AN EXTENSION OF THE CENTRAL SERVER MODEL

Udo Pooch
Don Warner

ABSTRACT

An extension of the central server model as developed by Buzen is analytically developed. This model concerns itself with a computer system that is not heavily loaded. A simulation model is then developed for this lightly loaded computer system, and is used to validate the analytical results of this extended central server model. The analytical calculations regarding resource utilization, as well as the average degree of multiprogramming, generate very close results to those measured from the system.

INTRODUCTION

Queueing models have been an important tool in analyzing the performance of computer systems. Simulation is an alternative to the analysis of computer systems, however, this can lead to high expense in both programming and computer time. The attraction of analytic models for performance evaluation lies in the low cost of computation. Also, a combination of analytic modelling and simulation has been successfully used (1,3). In this paper an analytic model is derived that extends the central server model of Buzen (2) which allows for the study of a computer system that is not heavily loaded. Also included is a validation of the model which uses real data for a system studied by Agrawala and Larsen (1). A simulation model that provides for a lightly loaded system is described in the paper by Agrawala and Larsen (1).

THE MODEL

Buzen (2) develops his central server model to analyze a multiprogrammed computer system using the queueing model solution technique of Gordon and Newell (4). Figure 1 shows a schematic of the central server model. The route taken by a job with probability p_0 is to depict the leaving and concurrent arrival of a job in the system.

To model the system without the assumption of simultaneously arriving and departing jobs, we add another "server" in the system as shown in Figure 2. This additional server is a multiserver with as many servers as the maximum degree of multiprogramming allowed in the system. This station serves as a delay station to model the arrival of jobs at the system at random intervals.

Queueing models of the central server type (2) must have an integer value for the degree of multiprogramming. However, when measuring an actual system the average degree of multiprogramming is very seldom an integer. Thus, the results are presented by interpolating the desired values between those that are obtained from the model with the degree of multiprogramming varied.

The degree of multiprogramming is defined as the number of jobs not at the multiserver. This allows for non-integral degrees of multiprogramming.

Let there be M stations in the system with the M^{th} station the multiserver with N servers. All the other stations are assumed to be the single servers. Let the service time distribution of each station be exponential with parameter μ_i for $i = 1,2,\ldots,M$. The state of the system is determined by the M-tuple $\langle n_1,n_2,\ldots,n_M\rangle$ such that $\sum_{i=1}^{M} n_i=N$, where n_i is the number of jobs at station i. The total number of states of the system is the partition of N requestors over M stations, $\binom{N+M-1}{M-1}$. Notice that the number of servers at station M is N, the maximum degree of multiprogramming. The degree of multiprogramming will be interpreted as the number of jobs not at the multiserver. The multiserver merely serves as a delay station for departing jobs. This station is a multiserver with N servers since a job leaving the system would not have to wait in a queue.

Using $p(n_1,n_2,\ldots,n_M)$ as the probability that the system is in state $\langle n_1,n_2,\ldots,n_M\rangle$, we can write the queueing equations as in (4).

Gordon and Newell (4) obtain the solution to the system by using the following set of equations

$$\sum_{i=1}^{M} P_{ij}(\mu_i x_i) = \mu_j x_j \ , \ j = 1,2,\ldots,M. \qquad (1)$$

or $\underline{y}P = \underline{y}$
where $\underline{y} = \langle \mu_1 x_1,\ldots, \mu_M x_M\rangle$ and

FIGURE 1

Buzen's Central Server Model

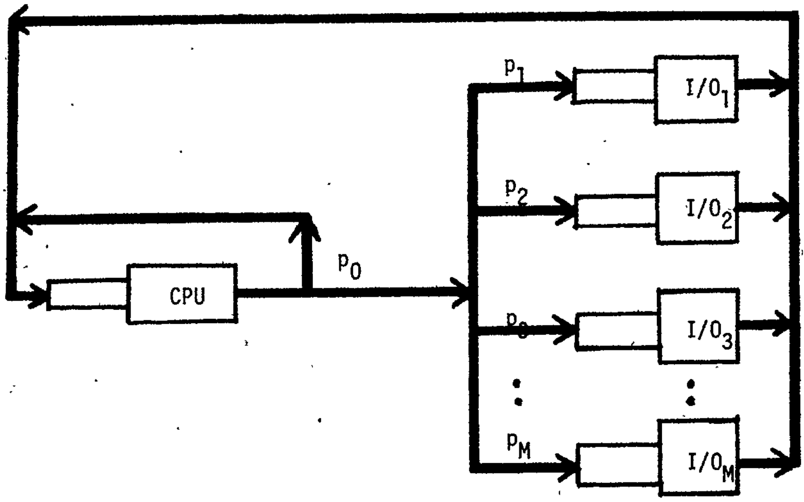
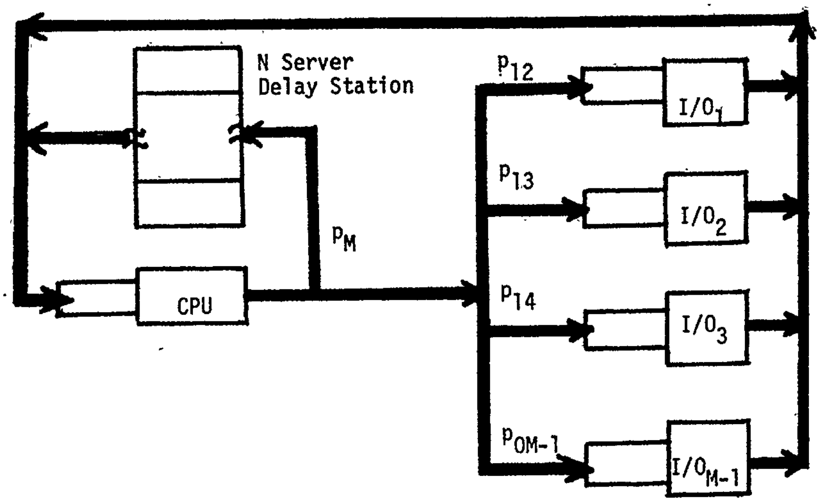


FIGURE 2

Proposed Alternate Model



$$P = \begin{bmatrix} P_{11} & \cdots & P_{1M} \\ \vdots & & \\ P_{M1} & \cdots & P_{MM} \end{bmatrix}$$

is the probability transition matrix. The μ_i's are the mean service times (as for example defined in Table 3), while the x_i is a set of unknown values that will be determined subsequently. The solution for $p(n_1,n_2,\ldots,n_M)$ is given as

$$p(n_1,n_2,\ldots,n_M) = \left\{ \prod_{i=1}^{M} \frac{x_i^{n_i}}{\beta_i(n_i)} \right\} G^{-1}(N) \qquad (2)$$

where $G(N)$ is a normalizing factor, and the function $\beta_k(n)$ for the k^{-th} station is defined as follows:

$\beta_k(0) = 1$

$\beta_k(n) = \alpha_k(n)\cdot\beta_k(n-1)$

$$\text{where } \alpha_k(n) = \begin{cases} n \text{ if } n \le R_k \\ R_k \text{ if } n > R_k \end{cases} \qquad (3)$$

and R_k is the number of servers at station k.

Since all stations except M contain a single server, then in equation (3) $R_i = 1$, $i = 1,2,\ldots,M-1$. Also, we have assumed that $R_M = N$. This leads to

$$\beta_i(n_i) = 1,\ i = 1,2,\ldots,M-1 \qquad (4)$$

and

$$\beta_M(n_M) = B(n_M) = n_M!$$

Let $n = \sum_{i=1}^{M-1} n_i$ then $n_M = N-n$ since there are N jobs, equation (2) because

$$p(n_1,n_2,\ldots,n_M) = \left\{\frac{1}{(N-n)!} \prod_{i=1}^{M-1} x_i^{n_i}\right\} G^{-1}(N). \qquad (5)$$

Since the solution to (1) is arbitrary to within a constant, we can choose $x_M = 1$. This yields

$$G(N) = \sum_{\substack{\forall \text{ states} \\ \sum_{i=1}^{M-1} n_i \le N}} \beta^{-1}(N-n) \prod_{i=1}^{M-1} x_i^{n_i}$$

Using a combinatorial lemma proved by Moore (5) for combinations with repetitions for N distinct objects distributed into M slots

$$G(N) = \sum_{n=0}^{N} \frac{1}{\beta(N-n)} \sum_{i=1}^{M-1} \frac{x_i^{n+M-2}}{\prod_{\substack{j=1\\j\neq i}}^{M-1} (x_i-x_j)} .$$

This may be rearranged into

$$G(N) = \sum_{i-1}^{M-1} \frac{x_i^{M-2}}{\prod_{\substack{j=1\\j\neq i}}^{M-1} (x_i-x_j)} \cdot \sum_{n=0}^{N} \frac{x_i^n}{\beta(N-n)} .$$

This last expression, using (4), simplifies to

$$G(N) = \sum_{i=1}^{M-1} \frac{x_i^{M-2}}{\prod_{\substack{j=1\\j\neq i}}^{M-1} (x_i-x_j)} \cdot \sum_{n=0}^{N} \frac{x_i^n}{(N-n)!} . \qquad (6)$$

ANALYSIS OF THE UTILIZATION OF THE MULTISERVER

Consider the probability that the multiserver is not being utilized, that is, when $n_M = 0$. This may be expressed as

$$\sum_{\substack{<n_1,\ldots,n_{M-1}> \\ \sum_{i=1}^{M-1} n_i = N}} p(n_1,n_2,\ldots,n_{M-1},0).$$

Let U_M be defined as the probability that the multiserver is busy. Then

$$U_M = 1 - \sum_{\substack{<n_1,\ldots,n_{M-1}> \\ \sum_{i-1}^{M-1} n_i = N}} p(n_1,n_2,\ldots,n_{M-1},0).$$

Using (5) and (6)

$$U_M = 1 - \left[\sum_{i=1}^{M-1} \frac{x_i^{M-2}}{\prod_{\substack{j=1\\j\neq i}}^{M-1} (x_i-x_j)} \quad x_i^N \right] G^{-1}(n) \qquad (7)$$

UTILIZATION OF THE SINGLE SERVER

In a manner similar to the method used in the previous section, an expression for the single server utilization may be obtained. Let U_i, $i = 1,2,\ldots,M-1$, denote the utilization of the i^{th} server. To find U_j, $j = 1,2,\ldots,M-1$, let $n_j = 0$ and consider

$$U_i = 1 - \sum_{\substack{<n_1,\ldots,n_{M-1}> \\ \sum_{i=1}^{M-1} n_i \le N}} p(n_1,\ldots,n_{i-1},0,n_{i+1},\ldots,n_M).$$

$$= 1 - \sum_{<n_1,\ldots,n_M>} \frac{1}{\beta(N-n)} \prod_{\substack{j=1\\j\neq i}}^{M-1} x_j^{n_j} \quad G^{-1}(n).$$

Using (6)

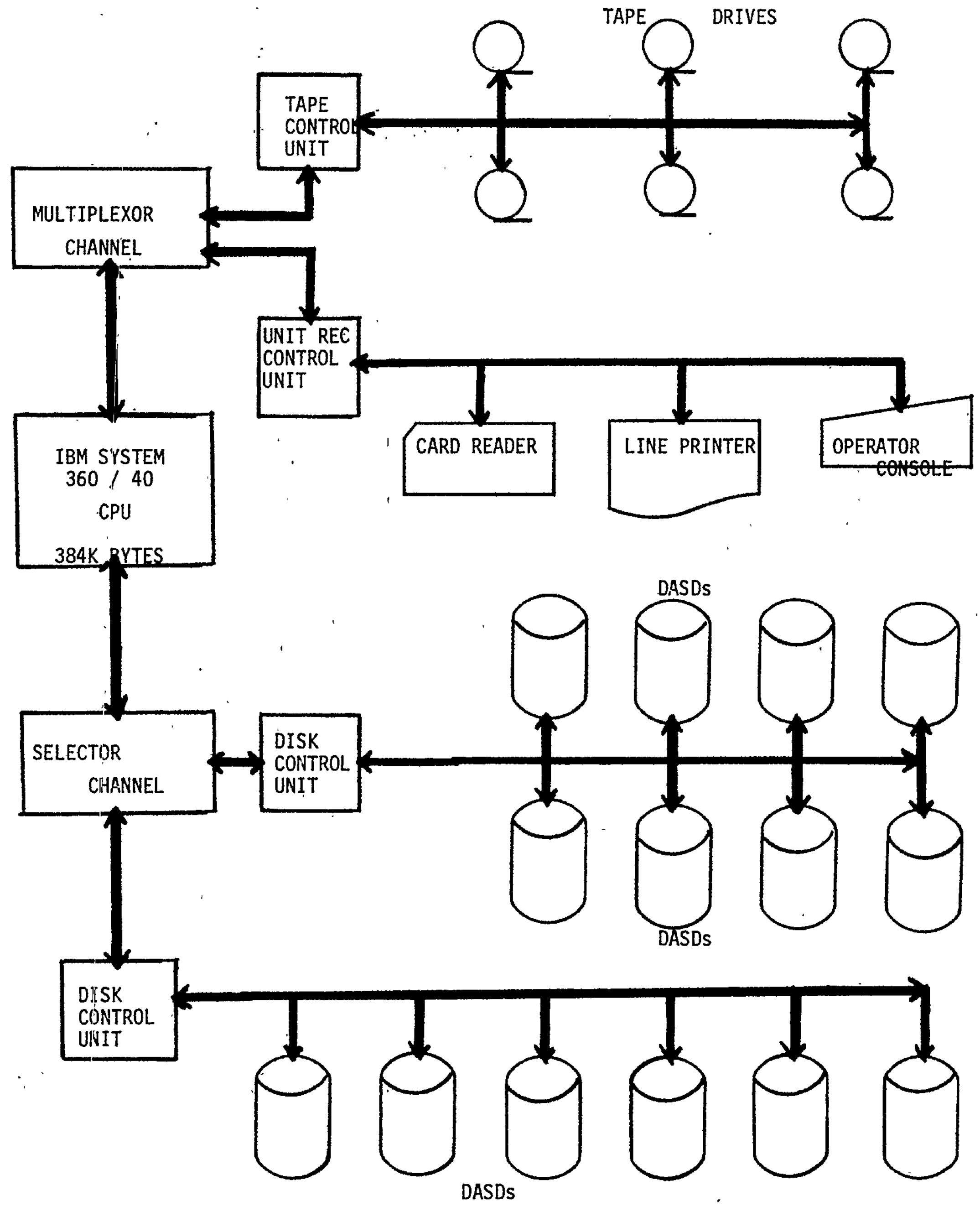
$$U_i = 1 - \sum_{\substack{j=1\\j\neq i}}^{M-1} \frac{x_j^{M-2}}{\prod_{\substack{j=1\\j\neq i}}^{M-1} (x_i-x_j)} \quad \frac{x_j-x_i}{x_j} \sum_{n=0}^{N} \frac{x_i^n}{(N-n)!} . \qquad (8)$$

DEGREE OF MULTIPROGRAMMING

To determine the degree of multiprogramming of the system the average number of jobs at the multiserver must be known. This may be obtained by considering A_M, the average number of jobs at the multiserver. By definition

FIGURE 3

Modelled Computer System Configuration



$$A_M = \sum_{\substack{<n_1,\ldots,n_{M-1}> \\ \sum_{i=1}^{M-1} n_i \leq N}} J(n)\, p(n_1,\ldots,n_M)$$

where

$J(n) = N-n.$

Using equations (2), (4), (6)

$$A_M = \sum_{\langle n_1,\ldots,n_M\rangle} \frac{J(n)}{\beta(N-n)} \prod_{i=1}^{M-1} x_i^{n_i} \; G^{-1}(N)$$

$$= \sum_{n=0}^{N} \frac{J(n)}{\beta(N-n)} \sum_{i=1}^{M-1} \frac{x_i^{M-2}}{\prod\limits_{\substack{j=1\\ j\neq i}}^{M-1} (x_i - x_j)} \; x_i^n \; G^{-1}(N)$$

$$= \sum_{i=1}^{M-1} \frac{x_i^{M-2}}{\prod\limits_{\substack{j=1\\ j\neq i}}^{M-1} (x_i - x_j)} \sum_{n=0}^{N} \frac{(N-n)\,x_i^n}{(N-n)!} \, . \qquad (9)$$

Using equation (9) the average degree of multiprogramming, D, may now be found as $D = N - A_M$. (10)

VALIDATION OF THE MODEL

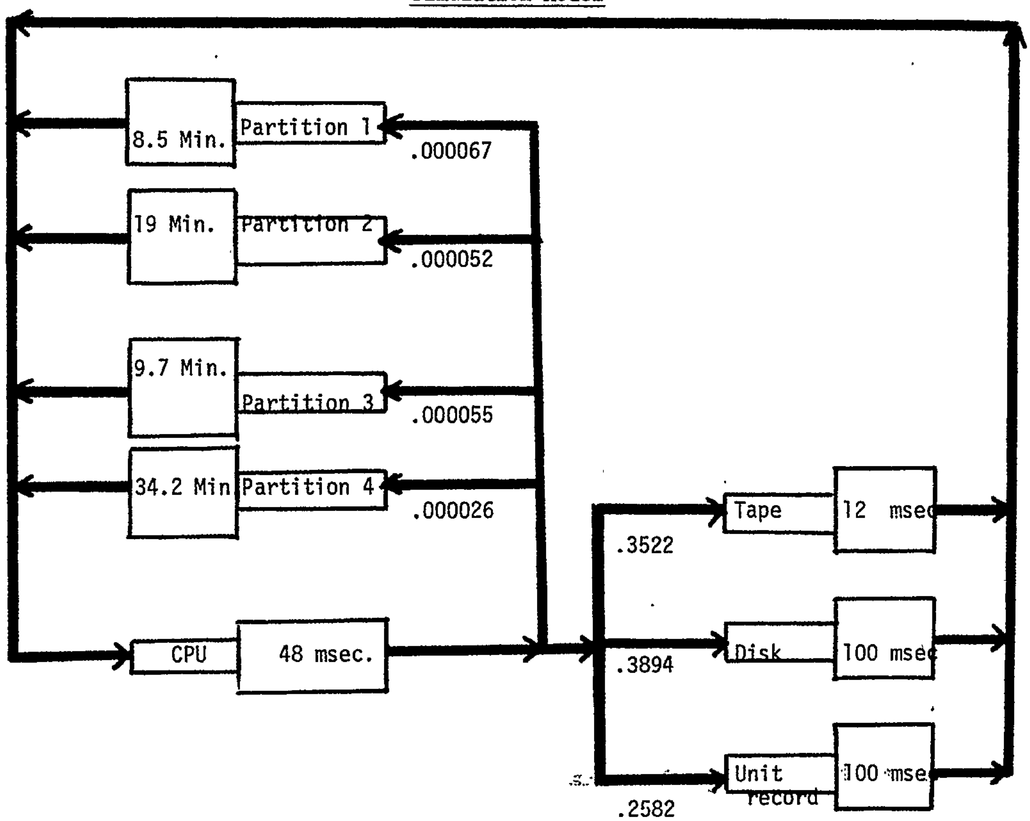
The system modelled by Agrawala and Larsen is an IBM System 360/40 computer system operating under OS/MFT with HASP. The configuration of this system is depicted in Figure 3, with the corresponding simulation model illustrated in Figure 4. The system contains a selector channel and a multiplexor channel. Attached to the selector channel are two control units, each of which controls a bank of disk drives. The multiplexor channel also is connected to two control units. One controls six tape drives, while the other manages the unit record equipment -- card reader, line printer, and the operator's console.

The performance specifications for each I/O device is taken from the appropriate IBM documentation. Requests to and from the operator's console is assumed to be negligible. The mean service time of the processor is estimated by dividing the total processor time by the number of service periods (i.e. total I/O requests). The probability that a job terminates after a processor service period is estimated by dividing the total number of jobs completed by the total number of service periods. The probability of requesting the service of a particular I/O device class after completing a CPU service period is estimated as the joint probability that a job does not terminate and that the request is for that class of device. The probability for a request of a particular I/O device is computed by dividing the total number of requests for that class by the total number of I/O requests.

The model developed by Agrawala and Larsen (1) adds four additional "servers" to model the partitions. However, the model allows a queue capacity of one

FIGURE 4

Simulation Model



at each partition server. Since Buzen's model does not allow this change, Agrawala and Larsen are forced to simulate this modification. In the simulation model, if a job is routed to a partition that is already busy, the new arrival is turned away and the simulation model attempts to reroute the job by selecting another server in accordance with the transition probabilities.

To validate the analytic model developed in this paper, the transition probabilities and service times of the CPU, tape, disk, and unit record device classes are those as obtained by Agrawala and Larsen (Tables 1, 2, and 3). The individual partition servers are consolidated into a service station with four servers. The transition probability to this server remains the same as that obtained in (1). The holding (service) time of this "server" is determined by using the weighted average of the idle times of each partition (Table 4).

TABLE 1

System Accounting Data

Period of measurement	11.25 hours
Total processor time	6.12 hours
I/O count for tape	161000
I/O count for disk	178000
I/O count for unit record	118000
Total I/O count	457000
Number of completed jobs	
Partition 1	33
Partition 2	26
Partition 3	27
Partition 4	13
Total	99

TABLE 2

Transition Probabilities

$$P\text{ (termination of a job)} = 99/457000 = .0002$$

$$P\text{ (CPU to tape)} = \frac{161000}{457000}(1 - .0002) = .3522$$

$$P\text{ (CPU to disk)} = \frac{178000}{457000}(1 - .0002) = .3894$$

$$P\text{ (CPU to unit record)} = \frac{118000}{457000}(1 - .0002) = .2585$$

TABLE 3
Mean Service Times
μ_1 (CPU) = 48 msec
μ_2 (tape) = 12 msec
μ_3 (disk) = 100 msec
μ_4 (unit record) = 100 msec
μ_5 (delay station) = 894 sec

TABLE 4

Mean Idle Time for Each Partition

Partition	Mean Idle Time
1	8.5 minutes
2	19 minutes
3	9.7 minutes
4	34.2 minutes

Table 5 presents the utilization of the system as measured, predicted by the simulation model in (1), and predicted by the analytic model developed in this section. The utilization of each processor was determined using equation (8). The average degree of multiprogramming was determined by equation (10).

CONCLUSIONS

The extension of the central server model to include lightly loaded systems is segnificant. The model presented here demonstrates that analytic models may be as accurate and still less expensive than similar simulation models.

The program used to evaluate this model was run on an IBM 360/95 and took .05 minutes to run.

REFERENCES

1. Agrawala, A.K., and Larsen, R.L. Experience with the central server model on a lightly loaded system. Symposium on Simulation of Computer Systems.
2. Buzen, J.P. Queueing network models of multiprogramming. Ph.D. Thesis, Harvard University (May 1971).
3. Chiu, W., Dumont and Wood, R. Performance analysis of a multiprogrammed computer system. IBM J. Res. Develop. (May 1955), 263-271.
4. Gordon, W.J., and Newell, G.F. Closed queueing systems with exponential servers. Oper. Res. 15 (April 1967), 254-265.
5. Moore, F.R. Computational Model of a closed queueing network with exponential servers. IBM J. Res. Devel. (November 1972), 567-572.

TABLE 5

Measured and Simulated Utilization of the Computer System

	Measured	Agrawala and Larsen	Warner and Pooch
CPU	.54	.52	.52
Tape	.05	.05	.05
Disk	.44	.42	.43
Unit Record	.29	.27	.28

AVERAGE DEGREE OF MULTIPROGRAMMING

	Measured	Agrawala and Larsen	Warner and Pooch
	1.83	1.69	1.81