Running Time Delays in Processor-Sharing System

MISAKO ISHIGURO*

An analytic method is developed to estimate the average running time for a job in a processor-sharing system, where the running time includes the delays caused from the execution of coexisting jobs. The processor-sharing mechanism is approximated by a finite number of pseudo processors called "init". In the present treatment, Poisson arrival and exponential service time distributions are assumed.

An integrated expression is given to obtain the average delayed job running time for the following three types of systems.

(1) Multiple separate system with multiple separate queues, where the inits correspond to the individual queues.

(2) Common init system with a single queue, where all the inits correspond to the queue.

(3) Job class system with separate job class queues, which is a melting of the preceding two systems, where the individual queues are assigned to some of the inits.

The results calculated from the present method are shown to be reasonable in light of validation of the results by simulation.

1. Introduction

It is an interesting problem to establish a reasonable analytic method to estimate the running-time delays of jobs in a processor-sharing (PS) system under an open batch type of job scheduling. In the system, a single processor is shared among a finite number of jobs, thus the running time of a job may become longer, depending on the extent that the processor is occupied by other jobs.

As for the PS mechanism, several studies have been made [1, 2] in which the associated mean waiting time, $W(t)$, was evaluated for a service requirement, t. The studies, however, were intended to investigate, not the batch type of system, but rather the time sharing system (TSS). In contrast with the TSS, the batch system jobs are scheduled by job classes, and a job arriving at the system may not immediately enter the PS service stage, but will be held in the queue to keep the bounds of the multiprogramming level. The level is always designated to each job class according to the practical limitations. Consequently, in the present PS service stage, several classes of jobs are served in parallel, under the finite multiprogramming levels. On the other hand, other models that appeared in [3, 4] are the so-called "two queues attended by a single server". Though the server is accompanied by two distinct queues, they were concerned with scheduling models that are dissimilar to the PS system. That is, in the model by Takács [3], two queues are alternately served, while in the model by Taube-netto [4], two service stage queues in tandem-attended are served by switching one stage to another.

In order to take the batch type multi-job-class queues into account, a PS queuing system, as shown in Figs. 1 and 2, is considered. The PS mechanism is shown to be approximated by a finite number of pseudo processors called "init". Then, we are led to the concept of init by the job scheduling scheme used in IBM S/360 OS/MFT

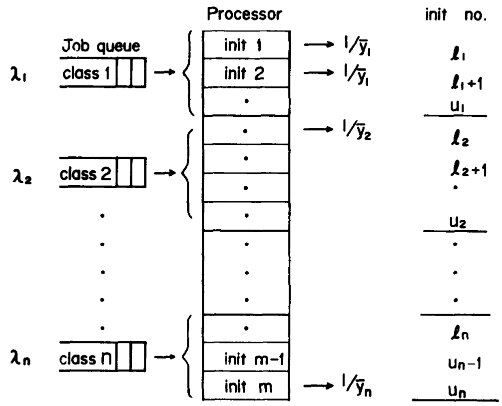

Fig. 1 Processor shared-queuing model.

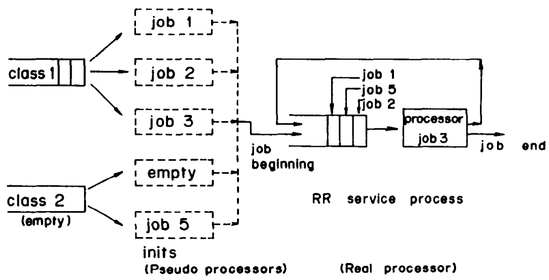

Fig. 2 RR service queue in the PS model.

*Japan Atomic Energy Research Institute, Tokai, Ibaraki, Japan.

[5]. Thereupon, job scheduling is performed by an initiator-terminator task that consists of a finite number of job initiators, and each of the initiators, abbreviated to "init," corresponds to a job class queue, or queues, and schedules jobs for the queues assigned to it by selecting a candidate by the FIFO rule. In the present system, all the jobs actively served by the individual inits are alternately processed by a real processor through the Round-Robin (RR) service discipline. Here, it is assumed that the real processor is equally shared-used among the active jobs, regardless of their job classes. Because a time slice must be very small compared with the service requirement of such a batch type of job, a job will spend a good many cycles of the RR processes. Consequently, at the PS stage, the use of the real processor is considered to be averaged among the coexisting jobs.

Now, the generalized effective models of such a job-scheduling system of queues are difficult to analyze, except under restrictive assumptions. Accordingly, in the present system macroscopic behaviors of the computing system are analyzed, though in an actual computing system, a job is processed through the CPU and input/output iteration with a requisite storage. Here, a basic part of the multiprocessing is modelled and analyzed to obtain a rough estimate, so the input/output times for auxiliary storages and others are not considered individually. All such factors are assumed to be lumped under a behavior of a single processor.

From the above discussion, we shall make the usual assumptions that a job entered into the i-th queue has Poisson arrivals at a mean rate, λ_i, exponential service times at a mean value, $\bar{s}_i$, and the FIFO scheduling rule is used for each queue.

To illustrate, the motivation to the present problem, we shall consider the simplest system, which has two inits corresponding to two individual queues, as shown in Fig. 3. Let us denote the average (delayed) running time for the j-th init associated with the average (real) service requirement $\bar{s}_i$ (for $i=1$, 2) by $\bar{y}_i$. We shall roughly estimate the $\bar{y}_i$ by taking into account the mutual influences between the two inits. Precisely, the running time of an init, say init 1, will double if another init (init 2) is busy, assuming that the two jobs are alternately served in the RR service processes. Then $\bar{y}_1$ is presumably calculated as

$$\bar{y}_1 = 2\bar{s}_1\eta_2 + \bar{s}_1(1-\eta_2) = \bar{s}_1(1+\eta_2), \quad \text{where } \eta_2 = \lambda_2\bar{y}_2, \tag{1}$$

Similarly, for init 2

$$\bar{y}_2 = \bar{s}_2(1+\eta_1), \quad \text{where } \eta_1 = \lambda_1\bar{y}_1. \tag{2}$$

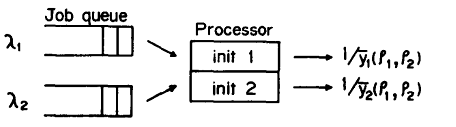

Fig. 3 Two separate queues.

In these two equations, we have introduced the definition, η_i, to be the pseudo utilization rate for the i-th init. From the consequence of (1) and (2), η_i ($i=1$ and 2) is given as a function of ρ_1 and ρ_2, where ρ_i is defined as the real utilization rate for the i-th init and given by $\rho_i = \lambda_i \bar{s}_i$. We have

$$\bar{y}_1 = \bar{s}_1(1+\rho_2)/(1-\rho_1\rho_2), \quad \bar{y}_2 = \bar{s}_2(1+\rho_1)/(1-\rho_1\rho_2). \tag{3}$$

The equations (3) resemble those obtained in Sakata's work [1], when $\lambda_1 = \lambda_2$ and $\bar{s}_1 = \bar{s}_2$. That is, the mean time spent in the PS system, $\bar{y}$, has been given by $\bar{y} = \bar{s}/(1-\rho)$, where $\bar{s}$ is the mean of the service time distribution and $\rho = \lambda\bar{s}$, and λ is the mean arrival rate to the system. However, the results calculated from the rough estimation are found to be considerably different from those calculated from the simulation by GPSS, as shown in Fig. 4.

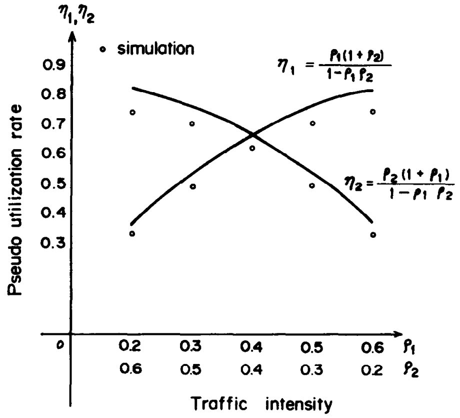

Fig. 4 Roughlycalculated pseudo utilization rate when $\rho_1+\rho_2=0.8$.

It is clear that an advanced analytic approach is needed to estimate the average (delayed) job running time. Hence, in this paper, a reasonable method will be developed to predict it with sufficient accuracy.

Although this approach cannot give explicit solutions for equilibrium queue length probabilities, once the average job running time is derived, both the average waiting time and the average queue length for a job class can be easily estimated from the usual waiting time law, namely $W = \bar{y}/(1-\eta)$, and Little's results, where $\bar{y}$ and η denote average running time and pseudo utilization rate of the class, respectively.

2. Multiple Separate System

First we consider a system with multiple queues, which are assigned to the individual inits. Then, we assume that the i-th init corresponds to the i-th queue ($1 \leq i \leq m$).

The following notation is used:

λ_i: average Poisson arrival rate to the i-th queue.
$\bar{s}_i$: average service requirement for the i-th init.

ρ_i: real utilization rate (=busy rate) for the i-th init and $\rho_i = \lambda_i \bar{s}_i$.

$\bar{y}_i(\rho_1, \rho_2, \cdots, \rho_m)$
expected job running time in the i-th init, including the "delays", for given $\rho_j (1 \leqq j \leqq m)$.

$\eta_i(\rho_1, \rho_2, \cdots, \rho_m)$
pseudo utilization rate of the i-th init and defined as $\eta_i = \lambda_i \bar{y}_i$.

Two-Queues—Two-Inits System

In viewing the present method, we take up the previous two-queues-two-inits system shown in Fig. 3. In this case, we show an approach to calculate the expected job running time $\bar{y}_i(\rho_1, \rho_2)$ for each i ($i = 1, 2$).

Let us first focus our attention on init 1. The method of approach is based on the fact that the job running time delays of init 1, at a certain time interval, may be significantly affected by the busy rate of init 2 and, in turn, init 2 is affected by that of init 1 at the same interval, and the same argument is applied successively. Hereupon, the busy rate for an init may be interpreted as utilization rate for it.

Let $\bar{u}_1$ denote the current average service requirement of jobs in init 1. For a small increase of the $\bar{u}_1$, $\Delta\bar{u}_1$, we want to evaluate a transition of the running time from $\bar{y}_1(\gamma_1, \rho_2)$ to $\bar{y}_1(\gamma_1 + \Delta\gamma_1, \rho_2)$, where $\gamma_1 = \lambda_1 \bar{u}_1$. Here, "current" means a certain time interval, including a tagged moment. Let us suppose that init 2 is always used with a uniform busy rate, ρ_2. From the assumption of the queuing model, the ρ_2 is independently given, whether or not init 1 is currently busy. We can apply the previous view of the equation given in (1) to the small time interval $\Delta\bar{u}_1$. This is because the time spent for a cycle of RR process for a job may double if another init is busy. Thus, we obtain the difference equation,

$$\Delta\bar{y}_1(\gamma_1, \rho_2) = \bar{y}_1(\gamma_1 + \Delta\gamma_1, \rho_2) - \bar{y}_1(\gamma_1, \rho_2) = \Delta\bar{u}_1(1 + Z_2). \quad (4)$$

Here, the value Z_2 is interpreted as the current pseudo busy rate in init 2. The value of Z_2 also increases by the rate of init 1, Z_1. Consequently,

$$Z_2 = \rho_2(1 + Z_1). \quad (5)$$

As for Z_1, because it can be supposed that the elapsed time in init 1 increases in proportion to the current increasing rate $\Delta\bar{y}_1(\gamma_1, \rho_2)/\Delta\bar{u}_1$, we have,

$$Z_1 = \lambda_1 \bar{u}_1 \Delta\bar{y}_1(\gamma_1, \rho_2)/\Delta\bar{u}_1. \quad (6)$$

To give an intuitive meaning for the last three equations, we would rather express them as the single equation:

$$\Delta\bar{y}_1(\gamma_1, \rho_2) = \Delta\bar{u}_1[1 + \rho_2\{1 + \lambda_1\bar{u}_1\Delta\bar{y}_1(\gamma_1, \rho_2)/\Delta\bar{u}_1\}]. \quad (7)$$

Solving (8) for $\Delta\bar{y}_1(\gamma_1, \rho_2)/\Delta\bar{u}_1$,

$$\frac{\Delta\bar{y}_1(\gamma_1, \rho_2)}{\Delta\bar{u}_1} = \frac{1 + \rho_2}{1 - \lambda_1\rho_2\bar{u}_1}. \quad (8)$$

By taking the limit $\Delta\bar{u}_1 \to 0$, we obtain the differential equation,

$$\frac{d\bar{y}_1(\gamma_1, \rho_2)}{d\bar{u}_1} = \frac{1 + \rho_2}{1 - \lambda_1\rho_2\bar{u}_1}. \quad (9)$$

Integrating over $\bar{u}_1$ from 0 to $\bar{s}_1$, we obtain the solution to yield an estimate for the expected job running time,

$$\bar{y}_1(\rho_1, \rho_2) = -\frac{1 + \rho_2}{\lambda_1\rho_2}\log(1 - \rho_1\rho_2). \quad (10)$$

Once the $\bar{y}_1(\rho_1, \rho_2)$ is calculated, the pseudo utilization rate is easily determined from the relation $\eta_i(\rho_1, \rho_2) = \lambda_i \bar{y}_i(\rho_1, \rho_2)$, by

$$\eta_1(\rho_1, \rho_2) = -\frac{1 + \rho_2}{\rho_2}\log(1 - \rho_1\rho_2). \quad (11)$$

Similarly, for init 2

$$\bar{y}_2(\rho_1, \rho_2) = -\frac{1 + \rho_1}{\lambda_2\rho_1}\log(1 - \rho_1\rho_2),$$

$$\eta_2(\rho_1, \rho_2) = -\frac{1 + \rho_1}{\rho_1}\log(1 - \rho_1\rho_2). \quad (12)$$

Multi-Queues—Multi-Inits System

The same approach is easily applied to the system with more inits. Four-queues-four-inits system is illustrated.

Herein, the job running time for an init, say init 1, should depend on the busy-situation of the other three inits. Then, for a small increase $\Delta\bar{u}_1$, increase of the associated job running time $\Delta\bar{y}_1(\gamma_1, \rho_2, \rho_3, \rho_4)$ for given ρ_2, ρ_3, ρ_4 can be written as

$$\Delta\bar{y}_1 = \Delta\bar{u}_1(1 + Z_2 + Z_3 + Z_4), \quad \text{where } \gamma_1 = \lambda_1\bar{u}_1,$$

$$Z_2 = \rho_2(1 + Z_1 + Z_3 + Z_4), \quad Z_3 = \rho_3(1 + Z_1 + Z_2 + Z_4),$$

$$Z_4 = \rho_4(1 + Z_1 + Z_2 + Z_3), \quad Z_1 = \lambda_1\bar{u}_1\Delta\bar{y}_1/\Delta\bar{u}_1, \quad (13)$$

henceforth, $\bar{y}_i(\cdots)$ is often abbreviated to $\bar{y}_i$. Solving the last equations for $\Delta\bar{y}_1/\Delta\bar{u}_1$, and using the same process as in the previous system, we can obtain the solution for init 1 as follows:

$$\bar{y}_1(\rho_1, \rho_2, \rho_3, \rho_4) = -\frac{1 + A_1}{\lambda_1 A_1}\log(1 - A_1\rho_1). \quad (14)$$

Here, A_1 is given by,

$$A_1 = \frac{(\rho_2 + \rho_3 + \rho_4) + 2(\rho_2\rho_3 + \rho_2\rho_4 + \rho_3\rho_4) + 3\rho_2\rho_3\rho_4}{1 - \{\rho_2\rho_3 + \rho_2\rho_4 + \rho_3\rho_4 + 2\rho_2\rho_3\rho_4\}}, \quad (15)$$

Thus, A_1 is interpreted as a summary factor to init 1's delay that is caused by coexisting jobs. The solutions for the other inits can be found in a way analogous with the last three expressions.

Furthermore, the present approach can be generalized into an m queues system, for an arbitrary m. The solution is written as,

$$\bar{y}_i(\rho_1, \rho_2, \cdots, \rho_m) = -\frac{1+A_i}{\lambda_i A_i} \log(1 - A_i\rho_i), \quad (16)$$

for each i $(1 \leqq i \leqq m)$.
Here, A_i is similarly interpreted and is given by,

$$A_i = \sum_{k=2}^{m-1} ka^{(k)} \Big/ \left\{1 - \sum_{k=2}^{m-1} (k-1)a^{(k)}\right\}. \quad (17)$$

And the $a^{(k)}$ $(1 \leqq k \leqq m-1)$ is given by,

$$a^{(k)} = \sum_{(h_1, h_2, \cdots, h_k), \neq i} \rho_{h_1}, \rho_{h_2}, \cdots, \rho_{h_k} \quad (18)$$

where for a certain i, summation $\sum_{(h_1, h_2, \cdots, h_k), \neq i}$ is taken over all the k-combinations formed from 1 to m, except for i.

Fig. 5 shows the pseudo utilization rates calculated for three-, four-, and six-inits systems.

We now show the cases where distinct rates are assumed for the individual queues.

Case 1: three-queues – three-inits system

$\rho_1 = 0.16$ $\rho_2 = 0.26$ $\rho_3 = 0.48$

$\eta_1 = 0.38$ $\eta_2 = 0.56$ $\eta_3 = 0.86$ (calculated by our method)

$\eta_1 = 0.37$ $\eta_2 = 0.56$ $\eta_3 = 0.83$ (simulation)

Case 2: six-queues – six-inits system

$\rho_1 = \rho_2 = \rho_3 = \rho_4 = 0.1$ $\rho_5 = \rho_6 = 0.2$

$\eta_1 = \eta_2 = \eta_3 = \eta_4 = 0.28$ $\eta_5 = \eta_6 = 0.48$ (calculated by our method)

$\eta_1 = \eta_2 = \eta_3 = \eta_4 = 0.27$ $\eta_5 = 0.47$ $\eta_6 = 0.46$ (simulation)

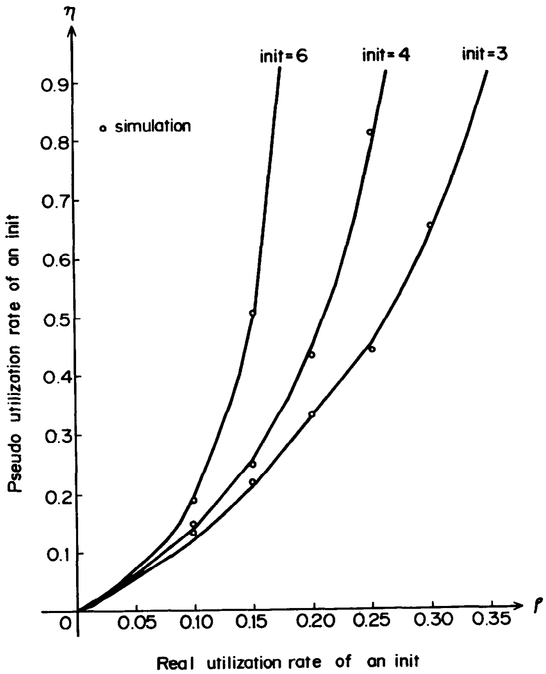

Fig. 5 Multiple separate model.

3. Common Init System

We now proceed to the second type of queuing system in which all inits are accompanied by only one queue. The following notation is used:

λ: average arrival rate to the system.
$\bar{s}$: average service requirement for an init.
ρ: real utilization rate for an init, and $\rho = \lambda\bar{s}/m$, if m-inits system is assumed.
$\bar{y}(\rho, m)$: expected utilization rate for an init when real utilization rate of an init is given by ρ.
$\eta(\rho, m)$: pseudo utilization rate for an init and defined as $\eta(\rho, m) = \lambda\bar{y}(\rho, m)/m$.

The two-inits system shown in Fig. 6 is considered first. The system seems to be an interesting variation of the basic two servers system in which service times are not mutually influenced. But in the present system, a single processor is assumed to be shared-used between the two inits, even though the inits themselves individually behave as the servers.

Let us assume again that init 1 is currently used. Then, the current busy rate of init 2 should be evaluated, because the busy probabilities of them are considered to be dependent, as may be identified with the conception of conditional probabilities in the basic $M/M/2$ system. We now formulate the queue length probabilities in the basic $M/M/m$ system [6]. Let p_k denote the statistical equilibrium probability when the system contains k jobs waiting or in execution. Then the p_k is given by:

$$p_k = \begin{cases} \dfrac{(m\rho)^k}{k!} P_0, & 1 \leqq k \leqq m-1 \\ \dfrac{m^m}{m!} \rho^k P_0, & k \leqq m \end{cases}$$

$$p_0 = \left\{\frac{(m\rho)^m}{m!(1-\rho)} + \sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!}\right\}^{-1}, \quad \rho = \lambda/m \quad (19)$$

Let us define $q_{(l), i_1, i_2, \cdots, i_k}$ as the conditional probability for an init-busy status (l), i_1, $i_2, \cdots$, i_k in the k-th multiprogramming level, namely

$$q_{(l), i_2, i_3, \cdots i_k} = \Pr\,[\text{inits } i_2, i_3, \cdots, i_k \text{ are busy} \mid \text{init } l \text{ is busy}]$$

Hereafter, we should pay attention to the particular case where $l=1$, because the conditional probability for the init 1, namely $q_{(1), i_2, i_2, \cdots, i_k}$, is definitely obtainable as a common value for all l $(1 \leqq l \leqq m)$. Furthermore, this probability is found equal to every such $(k-1)$-combination. Consequently, we need to calculate only $q_{(1),2,3,\cdots,k}$ for each k $(2 \leqq k \leqq m)$.

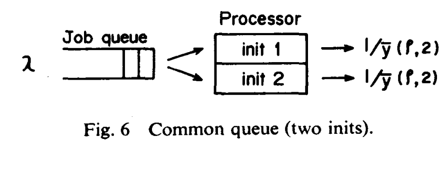

Fig. 6 Common queue (two inits).

$q_{(1),2,3,\cdots k} = \Pr$ [inits 1, 2, $\cdots$, k, are busy] /Pr [init 1 is busy]

From the results of (19),

$$q_{(1),2,3,\cdots k} = \begin{cases} \dfrac{b_k p_k}{\sum_{h=1}^{m-1} a_h p_h + \sum_{h=m}^{\infty} p_h}, & 2 \leq k \leq m-1 \\ \dfrac{\sum_{h=m}^{\infty} p_h}{\sum_{h=1}^{m-1} a_h p_h + \sum_{h=m}^{\infty} p_h}, & k=m \end{cases} \tag{20}$$

where a_h and b_k are given by

$$a_h = h/m, \quad b_k = 1/{}_mC_k,$$

the ${}_mC_k$ denotes the number of k-combinations formed from 1 to m.

Illustrated again is the four-inits system. Here we explain the derivation of its difference equation in correspondence with the previous four-separate-queues system.

At present, we want to evaluate the associated running time increase of an init, say init 1, $\Delta\bar{y}(\gamma, 4)$ for an increase of the service requirement $\Delta\bar{u}$, where $\lambda_1 = \lambda/4$ and $\gamma = \lambda_1 \bar{u}$.

Now let us recall the previous equations (13) and rewrite them as a single expanded equation.

$$\begin{aligned} \Delta\bar{y} = \Delta\bar{u}[1 &+ \rho_2\{1 + Z_1 + \rho_3(1 + Z_1 + \rho_2(\cdots) + \rho_4(\cdots)) \\ &\qquad + \rho_4(1 + Z_1 + \rho_2(\cdots) + \rho_3(\cdots))\} \\ &+ \rho_3\{1 + Z_1 + \rho_2(1 + Z_1 + \rho_3(\cdots) + \rho_4(\cdots)) \\ &\qquad + \rho_4(1 + Z_1 + \rho_2(\cdots) + \rho_3(\cdots))\} \\ &+ \rho_4\{1 + Z_1 + \rho_2(1 + Z_1 + \rho_3(\cdots) + \rho_4(\cdots)) \\ &\qquad + \rho_3(1 + Z_1 + \rho_2(\cdots) + \rho_4(\cdots))\}] \end{aligned}$$

$$Z_1 = \lambda_1 \bar{u} \Delta\bar{y}/\Delta\bar{u}. \tag{21}$$

According to the structure of the equation (21), we would draw up a diagram to represent the paths by which every init-busy status is reached. Fig. 7 shows a busy status tree for the present purpose. Then, the equation (21) is reformulated in due consideration to the multi-programming levels, enumerating the relevant init busy occurrences along the corresponding path, from bottom to top. Thus we have,

$$\begin{aligned} \Delta\bar{y} = \Delta\bar{u}[1 &+ \{(q_{(1),2} + q_{(1),2,3} + q_{(1),2,4} + q_{(1),2,3,4})(1 + Z_1) \\ &+ (q_{(1),2,3} + q_{(1),2,3,4}) + (q_{(1),2,4} + q_{(1),2,4,3}) \\ &+ 2q_{(1),2,3,4}\} + \{\text{similarly for init 3}\} \\ &+ \{\text{similarly for init 4}\}]. \end{aligned} \tag{22}$$

As the conditional probability $q_{(1),i_2,i_2,\cdots,i_k}$ was known as an identical value for every such $(k-1)$-combination from the preceding discussion, the equation (22) is reduced to,

$$\begin{aligned} \Delta\bar{y} = \Delta\bar{u}\{1 &+ 3q_{(1),2}(1 + Z_1) + 6q_{(1),2,3}(2 + Z_1) \\ &+ 3q_{(1),2,3,4}(5 + Z_1)\}. \end{aligned} \tag{23}$$

Using the relation $Z_1 = \lambda_1 \bar{u} \Delta\bar{y}/\Delta\bar{u}$, and solving (23) for $\Delta\bar{y}/\Delta\bar{u}$, we have,

$$\frac{\Delta\bar{y}}{\Delta\bar{u}} = \frac{1+B}{1 - \lambda_1 A \bar{u}}, \tag{24}$$

where: $A = 3q_{(1),2} + 6q_{(1),2,3} + 3q_{(1),2,3,4}$

$B = 3q_{(1),2} + 12q_{(1),2,3} + 15q_{(1),2,3,4}$

In a similar manner, we now obtain

$$\bar{y}(\rho, 4) = -\frac{1+B}{\lambda_1 A} \log(1 - A\rho), \quad \text{with } \lambda_1 = \lambda/4 \tag{25}$$

The method used can be readily generalized into the system with an arbitrary number of inits. The difference equations for an m-inits system are given by:

$$\Delta\bar{y} = \Delta\bar{u}\left(1 + \sum_{k=2}^{m} (k-1)\,{}_{m-1}C_{k-1}\, Z_{2,3,\cdots,k}\right), \tag{26}$$

$$Z_1 = \lambda_1 \bar{u} \Delta\bar{y}/\Delta\bar{u}, \quad \text{with } \lambda_1 = \lambda/m,$$

for each k $(2 \leq k \leq m)$,

$$Z_{2,3,\cdots,k} = q_{(1),2,3,\cdots,k}(d_{k-1} + Z_1).$$

The quantity, d_k, is introduced to denote the busy degree for the $(k+1)$-th level busy status. Constructing the busy status tree to a higher level in analogy with the preceding system, we can compute the d_k from the recurrence functions:

$$\begin{aligned} d_1 &= 1, \\ d_k &= 1 + (k-1)d_{k-1}. \end{aligned} \tag{27}$$

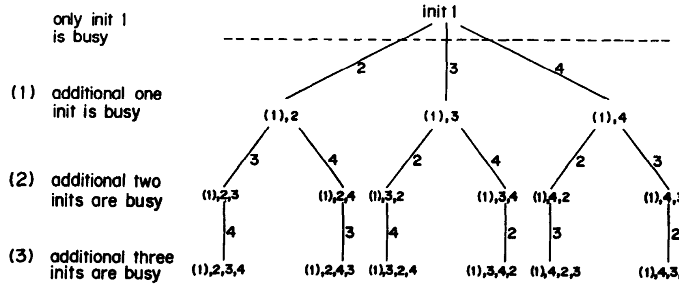

Fig. 7 Busy status tree.

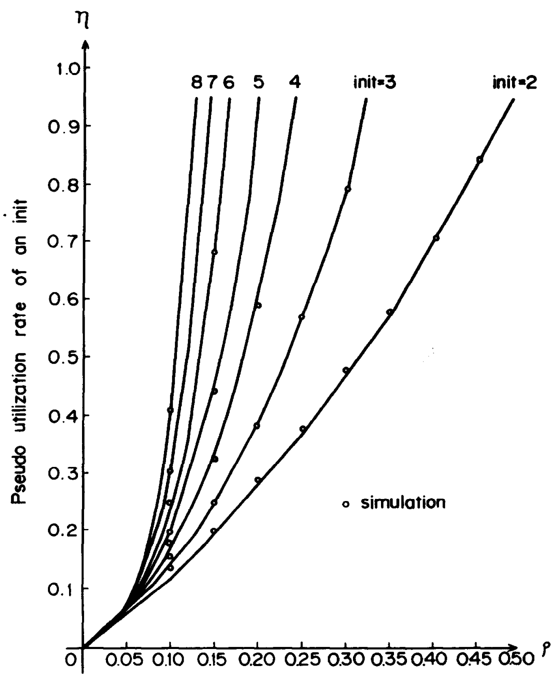

Fig. 8 Common init model.

From (26), the following solution is obtained:

$$\bar{y}(\rho, m) = -\frac{1+B}{\lambda_1 A}\log(1-A\rho), \tag{28}$$

where:

$$A = \sum_{k=2}^{m} (k-1)\,{}_{m-1}C_{k-1}\, q_{(1),2,3,\cdots,k},$$

$$B = \sum_{k=2}^{m} (k-1)d_{k-1}\ {}_{m-1}C_{k-1}\, q_{(1),2,3,\cdots,k}. \tag{29}$$

The results shown in Fig. 8 are calculated to validate the present solutions for systems with from two to eight inits.

4. Job Class System

Now, we recall the queuing system in Fig. 1 where the previous two types of systems are collectively modeled. Here, we assume that there is a separate job class associated with each average value $(\lambda, \bar{s})$. The following notation is used:

λ_i: average arrival rate to the i-th job class.

$\bar{s}_i$: average service requirement to an init of the i-th class.

m_i: number of inits of the i-th class, namely multiprogramming level of the i-th class.

ρ_i: real utilization rate for an init of the i-th class and $\rho_i = \lambda_i \bar{s}_i / m_i$.

$\bar{y}_i(\rho_1, m_1, \rho_2, m_2, \cdots, \rho_h, m_h)$: pseudo utilization rate of an init of the i-th class.

Two Queues—Five Inits System

Fig. 2 shows a two-job-class system, in which the first job class queue corresponds to the first three inits, and the second corresponds to the remaining two inits.

Once again, we shall concentrate on the first job class, especially on the behavior of init 1. Then, we estimate, $\Delta\bar{y}_1(\gamma_1, 3, \rho_2, 2)$, where $\lambda_{11} = \lambda_1/3$, $\gamma_1 = \lambda_{11} u_1$.

It can now be assumed that the init-busy probabilities are independently given among job classes, but they are interdependent within the class. The previous approaches used for the two distinct systems are melted to derive the difference equation; that is, the approach for the multiple separate system is applied to the relation among the job classes, while within the job class, the approach for the common init system is applicable. Then, we have:

$$\begin{aligned}
&\Delta\bar{y}_1 = \Delta\bar{u}_1\{1 + Z^{(1)} + Z^{(2)}\},\\
&Z^{(1)} = Z_2 + Z_3 + 2Z_{2,3}, \qquad Z^{(2)} = Z_4 + Z_5 + 2Z_{4,5},\\
&Z_2 = q_{(1),2}\{1 + Z_1 + Z^{(2)}\}, \qquad Z_3 = q_{(1),3}\{1 + Z_1 + Z^{(2)}\},\\
&Z_{2,3} = q_{(1),2,3}\{2 + Z_1 + Z^{(2)}\},\\
&Z_4 = q_4\{1 + Z_1 + Z^{(1)}\}, \qquad Z_5 = q_5\{1 + Z_1 + Z^{(1)}\}\\
&Z_{4,5} = q_{4,5}\{2 + Z_1 + Z^{(1)}\}\\
&Z_1 = \lambda_{11}\bar{u}_1 \Delta\bar{y}_1 / \Delta\bar{u}_1,
\end{aligned} \tag{30}$$

where, $q_{i_1, i_2, \cdots, i_k}$ denotes the init-busy probability given by

$$q_{i_1, i_2, \cdots, i_k} = \Pr\,[\text{inits } i_1, i_2, \cdots, i_k \text{ are busy}].$$

From (30), and applying similar processes, we have:

$$\bar{y}_1(\rho_1, 3, \rho_2, 2) = -\frac{1+B_1}{\lambda_{11}A_1}\log(1 - A_1\rho_1), \tag{31}$$

where $\lambda_{11} = \lambda_1/3$,

and A_1 and B_1 are given by:

$$\begin{aligned}
&A_1 = (e_1 + e_2 + 2e_1e_2)/(1 - e_1e_2),\\
&B_1 = \{e'_1(1+e_2) + e'_2(1+e_1)\}/(1 - e_1e_2),\\
&e_1 = q_{(1),2} + q_{(1),3} + 2q_{(1),2,3} = 2q_{(1),2} + 2q_{(1),2,3},\\
&e'_1 = q_{(1),2} + q_{(1),3} + 4q_{(1),2,3} = 2q_{(1),2} + 4q_{(1),2,3},\\
&e_2 = q_4 + q_5 + 2q_{4,5} = 2q_4 + 2q_{4,5},\\
&e'_2 = q_4 + q_5 + 4q_{4,5} = 2q_4 + 4q_{4,5},
\end{aligned} \tag{32}$$

With respect to the second job class queues, init 4 is chosen as a tagged init. Consequently, the equations (32) are slightly modified.

The results obtained by this analysis are validated against those of simulation, as shown in Fig. 9.

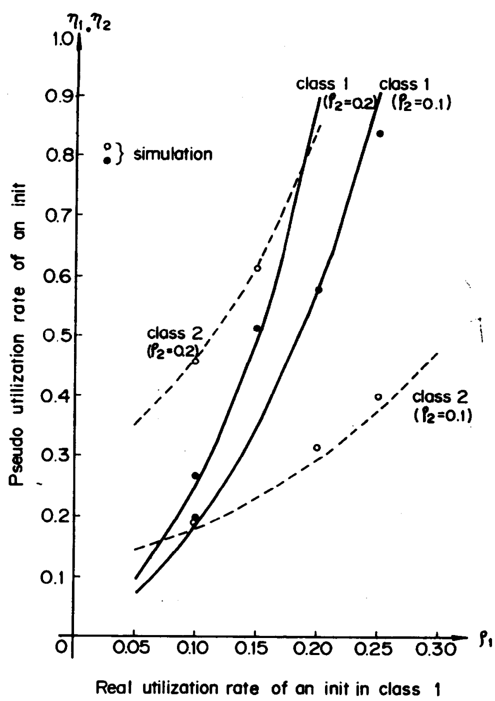

Fig. 9 Job class model.

n-Queues—m-Inits System

Finally, we reach a generalized job class system with n job classes, as shown in Fig. 1. The solution to the i-th job class is obtainable as:

$$\bar{y}_i(\rho_1, m_1, \rho_2, m_2, \cdots, \rho, m_n) = -\frac{1+B_i}{\lambda_{i1} A_i} \log(1 - A_i \rho_i),$$
$$\text{where } \lambda_{i1} = \lambda_i / m_i, \tag{33}$$

where A_i and B_i are analogously given by:

$$A_i = \sum_{k=1}^{n} k a^{(k)} \Bigg/ \left\{ 1 - \sum_{k=2}^{n} (k-1) a^{(k)} \right\},$$

$$B_i = \sum_{k=1}^{n} e'_k \left[1 + \sum_{g=1}^{n-1} \left\{ \sum_{(f_1, f_2, \cdots, f_g), \neq k} e_{f_1} e_{f_2} \cdots e_{f_g} \right\} \right] \Bigg/ \left\{ 1 - \sum_{k=2}^{n} (k-1) a^{(k)} \right\}, \tag{34}$$

Here, for each k $(1 \leqq k \leqq n)$, $a^{(k)}$ is given by

$$a^{(k)} = \sum_{(f_1, f_2, \cdots, f_k)} e_{f_1} e_{f_2} \cdots e_{f_k}, \tag{35}$$

where the notation $\sum_{(f_1, f_2, \cdots, f_g)}$ in (35) denotes the summation taken over all the g-combinations formed from 1 to n, while the notation $\sum_{(f_1, f_2, \cdots, f_g) \neq k}$ in (34) denotes the summation formed from 1 to n, except for k. In (35), for each f $(1 \leqq f \leqq n)$, e_i and e'_i are calculated as:

$$e_i = \begin{cases} \sum_{j=1}^{m_f - 1} j \; {}_{m_f - 1}C_j \; q_{(l_f), l_f + 1, l_f + 2, \cdots, l_f + j}, & f = i \\ \rho_f, & f \neq i \end{cases} \tag{36}$$

$$e'_i = \begin{cases} \sum_{j=1}^{m_f - 1} d_j j \; {}_{m_f - 1}C_j \; q_{(l_f), l_f + 1, l_f + 2, \cdots, l_f + j}, & f = i \\ \sum_{j=1}^{m_f} d_j j \; {}_{m_f}C_j \; q_{l_f, l_f + 1, \cdots, l_f + j - 1}, & f - i \end{cases} \tag{37}$$

Herein, the quantity d_j in (37) is the value recurrently calculated by function (27).

It is easily seen that the expressions obtained for this final queuing system are consistent with those for each of the previous two types of systems. That is, when we let $m = n$ and $m_i = 1$ for all i $(1 \leqq i \leqq n)$, the expressions (33)–(37) are reduced to (16)–(18) in the m-multiple-separate system. On the other hand, by letting $n = 1$ and $m = n$, the expressions are reduced to (28)–(29) for the m-common init system.

Acknowledgement

The author wishes to thank K. Isoda in Ibaraki University for his kind discussion.

References

1. Sakata, M., Noguchi, S. and Oizumi, J. An Analysis of the M/G/1 Queue under Round-Robin Scheduling, *Opns. Res.* 19 (1971) 371–385.
2. Coffman, E. G., Jr., Muntz, R. R. and Trotter, H. Waiting Time Distribution for Processor-Sharing Systems, *J. ACM* 17–1 (Jan. 1970) 123–130.
3. Takacs, L. Two Queues Attended by a Single Server, *Opns. Res.* 16 (1968) 639–650.
4. Taube-Netto, M. Two Queues in Tandem Attended by a Single Server, *Opns. Res.* 25 (1977) 140–147.
5. IBM Corp., OS/VS2 MVS System Programming Library: JES2, No. GC23-0001-0, IBM Corp. (1976).
6. Coffman, E. G. Jr. and Denning, P. J. Operating Systems Theory, Prentice-Hall (1973).
7. Bard, Y. An Analytic Model of the VM/370 System, *IBM J. Res. Develop.* V22-5 (1978) 498–508.

(Received January 22, 1979: revised January 21, 1980)