International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Special Issue-1 E-ISSN: 2347-2693

Minimizing Page Fault Using Queueing Theory

S. Biswas[1], D.Sengupta[2], R. Bhattacharjee[3] and M. Handique[4*]
[1234]Department of Information Technology, School of Technology
Assam University, Silchar-788011, India
[4]mousum.smit@gmail.com

***Abstract*—In this paper we recommend a compelling approach to minimize page fault rate utilizing the hypothesis of waiting lines, i.e. Queuing Theory. The Queuing Theory has wide applications such as making business decision, hospital management, traffic regulation etc. but in this paper we are simply using it to reduce the number of page faults. Hence for the said purpose a comparative analysis is made in this paper between the typical performance from ordinary execution of page replacement algorithm and the performance after actualizing these conventional algorithms on the queuing models. Therefore the performance is measured on the basis of page fault rate and the result from this comparative dissection helps us determine conclusion on the effective way of page fault reduction.**

Keywords- Memory Management, Page Fault, Queueing Theory, FIFO, LRU, OPT, M/M/1, M/M/s, M/D/1

I. INTRODUCTION

Memory management is important aspect of an operating system. It is always desirable for efficient use of memory and CPU that more and more process execute simultaneously in system. This process is known as multiprogramming. To achieve this dynamic paging is required in which occurrence of page fault is mandatory [1] as under this scheme we don't load whole program in memory but only a part of the program that is required for current execution but when CPU demands another page to execute next instruction of the program and that page is not in the memory then a page fault occurs.

A page fault is therefore a trap to the software raised by the hardware when a program accesses a page that is mapped in virtual address space, but not loaded in physical memory [2]. Individual programs face extra latency when they access a page for first time hence for enhancing system performance it becomes inevitable to reduce page faults. There are hard and soft page faults but in this literature we are concerned with the hard page faults.

Therefore, in this paper we endeavor to minimize the number of page faults using the Queuing Theory. Queuing theory is the mathematical study of waiting lines on queues. In queuing theory a model is constructed so that queue lengths and waiting lines can be predicted [3]. Queuing theory is gaining much weightage in recent times due to its applications which includes providing faster customer service, improving traffic flow, shipping orders efficiently from a ware house etc. but in this literature we are simply using the theory to analyze page fault rate so as to minimize it.

This section therefore gives the overview and motivation of our work. Section 2 states the terms and definitions that are required to be understood before one proceeds to further sections. Section 3 illustrates the previous work done on the concerned field. And the section 4 describes our main work i.e. illustration of page replacement algorithms using the queuing models. Finally we conclude with our future work in section 5.

II. BACKGROUND

This section provides the basic concept necessary for understanding of rest of the paper.

The page fault rate can be evaluated using a parameter known as the effective access time. To compute this effective access time the TLB hit ratio must be known for any given memory reference sting from which the probability of occurrence of the page fault for that reference string can be found which is then denoted by p. Further an average memory access time should also be known, denoted by ma. Hence the formula for the effective access time and requirements for its calculation are summarized below.

Definition 1
TLB hit ratio: TLB hit ratio is nothing but the ratio of TLB hits/Total no of queries into TLB. For example, if we asked the TLB 10 times for virtual-to-physical mappings, and found the mapping 4 times, then our hit ratio is 4/10 i.e. 40 percent

Definition 2
Probability of occurrence of page fault, p = (1-hit ratio).

Definition 3
Effective Access Time (EAT): Effective access time = (1−p)* ma + p * page fault service time [4].

For most computer systems the memory access time ranges from 10 to 200 nanoseconds. As long as we have no longer page fault the effective access time is equal to the memory access time [4]. The page fault service time, i.e. page-switch time, will probably be close to 8 milliseconds (with average latency of 3 milliseconds, a seek of 5 milliseconds, and transfer time of 0.05 milliseconds). The lesser the effective access time the lesser will be the number of page fault, as effective access time is directly proportional to the page fault rate [4].

III. PREVIOUS WORK

The first paper on queuing theory, The Theory of Probabilities and Telephone Conversations was published in 1909 by A.K. Erlang, now considered the father of the field. His work with the Copenhagen Telephone Company is what prompted his initial foray into the field.

He pondered the problem of determining how many telephone circuits were necessary to provide phone service that would prevent customers from waiting too long for an available circuit. In developing a solution to this problem, he began to realize that the problem of minimizing waiting time was applicable to many fields, and began developing the theory further. Erlang's switchboard problem laid the path for modern queuing theory [5]. In this paper the concept of the theory is used to lay down solutions to optimize system performance reducing the huge number of hard page faults.

Following the demand pagination scheme, when there are no free frames available to read in a page from secondary memory to primary memory a need arises to choose a victim page and swap it out to swap the required page in, this crisis gave rise to the basic page replacement algorithms. In this paper we consider the three most popular basic replacement algorithms. These conventional algorithms therefore help us to draw out a conclusion on system stability after comparative analysis of its general performance and performance after implementing these in queuing models. These traditional algorithms are briefly stated underneath.

A. FIFO

The simplest-page replacement algorithm is a first-in, first-out algorithm. When the buffer is full, the oldest page is replaced. A FIFO queue can be created to hold all the pages in memory [4]. It suffers from Belady's anomaly which implies that the page fault rate may increase with increased number of allocated frames.

B. Optimal Replacement

It is an algorithm that never has to suffer from Belady's anomaly. It replaces the page that will not be used for longer period of time. Use of this page-replacement algorithm guarantees the lowest possible page fault rate for a fixed number of frames [4]. Such an algorithm does not really exist and is impossible to implement practically.

C. Least Recently Used

If the optimal algorithm is not feasible, perhaps an approximation of the optimal algorithm is possible. If we use the recent past as an approximation of the near future, then we can replace the page that has not been used for the longest period of time. This approach is the least recently used algorithm [4]. The LRU is often used as a page replacement algorithm and is considered to be good.

These conventional algorithms will be utilized within a newer possibility to get to further areas.

IV. ILLUSTRATION OF CONVENTIONAL ALGORITHMS USIG QUEUING MODELS

This section is the representation about routine calculations utilizing queuing models. It depicts the portrayal for our primary worth of effort. We are demonstrating another way of applying these page replacement algorithms. We are not proposing alteration in the basic principles of these traditional algorithms rather using the same and applying it on the queuing models. High rate of page fault may result in thrashing, degrading the system performance therefore it becomes essential to bring about a constructive approach of reducing it. One such procedure is proposed in this paper.

There are many a queuing models but for simplicity in evaluation and computation in this paper we illustrate only three models. In this literature limited population model of queuing theory does not get highlighted. Queuing models such as M/M/1, M/M/s and M/D/1 are demonstrated.

The M/M/1 system is made of a Poisson arrival, one exponential (Poisson) server, FIFO (or not specified) queue of unlimited capacity and unlimited customer population. It should be noted that these assumptions are very strong, not satisfied for practical systems (the worst assumption is the exponential distribution of service duration - hardly satisfied by real servers). Nevertheless the M/M/1 model shows clearly the basic ideas and methods of Queuing Theory [6]. Here important point to remember is that M/M/1 follows the FIFO service discipline.

Before learning about the M/M/s model some notations important in the queuing theory should be known, they are λ which is the inter-arrival time and the other is μ which is the service time for any queuing model. All inter-arrival and service times are independently and identically distributed according to an exponential distribution. The number of servers is s. This model is a special case of the birth-and-death process where the queuing systems mean arrival rate and mean service rate per busy server are constant. When the system has just a single server (s= 1), the parameters for the birth-and-death process are $\lambda_n = \lambda$ and $\mu_n = \mu$. When the system has multiple server, ($s \geq 1$) μ_n represents the mean service rate for the overall queuing system when there are n customers in the system [7].

M/D/1 is Kendall's notation of this queuing model. The first part represents the input process, the second the service distribution, and the third the number of servers. M represents an exponentially distributed inter-arrival or service time; specifically M is an abbreviation for Markovian. The D represents a deterministic or constant inter-arrival or service time [8].

Finally applying the traditional algorithms on these queuing models we have-

A. *M/M/1*

Since the M/M/1 follows the FIFO discipline we can only apply the FIFO page replacement on it and the result for effective access time will be similar to what general FIFO page replacement algorithm would yield. Hence, since there is no change in the effective access time we cannot claim this model to be effective in minimization of page fault but the similar cannot be said for the M/M/s model.

B. *M/M/s*

The M/M/s is a multi-server model. Obviously one server may also be chosen for which condition is stated above but this would not give profitable results to serve the purpose. In effective access time calculation each page fault occurring is multiplied by the page fault service time but when more than one server is used the number of page fault gets divided by the number of servers. That means, if we have two servers then when they parallel work to divide the job for which the number of page faults are reduced as one server will be entertaining only half of the number of page faults asking for service. From this it may be concluded that in M/M/s model considering more than one server effective access time will become much lesser signifying the lower page fault rate. For practical implementation it is better not to consider more than three servers at a time.

In general implementation of the conventional algorithm the optimal replacement algorithm results in least number of page faults but this algorithm cannot be practically implemented rather it is only theoretical (as stated before) because it requires future knowledge of the string which is not possible. So this algorithm is used only for approximation purpose so that if any constructive algorithm behaves as optimal replacement algorithm it may be concluded as good algorithm with reduced number of page faults. The advantage of FIFO is that is can hold as

many process as given but it gives degraded result as compared to others. So in such case the LRU is found to approximate the results of optimal replacement. It is often used as a good page replacement algorithm, though much hardware assistance in this algorithm can improve it but in general too it gives optimal results.

So, if the combination of LRU and M/M/s queuing model is considered it would be highly effective declining page fault rate. Implementing LRU in M/M/s we would get less effective access time with lesser number of page faults.

C. M/D/1

The M/D/1 model signifies deterministic service time. So if for comparison purpose we consider same arrival rate and service time for all the models then the M/D/1model is found to behave just similar as M/M/1 model. The distinction in performance remains that for M/M/1 only FIFO page replacement can be implemented which gives higher page faults but in M/D/1 with specific service time OPR and LRU can also be implemented. So if we consider the LRU performance result in M/D/1 then, the M/D/1 model behaves better than M/M/1 queuing model.

The illustrated matter could be visualized properly in the next section.

V. EXPERIMENTAL RESULTS

The above exhibited matter can be better clarified in this segment with the help of an example.

Example: Let us consider a memory reference string (used widely for demonstration purpose) [4]:
7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

For the given reference string the FIFO page replacement algorithm results in 15 number of page faults with a higher effective access time.

The same result could be viewed in case in of FIFO implementing in M/M/1 model. The optimal replacement algorithm results in 9 page faults and LRU results in 12 page faults. The effective access time if found simply for these conventional algorithms without implementing queuing models would yield terrible outcome. In contrast, if the queuing models were used implementing these replacement algorithms at times it would give very profitable results. The comparison can be seen in the table below if effective access time is calculated in each case–

TABLE I: Without using queuing theory

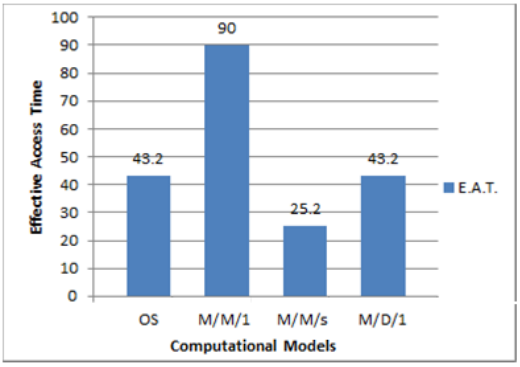
S. No	Parameters	FIFO	OPR	LRU
1	effective access time	90.000050	43.200110	57.600080

TABLE II: Using queuing theory

S. No	Models	M/M/1	M/M/s	M/D/1
1	FIFO	90.000050	45.000050	90.000050
2	OPR	43.20010	16.200050	43.200110
3	LRU	57.600080	25.200080	57.600080

The above computed values are found through formulae and methods stated in past segments of this paper. All the calculated effective access times above are in **milliseconds**.

Any comparative analysis is incomplete without demonstrating the graphical representation for the same. It gives the readers a clear and complete idea of the computational results without harassing the readers with complex formulations. So, the result of the comparative analysis is given below.



The OS considers the LRU performance, optimal being only a theoretical estimation. The same is followed by M/D/1 but the M/M/s implemented with LRU is found to yield optimal and practical results from the above graphical representation serving the purpose of a reduced page fault rate.

Obviously there are other procedures to reduce page faults. But we claim the above mentioned as an optimal one.

VI. CONCLUSION AND FUTURE WORK

The paper provides reader with easier way of interpretation of the matters depicted in it. Therefore from the experimental results it can be concluded that implementing LRU in M/M/s queuing model the page fault rate can be reduced highly thus improving CPU utilization and decreasing program latency. Further for future work other queuing models may also be considered. Also in this paper we are only concerned with the proposed theory and not its hardware implementation so in future some research of hardware aids for the proposed procedure in the literature may be done.

REFERENCES

[1] G. K. Vijay Srivastava, "A new approach to minimize page fault," International Conferrence on Information and Computer Networks, vol. IPCSIT 27, p. 99, 2012.

[2] Wikipedia, "Page fault — wikipedia, the free encyclopedia," 2014, [Online; accessed 21-December-2014]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Page\ fault\&oldid=638911887

[3] ——, "Queueing theory — wikipedia, the free encyclopedia," 2014,[Online; accessed 21-December-2014]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Queueing\ theory\&oldid=635735110

[4] G. G. Abraham S., Peter B. Galvin, "Operating systems: Design and implementation," vol. 8th Edition, pp. 365–376, 2010.

[5] R. Berry, "Queueing theory," Senior Project Archive, p. 1, 2006. [Online]. Available: https://www.whitman.edu/mathematics/SeniorProjectArchive/2006/berryrm.pdf

[6] "The m/m/1 queueing system," 2000, [Online; accessed 22-December-2014]. [Online]. Available: http://staff.um.edu.mt/jskl1/simweb/mm1.htm

[7] J. Y. Wang, "Operation reasearch ii," pp. 17–9, Spring 2009.

[8] R. L. Fink, "M/d/1 waiting line," 2000, [Online; accessed 22-December-2014]. [Online]. Available: http://bradley.bradley.edu/~rf/wait-md1.htm