Simulation of a Computer System

Masanori Kanazawa*, Hajime Kitagawa* and Hiroshi Hagiwara**

Abstract

The intention of this paper is to describe the use of simulation method in the analysis of a multiprogramming computer system. In this simulation the model and input data are given on the basis of the measurements on an actual system. We describe a few techniques, taking the efficiency of simulation into consideration. The results of the simulation and the effectiveness of the presented techniques are shown.

1. Introduction

We have analyzed and evaluated the performance of a multiprogramming system by means of the simulation method, while a few techniques have been examined which might serve to enhance the efficiency of simulation.

In the stage of constructing a model of the actual system, it is important to take the points, which are supposed to be critical to the system performance, into the model to a sufficient degree, and to formulate assumptions within such a scope that their influence on the result of simulation should be exerted as little as possible. The difference between the actual and the simulated systems also comes from that of actual data and input ones. Not only in simulation but also in an effective analytical method, if any, we shall have to utilize the actual data in order to undertake a practical analysis.

Another problem is one of technical treatments, which concerns, first of all, the computing time in each simulation run and the accuracy of the results. This is to balance between computing time and the accuracy according to the initial conditions and termination conditions for bringing the simulation to an end. In the system simulation, then, we often want to understand the relation between various parameter values and their results, or to pursue the parameter value

This paper first appeared in Japanese in Joho – Shori (Journal of the Information Processing Society of Japan), Vol. 13, No. 10 (1972), pp. 684～691.

* Data Processing Center, Kyoto University

** Faculty of Engineering, Kyoto University

which gives the optimal performance. As it typically requires a great deal of computing time of the simulation, it is effective for us to find how to shorten the time necessary for the whole simulation run on behalf of the efficiency of simulation.

The subject matter in our simulation was FACOM 230-60 multiprogramming system in Data Processing Center, Kyoto University.

2. Analysis

For understanding of the dynamic behavior of the object system, we make use of CPU time(T_{CPU}), core time(T_{CORE}), program size, number of lines printed(L) et al recorded as the accounting informations.

FORTRAN jobs, which may be considered to represent the ordinary users' jobs, were processed in the uniprogramming scheme(i.e. one degree of multiprogramming), and the distributions of net I/O(input/output) and scheduling times can be found. Statistics of them is obtained on each jobstep. See Fig. 1 and 2. In Fig. 1, we formulated the realation between the indicator of I/O complexity and I/O time in such a equation as

$$T_{I/O} = ax + b, \qquad x = T_{CPU} \text{ or } L.$$

3. Model

We have constructed the model of a proposed computer configuration, with emphasis on scheduling run and blocked I/O operations which are key factors exerting large influence upon the performance of the multiprogramming system. See Fig. 3. A job is composed of some jobsteps, each of which is devided into three parts; initiator, processing program and terminator. We showed the structure of them in Fig.4. Such a job stream flows with others in several degrees of multiprogramming

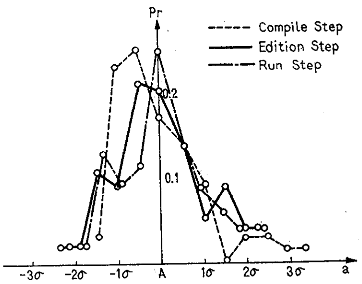

Fig. 1 Probability Density of 'a'

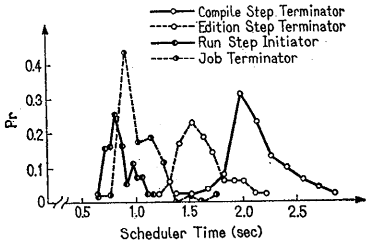

Fig. 2 Probability Density of Scheduler Time

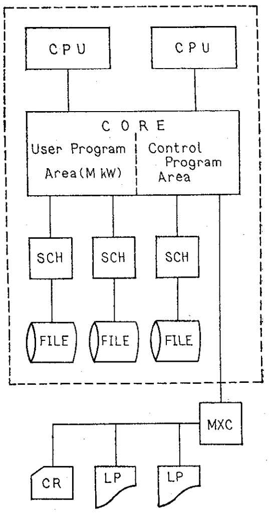

Fig. 3 System Configuration

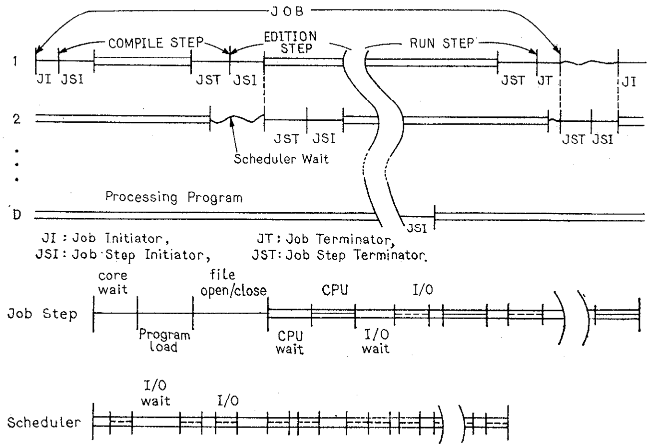

Fig. 4 Flow of D Jobs Processed in Multiprogramming Environment

set by operators.

4. Structure of the simulator and input data

The simulator is, as shown in Fig. 5, composed of three parts; the strategy program, the simulation program and the monitoring program. The strategy program sets the parameter values for termination conditions and determines the parameter values for the simulation program. The monitoring program examines the fluctuation of results obtained by the simulation program.

CPU time and program size are given as input data from the accounting information, on the other hand, I/O and scheduling times are generated in terms of random variables according to the analysis of the system. And then, we reproduced workload of a particular day on the object system.

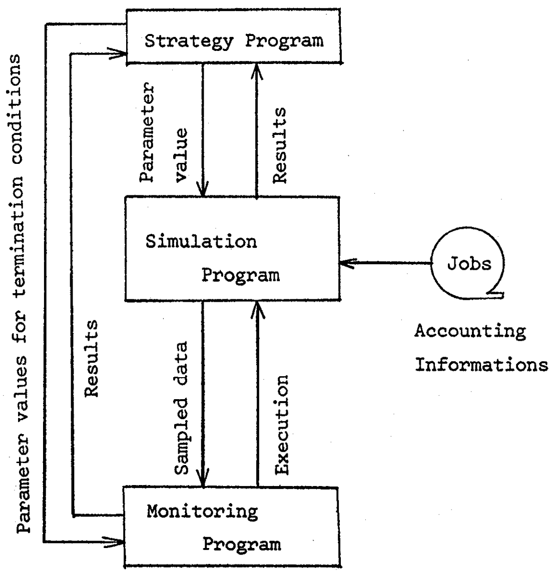

Fig. 5 Simulator

5. Measures of the system performance

CPU speed ratio, core size of users' area and degree of multiprogramming are selected as important parameters*, and utilization of each resource, scheduler run(ratio of sum of scheduling time to system running time), effective degree of multiprogramming and so on are picked up as measures of the performance.
We define the measure of the total system as the value P_c.

$$P_c = (T / \sum_i C_i)(\sum_i C_i U_i / \sum_i C_i) \tag{1}$$

where C_i: cost of resource i

T : number of jobsteps processed in unit time

U_i: utilization of resource i

i : resource (CPU, SCH/FILE, CORE)

The cost of CPU is assumed to be inversely proportional to the square root of CPU speed ratio, i.e. Grosch's law.

6. Efficiency of simulation

(i) convergence

It becomes material for discussion to determine the necessary and sufficient computing time for simulation in order to acquire the proposed accuracy at the equilibrium in simulation stage.

Using the central limit theorem and the interval estimation method, the confidence interval of the mean value is

$$2 \cdot t_\phi(\varepsilon/2) \cdot S / \sqrt{n - 1} , \tag{2}$$

where $t_\phi(\varepsilon/2)$: t-distribution, $\phi = n - 1$ (degree of freedom) and

S^2 : the maximum likelihood estimator of the standard deviation.

By means of them, the proposed accurate results are acquired without extra computation in a simulation.

(ii) initial conditions

The convergence of results becomes slow when the simulation run begins with empty and idle status. This is the reason the initial bias exerts the large

---

* Other parameter values were determined to be corresponding to the object system, CPU speed ratio is average instruction execution time, where FACOM 230-60 CPU speed ratio is 1.0.

influence into the estimation of the confidence interval. To reduce this initial bias, following two methods can be considered.

(*a*) Several sampled results obtained near the initial status are neglected.

(*b*) The simulation program keeps its own status at end of an experiment and continue to simulate another with only a slightly different parameter value from last one. This is called the *slide method.*

(iii) Direct search method

Pursuing the parameter value which gives the optimal performance with respect to P_c, the direct search method seems to be useful. In our simulator, the strategy program includes the routine which selects the parameter value according to the algorithm of Hooke and Jeeves method as well as the slide method.

7. Results

In order to evaluate the performance of the system under equilibrium or stability, the mean values of performance measures are sampled by 8,000 task-switches and their confidence intervals are estimated. When they are within 5 % of mean values, the simulation run is brought to an end. The resultants of the simulation

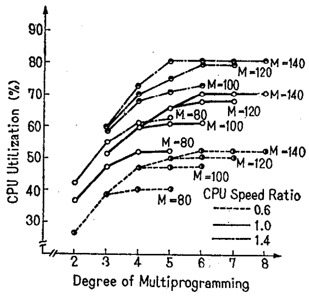

Fig. 6 CPU Utilization

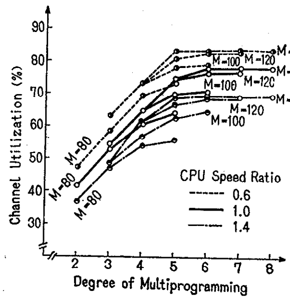

Fig. 7 Channel Utilization

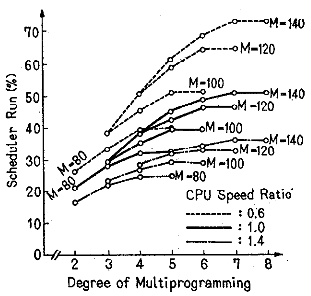

Fig. 8 Scheduler Run

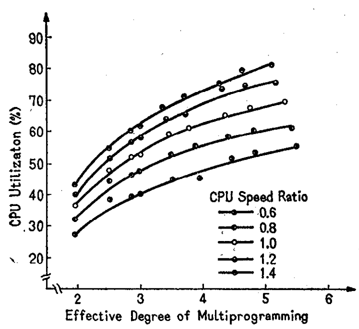

Fig. 9 CPU Utilization

are shown in Fig. 6, 7, 8 and 9.

The slide method helps reducing the computing time in this simulation. Seeing Fig. 10, the effectiveness of the slide method is obvious. In Fig. 11, the optimal performance of P_c was obtained by the direct search method. This method is also effective for shortening the total computing time for simulation.

The model with parameter value (1.0, 120, 5 or 6) corresponds to that of the object system. In order to use CPU more effectively, the characteristics of the jobs in this simulation was taken into account, and now the system is operating

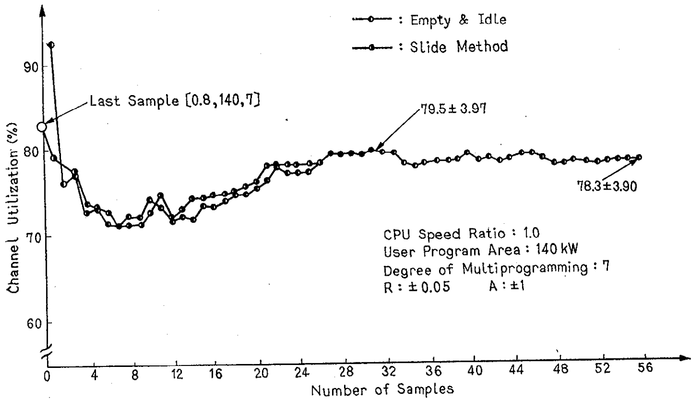

Fig. 10 Comparison of Convergence (Channel Utilization)

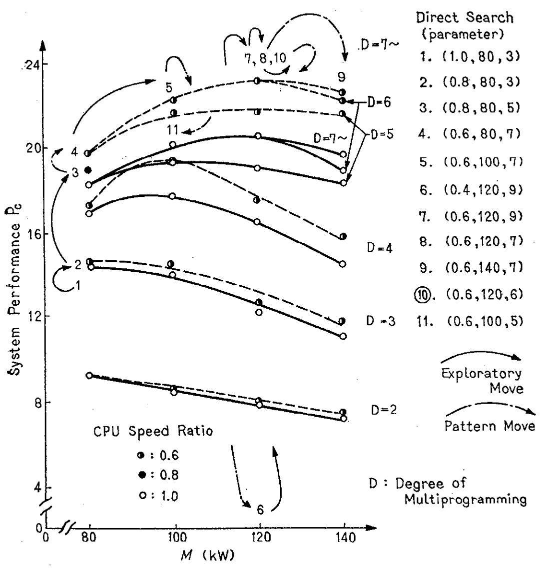

Fig. 11 System Performance and Maximum Value by Direct Search Method

on the condition of job-mix including long run jobs. As a result, it turns out by measurements that CPU utilization is about 10 % higher and channel utilization is about 20 % lower than the case of the simulation. Hence the system can be thought to attain to the optimal performance.

8. Conclusion

We have tried to evaluated the performance of the computer system under equilibrium by means of simulation and had more knowledge of the dynamical behavior of the system. The slide method and the direct search method are helpful to shorten not only the computing time of simulation run but also thinking time for setting the parameter value. It is important for the more excellent evaluation to use simulation method together with monitoring method.

This work was supported in part by the Project of Data Processing Center, Kyoto University.

Reference

1) Calingaert, P. : "System performance evaluation: survey and appraisal", Comm. ACM, Vol. 10, No. 1, pp.12-18, 1967.

2) Fine, G.H. and P.V. McIsaac : "Simulation of a time-sharing system", Management Science, Vol. 12, No. 6, pp.180-194, 1966.

3) Kowalik, J. and M.R. Osborne : "Methods for Unconstrained Optimization Problem", Elsevier, 1968.

4) Fujitsu : "Manual, FACOM 230-60 Monitor-V System", EX-041-2/EX-043-2, 1969.

5) Kitagawa, H., M. Kanazawa, H. Hagiwara and Y. Fujiwara : "Monitoring and evaluation of system performance", J. of Information Processing Society of Japan, Vol. 13, No. 11, pp.789-797, 1972.