IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 12, Issue 1 Ver. III (Jan. - Feb. 2016), PP 41-44
www.iosrjournals.org

Network of Queues in Computing Systems with Memory Management and Non-exponential Service Times

G. Hemalatha[1], S. Navaneetha Krishnan[2] and C. Elango[3]
[1]*(Department of Mathematics, Devanga Arts College, Aruppukottai , India)*
[2]*(Department of Mathematics, VOC College of Arts and Science, Tuticorin, India)*
[3]*(Department of Mathematics, Cardamom Planters' Association College, Bodinayakanur, India)*

Abstract : *In this article we study a more general queuing network model of computing systems with non - exponential service times product form solution is obtained for this system and numerical examples are provided to illustrate the more general network of servers in system computer with optimal memory management.*
Keywords: *Network of queues, memory management, non-exponential service, product form solutions.*

I. Introduction

The network of computers with general service time is a real situation in communication systems as well as computing systems. The question before us is the product form solution for such systems. Differtiable service - time distribution at each node is a network in a variable assumption to get product for solution (see Chandy, K.M.et.al [6,7]. In this model first we considered a simple cyclic queuing system, (Trivedi, K.S. [15]) with non-exponential service time distribution, (hyper-exponential). The service discipline is assumed to be processor sharing (PS). This kind of service discipline is the limiting case of the quantum oriented. RR discipline, where the quantum size is allowed to approach to zero. The (0, S) policy is used for replishment of S memory slots, whenever number of memory slots becomes zero in CPU node. Rest of the paper is 1 organized as follows. Section 1 gives a brief introduction of the computing network of CPUs and I/O[s] with (0, S) policy based memory management. In section 2, problem formulation is done and in section 3 analysis of the model is described. Section 4 deals with numerical examples to illustrate the model.

The introduction of the paper should explain the nature of the problem, previous work, purpose, and the contribution of the paper. The contents of each section may be provided to understand easily about the paper. (1

II. Model Formulation

Consider a cyclic queuing model of the computing system as shown in Fig 1.

This is usual a closed queuing network with $p_0 = 0$. We assume that I/O service times are exponentially distributed with parameter μ_2 but the CPU service times are non-exponential, that is they are hyper exponentially distributed with two phase ($\alpha_{i,}$ i = 0; 1). The pdf of CPU service distribution is $f(t) = \alpha_0\mu_0 e^{-\mu_0 t} + \alpha_1\mu_1 e^{-\mu_1 t}$. Let n be the total number of jobs cycling in the system and $X_t = k$ be the number of jobs in the CPU node at time t.

Then n – k jobs in the I/O node at time t.Clearly $\{X_t \ : t > o\}$, is a stochastic process with state space $E = \{1,2,3, \dots , n\}$, but it is not a Markovian process. The future behavior of the process depends not only on the arrival processes but also on the time it spends in the state.

This non-Markovian nature of the process keeps one away from getting a steady state probabilities of the system states. But this difficulty can be avoided by observing the phase in which the job is sitting at present (Fig .2).

4 job schedule for theCPU chooses phase i with probability _i. For simplicity we assume that there is only two jobs in the network and the CPU[s] scheduling discipline is process sharing (PS). That is the CPU is equally shared away all the jobs waiting before CPU. If there is k jobs in the queue each job process the CPU to be slowed by a factor k.

Suppose there are X_t net jobs (net jobs = total jobs - memory slots) in system at time t in the 0 phase and Y_t denote the number of jobs in phase 1 at time t. Then $\{(X_t;\ Y_t): t\ \geq 0\}$ is a stochastic process with state space $\{E = (i, j); (i + j \leq 2 - S, i \in E_1, j \in E_2)\}$.

Here system is in state (i, j), if i jobs are in phase 0 and j jobs in phase 1. Now the state space of the stochastic process $\{(X_t;\ Y_t): t\ \geq 0\}$ is $E = \{E_0\ X\ E_1\ \}$, where $E_i = \{-S, 1 - S, 2 - S\}$
$E = \{(-S, S), (-S, S + 1), (-S + 1, -S), (-S + 1, -S + 1), (-S, -S + 2), (-S + 2, -S)\}$,

where i + j $\le$ 2 (-S)S ≥ 0: The state transition graph is given in Fig.3

Notations:
α_i - Phase probability i = 0; 1:
μ_i - Phase service times i = 0; 1:
μ_2 - Service rate at I/O.
$E_i = \{o, 1, 2, \} i = 0, 2.$
$E = E_0 \; X \; E_1$

For the reconstructed stochastic process all inter-event times are exponentially distributed and hence the process is Markovian. After a job finishes I/O service it leads to one of the CPU phases with probabilities α_0 and α_1 . The system state is (-S + 1, -S + 1) means two jobs are in CPU queue, each of which act CPU of half the speed.
The steady state balance equation for the system is given below.

$$\mu_2 p(-S, -S) = \mu_0 p(-S + 1, -S) + \mu_1 p(-S, -S + 1),$$

$$(\mu_0 + \mu_2) p(-S + 1, -S) = \mu_0 p(-S + 2, -S) + \mu_2 \alpha_0 p(-S, -S) + \frac{\mu_1}{2} p(-S + 1, -S + 1),$$

$$(\mu_1 + \mu_2) p(-S, -S + 1) = \mu_1 p(-S, -S + 2) + \mu_2 \alpha_1 p(-S, -S) + \frac{\mu_0}{2} p(-S + 1, -S + 1),$$

$$\mu_0 p(-S + 2, -S) = \mu_2 \alpha_0 p(-S + 1, -S),$$

$$\mu_1 p(-S + 2, -S) = \mu_2 \alpha_0 p(-S, -S + 1),$$

$$(\frac{\mu_1 + \mu_2}{2}) p(-S + 1, -S + 1) = \mu_2 \alpha_0 p(-S, -S + 1) + \mu_2 \alpha_1 p(-S + 1, -S).$$

Above equations may be solved recursively to get the solution in terms of $p(-S, -S)$ as follows:

$$p(-S + 1, -S) = \frac{\mu_2 \alpha_0}{\mu_0} p(-S, -S),$$

$$p(-S, -S + 1) = \frac{\mu_2 \alpha_1}{\mu_1} p(-S, -S),$$

$$p(-S, -S + 2) = (\frac{\mu_2 \alpha_1}{\mu_1})^2 p(-S, -S),$$

$$p(-S + 1, -S + 1) = \frac{2(\mu_2 \alpha_0)(\mu_2 \alpha_1)}{\mu_0 \mu_1} p(-S, -S),$$

value $p(-S, -S)$ can be obtained from the normalizing condition

$\sum_{(i,j) \in E} p(i, j) = 1$. Suppose $\frac{1}{\mu} = \frac{\alpha_0}{\mu_0} + \frac{\alpha_1}{\mu_0}$ be the mean CPU service time per visit to the CPU, then the above equations are the special case of the birth- death recursion equations. Clearly the system has a product form solution:

$$p(n_0, n_1) = (1 - \rho_0)\rho_0^{n_0}, (1 - \rho_1)\rho_1^{n_1}, n_0, n_1 \ge 0, n_0 + n_1 \le -2(S - 2).$$

III. System Performance Measures

1. The utilization of CPU phase 1 is given by

$$U_0 = p(1, 0) + p(1, 1).$$

2. The utilization of CPU phase 2 is given by

$$U_1 = p(0, 1) + p(1, 1).$$

Average throughput at phase 1 of CPU $= E[T_1] = \mu_0 U_0$ and that of phase 2 is given by $(E[T_2] = \mu_1 U_1)$

IV. Numerical Example

Consider a computing network having 2 nodes, one CPU with two phases and one I/O node. Label them as 0, 1 and 2 respectively. Let μ_0, μ_1 and μ_2 denote exponential service rates at CPU and I/O and $\mu_0 = 2$, $\mu_1 = 3$, $\mu_2 = 2.5$, $\alpha_0 = 0.4$, $\alpha_1 = 0.6$, where α_0 and α_1 are branching probabilities and S the maximum inventory.

Let $a = p(-S, -S)$,

$p(-S+1, -S) = \frac{\mu_2 \alpha_0}{\mu_0} p(-S, -S)$

$= 0.5 \times \text{a}$ $p(-S, -S+1) = \frac{\mu_2 \alpha_1}{\mu_1} p(-S, -S) = 0.25 \times \text{a}$

$p(-S, -S+2) = (\frac{\mu_2 \alpha_1}{\mu_1})^2 p(-S, -S) = 0.25 \times \text{a}$

$p(-S+1, -S+1) = \frac{2(\mu_2 \alpha_0)(\mu_2 \alpha_1)}{\mu_0 \mu_1} p(-S, -S) = 0.25 \times a$

by the normalizing condition we get a = 0:3637.

The average CPU service time per $\frac{1}{\mu} = 0.2 + 0.2 = 0.4$. Since the number of jobs circulating the cycle is 2, the optimal value of S is 2. In general optimal value of s the maximum inventory level is $S^* = n$, the number of job.

V. Figures and Tables

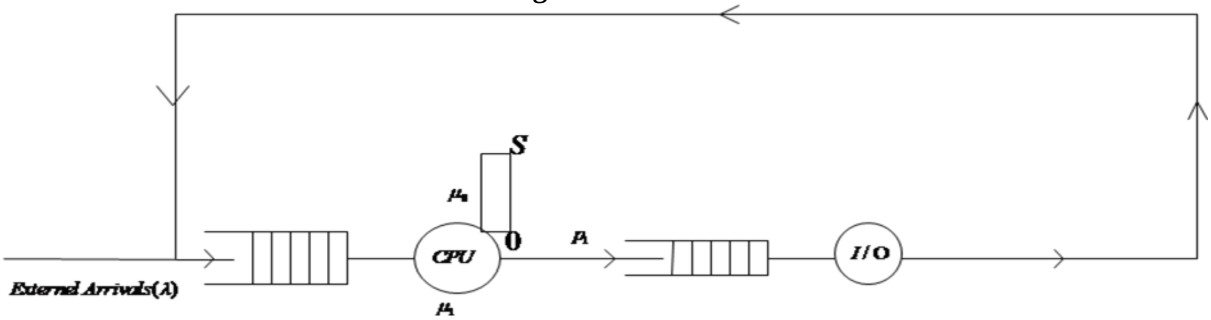

Fig. 1

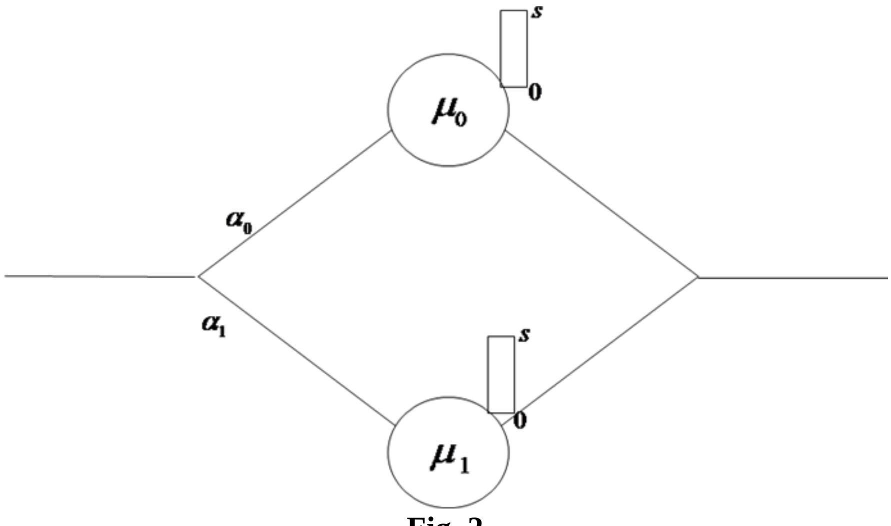

Fig. 2

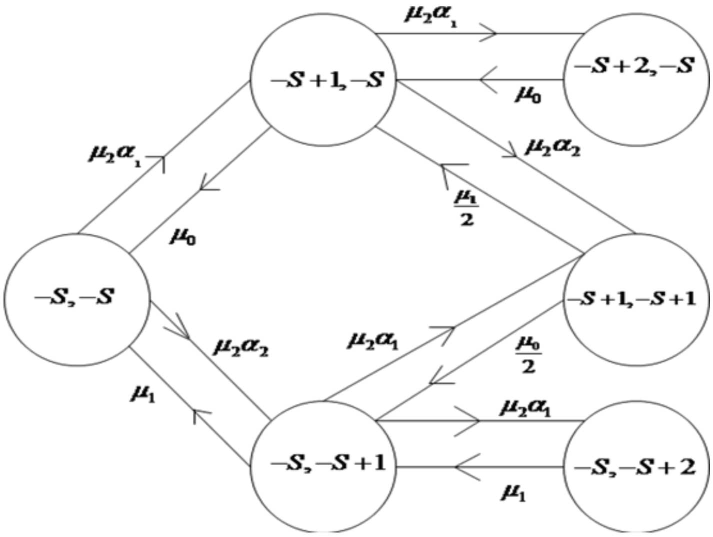

Fig. 3

VI. Conclusion

This model is a simple in which we have considered a cyclic network with phase type service at CPU this may be extended to all nodes including I/O and also to different class of customers.

References

[1]. Beutler, F.J., and Melamed, B., (1978)'"Decomposition of Customers streams of Feedback Network of Queues in Equillibrium'",Operartions Research, 26:6, pp. 1059-72.
[2]. Burke, P.J.,'"Output of a Queuing Systems'",Operartions Research, Volume 4, pp. 699-704.
[3]. Burke, P.J.,'"Proof of a Conjecative on the Inter arrival Time Distribution in a M/M/1 Queue with Feedback'",IEEE Trans on Communications, Volume 24, pp. 175-78.
[4]. Buzen, J.P.,(1978)'"A Queuing Network Model of MVS'",ACM Computing Surveys
[5]. Chandy, K.M.,Howard,J.P., and Towsley,D.F.,'"Product From and Local Balance in Queuing Networks,'"JACM, 24:2,pp250-263
[6]. Chandy, K.M.and Sauer,C.H.,'"Approximate Methods for Analyzing Queuing Network Models of Computer System,'"ACM Computing Surveys, 10,pp.281-317.
[7]. Ferrari , D.,'"Computer System Perfomance Evaluations Prentice Hall,'" Englewood Cliffs, N.J.
[8]. Giammo,T.,'"Validation of a Computer Perfomance Model of the Exponential Queuing Network Family,'"ACTA Informatica,7,pp.137-52.
[9]. Gordan, W and Newell,G.,'"Closed Queuing Systems with Exponential Services,'" Operations Research,.
[10]. Gross,D.,Shortle,J.F.,Thompson,J.M.,and Haris,C.M.,(2008)'"Fundamentals of Queuing Theory,'"John Willy and Sons,Inc.
[11]. Jackson,J.R.,'"Networks of Waiting Lines,'" Operations Researchs,5,pp.518-21.
[12]. Saner,C.H.,Chandy,K.M.,'"computer systems Performance Modeling,'"Prentice Hall, Englewood Cli_s,N.J.
[13]. Seveik, K.C., Graham,G.S., and Zahorjan,J.," 'Configuration and Capacity Planning in a Distributed Processing System,'"Proceeding Computer Performance Evaluation Users Group Meeting, Grlando,Fla.
[14]. Tripathi,S.K.,'"On Apporximate Solution Techniques for Queuing Network Models of Computer Systems,'"Computer System Research Group Technical Report, University of Toronto, Canada.
[15]. Trivedi,K.S.,and leech,R.L.,(1978)"`The Design and Analysis of Functionally Distributed Computer System,'"International Conference on Parallel Processing.
[16]. Williams,K.C.,and Bhandiwad,R.A.,"`A generating Function Approach to Queuing Network Analysis of Multiprogrammed Computers,'"Networks,pp.6-12.