
                                                                                     8   0.
                                                                                     9   0.

INDIVIDUAL CHANNEL ACTIVITY
        00  55.10
        01   2.87
        02  14.97
        03   0.
        04    .96
        05    .64
        06  10.83
        07    .96
        10  96.18
        11  11.15
        12   0.
        13   0.
```

Fig. 2. Last page of output from EYE.

XBL 684-613

```
COMPILER SURVEY FROM 22.56.09 TO 00.10.10, 09/08/67.
(TOTAL NUMBER OF JOBS STARTED= 19.)

   TOTAL SAMPLES (=SIGMA N(T), WHERE N(T)=NUMBER OF USERS AT TIME T, SAMPLED EVERY 4.096 MILLISECONDS)= 2035360.

   TOTAL COMPILE SAMPLES=   33643, ( 1.653 PERCENT).  TOTAL NON-COMPILE SAMPLES= 2001717, (98.347 PERCENT).

      COMPILE ACTIVE =   12965,   .637 PERCENT.        NON-COMPILE ACTIVE = 1009079, 49.577 PERCENT.
      COMPILE RECALL =   16858,   .828 PERCENT.        NON-COMPILE RECALL =  283259, 13.917 PERCENT.
      COMPILE WAITING=    3820,   .188 PERCENT.        NON-COMPILE WAITING=  709379, 34.853 PERCENT.
      COMPILE ACT/RCL=    .769.                        NON-COMPILE ACT/RCL=   3.562.


DISK SURVEY

(THERE WERE   4851 DISK REQUESTS, OF WHICH   1853 ( 38.20 PERCENT) INCLUDED REPOSITIONING, DURING THE SURVEY PERIOD)

TOTAL SAMPLES (ONE PER DISK EVERY 4.096 MILLISECONDS)= 2166784.

        UNRESERVED  2029058, 93.644 PERCENT
RESERVED, INACTIVE     4982,   .230 PERCENT
            ACTIVE   132744,  6.126 PERCENT

   BREAKDOWN OF ACTIVITY
                  COMPILE     9722,  7.324 PERCENT
           USER EXECUTION    78842, 59.394 PERCENT
       OPERATOR EXECUTION    44180, 33.282 PERCENT
```

Fig. 3. Output from SEE.

XBL 684-614

Disk requests: Number of requests for a disk channel that were granted

Total samples: Twice the number of sample periods (there are two disks)

- Unreserved: A disk channel is not reserved
- Reserved, inactive: A disk channel is reserved, but not active
- Active: A disk channel is (reserved and) active

Breakdown of activity:

- Compile: A disk channel is active and assigned to a user job in the compile phase
- User Exec: A disk channel is active and assigned to a user job not in the compile phase
- Operator Exec: A disk channel is active and assigned to an operator (system I/O) job

These programs have been of great assistance in the evaluation of major changes in the operating system. For example, early in 1967 it became apparent that one major obstacle to improvement of throughput was the handling of the disks. The available alternatives were (1) to redesign the entire operating system to provide more efficient allocation of disk space, and (2) to reduce the number of disk requests per job. The second was selected, and the problem was attacked on two fronts: by altering the track-chaining method, and by assigning the standard output file to the 6411 I/O Module via the communication link on channel 2 (instead of to a disk). Before these improvements were implemented, we generated 360 disk requests per job; the new track-chaining scheme reduced the number to 300; the reassignment of the output file, to 200. Our disk channel activity used to approach 80% (40% on each channel); the output samples in figs. 2 and 3 are typical of our current activity (40% total).

They have also been used to compare two operating systems. Their simplicity rendered it easy to modify them to run under the new system as well as under our standard system. Although much of the comparison could be (and in fact was) carried out by more conventional methods, those methods would not have indicated that the new system required 25-50% more PPUs and nearly twice as much disk channel activity, and that it generated almost 50% more disk requests than our standard system.

The investigations described above have been the obvious ones. A PPU monitoring routine could also examine some more interesting questions. Among them might be

1. An investigation of CP burst-times; how their lengths and distribution vary between I/O bound and compute-bound jobs.
2. An examination of the vagaries of PPU scheduling: Which PPUs are used most? For what? Is the system interference caused by the introduction of an MR independent of the PPU to which the MR is assigned?
3. An investigation of channel burst-times.
4. How does the efficiency of the scheduling algorithm vary with the length of the input queue?

Perhaps if computer manufacturers could be persuaded to include a small, programmable hardware monitor in their original designs such questions, and others, would receive the attention they deserve.

REFERENCES

[1] G. Estrin, D. Hopkins, B. Coggan, and S. Crocker, Snuper computer - a computer instrumentation automaton; AFIPS Conf. Proc., Vol. 30, 645-656 (Thompson Book Co., Washington, D. C., 1967).

[2] F. Schulman, Hardware measurement device for IBM system/360 time-sharing evaluation, in: Proc. of 22nd National Conf. ACM 103-109 (Thompson Book Co., Washington, D. C., 1967).

[3] Control Data 6000 series computer systems reference manual. Control Data Corporation Publication Number 60100000.

[4] Control Data 6000 series computer systems Chippewa Operating System reference manual. Control Data Corporation Publication Number 60134400.

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.