International Journal of Electrical Sciences & Engineering (IJESE)
Online ISSN: 2455 6068; Volume 1, Issue 1; January 2016 pp. 1-5

Improved CPU Utilization using Advanced Fuzzy Based CPU Scheduling algorithm (AFCS)

Xiao-Zhi Gao[1], Sasikumar Gurumurthy[2], S. Venkatesan[3]

[1]Professor, Department of Electrical Engineering and Automation,
Aalto University School of Electrical Engineering, 00076 Aalto, Finland
xiao.z.gao@gmail.com
[2,]Associate Professor, Department of Computer Science and Engineering,
Dayananda Sagar College of Engineering, Bangalore, India
[3]Professor, Department of Computer Science and Engineering,
Dayananda Sagar College of Engineering, Bangalore, India
[2]mithrangurugsk@gmail.com; [3]selvamvenkatesan@gmail.com

Absract: **The operating system in our computer machines have changed a lot during the course of time, where in the initial stage of their development they were used to process a single task (process) at a time but now, in the era of supercomputers we have multiprogramming operating system running in our machines. At present we have a number of scheduling algorithms which are used to decide the order in which the processes loaded into the memory are to be executed. But none of the conventional scheduling algorithms is ideal, they have their own drawbacks. In this paper, an advanced fuzzy-based logic has been proposed for soft real time system toovercome the drawbacks of other algorithms for better CPU utilization and to minimize waiting, turn-around and response time. The proposed algorithm is preemptive in nature with minimum context switching and work to complete process within its deadline.**

Keywords: Fuzzy logic, CPU scheduling, soft real time system deadline, preemptive process deadline, Dynamic priority

1. INTRODUCTION

As of now scheduling real time system involve allocation of resources and CPU time to task in such a way that certain performance requirements are happened. In real-time system scheduling has played a more acute role than non-real time system because in this system having the right answer too late is as bad as not having it at all[10].

Such a system reacts to the request within a fixed amount of time which is called deadline. In general, real time system can be categorized into two important groups: Hard real time system and Soft real time system. In hard real time systems, when task occurs it strictly completed at a given deadlines. While in soft real time system missing some deadlines is acceptable. In both cases, the scheduler is to be schedule in such a way that guarantees the deadline to meet when a new task is arrived.

Scheduling algorithm is necessary and important task when more than one jobsare present in ready queue. Criteria forchoosing best scheduling is depend upon following basic featuressuch as:

- Waiting Time
- Turnaround Time
- Response Time
- Utilization of CPU
- Throughput

There are various type of scheduling algorithms such as first come first serve, priority based scheduling, shortest job first etc. The main constrain of real time task is that it should be completed within deadline time. The above scheduling algorithms are inefficient for real time operating system task. Hence we have proposed a new scheduling algorithm to find out the dynamic priority of process using fuzzy logic.

2. RELATED WORK

New era of possibilities were open when Lotfi A. Zadeh introduced the term “fuzzy logic” with proposal of fuzzy set theory. To make concept of approximation[1] reality this fuzzy logic can be used. Process scheduling with fuzzy logic has also been thought by many researchers[2][3][4]. Scheduling with deadline concept is basic requirement for real time system [10]. This schedule can be preemptive or non preemptive. Soft real time system with optimal time slice and dead line [5] is considered in this paper.

3. SCHEDULING ALGORITHMS

3. 1 FCFS Scheduling Algorithm

Even with all evolution in scheduling algorithm the FCFS serves as base algorithm. It is as simple as it sounds. The task

is executed as it comes to ready queue in arrival time order. There are some disadvantages of FCFS such as follows:

a. This does not support preemption.

b. Throughput decreases as CPU holding time of a task increases.

c. There is no concept of priority. Turnaround time, waiting time and response time is very high which can reduces the performance.

3. 2 Priority Based Scheduling Algorithm

In this algorithm priority is associated to each process and depends upon the highest priority the process is assigned to the CPU. If process has equal priority then it scheduled in FCFS. We know that priority is assigned by operating system. The disadvantages of this algorithm are as follows

a. The major disadvantage of this algorithm is indefinite blocking it also called as starvation. We know that Low priority process gets interrupted by highest priority process. But if there is large number of highest priority process are present then each time it interrupted to low priority process then starvation occurred.

b. Another disadvantage is that the waiting time and turnaround time depend upon the priority of process.

3. 3 Shortest Job First Scheduling Algorithm

In this scheduling algorithm we are select the process with smallest burst time to execute the process. This is one of the best scheduling algorithm in which we get minimum waiting and turnaround time as compare to other scheduling algorithm. But there are some disadvantages of this algorithm are as follows:

a. It is very difficult to know the burst time for next CPU request.

b. Again this algorithm is not implemented for the shortest level CPU scheduling.

c. One major drawback is that process starvation for the process whose burst time is long if smallest burst time process is continuously arrived.

4. FUZZY LOGIC

Fuzzy logic is the superset of Boolean logic which deals with the truth values that is 0's and 1's. It is the nonlinear mapping form input data to the output data. The fuzzy logic system first collect the crisp set of inputs and convert it's to the fuzzy set using fuzzy linguistic variable, terms and membership function, this process is called as Fuzzification. This fuzzy set is use for making inference. Finally, we used the defuzzification step in which the resulting output is mapped with crisp output using membership function.

There are twokinds of Fuzzy Inference System such as (i) Mamdani's fuzzy inference method and (ii) Sugeno fuzzy inference method.

5. PROCESS DEADLINE

In any of the real time system the tasks are assigned some deadline, failure to meet the deadline is not tolerable in hard real time system but the soft real time system does not lead to system failure only performance degradation happens. In this paper an algorithm is proposed to avoid process starvation with deadline concept using some optimal time slice to execute process. Preemptive process deadline is used to denote the maximum time till which the process can be preempted.

Proposed Algorithm

1. Check weather new process is arrived than add to ready queue else continue

2. While (ready queue != NULL)

3. Set dynamic priority to output FIS

Calculate dynamic priority (DPi):-

1. For each process Pi in ready queue fetch its parameters burst time (BTi), static priority (PTi), and arrival time (ATi) and give them as input to FIS.

2. For each process (Pi),
   Evaluate membership function of priority (μp)
   μp=PTi/(max (PTi) +1); where 1<=i<=n

3. For each process (Pi),
   Evaluate membership function of burst time (μb)
   μb=1-(BTi/(max (BTi) +1)); where 1<=i<=n

4. For each process (Pi) in ready queue
   find minimum priority process.
   To calculate dynamic priority (DPi)

5. If process Pi has minimum priority then
   DPi= (μp+μb)
   Else
   DPi= max {μp, μb}
   where 1<=i<=n

4. Calculate optimal time slice (OTS) only once for each process

X=half of the highest burst time in ready queue (upper bound)
Y=average burst time in ready queue (consider upper bound)
Z=highest burst time-(OTS of 1stprocess in queue)
calculate Zevery time new process gets in ready queue

For 1st process in ready queue:
If(X <= Y)
OTS(Pi) =X [i=priority no 1 to 0]
Else
OTS(Pi) =Y [i=priority no 1 to 0]

From 2nd process in ready queue:
OTS(Pi) = Z [i=priority no 1 to 0]

5. Calculate deadline for each process in ready queue only once
For the process with highest priority:
[i =highest priority]
[i-1 =second highest priority]
[D(Pi) =deadline of process Pi]
[BT(Pi) =burst time of process Pi]
[RBT(Pi-1) = remaining burst time of next process in ready queue]
[AT(Pi) =arrival time of process Pi]

D(Pi)= AT(Pi) +BT(Pi)+RBT(Pi-1)
if(BT(Pi)<=OTS(Pi))
complete total task
else
switch on OTS(Pi)
Gotostep(7)
For next process except last process:
[SPBT(Pi)= sum of previous burst time completion in CPU of process Pi]
[PPD(Pi)=preemptive process deadline of process Pi+1]

D(Pi)=AT(Pi) + D(Pi+1)+RBT(Pi-1)

If(BT(Pi)<=OTS(Pi)&&(SPBT(Pi)+BT(Pi)<=PPD(Pi-1))
complete task
Else
switch on PPD(Pi)
Otherwise
switch on OTS(Pi)
Gotostep(7)

For last process:
D(Pi)=D(Pi-1)
If(BT(Pi)<=OTS(Pi)&&(SPBT(Pi)+BT(Pi)<=PPD(Pi-1))
complete task
Else
switch on PPD(Pi)
Otherwise
switch on OTS(Pi)
Gotostep(7)

6. Calculate remaining burst time(RBT):
[RBT(Pi)=remaining burst time of process Pi, i denote priority no 1 to 0]
[EBT(Pi)=total execution burst time of Pi, i denote priority no 1 to 0]
RBT(Pi)=BT(Pi)-EBT(Pi)
Gotostep(7)

7. Calculate deadline of preemptive process for every process with priority lower than equal to latest executed process :
for arrival in CPU:
PPD(Pi)=D(Pi)-RBT(Pi)
If(PPD(Pi-1)<=PPD(Pi)) ||(PPD(Pi-1)<D(Pi))
Then
PPD(Pi)=PPD(Pi-1)-RBT(Pi)
Gotostep(1)

8. Removing process from ready queue
a) Remove from queue when next lower process completes
2) Else remove if no lower priority process

9. If new process coming then Goto step (1).

6. RESULT AND PERFORMANCE EVALUATION

To demonstrate proposed algorithm some case studies have been considered with comparison to other algorithms on same cases. The results are denoted in terms of Gant chart and some statistical representation.

Case Study 1:

TABLE 1: Case Study 1 Data Set

Process ID	Arrival Time (ATi)	Burst Time (BTi)	Static Priority (PTi)	Dynamic Priority (DPi)	Deadline (D)
P1	0	3	2	0. 25	3
P2	2	6	7	0. 875	9
P3	4	4	5	0. 43	17
P4	6	5	6	0. 62	17
P5	8	2	1	0. 72	22

Gant Chart for priority Scheduling

PID	P1	P2	P4	P3	P5
0	3	9	14	18	20

Gant Chart for Improved Fuzzy based CPU Scheduling

PID	P1	P2	P5	P4	P3
0	3	9	11	16	20

Gant Chart for Advanced Fuzzy based CPU Scheduling

PID	P1	P1	P2	P2	P5	P4	P4	P3
0	2	3	7	9	11	13	16	20

RBT	1	0	2	0	0	3	0	0
PPD	2	3	7	9	22	10	13	13

Comparison Table

TABLE 3:Comparison between various algorithms for case study1

Algorithm	Average Waiting Time	Average Turnaround Time	Average Response Time
Priority Algorithm	4. 8	8. 8	4. 8
IFCS	3. 8	7. 8	3. 8
AFCS	3. 8	7. 8	3. 8

Case Study 2:

TABLE 4: Case Study 2 Data Set

Process ID	Arrival Time (ATi)	Burst Time (BTi)	Static Priority (PTi)	Dynamic Priority (DPi)	Deadline (D)
P1	0	18	1	0. 136	53
P2	0	2	3	0. 894	18
P3	0	1	2	0. 95	13
P4	0	4	6	0. 79	35
P5	0	3	5	0. 84	22
P6	0	12	11	0. 917	15
P7	0	13	7	0. 58	53

Gant Chart for priority Scheduling

PID	P6	P7	P4	P5	P2	P3	P1
0	12	25	29	32	34	35	53

Gant Chart for Improved Fuzzy based CPU Scheduling

PID	P3	P6	P2	P5	P4	P7	P1
0	1	13	15	18	22	35	53

Gant Chart for Advanced CPU Scheduling

PID	P3	P6	P2	P6	P5	P4	P7	P7	P1	P1
0	1	11	13	15	18	22	32	35	45	53

TABLE 5: Gantt chart for case study 2

RBT	0	2	0	0	0	0	3	0	8	0
PPD	3	13	18	13	22	35	32	35	45	53

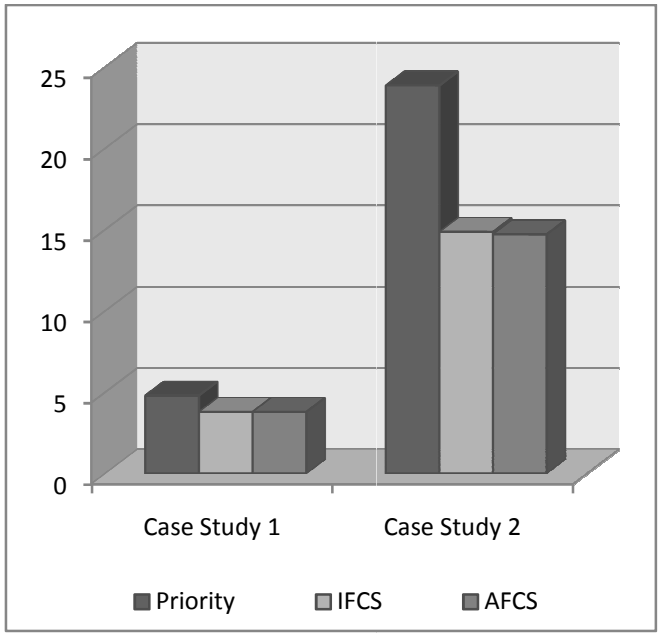
Comparison Table

TABLE 6: Comparison between various algorithms for case study

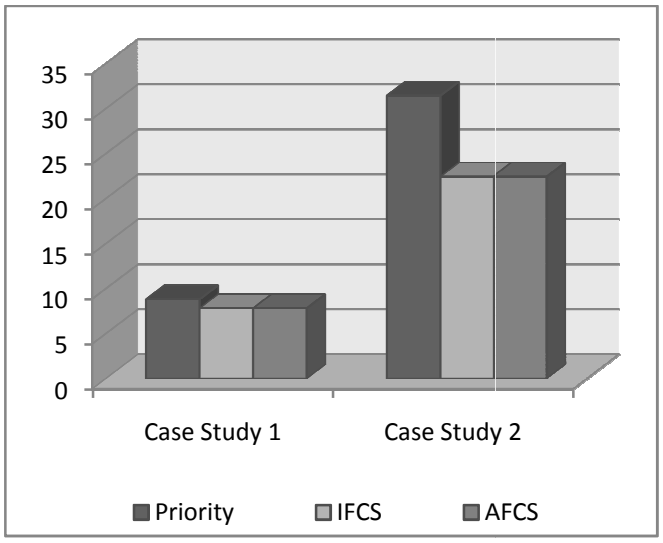
Algorithm	Average Waiting Time	Average Turnaround Time	Average Response Time
Priority Algorithm	23. 86	31. 43	23. 86
IFCS	14. 86	22. 43	14. 85
AFCS	14. 71	22. 43	14. 57

Statistical analysis of the proposed and existing algorithm

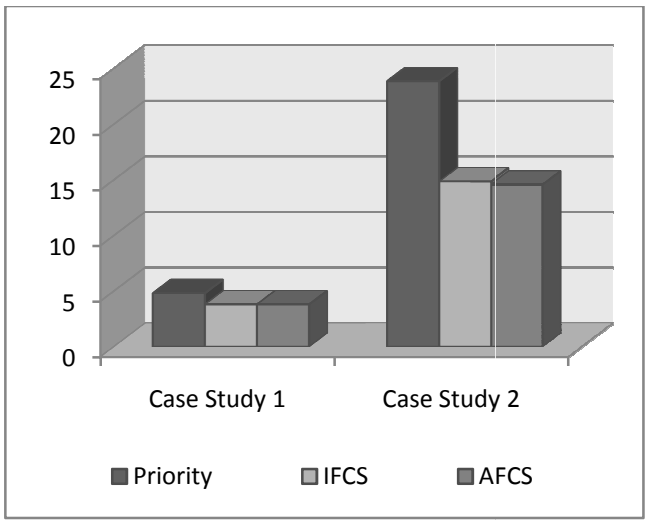
1. Waiting time vs. No. of Process



Turnaround Time vs. No. of process



3. Response Time vs. No. of process



7. CONCLUSION AND FUTURE WORK

The proposed algorithm reduces the response time and waiting time by minimal difference but with different case studies the difference of time may occur. This algorithm completes the given processes within deadline. Preemption of processes occurs. The time slice value is kept optimal to minimize context switches and increase the response time of processes. This algorithm can be further improved by choosing good membership function in fuzzyfication process. The time slice value can also be calculated with different way of thinking to further reduce the context switches.

REFERENCES

[1] Sankar K. Pal and Deba Prasad Mandal," fuzzy logic and approximate reasoning: an overview", Electronics and Communication Science Unit, Indian statistical institute, Calcutta 700 035, India

[2] Shatha J. Kadhim and Kasim M. Al-Aubidy," Design and Evaluation of a Fuzzy-Based CPScheduling Algorithm", Springer-Verlag Berlin Heidelberg 2010

[3] H. S. Behera, RatikantaPattanayak, PriyabrataMallick,"An Improved Fuzzy-Based CPU Scheduling (IFCS) Algorithm for Real Time Systems", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012

[4] PrernaAjmaniand ManojSethi," Proposed Fuzzy CPU Scheduling Algorithm (PFCS) for Real Time Operating Systems", BIJIT - BVICAM's International Journal of Information Technology BharatiVidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi (INDIA)

[5] Vikash Chandra Sharma, Priyadarshini, Ajay Chaudhary, Manindarsinghnehra," CPU Scheduling Algorithm with Deadline and Optimize Time Slice for soft real time systems", international journal of enhanced research in management & computer applications

[6] RajaniKumari, Vivek Kumar Sharma, Sandeep Kumar," Design and Implementation of Modified Fuzzy basedCPU Scheduling Algorithm", International Journal of Computer Applications (0975 – 8887) Volume 77 – No. 17, September 2013

[7] M. M. M. Fahmy," A fuzzy algorithm for scheduling non-periodic jobs on soft real-time single processor system", Ain Shams Engineering Journal (2010)

[8] Patricia Balbastre, Ismael Ripoll and Alfons Crespo," Minimum deadline calculation for periodic real-time tasksin dynamic priority systems", Spanish Government Research Office (CICYT)

[9] Mamdani E. H., Assilian S, "An experiment in linguistic synthesis with a fuzzy logic controller", *InternationalJournal of Man-Machine Studies*, Vol. 7, No. 1, 1975.

Book References:

[10] William Stallings : Operating Systems 5th Edition. Pearson Education India, 2006.