CHAPTER 5

THE ART OF WORKLOAD SELECTION

"Would you tell me, please, which way I ought to go from here?"
"That depends a good deal on where you want to get to," said the Cat.
"I don't much care where—" said Alice.
"Then it doesn't matter which way you go," said the Cat.
—Lewis Carroll, *Alice's Adventures in Wonderland*

The workload is the most crucial part of any performance evaluation project. It is possible to reach misleading conclusions if the workload is not properly selected. When the conclusions of a study are found to be unacceptable, the inappropriateness of the workload is commonly used as a criticism of the study. Like other aspects of performance evaluation, proper selection of workloads requires many considerations and judgments by the analyst, which is a part of the art of performance evaluation that comes with experience. In this chapter, a number of considerations are described that will help you make the right selection and justify your choice.

The four major considerations in selecting the workload are the services exercised by the workload, the level of detail, representativeness, and timeliness. These are discussed in the next four sections. Other minor considerations, such as loading level, impact of other components, and repeatability, have been grouped together in Section 5.5.

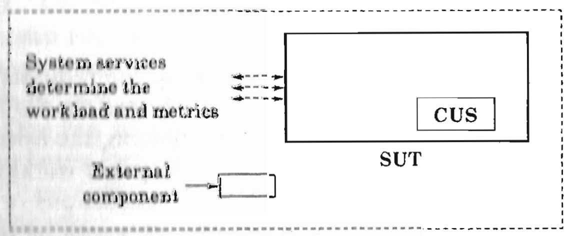
5.1 SERVICES EXERCISED

The best way to start the workload selection is to view the system as a service

and making a list of services is one of the first steps in a systematic performance evaluation study. Often the term **System Under Test (SUT)** is used to denote the complete set of components that are being purchased or being designed by the organization. Sometimes there is one specific component in the SUT whose alternatives are being considered. This component is called **Component Under Study (CUS)** as shown in Figure 5.1. For example, a Central Processing Unit (CPU) design team might want to understand the impact of different Arithmetic-Logic Unit (ALU) organizations. In this case, the CPU is the SUT and the ALU is the CUS. Similarly, a bank purchasing a transaction processing system may want to compare different disk devices. In this case, the transaction processing system is the SUT and the disk devices are the CUS. Clearly, identifying the SUT and CUS is important since the workload as well as the performance metrics are determined primarily by the SUT. Confusing CUS with SUT and vice versa is a common mistake that leads to misleading results. In the remainder of this chapter, the word *system* will be used to mean SUT and the word *component* will be used to mean CUS.

The metrics chosen should reflect the performance of services provided at the system level and not at the component level. For example, the MIPS is a justifiable metric for comparing two CPUs, but it is not appropriate for comparing two timesharing systems. The CPU is only one component of the timesharing system. A timesharing system may provide services such as transaction processing, in which case the performance would be measured by transactions (as opposed to instructions) per second.

The basis for workload selection is also the system and not the component. For example, the services provided by the CPUs are the so-called instructions, and the CPU designers may want to use instruction frequency as a possible representation of workload. The services provided by the turn-key banking systems are generally called "transactions," and so the bank may use the transaction frequencies as the workload. Notice that using instruction frequencies to specify the workload of a banking system is not appropriate, since the performance of the banking system depends on several components in addition to that of the CPU. Similarly, using transactions to compare two CPUs may or may not be appropriate, since the performance may be affected by other components such as I/O devices. However, if a manufacturer offers two banking systems that are identical except for the CPUs, the two systems can be



compared using transaction frequencies as the workload. This latter study may sometimes be inaccurately referred to as the comparison of the two CPUs.

If the system provides multiple services, the workload should exercise as complete a set of services as possible. Thus, it is inappropriate to measure the performance of an ALU that provides both floating and integer arithmetic by only using an integer workload. In this case, the workload does not exercise all the key services provided by the system.

In considering the services exercised, also take into account the purpose of the study. A workload may exercise the most efficient features of the system or the least efficient. For example, a graphics editor may not be an efficient text editor. Thus, a text-editing workload used to compare two editors may bring out the worst in the graphics editor and vice versa. A text-editing workload is not appropriate if the purpose of the study is to find an editor to be used primarily for graphics work.

To summarize the discussion so far, the requests at the service-interface level of the SUT should be used to specify or measure the workload, and one should carefully distinguish between the SUT and CUS since it is easy to confuse one for the other.

The types of workloads discussed in Chapter 4 can be shown to be an application of the principle outlined here. Figure 5.2 shows a hierarchical view of a timesharing system. A user typically types in a high-level request, for example, to make a withdrawal from a bank account. The application software may translate the transaction into a number of requests for the operating system. These requests in turn make a number of requests to be executed by various hardware components of the system, which may include a number of specialized processors, general purpose processors, I/O devices, and network links. The CPU requests may be translated into a number of instructions, each of which may make one or more requests to the ALU. Thus, as shown in Figure 5.2, there is a hierarchy of interfaces at which the requests are serviced. A single request at a higher level may result in one or more requests at the lower level. As shown in the figure, the interface levels are:

1. Arithmetic-logic unit
2. Central processing unit
3. Operating system
4. Applications

The workload could be described by summarizing the requests at any one of these interface levels, depending upon what constitutes the SUT. If two ALUs are being compared, that is, the ALUs are the systems, the arithmetic instructions constitute the services or requests. The appropriate workload in this case is to specify the frequency of various arithmetic instructions or to specify the most frequent arithmetic instruction, which may very well be the addition instruction.

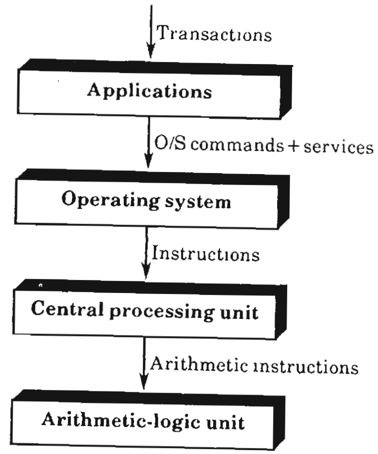


FIGURE 5.2 Hierarchical view of timesharing system.

If two CPUs are being compared, the instruction set of the processors constitutes the service provided. One possibility in this case is to use the instruction mix. However, if the performance of one instruction depends upon that of other neighboring instructions, the workload should be specified in terms of a set of instructions that are commonly used together. The popular kernels discussed in Section 4.6 are examples of such sets.

If two systems are being compared at the operating system level, the services provided by the operating systems, including the operating system commands and system services, should form the basis of the workload. In this case, synthetic programs discussed in Section 4.4 can be used.

If two turn-key transaction processing systems are being compared, the application interface is the SUT level and the requests at this level, namely, the transactions, would form the basis of workload. The workload could be described by specifying the frequency of various types of transactions or the most frequent transaction. The debit-credit workload described in Section 4.6.7 is an example of the workload to be used for such a study.

It should be obvious that the idea of a service interface level can be applied to other types of computer systems. The following example shows its application to computer networks.

Example 5.1 Consider the problem of selecting or designing the workload to compare two networks. In the literature, the term *network* is used loosely to mean *anything* from a cable between two computers to networking applications, such as mail, that allow communications between several computers. One way to represent the hierarchy of levels in this case is to use the seven layers identified by the ISO/OSI reference model as shown in Figure 5.3. These layers and the corresponding workloads are given next. Readers not familiar with networking terminology should see a networking textbook such as Tanenbaum (1988).

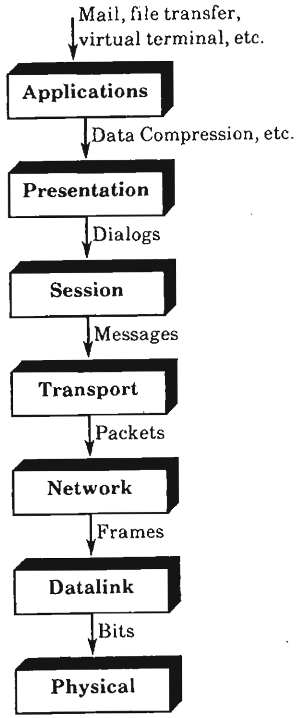


FIGURE 5.3 Workloads corresponding to the seven layers of the ISO/OSI reference model for computer networks.

1. *Physical Layer*: This layer deals with the transmission of individual bits over the physical medium, which may be a twisted-pair wire, a cable, or a fiber-optic link. The key service provided by this layer is the transmission of a bit. In some networks, the physical layer deals with a group of bits called symbols. Thus, the workload to compare two links should match the frequency of various symbols or bit patterns observed on real networks.
2. *Datalink Layer*: This layer deals with the transmission of frames over a single link. The workload to compare two datalink protocols would require specifying the characteristics of frames transmitted, such as their lengths and their arrival rates. On links that connect multiple stations, for example, on local-area networks, one would also want to specify the source-destination matrix.
3. *Network Layer*: This layer routes packets from a given source node to a given destination over multiple links. The workload in this case would require specifying the source-destination matrix, the distance between source and destination, and the characteristics of packets transmitted.
4. *Transport Layer*: This layer deals with the end-to-end aspects of com-

provided include segmentation and reassembly of large messages. The workload would consist of specifying the frequency, sizes, and other characteristics of various messages.

5. *Session Layer*: The complete dialog between the user processes on the two end systems is called a session. The session layer provides services related to the initiation, maintenance, and disconnection of the sessions. The workload for the session layer would thus include the frequency and duration of various types of sessions.
6. *Presentation Layer*: This layer provides data compression and security. The workload for this layer would consist of specifying the frequency of various types of security and data compression requests.
7. *Application Layer*: This layer consists of user applications such as mail, file transfer, and virtual terminal. The workload at this layer would consist of specifying the frequency of various types of applications and their associated characteristics.

Thus, the choice of workload depends upon the layer at which the networks are being compared. While a typical mail session may be the right workload to compare two mail systems, it may not be appropriate to compare two fiber-optic links. □

The following example illustrates the application of the services concepts to the analysis of an off-line backup storage system using magnetic tape drives.

Example 5.2 A magnetic tape backup system consists of several tape data systems, each containing several tape drives. The drives have separate read and write subsystems. Each subsystem makes use of magnetic heads. Thus, starting from a high level and moving down to lower levels, the services, factors, metrics, and workloads are as follows:

1. *Backup System*:
   (a) Services: Backup files, backup changed files, restore files, list backed-up files.
   (b) Factors: File system size, batch or background process, incremental or full backups.
   (c) Metrics: Backup time, restore time.
   (d) Workload: A computer system with files to be backed up. Vary frequency of backups.
2. *Tape Data System*:
   (a) Services: Read/write to the tape, read tape label, autoload tapes.
   (b) Factors: Type of tape drive.
   (c) Metrics: Speed, reliability, time between failures.
   (d) Workload: A synthetic program generating representative tape I/O requests.

3. *Tape Drives*:
   (a) Services: Read record, write record, rewind, find record, move to end of tape, move to beginning of tape.
   (b) Factors: Cartridge or reel tapes, drive size.
   (c) Metrics: Time for each type of service, for example, time to read record and to write record, speed (requests per unit time), noise, power dissipation.
   (d) Workload: A synthetic program exerciser generating various types of requests in a representative manner.
4. *Read/Write Subsystem*:
   (a) Services: Read data, write data (as digital signals).
   (b) Factors: Data-encoding technique, implementation technology (CMOS, TTL, and so forth).
   (c) Metrics: Coding density, I/O bandwidth (bits per second).
   (d) Workload: Read/write data streams with varying patterns of bits.
5. *Read/Write Heads*:
   (a) Services: Read signal, write signal (electrical signals).
   (b) Factors: Composition, interhead spacing, gap sizing, number of heads in parallel.
   (c) Metrics: Magnetic field strength, hysteresis.
   (d) Workload: Read/write currents of various amplitudes, tapes moving at various speeds. □

5.2 LEVEL OF DETAIL

After the service interface of the SUT has been identified and a list of services has been made, the next step in the workload selection is to choose the level of detail in recording (and thus reproducing) the requests for these services. A workload description may be as long as a time-stamped record of all requests or it can be as short as the single most commonly used request. A list of possibilities is as follows:

1. Most frequent request
2. Frequency of request types
3. Time-stamped sequence of requests
4. Average resource demand
5. Distribution of resource demands

The least detailed alternative is to select the most frequently requested service and use it as the workload. While this may not provide enough information about the system, this is commonly used as the initial workload.

processors, and the debit-credit benchmark to compare transaction processing systems are examples of this approach. It is particularly valid if one type of service is requested much more often than others or is a major consumer of resources in the system.

The second alternative is to list various services, their characteristics, and frequency. The instruction mixes are examples of this approach. If the performance of a service depends upon the context, that is, on the services required in the past, the set of services that are expected to be context free are more appropriate than the individual services. History-sensitive mechanisms, such as caching in computer systems, often make this grouping necessary.

The third alternative is to get a time-stamped record (called a **trace**) of requests on a real system and use it as a workload. The problem with this alternative is that it may be too detailed. It is certainly inconvenient for analytical modeling. Also, for simulation it may require exact reproduction of component behavior to maintain the timing relationships. The advantages and disadvantages of trace-driven simulations are further discussed in Section 24.5.

In analytical modeling, the resource demands placed by the requests, rather than the requests themselves, are used as the workload. For example, the statement that each user has an average CPU time of 50 milliseconds and makes 20 I/O's per request may be a sufficient workload description for analytical modeling. In case of multiple services, several similar services can be grouped into a class, and each class may be characterized by its average resource demands.

The average demand may not be sufficient in some cases, and it may be necessary to specify the complete probability distribution for resource demands. This is particularly the case if there is a large variance in resource demands or if the distribution is different than that used in the model. Particularly, in simulations, it is easy to use different distributions. The analytical models are generally restricted to a given distribution.

The workload descriptions used for analytical and simulation modeling are also referred to as **nonexecutable** workloads since they are not in a form suitable for execution on a real system. On the other hand, a trace of user commands that can be executed directly on a system would be called an **executable** workload.

5.3 REPRESENTATIVENESS

A test workload should be representative of the real application. One definition of representativeness is that the test workload and the real application match in the following three respects:

1. *Arrival Rate*: The arrival rate of requests should be the same or proportional to that of the application.
2. *Resource Demands*: The total demands on each of the key resources should be the same or proportional to that of the application.

3. *Resource Usage Profile*: Resource usage profile relates to the sequence and the amounts in which different resources are used in an application. In a multiprogramming environment, it is important that the test workloads have a resource usage profile similar to that of the applications. Otherwise, it is possible that total resource demands of individual workloads may be representative of their respective applications, but when several workloads are run together, they may not produce results representative of combined applications.

5.4 TIMELINESS

Workloads should follow the changes in usage pattern in a timely fashion. User behavior has changed considerably over the years. For example, the trend has been to move away from machine languages toward higher level languages and from batch toward timesharing. Users change their usage pattern depending upon the services provided by the new systems. Thus, workloads become obsolete as soon as they become well understood. Nonetheless, these obsolete workloads keep getting used. The debit-credit workload used in transaction processing is an example of an outdated, but popular, workload. It is important that the workload represent the latest usage pattern.

Timeliness is a difficult goal to achieve. The main problem is that real user behavior is a moving and fuzzy target. Any observed user behavior applies to a specific environment for a specific system at a specific time. Users change their behavior with time, and any change in system performance prompts users to change their usage behavior. Users tend to change their demands to optimize the response for a given system, focusing on those features that the system can perform efficiently. For example, the processors that provide fast multiplication have a much higher frequency of multiplication instructions than the processors with slow multiplication instructions. Even on the same system, the usage pattern changes with time. Initially, computers were primarily used for scientific applications that required fast arithmetic operations. Today, some may argue that applications such as databases are I/O intensive, so the speed of I/O is more important than the speed of arithmetic operations.

This interdependence of system design and workload provides opportunity for debate at every performance evaluation presentation. This is particularly true for the evaluation of systems under design. A workload based on an existing system, even if modified for the new system features, cannot be guaranteed to be an accurate representation of future workloads. Furthermore, designs that are optimized on the basis of one or more workloads cannot be guaranteed to operate efficiently in other environments. It is therefore very important that the user's behavior be monitored on an ongoing basis and that

5.5 OTHER CONSIDERATIONS IN WORKLOAD SELECTION

There are a few more issues, in addition to the services exercised and the level of detail, to consider in workload selection. These issues—loading level, impact of external components, and repeatability—are explained next.

1. *Loading Level*: A workload may exercise a system to its full capacity (best case), beyond its capacity (worst case), or at the load level observed in real workload (typical case). For procurement purposes, a workload measured in a similar existing environment may be good enough. However, for computer system design, you may have to identify all the environments where the system might be used and study performance under best, worst, and typical workloads. The correct choice of the loading level varies from case to case. For example, to measure the effectiveness of a congestion control scheme, the network should be exercised beyond its capacity, while the packet retransmission schemes should be tested for normal as well as heavy load, since the retransmissions may be required under both circumstances.
2. *Impact of External Components*: Sometimes components outside the system may have a significant impact on the system's performance. For example, the completion time of a synthetic program may depend heavily on I/O device performance. This is not a good workload to compare two processors. The speed difference between the processors may be masked completely by the I/O device. Ignoring this issue often leads to the conclusion that all alternatives in the system give equally good performance. The reason may be that the workload does not exercise the system or that there is a component external to the system that is the bottleneck. Therefore, either the workload should be modified to put less load on this other component, the system configuration should be modified to make this external component less significant, or the results should be analyzed to predict the performance of the system in the absence of the external bottleneck.
3. *Repeatability*: A workload should be such that the results can be easily reproduced without too much variance. Workloads that make a highly random demand on resources are less desirable since one would need to make many more runs to get a meaningful estimate of performance.

EXERCISES

5.1 Decide the metric and workload you would choose to compare the following:

a. Two systems for very different applications: IBM PC versus Macintosh