Computing systems are analyzed from a system performance profile, which shows usage of the different system components, such as channels and central processing units. Component usage data are obtained by a system monitor.

The profile indicates where a system configuration or a program might be modified to improve performance. The profile also suggests areas where more detailed monitoring and analysis appear promising.

Using system monitor output to improve performance

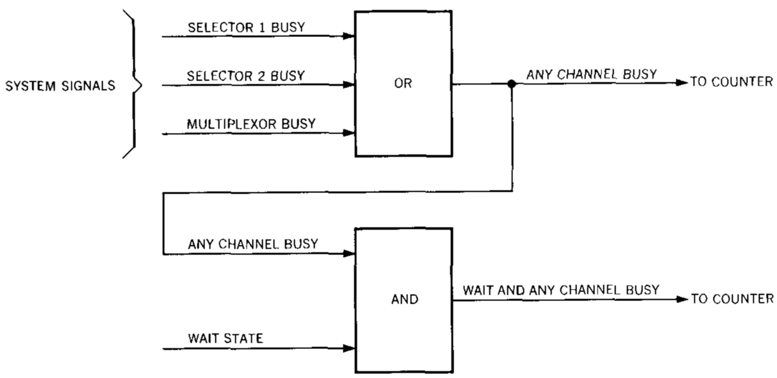
by A. J. Bonner

The wider choice of compatible central processing units and associated input/output devices now available has made feasible data processing system configurations designed for particular workload patterns. However, this flexibility has also increased the likelihood of arriving at designs that use data processing equipment inefficiently.

In the past, if a specific data processing capability were required, a computer system could be selected to meet that need. Such systems offered only a small selection of input/output devices because the means of attaching devices differed among systems. In selecting an efficient configuration, compromises between processor and I/O performance were often necessary. This barrier has been removed in current lines of compatible systems, such as the IBM SYSTEM/360,[1] by providing I/O controllers to fit all models of the processors. Thus, it is possible to organize from the available processor models and I/O devices a system configuration that is tailored to requirements at a particular installation.

The proper selection of a configuration requires knowledge of the distribution of work among the individual system components. For example, in a significant commercial data processing situation, which characteristically requires the input and output of large amounts of data, it is important that the data be stored on devices that offer a good balance of capacity, access, and transfer speeds to avoid delays. Conversely, a work load consisting of large mathematical calculations would be delayed if a central processor model with inadequate arithmetic speed were selected. Thus, the capability to measure the use of system components is necessary

Figure 1 Example of use of combinatorial logic



to make the proper selection. After a system is in operation modifying its characteristics to accommodate a changing workload also requires computer component usage data.

This paper describes the use of a system monitor[2] to record usage of the different components of a computing system, either individually or in combination. The data obtained are used to create a measured system performance profile to illustrate component usage (e.g., central processing unit (CPU), multiplexor channels, selector channels, I/O devices).[3] After evaluating this gross information, the monitor can be used to further analyze critical performance areas—those portions of the system that the profile indicates have potential for performance improvement. The monitor has been used extensively to measure IBM SYSTEM/360 computer performance.

The monitor

The system monitor can obtain the data necessary for evaluating computing system performance at a user's site. It is connected to the computer by measurement probes, which accept status signals from the computer being monitored, similar to the signals used for system checkout. For example, a measurement probe can monitor a signal that is present only when the computer is in the wait state. Absence of this signal indicates that the CPU is active. Monitoring such signals has no effect on system performance and requires no program modifications.

The unit is equipped with a patchboard to combine system signals where required to obtain additional data. For example, if both I/O and CPU activity are occurring simultaneously, the channel busy and CPU busy signals are ANDed together, and the time that the two conditions prevail is accumulated in one of the sixteen counters, as shown in Figure 1. Each of these counters, which are eleven decimal digits wide, can count events or can record durations of events with a resolution of one microsecond.

Output from the equipment monitor is provided in either of two forms: the contents of any of the sixteen counters are displayed singly by means of a rotary switch; or the contents of all sixteen counters, along with counter identification, are punched on cards.

Monitoring system operation

It is possible to monitor the durations of any or all of the following states of a computing system:

- Executing instructions but not performing I/O operations
- Performing I/O operations but not executing instructions
- Simultaneously executing instructions and performing I/O operations
- Neither executing instructions nor performing I/O operations

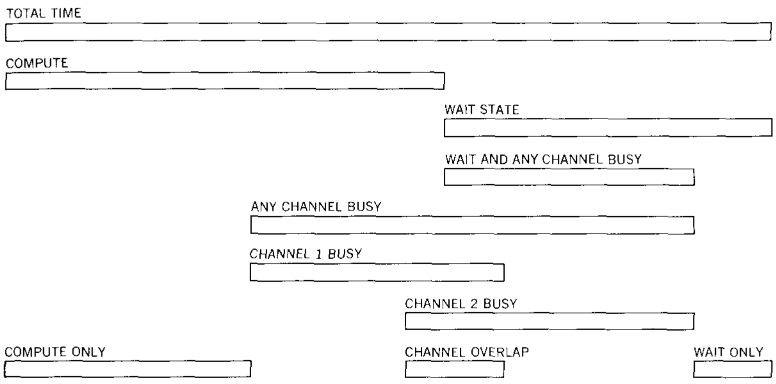
The monitor also provides the following information for the construction of a performance profile:

- Total time the system is active, derived from times the system is neither stopped nor in operator intervention mode.
- Total time the CPU is executing instructions.
- Channel busy time, recorded separately when each selector channel and when the multiplexor channel are being used. (In SYSTEM/360, the multiplexor channel signal may represent several slower units operating simultaneously, since they may share a single channel at the same time.)
- Any channel busy, created by ORing the separate channel-busy signals present.
- CPU wait and any channel busy, created by ANDing the any channel-busy signal with the CPU wait-state signal.
- Channel overlap, obtained by ANDing the channel-busy signals for any combination of the eight selector channels permitted in SYSTEM/360.

From these measurements, other performance data can be calculated:

- Total wait time, obtained by subtracting the total time that the CPU is executing instructions from the total time the system is active.
- System idle time, obtained by subtracting the time that the system is both in a wait state and performing I/O operations from total wait time.
- Overlap of CPU and I/O activity, obtained by subtracting the time that the system is both in a wait state and any channel is busy from the total time that any channel is busy.
- Compute-only time, obtained by subtracting the time that CPU and I/O activities are overlapped from the total time that the CPU is active.

Figure 2 Measured system profile



A system performance profile can be drawn from these measured and derived values, such as is illustrated in Figure 2. Such a profile should indicate any of the more prevalent performance problem conditions.

performance limitations

If there is an excessive amount of compute-only time, the system is said to be compute limited, which could indicate that a faster processor model is required. However, the various parts of computer systems interact to such a degree that any performance change to one part must be considered in relation to the remainder. Thus, if a faster CPU were installed, consideration would also have to be given to the speed of the input/output equipment to avoid changing a basically processor-limited system to a basically input/output-limited system.

If nonoverlapped I/O operations are dominant in the profile, the system is said to be I/O limited and may require additional I/O capability. Again, however, some thought is needed to correctly interpret measured results. Upgrading I/O units may not be the best approach if many input/output operations refer to intermediate storage files. This situation suggests that such files are being used as extensions of main storage. Thus, increasing main storage would appear then to be a sounder approach, since main storage affects both CPU and I/O performance.

When the profile of a system with more than one selector channel shows a wide disparity of work load per channel, channel imbalance is indicated. We have encountered this problem surprisingly often in using the monitors, and it seems to occur irrespective of distribution of I/O units on the channels. Configurations with an even number of I/O units per channel have been observed with as much as a ten-to-one ratio of I/O use between channels one and two because of improper data-set assignments. In some observed samples, this disparity of channel use has been found to be a cause of a portion of the I/O-only time, in that it results in conflicts between I/O functions. In most instances, this problem is readily solved by reassigning the data sets.

The measured system profile offers the user of a computer system the opportunity to adjust its operation based on observed data rather than approximations. In the profile of Figure 2, only gross system measurements are reflected (i.e., total I/O wait time but not by device, compute time but with no indication of the types of instructions being executed). However, the monitor can provide data to perform a more detailed analysis of the use of system components. The profile is a first indication of those areas of the system where effort can best be expended to improve throughput. Further analysis of different sections of the system may be performed based on the information contained in the profile. For example, the monitor can measure the amount of time that the CPU spends executing instructions in one main storage partition of a multiprogramming system, the time that individual I/O units are used in the course of executing a program, or the data flow in number of bytes by device.

detailed analysis

Use of the monitor for such detailed analyses can be demonstrated by considering a multiprogramming system in which partitions or regions of main storage are allocated to the programs competing for the use of system components. In measuring performance of such systems, it is useful to be able to record the amount of processing time for the programs in each of the partitions. To facilitate this measurement in SYSTEM/360 computers, the monitor has a four-by-sixteen decoder.

four-by-sixteen decoder

Main storage partitions are allocated in blocks that correspond to the sixteen storage-protect combinations available in SYSTEM/360. The signals from the four storage-protect bits are made available as inputs to the four-by-sixteen decoder. Whenever the processor is executing instructions, the decoder allocates CPU time to the storage partition that corresponds to the storage-protect key currently in use. The decoder can also be used to measure the time required to execute a smaller segment of a program if the programming system has been set up to allocate a unique protect key to the main storage area containing that segment.

An example of the usefulness of the four-by-sixteen decoder was provided by an analysis of a telecommunications system. In this case, the efficiency of message processing done in a separate partition[4] is compared with that of the remainder of the system. The time that the message-processing program used the CPU was compared to the total time that the CPU was in use. When the results of these measurements were plotted against the number of inquiries processed, it was discovered that the polling rate could be reduced without affecting inquiry response time. In this instance, the monitor was used to achieve a balance between processor use and message rates, the objective being to obtain the lowest polling rate necessary to avoid message delay and keep the processor available for programs in the remaining partitions.

Another method of evaluating performance is to break down the use of the CPU according to the types of instructions executed, providing the data needed to choose a processor model with the optimum performance for that distribution of instructions. This is done by connecting probes to the CPU to determine the operation code. Information that can be determined from the operation code includes:

- Instruction class, such as arithmetic or branch instructions
- Whether operand data is variable length or fixed length
- Whether operand is in binary, floating-point, or decimal format
- Whether operands are in main storage or general registers

With the operation code, the four-by-sixteen decoder can be used together with an AND function to determine the distribution of processor time by operation class. This information is useful in selecting a SYSTEM/360 processor model, since each of the different processor models has a rated internal performance based on an expected distribution of executed instructions by class. The ability to determine actual instruction class distribution can be useful in substantiating pre-installation estimates. Such approximations, if incorrect, can adversely affect performance. An example of the effect of using certain classes of instructions is provided by decimal arithmetic instructions. This class of instructions eliminates the need for converting data to binary form in order to perform calculations. However, frequent use of decimal arithmetic to perform certain calculations may require more time than to perform the same calculations on binary data, notwithstanding the conversion time required.

Analyzing input/output

Overall system performance is affected by the manner in which data are transferred between main storage and external devices. Information about device utilization can assist in achieving maximum efficiency in organizing and storing the data associated with the programs processed at an installation.

By attaching probes to monitor signals from the system input/output control units, the amount of activity (by type) of the various I/O devices can be measured. In this context, type implies such activities as rewinding of tapes, seeking, and actual transferring of data. If the profile indicates a large percentage of I/O and wait-state time, showing a system that is I/O limited, individual device activity is of interest. If this is true, I/O-only time of the separate devices should be examined further to determine the source of this problem.

Through its ability to count records and bytes transferred to and from an I/O unit, the monitor can be helpful in choosing the proper device for data files and programs. This is accomplished by monitoring the control unit for specific activities, such as the numbers of bytes transferred, records transferred, direct access seeks, polls of terminals, etc. Once these data are obtained, they are analyzed in the light of input/output device characteristics to fit the data to the characteristics of the I/O equipment.

An example of the use of I/O device utilization data was a study of the distribution of accesses to the different modules of a four-module, direct-access storage system. The monitor was connected to the system to obtain data on the total number of seeks to each module, the total number of cylinders traversed during seeks, and the duration of seeks. The data files and programs stored on these modules were: modules 0 and 1, work area and reader, printer, and punch queues; module 2, catalog and application library; and module 3, system library. The data obtained are shown in Table 1.

Table 1 Input/output device utilization

Module	*Cylinders traversed per seek*	*Seek time (msec)*
0	20	39
1	17	48
2	100	72
3	21	37

Module two, in this study, considerably exceeded the other modules in cylinders traversed, which resulted in a higher average access time. The cause proved to be a conflict between a catalog and an application library. The catalog is an index used to translate program references to data or to other programs from a symbolic name to a physical I/O device address.[5] Since each reference to the application library first required a reference to the catalog, a conflict existed on that module. Based on this data and a knowledge of the programming system, three alternative solutions to improve average access time were considered:

- Reposition the catalog on the module to reside on the middle cylinders, surrounded by the application library, or on the cylinders adjacent to those application programs most frequently used.
- Place the catalog on a different module.
- Place the catalog on a drum storage unit.

The catalog was subsequently moved to another available module (there are eight on the device), which brought the average access time more into accord with that of the other modules.

Unit record equipment

Unit record equipment, such as printers, punches, card readers, and the console typewriter, can be monitored to measure their usage. The IBM 2821 control unit contains signal points that, when monitored, provide:

- The time any device attached to the control unit is busy
- The number of lines printed
- The number of pages printed
- The number of cards read
- The number of cards punched

Each of the busy and wait-state lines of the unit record devices may be wired into an AND function to monitor the amount of I/O-only delay contributed by each unit. The instances where this has been done with the console typewriter have proved fruitful, in that there has generally been a sharp reduction in the length and number of console messages. Quantitative data proved that frequent or long console messages have a bearing on processor wait-state time.

Summary comment

The items measured here are but a few of the possible uses of the system monitor. To date, concentration has been on items of interest to the user of the data processing system. Even here, the measurements taken have been those that concern overall system performance. Once these adjustments are made, attention will probably be turned to more detailed measurements, such as an interruption-handling routine, channel contention, a critical subroutine, and other small but important aspects of the system. Also, the monitor may be used in the future to acquire data to assist the designers of new equipment, an area where monitors have proved useful in the past.

This paper does not discuss the relationship of computer system monitors to system simulators and analytic models. These tools require computation time to simulate or analyze a system. If the equipment monitor is used first to obtain overall information about the system, the model can be confined to more detailed portions of the system to reduce the overall time required to analyze performance. The monitor can also be useful to verify assumptions made in developing input to the simulator.

Although this paper covers the use of the monitor only on SYSTEM/360, the unit can be, and has been, attached to other systems, such as the IBM 1130 and the IBM 1800 systems, to perform similar studies.

ACKNOWLEDGMENTS

The system monitor documented here represents a design that has evolved from many of the ideas contained in earlier monitors. The author would therefore like to acknowledge the contributions of those individuals who thus indirectly contributed to this unit. Contributions to the engineering design of the present unit were made by J. D. McNeill, N. K. Powers, T. A. Papastasiou, and R. Harris. In addition, the author acknowledges the contribution of R. R. Stuart for the direct-access device work.

CITED REFERENCES AND FOOTNOTES

1. G. A. Blaaw and F. P. Brooks, Jr., "The structure of SYSTEM/360, Part I, Outline of the logical structure," *IBM Systems Journal* **3**, Nos. 2 and 3, 119–135 (1964).
2. The system monitor, a laboratory test device, is used to evaluate system design and to test programs.
3. The concept of a computer monitoring device evolved in the IBM Product Development Laboratory as a means of obtaining data on systems and program utilization. In 1961, an early version, the Machine Usage Recorder,

was built. This unit was used in IBM Data Centers to help customers evaluate their 7090 programs. A solid-state version, the 7090/7094 Portable Monitor (Channel Analyzer) followed in 1962 and was used extensively at customer sites, also to evaluate programming and configuration efficiency. The monitor described here is a third-generation device, which is an outgrowth of these earlier efforts.

4. The teleprocessing system studied was similar to those analyzed by Carol Hauth, "Turnaround time for messages of differing priorities," *IBM Systems Journal* **7**, No. 2, 103–122 (1968).
5. W. A. Clark, "The functional structure of OS/360, Part III, Data management," *IBM Systems Journal* **5**, No. 1, 30–51 (1966).