TO: Multics Performance Log

FROM: A. Sekino

SUBJECT: Summary of the 3.1.1 system performance

DATE: August 11, 1969

1. Introduction

The performance of the 3.1.1 system observed during console sessions is summarized in this note. The following three performance measures were considered in detail:

(1) CPU time

(2) number of page faults

(3) system response time

where the system response time is the time elapsed since the last input at the console till the first response from the system. Therefore, the system response time includes a CPU time, queueing time, and so on.

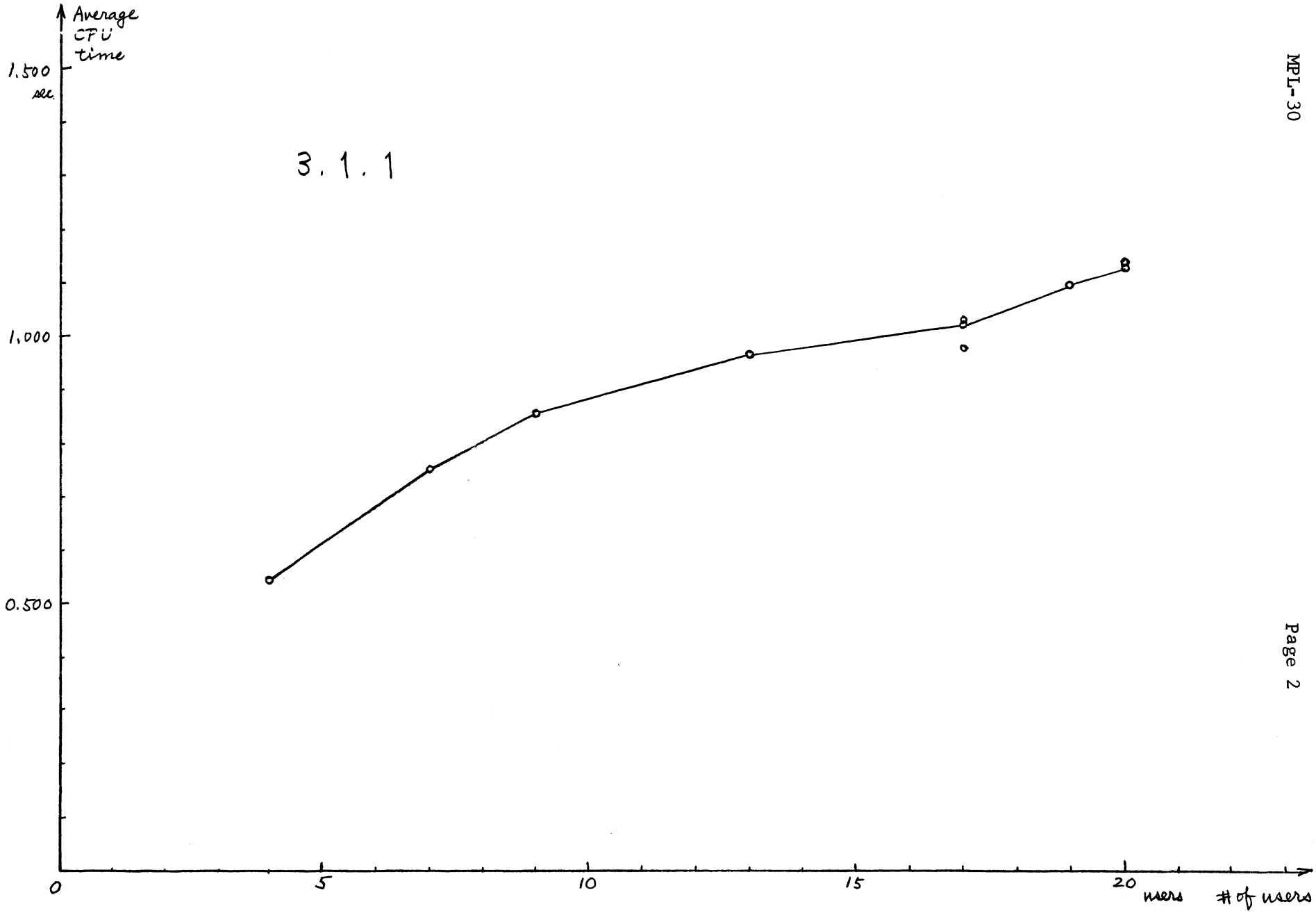
The results concerning the CPU time and the number of page faults are based on a daily Multics performance measurement of the period, July 23 - August 2. Therefore, these results are obtained by running a standard script, MFTN3, which contains 66 interactions (34 interactions which involve waiting and 32 interactions which do not involve waiting) respectively associated with 30-second think time, with other normal users (system programmers). On the other hand, the system response time was measured by manually inputting the lines contained in the first half of the same script, MFTN3. The system response time cannot be measured accurately by a PDP-8 user simulator system because the current time contained in a Multics ready message does not have enough accuracy.

2. The CPU time and the number of page faults

The results obtained from the daily Multics performance measurement on the 3.1.1 system are summarized in Figures 1, 2 and 3.

The behavior of the CPU time with an increasing number of simultaneous users is shown in Figure 1. It is observed that the CPU time increases gradually as the number of users increases. Similarly, the behavior of the number of page faults encountered in running the standard script, with an increasing number of simultaneous users, is shown in Figure 2. In this case the number of page faults increases fairly rapidly as the number of users increases.

Figure 3 indicates the relationship between the number of page faults and the CPU time, and the behavior of the components of the CPU time with an increasing number of page faults.



3.1.1

Figure 1 Effect of the number of users on the CPU time

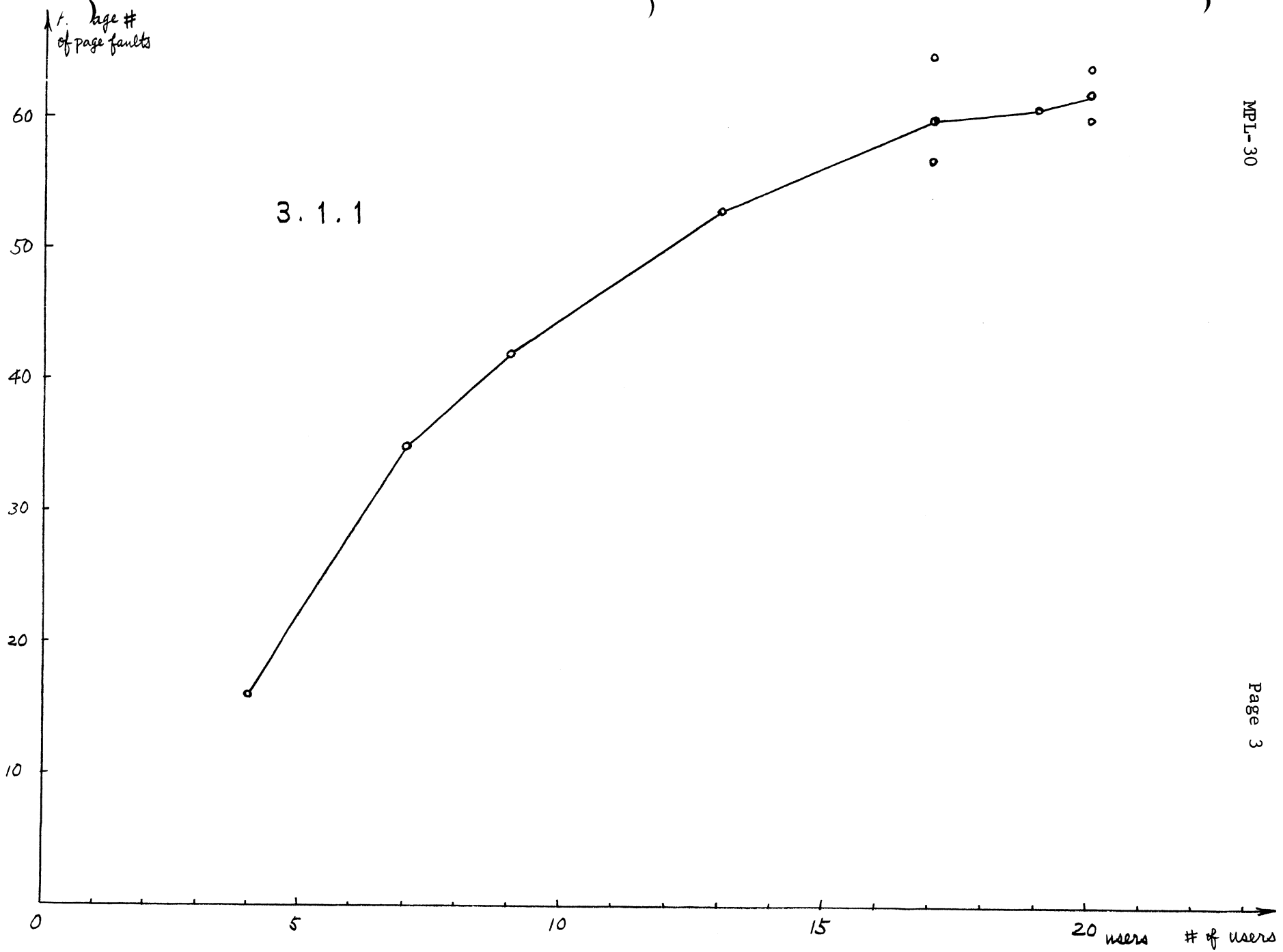


Figure 2 Effect of the number of users on the number of page faults

Since the four-user experiment in the graph was done with three special backup-and-housekeeping system users simultaneously, the CPU time and the number of page faults observed in this experiment could be regarded as the smallest possible CPU time and the smallest possible number of page faults. The measurement of this experiment shows that the average CPU time per interaction and the average page-fault- processing time per interaction are respectively 545 milliseconds and 120 milliseconds (the average page-fault-processing time per single page fault is assumed to be 7.5 milliseconds). The result of this four-user experiment done by R. Feiertag inside the processing system suggests that the average CPU time per interaction and the average page-fault-processing time per interaction are respectively 518 milliseconds and 111 milliseconds (See MPL-26). His experiments also suggest that the run time (defined to be non-fault CPU time) stays almost constant (a little over 300 milliseconds per interaction) as the number of users increases.

Therefore, it is estimated that the minimum per interaction page-fault-processing time is about 120 milliseconds and that the run time stays around 310 milliseconds. Assuming again that an average of 8.0 millisecond page faults are introduced as the number of users increases, the components of CPU time is estimated as shown by the broken lines in Figure 3. It is observed that the page-fault-processing time and other fault-processing time increase linearly and the percentage of the run time in the CPU time decreases from 57% to 28% as the number of users on the system increases from four to twenty.

---

* The reader should note that the "interaction" may be either an interaction which involves waiting or an interaction which does not involve waiting. An average CPU time and an average number of page faults are obtained by dividing a total CPU time and a total number of page faults respectively by the total number of interactions of either kind. However, an average system response time is obtained by dividing a total system response time by the total number of interactions which involve waiting.

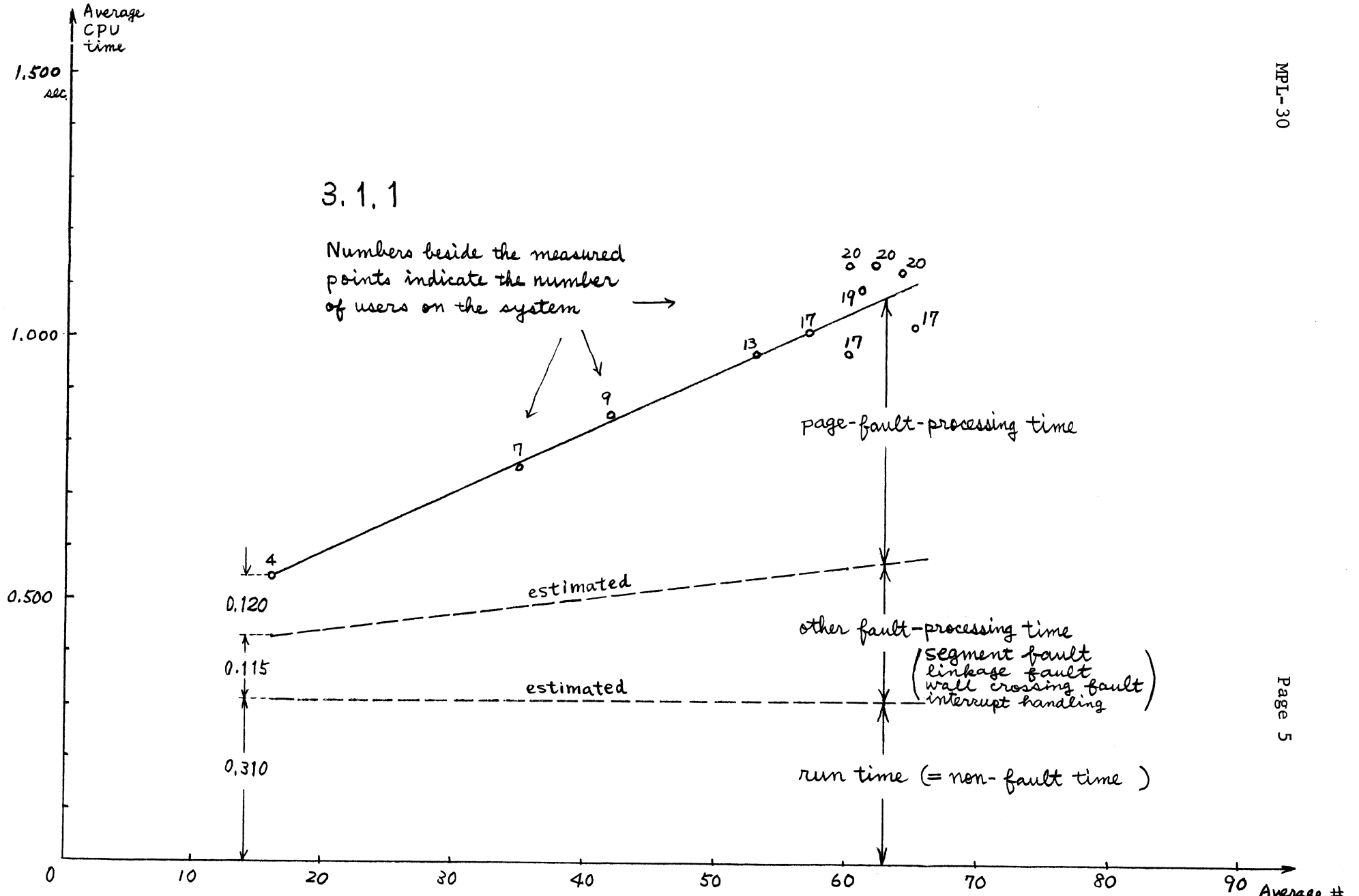


Figure 3 CPU times vs. Number of page faults

3. The system response time

The lines contained in the first half of the standard script were manually inputted with proper think times on the console and the system response time at each interaction was measured in four cases. There are 17 interactions which involve waiting in the first half of the script. The results are shown in Table 1. It is seen that the system response to the heavy command such as FORTRAN is very slow. Therefore, two averages, namely the average of all responses and the average of responses except the ones to FORTRAN, are shown. Similarly, Figure 4 shows the behavior of the system response time measured in these experiments with an increasing number of users. It should be noted that the average system response time at 15 users is about 19 seconds, while the typical system response time of CTSS is about 5 or 6 seconds.

4. Some other observations

(a) The 3.1.1 system performance realtive to CTSS.

The twin script, 6MAD1, was run on CTSS and the following result was obtained:

Run #	CPM01
Date and time	8/1/69 9:28
System status	(1) 17 simultaneous users (2) 116 entries in a user file directory (3) disk file system not reloaded recently
Total real time	2454 sec.
Total run time	42.960 sec.
Total swap time	27.975 sec.
Total CPU time	70.935 sec.
Average run time	0.650 sec.
Average swap time	0.423 sec.
Average CPU time	1.073 sec.
Average system response time	6.3 sec.

Because the average Multics CPU time at 17 users is about 1.02 seconds from Figure 1, MULTICS and CTSS are quite comparable in CPU time. On the other hand, it is again observed that the average system response time of CTSS is much shorter than that of MULTICS.

(b) The effect of think time on the performance

The experiment to see the effect of think time length on the system performance (the CPU time and the number of page faults) was done. In this experiment a certain command was issued periodically with a period

Table 1 System response time

Interaction number	MULTICS system response time (sec.) number of users 6	10	13	17	CTSS system response time # of users 15	Interaction number
edm 1	8.5	5.5	7.0	7.0	2.0	1 edl
2	1.5	2.5	6.5	2.5	4.5	2
3	2.0	3.5	8.5	9.0	9.0	3
fortran 4	10.5	25.0	93.5	117.0	6.5	4 mad
edm 5	1.5	10.5	8.5	15.0	4.5	5 edl
6	2.0	7.5	8.0	4.0	3.5	6
7	1.5	7.5	7.0	4.0	2.0	7
8	1.5	6.0	11.0	3.0	2.0	8
9	1.5	6.5	10.5	2.5	13.0	9
10	1.5	8.5	8.0	19.5	2.5	10
fortran 11	9.5	26.0	58.5	110.0	6.0	11 mad
rename 12	4.0	9.5	7.0	7.5	3.5	12 rename
print 13	2.0	6.0	9.5	13.5	2.5	13 print
a.-prime $ prime 14	2.5	5.5	3.5	10.5	9.5	14 Loadgo
List 15	4.0	4.5	7.0	9.5	3.0	15
setacl 16	6.5	6.0	15.0	12.5	2.0	16 Listf
remove 17	4.0	4.5	10.0	10.0	16.0	17 delete
sum	64.5	145.0	279.0	357.0	92.0	sum
average (all)	3.8	8.5	16.4	21.0	5.4	average (all)
average (except fortran)	3.0	6.3	8.5	8.7		

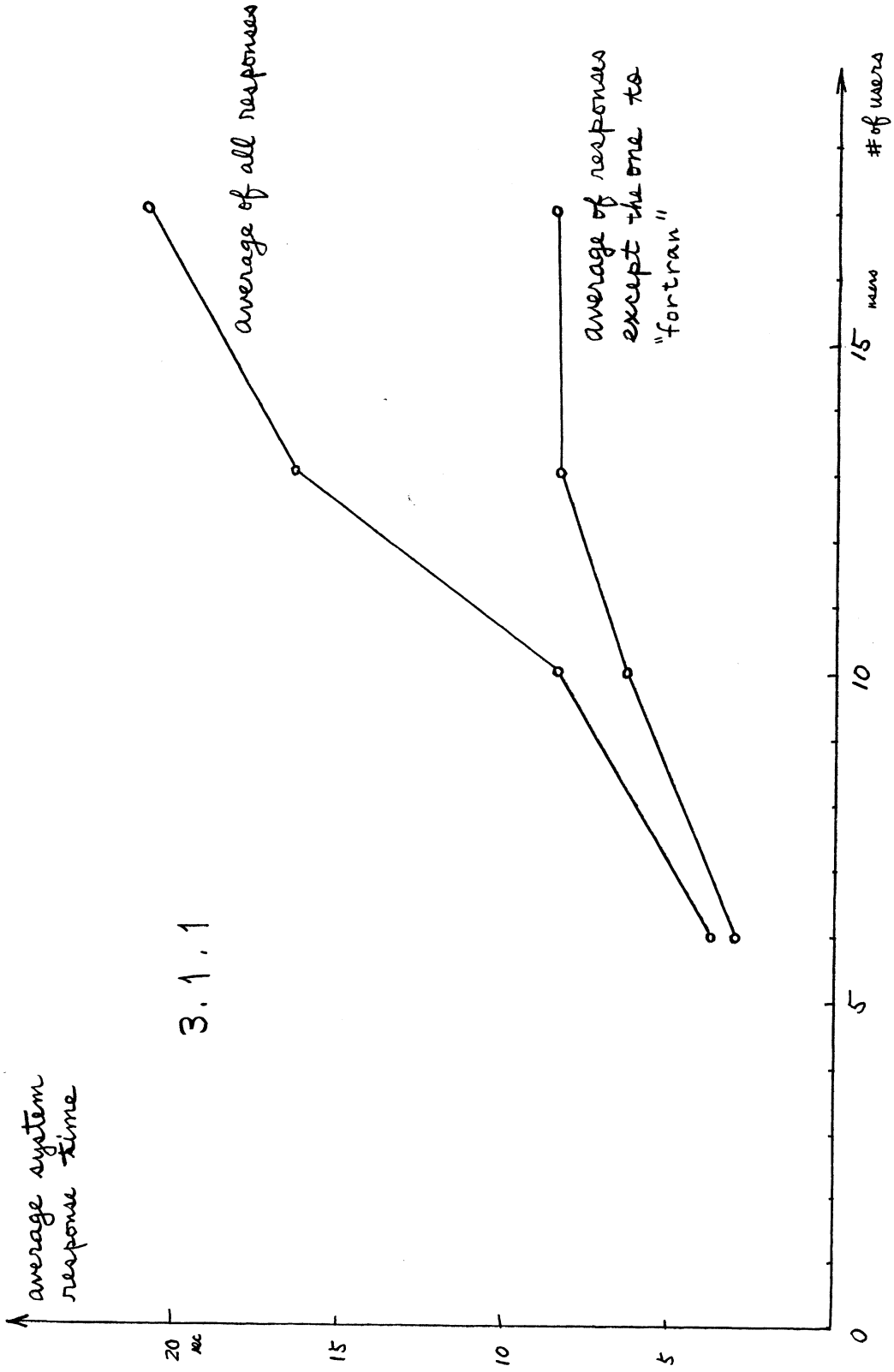


Figure 4 Effect of the number of users on the system response time

of think time ten times for each fixed length of think time during the console session. But any significant result has not yet been obtained presumably because of an improper setting of the think time length and the number of experiments. However, further experiments of this kind are planned.