IFERP

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 3, Issue 10, October 2016

# Markov Chain Modeling to Analyze Multi Process Fair Share Scheduling for Efficient Processing

[1]Manish Vyas, [2] Dr. Saurabh Jain
[1]Research Scholar, Faculty of Computer Science, Pacific Academy of Higher Education and Research University, Udaipur, India,
[2]Associate Professor, Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, India
[1] mai.vyas@gmail.com, [2] iamsaurabh_4@yahoo.co.in

***Abstract:* --- In multi process scheduling to ensure that no has excessively waiting for execution or resources, fairness is desired in predictable and responsive way. In case of innovative real time applications a fair share scheduler is required during execution to ensure that all resources will be allocated through fair manner on share and usage criteria. By fair share scheduling, improved processor utilization can be achieved by assigning dynamically priorities to executing processes. In this paper elementary scheme of multi process fair share scheduling is considered and by putting certain state of affairs over it, one more scheduling scheme is shaped. Comparative study of obtained scheme with elementary one is analyzed through simulation study under Markova chain model.**

***Keyword:* -- *CPU* Scheduling, Fair Share Scheduler, Markov Chain, State Probability, Transition Diagram**

## I. INTRODUCTION

Inclusive performance and productivity of multitasking operating systems depend on scheduling algorithms.For optimization in multi process scheduling, a flexible organized scheduler needed to ensure that each job will get a certain percentage of scheduler during execution [7][8].By using appropriate mechanism of fair share resources allocation during execution of processes, efficiency of scheduling can be improved [5]. Main objectives of a fair share scheduler are to ensure fairness, fast response time and load spreading without waiting as too long. It ensures that all resources will be allocated in a fair manner on share and usage criteria. By fair share scheduling, improved processor utilization can be achieved by assigning dynamically priorities to executing processes [6].CPU decides which process is to be executed next from ready queue. Processes are managed in terms of size, memory requirement, burst time etc. through various scheduling algorithms [1][4].

Scheduling involves randomization which can be studied by probabilistic study. The movement of scheduler over multiple processes can be analyzed through stochastic study of the system. Stochastic processes and their application in various fields have given an elaborated study in the field of computer science [2][3]. There are many scheduling algorithms are proposed for fair share scheduling. A Fair shared scheduling algorithm proposed as generic frame work for fair share algorithms from where various scheduling algorithms with different fairness characteristics can be derived [9]. For embedded multimedia applications, a real time scheduling method is presented which may schedule many processes by mixed scheduling using critical earlier based dead line first (EDF) and round robin method to utilize the CPU time [10]. A proportional share scheduling algorithm for fair share scheduling is proposed to reduce scheduling overhead by giving more chances to those processes which are close to completion [11]. Proportional Share CPU Scheduling Algorithm for Symmetric Multiprocessors as Surplus Fair Scheduling suggested which described by analysis that it is desirable for server operating systems.[12]. Recourse Management with fairness is described by a Fair Share Scheduler in efficient manner [13]. A new Fair share scheduling algorithm with weighted time slice for real time systems is proposed and by experimental analysis it is indicated that proposed approach may leads to gives better result [14].

## II. MARKOV CHAIN

A stochastic process is collection of random variables {Xn} indexed by time where 'n' will represent time which develops according to probabilistic rules. The set of possible values of Xn is known as state space. If in a stochastic process, present state Xk is independent of past states ( Xk−1, Xk−2, Xk−3, . . . , X1 ) that is state of a system at time t+1 depends only on its state at time t then it will satisfy markov property.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 10, October 2016

Markov process is a stochastic model that has Markov property. It can be used to model a random system that changes states according to a transition rule which depends only on current state and probabilities remain constant over time. If ‘x0’ is a vector which represents initial state of a system, then there is a matrix ‘M’ such that, state of system after first iteration is given by vector Mx0. Thus chain of state vectors x0, Mx0, M2x0, . . . , Mnx0 is called a Markov chain and matrix ‘M’ is called transition matrix.

## III. DESIGN OF MULTI PROCESS FAIR SHARE SCHEDULING SCHEMES

### A. *Elementary Scheme*

- Consider a fair share scheduling scheme with five processes P1, P2, P3, P4 and P5 in ready queue waiting for processing.
- In each round a fixed time quantum is given to each process for processing. The time quantum may varyfor different rounds of processing.
- Scheduler may begin with any of process from P1, P2, P3, P4 or P5 initially with a pre-defined equal priority such that,

$$\left( \sum_{i=1}^{5} pr_i = 1 \right)$$

- Movement of scheduler is random over all the states during execution all processes get finished. The transition diagram for above scheme will be as,

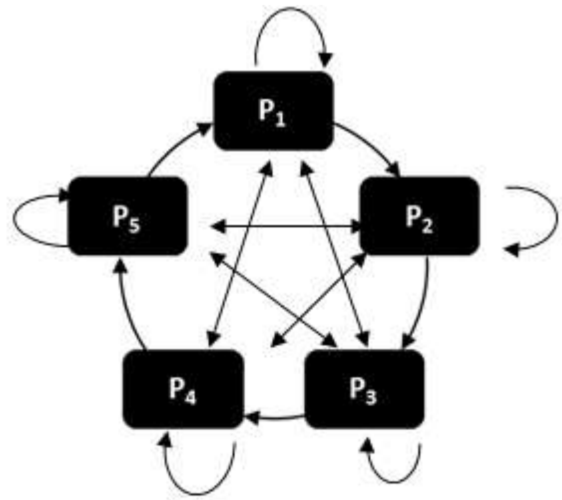


***Figure III (A): Transition Diagram of Elementary Scheme***

Solid filled shapes in above diagram show that initially scheduler may pick any of process and after completion of each allotted time quantum, it may reach to any of other process for execution or may remain at same process.
Now to analyze above scheme consider a markov chain as $\{x^n, n\geq1\}$ where ‘$x^n$’ indicates scheduler state and ‘n’ as time quantum. Scheduler can move over different state spaces $p_1$, $p_2$, $p_3$, $p_4$, and $p_5$ in different time quantum. Each state will have uniform initial probability as $pr_1$, $pr_2$, $pr_3$, $pr_4$ and $pr_5$.

Now consider $S_{ij}$ (for i,j=1,2,3,4,5) as transition probabilities of $X^n$ over different states then transition probability matrix for above scheme will be

$X^{(n)}$ ⟵ ⟶

$X^{(n-1)}$ ↕

| | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ |
|---|---|---|---|---|---|
| $P_1$ | $S_{11}$ | $S_{12}$ | $S_{13}$ | $S_{14}$ | $S_{15}$ |
| $P_2$ | $S_{21}$ | $S_{22}$ | $S_{23}$ | $S_{24}$ | $S_{25}$ |
| $P_3$ | $S_{31}$ | $S_{32}$ | $S_{33}$ | $S_{34}$ | $S_{35}$ |
| $P_4$ | $S_{41}$ | $S_{42}$ | $S_{43}$ | $S_{44}$ | $S_{45}$ |
| $P_5$ | $S_{51}$ | $S_{52}$ | $S_{53}$ | $S_{54}$ | $S_{55}$ |

Where $0 \leq S_{ij} \leq 1$and sum of each row is 1

If we apply markov chain model then generalized expressions of state probabilities after ‘*n*’time quantum will be,

$$p\left[X^{(n)}=P_1\right]=\sum_{m=1}^{5}\cdots\cdots\sum_{l=1}^{5}\sum_{k=1}^{5}\sum_{j=1}^{5}\sum_{i=1}^{5} pr_i . S_{ij} S_{jk} \cdots\cdots S_{m1}$$

$$p\left[X^{(n)}=P_2\right]=\sum_{m=1}^{5}\cdots\cdots\sum_{l=1}^{5}\sum_{k=1}^{5}\sum_{j=1}^{5}\sum_{i=1}^{5} pr_i . S_{ij} S_{jk} \cdots\cdots S_{m2}$$

$$p\left[X^{(n)}=P_3\right]=\sum_{m=1}^{5}\cdots\cdots\sum_{l=1}^{5}\sum_{k=1}^{5}\sum_{j=1}^{5}\sum_{i=1}^{5} pr_i . S_{ij} S_{jk} \cdots\cdots S_{m3} \quad \cdots\cdots III(A)$$

$$X^{(n-1)}$$

$$p\left[X^{(n)}=P_4\right]=\sum_{m=1}^{5}\cdots\cdots\sum_{l=1}^{5}\sum_{k=1}^{5}\sum_{j=1}^{5}\sum_{i=1}^{5} pr_i . S_{ij} S_{jk} \cdots\cdots S_{m4}$$

$$p\left[X^{(n)}=P_5\right]=\sum_{m=1}^{5}\cdots\cdots\sum_{l=1}^{5}\sum_{k=1}^{5}\sum_{j=1}^{5}\sum_{i=1}^{5} pr_i . S_{ij} S_{jk} \cdots\cdots S_{m5}$$

### *A. Scheme 1 (Execute Next Process at Random)*

- By putting certain state of affairs over elementary scheme, a different scheduling scheme is shaped as scheme 1.
- This scheme is based on imposing of scheduler to execute any of next process at random after completion of each time quantum.
- The scheduler movement is bounded in such a way that it can pick process $p_1$ initially and after completion of allotted time quantum, scheduler can move towards any of process but it can't remain at same state.
- Here initial probability for $p_1$ will be 1 while for remaining states it will be 0.

***Transition diagram for above scheme will be as,***

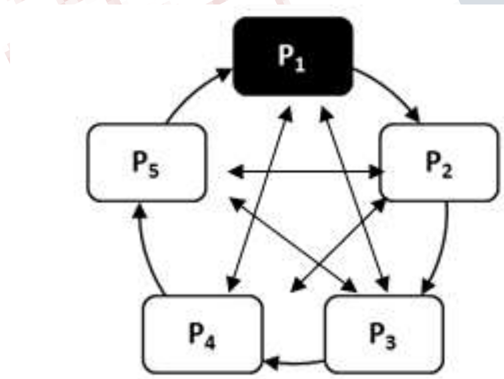


***Figure III (B): Transition Diagram of Scheme 1***

Transition probability matrix for above scheme will be $X^{(n)}$

| | $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ |
|---|---|---|---|---|---|
| $P_1$ | 0 | $S_{12}$ | $S_{13}$ | $S_{14}$ | $S_{15}$ |
| $P_2$ | $S_{21}$ | 0 | $S_{23}$ | $S_{24}$ | $S_{25}$ |
| $P_3$ | $S_{31}$ | $S_{32}$ | 0 | $S_{34}$ | $S_{35}$ |
| $P_4$ | $S_{41}$ | $S_{42}$ | $S_{43}$ | 0 | $S_{45}$ |
| $P_5$ | $S_{51}$ | $S_{52}$ | $S_{53}$ | $S_{54}$ | 0 |

If state probabilities is to be obtained with the decided criteria of scheme on the basis of above transition probabilities then, an indicator function $L_{ij}$(for i, j=1,2,3,4,5) need to be define such that,

$\mathbf{L_{ij} = 0}$ when (i=1, j=1), (i=2, j=2), (i=3, j=3), (i=4, j=4), (i=5, j=5)

$\mathbf{L_{ij} = 1}$ otherwise

Now by using markov chain model, generalized expressions of state probabilities after '*n*' time quantum will be,

$$p\left[X^{(n)}=P_1\right]=\sum_{m=1}^{5}\cdots\cdots\sum_{k=1}^{5}\sum_{j=1}^{5}\sum_{i=1}^{5} S_{1i}.L_{1i}\ S_{ij}.L_{ij}\ S_{jk}.L_{jk}\cdots\cdots S_{m1}.L_{m1}$$

$$p\left[X^{(n)}=P_2\right]=\sum_{m=1}^{5}\cdots\cdots\sum_{k=1}^{5}\sum_{j=1}^{5}\sum_{i=1}^{5} S_{1i}.L_{1i}\ S_{ij}.L_{ij}\ S_{jk}.L_{jk}\cdots\cdots S_{m2}.L_{m2}$$

$$p\left[X^{(n)}=P_3\right]=\sum_{m=1}^{5}\cdots\cdots\sum_{k=1}^{5}\sum_{j=1}^{5}\sum_{i=1}^{5} S_{1i}.L_{1i}\ S_{ij}.L_{ij}\ S_{jk}.L_{jk}\cdots\cdots S_{m3}.L_{m3}$$

$$p\left[X^{(n)}=P_4\right]=\sum_{m=1}^{5}\cdots\cdots\sum_{k=1}^{5}\sum_{j=1}^{5}\sum_{i=1}^{5} S_{1i}.L_{1i}\ S_{ij}.L_{ij}\ S_{jk}.L_{jk}\cdots\cdots S_{m4}.L_{m4} \quad \cdots III(B)$$

$$p\left[X^{(n)}=P_5\right]=\sum_{m=1}^{5}\cdots\cdots\sum_{k=1}^{5}\sum_{j=1}^{5}\sum_{i=1}^{5} S_{1i}.L_{1i}\ S_{ij}.L_{ij}\ S_{jk}.L_{jk}\cdots\cdots S_{m5}.L_{m5}$$

## III. SIMULATION STUDY

By the means of simulation study performance of scheduling schemes are evaluated and compared through data modeling under a common system. Data modeling requires various computation techniques to put on system behaviour. It is an interactive process in which makeover of qualitative information is convoyed into quantitative data. It simplifies possibility of outcomes of various processes in view of different input values. By data model approach probabilistic behavior of the processor over different processes can be also described. They determine that which data set can be used effectively and efficiently. For our consideration we have taken row dependent model *a + d.i* for analyzing the content. The model has two parameters'a' and 'd' whose values are come to be in linear order of increasing. Parameter *'a'* is origin while'd' is termed as scale. Here origin fixes while scale is maintained as varying and for any specific value of 'a', values of 'd' is increased in linear order. 'i'stands for process number and its values increase according to row wise (i=1,2,…). In the model, values are obtained in increasing order and the way of increment is linear. Row wise values will be obtained as *a*, *(a+d.i), (a+2d.i), ….*

Obtained values are transition probability values of processes. For balancing probability, a fair share factor 'F' is proposed by which last value is equally spread among all processes. Here 'F' is calculated by dividing last value to number of processes and it is added to all values as *a*F*, *(a+d.i)*F* . Last probability value will remain same as 'F'.

## IV. GRAPHICAL ILLUSTRATION

Simulation study through graphical analysis of state probabilities obtained from transition probabilities after applying markov chain model for proposed schemes is represented below. X –axis point to time quantum while Y –axis denotes probability.

### a. Elementary Scheme

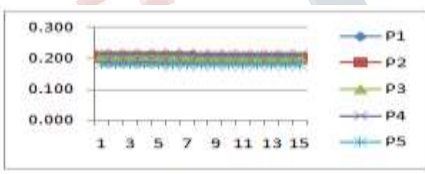


*Fig. V (a1) for a = 0.001 and d = 0.002*

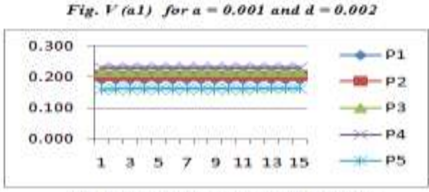


*Fig. V (a2) for a = 0.001 and d = 0.004*

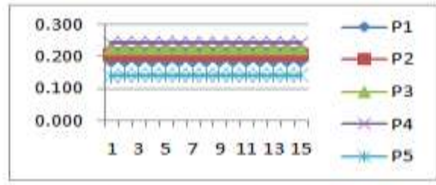


*Fig. V (a3) for a = 0.001 and d = 0.006*

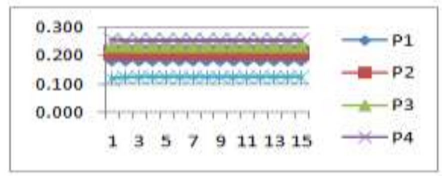


*Fig. V (a4) for a = 0.001 and d = 0.008*

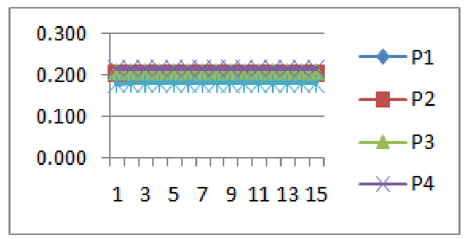


*Fig. V (a5) for a = 0.002 and d = 0.002*

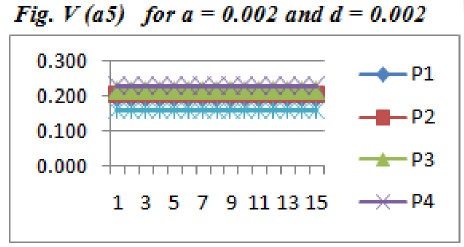


*Fig. V (a6) for a = 0.002 and d = 0.004*

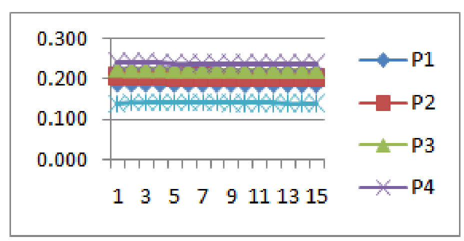


*Fig. V (a7) for a = 0.002 and d = 0.006*

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 10, October 2016

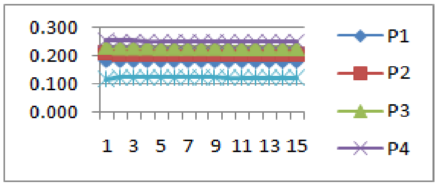


***Fig. V (a8) for a = 0.002 and d = 0.008***

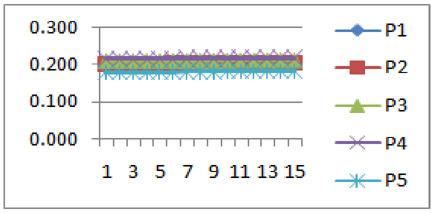


***Fig. V (a9) for a = 0.003 and d = 0.002***

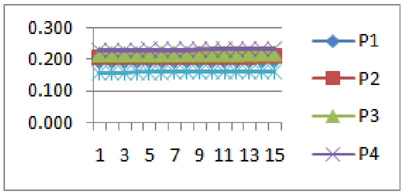


***Fig. V (a10) for a = 0.003 and d = 0.004***

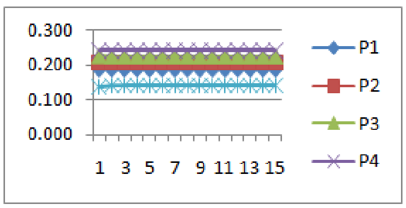


***Fig. V (a11) for a = 0.003 and d = 0.006***

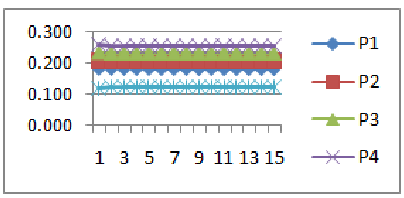


***Fig. V (a12) for a = 0.003 and d = 0.008***

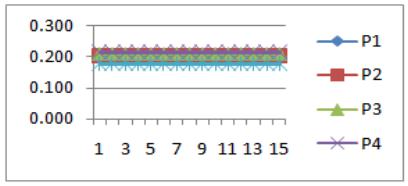


***Fig. V (a13) for a = 0.004 and d = 0.002***

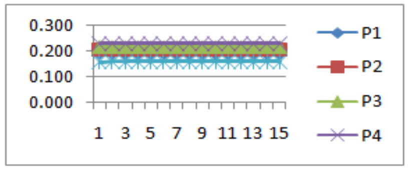


***Fig. V (a14) for a = 0.004 and d = 0.004***

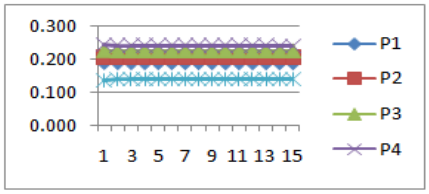


***Fig. V (a15) for a = 0.004 and d = 0.006***

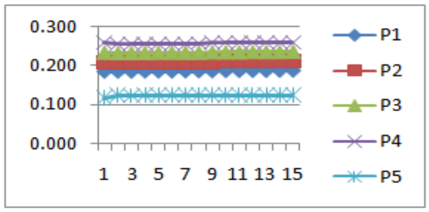


***Fig. V (a16) for a = 0.004 and d = 0.008***

***b. Scheme 1 (Execute Next Process At Random)***

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 10, October 2016

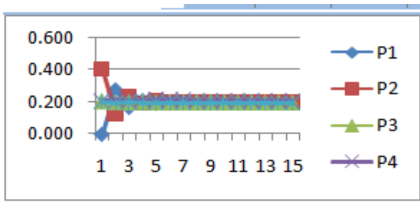


*Fig. V (b1) for a = 0.001 and d = 0.002*

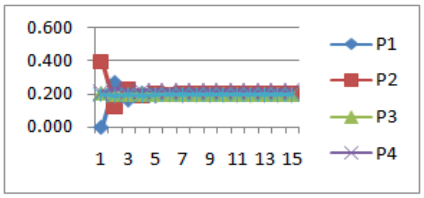


*Fig. V (b2) for a = 0.001 and d = 0.004*

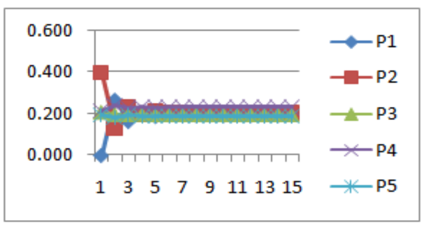


*Fig. V (b3) for a = 0.001 and d = 0.006*

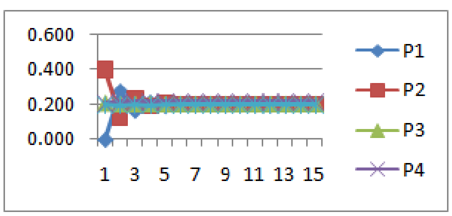


*Fig. V (b6) for a = 0.002 and d = 0.004*

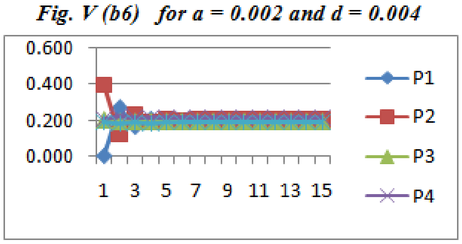


*Fig. V (b7) for a = 0.002 and d = 0.006*

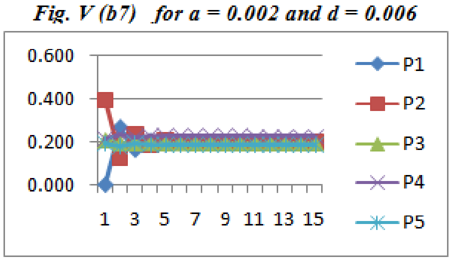


*Fig. V (b8) for a = 0.002 and d = 0.008*

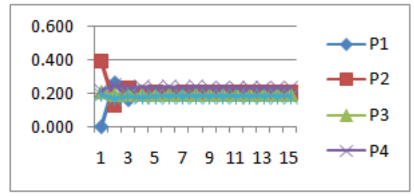


*Fig. V (b4) for a = 0.001 and d = 0.008*

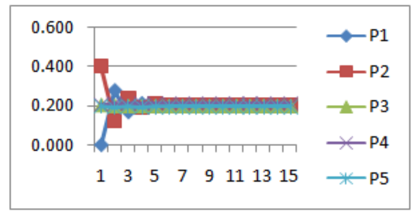


*Fig. V (b5) for a = 0.002 and d = 0.002*

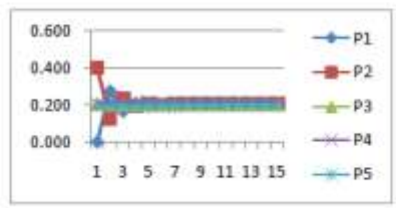


*Fig. V (b9) for a = 0.003 and d = 0.002*

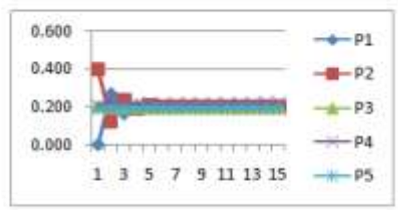


*Fig. V (b10) for a = 0.003 and d = 0.004*

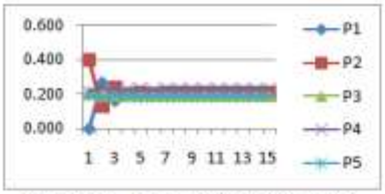


*Fig. V (b11) for a = 0.003 and d = 0.006*

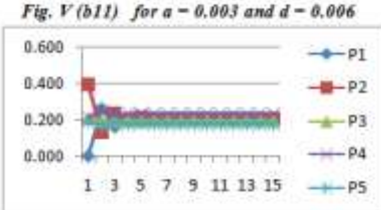


*Fig. V (b12) for a = 0.003 and d = 0.008*

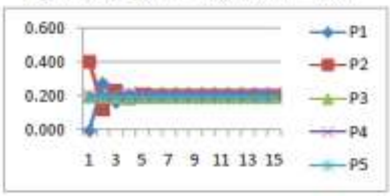


*Fig. V (b13) for a = 0.004 and d = 0.002*

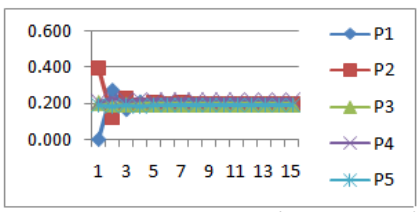


*Fig. V (b14) for a = 0.004 and d = 0.004*

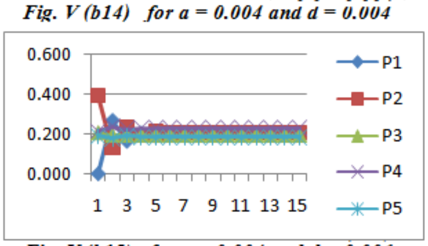


*Fig. V (b15) for a = 0.004 and d = 0.006*

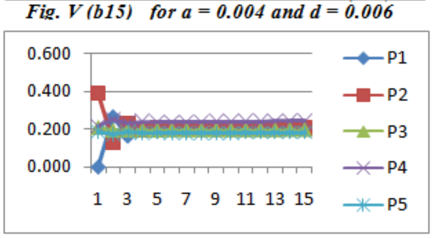


*Fig. V (b16) for a = 0.004 and d = 0.008*

## VI. CONCLUSION

In this paper multi process fair share scheduling is conferred which incorporate two schemes. Both schemes are designed and then evaluated on the basis of simulation study. Analysis shows that elementary scheme is a traditional fair share scheme in which scheduling pattern is uniform and there is no change in probabilities of processes during execution. Eventually scheme 1 may provide better fairness environment with lower scheduling overheads. It makes a better raised area than conventional fair share scheduling. Analysis can be concluded in view of Markova approach on multi process fair share scheduling that scheme 1 may be supportive for process scheduling as it build an environment that each process will obtain processor share in fair manner. This scheme supposed to be more operative for scheduler.

## REFERENCES

1) Tanenbaum, A.S. and Woodhull, A.S. (2009): Operating System: Design and Implementation, third Ed., Prentice Hall of India Private Limited, New Delhi

2) Parzen, E. (2015): Stochastic Process, Holden– day, Inc. San Francisco, California

3) Medhi, J. (2010): Stochastic processes, , Wiley Limited Third Edition (Reprint)

4) Silberschatz, Galvin and Gagne (2009): Operating systems concepts, Eighth Ed., Wiley

5) Dhotre, I.A.(2008): Operating Systems, Technical Publications

6) Hharris, A.J.(2008): Operating Systyems, Tata McGraw-Hill

7) Deitel , H.M., Deitel, P.J. and Choffnes,D.R.(2009) : Operating Systyems, Pearson

8) Dhamdhere, D.M.(2009): Operating System– A Concept Based Approach, Tata McGraw-Hill

9) Aravind, A. A. and Chelladurai, J. (2005):fair shared round robin CPU scheduling algorithms, Journal of Systems and Software, Elsevier Science

10) Ahn, B. , Kim, J., Lee, D. and Lee, S. (2006): A Real Time Scheduling Method for EmbeddedMultimedia

Applications, http://beru.univbrest.fr/~singhoff/cheddar/contribs/examples_of_use/ahn06.pdf

11) Helmy, T. andDekdouk, A.(2007):Burst Round Robin as a Proportional-Share SchedulingAlgorithm, https://www.researchgate.net/publication/228570872

12) Chandra, A., Adler, M., Goyal, P., and Shenoy, P. (2000):Surplus Fair Scheduling: A Proportional Share CPU Scheduling Algorithm forSymmetric Multiprocessors, http://scholarworks.umass.edu/cs_faculty_pubs/22

13) Mana, S.C.(2012): Recourse Management Using a Fair Share Scheduler, International Journal of Computer Science and Security (IJCSS), Vol.(6), No.(1)

14) Behera, H.S. ,Mohanty,R., Panda,J., Thakur, D. and Sahoo, S.(2016):Experimental Analysis of New Fair-Share SchedulingAlgorithm with Weighted Time Slice for Real TimeSystems, Journal of Global Research in Computer Science

15) Shukla, D., Jain, S. (2010): Performance Evaluation of a General Class of Multi-Level Queue Scheduling Scheme, Computer Science and Telecommunications (GESJ), Vol.(3).,No.(26)

16) SHUKLA,D., OJHA,S. JAIN, S.(2010):Data Model Approach and Markov Chain Based Analysis of Multi-Level QueueScheduling, Journal of Applied Computer Science & Mathematics, Vol. 8, No. (4)

17) Sendre, R.(2015):A Survey on Research Trends for Modeling CPU Scheduling Algorithm, International Journal in IT and Engineering