Prediction of Ready Queue Processing Time in Multiprocessor Environment Using Lottery Scheduling (ULS)

Diwakar Shukla[1] **Anjali Jain[2]** **Amita Choudhary[3]**

[1]Reader [2]Research Scholar [3]Lecturer

[1]Deptt. of Statistics and Mathematics, Dr.H.S.Gour Central University, Sagar- 470003, INDIA
[2]Deptt. of Computer Sc. and Applications, Dr.H.S.Gour Central University Sagar-470003, INDIA
[3]Deptt. of Computer Sc. and Applications, DAVV, Indore (M.P.), INDIA
E-mail Id:[1]diwakarshukla@rediffmail.com, [2]apscin30@yahoo.com, [3]amita411@rediffmail.com

Abstract

While in multi-user environment, CPU has to manage lot of requests generated over the same time. Waiting queue of processes generates a problem of scheduling for processors. Designers and hardware architects have suggested system of multiprocessors to overcome the queue length. Lottery scheduling is one such method where processes in waiting queue are selected through a chance manner. This opens a way to use probability models to get estimates of system parameters. This paper is an application where the processing time of jobs in ready queue is predicted using the sampling method under the k-processors environment (k>1).The random selection of one process by each of k processors through without replacement method is a sample data set which helps in the prediction of possible ready queue processing time. Some theorems are established and proved to get desired results in terms of confidence intervals.

Keywords: Scheduling, Lottery Scheduling, Bias, Variance, Confidence interval Lottery Scheduling (ULS), Estimator, Sampling.

I. INTRODUCTION

Scheduling is one of the important features of CPU where the multiple arrivals and requests are being finalized in optimum manner. Basic objective for this is to organize arrivals in specific way so as to achieve the goal of efficient performance algorithm of the system. Lottery scheduling is a procedure where, instead of purposive, the random selection of jobs is taken into account. This provides avenue to apply probability models to derive properties of system and to estimate various system parameters. This paper is an attempt to predict ready queue processing total time using lottery scheduling assuming k-processors environment. Carl et al. [1] discussed the proportional share resource management technique in lottery scheduling. David et al. [3] presented the specialization matching methodology in context to lottery scheduling. Shukla et al. [7] discussed a new variant of Lottery scheduling like SL Scheduling where the job selection is performed in random as well as in systematic manner both. The drawback with this is that it generates high variability in predicted estimates obtained and does not take into account the size measure of the process. A similar contribution is [8]. Shukla and Jain [5], [6] worked over multi-level queue scheduling with application of probability models in analysis. Sampling techniques and its wide applications are in [2] and [10]. Description of methodological part of scheduling is in [10], [11] and [12]. Raz et al. [4] presented procedure of deciding priorities among jobs by maintaining fairness in selection procedure.

2. MOTIVATION

Assuming a system has 5000 jobs in ready queue and after processing of 30 jobs, selected randomly the breakdown appears. System backup manager want to know how much additional time more required to process remaining jobs in the ready queue before shutting down the whole system. The backup management could be made accordingly. Deriving the idea from the above mentioned problem this motivates to explore the theoretical procedure to get probability based estimate or prediction for processing time using sampling methods with the help of lottery scheduling as a base scheme under k-processors environment.

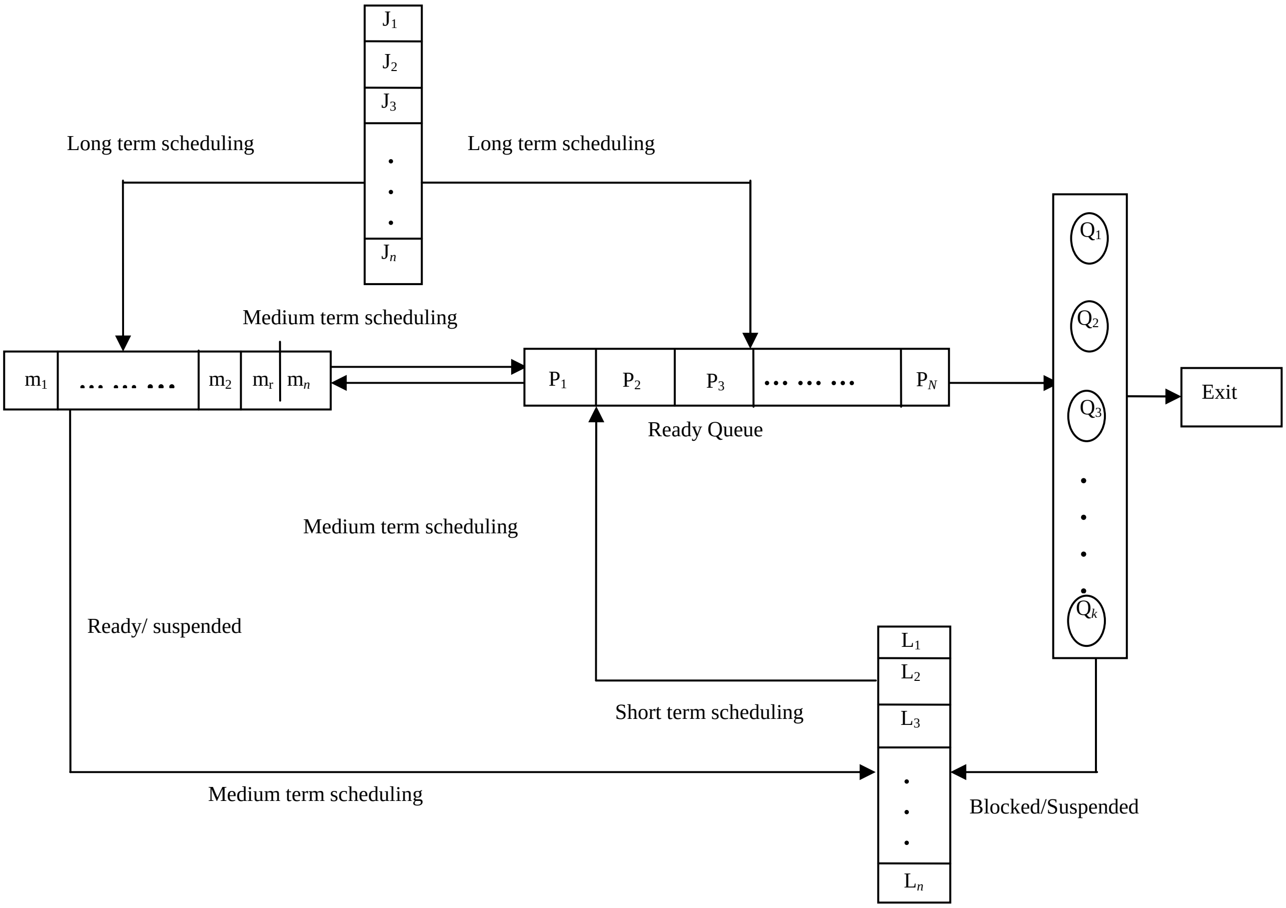

Fig. 1: Ready Queue Processing Under Lottery Scheduling

3. PROCESSOR STRUCTURE AND ROTATIONS

Let Q_1, Q_2, Q_3….Q_k be K processors who take intake from the ready queue containing P_1, P_2, P_3….P_N processes (K<N). Processes are related to long, medium and short term scheduling queues each having n processes, transferring to ready queue time. When a process remains suspended, they are back to respective queue. The Figure 1 shows the diagram of scheduling process structure with k processors.

(A) Multiprocessor Lottery Scheduling

Step 1: When a process enters into ready queue, it is allotted a random number (in specified range).

Step 2: Each processor Q_1, Q_2, Q_3….Q_k generates unique and uncommon random number in similar specified range stated in Step I.

Step 3: Matching of both random numbers takes place between process and processor. If both random numbers are same for a process in ready queue, it is assigned to that processor.

Step 4: Processor either blocks or processes the job. It selects another process by random manner.

Step 5: After when one job processed completely or partially processors generate time consumed in processing as t_i (time by j^{th} processor) (j=1, 2, 3…..K).

(B) Estimation of Ready Queue Processing Time

Suppose that P_1, P_2, P_3… P_k are k selected processes from ready queue of size N in random manner through lottery scheduling. These k selected processes P_j are assigned to processors Q_j (j =1, 2, 3...k). Processors have generated time consumed in processing as t_i. Now, suddenly system failed and there is need for backup management. We have some basic results discussed below used for prediction purpose.

Theorem 1: The probability of selecting a specified process for processing from N processes of ready queue at any given draw is equal to the probability of its being processed at first draw in k-processors setup by lottery scheduling with k processors.

Proof: The probability of drawing randomly a specified process for processing at first draw $\frac{1}{N}$.

Let P_r be the process selected at the r^{th} draw(r=1, 2, 3…..k).

$$P(P_r) = \prod_{i=1}^{r-1} P\{P_r \text{ is not processed at } i^{th} \text{ draw}\} \times \left\{ \begin{array}{l} P_r \text{ is processed at } r^{th} \text{ draw} \\ |\text{it is not processed at the previous } (r-1) \text{ draw} \end{array} \right\}$$

$$P(P_r) = \prod_{i=1}^{r-1}\left[1 - \frac{1}{N-(i-1)}\right]\left[\frac{1}{N-(r-1)}\right]$$

$$= \prod_{i=1}^{r-1}\left[\frac{N-i}{N-i+1}\right]\left[\frac{1}{N-r+1} = \frac{1}{N}\right]$$

Therefore $P(P_r) = \frac{1}{N}$

(C) Probability of selecting any Specified Process for processing

Since a specified process in ready queue of length N can be selected for the processing in k independent ways. It can be selected for processing at r^{th} draw (r = 1, 2, 3… k**)** and since $P(P_r) = \frac{1}{N}$; for r = 1, 2, 3… k.

By the addition theorem of probability, one can get probability a specified being processed by the processor $= \sum_{r=1}^{k} \frac{1}{N} = \left(\frac{k}{N} \right)$

Theorem 2: Let T_1, T_2, T_3 ....T_N be the actual processing time of N processes in ready queue with overall mean $\bar{T} = (N)^{-1} \sum_{i=1}^{N} T_i$ .The time consumed by processor to process K (k<N) jobs before the occurrence of sudden failure are ($t_1,t_2,t_3 \ldots t_k$) with mean $\bar{t} = (k)^{-1} \sum_{j-1}^{k} t_i$ .The Ti differ from ti since many may be partially finished or blocked while failure occurs

Proof: Define an indicator function a_i as

$$a_i = \begin{cases} 1 & \text{if } i \text{ th process of ready queue is selected} \\ & \text{for processing; (i = 1,2,3....N)} \\ 0 & \text{otherwise} \end{cases}$$

The expected value of a_i is $E(a_i) = \frac{k}{N}$

So $E(\bar{t}) = E\left[\frac{1}{k} \sum_{j=1}^{k} t_j \right]$

$$= \frac{1}{k} \left[\sum_{i=1}^{N} E(a_i T_i) \right]$$

$$= \frac{1}{k} \left[\sum_{i=1}^{N} E(a_i) T_i \right] = \frac{1}{k} \left[\sum_{i=1}^{N} \frac{k}{N} T_i \right] = \bar{\mathrm{T}}$$

So, sample mean time $\bar{t}$ is unbiased estimator of entire ready queue mean $\bar{T}$.

(a) The ready queue mean square is

$$S^2 = \frac{1}{N-1} \left[\sum_{i=1}^{N} (T_i - \bar{T})^2 \right] = \frac{1}{N-1} \left[\sum_{i=1}^{N} T_i^2 - N\bar{T}^2 \right]$$

(b) The sample based processor mean square is

$$s^2 = \frac{1}{k-1} \left[\sum_{j=1}^{k} (t_j - \bar{t})^2 \right] = \frac{1}{k-1} \left[\sum_{j=1}^{k} t_j^2 - k\bar{t}^2 \right]$$

Theorem 3: In lottery scheduling with k processors the process based time mean square is an unbiased estimator of ready queue mean square while the system failure occurs, i.e.

Proof:

$$s^2 = \frac{1}{k-1} \left[\sum_{j=1}^{k} t_j^2 - k\bar{t}^2 \right]$$

$$= \frac{1}{k-1} \left[\sum_{j=1}^{k} t_j^2 - \frac{1}{k} \left(\sum_{j=1}^{k} t_j \right)^2 \right]$$

$$= \frac{1}{k-1} \left[\sum_{j=1}^{k} t_j^2 - \frac{1}{k} \left(\sum_{j=1}^{k} t_j^2 + \sum_{j \neq i=1}^{k} t_j t_i \right)^2 \right]$$

$$= \frac{1}{k-1} \left[\sum_{j=1}^{k} t_j^2 \left(1 - \frac{1}{k} \right) + \frac{1}{k} \sum_{j \neq i=1}^{k} t_j t_i \right]$$

$$= \frac{1}{k-1} \left(1 - \frac{1}{k} \right) \left[\sum_{j=1}^{k} t_j^2 - \frac{1}{k(k-1)} \sum_{j \neq i=1}^{k} t_j t_i \right]$$

$$= \frac{1}{k} \left[\sum_{j=1}^{k} t_j^2 \right] - \frac{1}{k(k-1)} \left[\sum_{j \neq i=1}^{k} t_j t_i \right]$$

$$E(s^2) = \frac{1}{k} E\left[\sum_{j=1}^{k} t_j^2 \right] - \frac{1}{k(k-1)} E\left[\sum_{j \neq i=1}^{k} t_j t_i \right] \quad \textbf{…. (1)}$$

We use a_j as indicator function as in theorem (2), Similar could be function a_i $(i \neq j)$

$$E\left[\sum_{j=1}^{k} t_j^2 \right] = E\left[\sum_{j=1}^{N} a_j T_j^2 \right]$$

$$= \sum_{j=1}^{N} \mathrm{E}(a_j)\, T_j^2 = \frac{k}{N} \sum_{j=1}^{N} T_j^2 \qquad \text{.... (2)}$$

$$= E\left[\sum_{j \neq i=1}^{k} t_j t_i\right] = E\left[\sum_{j \neq i=1}^{N} (a_j a_i)\, T_i T_j\right]$$

$$= \sum_{j \neq i=1}^{N} E(a_j a_i)\, T_j T_i \qquad \text{.... (3)}$$

But $\mathrm{E}(a_j a_i) = 1.\mathrm{P}(a_j a_i = 1) + 0.\mathrm{P}(a_j a_i = 0)$

$$= \frac{k}{N} \frac{(k-1)}{(N-1)}$$

So,

$$\mathrm{E}\left[\sum_{j \neq i=1}^{k} t_j t_i = \right] \frac{k}{N} \frac{(k-1)}{(N-1)} \sum_{j \neq i=1}^{N} T_j T_i \qquad \text{....... (4)}$$

From equation (1), (2), (3) and (4), we have

$$E(s^2) = \frac{1}{k}\left[\frac{k}{N}\sum_{j=1}^{N} T_j^2\right] - \frac{1}{k(k-1)}\left[\frac{k(k-1)}{N(N-1)} \sum_{j \neq i=1}^{N} T_j T_i\right]$$

$$= \frac{1}{N}\sum_{j=1}^{N} T_j^2 - \frac{1}{N(N-1)} \sum_{j \neq i=1}^{N} T_j T_i$$

$$= \frac{1}{N-1}\left[\sum_{j=1}^{N} T_j^2 - N\overline{T}^2\right] = S^2$$

Theorem 4: In lottery scheduling with k processors the variance of the processed time sample mean is

$$V(\bar{t}) = \left(\frac{N-k}{N}\right)\frac{S^2}{k}$$

Proof: We have

$$V(\bar{t}) = E(\bar{t}^2) - \left[E(\bar{t})\right] = \left[E(\bar{t}^2) - \overline{T}^2\right] \qquad \text{... (5)}$$

Now,

$$E(\bar{t}^2) = E\left(\frac{1}{k}\sum_{i=1}^{k} t_j\right)^2 = \frac{1}{k^2} E\left(\sum_{i=1}^{k} t_j^2 + \sum_{i \neq j=1}^{k} t_j t_i\right)$$

$$= \frac{1}{k^2}\left[E\left(\sum_{i=1}^{k} t_j^2\right) + E\left(\sum_{i \neq j=1}^{k} t_j t_i\right)\right] \qquad \text{.... (6)}$$

We have

$$E\left(\sum_{i=1}^{k} t_j^2\right) = \frac{k}{N}\left[\sum_{i=1}^{N} T_j^2\right] \qquad \text{..... (7)}$$

From equation (2), we get

$$E\left(\sum_{i \neq j=1}^{k} t_j t_i\right) = \frac{k(k-1)}{N(N-1)}\left[\left(\sum_{i=1}^{N} T_j\right)^2 - \sum_{i=1}^{N} T_j^2\right]$$

$$= \frac{k(k-1)}{N(N-1)}\left[N(N-1)\overline{T}^2 - (N-1)S^2\right]$$

$$= k(k-1)\left[\overline{T}^2 - \frac{S^2}{N}\right] \qquad \text{.......... (8)}$$

Substituting in (6) we get,

$$E(\bar{t}^2) = \frac{1}{k}\left[\left(1 - \frac{1}{N}\right)S^2 + \overline{T}^2\right] + \left(1 - \frac{1}{k}\right)\left[\overline{T}^2 - \frac{S^2}{N}\right]$$

$$= \left[\overline{T}^2 + \left(\frac{1}{k} - \frac{1}{N}\right)S^2\right]$$

Substituting in (5) we get,

$$V(\bar{t}) = \left(\frac{1}{k} - \frac{1}{N}\right)S^2 = \left(\frac{N-k}{N}\right)\frac{S^2}{k}$$

Remark Theorems 1, 2 and 3 are derived from [2]. These theorems help to develop a prediction procedure in the form of constructing confidence intervals.

4. NUMERICAL DATA

Considered 30 processes in the ready queue and their CPU burst time as shown in table 1.

Table 1.Total Processes with CPU Burst Time (*N*=30)

Processes	P_1	P_2	P_3	P_4	P_5
CPU Burst Time	30	20	112	40	59
Processes	P_6	P_7	P_8	P_9	P_{10}
CPU Burst Time	60	33	43	101	69
Processes	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
CPU Burst Time	138	43	109	26	74
Processes	P_{16}	P_{17}	P_{18}	P_{19}	P_{20}
CPU Burst Time	89	123	67	58	84
Processes	P_{21}	P_{22}	P_{23}	P_{24}	P_{25}
CPU Burst Time	143	29	147	94	131
Processes	P_{26}	P_{27}	P_{28}	P_{29}	P_{30}
CPU Burst Time	79	46	59	72	22

5. COMPUTATION UNDER LOTTERY SCHEDULING (USL) SCHEME

Considered five processors environment, each receives on job and process it. It takes random samples of 5 processes from given 30 processes under lottery scheduling as shown in table 1 and calculated their sample mean time. There are total ${}^{30}C_5$ samples possible.

Table 2.Computation of Sample Mean Time for USL

Random Sample Number	**Sampled Process (*k*=5)**	**Sampled Processing Time**	**Sample Mean Time**
1.	$P_6, P_{11}, P_{21}, P_{23}, P_{29}$	60, 138, 143, 147, 72	112
2.	$P_9, P_{27}, P_{11}, P_7, P_3$	101, 46, 138, 33, 112	86
3.	$P_{15}, P_{20}, P_{26}, P_5, P_{10}$	74, 84, 79, 59, 69	73
4.	$P_{23}, P_{17}, P_{12}, P_3, P_{10}$	147, 123, 43, 112, 69	98.8
5.	$P_{30}, P_{21}, P_{13}, P_4, P_{16}$	22, 143, 109, 40, 60	74.8
6.	$P_{28}, P_{15}, P_{14}, P_2, P_6$	59, 58, 26, 20, 131	58.8
7.	$P_1, P_{14}, P_{16}, P_{24}, P_{29}$	30, 26, 89, 94, 72	62.2
8.	$P_{27}, P_{19}, P_{13}, P_2, P_6$	46, 58, 109, 20, 60	58.6
9.	$P_{23}, P_{17}, P_{11}, P_4, P_{13}$	147, 123, 138, 40, 109	111.4
10.	$P_{24}, P_{16}, P_8, P_5, P_{19}$	94, 89, 43, 59, 58	68.6

Table 3.Computational Values for Total Processes

Total Numbers of Processes N	=	30
Mean Time $\bar{Y}$	=	73.33
Square of Mean Time $\bar{Y}^2$	=	5377.28
Total Sum of Squares $\sum_{i=1}^{30} Y_i^2$	=	203712
Mean Square S^2	=	1461.8390
Variance of Usual Lottery Scheduling $V(\bar{t})_{ULS}$	=	243.6398

Confidence Interval: The 99% confidence interval for mean time is $\left[\bar{t}-3\sqrt{V(\bar{t})}, \bar{t}+3\sqrt{V(\bar{t})}\right] = 0.99$

Table 4.Computation of Confidence Interval

Random Sample	Sampled Processing Time(in units)	Total	Sampled Mean	Confidence Interval of Time for per unit process
1.	60,138,143, 147,72	60	112	(67.17,158.82)
2.	101,46,138, 33,112	430	86	(39.17,132.82)
3.	74,84,79, 59, 69	365	73	(26.17,119.82)
4.	147,123,43, 112,69	494	98.8	(51.97,145.62)
5.	22,143,109, 40,60	374	74.8	(27.97,121.62)
6.	59,58,26, 20,131	294	58.8	(11.97,105.62)
7.	30,26,59, 94,72	311	62.2	(15.37,109.02)
8.	46,58,109, 20,60	293	58.6	(11.77,105.42)
9.	147,123, 138,40,109	557	111.4	(64.57,158.22)
10.	94,59,43, 59,58	343	68.6	(21.77,115.42)

6. CONCLUSION

The sample mean of processed time duration $\bar{t}$ predicts about the average value required to process the entire ready queue. If sudden breakdown of system occurs and until then k jobs are already completed, the backup management required duration (n-k)$\bar{t}$ to pack up all the remaining processes. Most of sample estimates are within the 99% confidence interval which proves the efficiency of the estimation procedure of ready queue under the Lottery Scheduling with k processors. The average of upper limit of confidence intervals (or maximum value) could be taken as a standard to plan for system failure management.

REFERENCES

[1] Carl A. Waldspurger William E. Weihl (1994), "Lottery Scheduling a flexible proportional-share resource management", *Proceedings of the 1st USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp.1-11.

[2] Cochran, W.G (2005), *Sampling Technique*, Wiley Eastern Publication, New Delhi.

[3] David Petrou, Garth A. Gibson, and John W. Milford (1999), "Implementing Lottery Scheduling: Matching the specializations in Traditional Schedulers", *Proceedings of the USENIX Annual Technical Conference* USA, pp.66-80.

[4] Raz, D., B. Itzahak, H. Levy (2004), "Classes, Priorities and Fairness in Queuing Systems", Research report, Rutgers University.

[5] Shukla, D. and Jain, Saurabh (2007), "A Markov chain model for multilevel queue scheduler in operating system", *Proceedings of International Conference on Mathematics and Computer Science*, ICMCS-07, pp. 522-526.

[6] Shukla, D. and Jain, Saurabh (2007), "Deadlock state study in security based multilevel queue scheduling scheme in operating system", *Proceedings of National Conference on Network Security and Management*, NCNSM-07, pp. 166-175.

[7] Shukla,D.and Jain, Anjali, Choduary Amita (2010), "Estimation of ready queue processing time under SL scheduling scheme in multiprocessor environment"**,** *International Journal of Computer Science and Security* (IJCSS),4(1) pp. 74-81.

[8] Shukla,D.and Jain, Anjali, Choduary Amita (2010), "Estimation of ready queue processing time under Usual Group Lottery Scheduling(GLS) in Multiprocessor Environment**,"** *International Journal of Computer and Applications* (IJCA),Vol 8 no. 14 pp. 39-45.

[9] Silberschatz, A. and Galvin, P. (1999), *Operating System Concepts*, Ed.5, John Wiley and Sons (Asia), Inc.

[10] Singh, Daroga and Choudhary F.S. (1986), *Theory and Analysis of Sample Survey and Designs*, Wiley Eastern Limited, New Delhi.

[11] Stalling, W. (2004), *Operating System*, Ed.5, Pearson Education, Singapore, Indian Edition, New Delhi,.

[12] Tanenbaum, A. (2000), *Operating system*, Ed. 8, Prentice Hall of India, New Delhi.