Weighted Clustering Based Preemptive Scheduling For Real Time System

H.S Behera
Department of Computer Science & Engineering
Veer Surendra Sai University of Technology(VSSUT), Burla
Sambalpur, Orissa, India

Madhusmita Mishra
Department of Computer Science & Engineering
Veer Surendra Sai University of Technology(VSSUT), Burla
Sambalpur, Orissa, India

Simpi Patel
Department of Computer Science & Engineering
Veer Surendra Sai University of Technology(VSSUT), Burla
Sambalpur, Orissa, India

Abstract- **In this paper a new improved clustering based scheduling algorithm for a single processor environment is proposed. In the proposed method, processes are organized into non-overlapping clusters. For each process the variance from the median, is calculated and compared with the variance from the means of other clusters. Each process is assigned to the cluster associated with the closest median. The new median of each cluster is calculated and the procedure is repeated until the medians are fixed. Weight is assigned to each cluster using the externally assigned priorities and the burst time. The cluster with highest weight is executed first and jobs are scheduled using the Round Robin algorithm with calculated dynamic time slice.. The experimental study of the proposed scheduling algorithm shows that the high priority jobs can be executed first to meet the deadlines and also prevents starvation of processes at the same time which is crucial in a real time system.**

Keywords: K-Median Clustering, Service Time, Weight, Round Robin, Variance

I. INTRODUCTION

Scheduling is a key concept in operating system design. In a multiprogramming environment, the processes that are loaded into the memory compete for processor time [1].Scheduling determines which process will progress and which will wait. The main objectives that scheduling function must satisfy include fairness, efficient use of processor time, response time, turnaround and throughput Operating systems may be feature up to three distinct types of scheduling: long-term scheduling, medium-term scheduling and short-term scheduling. CPU scheduling is an essential operating system task; therefore its scheduling is central to operating system design. When there is more than one process in the ready queue waiting its turn to be assigned to the CPU, the operating system must decide through the scheduler, the order of execution. Many scheduling algorithms have been proposed for processor assignment, but there is no suitable algorithm for all purposes. So special care is needed to choose and coordinate parameters, e.g. waiting time, total time and utilization. Some of the commonly used algorithms are First Come First Serve(FCFS), Round Robin (RR), Shortest Job First (SJF), Shortest Remaining Time (SRT), High Response Ratio Next (HRRN) and Feed Back (FB) algorithm[3]. Non-preemptive algorithms like FCFS and SJF are not suitable for real time system, but preemptive scheduling like MLFQ and RR are used to provide good response time and fair dispatching of CPU time e.g. in interactive systems[3]. Clustering of processes is nothing but grouping of the processes into classes or clusters ,so that processes within a cluster have high similarity in comparison to one another, but are very dissimilar to processes in other clusters. Cluster analysis is an exploratory data analysis tool which aims at sorting different processes into groups in a way that the degree of association between two processes is maximal if they belong to the same group and minimal otherwise. To achieve different application purposes, a large number of clustering algorithms have been developed. Clustering techniques can be broadly divided into five classes: hierarchical methods, partitioning methods, density-based methods, grid based method, model based method Partitioning clustering algorithms, such as K-means, K-medoids, PAM which assign objects into k (predefined cluster number) clusters, and iteratively reallocate objects to improve the quality of clustering results [8]. Hierarchical clustering algorithms assign objects in tree-structured clusters, i.e., a cluster can have data points or representatives of low level clusters. Hierarchical clustering algorithms can be classified into categories according their clustering process: agglomerative and divisive [9].Density-based clustering is that for each instances of a cluster the neighborhood of a given radious has to contain at least a minimum number of instances. Grid-based clustering first quantize the clustering space into a finite number of cells.cells that contain more than certain number of points are treared as dense and the dense cells are connected to form the clusters. In model based method in addition to the observed or predictive attributes,there is a hidden variable which reflects the cluster membership for every case in the data set. One simple rule of thumb sets the number to $k \approx \sqrt{n/2}$ with *n* as the number of objects (data points).

In statistics and machine learning, ***k*-medians clustering** is a variation of *k*-means clustering where instead of calculating the mean for each cluster to determine its centroid, one instead calculates the median. This has the effect of minimizing error over all clusters with respect to the 1-norm distance metric, as opposed to the square of the 2-norm distance metric (which *k*-means does).This relates directly to the ***k*-median problem** which is the problem of finding *k* centers such that the clusters formed by them are the most compact. Formally, given a set of data points *x*, the *k* centers c_i are to be chosen so as to minimize the sum of the distances from each *x* to the nearest c_i. The criterion function formulated in this way is sometimes a better criterion than that used in the *k*-means clustering algorithm, in which the sum of the squared distances is used.

II. RELATED WORKS

Yaashuwanth C & R.Ramesh[5]proposed an architecture which eliminates the defects of implementing a simple round robin architecture in real time operating system by calculating dynamic time slice which depends on three aspects i.e, range, total no of processes in the system, priority of the process & total priority assigned in the system. Yaashuwanth C & R.Ramesh[6] proposed a new method to schedule the processes using scheduling component which is calculated by the help of task period component,task deadline component & priority component. Rami J. Matarneh[4] has proposed an modified round robin where time quantum is adjusted after each iteration by calculating the median of remaining burst time. Abdurazzag Ali Aburas, Vladimir Miho[3] proposed an non-preemtive dynamic priority based scheduling using fuzzy logic.

III. MATERIALS AND METHODS

A. CLUSTERING

The processes are clustered randomly into *k* clusters using "The rule of the Thumb" i.e. $k \approx \sqrt{n/2}$ with *n* as total number of processes. The *k*- median clustering algorithm can be used to cluster the processes iteratively into dense clusters suitable for scheduling .The median is obtained by arranging the processes in increasing order of their burst time and choosing the burst time of the middle process. If the number of processes is even the middle two processes are chosen and the mean of the burst time of the two processes is taken as the median burst time [4].

$$M = \begin{cases} Y_{(n+1)/2}, & \text{if n is odd} \\ 1/2\,[Y_{(n/2)} + Y_{(1+n/2)}], & \text{if n is even} \end{cases}$$

M=median
Y=number located in the middle of a group of numbers arranged in ascending order
n=number of processes

Thus *k*-medians are obtained. The variance of the burst time of each process from the k-medians is calculated and the process is assigned to the cluster with minimum variance.

$$Var(X_i)=E[(X_i-M_{cj})^2]$$

X=Burst time of the i^{th} process
M_{cj}=Median of j^{th} cluster

E represents expected value

This process is carried out iteratively till the medians become constant, obtaining dense clusters fit for scheduling. The weight for each cluster can be found on the basis of the priorities, burst time and waiting time of the processes in a cluster [2].Initially the WT for all the clusters is 0.

$$W_{cj} = \frac{WT + \sum P_m}{\sum BT_m}$$

WT=Waiting time of each cluster
W_{cj}=Weight of the j^{th} cluster
$\sum P_m$=Sum of User Assigned Priorities of m Processes in j^{th} cluster
$\sum BT_m$=Sum of burst time of m processes in j^{th} cluster

B. SCHEDULING

The heaviest cluster is chosen to be serviced first. The processes within the cluster are scheduled using Round Robin algorithm based on the Response Ratio [3] and Service Time of each process.

$$ST = P \times BT$$

$$RR = \frac{WT + ST}{ST}$$

ST=Service time of each process
P=User assigned Priority of each process
RR=Response Ratio of each process
BT=Burst Time of each process

The process with highest Response Ratio is chosen to be executed first in order to prevent starvation due to ageing. The Round Robin algorithm with dynamic time quantum is then followed to schedule the processes within the cluster.

C. PSEUDOCODE OF PROPOSED ALGORITHM

1) Divide n processes into k clusters randomly
   $k \approx \sqrt{n}/2$
2) for (j=1;j <=k ; j++)
   {
3) Arrange the processes in each cluster in increasing order of burst time
4) Obtain median of each cluster
5) $$M = \begin{cases} Y_{(n+1)/2}, & \text{if n is odd} \\ 1/2\,[Y_{(n/2)} + Y_{(1+n/2)}], & \text{if n is even} \end{cases}$$
6) $a[j]=M_j$
   }
7) While $a[j] \neq a[j+1]$
   {

8) for (i=1 to n)
{

Calculate variance of BT of i^{th} process from the median of each cluster

$Var(X_{ij}) = E[(X_i - M_{cj})^2]$

9) if $Var(X_{ij}) > Var(X_{ij+1})$
{
$X_{ij}=X_{ij}+1$// Send X_{ij} to $(j+1)^{th}$ cluster//
j=j+1// Set cluster number to j+1//
}
}

10)GOTO step 2

11)Find the weight of each cluster

$$W_{cj} = \frac{WT+\sum P_m}{\sum BT_m}$$

12) While ready queue ≠NULL

{

13) Choose the heaviest cluster

14)Choose the process with highest RR within the cluster

ST=P×BT

$$RR = \frac{WT+ST}{ST}$$

15) Use the Round Robin algorithm to schedule the Processes

Calculate Dynamic Time Slice for each process

$$DTS = \frac{(R)\times(N)}{(Pr)\times(P)}$$

N=Total no of processes in the cluster

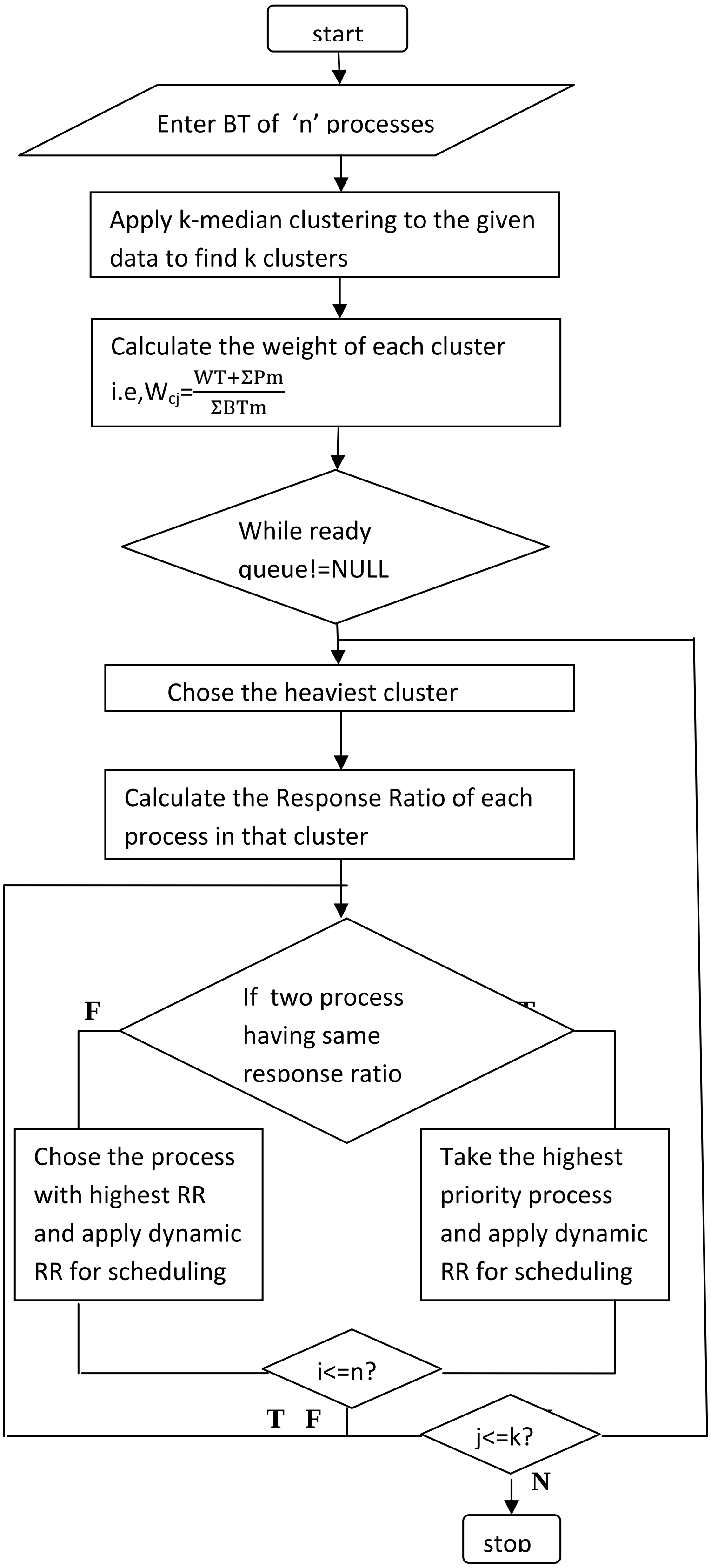
$$R = \frac{(\text{Max BT in the } j^{th} \text{ Cluster+Min Burst Time the } j^{th} \text{ cluster})}{2}$$

16) GOTO Step 12

}

17) End

D. FLOWCHART OF PROPOSED ALGOTITHM



IV. RESULTS AND DISCUSSIONS

The following experiments were conducted on processes with burst time chosen randomly between 1 to 100 in MATLAB. The unit of burst time is taken to be milliseconds.The pesudo-code is valid for any number of processes, however for simplicity,we have taken 50 processes.The priorities are assigned by the user which is considered external priority of a process.This external priority is particularly useful in an envionment where certain important tasks need to be executed first and thus are assigned higher priorities.

We experimented with 50 processes and using of the “rule of the thumb”, the number of clusters is found to be k=5.After applying k-medians clustering the lowest burst time processes form the highest weight cluster and highest burst time processes form the lowest weight cluster according to the weight calculation formula. The highest weight cluster is taken for scheduling first then the next highest cluster.This process continues until all the processes are scheduled with the help of round robin by calculating dynamic time slice.

Table 1: Processes, their Burst Time and Priority

PROCESSES	BT	PRIORITY
P1	5	1
P2	85	8
P3	27	2
P4	36	9
P5	7	3
P6	9	3
P7	77	2
P8	100	6
P9	3	5
P10	20	7
P11	10	2
P12	61	2
P13	14	6
P14	69	1
P15	24	4
P16	56	8
P17	89	8
P18	4	2
P19	6	5
P20	72	7
P21	1	3
P22	78	2
P23	11	6
P24	34	5
P25	29	7
P26	13	2
P27	66	9
P28	88	9
P29	68	1
P30	24	4
P31	47	7
P32	96	8
P33	67	2
P34	94	9
P35	74	5
P36	38	3
P37	8	1
P38	89	7
P39	22	4
P40	33	3
P41	12	2

P42	90	2
P43	78	6
P44	52	1
P45	91	4
P46	20	1
P47	18	8
P48	51	2
P49	46	9
P50	73	3

Table 2: Clustering Of Processes

CLUSTERS	MEDIAN	BURST TIME
Cluster 1	4	1,3,4,5,6,7
Cluster 2	78	66,67,68,69,72,73, 74,77,78,78,85,88, 89,89,90,91,94,96,100
Cluster 3	24	18,20,20,22,24,24, 27,29,33,34,36
Cluster 4	51	38,46,47,51,52,56,61
Cluster 5	11	8,9,10,11,12,13,14

Table 3:Calculation of Dynamic Time Slice

PROCESS	BT	PRIORITY	ST	DTS
P1	5	1	5	4
P2	85	8	680	11
P3	27	2	54	14
P4	36	9	324	3
P5	7	3	21	2
P6	9	3	27	4
P7	77	2	154	42
P8	100	6	600	14
P9	3	5	15	1
P10	20	7	140	4
P11	10	2	20	1
P12	61	2	122	25
P13	14	6	84	2
P14	69	1	69	83
P15	24	4	96	7

P16	56	8	448	7
P17	89	8	712	11
P18	4	2	8	2
P19	6	5	30	1
P20	72	7	504	12
P21	1	3	3	2
P22	78	2	156	42
P23	11	6	66	2
P24	34	5	170	6
P25	29	7	203	4
P26	13	2	26	6
P27	66	9	594	10
P28	88	9	792	10
P29	68	1	68	83
P30	24	4	96	28
P31	47	7	329	8
P32	96	8	768	11
P33	67	2	134	42
P34	94	9	846	10
P35	74	5	370	17
P36	38	3	114	17
P37	8	1	8	11
P38	89	7	623	12
P39	22	4	88	7
P40	33	3	99	9
P41	12	2	24	6
P42	90	2	180	42
P43	78	6	468	14
P44	52	1	52	50
P45	91	4	364	21
P46	20	1	20	27
P47	18	8	144	4
P48	51	2	102	25
P49	46	9	414	9
P50	73	3	219	28

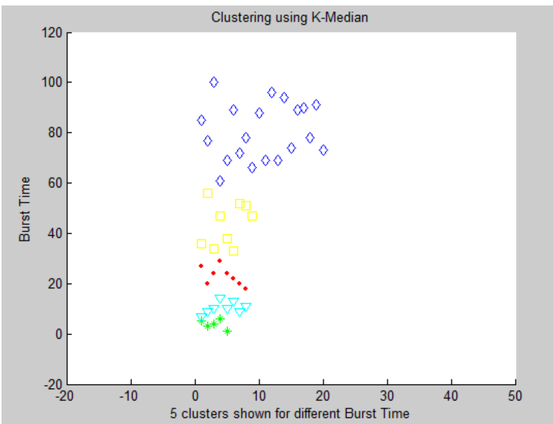


Fig 1: Processes clustered after 1st round of iteration

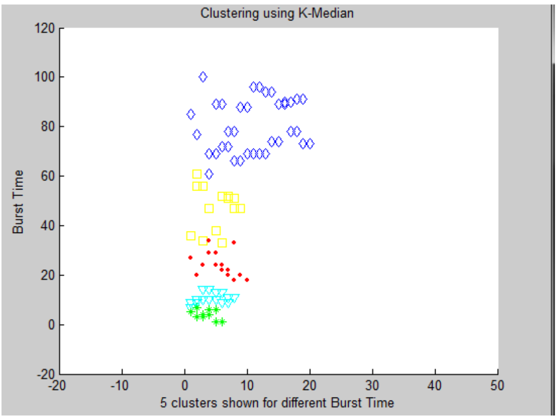


Fig 2: Processes clustered after 2nd round of iteration

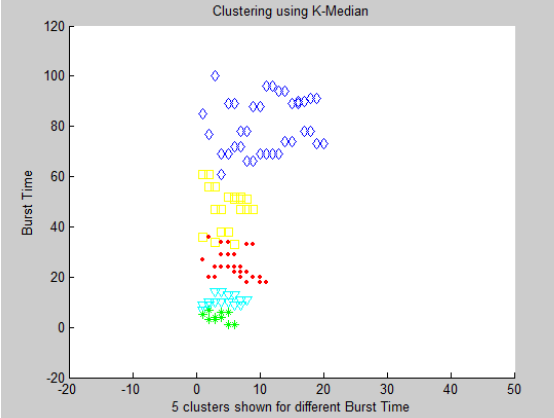


Fig 3: Processes clustered after 3rd round of iteration

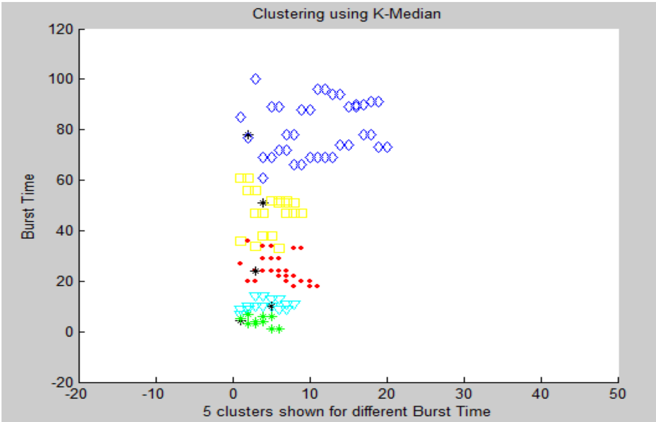


Fig 4: Processes clustered after 4th and final round of iteration

Table 1 represents the number of processes, burst time of processes and user assigned priority of each process. Table 2 represents the final clusters and medians. Table 3 represents scheduling of processes using Round Robin algorithm with the Dynamic Time Slice , service time and response ratio being the deciding factor for choosing the next process for execution. The time slice is allocated using the formula given in the pseudo-code.

Fig 1 shows the clustering of processes after 1st round of iteration. Fig 2 shows clustering after 2nd round of iteration. Fig 3 shows clustering after 3rd round of iteration and Fig 4 show the final clustering of processes after 4th round of iteration.

V. EXPERIMENTAL ANALYSIS

P1	P5	P21	P18	P9	P19	P1	P5	P18	P9	P19	P5	P9	P19	P5	P19	P19

0 4 6 7 9 10 11 12 14 16 17 18 20 21 22 23 24 25

P19

26

Fig 5:Gantt Chart For Cluster 1

P37	P11	P41	P26	P6	P23	P13	P11	P41	P26	P6	P23	P13	P26	P6

0 8 14 20 26 30 32 34 38 44 50 54 56 58 59 60 62

P23	P13	P23	P13	P23	P13	P23	P13	P13

64 66 68 70 72 73 75 77

Fig 6: Gantt Chart For Cluster 5

P46	P3	P40	P39	P15	P30	P24	P10	P25	P47	P4	P3	P40	P39	P15

0 20 34 43 50 57 64 70 74 78 82 85 98 107 114 121

P30	P24	P10	P25	P47	P4	P40	P39	P15	P30	P24	P10	P25	P47	P4

128 134 138 142 146 149 158 165 172 179 185 189 193 197 200

P40	P39	P15	P30	P24	P10	P25	P47	P4	P24	P10	P25	P47	P4	P24

206 207 210 213 219 223 227 231 234 240 244 248 250 253 259

P25	P4	P24	P25	P4	P25	P4	P4	P4	P4	P4

263 266 270 274 277 278 281 284 287 290 293

Fig 7: Gantt Chart For Cluster 3

P44	P48	P12	P36	P31	P16	P49	P44	P48	P12	P36	P31	P16	P49	P48

0 50 75 100 117 125 132 141 143 168 193 210 218 225 234 235

P12	P36	P31	P16	P49	P31	P16	P49	P31	P16	P49	P31	P16	P49	P16

246 250 258 265 274 282 289 298 306 313 322 329 336 337 344

P16	P16

351 358

Fig 8: Gantt Chart For Cluster 4

P29	P14	P33	P7	P22	P42	P50	P45	P35	P43	P8	P20	P38	P2	P17	P32

0 68 137 179 221 263 305 333 354 371 385 399 411 423 434 445 456

P27	P28	P34	P33	P7	P22	P42	P50	P45	P35	P43	P8	P20	P38	P2	P17

466 476 486 511 546 582 624 652 673 690 704 718 730 742 753 764

P32	P27	P28	P34	P42	P50	P45	P35	P43	P8	P20	P38	P2	P17	P32	P27

775 785 795 805 811 828 849 866 870 884 896 908 919 930 941 951

P28	P34	P45	P35	P43	P8	P20	P38	P2	P17	P32	P27	P28	P34	P45	P35

961 971 992 1009 1023 1037 1049 1061 1072 1083 1094 1104 1114 1124 1131 1137

P43	P8	P20	P38	P2	P17	P32	P27	P28	P34	P43	P8	P20	P38	P2	P17

1151 1165 1177 1189 1200 1211 1222 1232 1242 1252 1260 1274 1286 1298 1309 1320

P32	P27	P28	P34	P8	P38	P2	P17	P32	P27	P28	P34	P8	P38	P2	P17

1331 1341 1351 1361 1375 1387 1398 1409 1420 1426 1436 1446 1448 1453 1461 1472

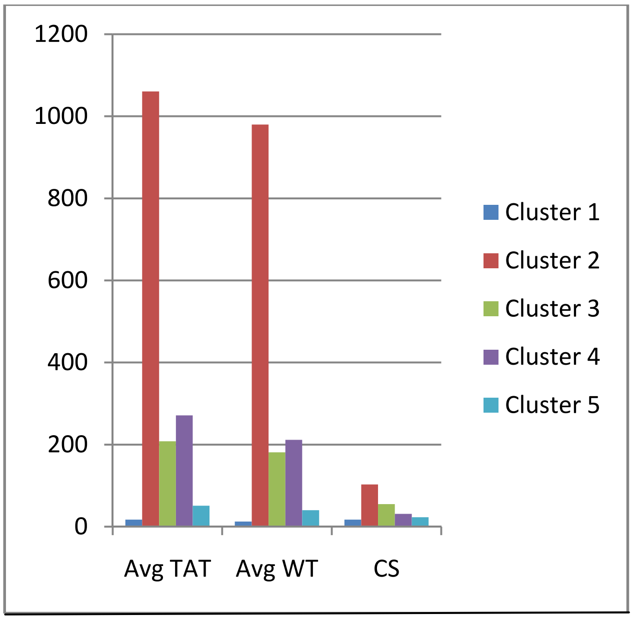
P32	P28	P34	P17	P32	P28	P34	P34

1483 1493 1503 1504 1512 1520 1530 1534

Fig 9:Gantt Chart For Cluster 2

Table 4: Calculation of TAT,WT & CS of each cluster
Comparision of AvgTAT, AvgWT and CS of 5 clusters

	Avg TAT	**Avg WT**	**CS**
Cluster 1	17	12.16	17
Cluster 2	1060.7	980.05	103
Cluster 3	208.09	181.18	55
Cluster 4	271	211.7	31
Cluster 5	51.2	40.2	23



VI. CONCLUSION

The experimental result shows that clustering of similar processes into dense clusters providing a better and easy scheduling of processes taking into account external priority and using Dynamic Round Robin algorithm which is suitable for real time systems. Clustering strategy will boost performance far beyond any tuning that might be done after the processes are loaded and executed will reduce your working set size by keeping similar size job into one cluster; yield higher data density; and reduce the overhead and save memory space [6].Heaviest weight

Cluster has lowest turn around time and lowest waiting time in comparison to other clusters. The above method can also be implemented in multi-processor system where the heavier clusters can be executed by high speed processors, thus increasing efficiency and decreasing the time complexity of today's complex working of operating system.

VII. REFERENCES

[1] Ke-Bing Zhang,"*Visual Cluster Analysis in Data Mining*",Department of Computing Division of Information and Communication Sciences Macquarie University, Australia,2007

[2] Mohammad Reza Effat Parvar1 et Al. "*A Starvation Free IMLFQ Scheduling Algorithm Based on Neural Network*", Electrical and Computer Engineering Department, University of Tehran, Tehran, Iran., (2008), 27–36p

[3] Abdurazzag Ali Aburas, Vladimir Miho, "*Fuzzy Logic Based Algorithm for Uniprocessor Scheduling*", Electrical and Computer Engineering Department International Islamic University Malaysia.2008,499-504p

[4] Rami J. Matarneh, "*Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes*", Department of Management Information Systems, Al-Isra Private University, Amman, Jordan,2009, 1831-1837p

[5] Yaashuwanth .C & Dr.R. Ramesh, "*Design of Real Time scheduler simulator and Development of Modified Round Robin architecture* ", Department of Electrical and Electronics Engineering, Anna University Chennai, 2010, 43-47p

[6] C.Yaashuwanth & Dr.R. Ramesh,"*A New Scheduling Algorithms for Real Time Tasks*",Department of Electrical and Electronics Engineering, Anna University Chennai 600 025,India-2009

[7] Nizar Grira, Michel Crucianu, Nozha Boujemaa, "*Unsupervised and Semi-supervised Clustering: Brief Survey*",INRIA Rocquencourt, B.P. 105,78153 Le Chesnay Cedex, France, August 15, 2005

[8] s.b. kotsiantis, p. e. pintelas,"R*ecent Advances in Clustering: A Brief Survey*", Department of Mathematics University of PatrasEducational Software Development Laboratory Hellas

[9] A.K.Jain, M.N Murthy,P.J.Flynn. "*Data Clustering:A Review*",ACM Computing Surveys, Vol. 31, No. 3, September 1999

BIODATA OF ALL THE AUTHORS

1.Prof H.S Behera is currently working as a Senior Lecturer in Dept. of Computer Science and Engineering in VSS University of Technology,Burla,Orissa,India.His research area of interest include Operating Systems,Data Mining and Distributed Systems.

2. Madhusmita Mishra is a MTech 2nd year student in Dept. of Computer Science and Engineering, VSS University of Technology,Burla,Orissa,India.

3.Simpi Patel is a final year MTech student in Dept. of Computer Science and Engineering, VSS University of Technology,Burla,Orissa,India.