IEEE Region 10 Conference on Computer and Communication Systems, September 1990, Hong Kong

Performance Modeling of Two Phase Service Policy in Distributed Systems

Rong-Chau Liu and Sheng–De Wang

Department of Electrical Engineering
National Taiwan University
Taipei 10764, Taiwan

Abstract

A Markovian queueing system is used to model a Two Phase Service (TPS) system. In this system, there are two queues, batch queue and individual queue, and a central server switching service between them. Expressions of mean task response time in the TPS are derived from the Markov–Chain concept and the Z–transform technique. Other performance parameters, such as the concept of power and speedup, and the optimization issues have also been taken into consideration. Furthermore, the performance improvement issue is investigated and the speedup upper bound is therefore obtained. Finally, numerical results are provided to verify the analytical model.

1. Introduction

In a distributed system, there are various applications, such as task allocation, scheduling, load sharing, and system reconfiguration, that have the characteristics of two phase execution on a central server. Tasks enter server which then probes the system for status information. This is the first phase. In the second phase, the server performs specific service, based on status information, for each task. We call the first phase a batch phase and the second one an individual phase.

One concrete example of two–phase service is load sharing using probing. Two important components of a load sharing policy are the transfer policy which generates status information and determines whether to process a task locally or remotely, and the location policy which allocates individual tasks to lightly loaded processors [1]. The batch service here is the transfer policy, and the individual service is the location policy.

Computer system analysts use some kind of models such as mathematical ones or simulation models to gain insight into the behavior of systems and to aid in system design. The analyst has several tools at his disposal to aid in estimating the performance of systems. Queueing models are becoming a more widely used method for analyzing computer systems, and have proved to be a powerful tool for performance analysis and prediction, such as for parallel processing systems [2], multiple bus multiprocessor systems [3–5], and distributed systems [6].

In this paper, we use a simple queueing system to model the TPS system. Markov–Chain used to describe the behavior of the TPS system is developed to obtain the expression of mean task response time and other performance parameters.

In section 2, we describe the structure of queueing system and the assumptions applied in this paper. In section 3, we analyze the characteristics of TPS system to obtain the Markov–Chain of the system state. The Z–transform technique is used to solve equilibrium equations and thus obtain the expression of mean task response time. Numerical results are provided in section 4. Section 5 concludes this paper.

2. The Queueing Model and Assumptions

The queueing model for the TPS system is shown in Fig. 1. The function of central server is to alternatively serve tasks in the batch queue and in the individual queue. In the batch service phase, the server first serves all tasks in the batch queue together at ones, and then switches to the individual queue. Tasks whose batch phase has been completed enter the individual queue, and then follow the FCFS (First Come First Serve) discipline to wait for service in the individual service phase.

There are assumptions described below for simplifying the problem so the resultant queueing system is mathematically tractable.

(a) Tasks enter the system according to a Poisson process with intensity λ.

(b) The batch service time and the individual service time are identically independent, exponentially distributed random variables with mean $1/\beta$, and $1/\mu$, respectively.

(c) The batch service time is independent of batch size, and whenever tasks enter the system when the server is in batch service phase, they will join the batch service immediately.

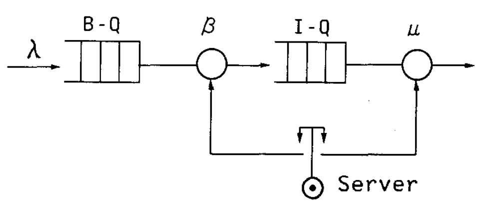

Fig. 1. TPS Queueing Model

3. Analysis

As described above, state of the TPS system can be modeled as a continuous time and discrete state Markov–Chain. The performance index is the mean task response time which is derived from the state equilibrium equations using Z–transform technique. We draw the Markov–Chain of the system in which the state (n_1,n_2) represents the situation where there are n_1 tasks in the batch queue and n_2 tasks in the individual queue, including tasks in the server. $P(n_1,n_2)$ denotes the probability that the system is in state (n_1,n_2).

3.1. General Case

In this section we consider a general case that the central server will switch to the individual queue with probability p, and will switch to batch queue with probability 1–p, when the number of tasks in the individual queue is nonzero. Markov–Chain of this case is shown in Fig. 2. The steady state equilibrium equations are as follows.

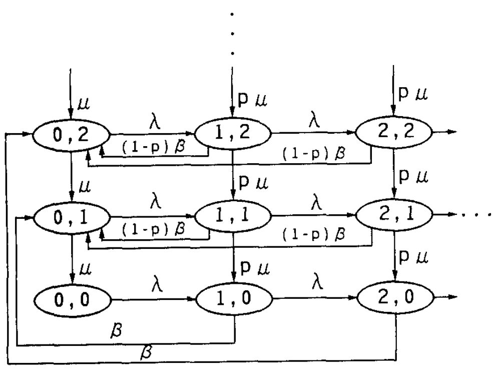

Fig. 2. Markov-Chain of TPS

$$\lambda P(0,0)=\mu P(0,1)\ , \tag{1}$$

$$(\lambda+\beta)P(i,0)=\lambda P(i-1,0)+p\mu P(i,1)\ ,\ i\geq 1\ , \tag{2}$$

$$(\lambda+\mu)P(0,j)=$$

$$\beta P(j,0)+\mu P(0,j+1)+(1-p)\beta\sum_{i=1}^{\infty}P(i,j)\ ,\ j\geq 1\ , \tag{3}$$

$$[\lambda+p\mu+(1-p)\beta]P(i,j)=$$
$$\lambda P(i-1,j)+p\mu P(i,j+1)\ ,i\geq 1,j\geq 1\ . \tag{4}$$

Define the following Z–transforms as

$$G(z,y)=\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}P(i,j)z^i y^j\ ,\ \Pi_j(z)=\sum_{i=0}^{\infty}P(i,j)z^i\ ,$$

$$\text{and } \Omega_i(z)=\sum_{j=0}^{\infty}P(i,j)z^j\ . \tag{5}$$

From above definitions, it is clear that

$$G(1,1)=\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}P(i,j)=1\ . \tag{6}$$

Let $\overline{N}$ be the mean number of tasks in the system, then

$$\overline{N}=\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}(i+j)P(i,j)=[\frac{d}{dz}G(z,z)]|_{z=1}. \tag{7}$$

By manipulating the above equations (1) to (4), we obtain the following result

$$G(z,y)=\frac{p(\beta y-\mu y+\mu)\Pi_0(z)-p\beta y\Pi_0(y)-F(y)}{p\mu+\lambda yz-\lambda y-p\mu y-(1-p)\beta y}\ ,$$

$$F(y)=y[\lambda+(1-p)\beta-p\lambda]\Omega_0(y)+p(1-p)\beta\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}P(i,j)y^{j+1}. \tag{8}$$

The detailed derivations of G(z,y) are given in the Appendix.

3.2. Exhaustive Service Case

To simplify the analysis, in this section, we consider the exhaustive service case. This means that, when the server finishes a batch service phase and then switches to the individual queue, it serves this queue exhaustively. In other words, the server serves all tasks in the individual queue before returning to the batch queue. Obviously, exhaustive service whose switching probability (p) equals to one is a special instance of the general cases. From (8), and let p=1, we have

$$G(z,y)=\frac{(\beta y-\mu y+\mu)\Pi_0(z)-\beta y\Pi_0(y)}{\mu+\lambda yz-\lambda y-\mu y}\ . \tag{9}$$

3.3. Performance Analysis

3.3.1. Mean Task Response Time

From (6), we have $lim_{z\to 1}G(z,1)=lim_{y\to 1}G(1,y)=1$. Using these relations and (9), after some algebra, we obtain

$$\Pi_0(1)=1-\frac{\lambda}{\mu}\ ,\ \text{and}\ \Pi_0'(1)=\frac{\lambda}{\beta}\ , \tag{10}$$

where the prime (') denotes the derivative with respect to its argument. From (7), (9), and (10), the mean number of tasks in the system

$$\overline{N}=[\frac{d}{dz}G(z,z)]|_{z=1}$$

$$=\mu\frac{\lambda\Pi_0(1)+(\mu-\lambda)\Pi_0'(1)}{(\mu-\lambda)^2}=\lambda\frac{\frac{1}{\beta}+\frac{1}{\mu}}{1-\frac{\lambda}{\mu}}\ . \tag{11}$$

The mean task response time can be obtained form the following two approaches.

(a) Little's formula [9],

$$\overline{T}=\overline{N}/\lambda=\frac{\frac{1}{\beta}+\frac{1}{\mu}}{1-\frac{\lambda}{\mu}}\ . \tag{12}$$

(b) Since Poisson arrivals see time averages, for a specified task, it's mean response time is composed of three parts, the mean batch service time $(\frac{1}{\beta})$, the mean waiting time in the individual queue $(\overline{N}\ \frac{1}{\mu})$, and the mean individual service time $(\frac{1}{\mu})$. Hence, using (11),

$$\overline{T}=\frac{1}{\beta}+\overline{N}(\frac{1}{\mu})+\frac{1}{\mu}=\frac{\frac{1}{\beta}+\frac{1}{\mu}}{1-\frac{\lambda}{\mu}}\ .$$

The mean task response time $\overline{T}$ in the above equation has similar expression to that of M/M/1 queueing system. The mean response time T_{mm1} of M/M/1 system was given by

$$T_{mm1}=\frac{\frac{1}{\mu}}{1-\frac{\lambda}{\mu}}\ [7]\ .$$

3.3.2. Optimization Issues

In this section, we apply the definition of power used in queueing system to TPS system. From the optimization point of view, we find the optimal system operating point (arrival rate of tasks) and the optimal number of tasks in the TPS system in order to maximize power. In [8], power P was defined as

$$P=\frac{U}{T}\ , \tag{13}$$

where U was the utilization of the server and T was the response time of the system. We use the symbol * to denote the optimal values maximize power. It was found that for any M/G/1 queueing system, power is optimized when $\overline{N}^*=1$. We also assume that the full–loaded condition of the computer system occurs when the system response time approaches infinity. It is easy to show that the optimal operating point of the M/M/1 queueing system is $\lambda^*=\frac{\mu}{2}$, i.e., 50% of the full load, since the full load occurs at $\lambda=\mu$.

For TPS systems, we define the utilization of the server is the ratio of task arrival rate to individual service rate, i.e., $U=\frac{\lambda}{\mu}$. From (12) and (13), we have

$$P=(\frac{\lambda}{\mu})(\frac{\mu-\lambda}{1+\frac{\mu}{\beta}})\ .$$

Optimizing P with respect to λ, we have $\frac{d}{d\lambda}P=0$, which leads to $\lambda^*=\frac{\mu}{2}$. We obtain the same results as in the M/M/1 queueing system. The optimal number of tasks in the system is, using (11),

$$\overline{N}^*=1+\frac{\mu}{\beta}\ . \tag{14}$$

This indicates that the optimal number of tasks in the TPS systems increases linearly with the mean batch service time, and that the slope is mean individual service rate.

3.3.3. Performance Improvement and Responsive Server

Tasks executed in the TPS systems can be regarded as fully parallel in the batch phase and fully serial in the individual phase. In other words, performance of the TPS systems is limited in the individual phase. Based on this observation, if we increase the number of servers in the individual phase, the individual service time will decrease, and consequently the system performance will be improved.

In this section, we let the speedup (SP) be the performance measure to describe how faster a task can be processed in an improved TPS system, as opposed to uniserver TPS system. We are interested in finding what the highest SP value could be. Intuitively, the more processors are used to serve the individual phase, the better performance improvement is obtained, if the communication overhead is ignored.

In the following, we consider and investigate, the ideal case, the effect of responsive (infinite) server [7], i.e., there is always a server available for each arriving task during the individual service phase, in the TPS system. Two theorems are provided to describe this phenomenon.

Theorem–1. In a TPS system with responsive server, the mean number of tasks in the individual queue ($\overline{N}_i$) and the mean number of tasks in the system during the batch service phase are $\frac{\lambda}{\mu}$, and $\frac{\lambda}{\beta}$, respectively.

<*Proof*> Following the same procedures mentioned above, we obtain, after some algebraic manipulations which we omit here, the Z–transform, denoted by $G_{rs}(z,y)$, of the TPS system with responsive server.

$$G_{rs}(z,y)=\frac{\beta\Pi_0(z)-\beta\Pi_0(y)-\mu(1-y)H(z,y)}{\lambda(z-1)},$$

$$\text{where } H(z,y)=\frac{d}{dy}G_{rs}(z,y)\,. \qquad (15)$$

(a) Let z=y and y approaches to one, we have

$$G_{rs}(1,1)=\frac{\mu}{\lambda}H(1,1)=1.$$

Thus, the mean number of tasks in the individual queue $\overline{N}_i$ is as follows.

$$\overline{N}_i=H(1,1)=\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}jP(i,j)=\frac{\lambda}{\mu}\,.$$

(b) Let y=1, we have

$$G_{rs}(z,1)=\frac{\beta\Pi_0(z)-\beta\Pi_0(1)}{\lambda(z-1)}.$$

Let z approaches to 1, after some algebra we obtain

$$\lim_{z\to1}G_{rs}(z,1)=1=\frac{\beta}{\lambda}\Pi_0'(1).$$

Thus the mean number of tasks in the system during the batch service phase is as follows.

$$\Pi_0'(1)=\sum_{i=0}^{\infty}iP(i,0)=\frac{\lambda}{\beta}\,. \qquad \text{Q.E.D.} \qquad (16)$$

Theorem–2. The performance improvement (speedup) upper bound of the TPS systems is $\frac{1}{1-\rho}$, where load factor (ρ) is defined as $\frac{\lambda}{\mu}$.

<*Proof*> As mentioned above, the minimum of mean number of tasks in the system can be obtained in a TPS system with responsive server. From (15) and engage in some algebraic manipulations, we have

$$\overline{N}_{rs}=\overline{N}^*=\sum_{n=0}^{\infty}[\Pi_n'(1)+n\Pi_n(1)]$$

$$=\Pi_0'(1)+\sum_{n=0}^{\infty}\sum_{i=0}^{\infty}nP(i,n)+\sum_{n=1}^{\infty}\Pi_n'(1)\,. \qquad (17)$$

By applying the results of Theorem–1, we obtain

$$\overline{N}^*=\frac{\lambda}{\beta}+\frac{\lambda}{\mu}+\sum_{n=1}^{\infty}\Pi_n'(1)\geq\frac{\lambda}{\beta}+\frac{\lambda}{\mu}\,.$$

$$\text{Thus, } \overline{T}^*=\overline{N}^*/\lambda\geq\frac{1}{\beta}+\frac{1}{\mu}\,. \qquad (18)$$

From (12) and (18), we obtain the upper bound of speedup

$$SP^*\leq\frac{(\frac{1}{\beta}+\frac{1}{\mu})/(1-\frac{\lambda}{\mu})}{\frac{1}{\beta}+\frac{1}{\mu}}=\frac{1}{1-\rho}. \quad \text{Q.E.D.} \quad (19)$$

From (19), it is clear that, the maximum of speedup obtainable from the TPS system with responsive server is small when the system is lightly loaded (small load factor). However, speedup will grow sharply when the system is heavily loaded, and it will approach infinity if the system is fully loaded (ρ=1).

4. Numerical Results

In this section, we present some simulation results to verify the analytical models mentioned above. In the following experiments, we fix the mean batch service time ($\frac{1}{\beta}$) and the mean individual service time ($\frac{1}{\mu}$) while varying the mean task interarrival time ($\frac{1}{\lambda}$) to obtain the mean task response time ($\overline{T}_{sim}$) by numerical simulation. The simulation results and the analytical results, computed from (12), of exhaustive service (uniserver) are listed in Table 1. The table also contains relative deviations which are computed by:

$$\delta(\%)=[(|\overline{T}_{sim}-\overline{T}|)/\overline{T}_{sim}]*100\,.$$

This table shows a good consistence between the analytical results and the simulation results.

In the proof of Theorem–2, we ignore the third part of mean number of tasks, i.e., those in the batch queue during the individual service phase, in the TPS system with responsive server. Table 2 shows the results. The approximation mentioned above leads to larger deviations than that in the table 1. However, they stay in an acceptable region (lower than 10%). Note that the deviation increases with the decreases of mean task interarrival time. This is due to the fact that mean number of tasks queued in the batch queue during the individual phase will increase under the heavy load condition.

Table 1.

Simulation and Analytical Results of Exhaustive Service with Uniserver, Mean Batch Service Time and Mean Individual Service Time are fixed at one.

$1/\lambda$	T_{sim}	T	$\delta(\%)$
10.0	2.208	2.222	0.634
7.5	2.292	2.308	0.698
5.0	2.485	2.5	0.604
2.5	3.3	3.33	0.909
1.5	5.974	6.0	0.435

Table 2.

Simulation and Analytical Results of Exhaustive Service with Responsive Server, Mean Batch Service Time and Mean Individual Service Time are fixed at one.

$1/\lambda$	T_{sim}	T	$\delta(\%)$
10.0	2.038	2.0	1.86
7.5	2.053	2.0	2.58
5.0	2.07	2.0	3.38
2.5	2.121	2.0	5.71
1.5	2.185	2.0	8.47

5. Conclusion

We have analyzed a Markovian queueing system and have used it to model a two–phase service system. Markov–Chain and Z–transform techniques are used to develop the expression of mean task response time in this system.

Power is defined as the ratio of the server utilization to the system response time. To maximize system power, we find out the optimal operating point (task arrival rate), and the optimal number of tasks in the system. The optimal operating point of the TPS system is 50% of the full load.

We also take the performance improvement issue into consideration. The system performance is improved when more servers are used to serve the individual phase. Speedup is defined as the ratio of the mean task response time of the uniserver system to that of the improved system. We investigate the effect of the responsive server in the TPS system, and find a phenomenon that, even that the communication overhead can be ignored, the maximum speedup obtainable from an infinite number of servers is $\frac{1}{1-\rho}$, ρ is the system load factor.

Finally, numerical results are provided to verify the analytical models. The results show a good consistence between the simulation approach and the analytical approach. Furthermore, the approximation applied in the derivation of performance improvement upper bound is acceptable.

References

[1] D. L. Eager, E. D. Lazowska, and J. Zahorjan, "Adaptive Load Sharing in Homogeneous Distributed Systems," IEEE Trans. Software Eng. vol. SE–12, May 1986, pp. 662–675.

[2] R. Nelson, D. Towsley, and A. N. Tantawi, "Performance Analysis of Parallel Processing Systems," IEEE Trans. Software Eng. vol. SE–14, Apr. 1988, pp. 532–540.

[3] M. A. Marsan and M. Gerla, "Markov Models for Multiple Bus Multiprocessor Systems," IEEE Trans. Comput. vol. C–31, Mar. 1982, pp. 239–248.

[4] K. B. Irani and I. H. Onyuksel, "A Closed–Form Solution for the Performance Analysis of Multiple–Bus Multiprocessor Systems," IEEE Trans. Comput. vol. C–33, Nov. 1984, pp. 1004–1012.

[5] T. N. Mudge and H. B. Al–Sadoun, "A Semi–Markov Model for the Performance of Multiple Bus Systems," IEEE Trans. Comput. vol. C–34, Oct. 1985, pp. 934–942.

[6] L. Kleinrock, "Distributed Systems", Computer, Nov. 1985, pp. 90–103.

[7] L. Kleinrock, *Queueing Systems, Vol. 1, Theory*, Wiley, New York, 1975.

[8] L. Kleinrock, "Power and Deterministic Rules for Probabilistic Problems in Computer Communications," in Proc. Int. Conf. Commun., 1979, pp. 43.1.1–43.1.10.

[9] J. D. C. Little, "A proof of Queueing Formula L=λW," Oper. Res. vol. 9, May 1961, pp. 383–387.

Appendix

This appendix describes the utilization of Z–transform technique used to solve state equilibrium equations and consequently obtain the expression of $G(z,y)$.

By multiplying both sides of (4) by z^i and summing up with respect to i, we have

$$\sum_{i=1}^{\infty}[\lambda+p\mu+(1-p)\beta]P(i,j)z^i$$
$$=\lambda\sum_{i=1}^{\infty}P(i-1,j)z^i+p\mu\sum_{i=1}^{\infty}P(i,j+1)z^i \text{ , and}$$
$$[\lambda+p\mu+(1-p)\beta][\Pi_j(z)-P(0,j)]$$
$$=\lambda z\Pi_j(z)+p\mu[\Pi_{j+1}(z)-P(0,j+1)] \text{ ,}$$

which leads to

$$p\mu\Pi_{j+1}(z)+[\lambda z-\lambda-p\mu-(1-p)\beta]\Pi_j(z)-$$
$$p(\lambda+\mu)P(0,j)+p\beta P(j,0)+p(1-p)\beta\sum_{i=1}^{\infty}P(i,j)+$$
$$[\lambda+p\mu+(1-p)\beta]P(0,j)=0 \text{ .} \tag{20}$$

By multiplying both sides of (20) by y^j, summing with respect to j, and applying (3) to this summing, after some algebraic manipulations, we obtain

$$\frac{p\mu}{y}[G(z,y)-y\Pi_1(z)-\Pi_0(z)]+$$
$$[\lambda z-\lambda-p\mu-(1-p)\beta][G(z,y)-\Pi_0(z)]+p\beta[\Pi_0(y)-P(0,0)]+$$
$$[\lambda-p\lambda+(1-p)\beta][\Omega_0(y)-P(0,0)]+$$
$$p(1-p)\beta\sum_{j=1}^{\infty}\sum_{i=1}^{\infty}P(i,j)y^j=0 \text{ .} \tag{21}$$

Follow the same procedures mentioned above, manipulation of (1) and (2) yields

$$p\mu\Pi_1(z)=(\lambda+\beta-\lambda z)\Pi_0(z)-\lambda(1-p)P(0,0) \text{ .} \tag{22}$$

Substituting (22) into (21) and engaging in some algebra, we finally obtain

$$G(z,y)=\frac{p(\beta y-\mu y+\mu)\Pi_0(z)-p\beta y\Pi_0(y)-F(y)}{p\mu+\lambda yz-\lambda y-p\mu y-(1-p)\beta y} \text{ ,}$$

$$F(y)=y[\lambda+(1-p)\beta-p\lambda]\Omega_0(y)+p(1-p)\beta\sum_{i=1}^{\infty}\sum_{j=1}^{\infty}P(i,j)y^{j+1}.$$