Performability Modelling of Identical Multiprocessors System with remodelling and restarting delay

C. Chandra Sekhar Reddy[1], Dr. K. R. Prasad[2]

Abstract

Computer system models provide detailed answer to system performance. In designing, the models play an important role of helping to resolve architecture of remodelling and the system breakdown. In this paper, the performance modelling of homogeneous multiprocessor systems are covered. To account for delays due to remodelling and start up of the system, such systems are modelled and solved. Numerical results for various cases are presented.

Keywords : Infinite queue, QBD, mql, Markov process

1. Introduction

There are many computer, communication, and manufacturing systems which give rise to models where a single bounded/unbounded queue evolves in an environment which changes state from time to time. Multiprocessor system models at present are very important and widely used in modelling transaction processing systems, communication networks, mobile networks, and flexible machine shops with groups of machines. In this paper, the performance modelling of homogeneous multiprocessor systems are studied.

Multiprocessor system model have been extensively studied Trivedi [11]; Harrison and Patel [6]. Particularly, Stecke and Kim [8]; Stecke [9]; Righter [10]; Buzacott and Shantikumar [1]; Fiems et al. [4] contributed their work on nodes in communication networks, and flexible machine shops in a manufacturing environment. These systems use more than one processor and can be homogeneous (all processors are identical) or heterogeneous (at least one processor is different from others). Furthermore, such systems are prone to breakdowns. In this paper we study the approaches to model homogeneous with remodelling and delay of the system by suitably extending the resulting quasi birth death (QBD) process in the performance models of multiprocessor systems Chakka and Mitrani [2]; Chakka et al. [3]. Modelling homogeneous multiprocessor systems with rebooting and remodelling delays was considered and approximate performance models are presented based on Markov reward models Gemikonakli et al. [5].

An approximate performance modelling of this system was carried out in Trivedi et al. [12]. We intend to carry out the performance evaluation of this system by spectral expansion method considering leading eigen vector, as explained in Gemikonakli et al. [5].

2. Study Of The Multiprocessor System

The homogeneous multiprocessor system shown in Figure 1, consists of K identical parallel processors, numbered 1, 2,..., K, with a common queue, including the jobs in service. Jobs arrive at the system in a Poisson stream at rate , and join the queue. Jobs are homogeneous and the service rates of the processors assumed identical. Thus, the service times of jobs serviced by processor k (k=1, 2,..., K) are distributed exponentially with a mean value of $1/\mu$. The failure rate of processor is ξ and processor k

[1]Assistant Professor, RGM Engineering College
email : srisaimax@gmail.com
[2]Professor, BOS, SV University, Tirupati

executes jobs only during its operative periods which are distributed exponentially with mean $1/\xi$. At the end of an operative period, processor k breaks down and requires an exponentially distributed repair time with a mean value of $1/\eta$. The number of repairs that may proceed in parallel could be restricted. This is expressed by saying that there are R repairmen ($R \leq K$), each of whom can work on at most one repair at a time. Thus, an inoperative period of a processor would also include the possible waiting time for a repairman. No operative processor can be idle if there are jobs awaiting service and no repairman can be idle if there are broken-down processors waiting for repair. All inter-arrival, service, remodelling, restarting, operative and repair time random variables are independent of each other. The remodelling delay $1/\delta$ and the restarting delay $1/\varphi$ relate to the system and not to individual processors.

If the operative processors are more than jobs in the system, then the busy processors are selected randomly. Services that are interrupted by break-downs are eventually resumed, perhaps on a different processor but at a similar service rate. Similarly, if $R < K$ and the repair strategy allows pre-emption of repairs, then interrupted repairs are eventually resumed from the point of interruption and there are no switching delays.

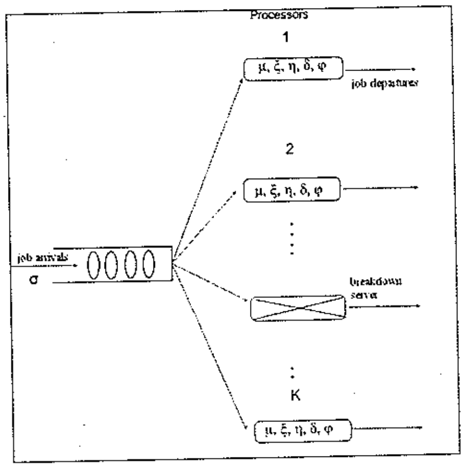

Figure 1 : Multiprocessor System with breakdowns and Repairs Remodelling and Rebooting Delays

3. Mathematical Descriptions Of QBD Processes

Consider a queuing system that can be modeled by a discrete time, two dimensional Markov process on semi-infinite or finite lattice strip. The process has a Markovian property and the state of system at observation time t can be described by two integer random variables I(t) and J(t). The former one is bounded and referred to as a phase; the latter one may be either unbounded (infinite case) or bounded (finite case) and is referred to as a level of the system. The Markov process is denoted by $Z = \{I(t), J(t); t \geq 0\}$ and its state space is $([0, 1,2, \ldots, N] \times [0, 1,2, \ldots])$ in the infinite case and $([0, 1,2, \ldots, N] x [0, 1,2, \ldots L])$ in finite case, respectively. If the possible jumps of system's level in transition are only 0, -1 or 1, the corresponding process is known as Quasi Birth-Death (QBD) process.

The system now can be represented by a QBD process with finite or infinite state space. The state of the system can be defined by [I(t), J(t)] where I(t) represent the operative states of the system and J(t) is the number of jobs in the system. Let the operative states be represented in the horizontal direction and the number of jobs in the vertical direction of a two-dimensional lattice strip. Here A is the matrix of instantaneous transition rates from operative state i to operative state k with zeros on the main diagonal, caused by change in the operative state i.e. a break down followed by remodelling or restarting and repair. These are the purely lateral transitions of the model Z. Matrices B and C are transition matrices for one-step upward transition rate matrix caused by job arrival rate and one-step downward transitions matrix caused by departure of a serviced job respectively. When the transition rate matrices depend on j for $j \geq M$, where M is a threshold having an integer value, the process Z evolves with the following instantaneous transitions for j = 0,1,2,....

- Aj : purely phase transitions - From state (i, j) to state (k, j) $(0 \leq i, k \leq N; i \neq k.)$
- Bj : one-step upward transitions - From state (i, j) to state (k, j + 1) $(0 \leq i, k \leq N)$
- Cj : one-step downward transitions - From state (i, j) to state (k, j - 1)$(0 \leq i, k \leq N)$.

4 Study Of The System

In multiprocessor systems, in practice however, some delay is encountered when a failed processor is being mapped out of the system, and when a repaired processor is being admitted into the system. A Markov model of the availability of a homogeneous multiprocessor system is developed and parameters such as probability of rejection, probability of interruption of an accepted task, and probability of late completion of an accepted task are also computed Trivedi et al. [12]. It is possible to model the system affected by such remodelling and restarting delays effectively using the spectral expansion method for performability measures. In this section such models are developed for multiprocessor systems with breakdowns, repairs, remodelling and restarting delays.

In the homogeneous multiprocessor system with K processors, it is assumed that there is a single repair facility with repair rate η. When a processor fails the fault is covered with probability c and is not covered with probability 1-c. Subsequent to a covered fault, the system comes up in a degraded mode after a brief remodelling delay, while after an uncovered fault a longer reboot action is required to bring the system up at a degraded mode. Here, degraded mode indicates a state with one less operative processor than the previous state. Remodelling and restarting times are exponentially distributed with mean $1/\delta$ and $1/\varphi$ respectively.

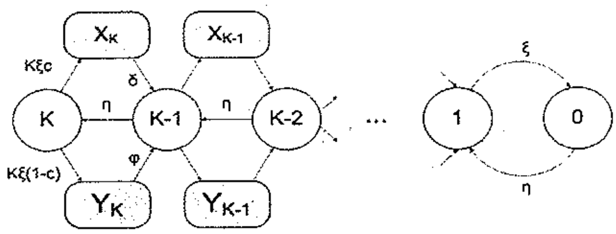

Figure 2 : Multiprocessor System with Breakdowns, Repairs, Restarting and Remodelling Delays

Figure 2 is the Markov chain that represents the operative states of the homogeneous multiprocessor system considered. The states labelled 1, 2, ..., K are the K working states of the multiprocessor system, with that many number of operative processors in each state. State 0 means no processor is operational. The K-1 states, labelled as $X_2, X_3, \ldots, X_K$, are the states representing the case where the system is out of service for mapping a failed processor out and a remodelling delay is needed to bring the system up at a degraded mode. The K-1 states labelled as $Y_2, Y_3, \ldots, Y_K$, are the states representing the case where the system is out of service for mapping a failed processor out and a rebooting delay is needed to bring the system up at a degraded mode. Hence, the total number of operative states is 3K-1. Let these be renumbered as, states 0,1,...,K unchanged, states $X_2, X_3, \ldots, X_K$ as K+1, K+2, ..., 2K-1, and the states $Y_2, Y_3, \ldots, Y_K$ as 2K, 2K+1, ..., 3K-2; hence, resulting in 3K-2 states for a K-processor system.

Clearly, the elements of the matrix A depend on the parameters K, ξ, η, c, δ, and φ. By following the model presented in Figure 3, the state transition matrices A, Aj, B, Bj, C, and Cj, can be given as follows.

$$A=\begin{bmatrix} h & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & h & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & h & \cdots & 0 & 0 & 2\alpha & 0 & \cdots & 0 & 2c & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 3\alpha & \cdots & 0 & 0 & 3c & \cdots & 0 \\ \vdots & & & & & & & & & & & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & h & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & K\alpha & 0 & 0 & \cdots & Kc \\ d & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & d & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & & & & & & & & \vdots \\ 0 & 0 & 0 & \cdots & d & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ j & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & j & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & & & & & & & & \vdots \\ 0 & 0 & 0 & \cdots & j & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

$B_j = B$, $j=0, 1, 2, \ldots$;

$B = \mathrm{Diag}\,[\sigma, \sigma, \ldots, \sigma\,]$ of size (3K-1)x(3K-1).

$C_j = C$ for $j \geq K$;

$C = \mathrm{Diag}\,[w(0)\mu, w(1)\mu, \ldots, w(3K-2)\mu]$;

C_0 = Null Matrix;

$C_j = \mathrm{Diag}[\mathrm{Min}\{w(0), j\}\mu, \mathrm{Min}\{w(1), j\}\mu, \ldots,$

$\mathrm{Min}\{w(3K-2), j\}\ \mu\ ]$ for $1 \leq j < K$

where, w(i) (i=0, 1, 2, …, 3K-2) is the number of working processors in the operative state i.

The matrix A is given for general K processor system. Since time-dependent failures are considered, the matrices Aj do not depend on j, and hence, Aj =A for all values of j.

5 The Steady State Solution

The solution for the system explained above is given for an unbounded queue (i.e. $K \leq L < \infty$) and the given solution is valid for study states of homogeneous systems. The study state probabilities of the system considered can be expressed as:

$$p_{i,j} = \lim_{t\to\infty} P\{I(t)=i, J(t)=j\};\quad 0 \leq i \leq N,\quad 0 \leq j \leq \infty \qquad (1)$$

It is convenient to define row vectors of probabilities corresponding to state with j jobs in the system:

$$v_j = (p_{0,j}, p_{1,j}, \ldots, p_{n,j})\ ;\quad j = 0,1,2,\ldots \qquad (2)$$

Then the balance equations for the equilibrium probabilities can be written as

$$V_j[D_j^A + D_j^B + D_j^C] = V_{j-1} B_{j-1} + V_j A_j + V_{j+1} C_{j+1},\quad 1 \leq j \leq M-1 \qquad (3)$$

Where Dj^A, Dj^B, Dj^C, are diagonal matrices whose i[th] element is the sum of i[th] row sum of the matrices $A_{j's}$, $B_{j's}$, $C_{j's}$ respectively of size (N+1)X(N+1). By definition $V_{-1}, = 0$ and $B_{-1} = 0$.

When j is greater than the threshold M, those equations become

$$V_j[D^A + D^B + D^C] = V_{j-1}B + V_jA + V_{j+1}C,\quad j = M+1, \ldots \qquad (4)$$

In addition, for j = 0, the balance equation is:

$$V_0[D_0^A + D_0^B] = V_0A_0 + V_1C_1 \qquad (5)$$

And all probabilities must sum up to 1:

$$\sum_{j=0}^{\infty} v_j e = 1.0 \qquad (6)$$

Where e is a column matrix whose elements are unit values of size N+1.

The balance equations with constant coefficients (4), are usually written in the form of a homogeneous vector difference equation of order 2:

$$V_j Q_0 + V_{j+1} Q_1 + V_{j+2} Q_{j+2} = 0,\ j = M, M+1, \ldots \qquad (7)$$

Where $Q_0 = B$, $Q_1 = A - D^A - D^B - D^C$ and $Q_2 = C$.

Furthermore, the characteristic matrix polynomial Q (λ) can be defined as:

$$Q(\lambda) = Q_0 + Q_1\lambda + Q_2\lambda^2 \qquad (8)$$

$$\text{Where } \psi Q(\lambda) = 0;\quad \det[Q(\lambda)] = 0 \qquad (9)$$

λ and ψ are N+1 eigenvalues and left-eigenvectors of $Q(\lambda)$ respectively with $|\lambda|<1$.

For unbounded queue system, and avoiding large numbers resulting from the positive powers of eigenvalues greater than 1.0, one can obtain the general solution as:

$$V_j = \sum_{t=0}^{N} \alpha_i \psi_i \lambda_i, \; j = M, M+1, ... \qquad (10)$$

where αi, i = 0,1,2,....N are constants may be complex.

Note that if there are non-real eigenvalues in the unit disk, then they appear in the complex conjugate pairs. The corresponding eigenvectors are also complex conjugate. The same must be true for the appropriate pairs of constants ak, in order that the right hand side of (10) be real.

From eq (3), the constants α_k and probability vectors v_j for j = 0, 1, 2,... M are to be determined. This is a set of (M+1)(N+1) linear equations with M(N+1) unknown probabilities and the N+1 constants α_k. Since the generator matrix is singular hence only (M+1)(N+1)-1 of these equations are linearly independent. To make generator matrix non singular it requires another equation and eq (6) do this job.

The quadratic eigenvalue-eigenvector problem (9), for computational purposes can be reduced to a linear form $\psi Q = \lambda \psi$, where Q is a matrix of size (2N+2)(2N+2). The process of evaluation, in detail discussed R Chakka in [2].

In what follows, Proposition 1 will be used to derive approximations, rather than exact solutions. The approximate solution is in detail discussed Isi Mitrani [7]. A central role in those developments is played by the largest eigenvalue, λN+1, and its corresponding left eigenvector. When the queue is stable, λN+1, is real, positive and simple. Moreover, it has a positive eigenvector. From this onward, λN+1, will be referred to as the dominant eigenvalue, and is denoted by ? and its corresponding vector as u_{N+1}.

Expression (10) implies that the tail of the joint distribution of the queue size and the environmental phase is approximately geometrically distributed, with parameter equal to the dominant eigenvalue, γ. To see that, divide both sides of (10) by γ^j and j$\to\infty$. Since γ is strictly greater than in modulus than all other eigenvalues, all terms in the summation vanish, except one:

$$\underset{j\to\infty}{Lt} \frac{v_j}{\gamma^j} = \alpha_{N+1} u_{N+1} \qquad (11)$$

when j is large, the above form can be expressed as:

$$v_j \approx \alpha_{N+1} u_{N+1} \gamma^j \qquad (12)$$

This product form implies that when the queue is large, its size is approximately independent of the environment phase. The tail of the marginal distribution of the queue size is approximately geometric:

$$p_{.,j} \approx \alpha_{N+1} (u_{N+1}.e) \gamma^j \qquad (13)$$

Where e is the column matrix defined earlier.

These results suggest seeking an approximation form:

$$v_j = \alpha u_{N+1} \gamma^j \qquad \alpha \text{ is a constant.} \qquad (14)$$

Note that γ and μN+1 can be computed without having to find all eigenvalues and eigenvectors. There are techniques for determining the eigenvalues that are near a given number. Here we are dealing with the eigenvalue that is nearest to but strictly less than 1.

One could decide to use the approximation (14) only for j = M. Then the coefficient a and the probability vectors vj for j = 0, 1, 2, . . . , M - 1 can be obtained from the balance equations (4), e.g., for $j \geq M$, and the normalizing equation (6). In that case, one would have to solve a set of M (N + 1) + 1 simultaneous linear equations with M (N + 1) + 1 unknowns. If that approach is adopted, then the approximate solution satisfies all balance equations of the Markov process except those for j = M.

Alternatively, and even more simply, (14) can be applied to all v_j, for j = 0, 1, 2,.... Then the approximation depends on just one unknown constant, α. Its value is determined by (6) alone, and the expressions for v_j become

$$v_j = \frac{u_{N+1}}{(u_{N+1}.e)}(1-\gamma)\gamma^j, j = 0,1,2,... \qquad (15)$$

and the mean queue length is calculated by

$$MQL = \frac{u_{N+1}}{(u_{N+1}.e)}\frac{\gamma}{(1-\gamma)}, \qquad (16)$$

This last approximation avoids completely the need to solve a set of linear equations. Hence, it also avoids all problems associated with ill-conditioned matrices. Moreover, it scales well. The complexity of computing γ and $\mu N+1$ grows roughly linearly with N when the matrices A, B, and C are sparse. The price paid for that convenience is that the balance equations for $j \leq M$ are no longer satisfied. The geometric approximation is asymptotically exact when the offered load increases i.e. the arrivals or services of the jobs are done rapidly so that system becomes heavily loaded and approaches saturation.

6. Numerical Results

To show the effectiveness of the method presented and evaluate the performance of the multiprocessor system with homogeneous processors, we considered, 4, 5, and 6-processor systems with break-downs with an infinite queue. Other parameters are given as ξ=0.01, η=0.05, μ= 1, φ= 2, and δ= 1 unless stated otherwise.

Figure 3 shows the relationship between the mean queue length and the mean arrival rate , for different number of servers and is observed that performances increases with proportion to number of servers. Here c is considered as zero. Figure 4 shows the mean queue length as a function of c. It is clearly evident that an increase in c results a decrease in the mean queue length because remodelling delays are shorter than restarting delays.

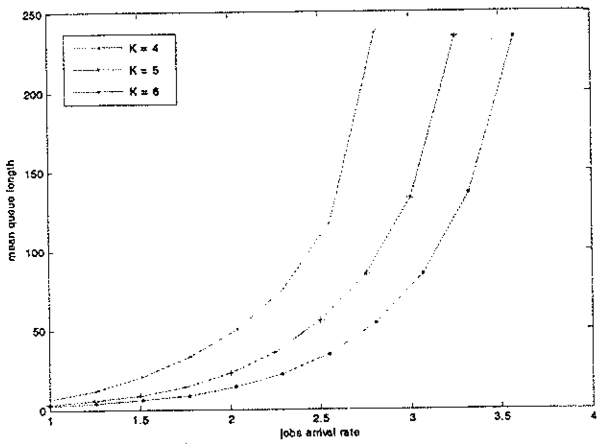

Figure 3: MQL Versus Mean Arrival Rate

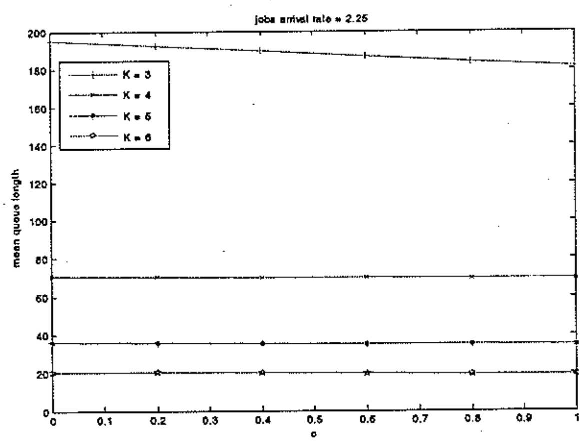

Figure 4 : MQL vs. c for different servers

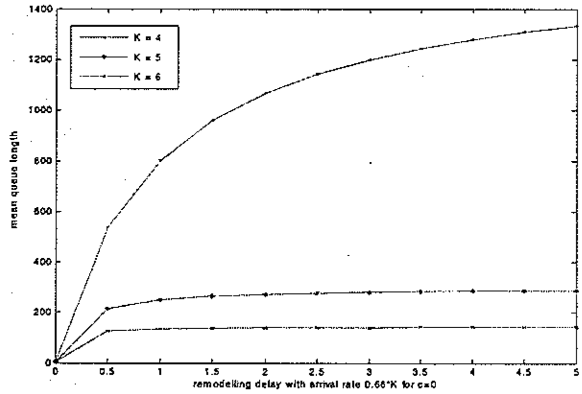

Figure 5 : MQL as a Function of δ with Job Arrival Rate 0.66K.

Figure 5 shows that number of jobs in the queue decreases as number of servers increases with arrival rate 0.66*K and queue length increases as remodelling delay increases. Here again c=0 is considered.

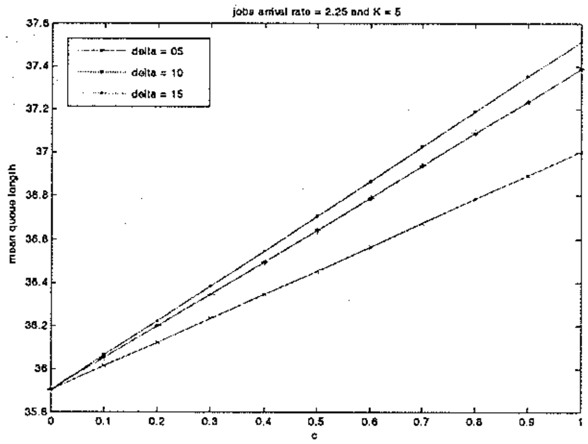

Figure 6 : MQL as a function of c and δ for homogeneous multiprocessor systems.

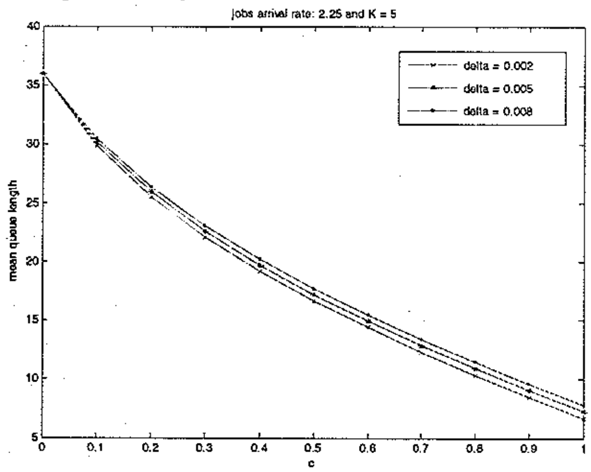

Figure 7 : MQL as a Function of c and δ For Homogeneous Multiprocessor System With 3 Nodes.

Figure 6 shows mean queue length as a function of c, with K = 5. This shows that as c increases mean queue length increases along with δ increases. Where as figure 7 shows that, for small δ values mean queue length decreases as c increases and increases as δ increases. Figure 8 shows mean queue length decreases for constant arrival rate of jobs with respect to number of server increases. It is observed that as c increases, number of jobs in the queue decreases.

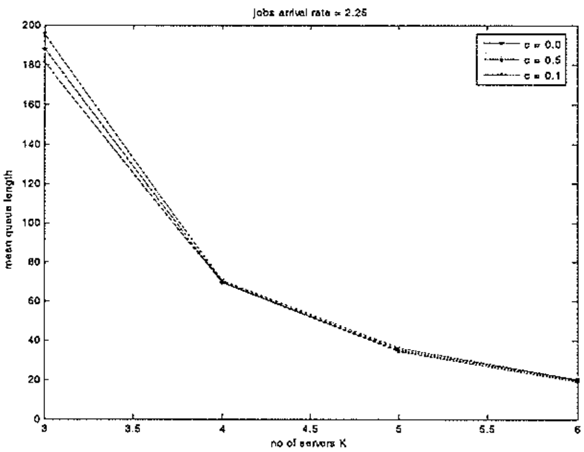

Figure 8 : MQL Vs. No. of Servers For Different c

7. Conclusions And Recommendations

In this paper multiprocessor system with break-downs, repairs and remodelling delays have been modelled for exact solution. Numerical results have been obtained and presented for various performability parameters, for unbounded as well as bounded queuing systems.

In conclusion, the discussion is out of scope for a processor to be added. Depending on various parameters, such as queuing capacity, processor speeds, and cover, an informed choice can be made. The approach to evaluating the performance of multiprocessor systems presented here lends itself as a most reliable tool in making such decisions.

The method can be extended to the case of heterogeneous multiprocessor system with on identical servers and many of high performance, highly reliable computer architectures.

References

[1] Buzacott. J. A and J.G. Shanthikumar, *"Stochastic Models of Manufacturing Systems"*, Prentice-Hall. 1993.

[2] Chakka R, *"Performance and reliability modelling of computing systems using spectral expansion"*, PhD thesis, University of Newcastle Upon Tyne, UK. 1995.

[3] Chakka. R, O. Gemikonakli and P. Basappa, *"Modelling Multiprocessor Systems with Time or Operation Dependent Breakdowns, Alternate Repair Strategies, Reconfiguration and Rebooting Delays"*, In Proceedings of SPECTS (San Diego, CA. PP 266-277. July 2002.

[4] Fiems D, B. Steyaert and H. Bruneel, *"Discrete-time queues with generally distributed service times and renewal type server interruptions"*, Performance Evaluation, 55, PP. 277-298, 2004.

[5] Gemikonakli O, Do, T. V, Chakka R, and Ever E, *"Numerical Solution to the Performability of a Multiprocessor System with Reconfiguration and Rebooting Delays"*, In Proceedings of ECMS, (Riga, Latvia, PP 766-773. June 2005.

[6] Harrison. P. G. and N.M. Patel, *"Performance Modelling of Communication Networks and Computer Architecture"*, Addison-Wesley. 1993.

[7] Isi Mitrani, *"Approximate solutions for heavily loaded Markov-modulated queues"*, Performance Evaluation, 62, PP 117-131, 2005.

[8] Stecke. K. E and I. Kim, *"Performance evaluation for systems of pooled machines of unequal sizes: unbalancing vs. blocking"*, European Journal of Operations Research 42, 22-38, 1989.

[9] Stecke. K.E, *"Machine interference: Assignment of machines to operators"*, Handbook of Industrial Engineering, Second Edition, Chapter 57, Gavriel Salvendy (Ed.), 1992.

[10] Righter. R, *"Optimal Policies for Scheduling Repairs and Allocating Heterogeneous Processors"*, Journal of Applied Probability, 33, PP. 536-547, 1996.

[11] Trivedi. K. S, *"Probability and Statistics with Reliability, Queuing, and Computer Science Applications"*, Wiley, NY, USA, 2002.

[12] Trivedi, K. S. A.S. Sathaye, O.C. Ibe and R.C. Howe, *"Should I add a processor?"*, In Proceedings of 23rd Annual Hawaii International Conference on System Sciences, IEEE Computer Society Press, (Los Alamos, CA), 214-221, 1990.

Authors Biography

Mr. Chandra Sekhar Reddy is pursuing Ph. D. in Mathematics at Sri Venkateswara University, Tirupati. He obtained M. Sc. Form SVU and has presented and published six technical papers in national, international conferences and journals. He has been working as Asst. Professor for past eight years in RGM engineering college, Nandyal. His current areas of research are modeling and performance evaluation of multiprocessor system, markov processes.

Dr. K. R. Prasad received his Ph. D. degree from IIT Khanpur. He is having more than three decades of experience in teaching at engineering colleges and P. G. college of S. V. University. Now he is working as senior professor at the Dept. of Mathematics and Board of Studies at S. V. University, Tirupati. He has guided and guiding several Ph. D. students.