www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 10 Oct. 2016, Page No. 18568-18574

Stochastic Modeling for Analyzing Scalability Impact of Lottery Scheduling using Proportion Reformation

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

Abstract: For effective processor scheduling, algorithms are required to develop not only for fair scheduling but also for efficient implementation of resource management with rapid adjustment to control over relative execution rates. Proportional share scheduler assure that each job obtain a certain percentage of processor time. Lottery scheduling is based on randomized approach to achieve proportional share resource management where resources are allocated to the clients in proportion to their respective weights. In this paper conventional lottery scheduling scheme is designed and extended along with some conditions to get new scheduling schemes. Stochastic modeling is applied for study and analysis.

1. Introduction

Process management is one of the major responsibilities of any operating system. It involves allocating various resources to processes including processor which must be shared efficiently among all processes. The fundamental objective of scheduling is to provide efficient and fair scheduling by modular resource management with ensuring that each process get equal share of processor over long run. To accomplish this objective a novel scheduler known as proportional-share scheduler is carried out with an elementary concept that scheduler put efforts for obtaining certain amount of time to each job. Initially a randomized resource allocation algorithm is proposed as lottery scheduler. It efficiently implements proportional–share resource management with probabilistically fair.

Lottery scheduling is a probabilistic scheduling algorithm which states that a lottery will determine that which process will get to run next. By the means of lottery, probabilistically fair selection of next resource holder is selected.

All processes are assigned some lottery tickets which are in terms of abstract, relative and uniform resource rights and can be used to represent share of a resource that a process should receive. The percent of tickets that a process has will be its share of receiving system resource. Scheduler draws random ticket to select a process. Distribution of tickets need not be uniform and granting a process more tickets provides it a relative higher chance of selection. It is a randomized approach to achieve proportional share resource management in proportion to respective weights of processes. It provides a flexible and useful concept for multiplexing scarce resources among processes.

2. Literature Review

Proper use of dynamic ticket adjustments in a lottery scheduler can improve interactive response [8]. Scheduling in a queuing system is proposed with asynchronously varying service rates to describe state of server as well as queues [9]. Weight readjustment algorithm presented to indicate that it can reduce unfairness in resources allocation and may be desirable for server operating systems as well as wireless networks[10][11]. Flow control mechanism proposed by lottery scheduling and stated through stochastic simulation that adjustment of resource scheduling can increase network performance and throughput [12]. Lottery scheduler for the Linux kernel is contributed as probabilistically fair with prevention of starvation [15]. A proportional share scheduler is suggested for providing accurate proportional sharing [16]. For improvement in quality of service parameters, a Markov based performance model is intended for resource allocation [14]. [13]

DOI: 10.18535/ijecs/v5i10.40

Presented resource elasticity fairness to determine execution of each process as fair share. Generalization of max-min fairness approach is suggested for providing fair allocation of multiple resource and by analysis it is expressed that it leads to better throughput and fairness than slot-based fair sharing schemes [18]. Similar contribution provided by [19] by implementing lottery scheduling in Linux kernel. Lottery Scheduling as novel randomized resource allocation mechanism discussed for service requests based systems like database; media based and networks applications by [22] to provide efficient modular resource management. For generalized processor sharing in more efficient way, deterministic fluid models of fair schedulers are presented [23].

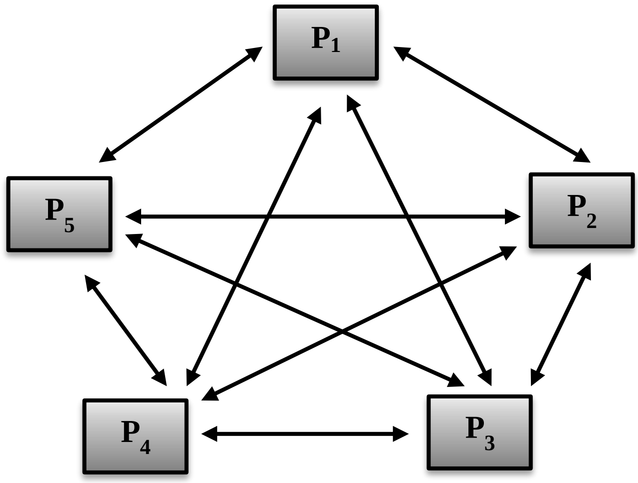
To attempt efficient fair scheduling with a probability proportional and generalization of modular resource management with dynamic ticket adjustments in lottery scheduler, we designed a conventional lottery scheduling scheme and extended it to get some new scheduling schemes. Stochastic modeling is applied for graphical study and analysis.

3. Formation and Analysis of Lottery Scheduling

To analyzing scalability impact of lottery scheduling we design some schemes based on randomized approach along with flexible proportional share resource management. All schemes are compared under data model approach. Keeping essential impact of lottery scheduling, initially we picked structural scheme and then some additional scheduling schemes are shaped.

3.1 Structural Scheme

This scheme is formulated on fundamental principle of lottery scheduling. Consider a multiprocessing scheduling scheme with five processes P_1, P_2, P_3, P_4 and P_5 in ready queue. A fixed time quantum is set for each process to be executed. Here as per decided lottery value, a process which holds minimum tickets can be picked by scheduler. This will be achieved probabilistically in random manner. Initial probability of each process to be executed is equal as Scheduler will move to any other process after completion of time quantum. If any process gets complete within allotted time quantum, it comes out from ready queue otherwise it remains in waiting queue and wait for next quantum to allot for processing.



Structural Scheme

Filled rectangle shows that scheduler may pick any of process initially and arrows show scheduler movement after completion of each allotted time quantum.

Now by considering stochastic modeling for analyze above scheme, if we apply markov chain model then unit step transition matrix will be

	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

To get data set effectively and efficiently row dependent model $\mathbf{p_{ij} = \alpha_i + i.(d_j)}$ is taken, where 'i' & 'j' are rows and columns respectively. 'p_{ij}' is probability value of a specific process to be executed subsequently. 'α' and 'd' are two model parameters whose value is obtained in linear order. By probabilistic data model approach transition probability of all processes will be obtained

	P_1	P_2	P_3	P_4	P_5
P_1	α_i	$\alpha_i + i.d_j$	$\alpha_i + (i+1).d_j$	$\alpha_i + (i+2).d_j$	$1-(4\alpha_i + 6i.d_j)$
P_2	$\alpha_i + i.d_j$	$\alpha_i + (i+1).d_j$	$\alpha_i + (i+2).d_j$	$\alpha_i + (i+3).d_j$	$1-(4\alpha_i + 10i.d_j)$
P_3	$\alpha_i + (i+1).d_j$	$\alpha_i + (i+2).d_j$	$\alpha_i + (i+3).d_j$	$\alpha_i + (i+4).d_j$	$1-(4\alpha_i + 14i.d_j)$
P_4	$\alpha_i + (i+2).d_j$	$\alpha_i + (i+3).d_j$	$\alpha_i + (i+4).d_j$	$\alpha_i + (i+5).d_j$	$1-(4\alpha_i + 18i.d_j)$
P_5	$\alpha_i + (i+3).d_j$	$\alpha_i + (i+4).d_j$	$\alpha_i + (i+5).d_j$	$\alpha_i + (i+6).d_j$	$1-(4\alpha_i + 22i.d_j)$

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

To ensure flexible proportional share, scheduling indicator P is suggested which balance probability values. It is obtained from last probability value. And its proportional share is added to each column value of respective row.

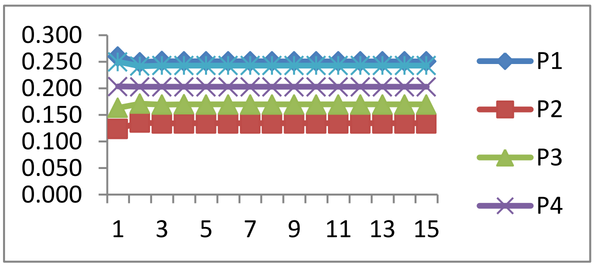
$$\left(\sum_{i=1}^{5} pr_i = 1\right)$$

DOI: 10.18535/ijecs/v5i10.40

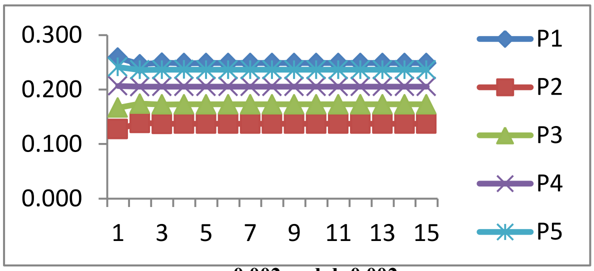
For example for first row P will be $1 - 4.\alpha_i + 6i.d_j$ and it is added in first row as $(\alpha_i *0.10P)$, $\alpha_i + i.(d_j) *0.15P$, $\{\alpha_i + (i+1).d_j\}*0.20P$ ……..

Markov chain model is applied on transition probabilities obtained from above approach and state probabilities are attained.

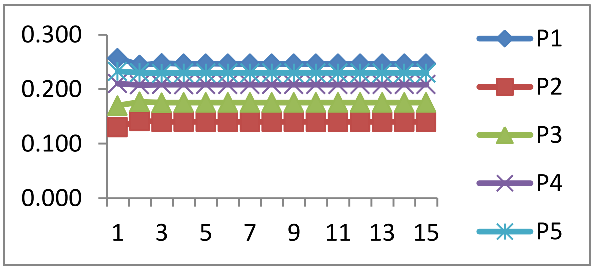
Graphical analysis of state probabilities is,



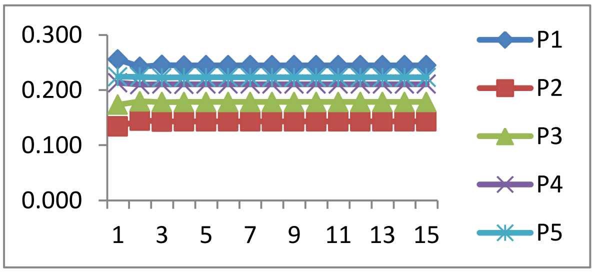
α = 0.002 and d=0.001



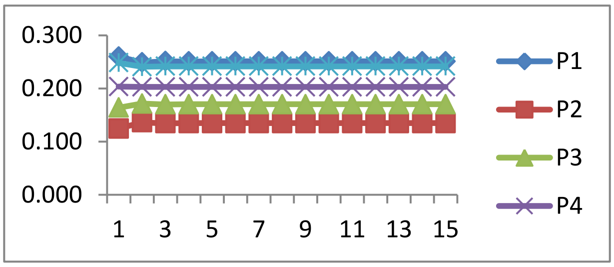
α = 0.002 and d=0.002



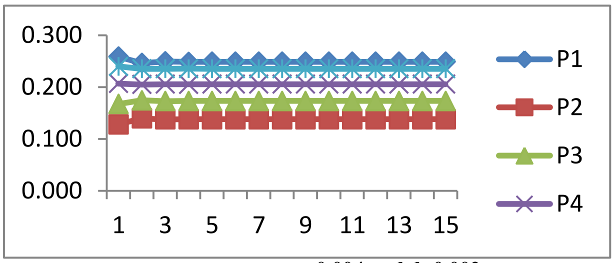
α = 0.002 and d=0.003



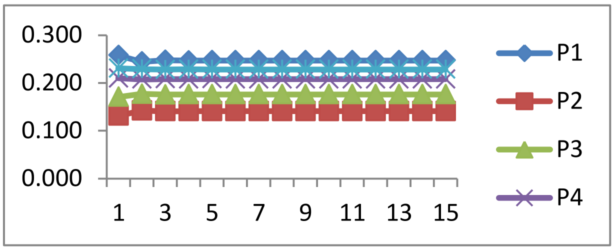
α = 0.002 and d=0.004



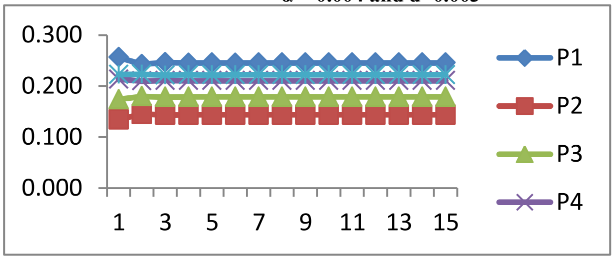
α = 0.004 and d=0.001



α = 0.004 and d=0.002



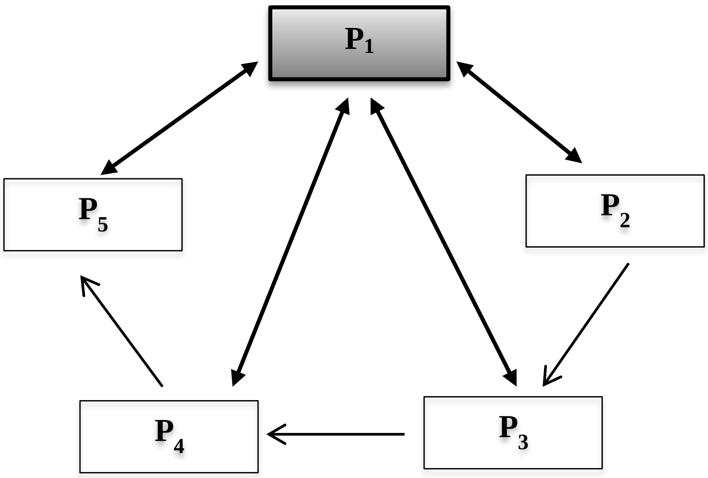
α = 0.004 and d=0.003



α = 0.004 and d=0.004

3.2 Subsequent or Outset Execution

This scheme is designed by supposing that initially process P_1 going to be executed by earning maximum tickets. After completion of first time quantum, scheduler can execute any of process excluding P_1 which means that any of process can win lottery. Afterwards scheduler can move towards next process or return to process P_1. So after completion of each time quantum either process P_1 has chance to gain maximum tickets or that process which is next to process executing currently. This provision of scheduler moment carries on till all processes get concluded. Initially state probability for P_1 will be 1 while that of remaining it will be 0.

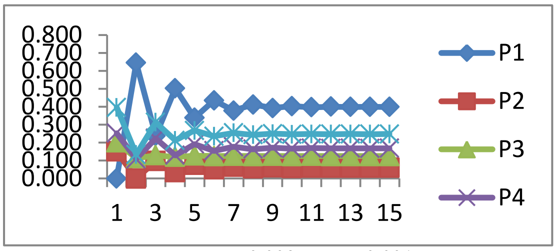


Transition Diagram

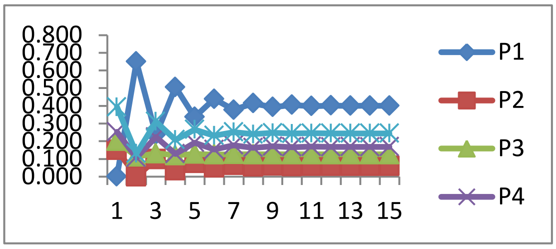
Unit step transition matrix for the above scheme will be

	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}	0	0
P_3	S_{31}	0	0	S_{34}	0
P_4	S_{41}	0	0	0	S_{45}
P_5	S_{51}	0	0	0	0

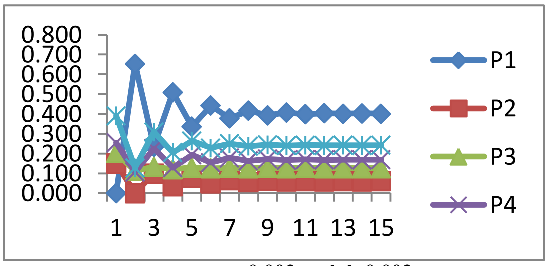
Transition probabilities obtained through data model and markov chain model applied to get state probabilities same as structural scheme. Graphical analysis for simulation study for above scheme,



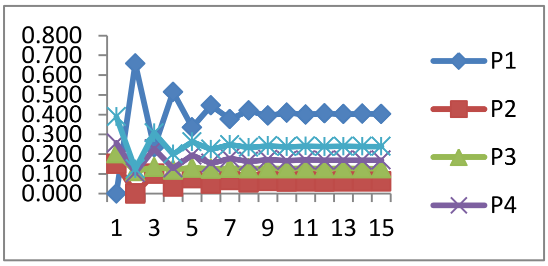
α = 0.002 and d=0.001



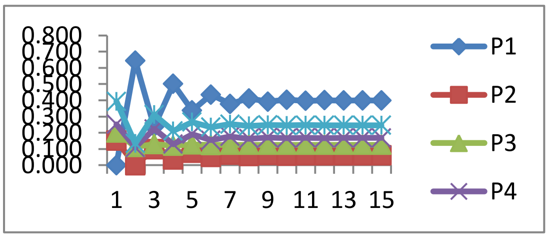
α = 0.002 and d=0.002



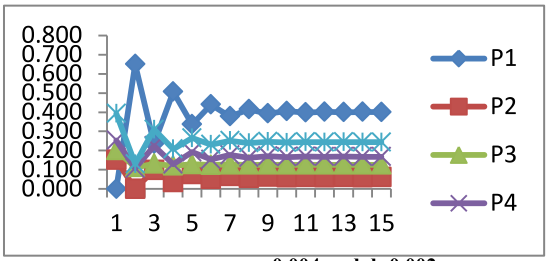
α = 0.002 and d=0.003



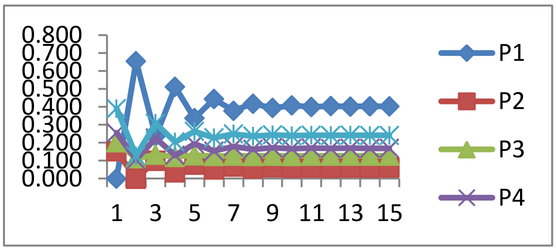
α = 0.002 and d=0.004



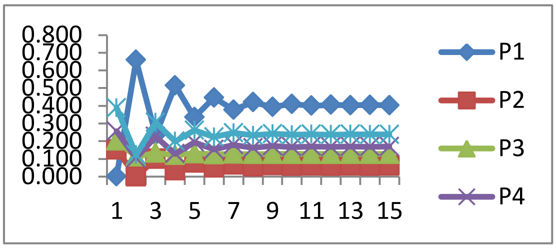
α = 0.004 and d=0.001



α = 0.004 and d=0.002



α = 0.004 and d=0.003



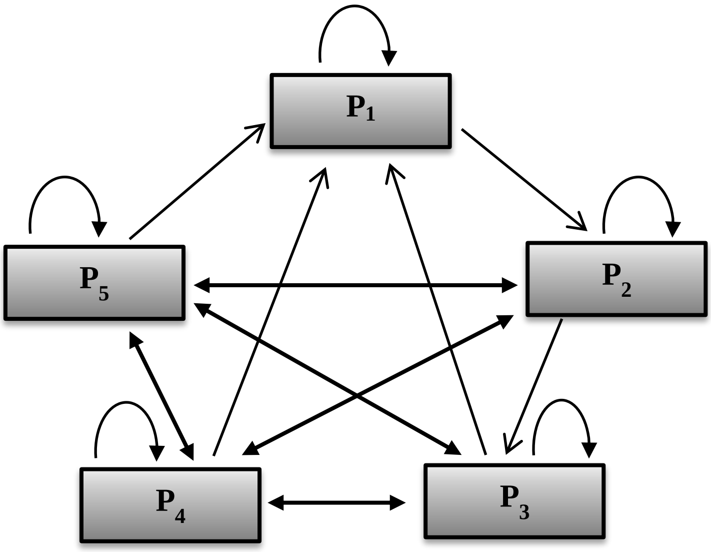
α = 0.004 and d=0.004

3.3 Lined order expansion

In this scheme initially any of process may be executed by winning lottery as earning maximum tickets. That is scheduler can pick any one process in beginning. Structure of scheme is such that if process P_1 is executing currently by having largest number of tickets then after completion of allotted time quantum, either scheduler will hold P_1 or process P_2 will be executed.

Similarly if P_2 is executing currently then in next quantum scheduler may continue with P_2 or P_3 or P_4. So after completion of each time quantum, collective processes in linear order has chance to gain maximum tickets and to be executed next.

As scheduler can pick any of process initially hence initial state probabilities for all will be equal as pr_1, pr_2, pr_3, pr_4, pr_5. Transition diagram of the scheme

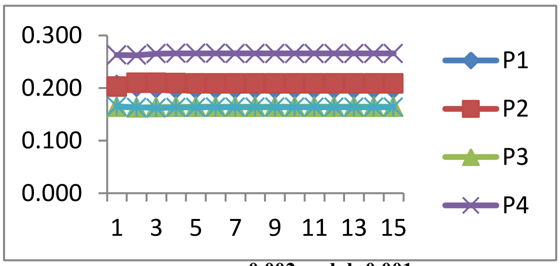


Lined Order Expansion

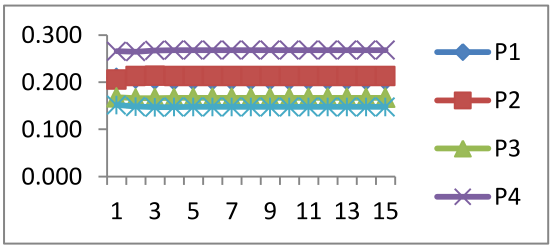
Unit step matrix will be

	P_1	P_2	P_3	P_4	P_5
P_1	S_{11}	S_{12}	0	0	0
P_2	0	S_{22}	S_{23}	S_{24}	0
P_3	S_{31}	0	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}

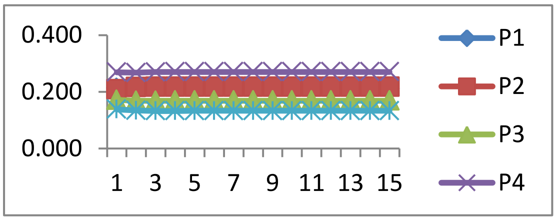
Graphical analysis is,



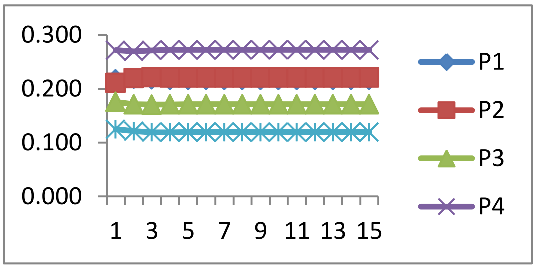
α = 0.002 and d=0.001



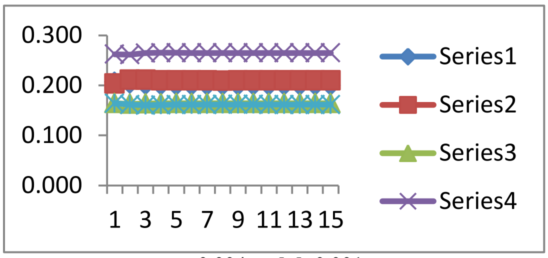
α = 0.002 and d=0.002



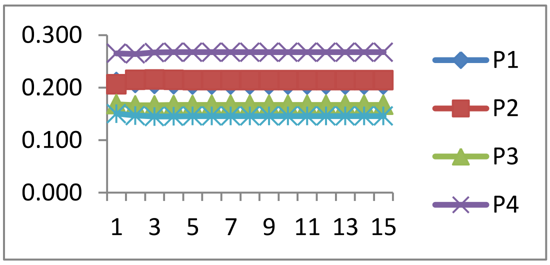
α = 0.002 and d=0.003



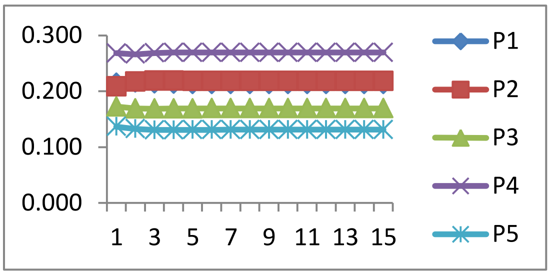
α = 0.002 and d=0.004



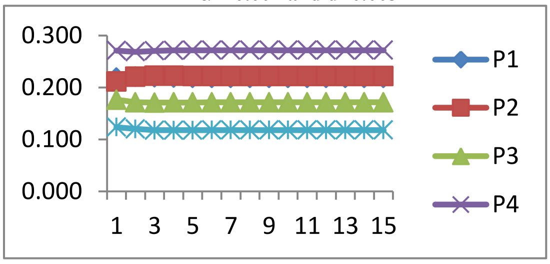
α = 0.004 and d=0.001



α = 0.004 and d=0.002



α = 0.004 and d=0.003



α = 0.004 and d=0.004

4 Concluding Remark

Inclusive behaviour of structural scheme is found on symmetry with respect to each process. Here movement of all processes during execution is more or less consistent with slight increase or decreases at initial level. Afterwards they almost remain unaffected during entire scheduling. Overall in this scheme, each process scheduled with its initial level priority.

Subsequent or Outset Execution scheme makes an effort for getting primitive aspect of lottery scheduling to some extent with raise in probability values of processes during execution. In beginning of execution, probability value of each process cutback for few time quanta and then there is gain in each one. Later on all becomes steady with certain development. So here general pattern of scheme seems to be supportive for proportional share resource management.

Comprehensive impression of Lined order expansion scheme is nearly similar as structural scheme in which somewhat symmetry is found with respect to each process. Processes P_1, P_3 & P_4 performs in steady manner and remain nearly unaffected during entire scheduling. Process P_2 turn into stable through some gain and P_5 is resulted to same with some downturn.

After analysis it can be inferred that both structural scheme and lined order expansion scheme follows scheduling pattern in uniform fashion and there is no sizable development in probabilities of any processes during execution. Meanwhile subsequent or outset execution scheme makes an effort for getting raise in probability values of processes during execution. Although addition in probability is less than that of initial value, but it is somewhat considerable and both schemes appear to be helpful for proportional share resource management.

Analysis can be concluded by considering Stochastic modeling that subsequent or outset execution scheme supposed to be operative and can be put forward for providing a supportive environment for randomized scheduling.

5 References

1. Murrell, H.(2006): Operating Systems , University of KwaZulu-Natal
2. Taylor, H.M. and Karlin, S. (1998):An Introduction To Stochastic Modeling, Third Ed., Academic Press, Elsevier
3. Bower, T.(2009):Operating Systems Study Guide, *http://faculty.salina.k-state.edu/tim/ossg*
4. Stallings, W. (2014): Operating Systems – Internals and Design Principles, Eight Ed., Pearson
5. Tanenbaum, A.S. and Herbert, B.(2015): Modern Operating Systems, Fourth Ed., Pearson
6. Silberschatz, Galvin and Gagne (2009): Operating systems concepts, Eighth Ed., Wiley
7. Parzen, E. (2015): Stochastic Process, Holden–day, Inc. San Francisco, California
8. Medhi, J. (2010): Stochastic processes, , Wiley Limited Third Ed. (Reprint)
9. Petrou, D. , Milford, J.W. and Gibson, G.A. (1999): Implementing Lottery Scheduling, Matching the Specializations in Traditional Schedulers, Proceedings of USENIX Annual Technical Conference, California
10. Andrews, M. , Kumaran, K., Stolyar, A., Vijayakumar, R. and Whiting, P. (2004): Scheduling In A Queuing System With Asynchronously Varying Service Rates, Journal Probability in the Engineering and Informational Sciences Vol. 18 No. 2
11. Chandra, A. , Adler, M., Goyal, P. and Shenoy, P.(2000): Surplus Fair Scheduling – A Proportional-Share CPU Scheduling Algorithm for Symmetric Multiprocessors, *http://scholarworks.umass.edu/cs_faculty_pubs, Computer Sciences Commons*
12. Eugene, T.S., Stoica, I. and Zhang, H. (2000): Packet Fair Queuing Algorithms for Wireless Networks with Location-Dependent Errors, *http://www.cs.cmu.edu/~hzhang/papers/INFOCOM98a.pdf*
13. Kasilingam, S. and Seshadhri, V. (2015): Design of Flow Control Mechanism Model for Markovian Multi Stage Queuing Network, World Engineering and Applied Sciences Journal Vol. 6, No.3
14. Zahedi, S.M. and Lee, B.C. (2014):

Resource Elasticity Fairness With Sharing Incentives For Multiprocessors, *http://people.duke.edu/~bcl15/documents/zahedi14-asplos.pdf*

15. Kawasaki1,R.Y., Guedes, L.A., Cardoso1, and Vijaykumar, N.L. (2010): A Markovian Performance Model for Resource Allocation Scheduling on GNU/Linux,
http://dblp.uni-trier.de/pers/hd/c/ Costa:Jo=atilde=o_Cris=oacute=stomo_Weyl_Albuquerque

16. Mejía, M., Betancourt, A.M. and Patki, T. (2015): Lottery scheduler for the Linux kernel,
Dyna rev.fac.nac.minas Vol. 82, No. 189, Medellín,

17. Nieh, J. Vaill, C. and Zhong, H. (2001): Virtual-Time Round-Robin: An O(1) Proportional Share Scheduler,
Proceedings of the USENIX Annual Technical Conference Boston, USA

18. Bitirgen, R. Ipek, E. and Martinez, J.F.(2008): Coordinated Management of Multiple Interacting Resources in Chip Multiprocessors: A Machine Learning Approach,
ACM Digital Library,
http://dl.acm.org/purchase.cfm?id=1521805

19. Ghodsi, A. Zaharia, M. , Hindman, B. Konwinski, A. Shenker, S. and Stoica, I. (2011): Dominant resource fairness: fair allocation of multiple resource types,
Proceedings of Eighth USENIX conference on Networked systems design and implementation Berkeley , USA

20. Zepp, D.(2012):Lottery Scheduling in the Linux Kernel: a Closer Look,
https://pdfs.semanticscholar.org/f272/36e8b34042a8ef899832541fb681c01a2ebc.pdf

21. Vyas, M. and Jain, S.(2016): Comparative Study of Extensive Round Robin Scheduling by Data Model Approach under Markov Chain, International Journal of Recent Research Aspects, Vol. 3, No. 2

22. Hoque, E. and Dey, T.(2009): Comparing Lottery and EEVDF Scheduling Algorithm for Real-time Applications,
https://www.cs.virginia.edu/~td8h/docs/os_paper_ET2.pdf

23. Shukla, Y.C. and Katiyar, P. (2011): Lottery Scheduling,
https://docs.google.com/document/d/1Y8C8Knx46gAzxAQx33bc1mTTiL5h77YmSYLsabW-foI/edit?hl=en_US

24. Szabó, R. (2000): Admission and Flow Control in Generalized Processor Sharing Schedulers,
Ph.D. Thesis submitted at Budapest University of Technology And Economics Budapest, Hungary
https://repozitorium.omikk.bme.hu/bitstream/handle/10890/146/ertekezes.pdf?sequence=1