



A Priority Switch Mechanism to Improve SJF Scheduling Approach under Multiprocessor Environment

Deepak Kumar Mehra¹,
Student, GJU¹
Hisar, Haryana, India

Sunil Nandal²
Asstt. Professor, GJU²
Hisar, Haryana, India

Abstract: In a computer system, multiple processes are get executed by different users. It is the responsibility of the processor to execute these processes effectively and to avail the CPU time and resources to these processes effectively. The order of the execution of these processes is called the scheduling approach. There are number of scheduling approaches to achieve the effective throughput and maximum utilization of processes. Even though, there is the problem of starvation in such scheduling schemes. The presented work has defined an improved priority shift mechanism to reduce the waiting time and to avoid the starvation. The work includes the dynamic prioritization of processes as well as priority change mechanism. The work is compared with shortest job first scheduling approach and results shows that the presented work has improved the work under all parameters.

Keywords: Dynamic Prioritization, Resource Allocation, SJF, Maximizing Throughput

I. Introduction

Today most of the available computer system is able to provide the multi processor architecture to enhance the efficiency and throughput provided by the system. The multi processor system is capable to manage the system resources and to provide the efficient turnaround time over the processor system. These kind of systems are divided in two main categories called the tightly coupled multiprocessor system and the loosely coupled multiprocessor system. The tightly coupled multi processor system having the single memory and the queue for all the processors where as in case of loosely coupled, each processor have a separate queue and the memory to manage the processes[1][2]. The presented work is focused on a tightly coupled multiprocessor system. The multiprocessor system is having the single job queue to accept all the processes executed by the users. As a user, execute a process, the process enter to the job queue. It is the work of scheduler is to select a process from the job queue and avail it to the processor for execution as well as to load it in the memory. There are number of existing scheduling approaches to take out the processes from the queue. These approaches includes the FIFO(First In First Out), LIFO(Last In First Out), SJF(Shortest Job First). The framework provided by Processor Computing is basically to achieve the better utilization of resources as well as to improve the probabilistic efficiency of the processing device with better resource usage as well as better processing throughput. The easiest use of Processor computing would be to run an existing application on several machines. The work is defined for the multiprocessor system in which the effective queue management is done so that if the devices are busy, the execution of the task would be delayed[3][4].

Processor computing deals with all aspects related to the scheduling mechanism. The scheduling mechanism aspects includes the processor utilization, processor contribution and the time state analysis of different processes. The processor computing becomes more complex in case of heterogeneous systems, where the work is divided among the processors respective to the complexity of the task or type of the process. The processor computing is also having the responsibility to manage the resources to different processes in such way effective utilization of resources will be done and the situation of deadlock will never occur. The scheduling mechanism adapted by a processor system can be preemptive or non-preemptive. In case of preemptive scheduling approach, the process under execution can be interrupted by other process of higher priority. Whereas the non-preemptive processes are never interrupted by other processes. The processor computing is effective enough to take the decision regarding the allocation of the resources respective to scheduling or based on the requirement of the resources. In case of multi processor and multi processes systems, each resource also having a separate job queue to keep the processes in wait if the resource is not sharable[5][6].

The scheduling approach is defined along with each resource provider under three main perspectives called the capability of a resource, the job queue and the cost of the execution of a process. The capability is here been defined in terms of speed of the execution of the process either by the processor or by the IO controller. It also depends on the sharable property of the resource. Another aspect included here is the throughput ratio or the utilization ratio respective to the speed. Based on these parameters, the execution time of the process can be estimated. Another important perspective is the job queue. The size of the job queue and its basic characteristics play an important role while deciding the sequence of the scheduling. The queue can be a priority queue, double ended queue etc. The final aspect is the cost of the process execution; the cost of the process depends on the number of resources required to a process, number of CPU clocks required[8].

A. Processor Computing

A computer system can solve complex problems and even the scientific problems. These systems are capable to solve the problems in different real time application areas such as engineering, medicine, security system etc. Now the computer system is more capable than a human brain to solve these problems in adaptive time and to provide the more effective computational solution over these problems. The processor system is basically defined to achieve the high throughput over the computing devices. The main objective of work is to achieve the essential computing over the system with shared resources mechanism. These systems are basically the mainframe systems that provide the power of unity in the computing system. The computing resources along with the system include the resource utilization, enhancing the capabilities of the computing system. The computer system itself having the several components including the CPU, memories, network system etc.

B. Scheduling in Processor

In the traditional systems, a static model is been presented to manage the resources over the system. To manage these resources, a centralized controller is defined to manage the resources and job respectively. There are number of static and dynamic approaches to manage these jobs as well as the resources. The main objective of the scheduling is to defined the inter-relation between these resources and the processes so that effective resource allotment and the process execution will be done. The processor system is defined to compute the cost estimation on these devices and to handle the challenges in terms of resource management such as scalability, adaptability fault tolerance and the reliability.

II. Existing Work

Lot of work is already done in the area of resource allocation and the process scheduling. Some of the earlier work done in same area is presented here. In year 2006, Vikki Tang has defined a work to reduce the instruction scheduling under the dynamic compilers. Author defined a scheduling approach under the feedback analysis so that effective allocation will be done. The presented framework is defined to benefit the instruction scheduling under multi threaded server applications[1]. In year 2013, Lichen Weng has defined a work on multithreaded processor system to perform the dynamic modeling. The paper describe the design under three steps. At first, author convert a scheduling policy to dynamic to evaluate the runtime of pattern mapping. The another step is to define the regression model to achieve the scheduling policy to identify the changing behavior of the threading system. The main objective of author was to define a scalable heuristic approach for estimating the growth of the system count[2]. Hsiang-Yun Cheng is defined as an analytical model to achieve the task scheduling under the analytical modeling. Author estimated the potential aspects under the memory and bandwidth analysis to restrict the number of task. Author implemented the scheduling under the real hardware[3]. In year 2013, Vishakha gupta has performed the performance analysis for the functionality analysis under asymmetric platforms. Author has performed the analysis under the heterogeneity under the utility and applicability analysis. Author has defined the work under the workload analysis and defined it under different processes and different configuration for the resource analysis[4].

Morris A. Jette defined the characteristics analysis under the scheduling process for multi programmed environments. Author defined a time and space slicing mechanism for the parallel programming and defined the concurrent job execution under single processor environment. Author has defined a performance analysis system under the utilization and responsiveness under different computing platforms[5]. Another work for the heterogeneous scheduling policies for real time multi processor system is considered for the multimedia mapping for design space. Author has defined a suitable scheduling policy so that system energy can be minimized. The presented framework includes the analysis on energy reduction approaches for dynamic power management [6]. Another work on power management for multi-core architecture for the process scheduling is defined for the process estimation under platform evaluation. Author defined the effectiveness and scalability of the system. Author highlighted the scalability limitations for the thread scheduling algorithm for small scale multi processor system. Author has defined the scheduling overhead without loss of accuracy[7]. In Year 2005, Rony Ghattas presented some approach to improve the functionality of the micro processor system under the energy and power constraints. This system was defined under low bit system and to enhance the system performance. The main advantage of the system is to reduce the cost and complexity of this new micro processor system along with the reduction of power consumption[8].

In Year 2003, Andrei Terechko defined the scheduling under the high level language with some variable definition with global values. Author defined the long range and large impact schedule for the compiler optimization for local values under the scheduling units. The paper has defined three main algorithms for assigning the values to different cluster under the multi pass scheduling approach under the variable definition. Author also defined the performance measures for optimizing the algorithm[9]. In Year 2004, Andrew Riffel also defined a multi pass partitioning problem with recursive denominator split along with heuristic algorithm so that the robustness over the approach will be achieved. This paper redefines the MPP as a scheduling problem and uses scheduling algorithms that allow incremental resource estimation and pass computation in effective time[10]. Another work on improvement over the energy efficiency was presented by Hiroshi Sasaki. The proposed method groups several instructions as a single issue unit and reduces the required number of ports and the size of the structure for dispatch, wakeup, select, and issue. The present paper describes the micro architecture mechanisms and shows evaluation results for energy savings and performance[11]. Flavius Gruian presented an addresses scheduling approach for reduced energy of hard real-time tasks with fixed priorities assigned in a rate monotonic or deadline monotonic manner. The approach Author describe can be exclusively implemented in the RTOS. It targets energy consumption reduction by using both on-line and off-line decisions, taken both at task level and at task-set level[12].

Martin Schoeberl perform the investigation on the overhead analysis on object oriented operations. Author also presented the work so that the overhead over the system will be reduced as well as the dispatch and field access will be done effectively. Author presented this work for a real time embedded system. The main objective presented by the author to reduce the hardware cost and to optimize the application output[13]. In Year 2000, Jared Stark presented work on instruction scheduling for pipelined processing. Author defined the work to improve the pipelined scheduling. Author has defined the technique to eliminate the ability to improve the execution of dependent instruction under the consecutive cycles. The presented approach by the author has defined the frequency check with the sacrifice of IPC[14].

III. Proposed Work

To achieve the optimize utilization of processor and interrelated resources, it is required to provide a better scheduling mechanism. The main objective of the work is to achieve the maximum throughput from the processor and integrated components and to reduce the wait time. The presented work is focused on a multi processor system in which a process can require more than one processor and input output devices for the execution. Once a process is executed, it is loaded in a job queue. From this job queue, the process is taken out for the execution. If the process is an input output process, then the process is also loaded in the queue maintained by input output device. The work is about to reduce the waiting time in terms of processor execution either it is in process queue or it is in input output queue.

In this presented work, the main focus is to reduce the resource allocation wait time. The work is about to analyze the wait time of each process to get the access on the required input output device. Each process defined by the user is analyzed under the capabilities of the processor as well as the capabilities of input output devices. Number of access units or clocks are identified and analyzed to estimate the throughput and wait time of the processors. The objective of the work is to reduce the number of wait cycles so that overall reduction over the wait time will be done.

The another concept here is the prioritization. The initial priority assignment is here defined under the resource requirement. This requirement is defined based on the number of input output devices and the number of cpu required. Once the priority is decided, a limit threshold is applied to change the priority of the processes. The another aspect to change priority is defined based on the waiting threshold. If the wait time of a low priority process is increased by its threshold. The presented scheduling mechanism is given as under

ImprovedSJFScheduling()

```
{
  (i)    Generate N Number of Processes called P1,P2...PN.
  (ii)   For i=1 to N
  (iii)  {
  (iv)   Define the Parameters for Each Process P(i) called, Process Time, Arrival Time, IO Request and CPU
  Request
  (v)    }
  (vi)   Arrange the Processes in order of
  a.    Process Time
  b.    Arrival Time
  c.    CPU Requests
  (vii)  For i=1 to N
  (viii) {
  (ix)   If (Request(CPU, P(i))=High and Request(IO, P(i))=High)
  (x)    {
  (xi)   Set Priority(P(i)) => High
  (xii)  }
  (xiii) If (Request(CPU, P(i))=High or Request(IO, P(i))=High)
  (xiv)  {
  (xv)   Set Priority(P(i)) => Medium
  (xvi)  }
  (xvii) If (Request(CPU, P(i))=Medium or Request(IO, P(i))=Medium)
  (xviii) {
  (xix)  Set Priority(P(i)) => Low
  (xx)   }
  (xxi)  Else
  (xxii) {
  (xxiii) Set Priority(P(i)) => Very Low
  (xxiv) }
  (xxv)  }
  (xxvi) For i=1 to N
  (xxvii) {
  (xxviii) If (FrequencyCount(Priority(P(i))) > Threshold)
  (xxix)  {
  (xxx)   Set Priority(P(i))=Priority(P(i))+1
```

```

(xxxi) }
(xxxii) }
(xxxiii) For i=1 to N
(xxxiv) {
(xxxv) If (Wait time(P(i))> Threshold)
(xxxvi) {
(xxxvii) Priority(P(i))= Priority(P(i))+1
(xxxviii) }
(xxxix) }
(xl) Update the Scheduled Process according to Prioritized Sequence
(xli) Analyze the Processes Based of Wait time.
}

```

IV. Research Methodology

The presented work is implemented in java environment under a user friendly environment. To accept the user input in terms of number of processor and input-output requirements, a user friendly environment is been generated. The presented environment is implemented for the existing SJF algorithm as well as for the proposed algorithm. Based on the analysis, the results are taken from the driven work. The presented work is been executed for 5 processes and the results obtained from the system are presented in the form of graph shown in figure 1.

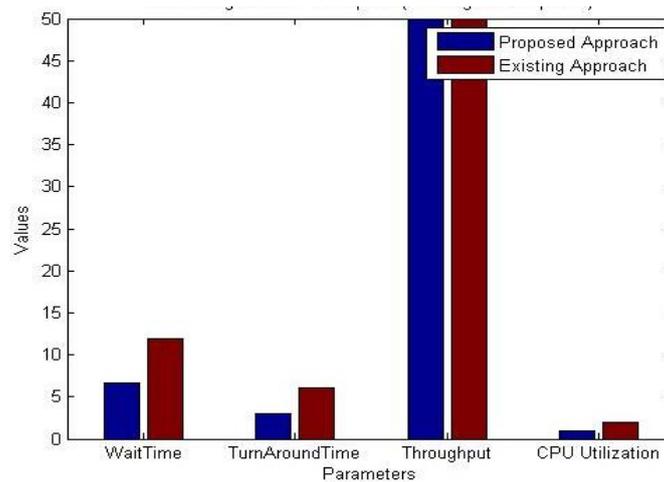


Figure 1 : Comparative Analysis (Proposed Vs. Existing)

As we can see, the presented work is analyzed with existing approach. The comparative analysis is shown in terms of wait time, turn around time, throughput and the CPU utilization. The proposed work is shown in the form of blue bars and existing work is presented in brown bars. The results shows that the presented work has reduced the wait time, turn around time of the process execution. More effective throughput and the CPU utilization is obtained from the work.

V. Conclusion

The presented work is about to improve the effectiveness of the Processor system along with reliability. In this system we will keep the most frequent data items in cache by estimating the data frequency. As the most required data items are kept in the cache itself, it will improve the hit ratio and improve the reliability of data access. The system will give the better service allotment such way that the starvation will not occur over the system.

References

- [1] Vikki Tang, " A Framework for Reducing Instruction Scheduling Overhead in Dynamic Compilers".
- [2] Lichen Weng, " Scheduling Optimization in Multicore Multithreaded Microprocessors through Dynamic Modeling", CF'13, May 14–16, 2013, Ischia, Italy. ACM 978-1-4503-2053-5
- [3] Hsiang-Yun Cheng, " An Analytical Model to Exploit Memory Task Scheduling", INTERACT-14, March 13, 2010, Pittsburgh, PA, USA ACM 978-1-60558-921-3/10/03
- [4] Vishakha Gupta, " Kinship: Efficient Resource Management for Performance and Functionally Asymmetric Platforms", CF'13, May 14–16, 2013, Ischia, Italy. ACM 978-1-4503-2053-5
- [5] Morris A. Jette, " Performance Characteristics of Gang Scheduling in Multiprogrammed Environments", 1997 ACM 0-89791-985-8/97/0011
- [6] Minyoung Kim, " Design Space Exploration of Real-time Multi-media MPSoCs with Heterogeneous Scheduling Policies", CODES+ISSS'06, October 22–25, 2006, Seoul, Korea. ACM 1-59593-370-0/06/0010
- [7] Jonathan A. Winter, " Scalable Thread Scheduling and Global Power Management for Heterogeneous Many-Core Architectures", PACT'10, September 11–15, 2010, Vienna, Austria. ACM 978-1-4503-0178-7/10/09

- [8] Rony Ghattas, " Energy Management for Commodity Short-Bit-Width Microcontrollers", CASES'05, September 24–27, 2005, San Francisco, California, USA. ACM 1-59593-149-X/05/0009
- [9] Andrei Terechko, " Cluster Assignment of Global Values for Clustered VLIW Processors", CASES'03, Oct. 30 – Nov. 1, 2003, San Jose, California, USA. ACM 1-58113-676-5/03/0010
- [10] Andrew Riffel, " Mio: Fast Multipass Partitioning via Priority-Based Instruction Scheduling".
- [11] Hiroshi Sasaki, "Energy-Efficient Dynamic Instruction Scheduling Logic through Instruction Grouping", ISLPED'06, October 4–6, 2006, Tegernsee, Germany. ACM 1-59593-462-6/06/0010
- [12] Flavius Gruian, " Hard Real-Time Scheduling for Low-Energy Using Stochastic Data and DVS Processors", ISLPED'01, August 6-7, 2001, Huntington Beach, California, USA. ACM 1-58113-371-5/01/0008
- [13] Martin Schoeberl, " Architecture for Object-Oriented Programming Languages", JTRES '07 September 26-28, 2007 Vienna, Austria ACM 978-59593-813-8/07/09
- [14] Jared Stark, " On Pipelining Dynamic Instruction Scheduling Logic", 0-7695-0924-X/2000© 2000 IEEE