Adapted from a speech given before the AICPA Fifth National Conference of Computer Users, held in Chicago in May. The Conference itself will be covered in the September-October issue—

SYSTEMATIC TECHNIQUES FOR COMPUTER EVALUATION AND SELECTION

By John R. Hillegass
Computer Conversions, Inc.

As you all know, there is a very wide variety of computer equipment and supporting software on the market today, with dozens of suppliers contending for your hardware and software dollars. You've probably heard more than one computer user say that it's all pretty much alike, and that there are few, if any, significant differences in the capabilities and features of the equipment and software available from the various manufacturers. On the basis of our own experience in equipment and software evaluation, we can say with great conviction that this simply isn't true—and chances are that the user who thinks it is will be spending a lot more money than he should to get the computing power he needs—as much as thousands of dollars more each month.

Among the available computers in any given class, there are very significant differences in their performance per dollar and their overall suitability for specific applications. Therefore, the use of systematic, objective procedures for computer evaluation and selection can save you and your clients a great deal of time and money. Moreover, it can guard against the serious disruptions that are occurring in all too many firms these days as a result of the installation of an inadequate computer.

Nothing would be more pleasing than to be able to tell you all about a simple, foolproof technique that would guarantee selection of the most suitable computer system for your needs. Unfortunately, no such technique is available today, and none is expected within the foreseeable future. The development of such a technique simply has not been possible—despite the great demand for it—because of the many nonhardware factors that have an important effect upon overall computer performance and economy but are extremely difficult to pin down in any quantitative manner, factors such as compatibility with your present equipment, expandability to handle new applications, ease of programing, quantity and quality of the manufacturer's support, availability and quality of maintenance service, back-up considerations in the event of equipment failure, and many more.

There are, however, a number of techniques available today that can aid very significantly in determining the most suitable equipment and software for your particular needs. This article will describe briefly some of these techniques and tell you about the advantages and drawbacks you can expect from each of them. It might be worthwhile to keep in mind that tht ideal computer evaluation technique would be easy to apply, inexpensive, comprehensive (in that no significant factors are overlooked), and totally valid (in that it always leads to the correct conclusion as to the most suitable hardware and software). Now let's see how close each of the available techniques comes to satisfying these criteria.

Standard benchmark problems, as featured in the widely used Auerbach EDP reference services, are designed to be representative of typical computer workloads in both business and scientific applications. The problems include file updating, sorting, matrix inversion, and polynomial evaluation.

Instruction mixes

First, there are the *instruction mixes.* To compare central processor speeds, several weighted mixes of instruction execution times have been developed.[1,2] Probably the most popular one is the Gibson Mix, originally developed by the Air Force. Each of these mixes is simply a weighted average of the execution times for a number of the most commonly used instructions. A weighting factor is assigned to each instruction in accordance with somebody's opinion of that instruction's frequency of occurrence in programs of a certain general type.

Instruction mix times are easy to calculate and compare, but they can only measure central processor speeds. Furthermore, these methods ignore the facts that the frequencies of different types of instructions vary widely in programs for different applications and that a single specialized instruction in one computer may be able to perform functions that require several of the basic instructions in other computers.

For the scientific computer user who applies them with due caution, instruction mixes can provide useful approximations of raw computing power. But for the business computer user who needs balanced computing and input/output power, they are likely to be misleading and virtually useless.

Second, there is the so-called *kernel* approach. A kernel is a simple problem, presumably representative of typical business or scientific computer applications, that is coded and timed for each of the computers under consideration.[1,2]

Kernels permit each computer's internal instruction repertoire to be used to best advantage, but, like the instruction mixes, they generally ignore input/output considerations and software performance factors. Moreover, it is usually difficult or impossible to relate the times for an assortment of kernels to a given user's real data processing applications.

Third, there are *standard benchmark problems,* as featured in the widely used Auerbach EDP reference services.[3] These standard problems are designed to be representative of typical computer workloads in both business and scientific applications. The problems include file updating, sorting, matrix inversion, and polynomial evaluation.

To help ensure objective comparisons, the standard problems are rigidly specified in terms of input data, computations, and results to be produced. On the other hand, factors such as master file arrangement and detailed coding methods are left flexible to permit optimum use of the distinctive capabilities of each computer. Finally, to assure realistic comparisons between competitive systems, the equipment configurations as well as the problems are standardized. Each computer's performance is measured in a number of different standardized configurations.

The execution time for each standard problem on each standard configuration is determined by calculating all input/output times and central processor times and then combining them with due regard for the system's capabilities for simultaneous operations. The results are presented in the form of graphs that show each computer system's performance over a wide range of problem parameters and equipment configurations.

These standard benchmark problems can give you a good idea of the overall performance characteristics of competitive computers on applications similar to your own. It is not always easy, however, to relate the standard problems and standardized equipment configurations to your own particular problems and equipment needs. Moreover, the important effects of software performance and of advanced operating techniques such as multiprograming, time sharing, and data communications are virtually ignored.

Fourth, there is an even more widely used type of benchmark problem which may be called *"live" benchmarks.* These are problems, designed to be as representative as possible of a specific user's workload, which are actually programed, compiled, executed, and timed for each of the computer systems under consideration.[4] Live benchmarks provide an excellent opportunity to observe each computer system's overall performance, including the effects of input/output simultaneity and software inefficiencies. They can also tell you a great deal about the ease or difficulty of programing and operating each system.

Drawbacks

There are three main drawbacks associated with the use of live benchmarks. First, they tend to be comparatively time-consuming and expensive to prepare. Second, it may be impossible to conduct live benchmark tests when the interest in a new computer system is highest—immediately after its introduction. Third, and most important, it is usually very difficult to accurately estimate a computer's overall performance on a user's entire workload on the basis of its performance on a few simple benchmark problems—particularly where the user has a wide range of applications or where some of these applications involve data communications or multiprograming.

JOHN R. HILLEGASS is president and chief executive officer of Computer Conversions, Inc., of Jenkintown, Pa. Previously he was manager of the computer technology unit of the Auerbach Corp., where he was the principal developer of the Auerbach Computer Notebook for Accountants. Mr. Hillegass received his BS in chemical engineering from Lehigh University and holds a DPMA certificate in data processing. He is a member of Phi Beta Kappa and the Association for Computing Machinery. Over a dozen of his articles on data processing equipment and techniques have been published.

The fifth evaluation technique is *computer simulation.* This involves the use of a computerized model to determine the run times for predefined program runs and equipment configurations.[5] The two best known examples are Comress' SCERT (Systems and Computers Evaluation and Review Technique) and a newer technique called CASE (Computer Aided System Evaluation).

Each of these techniques consists of a complex computer program and a library of hardware and software factors describing the key characteristics of most of the commercially available computer systems. Given a series of program run specifications and a series of equipment configuration definitions, the program determines the estimated execution time for each run on each configuration. The detailed reports produced by the program also specify other useful information such as estimated memory requirements, programing time requirements, environmental requirements, and equipment costs.

These simulation techniques are probably the most elaborate ones yet developed to aid in computer selection. The estimates they produce appear to be generally valid for straightforward, batch-type applications but distinctly less reliable for more sophisticated operational modes such as those involving multiprograming or data communications. Preparation of the

It is not always easy to relate the standard problems and standard equipment configurations to your own particular problems and equipment needs. Moreover, the important effects of software performance and of advanced operating techniques are virtually ignored.

detailed input specifications is time-consuming, and the techniques are by no means inexpensive to use. Moreover, no simulation technique can produce valid results when supplied with invalid data, and some of the required data regarding software efficiencies and programing time requirements are difficult, if not impossible, to obtain.

All of these computerized run-timing techniques are essentially limited to the determination of estimated running times for user-defined runs on user-specified equipment configurations. Therefore, they cannot relieve you of the basic problem of designing an effective system and finding the best complement of equipment to implement it; they can only make it a lot easier for you to investigate a large number of alternative cases.

The sixth type of evaluation technique can be referred to as the *weighted factor methods*.[6,7] A number of techniques have been developed to systematize the process of computer selection by assigning weighted point scores to each significant factor within the general categories of equipment characteristics, programing, software support, pricing, etc. An appropriately weighted score is assigned to each factor for each competing computer system. Then the scores are added up, and the system with the highest point score ostensibly is the best choice.

Weaknesses of technique

The fatal flaw here is that the factors to be considered and the weights to be assigned to them are, by necessity, chosen arbitrarily. No objective guidelines exist for matching them to a particular user's needs. It is unlikely that any two analysts, given the job of independently establishing appropriate factors and weights to select the best computer for a particular installation, would arrive at similar conclusions. The weights can easily be juggled to lead to virtually any desired result. Furthermore, there is a real danger that by the time the analyst has performed all the necessary weighting and scoring calculations, he may tend to lose sight of their shaky foundations and attach undue significance to the results, which are subject to error.

The published articles describing the various weighted factor techniques can provide convenient checklists of factors that should be considered in computer evaluation studies. But in terms of their use as objective selection techniques they should be viewed with deep suspicion.

The seventh evaluation method, the *cost/value technique*, is a variation of the weighted factor methods.[8] It represents a significant improvement over them in that it strives to establish meaningful relationships between the items of value to the user and their costs. Proposals from the manufacturers are ranked by a scheme called cost/value accounting. This involves taking the total cost of a proposed system and deducting the estimated values of all the desirable extra features included in that proposal. The difference is then considered to represent the cost of satisfying the mandatory requirements set forth in the user's request for proposals. The system with the lowest cost for satisfying the user's mandatory requirements is then judged the best choice, because the values of the desirable extra features offered have already been taken into account.

The cost/value technique is quite sophisticated and comprehensive. But, like the weighted factors methods, it forces the user to assign quantitative values to many factors for which no objective guidelines exist. Consequently, there is a very real possibility that the results will be biased and misleading.

You can see that each of these seven evaluation techniques has significant advantages and drawbacks associated with it, and each falls far short of being ideal. Some of these techniques, such as instruction mixes and kernels, deal strictly with hardware performance. Others, such as live benchmarks and simulation, introduce the important element of software performance as well. Still others, notably the weighted factors and cost/value techniques, attempt to give proper consideration not only to hardware and software but also to the many other factors which are important in computer selection, such as reliability, compatibility, expandability, manufacturers' support, and contract terms.

Since all of the evaluation techniques discussed here obviously have significant disadvantages associated with them, the picture may seem rather bleak. Fortunately, it isn't nearly as bad as it looks. It *is* possible to make objective computer selections with a high degree of confidence that the equipment and software selected will truly be the most suitable and economical choice. What is needed is a combination of one or more of the formal evaluation techniques just described with a systematic overall selection procedure and a good deal of old-fashioned common sense.

[1] Arbuckle, R. A., "Computer Analysis and Thruput Evaluation," *Computers and Automation*, January, 1966, pp. 12-15.

[2] Solomon, M. B., Jr., "Economies of Scale and the IBM System/360," *Communications of the ACM*, June, 1966, pp. 435-440.

[3] Hillegass, J. R., "Standardized Benchmark Problems Measure Computer Performance," *Computers and Automation*, January, 1966, pp. 16-19.

[4] Joslin, E. O., "Application Benchmarks: The Key to Meaningful Computer Evaluation," *Proceedings of the ACM 20th National Conference*, 1965, pp. 27-37.

[5] Canning, R. G., "Data Processing Planning Via Simulation," *EDP Analyzer*, April, 1968, pp. 1-13.

[6] Bromley, A. C., "Choosing a Set of Computers," *Datamation*, August, 1965, pp. 37-40.

[7] Schwartz, E. S., "Computer Evaluation and Selection," *Journal of Data Management*, June, 1968, pp. 58-62.

[8] Joslin, E. O., *Computer Selection*, Addison-Wesley, Reading, Mass., 1968, pp. 18-45.