Estimation of Ready Queue Processing Time Under SL Scheduling Scheme in Multiprocessors Environment

D. Shukla diwakarshukla@rediffmail.com
Deptt. of Mathematics and Statistics
Dr. H.S.Gour Central University
Sagar (M.P.),470003, INDIA

Anjali Jain anjalidcsa@rediffmail.com
Deptt. of Computer Science and Applications
Dr. H.S.Gour Central University
Sagar (M.P.), 470003, INDIA

Amita Chowdhary amita411@gmail.com
Deptt. of Physics and Electronics
Dr. H.S.Gour Central University
Sagar (M.P.), 470003, INDIA

Abstract

CPU Scheduling is an open area of research where computer scientists used to design efficient scheduling algorithms for CPU processes in order to get output in the efficient manner. There are many CPU scheduling schemes available in literature. Lottery scheduling is one of them which adopts random choice of processes by the processors. This paper presents a new CPU scheduling scheme in the form of SL Scheduling which is found useful and effective. By virtue of this, an attempt has been made to estimate the total processing time of all the processes present in ready queue waiting for their processing. A numerical study is incorporated in the content to support the mathematical findings related to the estimation of processing time.

Keywords: CPU, Ready Queue, Scheduling, SL Scheduling (SLS), Lottery Scheduling.

1. INTRODUCTION

The scheduling is a methodology of queue of processes to minimize delay and to optimize performance of the system in the multiple processor environment where queues of processes exist with servers. A scheduler is part of an operating system module whose primary objective is to optimize system performance according to the criteria set by the system designers. It refers to a set of policies and mechanism, built into the operating system, which governs the order in which work to be done by computer system [see Silberschatz and Galvin [13], Stalling [9] and Tanenbaum and Woodhull [15]]. There are many CPU scheduling schemes available like FIFO, Round Robin, LIFO, DRRA etc. The lottery scheduling is one more, based on a probabilistic scheduling algorithm for in which processes are assigned some numbers in the form of lottery tickets, and the scheduler draws a random ticket to select the process. The distribution of tickets need not be uniform; granting a process more tickets to provide a relatively higher chance of selection. This technique can be used to approximate other scheduling algorithms, such as shortest- job – next and fair- share scheduling etc.. In other words, lottery scheduling is highly responsive because it solves the problem of starvation also, giving each process at least one lottery ticket which guarantees that it has non- zero probability of being selected at each scheduling operation. Suppose that there are many processors and each fetches a process at a time from the ready queue under lottery scheduling scheme. Then this may be treated as a random sample from the long ready queue of processes. There are techniques available in the literature sampling theory by which one can improve upon the quality of sample. This paper presents a new scheduling scheme as SL scheduling (modified form of lottery scheduling) and the approach has been adopted to estimate total processing time likely to consume if entire ready queue becomes empty.

2. A REVIEW

Lottery Scheduling by Waldsparger et al. [3] has recently introduced proportional share scheduler that enables flexible control over the relative rates at which CPU- bound work loads consume processor time. David et al. [5] extended lottery scheduling, a proportional share resource management algorithm, to provide the performance assurances present in traditional non-real time process schedulers. They used dynamic tickets adjustments to incorporate into a lottery scheduler the specialization present in the Free BSD scheduler to improve interactive response time and reduce kernel lock contention, which enables flexible control over relative process execution rates with a ticket abstraction and provides load insulation among group of processes using concurrencies. Shukla and Jain [7, 8] examined the multilevel queue scheduling scheme and examined the deadlock property using stochastic process. Shukla and Jain [9] presented deficit round robin alternated (DRRA) scheduling algorithm under Markov chain model and examined variety of scheduling scheme and their relative mutual comparisons by simulation study. Raz et al. [6] described *n* jobs to service, *p* class of priority, and m servers for the queue which holds tasks to execute and introduce some simulation results for the formula for dynamic priority calculation for CMPQ. The goal is to assure that even in worst case situations starvation does not occur. Cochran [4] contains an introduction to the methods of sampling theory with applications over multiple data. One more contribution is due to Tanenbaum and Woodhull [15].

3. MOTIVATION

Deriving an idea from all these contributions, this paper is an attempt to estimate possible time duration in case when a bank server or power supply is suddenly shut down to avoid disaster for few minutes. If some processes are running on different machines then it is not wise to stop them all of a sudden. In such a case one may desire know after what time they all will be finished from ready queue, then after estimating time duration we will be able to stop processing. Therefore, it is an open problem for researcher to estimate the total time of all processes in the ready queue likely to be consumed before closing the systems. Efficient sampling methodologies could be useful at this level to develop computational technique.

4. SL SCHEDULING SCHEME

SL Scheduling (SLS) scheme employs a technique in which the complete and up-to-date list of the processes is available in the Ready Queue of the system. It selects only the first process in random manner and the rest being automatically selected according to some predetermined pattern. The random number *'i'* is random start whose value is determined by CPU logic unit. The CPU then estimates duration of possible processing time of all *N* processes at the end of a session. The SL scheduling is laid down as under:

a) Assume N processes in the ready queue and the number N is such that $N=nk$ holds for any positive number n and k. The system has k processors in multiprocessor environment. Every process in ready is assigned a token of serial number 1 to N while arrival.

b) The CPU restricts a session in which all N ready queue processes are available for execution.

c) Scheduling chooses randomly a serial number i $(1 \leq i \leq n)$.This process is assigned to the first processor Q_1.

d) The other processors $Q_2 \ldots . Q_k$ are assigned processes having serial number $[i+n, i+2n, i+3n \ldots \ldots . i + (n-1)\, k]$.

e) At the end of the first job processing session CPU computes mean time of all k jobs processed in a session.

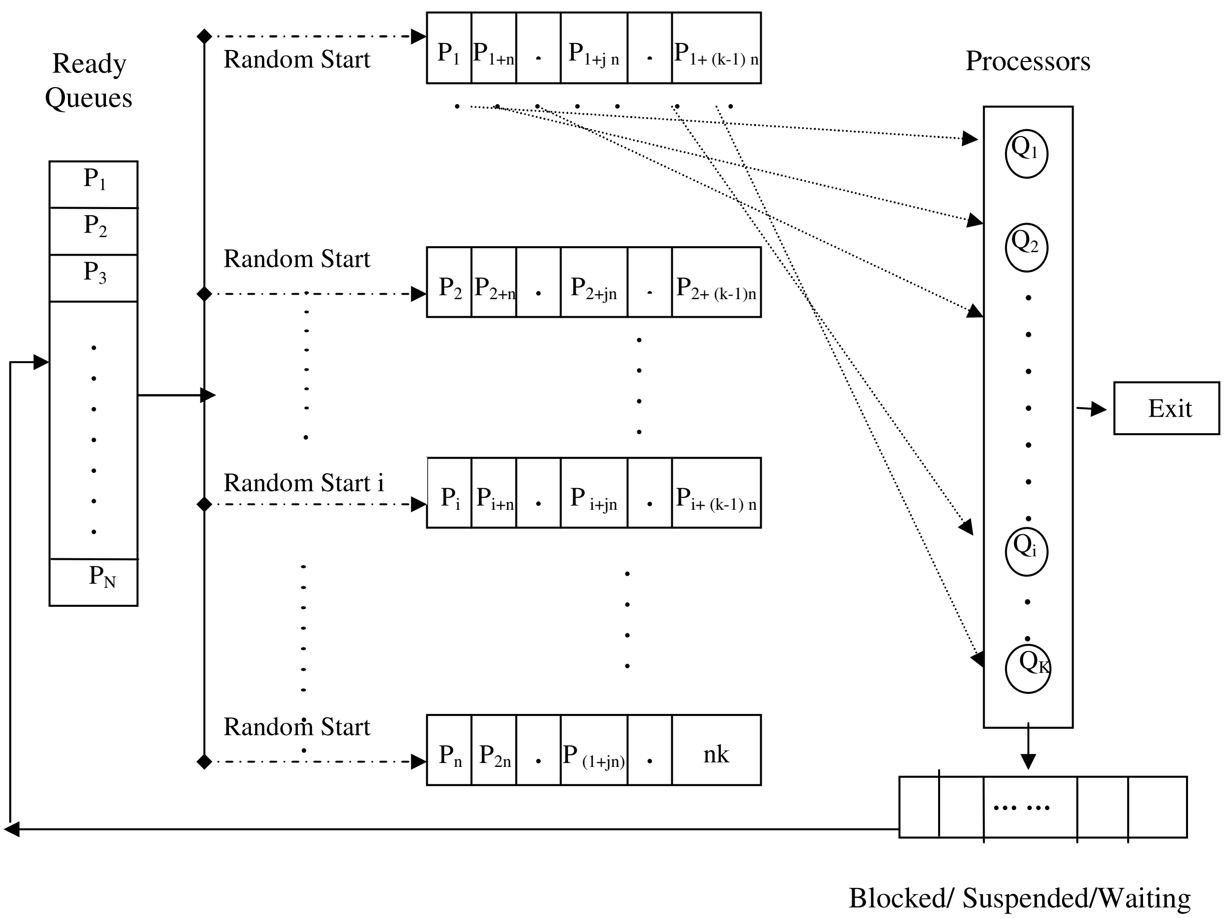

FIGURE 1: Processing of Ready Queue under Systematic Lottery Scheduling Scheme

5. ESTIMATION OF READY QUEUE PROCESS TIME IN A SESSION

Let t_{ij} denote the time of processing consumed for j^{th} process of the i^{th} sample, ($i=1,2,\ldots\ldots n; j=1,2,\ldots\ldots k$).

$\bar{t}_{i.}$ = Mean of the i^{th} systematic sample

$$=1/k\sum_{j=1}^{k}t_{ij} \qquad \text{…. (5.1)}$$

$\bar{t}_{..}$ = Overall process mean time of *N* processes in ready queue

$$=1/nk\sum_{i=1}^{k}\sum_{j=1}^{n}t_{ij}=1/n\sum_{i=1}^{n}\bar{t}_{i.} \qquad \text{…. (5.2)}$$

S^2 = Mean square of processing time for all N processes in ready queue

$$=1/(N-1)\sum_{i=1}^{k}\sum_{j=1}^{n}\left(t_{ij}-\bar{t}_{..}\right)^2=1/(nk-1)\sum_{i=1}^{k}\sum_{j=1}^{n}\left(t_{ij}-\bar{t}_{..}\right)^2 \qquad \text{…. (5.3)}$$

TABLE 1: The *k* possible systematic samples together with their means

Random Start	**Sample Composition** ***(Units in the sample)***	**Probability**	**Mean**
1	1 1+n……1+jn………1+ (k-1) n	1/n	$\bar{t}_{1.}$
2	2 2+n…….2+jn………2+ (k-1) n	1/n	$\bar{t}_{2.}$
.			
i	i i+n…… i+jn……..i+ (k-1) n	1/n	$\bar{t}_{i.}$
.			
n	K 2n…… (1+j) n……. nk	1/n	$\bar{t}_{k.}$

Thus *k* rows of the table 1 gives the *k*-systematic random samples. The probability of selecting i^{th} group of processes as the systematic sample is *1/n.* The $\bar{t}_{i.}$ is sample mean time consumed by K processors each to process one job in a session. The expected value of sample mean is

$$\text{E}\left(\bar{t}_{i.}\right)=1/k\sum_{i=1}^{k}\bar{t}_{.}=\bar{t}_{..} \qquad \text{….(5.4)}$$

So if *N= nk*, the process sample mean provides an unbiased estimate of the entire processes ready queue mean. Let $\bar{t}_{sys}$ is mean time of one systematic sample of size *k* units. Then $\bar{t}_{sys}$ is estimator of ready queue mean time and

$$\bar{t}_{sys}=\bar{t}_{i.}$$

5.1 Variance of the Estimated Mean

$$\text{Var}\left(\bar{t}_{sys}\right)=1/n\sum_{i=1}^{n}\left(\bar{t}_{i.}-\bar{t}_{..}\right)^2 \qquad \text{…. (5.5)}$$

$$\text{Var}\left(\bar{t}_{sys}\right) = ((N-1)/N)S^2 - ((n-1)k/N)S^2_{sys} \quad \text{.... (5.6)}$$

where $$S^2_{sys} = 1/k(n-1)\sum_{i=1}^{n}\sum_{j=1}^{k}\left(t_{ij} - \bar{t}_{i.}\right)^2 \quad \text{.... (5.6a)}$$

Which is the mean square among process time k units which lie within the same systematic samples.

6. NUMERICAL ILLUSTRATION

TABLE 2: Data Set

Processes	P_1	P_2	P_3	P_4	P_5
CPU Time	30	20	112	40	59

Processes	P_6	P_7	P_8	P_9	P_{10}
CPU Time	60	33	43	101	69

Processes	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
CPU Time	138	43	109	26	74

Processes	P_{16}	P_{17}	P_{18}	P_{19}	P_{20}
CPU Time	89	123	67	58	84

Processes	P_{21}	P_{22}	P_{23}	P_{24}	P_{25}
CPU Time	143	29	147	94	131

Processes	P_{26}	P_{27}	P_{28}	P_{29}	P_{30}
CPU Time	79	46	59	72	22

Considered *30* processes in the ready queue and their CPU time as shown in table 2 with *n=5*, *k=6* and *N=nk* holds.

6.1. Under Systematic Lottery Scheduling (SLS) Scheme

We have taken random samples of *6* processes from given *30* processes as shown in table 2 and find their sample mean time as shown in table 3.

TABLE 3: Computation of Sample Mean Time for SLS

Sample number for random start n=5	**Sampled Process (*k*=6) Sampled Processing Time**	**Sample Mean Time**
i=1	P_1 =30, P_6 = 60, P_{11} = 138, P_{16} = 89, P_{21} = 143, p_{26}=79	89.83

i=2	P_2 = 20, P_7 = 33, P_{12} = 43, P_{17} = 123, P_{22} = 29, P_{27} =46	49
i=3	P_3 = 112, P_8 = 43, P_{13} = 109, P_{18} = 67, P_{23} = 147, P_{28} = 59	89.5
i=4	P_4 = 40, P_9 = 101, P_{14} = 26, P_{19} = 58, P_{24} = 94, P_{29} = 72	65.16
i=5	P_5 = 59, P_{10} = 69, P_{15} = 74, P_{20} = 84, P_{25} = 131, P_{30} = 22	73.16

TABLE 4: Computational Values for Total Processes

Total Numbers of Processes *N*	30
Mean Time $\bar{t}_{..}$	73.33
Square of Mean Time	5377.28
Total Sum of Squares $\sum_{i=1}^{n}\sum_{j=1}^{k} t_{ij}^2$	203712
Mean Square S^2	1461.8390
Variance of SL Scheduling $Var(\bar{t}_{sys})$	238.48

Confidence Interval: The 99% confidence interval is $\left[\bar{t}_{sys} - 1.96\sqrt{V(\bar{t}_{sys})}, \bar{t}_{sys} + 1.96\sqrt{V(\bar{t}_{sys})}\right]$

TABLE 5: Computation of Confidence Intervals

Random Sample	Sampled Processing Time	Total Time	Sampled Mean	Confidence Interval of Time for per process	Confidence Interval for Total Time for complete Ready Queue
1.	30,60,138,89,143,79	539	89.83	(59.57,120.09)	(1787.1,3627)
2.	20,33,43,123,29,46	294	49	(18.74,79.26)	(562.2,2377.8)

3.	112,43,109,67,147,59	537	89.5	(59.24,119.76)	(1777.2,3592)
4.	40,101,26,58,95,72	391	65.16	(34.9,95.42)	(1047,2862.6)
5.	59,69,74,84,131,22	439	73.16	(42.9,103.42)	(1287,3102.6)

7. CONCLUDING REMARKS

It is observed that SL scheduling is a more scientific way of representing algorithm than usual lottery scheduling. The unique feature it has, to provide procedure of estimating ready queue processing time. Since sample representation is better by this procedure, so the queue time estimation is also sharper. In table 5, most of confidence intervals contain true value within the 99% confidence limits. It seems SL scheduling helps to estimate ready queue time processing length in advance. These estimates are useful when suddenly the system needs to shut down due to unavoidable reasons.

8. REFERENCES

1. Ankur Agarwal "System-Level Modeling of a Network-on-Chip", International Journal of Computer Science and Security (IJCSS), 3(3):154-174, 2009.
2. Agarwal, Rinki and Kaur Lakhwinder "On flexibility analysis of fault tolerant multistage interconnection networks, International Journal of Computer Science and Security (IJCSS),2(4), 01 – 08,2008.
3. Carl A. Waldspurger William E. Weihl "Lottery Scheduling a flexible proportional-share resource management", Proceedings of the 1st USENIX Symposium on Operating Systems Design and Implementation (OSDI): 1-11, 1994.
4. Cochran, W.G "Sampling Technique", Wiley Eastern Publication, New Delhi 2005.
5. David Petrou, Garth A. Gibson, John W. Milford "Implementing Lottery Scheduling: Matching the specializations in Traditional Schedulers", Proceedings of the USENIX Annual Technical Conference USA: 66-80, 1999.
6. Raz, D., B. Itzahak, H. Levy "Classes, Priorities and Fairness in Queuing Systems". Research report, Rutgers University, 2004.
7. Shukla, D. and Jain, Saurabh. "*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, 2007.
8. Shukla, D. and Jain, Saurabh. "*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, 2007.
9. Shukla, D. and Jain, Saurabh "A Markov chain model for Deficit Round Robin Alternated (DRRA) scheduling algorithm", Proceedings of the International Conference on Mathematics and Computer Science, ICMCS-08: 52-61, 2008.
10. Shukla D., Tiwari Virendra, Thakur Sanjay, And Deshmukh, A. K. "Share Loss Analysis of Internet Traffic Distribution in Computer Networks" International Journal of Computer Science And Security(IJCSS), 3(5): 414- 427,2009.
11. Shukla D., Tiwari Virendra, Thakur Sanjay, And Tiwari Mohan "A comparison of methods for internet traffic in computer network", International Journal of Advance Networking and Applications, 1(3): 164- 169, 2009.
12. Shukla D., Ojha, Shweta, And Jain, Sourabh, "Analysis of multilevel queue with the effect of data model approach", Proceedings of the National Conference on Research and Development Trends in ICT (NCRTICT – 10), 245- 251, 2010.

13. Silberschatz, A. and Galvin, P. "*Operating System Concepts"*, Ed.5, John Wiley and Sons (Asia), Inc. (1999)
14. Stalling, W. "*Operating System"*, Ed.5, Pearson Education, Singapore, Indian Edition, New Delhi. (2004)
15. Tanenbaum, A. and Woodhull "*Operating system"*, Ed. 8, Prentice Hall of India, New Delhi,(2000).