[Volume 4 issue 1 January 2016]
PageNo.1236-1243 ISSN : 2320-7167

Multi-Class Central Server Queuing Network In Computer System With Memory Management

G. Hemalatha, S. Navaneetha Krishnan and C. Elango

Devanga Arts College, Aruppukottai,VOC College of Arts and Science, Tuticorin
Cardamom Planters' Association College, Bodinayakanur
E-mail: chelleelango@gnail.com

Abstract: In this article we consider a closed central server network in computer system. Assuming different class of customers access the network for service. The memory slots are managed as inventory to meet the requirement of the jobs. System performance measures are obtained and numerical examples are provided to illustrate the model implementation.

Introduction

Product from networks with multi-classes of jobs in computing system are difficult to analyze. Various types of jobs define the customer classes in the network that are gathered in chains. Convolution and MVA algorithms are two major tools used in the analysis of large queuing network models with multiple classes. The tree convolution, tree MVA algorithms for multi-chain networks are based on tree data structure to optimize algorithmic computations. Simple multi-class queuing network has been investigated in Baskett, Chandy, K.M. et-al [6] and Chandy, K. M. Howard, J.H. et. al [5]. In this article we considered a closed central server multi-class network of queues in computer system. The memory slots are managed at each node with instantaneous replenishment policy. It is assumed that there are r (> 0) classes of job want to share the network resources. We also admit different types and disciplines of service at nodes. Utilization of nodes for different class of jobs has been computed as system performance measures. Rest of the paper is organized as follows. Section 2 and 3 are model formulations part and analysis of the system section. Section 4 deals with system performance measures and the

final section 5 contains numerical examples to establish the results obtained.

2 Model Formulation

First we consider a open queuing network with different types of jobs. The arrival rate be assumed to be different for each type of jobs. The routing matrix R^k is also different for each type of job. Here $\lambda = \sum_k \lambda^k$, where λ^k denote the arrival rate of type k job. This model becomes a simple generalization of the Open Jackson Network. (six assumptions for OJN hold good). Service times at node i has exponential distribution for each type of jobs. The following figure - 1 represent a simple open queuing network with different class of jobs with 4 nodes.

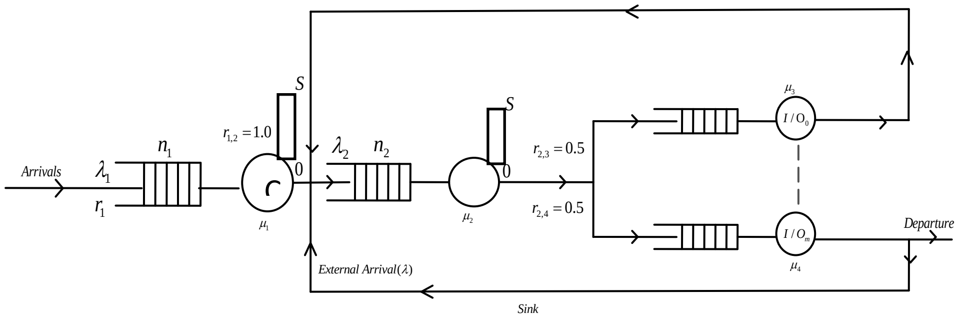

Figure 1

3 Analysis

Let X_j denote the random variable represent the net number of jobs in queue of node j (j = 0, 1, 2, … m_n). That is $X_j = Y_j - I_j$, Y_j be the number of jobs and I_j the inventory (memory slots) level at node j = 0, 1. Memory slots are maintained at node 0 and 1 (both CPVS). The state of random variables representing number of jobs in each queue at time t. Then the state of the entire open queuing network at time t is considered to a vector of real dimension m + 2. The state of the system at time t is given by s = (n_1, n_2, n_3, … n_m). The evolution of the state vector represents a continuous time Markov Chain. The joint probability - density function of all the queue length in the system is given by

$f(n_1, n_2, n_3, \dots n_m)$

$= \Pr \{X_1 = n_1, X_2 = n_2, \dots, X_m = n_m\}$

and the joint cumulative distribution function is of the form

$F(n_1, n_2, n_3, \dots, n_m)$

$= \Pr \{X_1 \le n_1, X_2 \le n_2, \dots, X_m \le n_m\}$.

By Jacksons result the product form solutions for the system is given by

$f(n_1, n_2, n_3, \dots, n_m) = f(n_1)\, f(n_2) \dots f(n_m)$.

Let $R^{(n)}$ denote the routing probability matrix for a job type k. Total arrival rate to the system is given by $\lambda = \sum_k \lambda^{(k)}$, where $\lambda^{(k)}$ is the arrival rate for the type k jobs. Suppose L_i denote the mean number (net) of jobs in nodes i = 0, 1, and number of jobs in nodes I = 2, 3, …, m using M/M/1 queuing system. Assuming that the all type of customers have the same average waiting time (each type has an exponential server time and FCFS service discipline). Little's formula can be used to compute the mean waiting time of each job. The average system size $L_i^{(k)}$ for type k job is obtained simply by weighting the node average total size by relative flow rate of type k job.

That is $L_i^{(k)} = \dfrac{\lambda_i^{(k)}}{\lambda_i^{(1)} + \lambda_i^{(2)} + \cdots + \lambda_i^{(n)}} L_i$, where L_i denote the system size at node *i*.

Example:

Consider a open queuing network having 3 nodes with the following parameters.

$\lambda = 35/\text{hr}$, $r_1^{(1)} = 19.25, r_1^{(2)} = 15.75$ and the routing probability matrices $R^{(1)} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0.2 \\ 0 & 0 & 0 \end{pmatrix}$ and $R^{(2)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & .01 & 0 \end{pmatrix}$ for the job types 1 and 2 respectively.

By solving the traffic equation,

$\lambda^{(k)} = R^{(k)} + \lambda^{(k)} R^{(k)}$ for k = 1, 2, we get the solutions

$$\lambda_1^{(1)} = 19.25, \lambda_2^{(1)} = 19.25, \lambda_3^{(1)} = 0.385,$$

$$\lambda_1^{(2)} = 15.75, \lambda_2^{(2)} = 0.1575, \lambda_3^{(2)} = 15.75$$

Total flow $\lambda = \sum_{k=1}^{3} \lambda^{(k)}$ gives

$\lambda = (35, 19.408\ 16.135)$.

By the M/M/1 queuing system result, we get $L_1 = 0.412$, $L_2 = 2.705$ and $L_3 = 6.777$. Total system size $L = 9.894$

Average system solution time = $\frac{L}{\lambda} = \frac{9.894}{35} = 0.28312$.

Average number of each logic of jobs k at each node is given by

$$L_i^{(k)} = \frac{\lambda_i^{(k)}}{\lambda_i^{(1)} + \lambda_i^{(2)} + \cdots + \lambda_i^{(n)}} L_i, i = 1, 2, 3; k = 1, 2$$

$$L_1^{(1)} = 0.227, L_2^{(1)} = 2.683, L_3^{(1)} = 0.612$$

$$L_1^{(2)} = 0.185, L_2^{(2)} = 0.022, L_3^{(2)} = 6.616.$$

4 Model Formulation - general

Consider a Closed multi-class queue network with n classes maintaining memory slots for service.

A class k (k = 1, 2, 3, …, n) customer has a transition probability matrix M_k and its service rate at node *i* is denoted by μ_{ik} (i = 0, 1, 2, …, m, k = 1, 2, …, n). For simplicity we assume 4 different types of services at node *i*.

1) A node *i* is said to be a type 1 node if it has a single server with exponentially distributed service times with FCFS scheduling and identical service rates for all job types $\mu_{ik} = \mu_i$.
2) A node *i* is said to be a type 3 node if it has a single server with PS (Processor Sharing) scheduling and service time distributions are differentiable and distinct.
3) A node *i* is said to be a type 3 node if it has an ample number of servers so that no queues ever forms at a node (self serving system). Differentiable service time distributions (distinct) are allowed.
4) A node is said to be a type 4 node if it has a single server with LCFS-PS scheduling. Any differentiable service time distribution is allowed and each job type may have a distinct service-time distribution.

Let q_{ik} be the number of jobs of type k at node *i*. Assume that there are q_k jobs of type k in the network so that we have, $\sum_{i=0}^{m} q_{ik} = q_k$ for

k = 1, 2, 3, …, n. define a vector

$X_i = (q_{i1}, q_{i2}, \ldots, q_{in})$ so that $(X_0, X_1, X_2, \ldots, X_m)$ is the state of the system at time t.

Let $r_i = \sum_{k=1}^{n} q_{ik}$ be the total number of jobs at node *i*.

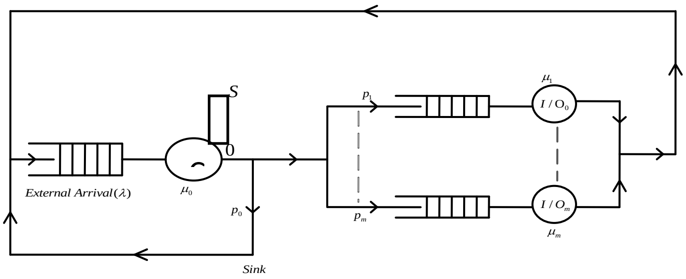

Figure 2

The joint probability in steady state for the system at state $(q_1, q_2, \ldots, q_n)$ is given by

$$P(X_0, X_1, X_2, \ldots, X_m) = \frac{1}{n(q_1, q_2, q_3, \ldots, q_n)} \prod_{i=0}^{m} g_i(X_i) \text{ where,}$$

$$g_i(X_i) = \begin{cases} r_i!\left(\prod_{k=1}^{n} \frac{1}{(q_{ik})!}(v_{ik})^{q_{ik}}\left(\frac{1}{\mu_i}\right)^{r_i}\right) & \text{for type 1 at node } i \\ r_i!\left(\prod_{k=1}^{n} \frac{1}{(q_{ik})!}\left(\frac{v_{ik}}{\mu_{ik}}\right)^{q_{ik}}\right) & \text{for type 2 or 4 at node } i \\ \prod_{k=1}^{n} \frac{1}{(q_{ik})!}\left(\frac{v_{ik}}{\mu_{ik}}\right)^{q_{ik}} & \text{for type 3 at node } i \end{cases}$$

If we define the relative utilization of the node *i* due to jobs of type k by $\rho_{ik} = \frac{v_{ik}}{\mu_{ik}}$, for *i* = 0, 1, 2, …, m (nodes) then the distribution function becomes,

$$g_i(X_i) = \begin{cases} r_i!\left(\prod_{k=1}^{n} \frac{1}{(q_{ik})!}\left(\frac{\rho_{ik}\mu_{ik}}{\mu_i^{r_i}}\right)^{q_{ik}}\right) & \text{for type 1 at node } i \\ r_i!\left(\prod_{k=1}^{n} \frac{1}{(q_{ik})!}(\rho_{ik})^{q_{ik}}\right) & \text{for type 2 or 4 at node } i \\ \prod_{k=1}^{n} \frac{1}{(q_{ik})!}(\rho_{ik})^{q_{ik}} & \text{for type 3 at node } i \end{cases}$$

5 System Performance Measures:

The real utilization for node *i* = 1, 2or 4 due to jobs of type k can be computed using the normalization constant C(.). Willams[16] has shown that utilization of i^{th} node is given by $u_i = \sum_{k=1}^{n} u_{ik}$,

$$u_{ik} = p_{ik} \frac{C(q_1, q_2, \cdots, q_{k-1}, q_k - 1, q_k, q_{k+1}, \cdots, q_n)}{C(q_1, q_2, \cdots, q_{k-1}, q_k, q_{k+1}, \cdots, q_n)}$$

For single job type (n = 1), the normalizer constant C(.) has the relation

$C(q_1, q_2, \cdots, q_n) = C_m(q_1, q_2, \cdots, q_n)$ where for i = 1, 2, 3, … m and

for j_k = 1, 2, 3, …, q_k, we have,

$$C_i(j_1, j_2, \cdots, j_n) = C_{i-1}(j_1, j_2, \cdots j_n) + \sum_{k=1, j_k \neq 0}^{n} C_i(j_1, j_2, \cdots, j_{k-1}, j_k - 1, j_{k+1}, \cdots, j_n),$$

with the initial condition

$$C_0(j_1, j_2, \cdots, j_n) = \frac{(j_1 + j_2 + \cdots + j_n)!}{j_1! j_2! \cdots j_n!} \left(\prod_{k=1}^{n} (\rho_{0k})^{j_k} \right)$$

and $C_i(0,0,\cdots,0)=1$. From this average throughput E[T_k] for type k job can be found for each job type.

6 Numerical Examples

Consider a central server closed queuing network with 2 different class of jobs labeled 1 and 2, circulating the network. Assume that total number of jobs in the network be n = 2.

Job 1 does not access I/O node 2 and job 2 doe not access I/O node 1. The mean service time of job 1 at CPU is $1/\mu_1$ and that of job 2 is $1/\mu_2$. The mean I/O service time of job is $1/\lambda_1$ and that of job on node 2 is $1/\lambda_2$. Assuming that there is no new program path, we get the probability that a job completing a CPU burst enters its respective I/O node is 1. Also assume that the CPU scheduling discipline is PS.

Let (n_0, n_1, n_2) denote the state of the system, where n_i denotes the number of jobs at node *i*. The transition diagram for the 3 dimensional Markov Chain is given in Fig 3.

The stochastic balance equation for the system we studied has been obtained as.

$(\lambda_1 + \lambda_2)\, p(0, 1, 1) = \mu_1\, p(1, 0, 1) + \mu_2\, p(1, 1, 0)$

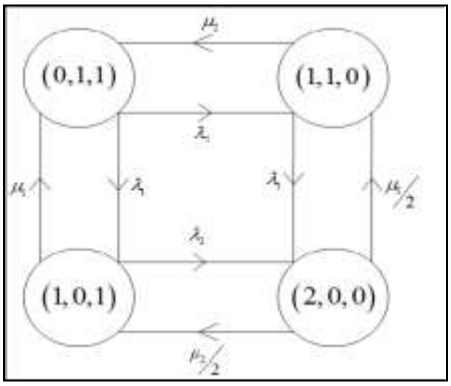

Figure 3

$$(\lambda_2 + \mu_1)\, p(1, 0, 1) = \lambda_1 p(0, 1, 1) + \frac{\mu_2}{2} p(2, 0, 0)$$

$$(\lambda_1 + \mu_2)\, p(1, 1, 0) = \lambda_2 p(0, 1, 1) + \frac{\mu_1}{2} p(2, 0, 0)$$

$$\left(\frac{\mu_1 + \mu_2}{2}\right) p(1, 1, 0) = \lambda_1 p(1, 1, 0) + \lambda_2\, p(1, 0, 1) \ldots.(1)$$

Solving the system of equations (1) we get

$$p(0,1,1) = \frac{1}{c}\frac{1}{\lambda_1\lambda_2},\ p(1,0,1) = \frac{1}{c}\frac{1}{\lambda_2\mu_1},$$

$$p(1,1,0) = \frac{1}{c}\frac{1}{\lambda_1\mu_2},\ p(2,0,0) = \frac{1}{c}\frac{2}{\mu_1\mu_2},$$

where the normalization constant C can be formed by using the probability condition. $\sum_{0 \le n_i \le 2} p(n_0, n_1, n_2) = 1,$

where, n_0, n_1, n_2 ∈ E. The value of normalizing constant C is given by $C = \frac{1}{\lambda_1 \lambda_2} + \frac{1}{\lambda_1 \mu_2} + \frac{1}{\lambda_2 \mu_1} + \frac{2}{\mu_1 \mu_2}$.

7 Performance Measures:

Utilization of I/O device 1 is given by:

$$U_1 = p(1,1,0) + p(0,1,1) = \frac{1}{C\lambda_2}\left[\frac{1}{\mu_2} + \frac{1}{\lambda_2}\right].$$

Utilization of I/O device 2 is given by:

$$U_1 = p(1,0,1) + p(0,1,1) = \frac{1}{C\lambda_2}\left[\frac{1}{\mu_1} + \frac{1}{\lambda_1}\right]$$

The average throughput of type *i* (= 1, 2) job is given by

$$E[\mathrm{T}_1] = U_1 \lambda_1 = \frac{1}{C}\left[\frac{1}{\mu_2} + \frac{1}{\lambda_2}\right]$$

$$E[\mathrm{T}_2] = U_2 \lambda_2 = \frac{1}{C}\left[\frac{1}{\mu_1} + \frac{1}{\lambda_1}\right].$$

Consider the following parameter values for the system we proposed:

$\lambda_1 = 2$, $\lambda_2 = 3$, $\mu_1 = 2$, $\mu_2 = 4$ and S = 10.

Steady state probabilities are computed as follows $C = \frac{1}{\lambda_1 \lambda_2} + \frac{1}{\lambda_1 \mu_2} + \frac{1}{\lambda_2 \mu_1} + \frac{2}{\mu_1 \mu_2} = 0.709$.

Here,

$$p(2,0,0) = \frac{1}{c}\frac{2}{\mu_1\mu_2} = 0.3526$$

$$p(0,1,1) = \frac{1}{c}\frac{1}{\lambda_1\lambda_2} = 0.1763$$

$$p(1,0,1) = \frac{1}{c}\frac{1}{\lambda_2\mu_1} = 0.2355$$

$$p(1,1,0) = \frac{1}{c}\frac{1}{\lambda_1\mu_2} = 0.2355,$$

Utilization of I/O device 1 is given by:

$$U_1 = p(1,1,0) + p(0,1,1) = 0.4118.$$

Utilization of I/O device 2 is given by:

$$U_1 = p(1,0,1) + p(0,1,1) = 0.4710.$$

The average throughput of type *i* (= 1, 2) job is given by

$$E[\mathrm{T}_1] = U_1 \lambda_1 = 0.8376$$

$$E[\mathrm{T}_2] = U_2 \lambda_2 = 1.413.$$

Example 2:

Consider a computer system with central processor having 3 types of jobs. Type 1 job need 1 second of CPU time and 10 seconds of I/O time and 1 unit of memory in each device. Jobs of type 2 are balanced. They need 10 seconds each of CPU and I/O and 2 units of memory at each device. Jobs of type 3 are CPU bound; they consume 100 seconds of CPU and 10 seconds of I/O time and 5 units of memory slots at each device.
Total memory unit available S=10 units and (0, S) policy is adapted to replenish the inventory instantaneously. One can admit either (one job of type 1, two jobs of type 2 and one job of type 3) or (three jobs of type 1, one job 11 of type 2 and one job type 3) for getting processed.
We can analyze the effect of two different choice of job combinations. The given data set is

Let $\rho_{ij} = E[R_{ij}]$. $\rho_{01} = E[\rho_{01}] = 1$, $\rho_{11} = \rho_{02} = \rho_{12} = \rho_{13} = 10$ and $\rho_{03} = 100$.

Case (i):

$n_1 = 1$, $n_2 = 2$, and $n_3 = 1$

Now the normalizing constants are

C_1 (1, 2, 1) = 1, 4, 10, 000; C_1(1, 2, 1) = 66, 000; C_1 (1, 1, 1) = 56, 400; C_1(1, 2, 0) =6, 600.

The mean throughputs of type *i* (= 1; 2; 3) jobs per second are computed as

$$E[T_1] = \frac{C_1(0,\ 2,\ 1)}{C_1(1,\ 2,\ 1)} = 0.04681$$

$$E[T_2] = \frac{C_1(1,\ 1,\ 1)}{C_1(1,\ 2,\ 1)} = 0.04$$

$$E[T_3] = \frac{C_1(1,\ 2,\ 0)}{C_1(1,\ 2,\ 1)} = 0.004681$$

Case (2):

$n_1 = 3$, $n_2 = 1$, $n_3 = 1$

The mean throughputs of type *i* (= 1, 2, 3) jobs per second are computed as

$$E[T_1] = \frac{C_1(2,\ 1,\ 1)}{C_1(3,\ 1,\ 1)} = 0.0776$$

$$E[T_2] = \frac{C_1(3,\ 0,\ 1)}{C_1(3,\ 1,\ 1)} = 0.0167$$

$$E[T_3] = \frac{C_1(3,\ 1,\ 0)}{C_1(3,\ 1,\ 1)} = 0.0056$$

7 Conclusion

In this article we studied central server queuing network & computer system with inventory (memory) management for complete service. The performance analysis is node to get its system utility. Cloud computing is the latest concept is which dynamic memory management is need of the hour to optimize the efficiency of cloud. The model we proposed can be studied in depth to make cloud computing models.

8 References

[1] Beutler, F.J., and Melamed, B., (1978) “Decomposition of Customers streams of Feedback Network of Queues in Equilibrium", Operations Research, 26:6, pp. 1059-72.

[2] Burke, P.J., "Output of a Queuing Systems", Operations Research, Volume 4, pp. 699-704.

[3] Burke, P.J., "Proof of a Conjecture 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, pp 250-263.

[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 Performance Evaluations Prentice - Hall," Englewood Cliffs, N.J.

[8] Giammo, T., "Validation of a Computer Performance 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 Research, 5, pp. 518-21.

[12] Saner, C.H., Chandy, K.M., "computer systems Performance Modeling, "Prentice Hall, Englewood Cliffs, 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 Approximate 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 Multi programmed Computers," Networks, pp.6-12.