University of Edinburgh

Department of Computer Science

EMAS Performance Measurement experiments using a remote terminal emulator

by

J.C. Adams and A.S. Wight

Internal Report CSR-57-80

James Clerk Maxwell Building,
The King's Buildings,
Mayfield Road,
Edinburgh,
EH9 3JZ.

February, 1980

EMAS - Performance measurement experiments using a remote terminal emulator.

J.C. Adams* and A.S. Wight
Department of Computer Science
University of Edinburgh
Scotland.

We describe the system measurement phase of a performance evaluation exercise. The performance of the Edinburgh Multi-Access System is measured during a set of experiments run within a controlled environment using a PDP 11/40 as a Remote Terminal Emulator (RTE). The system is monitored running on a fixed configuration with a number of different user loads and also with a fixed user load running on a variety of hardware and software configurations. Results obtained from internal system monitoring and RTE level monitoring are presented.

Key words: Remote terminal emulation; system configuration; time-sharing; virtual memory; workload characterisation.

Acknowledgement

This paper was presented at the 15th meeting of the Computer Performance Evaluation User's Group, San Diego, October 15-18, 1979.

* Now at Digital Equipment Corporation, Maynard, U.S.A.

1. Introduction

The Edinburgh Multi-Access System

The Edinburgh Multi-Access System (EMAS) [1][1] is a virtual memory time-sharing operating system designed and developed at Edinburgh University. EMAS currently provides the main interactive service for the university, supporting 90 simultaneous users on two ICL 4/75 mainframes [1]. This service has been available for seven years and during this period the performance of the system has been studied both in normal use and under experimental conditions. [2,3,5]. This paper describes a series of measurement experiments using the Edinburgh Remote Terminal Emulator (ERTE) to provide a controlled environment. EMAS was always intended to be a system suitable for studying the performance of an interactive virtual memory operating system in service. It is a modular system written in a high level language [4] and as intended these features have provided easy modification which has proved very useful. Dedicated machine-time for experimentation has been available on a regular basis. In particular research, funded by the Science Research Council, into many aspects of performance evaluation has capitalised on this favourable environment [5]. To study the system in a controlled fashion a remote terminal emulator was developed as a suitable method of providing workloads. Measurements taken at various levels can be used to attempt to identify reasons for variations in observed performance.

We describe two sets of experiments. In the first set the workload remains fixed and the performance of various system configurations monitored. In the second the system configuration is fixed and the workload allowed to vary. We restricted the system changes to:

a) process loading algorithm.
b) available main memory.
c) number of drum channels.

[1] Figures in brackets indicate the literature references at the end of this paper.

The workload used throughout was a benchmark defined from very detailed measurements of EMAS in normal use.

2. Description of Experiments

Two sets of experiments were as follows:

1. A factorial experiment in which the workload remained fixed while three system configuration parameters were varied.

    a) The amount of main memory available to the system. This was set by software test flag at initial system load time and ranged over the values ⅝, ¾, ⅞, 1 Mbyte.

    b) The number of drum channels available. This parameter was varied by means of a hardware switch set before loading the system (which is self-reconfiguring) and allowed either one or two channels to be used.

    c) The process scheduling algorithm. This had two variations. In the first processes were preloaded using a Working Set Replacement (WSR) scheme in conjunction with the standard EMAS category scheme (1). In the second a Pure Demand Paging (PDP) strategy was employed with a modified category scheme.

2. A series of experiments where the system configuration remained fixed (1 Mbyte of main store, 2 drum channels, WSR algorithm) and the number of simultaneous interactive users was varied from 24 to 48.

All experiments were carried out in a standard fashion (see figure 1). EMAS was loaded with an appropriate hardware configuration. After starting ERTE pseudo-users started to log on. The timing of the run was from the first pseudo-user logging on. Eight minutes after this various internal supervisor monitoring tables were cleared and a thirty-one minute measurement period was under way. Eight minutes were adequate for all pseudo-users to log on and the system to reach a suitably steady state. Two minutes after the start of the measurement window a detailed supervisor event trace was automatically switched on. At the end of the measurement window all the internal supervisor monitoring was automatically dumped and ERTE halted and made ready for the next experiment. All transactions between ERTE and EMAS are automatically time stamped and logged by ERTE for later analysis.

Even with defined workloads and standard system configurations there will be some variability introduced by considerations such as random positioning of rotating memories etc. Certain experiment runs were repeated in an attempt to obtain an estimate of the possible error introduced by this variability.

3. Workload Characteristics.

The workload used throughout was derived from an extensive measurement programme carried out on EMAS in 1974 [2]. In particular information about the following characteristics of interactive users was obtained:-

Types of command issued
Resources used per command (CPU time, paged I/O, file sizes, terminal I/O)
Think times
Typing rates.

Such information was collected over a period of two years and it was found to be very stable for continuously busy periods with for example discrepancies of the order of one per cent when comparisons were made between the distribution of commands measured over a 24 hour period and with that determined from an included continuously busy period (3-5 p.m.). On this basis it was decided to record one particular two hour session in the greatest possible detail for as many variables as possible. The data obtained in this way was transcribed into a set of thirty-two distinct scripts. Each script contained all the commands issued by a user and the think times between commands. The individual command totals were identical with those in the observed period and the programme files, contributed by users, were selected to exhibit the distribution of compilation and execution characteristics which had been observed. These scripts were used by ERTE [6]. The permanent base files, programme and data, used during the runs of the benchmark are never modified, all editing is done by copying from a base file into a temporary file which will eventually be destroyed at the end of any run.

Since the research has concentrated largely on studying the system under an exclusively interactive load the constructed workload does not contain any background or batch component.

The original 32 scripts formed the workload used in experiment 1. However experiment 2 called for a variable workload including the possibility of many more than 32 simultaneous users as provided by this benchmark workload. To meet this requirement the MAVIS benchmark construction suite was used [7]. This gives a convenient way of generating loads with specific characteristics from a set of predefined building blocks (in this case based on the original benchmark).

A more detailed description of the contents of the benchmark is shown in figure 2. It may be noted that the workload has a strong bias towards programme preparation and interaction with running programmes. A large element of the input under the heading miscellaneous is interaction with user programmes e.g. supplying data. The use of CPU time is also heavily skewed, most (61%) being taken up with executing user programmes, followed by compilation of programmes, (24%), whilst editing only absorbs 8% of the total CPU time.

4. Performance Measures

No single measure can adequately represent the performance of a complex computing system. The combination of easily modified software and an attached remote terminal emulator provides an environment in which to collect data simultaneously within the system and at the 'terminals' and in which to attempt to relate observations at the user and system level. We are interested both in the level of service obtained by users and in identifying the causes of variations in the observed service. For a comprehensive discussion of the collection of user level measurements see the recent paper by Abrams and Treu [9]. Details of other emulators are surveyed in Watkins and Abrams [8].

a) User level measures.
All 'terminal' input and output is recorded by ERTE and analysed off line to obtain the following measures.

i) Reaction.
Time from stimulus given to system, i.e. carriage return at end of line of input, until first character of response is typed.

ii) Throughput.
Reactions per minute.

iii) Satisfaction.
Number of reactions which fall beneath some level of satisfactory response (e.g. 2 seconds for interaction).

Response time in our definition is the time from stimulus until the command is completed and the system issues another prompt, i.e. is ready for the next command. We can obtain distributions etc. for many measures but for general consumption these have to be reduced to single figures of merit.

b) System level measures.
EMAS includes many software monitoring features and

it is difficult to add more if required.

i) CPU accounting.
Time spent in Supervisor, User and Idle states.
Details of where Supervisor time is spent.
Paging rates.

ii) Samples.
The values of certain supervisor variables and queue length are recorded every ten seconds and are available as a performance summary either on demand or at the end of an experiment.

iii) Event trace.
A very detailed record of all scheduling and paging events is recorded for off-line analysis.

We present below some of the results obtained. Note that each experimental run produces approximately 3.5 megabytes of data.

ERTE gives the possibility of studying the performance of many components of the total system in a controlled fashion. Changes in hardware, software and the communications network can be evaluated knowing that everything else including the workload was as before.

That experiments can be repeated allows us to investigate unexpected or dubious results. Duplicating certain runs also allows estimates to be made of the errors for each of the performance metrics chosen.

5. Results

The three graphs figures 3.1 - 3.3 show the results obtained from the first experiment. Keeping the number of users (32) and the workload fixed we see the benefits gained from preloading and the use of two drum channels rather than one. The differences are most obvious in figure 3.1 illustrating the satisfaction level. This is describing the effect on short interactions and for these the initial processs load time is critical and the benefits of preloading worthwhile. The graphs of figure 3.2 are not quite as smooth as one might have hoped for but display the data as collected and the same overall effect as figure 3.1. Turning to figure 3.3 we find the overall result confirmed but note that in the case of one channel with preloading the number of interactions per minute eventually drops off as the core size is increased to one megabyte because there is then insufficient drum channel capacity to deal with the traffic generated. Overall however the benefits gained from preloading are greater than those gained by adding another drum channel.

N.B. The combination of two drum channels and one megabyte of memory was not realisable at the time the first set of experiments was done.

In experiment 2 the number of users was varied on a fixed configuration. Although on figure 4.1 the results for 40 users are not consistent with those around, figure 4.2 shows clearly that the interaction rate reaches a peak around 40 users. Again the reason is trouble with the drums. Study of the associated system data shows that the drum space is filled up and as the scheduling required is done on a global basis the effects of this start to dominate the service. The careful management of core is being nullified by a form of thrashing in the drum memory.

In the first experiment the reaction time data for a number of commands was analysed using the analysis of variance technique (ANOVA) [10] calculated using an algorithm due to Yates [11]. No adjustment was attempted for the missing data for the 1 Megabyte of memory level and the analysis was done for a 3x2x2 factorial experiment. An experimental error estimate is based on the effects attributed to higher order factors and used in a simple F-test [12] to test the significance of the average effect due to the major factors upon reaction time. Tables 1, 2 and 3 display the analysis for FORTRAN compilations, editing and file listing respectively.

6. Conclusions

The emulator proves a very good, if expensive, tool for doing realistic investigations of user and system behaviour. Work is proceeding to use the insights and data gained to develop useful models of the system and users. It is also intended to use the emulator to study other systems. A first step has seen it used as a development tool in the production of a version of EMAS for an ICL 2970 machine. Here the emulator has proved invaluable in providing realistic repeatable workloads for testing at a much earlier stage than is normal in system development.

References

[1] Whitfield, H. and Wight, A.S., The Edinburgh Multi-Access System. Computer Journal, Vol. 16, pp 331-346, 1973.

[2] Adams, J.C. and Millard, G.E., Performance Measurement on the Edinburgh Multi-Access System. Proceedings of the International Computing Symposium, Antibes, pp 105-112, 1975.

[3] Adams, J.C., Performance Measurement and Evaluation of Time-Shared, Virtual Memory Systems. Ph.D. Thesis, University of Edinburgh, 1978.

[4] Stephens, P.D., The IMP Language and Compiler. Computer Journal, Vol. 17, pp 216-223, 1974.

[5] Science Research Council grant BRG 93960. Modelling multi-access systems using detailed observations of user behaviour and system performance.

[6] Adams, J.C., Currie, W.S. and Gilmore, B.A.C., The Structure and Uses of the Edinburgh Remote Terminal Emulator. Department of Computer Science, University of Edinburgh Report CSR 12-77. Software-Practice and Experience, Vol. 8, pp 451-459, 1978.

[7] Brebner, G.J., MAVIS - A Script Generator for the Edinburgh Remote Terminal Emulator. Department of Computer Science, University of Edinburgh. Internal Report, 1977.

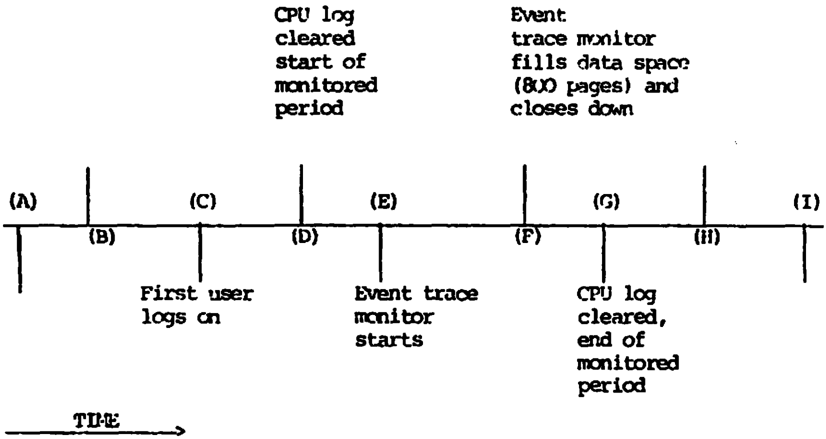
[8] Watkins, S.W. and Abrams, M.D., Survey of Remote Terminal Emulators. U.S. National Bureau of Standards Special Publication 500-4, 1977.

[9] Abrams, M.D. and Treu, S., A Method for Interactive Computer Service Measurement. C.A.C.M. Vol. 20, pp 936-944, 1977.

[10] Mendenhall, W., The Design and Analysis of Experiments, Wadsworth, 1968.

[11] Yates, F., The Design and Analysis of Factorial Experiments, Commonwealth Bureau of Soils, Technical Communication. No. 6, 1937.

[12] Johnstone, N.L., and Leone, F.C., Statistics and Experimental Design in Engineering and Physical Sciences, Wiley, 1964.



A - IPL 4/75 with appropriate hardware/software configuration.
B - Start emulator.
H - Stop emulator.
I - Copy event trace data to EMAS file.

C→D 8 minutes
D→E 2 minutes
D→G 31 minutes (monitored period)

The close of the event trace monitor (F) will normally take place before the end of the monitored period (G) when the data space is filled. If not then the event trace monitor is shut down at (G).

Figure 1 Normal Experimental Run

	Experiment 1	Experiment 2
Logon and Logoff	6.4%	6.9%
Compilation (IMP, FORTRAN)	4.0%(2.8,1.2)	4.0%(2.5,1.5)
Initiating Edits	5.2%	5.0%
Input to Edit sessions	40.4%	46.0%
Setting up streams	3.2%	4.8%
Running programmes	3.9%	4.3%
File and Library manipulation	9.7%	5.9%
Miscellaneous	21.4%	15.6%
Mean number of characters/input	10	10
(Mean number of characters out/response	75	75
Mean think time	12 secs.	13 secs.
Mean time in user wait state (think + type)	16 secs.	17 secs.

Figure 2 Input from Benchmark
(figures stated as percentage of all input lines.)

Table 1. ANOVA Table for Reaction Times-FORTRAN Compilation

SOURCE	AVERAGE EFFECT	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	MEAN SQUARE RATIO
MEMORY					
7/8 3/4 3/4 5/8 7/8 5/8	9.85) 42.50) 52.65)	2089.28	2	1044.64	36.95****
CHANNELS	22.35	499.52	1	499.52	17.67***
ALGORITHM	28.12	790.55	1	790.55	27.96***
HIGH ORDER FACTORS		197.89	7	28.27	
(ERROR ESTIMATE)					
TOTAL		3577.22	11		

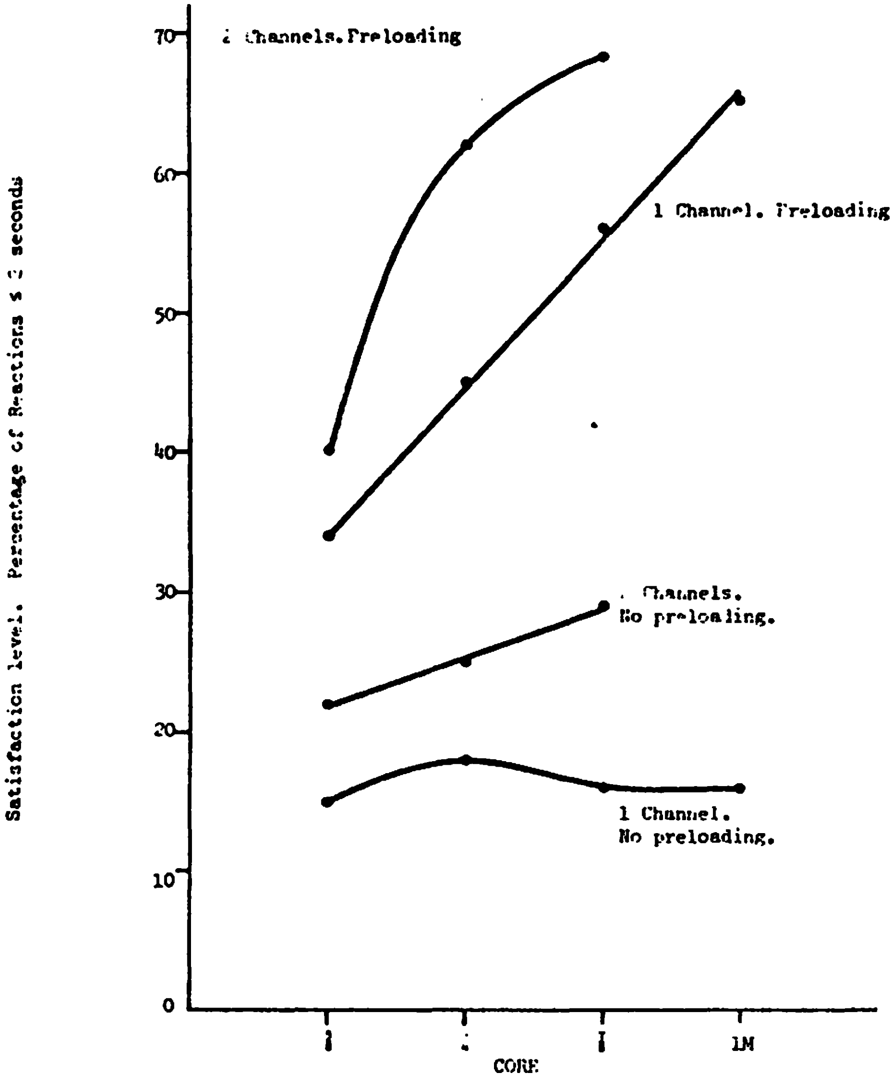
Table 2. ANOVA Table for Reaction Times-Editing

SOURCE	AVERAGE EFFECT	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	MEAN SQUARE RATIO
MEMORY					
7/8 3/4 3/4 5/8 7/8 5/8	2.17) 4.78) 6.95)	33.70	2	16.85	13.42***
CHANNELS	3.56	12.60	1	12.60	10.04**
ALGORITHM	8.72	75.98	1	75.98	60.51****
HIGH ORDER FACTORS		8.79	7	1.26	
(ERROR ESTIMATE)					
TOTAL		131.08	11		

Table 3. ANOVA Table for Reaction Times-Listing Files

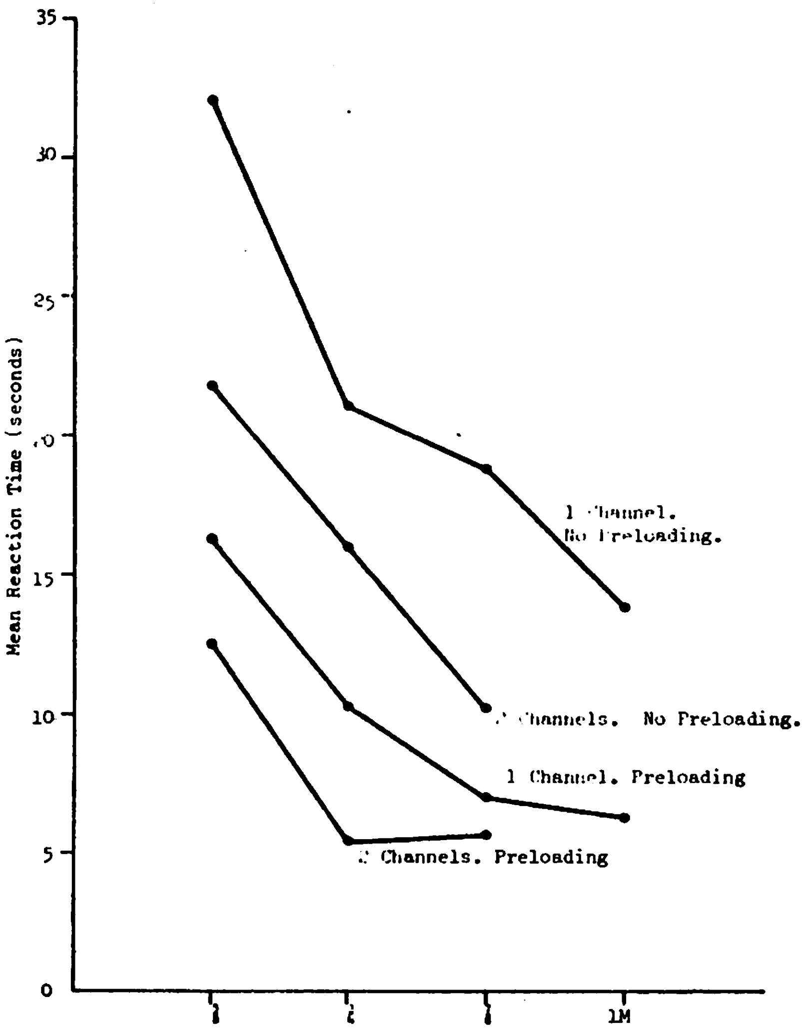
SOURCE	AVERAGE EFFECT	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	MEAN SQUARE RATIO
MEMORY					
7/3 3/4 3/4 5/8 7/3 5/8	5.80) 8.27) 14.08)	133.43	2	66.72	16.01***
CHANNELS	11.67	136.11	1	136.11	32.66****
ALGORITHMS	16.47	271.15	1	271.15	65.05****
HIGH ORDER FACTORS		29.18	7	4.17	
(ERROR ESTIMATE)					
TOTAL		569.87	11		

**** SIGNIFICANT AT 99.9% level (F-test)
*** SIGNIFICANT AT 99% level (F-test)
** SIGNIFICANT AT 97.5% level (F-test)



Satisfaction level. Percentage of Reactions ≤ 2 seconds
70
60
50
40
30
20
10
0
2 Channels. Preloading
1 Channel. Preloading
2 Channels. No preloading.
1 Channel. No preloading.
1M
CORE

Figure 3.1



35
30
25
20
15
10
5
0
Mean Reaction Time (seconds)
1 Channel. No Preloading.
2 Channels. No Preloading.
1 Channel. Preloading
2 Channels. Preloading
1M
CORE

Figure 3.2

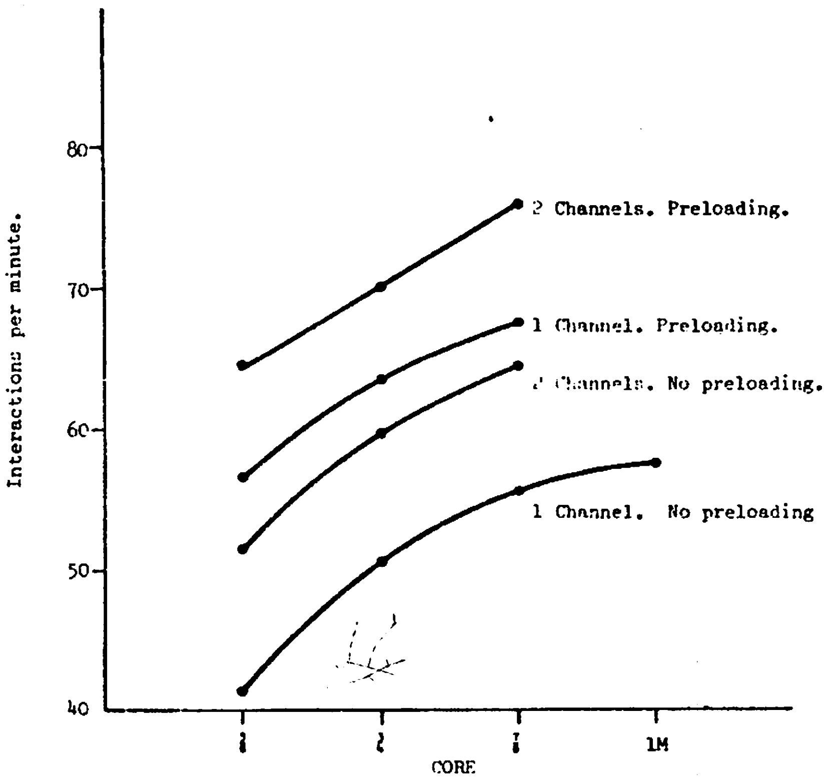

Interactions per minute.
80
70
60
50
40
2 Channels. Preloading.
1 Channel. Preloading.
2 Channels. No preloading.
1 Channel. No preloading
1M
CORE

Figure 3.3

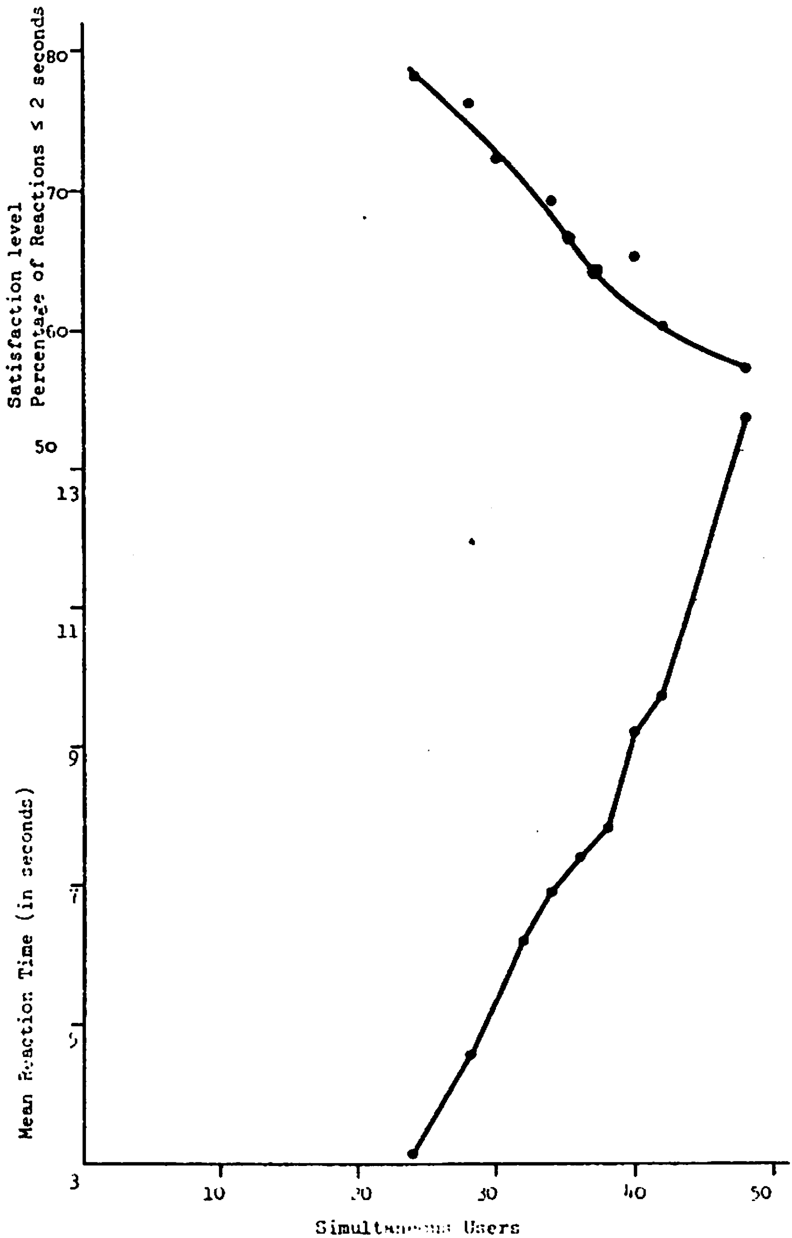
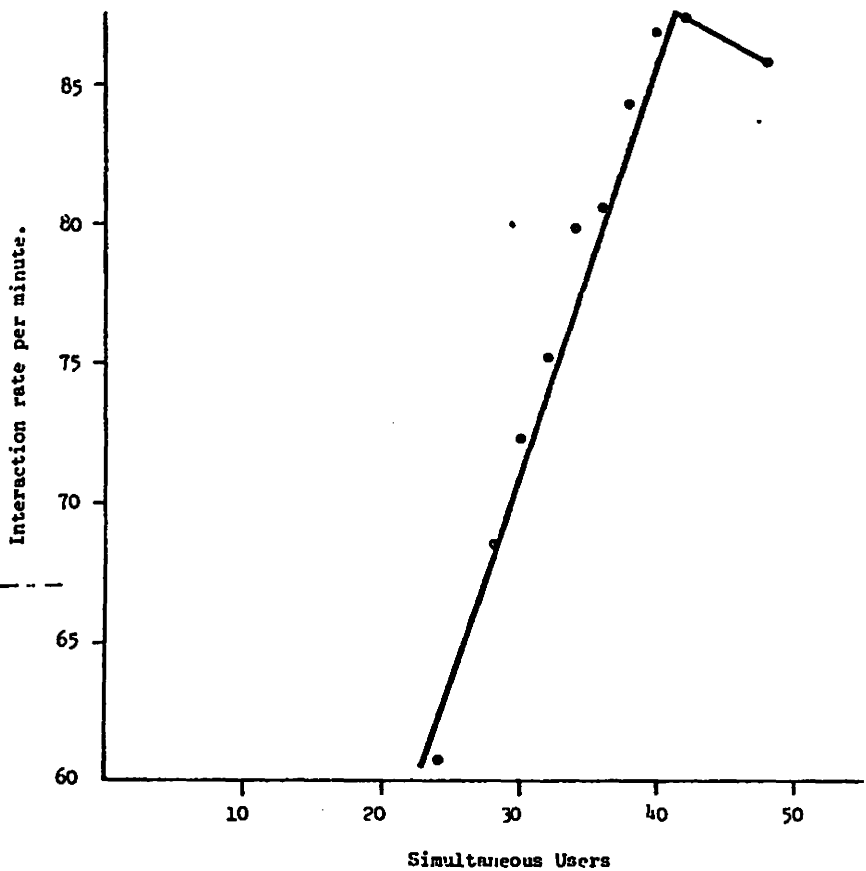

Satisfaction level
Percentage of Reactions ≤ 2 seconds
80
70
60
50
13
11
9
7
5
3
Mean Reaction Time (in seconds)
10
20
30
40
50
Simultaneous Users

Figure 4.1


Interaction rate per minute.
85
80
75
70
65
60
10
20
30
40
50
Simultaneous Users
Figure 4.2