Acta Informatica 7, 167—182 (1976)

Fundamental Operational Laws of Computer System Performance*

J. P. Buzen

Received June 4, 1976

Summary. A number of laws are derived which establish relationships between throughput, response time, device utilization, space-time products and various other factors related to computer system performance. These laws are obtained through the operational method of computer system analysis. The operational method, which is formally introduced in this paper, differs significantly from the conventional stochastic modeling approach and is based on a set of concepts that correspond naturally and directly to observed properties of real computer systems. The operational laws presented in this paper apply with complete precision to all collections of observational data, and they are similar to fundamental laws found in other areas of engineering and applied science

1. Introduction

Most analyses of computer system performance rely on either benchmarks or probabilistic models. Benchmarks, which may consist of real programs, synthetic programs or trace driven simulations, are most useful when it is necessary to determine system behavior under a precisely specified workload. However, benchmark results can be surprisingly sensitive to the nature of the workload that the system is assumed to be processing, and slight changes in the workload definition may sometimes lead to significantly different conclusions.

Table 1 illustrates the crux of the problem through a highly simplified example. Suppose that an analyst is comparing "round robin" (RR) and "first come first served" (FCFS) scheduling algorithms for a central processor. Assume that the workload consists of three jobs: Job A, with a duration of 7 s; Job B, with a duration of 1 s, and Job C, with a duration of 3 s. The first line of Table 1 gives the average response time for the three jobs in the case where the order of arrival is "ABC" with all jobs arriving at approximately the same time. Note that the average response for FCFS is 18% higher than the average response time for RR with a quantum of two seconds (see Appendix A for details). Thus the benchmark results in line one indicate a definite preference for RR.

In the second line of Table 1, everything is the same except that the order of arrival is reversed. In this case, average response time for RR is 11% higher than average response time for FCFS. The second set of benchmark results thus

* A preliminary version of this paper was published in the Proceedings of the IFIP/ACM SIGMETRICS International Symposium on Computer Performance Modeling, Measurement and Evaluation, Cambridge (Mass.) March 1976, p. 200–210

Table 1

Workload			Average response time	
first	second	third	FCFS	RR ($Q=2$)
A	B	C	$8\frac{2}{3}$	$7\frac{1}{3}$
C	B	A	6	$6\frac{2}{3}$
B	A	C	$6\frac{2}{3}$	$6\frac{2}{3}$

indicates a preference for FCFS, even though this benchmark contains the same set of jobs as the first. As a point of interest, the third line of Table 1 presents yet another arrival sequence in which the two scheduling algorithms produce exactly the same average response time.

Although the example in Table 1 is highly simplified, the dangers which it illustrates are very real. The reason for the discrepancies in Table 1, and for similar discrepancies in other cases, is that benchmark evaluations require specification of the system workload in complete detail: as a result, the analyst is often compelled to make subtle but critical decisions in areas where his knowledge is imprecise. This results in confusing situation where seemingly equivalent benchmark studies lead to different final conclusions.

2. Probablistic Models

Probabilistic models enable the analyst to deal directly with situations where only partial knowledge of the workload exists. Suppose, in the case of Table 1, that the workload is known to consist of Jobs A, B, and C, but the order in which the jobs will arrive is uncertain. If the uncertainty concerning the order of arrival can be represented with a probabilistic model, the analyst may be able to formulate a solution in terms of random variables. For example, suppose that it is reasonable to assume that the sequence "ABC" will occur 60% of the time, the sequence "CBA" will occur 10% of the time, and the sequence "BAC" will occur 30% of the time. In this case, the arrival sequence can be regarded as a random variable, and the expected response times for FCFS and RR can be computed as follows:

$$\begin{aligned}&\text{Expected FCFS response time}\\&=0.60\times 8\tfrac{2}{3}+0.10\times 6+0.30\times 6\tfrac{2}{3}\\&=7.80.\\&\text{Expected RR response time}\\&=0.60\times 7\tfrac{1}{3}+0.10\times 6\tfrac{2}{3}+0.30\times 6\tfrac{2}{3}\\&=7.07.\end{aligned}$$

Thus, the analyst can compute that—"on the average"—FCFS response time will be 10% higher than RR response time. The analyst is not troubled by the fact that individual benchmark tests fail to confirm this prediction; indeed, the analyst fully expects to find RR response higher than FCFS response time for 10% of the arrival sequences.

The potential discrepancies between predicted and observed average response times discussed in the preceding paragraph were magnified by the implicit assumption that the benchmark consisted of only a single three-job arrival sequence. In practice one would normally include a large number of three job sequences in a single benchmark. In such cases, results from probability theory (e.g., the Law of Large Numbers, the Ergodic Theorem, and the theory of confidence intervals) imply that predicted and observed average values will be reasonably close in almost all cases. This is the major reason for constructing simulation programs that run for long periods of time and constructing benchmarks that constitute adequately large samples from the set of anticipated workloads.

3. Operational Objectives

The assertion that a particular theoretical result can be validated in practice at some desired confidence level—if the experiment is run "long enough"—may be satisfactory for resolving certain performance evaluation issues, but such assertions are clearly insufficient to meet all the needs of empirically oriented computer performance analysts. Such individuals usually deal with sets of data that have been collected by direct measurement of actual systems during finite intervals of time. These analysts are basically interested in:

A) Precise mathematical expressions which relate existing measurement data to other quantities that were not measured but which could, in principle, be empirically determined.

B) Relationships that can be used to verify the internal consistency of existing sets of measurement data.

C) Formulas that predict the effect that certain modifications to the system or the workload would have on measured quantities such as throughput and response time.

Note that empirically oriented computer performance analysts are not primarily interested in relationships between random variables or expected values of random variables; rather, they are interested in relationships between measured quantities and quantities which can, in principle, be measured. The remainder of this paper will develop a theory of such relationships. This theory is based on an approach to systems analysis that will be called the "operational method".

4. The Operational Method

In the context of the operational method, an interval of time during which system behavior is monitored and measurement data is collected is called an observation interval. The quantities that are measured or computed during an observation interval are referred to as operational variables. Given this simple conceptual framework, the operational method consists of:

A) Defining a set of operational variables that correspond in a direct and natural manner to intuitive concepts of interest (e.g., throughput, device utilization, etc.).

B) Deriving mathematical relationships among operational variables which characterize system behavior during a single observation interval.

C) Applying these mathematical relationships to problems such as those described in Section 3.

The first set of operational variables that will be defined correspond to basic measurements taken during an observation interval.

T = Length of the observation interval. It is assumed that the time units in which T is expressed are also used to express all other time dependent quantities.

J = Number of jobs completed during the observation interval. A job is a basic unit of work and may refer to a program, a job step, a job, or an interaction, depending on the system being studied.

$B(i)$ = Amount of time that server i is busy (i.e., actually providing service) during the observation interval. It is assumed that there are a total of q separate devices and processors in the system being observed, and that each one is identified by an integer i ($i = 1, 2, \ldots q$). Thus, the term "server i" is understood to represent a unique device or processor for $i = 1, 2, \ldots q$. $B(i)$ is then defined for $i = 1, 2, \ldots q$.

$C(i)$ = Total number of service requests completed by server i during the observation interval ($i = 1, 2, \ldots q$).

An additional set of operational variables can now be defined in terms of T, J, $B(i)$ and $C(i)$.

X = Throughput (i.e., number of job completions per unit time). The operational variable X is expressed as follows:

$$X = J/T. \tag{1}$$

$U(i)$ = Utilization of server i ($i = 1, 2, \ldots q$). $U(i)$ is expressed as follows:

$$U(i) = B(i)/T. \tag{2}$$

$S(i)$ = Average service time for server i ($i = 1, 2, \ldots q$). $S(i)$ is expressed as follows:

$$S(i) = B(i)/C(i). \tag{3}$$

$D(i)$ = Average number of requests for server i per job ($i = 1, 2, \ldots q$). $D(i)$, which will be referred to as the demand per job for server i, is expressed as follows:

$$D(i) = C(i)/J. \tag{4}$$

5. The Throughput Law and Utilization Equality

Given the above definitions, it is possible to derive the following basic result.

Theorem (Throughput Law). Throughput is equal to device utilization divided by the product of average device service time and the average demand per job

for that device. Symbolically,

$$X = \frac{U(i)}{S(i) \cdot D(i)} \qquad \text{for } i = 1, 2, \ldots q. \tag{5}$$

Proof. The Throughput Law is an immediate consequence of Equations (1) through (4) since

$$\begin{aligned} \frac{U(i)}{S(i) \cdot D(i)} &= \frac{B(i)}{T} \cdot \frac{C(i)}{B(i)} \cdot \frac{J}{C(i)} \\ &= J/T \\ &= X. \end{aligned}$$

An important aspect of the Throughput Law is that it is independent of a large number of factors which must normally be specified in other analyses. These factors include: the degree of multiprogramming; the service time distribution for each device and processor in the system; the fact that the system is or is not in statistical equilibrium; the fact that overlap of CPU and I/O processing within a single program is or is not permitted. In other words, the Throughput Law is valid regardless of the status of these other factors.

Corollary (Utilization Equality). The utilization, average service time, and demand per job for any two servers in a system must satisfy the following relationship:

$$\frac{U(i)}{S(i) \cdot D(i)} = \frac{U(j)}{S(j) \cdot D(j)} \qquad \text{for } i = 1, 2, \ldots q;\ j = 1, 2, \ldots q. \tag{6}$$

Proof. Immediate from Throughput Law since Equation (5) must hold for all values of i.

6. Response Time Laws

In order to apply operational techniques to the analysis of interactive systems, it is first necessary to provide a basic model for describing the manner in which such systems operate. The model and terminology that will be employed in this discussion are based on the work of Scherr [14], Kleinrock [11], Moore [12], and Muntz and Wong [13]. The essence of this model is illustrated in Figure 1. There are a fixed number of interactive terminals, and each terminal has one interactive process associated with it. The interactive process is either in think state (i.e., blocked) or system state (i.e., active). Response time is the elapsed time between the instant that an interactive process enters system state and the instant that it next leaves system state. Each time that an interactive process leaves system state (i.e., completes a processing request), an interaction is said to occur.

Given this general context, let J represent the number of interactions completed during the observation interval, and let N be equal to the number of interactive terminals "logged on" during the observation interval. Define the operational variable R as follows:

R = Average response time (i.e., average amount of time in system state per interaction).

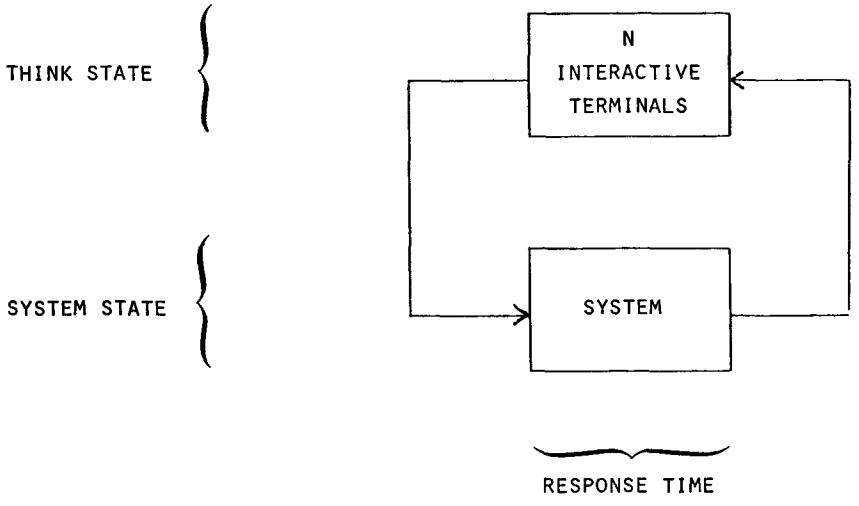

Fig. 1. Basic interactive model

Lemma

$$R = \frac{1}{J} \sum_{k=1}^{N} r(k) \tag{7}$$

where $r(k)$ = total time that the k-th interactive process (i.e., the interactive process associated with the k-th terminal) spends in system state during the observation interval ($k = 1, 2, \ldots N$).

Proof. The average amount of time in system state per interaction is, by definition, equal to the total time spent in system state by all interactive processes divided by the total number of interactions completed during the observation interval. Since $r(k)$ is the total time that the k-th interactive process spends in system state during the observation interval, the total time in system state for all interactive processes is $\sum_{k=1}^{N} r(k)$. The total number of completed interactions is, by definition, equal to J. Equation (7) follows directly.

Before presenting the main theorem, it is necessary to prove another lemma which is similar to the lemma that has just been derived. First define the operational variable Z as follows:

Z = Average think time (i.e., average amount of time in think state per transition from think state to system state).

Lemma

$$Z = \frac{1}{J'} \sum_{k=1}^{N} z(k) \tag{8}$$

where J' = total number of transitions from think state to system state during the observation interval.

$z(k)$ = total time that the k-th interactive process spends in think state during the observation interval ($k=1, 2, \ldots N$).

Proof. The proof is similar to that of the previous lemma. Simply replace R by Z, $r(k)$ by $z(k)$, and J by J'.

Theorem (General Response Time Law). The average response time of an interactive system is expressed as follows:

$$R=N\frac{S(i)\cdot D(i)}{U(i)}-\frac{J'}{J}Z. \tag{9}$$

Proof. The derivation of Equation (9) utilizes the fact that the total time that the k-th interactive process (the process associated with the k-th terminal) spends in think state, plus the total time that the k-th interactive process spends in system state, must be equal to the length of the observational interval. That is

$$z(k)+r(k)=T \quad \text{for } k=1, 2, \ldots N. \tag{10}$$

Thus,

$$\sum_{k=1}^{N} z(k)+\sum_{k=1}^{N} r(k)=N\cdot T. \tag{11}$$

Dividing both sides by J and using Equations (7) and (8) to simplify the left hand side,

$$\frac{J'}{J}\cdot Z+R=N\cdot\frac{T}{J} \tag{12}$$

By Equation (1) and the Throughput Law,

$$\frac{J}{T}=\frac{U(i)}{S(i)\cdot D(i)} \tag{13}$$

Equations (12) and (13) yield Equation (9).

Corollary (Response Time Law or Asymptotic Response Time Law). As the number of interactions that take place during the observation interval becomes large, average response time tends to the limiting value given in Equation (14).

$$\lim_{J\to\infty} R=N\frac{S(i)\cdot D(i)}{U(i)}-Z. \tag{14}$$

Proof. Let z' be equal to the number of terminals in think state at the start of the observation interval, and let z'' be equal to the number of terminals in think state at the end of the observation interval. Clearly,

$$z''=z'+J-J'. \tag{15}$$

Thus,

$$|z''-z'|=|J-J'|. \tag{16}$$

Next note that $|z''-z'|\leqq N$ since $0\leqq z'\leqq N$ and $0\leqq z''\leqq N$. Thus,

$$|J-J'|\leqq N. \tag{17}$$

Dividing by J,

$$|1-J'/J|\leqq N/J. \tag{18}$$

Now consider the effect of allowing J to become large by increasing the length of the observation interval. Since N is fixed, N/J will approach zero as J approaches infinity. By Equation (18), this implies $J'/J \to 1$ as $J \to \infty$. Combining this fact with Equation (9) yields (14).

7. Space-Time Product Laws

Space-time products are often used to evaluate program performance and assign accounting charges to programs in virtual memory systems. Essentially, a program's space-time product is equal to its execution time multiplied by the average amount of memory allocated to it during its execution. Since space-time products, response time, and throughput are all used as indicators of system performance, it is interesting to examine the manner in which these quantities are related.

Theorem (Space-Time Product Throughput Law). Throughput is equal to average amount of memory in use divided by average space-time product. Symbolically,

$$X = M/Y \tag{19}$$

where

Y = Average space-time product (i.e., space-time product per completed job);
M = Average amount of memory in use during the observation interval.

Proof. It is first necessary to define four auxiliary variables for use within the proof.

A = Number of jobs that arrive at the system during the observation interval (including those present at the start of the interval).

$f(k, t)$ = Amount of memory allocated to the k-th job at time t.

$$k = 1, 2, \ldots A \qquad t \in [0, T].$$

$y(k)$ = Space-time product for the k-th job. The operational variable $y(k)$ is defined as follows:

$$y(k) = \int_0^T f(k, t)\, dt \qquad k = 1, 2, \ldots A. \tag{20}$$

$m(t)$ = Amount of memory in use (i.e., allocated to some job) at time t. The operational variable $m(t)$ is defined as follows:

$$m(t) = \sum_{k=1}^{A} f(k, t) \qquad t \in [0, T]. \tag{21}$$

Since Y is defined as average space time product per completed job, Y is equal to the sum of all space-time products accumulated during the observation interval divided by the total number of completed jobs. Thus, Y is given by

$$Y = \frac{1}{J} \sum_{k=1}^{A} y(k). \tag{22}$$

Note that M is defined as the average value of $m(t)$ during the interval $[0, T]$.

Thus,

$$M = \frac{1}{T} \int_0^T m(t)\, dt. \tag{23}$$

By Equations (21) and (23),

$$M = \frac{1}{T} \int_0^T \sum_{k=1}^{A} f(k, t)\, dt. \tag{24}$$

Reversing the order of integration and summation, and applying Equations (20) and (22)

$$M = Y \cdot J/T. \tag{25}$$

Applying Equation (1) completes the proof.

Corollary (General Space-Time Product Response Time Law)

$$R = N\frac{M}{Y} - \frac{J'}{J} Z. \tag{26}$$

Proof. Immediate from Equations (1), (9) and (19)

Corollary (Space-Time Product Response Time Law)

$$R = N\frac{Y}{M} - Z \tag{27}$$

Proof. Immediate from Equations (1), (14) and (19).

Table 2 illustrates the basic structure of the operational variables defined in this paper. Note that some operational variables correspond to events, some correspond to time durations, and some correspond to space-time integrals. Within each of these three classes, certain quantities are expressed for the entire

Table 2. Operational variables

	Event counts	Time durations	Space-time integrals
Entire interval	J A J' $C(i)$	T $B(i)$ $r(k)$ $z(k)$	$y(k) = \int_0^T f(k, t)\, dt$
Per unit time	$X = \frac{J}{T}$	$U(i) = \frac{B(i)}{T}$	$M = \frac{1}{T} \int_0^T m(t)\, dt$
Per job	$D(i) = \frac{C(i)}{J}$	$R = \frac{1}{J} \sum_{k=1}^{N} r(k)$	$Y = \frac{1}{J} \sum_{k=1}^{A} y(k)$
Per service completion		$Z = \frac{1}{J'} \sum_{k=1}^{N} z(k)$	
		$S(i) = \frac{B(i)}{C(i)}$	

observation interval, certain quantities are expressed on a "per unit time" basis, certain quantities are expressed on a "per job" basis, and certain quantities are expressed on a "per service completion" basis.

8. Congestion Law

If it is assumed that each job has unit size (i.e., occupies one unit of memory space), M will be equal to the average number of occupied memory units or, equivalently, the average number of jobs present in the system. It also follows in this case that $y(k)$ is equal to the time-in-system (response time) for the k-th job multiplied by one unit of space. Thus Y is equal to average response time (R) multiplied by one unit of space.

Define the operational variable L as follows:

L = Average number of jobs present in the system during the observation interval.

From the previous paragraph,

$$M = L \times \text{one unit of space}, \tag{28}$$

$$Y = R \times \text{one unit of space}. \tag{29}$$

Theorem (Congestion Law). The average number of jobs present in a system is equal to the throughput of the system multiplied by the average response time of the system. Symbolically,

$$L = X \cdot R. \tag{30}$$

Proof. Substitute from (28) and (29) into (19) and cancel the space units.

Equation (30) is the operational counterpart of the well known result from queueing theory that is usually referred to as Little's formula [9]. Note, of course, that Equation (30) is expressed in terms of operational variables rather than expected values of random variables. Note also that Equation (30) involves throughput (i.e., departure rate) rather than arrival rate (i.e., λ). If it is assumed that the system being studied is in equilibrium in the sense that the same number of jobs are present at the beginning and the end of the observation interval, the departure rate will be equal to the arrival rate and a direct counterpart to Little's formula can be deduced as a special case.

9. Relationship to Probabilistic Results

Since most operational results presented in this paper have counterparts which can be expressed in terms of probabilistic models and random variables, it is useful to consider the relationship between probabilistic and operational analyses. Essentially, operational analyses are concerned with the properties of specific behavior sequences which systems follow during well defined intervals of time. Most results in this paper involve operational variables that represent averages computed over such time intervals, and as a consequence these results apply to all possible behavior sequences which produce a given set of average values.

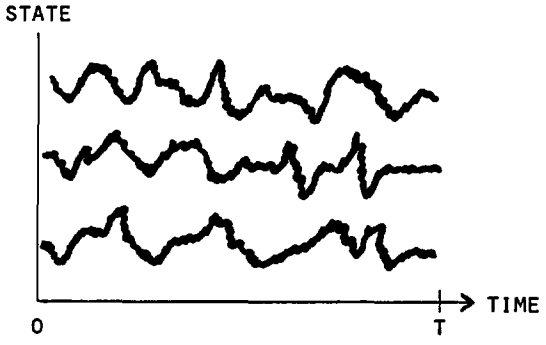

Fig. 2. Behavior Sequences

Figure 2 illustrates this basic idea by indicating three possible behavior sequences that a system may follow during an observation interval of length T. If each of these sequences has the same value of M and the same value of Y, then the Space-Time Product Throughput Law implies that each will have the same value of X. In other words, the Space-Time Product Throughput Law actually applies to families of possible behavior sequences: each family is made up of a set of behavior sequences that have the same values of M and Y, and the Space-Time Product Throughput Law asserts that all members of such a family have the same throughput (i.e., M/Y). Corresponding statements can, of course, be made for other operational laws as well.

In the case of benchmarks, trace driven models, and mathematical forms of deterministic analysis, the analyst must normally specify a set of parameters which completely determines the behavior sequence that the system will follow during the observation interval. Consequently, deterministic results apply to single behavior sequences rather than families such as those illustrated in Figure 2.

As pointed out in Section 1, analysts seldom have sufficient knowledge to specify actual behavior sequences in complete detail. The operational method circumvents this difficulty by only requiring specification of summary statistics such as $U(i)$, $S(i)$, and $D(i)$. Note that analysts are more likely to have accurate knowledge of summary statistics than detailed behavior sequences. In addition, summary statistics are more likely to remain constant from day to day. Thus the operational method can be far more effective than deterministic analyses in dealing with situations where only partial knowledge of the workload exists.

Probabilistic results can also be expressed in terms of summary statistics such as mean service time. Furthermore, probabilistic results also apply to ensembles (i.e., families) of possible behavior sequences of the type illustrated in Figure 2. However, probabilistic results for stochastic processes in equilibrium differ significantly from operational results because probabilistic results only apply precisely (i.e., with probability one) to ensembles which are comprised of infinitely long behavior sequences.

To elaborate upon this point, consider the probabilistic counterpart of the Thr oughput Law. In order to derive such a result, one would normally begin by

assuming that successive service times at device i are given by independent, identically distributed random variables with mean $S(i)$. One would also assume that the demand per job for device i is given by a sequence of independent, identically distributed random variables with mean $D(i)$. One would then add certain technical assumptions needed to guarantee ergodicity (i.e., the existence of an equilibrium probability that server i is active). Denoting this equilibrium probability by $U(i)$, one could then demonstrate that $U(i)/S(i) \cdot D(i)$ is the expected number of jobs completed per unit time in equilibrium (i.e., throughput).

This result would apply to the ensemble of infinite time behavior sequences associated with the stochastic process defined by $S(i)$, $D(i)$ and $U(i)$. In generating each member of the ensemble, the random variables that define "device i service time" and "demand per job for device i" would both be sampled an infinite number of times. Thus, for each member of the ensemble, the Law of Large Numbers will imply that the operational values of "average device i service time" and "average demand per job for device i" will be equal to the means of the corresponding random variables (i.e., $S(i)$ and $D(i)$). Furthermore, since the stochastic process is assumed to be in equilibrium and the observation interval is infinite, the operational values of utilization and throughput will be equal to $U(i)$ and X by the Ergodic Theorem. Thus, the probabilistic and operational versions of the Throughput Law will have the same formal appearance.[1]

This type of reasoning can clearly be applied to other operational laws that involve constants and summary statistics computed over an observation interval. However, probabilistic results cannot be regarded as equivalent to their operational counterparts because:

A) Probabilistic results only apply precisely (i.e., with probability one) in the case where the length of the observation interval is infinite.

B) The assumption of equilibrium sometimes obscures the true identity of critical parameters (viz., Congestion Law).

C) A number of artifical technical assumptions must often be introduced when deriving probabilistic results.

D) In probabilistic analyses, quantities such as throughput and device service time are associated with random variables and stochastic processes. However, such quantities are more naturally regarded as measurable properties of system behavior during specific intervals of time (i.e., as operational variables).

E) When specifying the expected value of a random variable used in a stochastic model, the analyst is in effect specifying the exact value of the corresponding operational variable for each member of the associated ensemble of infinite time behavior sequences. Thus, probabilistic models do not really enable the analyst to specify problems in greater generality; in fact, probabilistic versions of the operational laws derived in this paper are less general (i.e., apply to fewer cases) than the operational versions.

---

1 A precise statement of the points presented in this paragraph would require a lengthy digression into advanced probability theory. Such a digression would add little to the basic content of the argument. For further details, refer to Chapters 7, 8 and 11 of Feller [7]

F) The mathematical techniques required to analyze stochastic models are considerably more complex than those needed in operational analyses. This complexity limits the number of individuals who have access to stochastic models.

Despite these limitations and reservations, stochastic models have proven remarkably successful in predicting performance of actual systems [4]. This is due in part to the fact that many probabilistic results have operational counterparts or operational upper and lower bounds. Since the operational results require fewer assumptions, the probabilistic results sometimes exhibit surprising degrees of robustness. The operational method helps to explain this situation and also provides a bridge between probabilistic results and data collected during actual studies of computer system performance.

A number probabilistic counterparts of the operational laws presented in this paper have been derived previously. In particular, probabilistic counterparts have been derived for the Throughput Law [2], the Utilization Equality [2, 6], the Response Time Law [1, 11–13], and the Congestion Law [8–10]. In some cases, the derivations include arguments that are very similar to those used in the operational method. However, all previous analyses were primarily concerned with random variables and stochastic processes. Thus, additional assumptions were always required to insure ergodicity, and results were always expressed in terms of equilibrium distributions and underlying random variables.

10. Practical Considerations

One question which arises when appliying operational laws in practice concerns the problem of end effects (i.e., the state of the system at the initial and terminal points of the observation interval). Actually, end effects do not in any way impair the validity of operational laws themselves since these laws are internally consistent and valid for all possible initial and terminal states. However, the implications of certain operational definitions should be carefully understood to avoid possible misinterpretation of operational variables.

For example, Equation (3) states that average device service time is equal to total device busy time divided by the number of requests completed during the observation interval. If a service request is partially complete at the start of the observation interval, the value of $S(i)$ will, in a sense, be artificially reduced. Likewise, a service request which is only partially complete at the end of the observation interval will contribute to $B(i)$ but not to $C(i)$. This will artificially raise the value of $S(i)$. Fortunately, if the observation interval is at least moderately long, these effects will be negligible and can be safely disregarded.

The same considerations apply to all variables defined on a "per job" basis (e.g., X, Y, and R). In effect, the operational definitions presented in this paper are based on the assumption that measurements are collected during the observation interval and that "per job" and "per service completion" averages are computed at the end of the interval by simple division. This assumption appears to conform well to existing practice in the computer performance measurement and evaluation field.

Note that the above comments actually pertain to the problem of estimating intrinsic system parameters (e.g., average service time) on the basis of observations

taken during finite intervals of time. This problem arises in both operational and probabilistic analyses, and it is independent of the analysis method that is actually employed.

A somewhat different problem arises in the case of disk and drum subsystems which utilize rotational position sensing and block multiplexing [3]. The application of Equation (3) to such devices yields average service times which are approximately equal to average data transfer times. Thus, the additional delay due to seek and rotational latency will not appear in $S(i)$. Once again, this phenomenon does not affect the validity of the operational laws derived in this paper, but it does illustrate the need to fully understand the implications of all operational definitions.

11. Applications

Since the laws derived in this paper are directly applicable to all systems, they can be used to verify the internal consistency of: performance measurements collected by empirical methods; numerical values generated by simulation programs; algebraic equations derived through the explicit solution of probabilistic models of systems in equilibrium. The laws can also be used to express an unknown variable (e.g., device service time) in terms of other variables which are easier to measure. Additional applications are discussed in an earlier version of this paper [5].

12. Conclusions

Because they apply without restriction to broad classes of systems, the operational laws derived in this paper can be regarded as universal and fundamental laws of computer system performance. They are similar in certain respects to the fundamental laws found in such fields as basic mechanics, thermodynamics, and electrical engineering. Analysts who deal with problems which are not related to computer system performance, but which do involve queueing theory and stochastic models, should also be able to benefit from employing the operational method.

Acknowledgements. Many of the concepts presented in this paper were refined and sharpened during conversations between the author and D. B. Rubin, P. J. Denning, and E. Gelenbe The author was also influenced by the work of R R. Muntz and L. Kleinrock

Appendix A Benchmark Analysis

Tables A-1, A-2, and A-3 present the completion times of individual jobs and the average response time (completion time) of the entire benchmark for each of the three workloads presented in Table 1. It is assumed that the quantum size in the round robin scheduling algorithm is 2 seconds.

Table A-1. Workload = ABC

	FCFS	RR
A	7	11
B	8	3
C	11	8
Average	$8\frac{2}{3}$	$7\frac{1}{3}$

Table A-2. Workload = CBA

	FCFS	RR
C	3	6
B	4	3
A	11	11
Average	6	$6\frac{2}{3}$

Table A-3. Workload = BAC

	FCFS	RR
B	1	1
A	8	11
C	11	8
Average	$6\frac{2}{3}$	$6\frac{2}{3}$

Appendix B Glossary

A = Number of arrivals at system.
$B(i)$ = Amount of time server i is busy.
$C(i)$ = Number of requests completed by server i.
$D(i)$ = Average number of requests per job for server i.
$f(k, t)$ = Amount of memory allocated to the k-th job at time t.
J = Number of jobs (or interactions) completed during the observation interval.
J' = Number of transitions from think state to system state.
L = Average number of jobs in the system.
M = Average amount of memory in use.
$m(t)$ = Amount of memory in use at time t.
N = Number of interactive terminals.
R = Average response time.
$r(k)$ = Total time in system state for k-th interactive process (k-th terminal).
$S(i)$ = Average service time for server i.
T = Length of observation interval.
$U(i)$ = Utilization of server i.
X = Throughput.
Y = Average space-time product per job.
$y(k)$ = Space-time product for k-th job.
Z = Average think time.
$z(k)$ = Total time in think state for k-th interactive process (k-th terminal).

Appendix C Principal Operational Laws

Asymptotic Response Time Law—Equation (14)
Congestion Law—Equation (30)
General Response Time Law—Equation (9)
Response Time Law—Equation (14)
Space-Time Product Response Time Law—Equation (27)

Space-Time Product Throughput Law—Equation (19)
Throughput Law—Equation (5)
Utilization Equality—Equation (6)

References

1. Boyse, J. W., Warn, D. R.: A straightforward model for computer performance prediction. Computing Surveys **7**, 73–93 (1975)
2. Buzen, J. P.: Analysis of system bottlenecks using a queueing network model. Proc. ACM SIGOPS Workshop on Syst Perf Eval., April 1971, Cambridge (Mass.) p. 82–103
3. Buzen, J. P.: I/O subsystem architecture. Proc. IEEE **63**, 871–879 (1975)
4. Buzen, J. P.: Cost-effective analytic tools for computer performance evaluation. Proc. COMPCON 75—IEEE Computer Society Conference, Sept. 1975. p. 293–296
5. Buzen, J P.: Fundamental laws of computer system performance. Proc. ACM-IFIP International Symp. on Computer Performance Modeling, Measurement and Evaluation, March 1976, p. 200–210
6. Chang, A., Lavenburg, S. S.: Work rates in closed queueing networks with general independent servers. Operations Research **22**, 838–847 (1974)
7. Feller, W.: An introduction to probability theory and its applications, Vol. II. New York: Wiley 1966
8. Jewell, W. S.: A simple proof of: $L = \lambda W$. Operations Research **15**, 1109–1116 (1967)
9. Little, J. D. C. A proof of the queueing formula $L = \lambda W$. Operations Research **9**, 383–387 (1961)
10. Maxwell, W. L.: On the generality of the equation $L = \lambda W$ Operations Research **18**, 172–174 (1970)
11. Kleinrock, L.: Certain analytic results for time shared processors. Proc. IFIP Congress 1968, Amsterdam: North-Holland 1968, p. 838–845
12. Moore, C.: Network models for large scale time sharing systems. Dept. Industrial Eng., Univ. of Michigan, Ann Arbor TR-71-1, April 1971
13. Muntz, R. R., Wong, J.: Asymptotic properties of closed queueing network models. Proc. Eighth Annual Princeton Conference on Information Sciences and Systems, March 1974, p. 348–353
14. Scherr, A.: An analysis of time shared computer systems. Cambridge (Mass.): M.I.T. Press 1967

J. P. Buzen
Center for Research
in Computing Technology
Harvard University
Cambridge, Mass. 02138
and
BGS Systems, Inc.
Box 128
Lincoln, Mass. 01773
U.S.A.