A Survey on Research Trends for Modeling CPU Scheduling Algorithm

Rupesh Sendre

Assistant Professor, Dept. of Computer Science

International Institute of Professional Studies,

Devi Ahilya University, Indore, M.P., India.

ABSTRACT

An operating system is the brain of a computer of computer system who constantly and continuously manages the resources available around the system in optimum way. One of the basic and most important tasks, an OS needs to perform, is job scheduling where many processing requests arrive from multiple channels to a ready queue and system manages all in a way to achieve high efficiency level. CPU scheduling is a fundamental operating system function that determines which of the process have to be executed next when multiple run able process are waiting in the ready queue. The aim of CPU scheduling is to execute by the processor or processors over time, in a way that meets system objectives such as response time, throughput, and processor's efficiency. In the proposed paper I have discussed the various approaches that can be used for this purpose & elaborate the research trends in this field. A Markov chain analysis is done in order to determine the performance of this suggested algorithm. These efforts have found very efficient and useful.

Keywords: Operating system, CPU scheduling, Markov chain, Stochastic process.

INTRODUCTION

An operating system, termed as OS, is the brain of a computer of computer system who constantly and continuously manages the resources available around the system in optimum way. An operating system is a software (a program), which controls the execution of many other application programs and acts as an interface between computer hardware and applications. It has some attractive features like Multiprogramming, Multitasking, and Multi-users etc, which place it way ahead in the race with human mind. One of the basic and most important tasks, an OS needs to perform, is job scheduling where many processing requests arrive from multiple channels to a ready queue and system manages all in a way to achieve high efficiency level. For this, scheduling algorithms are required having pre-set systematic steps. In wider sense, one can assume random behavior of the scheduler as an extension to usual, in light of algorithm, and at the certainty level the specific algorithm or many similar may be obtained. This idea generates a class of algorithms and provides a common platform to compare many similar at a single base.

Scheduling of Processes

The aim of CPU scheduling is to execute by the processor or processors over time, in a way that meets system objectives such as response time, throughput, and processor's efficiency. In many systems, this scheduling activity is broken down into three separate functions: long-, medium-, short-term scheduling. The names suggest the relative time scales with which functions are performed. Long-term scheduling is performed when a new process is created. This is a decision to add a new process to a set of processes that are currently active. Medium-term scheduling is a part of swapping function. This is a decision to add a process to those that are partially in main memory and therefore available for execution. Short-term scheduling is the actual decision of which ready process to execute next. Scheduling affects the performance of the system because it determines which process will wait and which will progress. Fundamentally, scheduling is a matter of managing queues to minimize queuing delay and optimize performance of queuing environment.

Scheduling Algorithms

The commonly used criteria can be categorized along two directions. First, we can make a distinction between user-oriented and system-oriented criteria. User-oriented criteria relate to the behavior of the system as perceived by the individual user or process. An example is a response time in an interactive system. Response time is the elapsed time between the submissions of a request until the response begins to appear as output. This quantity is visible to the user and is naturally of interest of the user.

Use of Priorities

In many systems, each process is assigned a priority and the scheduler will always choose a process of higher priority over one of lower priority. When a scheduling selection is to be made, the scheduler will start with the highest-priority ready queue. If there are one or more processes in the queue, a process is selected using some scheduling policy. One problem with a pure priority-scheduling scheme is that lower-priority process may suffer starvation. This will happen if there is always a steady supply of higher-priority ready processes. If this behavior is not desirable the priority of a process can change with its age or execution history.

First-Come First-Served

The simplest scheduling policy is First-Come First-Served (FCFS), also known as first-in first-out (FIFO) or a strict queuing scheme. As each process becomes ready, it joins the ready queue. When the currently running process ceases to execute, the process has been in the ready queue the longest is selected for running. First the finish time if each process is determined. From this, we can determine the turnaround time. In terms of the queuing model, turnaround time is the residence time, or the time that the item spends in the system (waiting time plus service time). A more useful figure is the normalized turnaround time, which is the ratio of turnaround time to service time.

Round Robin

A straightforward way to reduce the penalty that jobs suffer with FCFS is to use preemption based on a clock. The simplest such policy is round robin. A clock interrupt is generated at periodic intervals. When the interrupt occurs, the currently running process is placed in the ready queue, and the next ready job is selected on a FCFS basis. This technique is also known as time slicing, because each process is given a slice of time before being preempted. With round robin, the principal design issue is the length of time quantum, or slice, to be used. If the quantum is very short, then short processes will go through the system very quickly. On the other hand, the processing overhead involved in handling the clock interrupt and performing the scheduling and dispatching function. Round robin is generally effective in general-purpose time-sharing systems or transaction processing system. One drawback of round robin is its relative treatment of processor-bound and I/O-bound processes.

Shortest Process Next

Another approach to reducing the bias in favor of long processes inherent in FCFS is the shortest process next (SPN) policy. This is nonpreemptive policy in which the process with shortest expected processing time is selected next. Thus the short process will jump to the head of the queue past longer jobs. However, the variability of response time is increased, especially for longer processes, and thus predictability is reduced. One difficulty with the SPN policy is the need to know or at least estimate the required processing time of each process. For batch jobs, the system may require the programmer to estimate the value and supply it to the operating system.

Shortest Remaining Time

The shortest remaining time (SRT) policy is a preemptive version of SPN. In this case, the scheduler always chooses the process that has shorted expected remaining expected time. When a new process joins the ready queue, it may in fact have a shorter remaining time than the currently running process. Accordingly, the scheduler preempted whenever a new process becomes ready. As with SPN, the scheduler must have an estimate of processing time to perform the selection function, and there is a risk of starvation of longer processes. SRT does not have the bias of favor of long processes found in FCFS. Unlike round robin, no additional interrupts are generated, reducing overhead. On the other hand, elapsed service times must be recorded, contributing the overhead. SRT should also give superior turnaround time performance to SPN, because a short job is given immediate preference to a running longer job.

Multi-Level Queue Scheduling

A multi-level queue scheduling algorithm partitions the ready queue into several separate queues. The processes are permanently assigned to one queue, generally based on some property of process, such as memory size, process priority, or priority type. Each queue has its own scheduling algorithm. For example, separate queues might bf for foreground and background processes. The foreground queue might be scheduled by RR algorithm, while the background queue is scheduled by FCFS algorithm. In addition, there must be scheduling between the queues, which is commonly implemented as a fixed-priority preemptive scheduling. For example, the foreground queue may have absolute priority over background queue.

Stochastic process

The set of possible values of an individual random variable X_n (or $X(t)$) of a stochastic process $\{X_n, n\geq1\}$, $\{X(t), t\in T\}$ is known as state space. The state space is discrete if it contains a finite or a denumerable infinity of points; otherwise, it is continuous. For example, if X_n is the total number of sixes appearing in the first n throws of a die, the set of possible values of X_n is discrete. We can write $X_n = Y_1+\ldots+Y_n$, where Y_i is discrete random variable denoting the outcome of the i^{th} throw and Y_i=1 or 0 according as the i^{th} throw shows six or not. Secondly, consider $X_n=Z_1+\ldots. +Z_n$, where Z_i is the continuous random variable assuming values in $[0.\infty)$. Here the set of possible values of X_n is the interval $[0.\infty)$, and so the state space of X_n is continuous.

Markov Chain

Consider a simple coin tossing experiment for a number of times. The possible outcomes of each trial are two: head with probability, say p, and tail with probability q, p + q = 1. Let us denote head by 1 and tail by 0 and random variable denoting the result of n^{th} toss by X_n. Then for n = 1, 2, 3,…

$$Pr\{X_n=1\}=p\ ,\ Pr\{X_n=0\}=q$$

There is a sequence of random variables X_1, X_2, X_3,… The trails are independent and the result of n^{th} trial does not depend in any way on the previous trial numbered 1,2,.(n-1).
Consider now the random variable given by the partial sum $S_n=X_1+\ldots+X_n$. The sum S_n gives the accumulated number of heads in the first n trials and its possible values are 0,1…n. We have $S_{n+1} = S_n + X_{n+1.}$ Given that $S_n = j$ (j=0,12…n), the random variable S_{n+1} can assume only two possible values: $S_{n+1} = j$ with probability p; these probabilities are not at all affected by the values of the variables $S_1,\ldots,S_{n-1}$. Thus

$$Pr\{S_{n+1} = j+1/S_n=j\}=p$$
$$Pr\{S_{n+1} = j/S_n=j\}=q$$

Definition: The stochastic process $\{X_n, n=0,1,2\ldots\}$ is called Markov chain, if, for $j,k,j_1,\ldots j_{n-1}$ € N (or any subset of I),

$$Pr\{X_n = k / X_{n-1} = j\ , X_{n-2} = j_1\ ,\ldots.,X_0 = j_{n-1}\}$$
$$= Pr\{X_n = k / X_{n-1} = j\} = p_{jk}\ \text{(say)}$$

Whenever first member is defined.

Transition Probability Matrix: The transition probabilities p_{jk} satisfy

$$p_{jk} \geq 0, \ \sum_k p_{jk} = 1 \text{ for all j.}$$

These probabilities may be written in the matrix form

$$P = \begin{array}{c|cccc} & \multicolumn{4}{c}{\longleftarrow X^{(n)} \longrightarrow} \\ \hline \uparrow & p_{11} & p_{12} & p_{13} & \cdots \\ X^{(n-1)} & p_{21} & p_{22} & p_{23} & \cdots \\ \downarrow & p_{31} & p_{32} & p_{33} & \cdots \\ & \cdots & \cdots & \cdots & \cdots \\ \hline \end{array}$$

This is called the transition probability matrix of the Markov chain. The P is a stochastic matrix, i.e. a square matrix with non –negative elements and unit row sums.

REVIEW OF LITERATURE

The operating system plays a major role in managing processes arriving in the form of multiple queues. The arrival of a process is random along with their different categories and types. All these require scheduling algorithms to work over real time environment with special reference to task, control and efficiency (see Stankovic (1984), Liu and Layland (1973), Garey and Johnson (1977) etc.). The randomization involved in scheduling procedure leads to perform a probabilistic study. Demer et al. (1989) have presented an analysis of Fair Queuing algorithm whereas Cobb et al. (1998) picked up fair scheduling of flaros with the consideration of time shifting approach in the area of high speed networks. Goyal,Guo,Vin (1996) derieved the Hierarchical CPU scheduler in the environment where the multimedia operating system is used. In the similar line, Hieh and Lam (2003) discussed smart schedulers for multimedia users. A time driven scheduling model is proposed by Janson,Lockey and Tokuda (1985) attracted the attention of researchers for the model formation over functioning and procedure on operating systems. Katcher et al. (1993) proposed an analysis of fixed priority schedulers and Horn (1974) generated some new scheduling algorithms useful for managing queues in operating system. David (1994) has a successful contribution over the study of real time and conventional scheduling with a comparative analysis.

Barthomew (1973), Medhi (1991 a) and Parzen (1962) have given an elaborate study of a variety of stochastic processes and their applications in various fields. Medhi (1976) developed a Markov chain model for the study of uncertain rainfall phenominon. Naldi (2002) presented a Markov chain model for understanding the internet traffic sharing among various operators in a compatetive market. Shukla et al. (2006) derived a Markov chain model for the transition probabilities in space division switches in computer networks. Medhi (1991 b) presented the use of stochastic process in the management of queues. Shukla and Jain (2007) have a discussion on the use of Markov chain model for multilevel queue scheduler in an operating system. Some other useful contributions are due to Silberschatz and Galvin (1999), Stalling (2004) and Tanenbaum and Woodhull (2000). Mohammad A.F. Al-Husainy (2007) has

presented a new CPU scheduling algorithm called Best-Job-First is suggested by mixing the functions of some well-known basic scheduling algorithm.
Shukla and Jain (2009) have a proposed on the use of Markov chain model for scheduling scheme which is the mixture of FIFO and round robin is found efficient in terms of model based study. Shukla and Ojha (2010) have a discussion on the use of data model based markov chain model for deadlock index analysis of multi-level queue scheduling in operating system. Shukla et al. (2010) have given elaborate study of a general class of multi-level queue scheduling schemes is designed and studied under a Markov chain model. Shukla et al. (2010) have present 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. Nigam and Jain (2010) have proposed new way of structuring the Markov model is proposed named as dynamic nested markov model for modeling the user web navigation sessions. Nigam and Jain (2010) have analyzed three different schemes for web prefetching and caching are proposed i.e. prefetching only, prefetching with caching and prefetching from caching. Pandey and Vandana (2010) have suggested approach uses two ready queues, wherein a process is returned to the second ready queue after the completion of its penultimate round. This policy reduces the average waiting time and increases the throughput, in comparison to the conventional round robin scheme. Shukla and Jain (2011) have present 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. Pandey et al. (2011) have proposed an attempt to analyze the collective effect of time of arrival, size of CPU burst and priority of the process, through a logical combination of all the three. Sisodia and Garg (2011) have presented a general class of round-robin scheduling in which both (FIFO & RR) the scheduling procedures are covered in the particular case. Shukla and Jain (2012) have presents an efficient method to predict about total time needed to process the entire ready queue if only few are processed in a specified time. Confidence internals are calculated based on PPS-LS and compared with SRS-LS. The PPS-LS found better over SRS-LS. Shukla and Jain (2013) have suggests two new estimation methods to predict the remaining total processing time required to process completely the ready queue provided sources of auxiliary information are negatively correlated.

CONCLUSION

In this paper the researcher introduces the concept of CPU Scheduling algorithms and states that how markov chain may be applied on CPU Scheduling for setting up approaches and trend. After reviewing the available literature the author explore some important issues and challenges associated with CPU Scheduling algorithms. This review work certainly will be helpful for the upcoming researchers who want to carry on their research in the application of Web mining on Web server log files.

REFERENCES

1. Bartholomew,D.J.(1973): Stochastic Models for Social Process , Second Edition ,John Wiley ,New York.

2. Cobb, J. Gouda, M. and EL-Nahas, A. (June, 1998): Time-Shift Fair Scheduling of Flaros in High-Speed Networks, IEEE/ACM Transactions of Networking, pp. 274-285.

3. David B. Goub, (1994): Operating System Support for Coexistence of Real-Time and Conventional Scheduling, Carneqie Mellon University, PiHsburg W. PA.

4. Goyal, P. Guo., X. and Vin, H.M.(Oct., 1996): A Hierarchical CPU Scheduler for Multimedia Operating Systems, In Proceedings of the Second Symposium on Operating Systems Design and Implementation (OSDI' 90), Secattle, WA, pp. 107-122.

5. Hieh, J. and Lam, M.S.(May, 2003): A SMART Scheduler for Multimedia Application, ACM Transactions on Computer System (TOCS), 21(2), pp. 117-163.

6. Horn, W. (1974): Some Simple Scheduling Algorithms, Naval Research Logistics Quaterly, 21,177-188.

7. Katcher et.al. (Oct., 1993): Engineering and Analysis of Fixed Priority Schedulers, IEEE Transactions of Software Engineering, pp 67-81.

8. Medhi, J. (1976): A Markov chain model for occurrence of dry and wet days, Ind. J. Met. Hydro. Geophys. , 27, 431-435.

9. Medhi, J. (1991 a): Stochastic processes, Ed. 4, Wiley Limited (Fourth Reprint), New Delhi.

10. Medhi, J. (1991 b): Stochastic models in queuing theory, Academic Press Professional, Inc, San Diego, CA.

11. Naldi, M. (2002): Internet access traffic sharing in a multi-user environment, Computer Networks, Vol. 38, pp. 809-824.

12. Parzen, E. (1962): Stochastic Process, Holden –day, Inc. San Francisco, California.

13. Silberschatz, A., Galvin, P. (2001): Operating system concept, Ed.6, John Wiley and Sons (Asia), Inc.

14. N. C. Wilhelm, “An anomaly in disk scheduling: a comparison of fcfs and sstf seek scheduling using an empirical model for disk accesses,” Communications of the ACM, vol. 19, no. 1, pp. 13–

18, 1976

15. E. G. Coffman, L. A. Klimko, and B. Ryan, "Analysis of scanning policies for reducing disk seek times," SIAM Journal on Computing, vol. 1, no. 3, pp. 269–279, 1972.

16. W. C. Oney, "Queueing analysis of the scan policy for moving-head disks," Journal of the Association for Computing Machinery, vol. 22, pp. 397–412, 1975.

17. View at Zentralblatt MATH C. C. Gotlieb and G. H. MacEwen, "Performance of movable-head disk storage devices," Journal of the ACM, vol. 20, no. 4, pp. 604–623, 1973.

18. M. Andrews, M. A. Bender, and L. Zhang, "New algorithms for the disk scheduling problem," in Proceedings of the 37th Annual Symposium on Foundations of Computer Science, pp. 550–559, Burlington, Vt, USA, 1996.

19. T.-H. Yeh, C.-M. Kuo, C.-L. Lei, and H.-C. Yen, "Competitive analysis of on-line disk scheduling," in Proceedings of the 7th International Symposium on Algorithms and Computation (ISAAC '96)

20. R. Geist, R. Reynolds, and E. Pittard, "Disk scheduling in system v," in Proceedings of the 1987 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '87), pp. 59–68, 1987.

21. M. Hofri, "Disk scheduling: Fcfs vs. sstf revisited," Communications of the ACM, vol. 23, no. 11, pp. 645–653, 1980

22. D. M. Jacobson and J. Wilkes, "Disk scheduling algorithms based on rotational position," Tech. Rep., Hewlett-Packard, Palo Alto, Calif, USA, February 1991.

23. Thomasian and C. Liu, "Some new disk scheduling policies and their performance," in Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '02), pp. 266–267, 2002.

24. Thomasian and C. Liu, "Performance evaluation for variations of the satf scheduling policy," in Proceedings of the Internationl Sympium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS '04), pp. 431–437, 2004.

25. David M. Jacobson and John Wilkes, "Disk scheduling algorithms based on rotational position" Hewlett Packard, May 1995.

26. Margo Seltzer, Peter Chen and John Ousterhout, "Disk Scheduling Revisited", Winter Usenix, Washington, January 1990.

27. Daniel T. Joyce, "An Investigation of Disk Scheduling Algorithms Laboratory", 2001.

28. Toby J. Teorey and Tad B. Pinkerton, "A comparative analysis of disk scheduling policies", March 1972, New York, NY, USA.

29. Helen D. Karatza," A Comparative Analysis of Scheduling Policies in A Distributed System Using Simulation", Thessaloniki, Greece, 2000.

30. Hu Ming, "Improved disk scheduling algorithms based on rotational position", October, 2005.

31. Stalling, W. (2004): Operating Systems, Ed.5, Pearson Eduaction,Singopore,Indian Edition ,New Delhi.

32. Stankovic, J.A. (June 1984): Simulation of three Adaptive, Decentralized controlled, Task scheduling algorithms, Computer Networks, vol. 8, No. 3, 199-217.

33. Tanenbaum, A.S. and Woodhull, A.S. (2000): Operating System: Design and Implementation, Ed. 8, Prentice Hall of India Private Limited, New Delhi.

34. Shukla, D., Gadewar, Surendra. Pathak, R.K. (2007): A Stochastic model for space-division switches in computer networks, Applied Mathematics and Computation (Elsevier Journal), Vol. 184, Issue 2, pp. 235-269.

35. Shukla, D. and Jain, Saurabh. (2007): A Markov chain model for multi-level queue scheduler in operating system, Proceedings of the International Conference on Mathematics and Computer Science, ICMCS-07, pp. 522-526.

36. Mohammad A.F. Al-Husainy (2007): Best-Job-First CPU scheduling algorithm, Information Technology Journal, Vol. 06, No. 02, pp. 288-293.

37. Shukla D. and Jain, Saurabh. (2009): A Markov chain model for the analysis of Round-Robin Scheduling Scheme, Journal of Advanced Networking and Applications, Vol. 01, No. 01, pp. 1-7.

38. Shukla, D. and Ojha, Shweta. (2010): Deadlock Index analysis of Multi-Level Queue Scheduling in Operating System using Data Model Approach, GESJ: International journal of computer science and telecommunication, Vol. 06, No. 29, pp. 93-110.

39. Shukla, D., Ojha, Shweta. And Jain, Saurabh (2010): Performance evaluation of a general class of multi-level queue scheduling scheme, GESJ: Computer Science and Telecommunications, Vol. 03, No. 26, pp. 99-122.

40. Shukla, D., Jain, Anjali and Chudhary, Amita (2010): Estimation of ready queue processing time under SL-Scheduling scheme in multiprocessors environment, International Journal of Computer Science and Security (IJCSS), Vol. 04, No. 01, pp. 74-81.

41. Nigam B. and Jain S. (2010): Generating a new model for predicting the next accessed web page in web usage mining, IEEE, pp. 485-490.

42. Nigam B. and Jain S. (2010): Analysis of Markov model on different web prefetching and caching schemes, IEEE.

43. Pandey, D. and Vandana (2010): Improved round robin policy a mathematical approach, IJCSE, Vol. 02, No. 04, pp. 948-954.

44. Shukla D. and Jain, Anjali (2011): Prediction of Ready queue processing time in multiprocessor environment using Lottery scheduling (ULS), Journal of Applied Computer Science and Mathematics, Vol. 05, No. 11, pp. 58-63.

45. Pandey, D. et al. (2011): Fuzzy better job first scheduling algorithm, International Journal of Computer Applications, Vol. 33, No. 09, pp. 13-16.

46. Sisodia, D. and Garg, S. (2011): A Markov chain model for round robin scheduling in operating system, JGRCS, Vol. 02, No. 06, pp. 1-5.

47. Shukla D. and Jain, Anjali (2012): Analysis of ready queue processing time under PPS-LS and SRS-LS scheme in multiprocessing environment, GESJ: Computer Science and Telecommunication, Vol. 01, No. 33, pp. 54-61.

48. Shukla D. and Jain, Anjali (2013): Estimation of Ready queue processing time using transformed factor type estimator (T-F-T) in multiprocessor environment, International journal of computer application (IJCA), Vol. 79, No. 16, pp. 40-48.