INTERACTIVE GRAPHICS IN PERFORMANCE MEASUREMENT

B.N. MARTENSEN

BELL-NORTHERN RESEARCH

ABSTRACT

This paper describes an interactive graphics program developed at Bell-Northern Research, to present computer analysts and management with graphic views of the performance of the IBM 370/168. Vital performance statistics, gathered at regular intervals from the computer, can be analysed by a PL/1 program and graphed interactively with the use of a GRAPPLE program. Over sixty performance quantities can be graphed on a computer graphics display or plotted for more permanent use. Through this application of interactive graphics, important relationships and bottlenecks in the computer software and hardware have been easier to detect and evaluate.

SYSTÈME GRAPHIQUE À ACTION RÉCIPROQUE SOUS L'ANGLE DE LA MESURE DU RENDEMENT

B.N. MARTENSEN

RECHERCHES BELL-NORTHERN

ABRÉGÉ

Cette étude décrit un programme graphique à action réciproque mis au point aux Recherches Bell-Northern pour fournir aux analystes informaticiens et à la direction des données graphiques sur le fonctionnement de la machine IBM 370/168. Des statistiques de rendement d'importance vitale, extraites à intervales réguliers de l'ordinateur, peuvent être analysées par un programme PL/1 et présentées graphiquement d'une facon réciproque par l'utilisation d'un programme "GRAPPLE". On peut rendre graphiquement plus de soixante valeurs de rendement sur un écran d'ordinateur ou les tracer en vue d'une utilisation plus permanente. Par cette application d'un système graphique à action récipoque on a pu détecter et évaluer plus facilement d'importantes relations et goulets d'étranglement dans le logiciel et l'ensemble ordinateur.

INTRODUCTION

Bell Northern Research installed an IBM 370/168 in December 1973. This is a large time-sharing computer running the operating system VM/370. Each user of the computer is provided with a virtual computer which appears to him just like an IBM/370. A majority of these users run the CMS operating system in their virtual machines. The user's virtual machine is more flexible than the real computer. He is able to interactively change the configuration of his computer. He is able to change the memory size, attach and detach card readers, printers, disks and tapes, or even disconnect his console.

The real computer is fixed in its configuration. Presently the hardware configuration is 1 CPU, 2 megabytes of real memory, two 2305 drums on one channel (9 megabytes per drum), 28 3330 disk drives and 8 tape drives. VM/370 shares these resources amongst the users' virtual machines. VM/370 allocates time slices to those virtual machines competing for the CPU. The drums are used for storage of the memory pages (4K blocks) of the virtual machines. Paging is the mechanism by which the pages that are being worked on are transfered to the real memory and by which least active pages are placed back on drum. A disk is used for extra pages when the drums fill up. The other disks are divided up to provide each user with his own disk storage. The tape drives are attached to virtual machines when they are needed.

The computer has two states of operation, supervisor and problem state. VM/370 runs in the supervisor state and the virtual machines run in problem state. Some instructions can only be executed in the supervisor state. When a virtual machine tries to execute one of these instructions, VM/370 catches and simulates it in the supervisor state. Thus VM/370 is able to maintain the appearance of many virtual machines.

Data Collection

As delivered, 70 users could simultaneously use the computer before response became noticably bad. Since then many software changes have been implemented in order to extend the capacity and life of the system. It now handles up to 120 users before response degradation.

In June 1974 we started collecting system performance data. With this data we have monitored the improvements to VM/370 and CMS. CP-67, the predecessor system to VM/370, has been extensively monitored and analysed by Y. Bard (Ref. 2) and by Rodriguez-Rosell (Ref. 3). Very little definitive literature exists on the performance of VM/370.

The performance statistics are collected by a virtual machine which runs disconnected (no on-line console). Every 3 minutes the machine runs a program which acquires data from the VM/370 Control Program (CP) and data from each virtual machine's control block. This data is time stamped and placed on disk storage. There are presently 63 different data items collected. These variables cover a wide range of system properties. A sample of such data items include: memory available for paging, paging rates, drum pages in use, multiprogramming level (queue length), CPU utilization, wait time either paging or I/O, throughput rates, and the number of users on the system. The data can be categorized into two classes, instantaneous and cumulative data. Such variables as number of users or queue length are instantaneous values measured at the sample time, where as page reads and writes or CPU utilization are accumulated over the 3 minute sample time. Instantaneous data items generally have a greater variance than accumulated ones.

Graphing the Data

We required a means to selectively view the large quantities of data being produced. This program had to be easy to use, yet be powerful enough to present the data in a concise form. For these reasons an interactive graphics display approach was taken. GRAPPLE (Graphics Applications Programming Language) Ref. 1, was the obvious choice for the development of the graphics software because it was developed and is being extensively used at BNR. The GRAPPLE program interfaces to a PL/1 program which takes the data from the disk and performs a preliminary analysis. This consists of calculating rates from cumulative data and producing new data items which are functions of two or more collected data items. The result is a set of 72 data items that can be displayed. Only the requested data is passed to the GRAPPLE program. The analyst who wishes to look at the data interacts with the GRAPPLE program through a menu of commands. This menu lists the 72 data names and a number of menu entries for invoking functions as shown below.

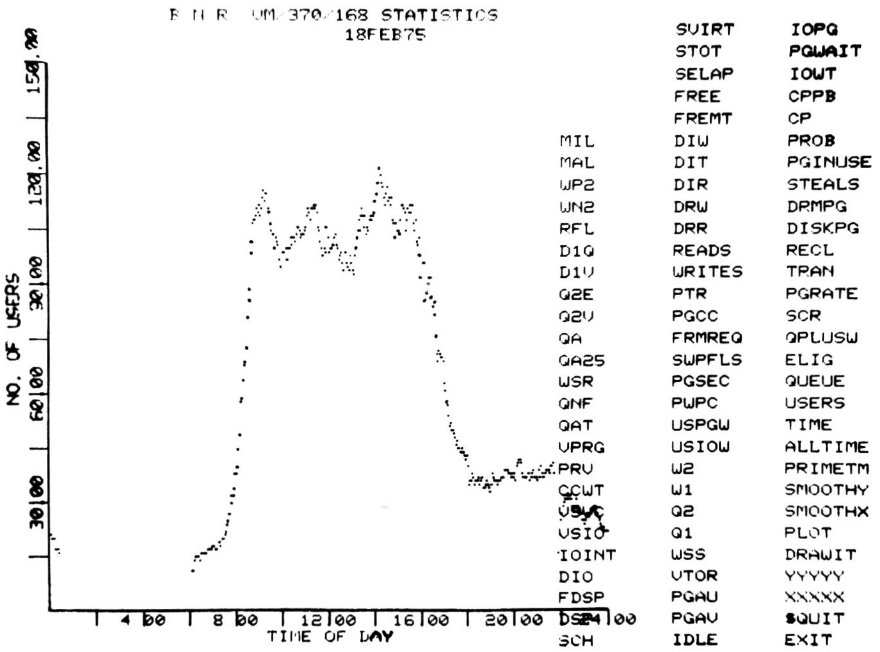
MENU name	FUNCTION performed
XXXXX	Places a data item on the horizontal(X) axis.
YYYYY	Places a data item on the vertical(Y) axis.
DRAW	Draws the graph of the Y versus X axis data.
PLOT	Adds the most recent graph to a file which

can be plotted later on a hard copy device.
SMOOTHX Place best fit line segments through the points.
SMOOTHY This is produced by averaging the data in intervals along the X axis or Y axis.
PRIMETM Use only prime time data(8.00am to 4.00pm)
ALLTIME Use the whole days data.

The day after the data is collected it is analyzed and displayed. To produce a graph, for example Fig. 8, of paging rate versus multiprogramming level an analyst would select the menu item PGRATE and then the YYYYY function. Similarly the menu item QUEUE (our synonym for multiprogramming level) would be selected and the XXXXX function invoked. DRAW would be selected and the resulting graph would appear on the display. An averaged line can be fitted through the data with the selection of SMOOTHX or SMOOTHY. The resulting display can be added to the plot file by selecting the PLOT function. Since the graphics device is usually a Tektronix storage tube a hardcopy picture can be produced on an attached hardcopy unit. This is of lower quality than a plot.

An additional feature of this system is that the averaged lines (produced by SMOOTHX or SMOOTHY) for other data items can be superimposed on the display. This enables relationships to be observed between many data items at one time.

A typical display is shown below. The left two columns of the menu are not displayed when a graph is drawn because of the overlap of menu and graph. The EXIT menu item is used to obtain the extra two columns.



RESULTS

The ability to draw performance curves interactively gives this system tremendous potential. An analyst can view over 2500 different graphs, though a large quantity of these would be meaningless because no relation exists between the data items. However many graphs show strong correlations, the strongest being page write rate versus page read rate. This is a straight line of slope 0.8. This means that 80% of the pages read from drum are written back on to drum because of modification while in memory.

One change which has extended the life of the system is page release. This involved modifications in the CMS operating system to indicate to CP pages no longer being used. VM/370 was then able to use the released drum space for more active pages. This meant that the average number of drum pages per user came down from 50 to 35 and allowed 20 more users to logon before pages were allocated on disk. Fig. 1 shows data before the change and Fig. 2 after the change. The difference in slope of the two graphs indicates the performance gain. Fig. 1 also shows a system bug whereby the count was not being correctly maintained. However, the slope of the graph is still accurate.

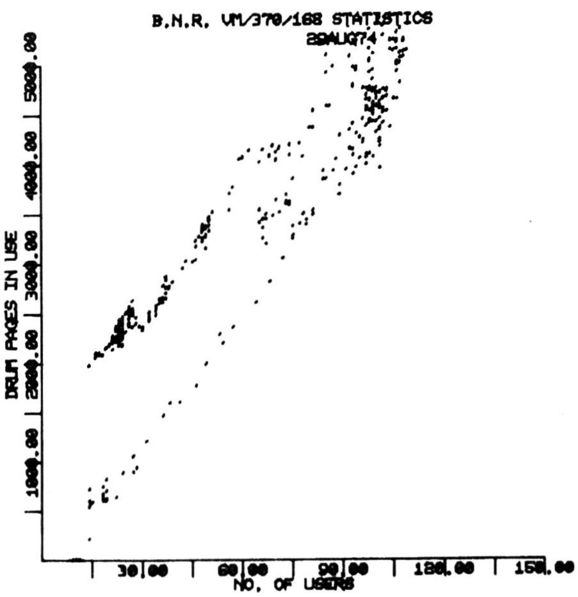


Fig. 1

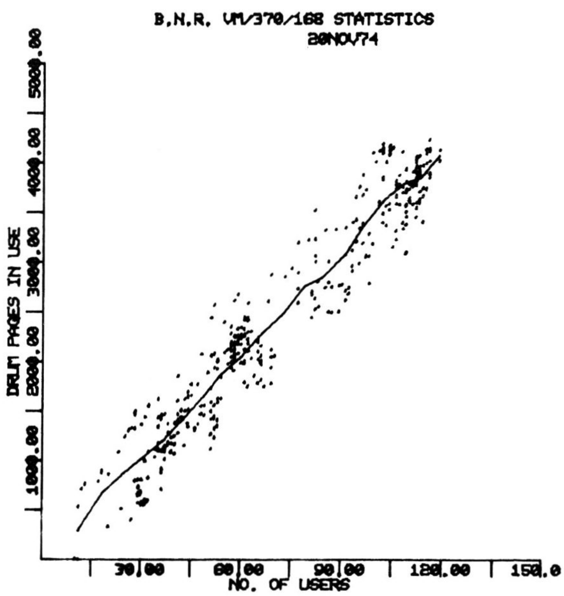


Fig. 2

We noticed that less work was being done by the system when the multiprogramming level increased beyond a certain point. (Fig. 3 and 4) The graphs show problem state time decreased but supervisor overhead stayed constant as the queue length grew. This situation was caused by a tendency for the scheduler to over commit main memory resources. (Fig 5 and 6)

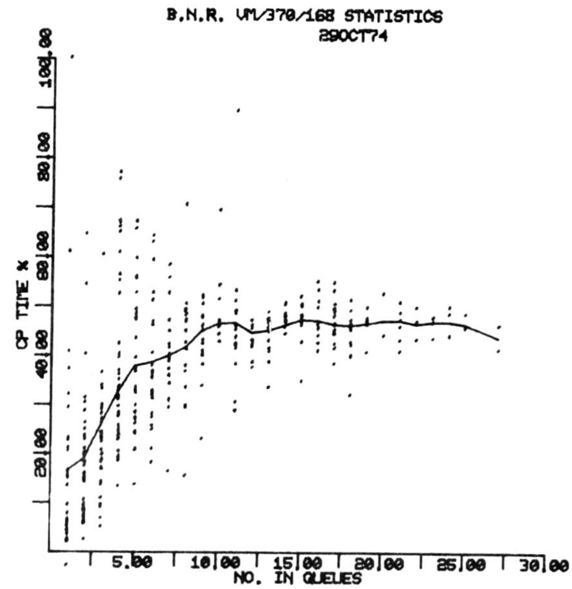


Fig. 3

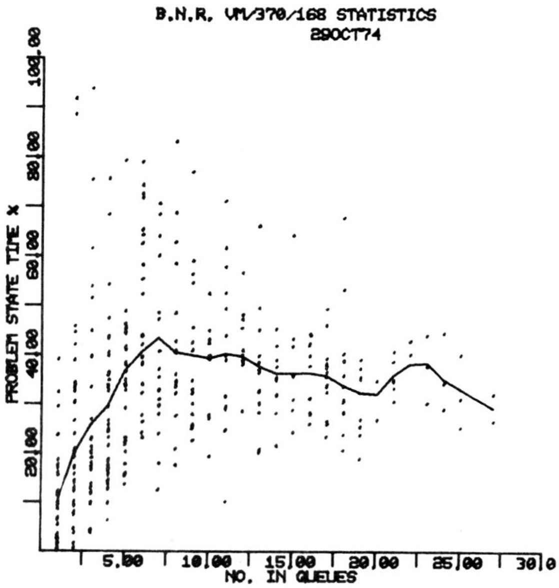


Fig. 4

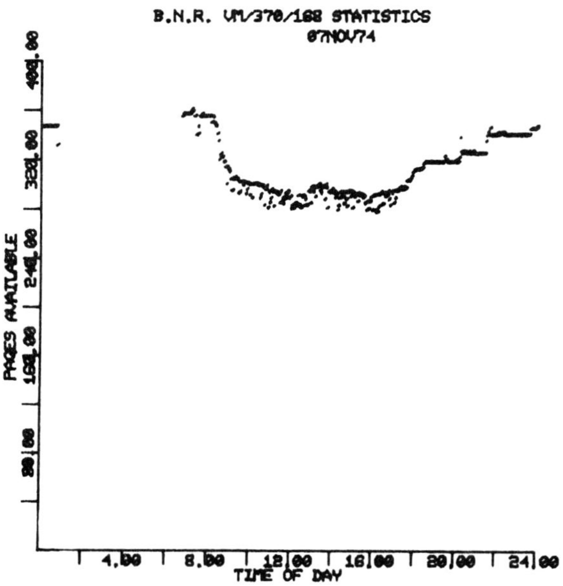


Fig. 5

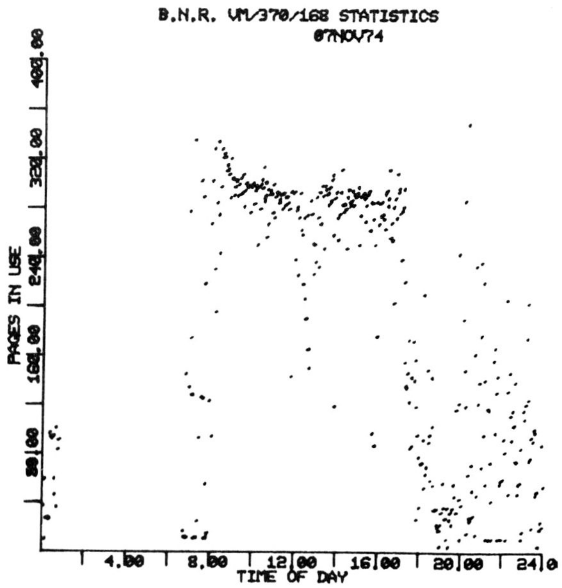


Fig. 6

We also observed that for every virtual machine scheduled, another would go into page wait. (Fig. 7) This caused an increased demand for paging which resulted in page wait time. Our first order solution to this was to restrict the multiprogramming level to 15. This improved the situation by preventing the degradation of the system. We are presently looking at a better working set size evaluation to remedy this problem.

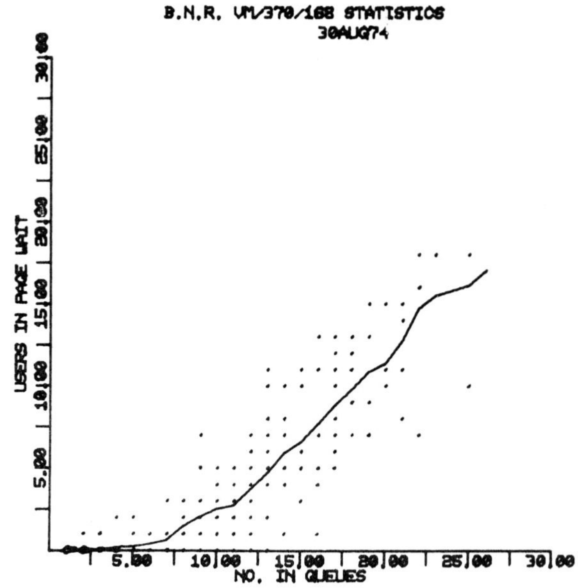


Fig. 7

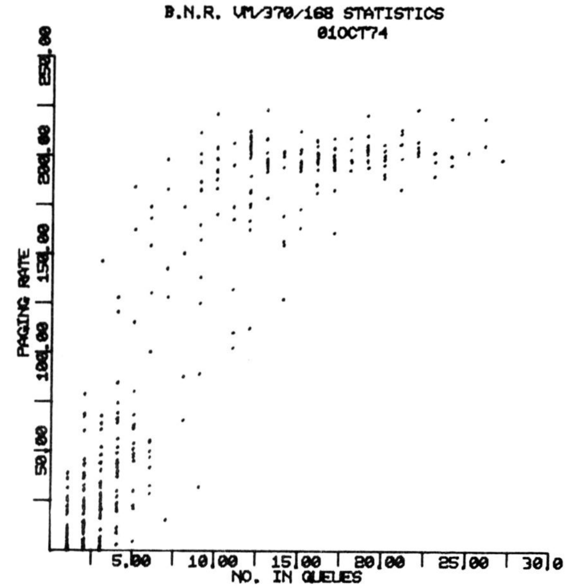


Fig. 8

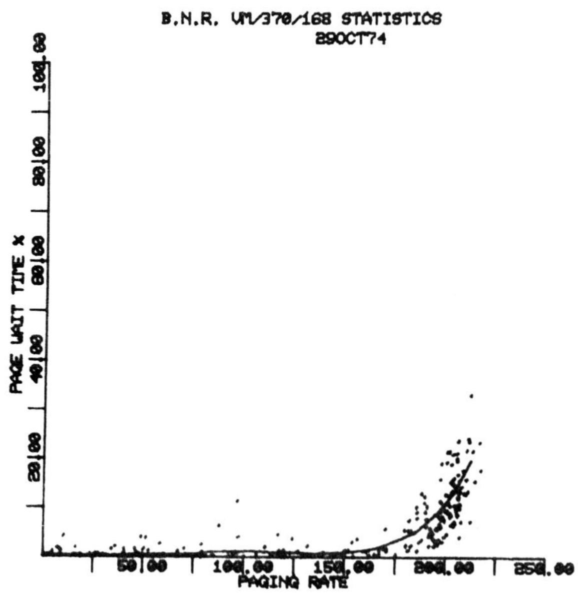


Fig. 9

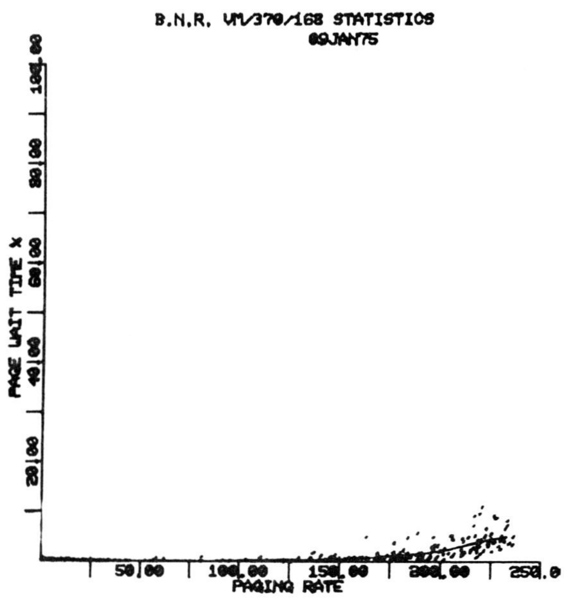


Fig. 10

An example of our ability to prove the benefit of a system change is shown in Fig. 9 and 10. With the drum paging algorithm we had in October we could not exceed a paging rate of 200 pages per second (Fig. 8) without incurring a large page wait time. (Fig 9) With a change to this algorithm supplied to us by another installation, we decreased the page wait time and increased our paging rate. (Fig 10)

Even with these changes our major bottlenecks still remain memory and paging. Presently the CPU spends 10% to 15% of its time in I/O or page wait. We anticipate major system changes during the next year that should reduce this inefficiency. We hope to show dramatically the effect of each change.

CONCLUSIONS

The objective of providing the analyst with a method to view performance data interactively was achieved. The graphs have provided valuable feedback concerning changes to the system and enabled evaluation of future requirements. The method does not do away with intelligent conjecture. Rapid presentation of graphs enables experimentation with such conjectures. We have been able to prove ideas both true and false about the computer system considerably faster than by other techniques.

Interactive graphics will be playing an ever increasing role in performance measurement. With the advent of cheap minicomputers and microprocessors now available, real-time graphical performance monitoring of software and hardware is becoming economically practical. Experimental systems such as GDM (Ref. 4), a monitor for Multics and SPY (Ref. 5), a monitor for OS/360, have paved the way in this area. This approach offers two major advantages.

1/ Only data collection, no analysis, takes place in the monitored computer, thus reducing interference with the processes being measured.

2/ Interactive real time display of data as well as large statistical analyses can be carried out naturally and easily on the remote minicomputer.

REFERENCES

1/ GRAPPLE - Graphics Applications Programming Language
D.L. Williams
The 3rd Man-Computer Communications Seminar May 30 1973

2/ Performance criteria and measurement for a time-sharing system. Y. Bard
IBM Systems Journal, Vol. 10, No. 3, 1971 pp193-216

3/ The evaluation of a time sharing page demand system
J. Rodriguez-Rosell and J. Dupuy.
Proc. AFIPS Spring Joint Computer Conference, Vol. 40, 1972, pp759-765

4/ Real-time graphics display of time-sharing operating characteristics. J. M. Grochow
Proc. AFIPS Fall Joint Computer Conference, Vol. 35, 1969, pp379-386

5/ SPY - A program to monitor OS/360
R.Sedgewick, R.Stone and J.W. McDonald
Proc. AFIPS Fall Joint Computer Conference, Vol. 37, 1970, p118.