# On Performance Evaluation of Multiprogramming Systems

Masatoshi MIYAZAKI*, Shingo TOMITA**, Shoichi NOGUCHI*
and Juro OIZUMI***

**Abstract**

For performance evaluation of multiprogramming systems, a queueing model with a finite waiting room is considered. The model assumes that a computer system consists of a central processor and identical input/output channels, and that the service times of both units are exponentially distributed independent random variables. Then the queueing model is described as M/M/S(N), where S is number of input/output channels and N is degree of multiprogramming.

Processor productivity is defined by a steady state probability. Compared with measured and calculated data of processor productivity, the model validation is discussed. Furthermore, an overhead of a monitor in multiprogramming systems is also considered.

## 1. Introduction

Many efforts for performance evaluation of multiprogrammed computer systems have been performed in various ways. There are three approaches in a general classification such as measurement of real systems, simulation and a mathematical model. Each method may have some advantages and disadvantages. In this paper, we use the technique of mathematical modeling.

A mathematical approach to performance evaluation has been discussed widely during the past several years.[1),2)] This method usually has several assumptions which cause different results from actual systems, consequently. On the other hand, for a more accurate description of the systems, the mathematical model may have a very complicated structure. It would not be feasible to apply the model to other general systems for its complication.

From the point of view of features in mathematical modeling, we consider a simple queueing model of multiprogrammed computer systems.

This paper first appeared in Japanese in Joho−Shori (Journal of the Information Processing Society of Japan), Vol. 16, No. 12 (1975), pp. 1,040~1,047.

* Tohoku University
** Yamaguchi University
*** University of Electro-Communication

## 2. Multiprogramming Model

A computer system discussed in this paper consists of a central processor(CPU) and identical input/output(I/O) channels. A CPU and each I/O channel are operable independently. There are always N processes in a main memory unit. Executions of these processes are concurrently performed by CPU and I/O channels in a multiprogramming manner. In other words, a system is in heavy load condition; that is, a new process comes in a memory whenever any process leaves the system after its completion of whole execution.

An execution of a process in single processing may be a series of two alternate states being served by CPU and by I/O channel. We assume that processing times of CPU are independent, identical, exponentially distributed random variables with mean $1/\lambda$, and that processing times of I/O channels are also exponentially distributed with mean $1/\mu$.

In multiprogramming systems, I/O requests from processes arrive at I/O channels through CPU with arrival rate $\lambda$ while at least one process which can be served by CPU exists. The I/O requests will make a queue in front of I/O channels. The maximum queue length may be equal to N, where N is the number of processes in a main memory and it is called degree of multiprogramming. When the queue length is N; that is, all processes are waiting the completion of their I/O operations; CPU becomes idle, and then no I/O request arrives at I/O channels.

No arrival of I/O request may be equal to that an I/O request leaves the system as soon as it arrives at I/O channels. This queueing system is well known as a queue with a finite waiting room. Thus, multiprogrammed computer systems are represented as M/M/S(N), where S is number of I/O channels. Figure 1 shows the multiprogramming model with a finite waiting room.

## 3. CPU Productivity

Let us consider only a CPU productivity in steady state of systems as a factor of performance. The queueing system M/M/S(N) is already analyzed. Let n ($0 \leq n \leq N$) be the number of I/O requests either waiting or being performed on I/O channels, and $P_n$ be the steady state

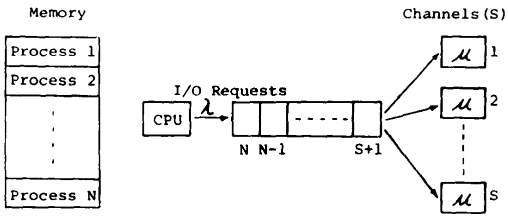


Fig.1 Multiprogramming model with a finite waiting room.

probability that n I/O requests exist in the system at any moment. $P_n$ is given as follows:

$$P_n = \begin{cases} \dfrac{\rho^n P_0}{n!} , & (1 \leqq n \leqq S) \\ \dfrac{\rho^n P_0}{S!S^{n-S}} , & (S < n \leqq N), \end{cases}$$

$$P_0 = \left\{ \sum_{n=0}^{S} \frac{\rho^n}{n!} + \frac{\rho^S}{S!} \sum_{n=1}^{N-S} \left(\frac{\rho}{S}\right)^n \right\}^{-1},$$

where

$$\rho = \lambda/\mu ,$$

and the constraint

$$\sum_{n=0}^{N} P_n = 1$$

should be given.

The CPU will not become idle but continue to execute any process while n is less than N. The probability that the CPU can execute any process is given by $1-P_N$. A CPU productivity denoted by $\alpha$ may be defined with this probability as follows:

$$\alpha = \begin{cases} 1 - \dfrac{\rho^N P_0}{S!S^{N-S}} , & (S < N) \\ 1 - \dfrac{\rho^N P_0}{N!} , & (S \geqq N) , \end{cases}$$

$$P_0 = \begin{cases} \left\{ \sum_{n=0}^{S} \dfrac{\rho^n}{n!} + \dfrac{\rho^S}{S!} \sum_{n=1}^{N-S} \left(\dfrac{\rho}{S}\right)^n \right\}^{-1} , & (S < N) \\ \left\{ \sum_{n=0}^{N} \dfrac{\rho^N}{n!} \right\}^{-1} , & (S \geqq N) \end{cases} \qquad (1)$$

Figure 2 gives a comparison between measured and calculated data of a CPU productivity for the sake of model verification. The measured data were obtained from the system (NEAC 2200 Model 700) implemented in Tohoku University Computer Center by a software metering technique.

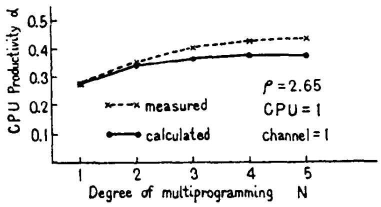


Fig. 2 A comparison between measured and calculated data of CPU productivity.

The value of $\rho$ can be given by the CPU productivity of the case of N=1. Then we get the calculated data from equation (1). The calculated productivity would be comparatively equal to the measured. Consequently, we might conclude that the queueing system with a finite waiting room can be applied to performance evaluation of multiprogramming systems.

The next two graphs give numerical examples. Figure 3 shows an effect of the degree of multiprogramming and values of $\rho$ upon a CPU productivity when S=1. A CPU productivity increases and reaches 1 according to increase of N when $\rho \leqq 1$; but in the case of $\rho > 1$, its maximum value cannot exceed $1/\rho$ . Figure 4 shows the effect of the number of I/O channels upon a CPU productivity in the case of $\rho=2$. It is seen that the number of I/O channels affects a CPU productivity considerably.

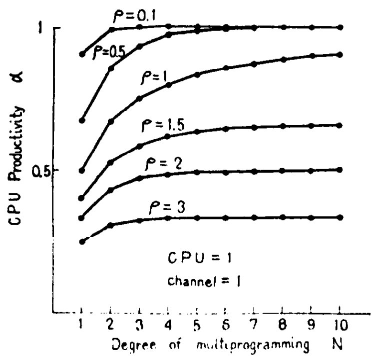


**Fig. 3** Effect of degree of multiprogramming and $\rho$ upen CPU productivity.

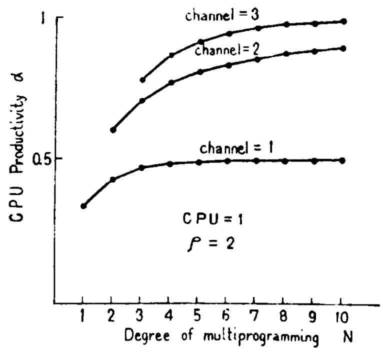


**Fig. 4** Effect of number of channels upon CPU productivity.

4. Overhead Effect

The CPU productivity discussed above includes an overhead of a monitor. The effective work of a CPU upon processes should decrease in the case of much overhead even if a CPU productivity may be comparatively large. In this section, a consideration on an overhead effect of a monitor is given briefly.

We assume that an overhead time $\delta$ is included in the mean time $1/\lambda$ of random variables of CPU processing times. Then,

$$\eta = \frac{1/\lambda - \delta}{1/\lambda} = 1 - \lambda\delta$$

means a ratio of the effective work in $1/\lambda$ . The effective work of a CPU upon processes, we call it a process productivity denoted by $\beta$ , is defined as follows:

$$\beta = \eta\alpha \ . \qquad (2)$$

A comparison between measured and calculated process productivity is shown in Figure 5. The measured data is based on Figure 2. The ratio $\eta$ can be obtained from the case of N=1 in the measured data. It is obvious that the process productivity calculated from equation (2) is nearly equal to the measured data.

5. Conclusions

A queueing system with a finite waiting room has been indroduced for performance evaluation of multiprogrammed computer systems. Although a number of assumptions were made to simplify the analysis of the model, we found reasonable correlation between the measured performance and the calculated prediction in the model validation. The queueing model M/M/S(N) is very tractable. It may be easy to apply the model to multiprocessor computer systems with slight modification.

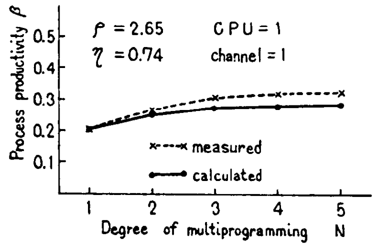

Fig. 5 A comparison between measured and calculated data of process productivity

Acknowledgment

The authors wish to thank S.Matsuzawa and S.Obata of Tohoku University Computer Center for their measurement of the system and many constructive suggestions.

References

1. Adiri,I.,Hofri,M.,and Yadin,M. A Multiprogramming Queue, J.of ACM, Vol.20, No.4, 1973, pp.589-603.

2. Gaver,D.P. Probability Models for Multiprogramming Computer Systems, J.of ACM, Vol.14, No.3, 1967, pp.423-438.