

Dynamic Fair Priority Optimization Task Scheduling Algorithm in Cloud Computing: Concepts and Implementations

Deepika Saxena, Scholar

Department of Computer Science & Applications Kurukshetra University, Kurukshetra (Haryana) India.
E-mail: 13deepikasaxena@gmail.com

Dr. R.K. Chauhan, Dr. Ramesh Kait

Department of Computer Science and Applications Kurukshetra University, Kurukshetra (Haryana) India.
E-mail: rkc.dcsakuk@gmail.com, ramesh.kait@gmail.com

Abstract—Cloud computing has become buzzword today. It is a digital service where dynamically scalable and virtualized resources are provided as a service over internet. Task scheduling is premier research topic in cloud computing. It is always a challenging task to map variety of complex task on various available heterogenous resources in scalable and efficient way. The very objective of this paper is to dynamically optimize task scheduling at system level as well as user level. This paper relates benefit-fairness algorithm based on weighted-fair Queuing model which is much more efficient than simple priority queuing. In proposed algorithm, we have classified and grouped all tasks as deadline based and minimum cost based constraints and after dynamic optimization, priority of fairness is applied. Here different priority queue (high, mid, low) are implemented in round-robin fashion as per weights assign to them. We recompile the CloudSim and simulate the proposed algorithm and results of this algorithm is compared with sequential task scheduling and simple constraints (cost and deadline) based task scheduling algorithm. The experimental results indicates that proposed algorithm is, not only beneficial to user and service provider, but also provides better efficiency and fairness at priority level, i.e. benefit at system level.

Index Terms—Cloud computing, priority, scheduling, weighted-fair queue.

I. INTRODUCTION

Computational world is very broad and complex. In this respect, cloud computing has undertaken almost entire space. Basically, cloud is a collection of resources (hardware and software) distributed at worldwide datacenters. There are many servers available at various datacenters which are provided by service providers throughout the world. We are paying as per our demand for using those resources. There are many popular issues for research in cloud computing like virtualization, data security, license management, scalable storage

management, mobile cloud, availability of services, task scheduling. But, scheduling of job is always a prime topic of research in cloud computing. There are heterogeneous resources available at various datacenters. So, traditional scheduling algorithms like FCFS, shortest job first, round-robin and priority etc, are not recommendable. The various challenges of task scheduling in cloud environment are:

- 1) To allocate resources to task
- 2) To decide in which order the cloud should execute the task
- 3) To schedule overhead when VMs prepare, terminate or switch task, communication overhead should be minimum.
- 4) It requires continuous VM status monitoring.
- 5) Cost of using VMs, dispatching VMs to different tasks etc.

Since scheduling world is very dynamic, heterogeneous and complex in cloud computing, we require some efficient scheduling technique that can optimize and improve the overall performance of scheduling system. Scheduling technique should be such that can do well and provide complete satisfaction at user end, at service provider end and also load balancing at system end.

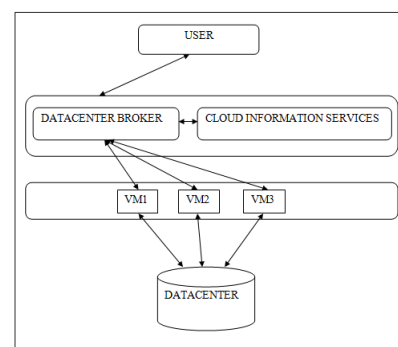


Fig.1. Scheduling in Cloud

In cloud computing, scheduling of task is done at two levels:

- 1) Resource level scheduling, deployment of VMs at available physical nodes is done
- 2) User level scheduling, tasks are assigned to VMs

Customer put QoS constraints like minimum cost and task done faster etc. CSP (Cloud Service Provider) requires maximum returns on investment. At system level, maximum resource utilization and load balancing is required. In cloud computing, Broker acts as an intermediary between user and CSP. Broker exists (as shown in fig.1) at system level, the broker decides where to map job or task submitted by user to the resource provided by CSP [2]. So, while designing any new scheduling algorithm all the changes are performed at DCB (Data Center Broker). Likewise, many researchers have provided many scheduling algorithms, which are working well in one or the other way.

But our proposed scheduling algorithm for task scheduling is highly improved and efficient which is based on concept of “Weighted Fair Queuing” technique to improve quality of service. It has removed the drawbacks of already existing priority based task scheduling algorithms, i.e. starvation of low priority queues (long waiting priority queue). Now, with our proposed algorithm, there occur no long waiting priority queues. It provides fairness at priority level by implementing combination of priority queue and round-robin fashion scheduling at grouped cloudlet level. This improved algorithm proves to be beneficial to all, (resource manager and load balancing) and satisfying QoS constraints at each level. The remainder of the paper is sectioned as follows: section 2 discusses literature review, section 3 discusses proposed methodology for cloudlet scheduling, section 4 discusses proposed algorithm, section 5 shows experimental data and results and finally section 6 concludes the overall study.

II. RELATED WORKS

Grouping task for scheduling after prioritization effectively reduces processing time in comparison to task scheduling without grouping. Cost of task scheduling is further reduced, if VM is selected dynamically on basis of cost and processing power [1]. Average turnaround time and average cost of overall task scheduling is minimized, when turnaround time and cost of each job is minimize individually. As a result, number of tasks increases, which improves the performance [2]. This paper has focused on grouping task, prioritization and greedy resource allocation. Criteria for calculating cost of every task must not be same, as some tasks are simple, some tasks are complex. Different task have different CPU requirement, memory requirement etc. So, activity based costing is better way of calculating cost of each task, which measures cost of objects and performance of activities and computes cost more accurately [3]. Task

execution cost can be reduced and user required QoS is improved using load balancing at resource level scheduling [4]. ERUA algorithm [5] satisfy user and cloud service provider through dynamic resource management where utilization ratio must fall under 1, leading to better resource utilization. This paper focus at resource level scheduling. Processing each job individually increases communication cost and time. Because of this communication overhead overall performance of task scheduling increases. But, job-grouping technique groups the small scaled user jobs in job groups which reduces overhead communication time [6].

There are different levels of elasticity structures offered by different cases of data flow structures, operator characteristics and other parameters etc for data flow schedule optimization on cloud [7]. Multiple QoS constraints based scheduling strategy to address multiple workflows in a decentralized cloud computing environment including two level task scheduling mechanism based on load balancing in cloud computing[8]. This task scheduling mechanism not only meet user’s requirements, but also get high resource utilization. Priority based dynamic resources allocation to tasks scheduling algorithm, which considers multiple SLA objective of job, by preempting best-effort job in cloud environment is described in [9]. In paper [10], it has been discussed that hierarchical scheduler exploit the multicore architecture for effective scheduling They have used diversity of task priority at local and global level for proper load balancing across heterogeneous processors. TDP algorithm is there, where ‘T’ stands for task selection, ‘D’ for deadline and ‘P’ means priority in terms of cost, which selects task according to its constraints and requirements, finally scheduling is done using single priority queue [11]. In heterogeneous environment of resources, the turnaround time of each job is minimized individually, to minimize the average turnaround time of all submitted jobs in a timeslot [12].

Though we have so many scheduling algorithms available, still some algorithm are better in one way or some are better in other way, none of them is completely efficient. Many algorithms make use of priority, but they all have disadvantage of long waiting queues. Generally, what happens in earlier proposed algorithms, they classify tasks as cost-based or deadline –based and then apply simple priority queues. In deadline-based grouping, task with higher deadline, having lower priority wait for longer duration for its execution, though it arrived so early. Also, in cost-based scheduling, the task with lesser execution cost (lower in priority) have to wait for longer duration for its execution. But, our proposed algorithm has included all good points of existing algorithms with new and enhanced version of priority that has surely remove the deficiency of all existing priority based task scheduling. We have added weighted fair queue to introduce priority of fairness in our proposed task scheduling algorithm.

The concept of “Weighted Fair Queue” is taken from the book William Stallings “High-Speed Networks and Internet, Performance and Quality of Service, Pearson

education”, under the topic “Scheduling”, which is a Technique to improve QoS.

III. THE PROPOSED METHODOLOGY

Our proposed scheduling method includes following concepts:

- (1) **Constraint-based grouping of tasks:** - Grouping means collecting similar kind of tasks altogether in one dimension, all of them having same requirements. Task grouping can have many types like; grouping of tasks can be based on length of task, location of task, deadline of task, cost of task, complexity of task etc. Here, we have included two-dimensional grouping of task. One dimension consist of constraint imposed by user i.e. Deadline based task grouping. Actually, almost every user wants his request to be met as earlier as possible, so applying deadline constraints over tasks will be quiet beneficial for user. Second dimension satisfy constraint of cloud service provider (CSP) which is, maximum resource utilization and earning maximum profit i.e. cost-based task grouping. From business point of view, applying cost-based constraint on tasks submitted by user will allow CSP to mend more money. As, CSP will his use minimum cost machine to execute maximum length task, so he is getting more money by utilizing benefit from minimum cost machine. Grouping of tasks certainly reduces communication overhead. For example, if 1000 tasks are given for execution, suppose 680 tasks are deadline-based and rests are cost-based. Then, to check for its constraint (deadline-based or cost-based) separately for each task at runtime would increase overhead for system. So, it is better to group them previously according to their constraints, before arranging them in different priority list.
- (2) **Weighted Fair Priority queue :-** Among all the available scheduling methods, Weighted Fair Queue is best scheduling technique where tasks(cloudlets) are assign to different priority queues for scheduling. Weighted Fair Queue Model is shown in fig 2; in these queues are weighted based on priority of queue. Selected VM process task in each queue based on round-robin fashion where number of tasks selected for execution from each queue depends on its queue weight. For example, let weight assigned to high priority queue is 3, weight of mid queue is 2 and weight granted to low priority queue is 1. then at each cycle(round) 3 tasks are processed from high priority queue, 2 tasks from mid priority queue and 1 task is executed from low priority queue. In this way, tasks would be executed in round-robin fashion and priority fashion scheduling, both method goes together side-by-side. So, there will be no long waiting low priority queues.
- (3) **Greedy resource (VM) allocation:** - This approach is greedy in respect that it selects resource, which

appears best at instant. It means scheduler or broker selects VM with minimum turnaround time for each individual task for deadline-based task scheduling. Minimizing turnaround time for each job will definitely reduce overall turnaround time and increase response time for all task taken together. This is a great enhancement regarding system performance, providing benefit at system level. For cost-based tasks, it selects VM with minimum cost and accurate processing power for cost-based tasks. The task with highest cost (which is decided based on task length) is assign to VM with minimum possible cost. This reduces cost of execution of each task. Here, cost refers to CSP’s minimum cost machine is best utilized to execute the maximum length task of the user and providing maximum money benefit to him. So, he is spending less and earning more. During dynamic optimization, Greedy allocation of resource searches local optima and finally reaches global optima.

Continuous VM-status monitoring is done by calculating waiting time and then updating turnaround time of respective VM at each VM selection. Here, figure 3 is depicting the model of proposed algorithm, which is making use of weighted fair queue i.e. shown in figure 2.

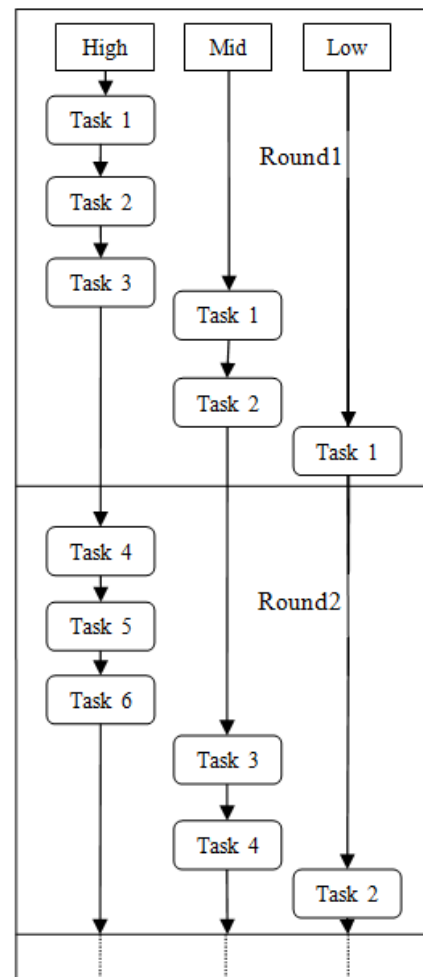


Fig.2. Weighted Fair Queue Model

Making use of this best scheduling technique surely enhances the degree of multiprogramming by making each VM busy executing at least one task at a time, definitely CPU utilization increases and response time decreases for each task in particular. Hence, it is three way optimization technique, as user is getting his task executed faster and Cloud Service Provider is getting maximum benefit at cost of machine level. Also resource utilization i.e. VM utilization is at best promising level.

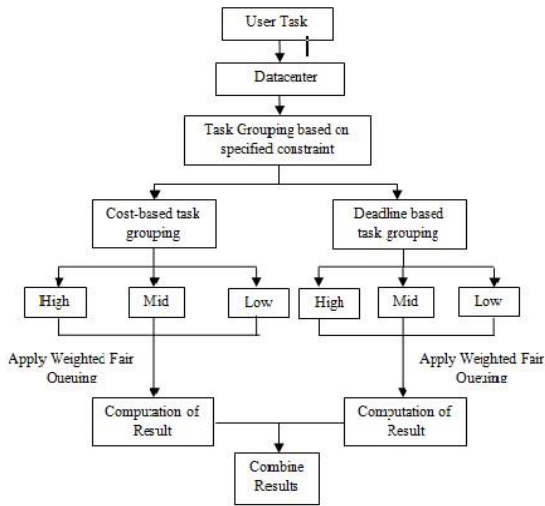


Fig.3. Proposed Algorithm Model

IV. PROPOSED ALGORITHM

Our proposed algorithm includes following steps:

User will submit number of cloudlets to DCB (Datacenter Broker), then random generator number will decide type of constraint for each task, so constraints are decided dynamically at runtime (they are not prefixed). Also, each task has different length which is changing by certain amount (for example, if length of previous task is 1000, then for next task it can be $1000 + 100 * \text{random generator number}$), length of task is also changing dynamically during execution. Deadline for each task is also decided using random generator number, so deadline is also not static. So, overall setup of experiment is truly dynamic which gives real view of practical situation in cloud computing:

- (1) Make groups of tasks based on constraints (deadline or cost-based) classification.
- (2) For deadline based task group, do:
 - (i). Arrange deadline based task in 3 priority queues – high, mid, low.
 - (ii). Take one task at a time from each priority queue, using weighted fair queue.
 - (iii). For each VM compute minimum turnaround time as, $\text{Turnaround time} = \text{waiting time} + (\text{task length} / \text{processing power})$. Also, calculate cost of task using formula $\text{cost of VM} * (\text{task length} /$

processing power). And update its waiting time accordingly.

- (iv). For each task, select VM with minimum turnaround time, so that computation time of each task is reduced individually.
- (3) For cost based task group, do:
- (i). Arrange task in high, mid, low priority queues according to descending order of task length.
 - (ii). Apply weighted fair queue, and take one job at a time.
 - (iii). Compute total execution time of task execution using (start time of task execution – end time of task execution) Also, Compute cost of each VM using formula:
 $\text{Cost} = \text{resource cost} * (\text{task length} / \text{processing power})$ and sort VM in descending order.
 - (iv). Assign task to minimum cost VM and task length \leq MIPS of VM (so optimization is done at cost level).

Finally, to get combined results, sum up total execution cost of all task based on cost-based and deadline-based constraint together. Similarly, sum up total execution time of all task from both the groups.

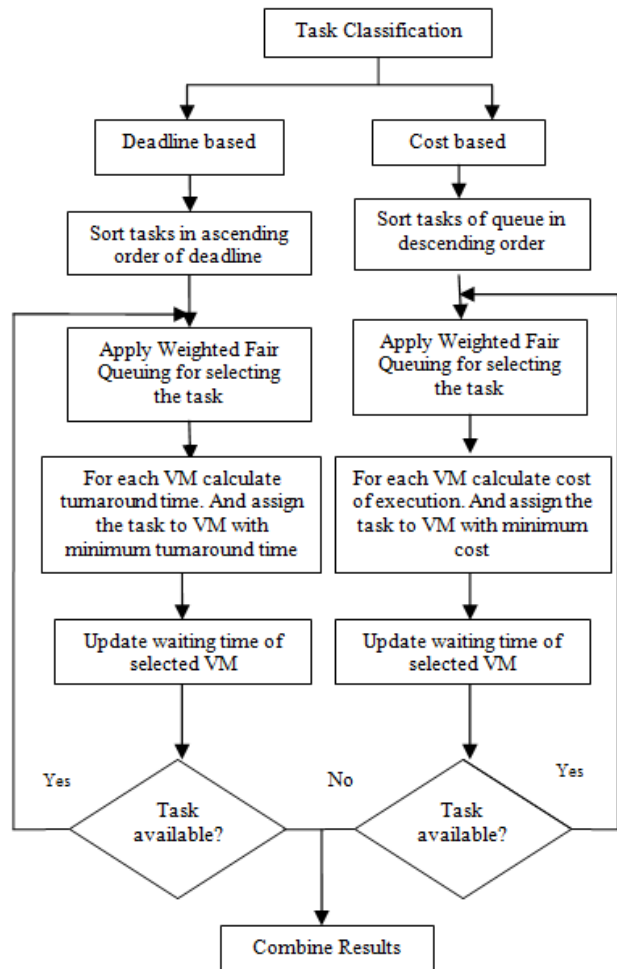


Fig.4. Flowchart of Proposed Algorithm

A. Pseudo code for our proposed algorithm is as follows:

It consists of three different modules namely, Schedule (), Deadline () and Cost ().

```

Schedule ()
For each task Ti
If priority: = deadline
Then invoke Deadline (Ti)
Else
If priority: = cost
Then invoke Cost (Ti)
End
    
```

```

Deadline (A)
Add A to priority queue P
Sort P in ascending order of deadline
Divide the tasks into three priority queues High, Mid and Low
Select a task using Weighted Fair Queuing approach
Assign the task to the VM having minimum turnaround time.
Calculate time of execution and cost of execution of task.
Update the selected VM status
    
```

```

Cost (B)
Add B to priority queue Q
Sort Q in descending order of profit
Divide the tasks into three priority queues High, Mid and Low
Select a task using Weighted Fair Queuing approach
Assign the task to the VM having minimum cost of execution and
length of task<= MIPS of VM. Calculate time of execution and
cost of execution.
Update the selected VM status
    
```

At last, compute total cost and total time of execution of all task in both groupings together.

V. EXPERIMENTAL DATA ANALYSIS

The CloudSim toolkit is used demonstrate the simulation. The simulation results are verified using CloudSim (2.1.1) to check the correctness of proposed algorithm [2]. The simulation results of proposed algorithm are compared with the Sequential assignment, which is in-build in CloudSim and Dynamic Optimization Algorithm for Task scheduling [2].

The configuration of Datacenter created is as shown below,

Table I shows the configuration of hosts and Table II depicts the configuration of virtual machines used in this simulation framework.

Here,

Number of processing element - 1

Number of hosts – 2

Table 1. Configuration of Host

RAM (MB)	10240
Processing Power (MIPS)	110000
VM Scheduling	Time shared

Table 2. Configuration of Vms

Virtual Machine	VM1	VM2
RAM (MB)	5024	5024
Processing Powe (MIPS)	22000	11000
Processing Elements (CPU)	1	1

The configuration of the system on which these results are obtained is as shown below:

Table 3. System Configuration

Processor	Pentium® Dual-core CPU T4400@2.20GHz, 2.20GHz
RAM	3.00GB
System types	32bit operating system
Operating System	Windows 7 Ultimate

Performance with Time: - The experimental results show the remarkable improvement in time over the sequential approach as well as Dynamic Optimization Algorithm [2].

Table 4. Results of Proposed Algorithm and Existing Algorithm

No. of cloudlets	Sequential algorithm	Dynamic optimization without fair priority	Proposed algorithm with Fair Priority
20	98.6154	67.6984	45.3135
40	417.5499	283.441	235.2001
60	1041.3430	539.131	485.3890
80	1953.1897	885.0639	716.0989
100	3103.9777	1486.6538	1166.1269

Comparing proposed algorithm with sequential i.e. FCFS scheduling algorithm and Dynamic optimization scheduling algorithm [2] shows the tremendous improvement in results. As the number of cloudlets are increasing definitely, the total execution time has decreased together for both deadline and cost based tasks. The result shown above are the average of total execution time obtained after several number of execution for each number of cloudlet (e.g. we have run the implementation 20 times for 100 cloudlets and calculated its average).

Performance with Cost: - The experimental results show the enhanced performance of proposed algorithm over existing algorithms.

Table 5. Results of Proposed Algorithm and Existing Algorithm

No. of cloudlets	Sequential algorithm	Dynamic optimization without fair priority	Proposed algorithm with Fair Priority
20	76.0	75.377	73.300
40	168.0	145.109	140.001
60	273.0	220.500	206.690
80	396.0	300.036	281.904
100	534	369.877	310.554

As the number of cloudlets are increasing the total execution cost has decreased together for both deadline and cost based tasks. The results shown above are the average of total execution cost obtained after several number of execution for each number of cloudlet (e.g. we have run the implementation 15 times for 100 cloudlets and calculated its average).

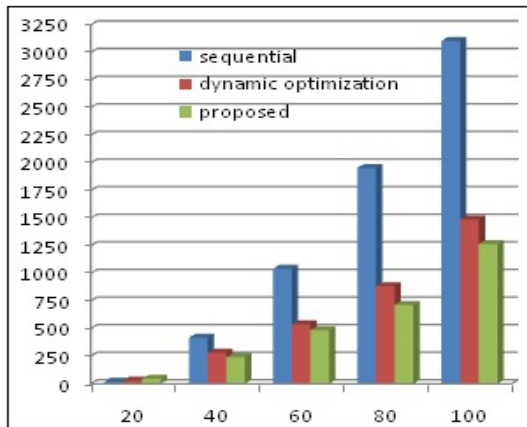


Fig.5. Traditional V/S Proposed Algorithm With Respect To Time

The above bar graph showing task completion time and comparison of traditional algorithm and proposed algorithm.

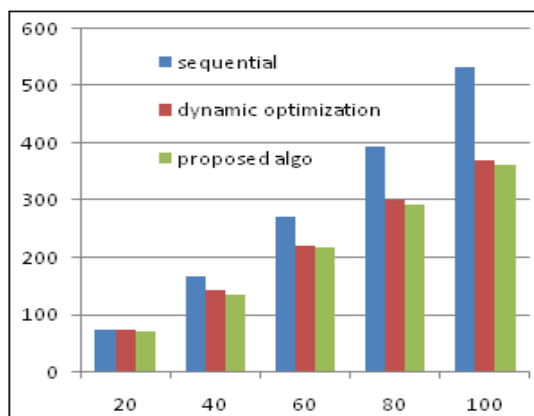


Fig.6. Cost Comparison Traditional V/S Proposed Algorithm With Respect To Cost

The above figure 6 depicts the information of task scheduling cost comparison of traditional algorithm with proposed algorithm.

The experimental results vary every time on running the simulation as the priorities of the incoming user tasks are not fixed, they are random for example, for 100 cloudlets, suppose at first execution, 40 tasks are cost based and rest are deadline-based, may be at next run of 100 task, 68 task are cost-based and rest are deadline-based, so somewhat results vary, but each time results are improved as compared to sequential or dynamic optimized algorithm without fair priority. Also, simulation result varies depending upon the system configuration (for example, system type, RAM, processor version, and different OS) on which the simulation is running.

VI. CONCLUSIONS AND FUTURE WORK

Scheduling of task in cloud environment is a highly researched and challenging issue in cloud computing. To meet thousand of service requests while making best possible use of available resources and simultaneously

satisfying both user as well as service provider, is the challenge for task scheduler. Traditional methods of scheduling lead to overpricing and slow processing for bulk of tasks. Some task scheduling algorithm is cost based, some are deadline based and many algorithms make use of priority based scheduling. But they suffer from long waiting priority queues. Our proposed algorithm definitely meet all the challenges, along with constraint based optimization scheduling. We have also, introduced fair-priority scheduling concept i.e. combination of priority with round-robin scheduling scheme. This brings fairness at priority level and increases utilization of resources at system level and thereby providing much more efficient results than that can be provided by any other existing task scheduling algorithm.

In future, more pragmatic algorithm can be devised using several other combination of scheduling schemes like, Shortest Job First (SJF) with priority or SJF with weighted fair queue, or round-robin with SJF etc. Grouping of tasks can be done based on complexity of task, or group by location etc., before resource allocation.

REFERENCES

- [1] Yogita Chawla and Mansi Bhonsle, "Dynamically optimized cost based task scheduling in Cloud Computing", International Journal of Emerging Trends & Technology in Computer Science, Volume 2, Issue 3 May-June 2013.
- [2] Monika Chaudhary and Sateesh Kumar Peddoju, "A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment" International Journal of Engineering Research and Application, Vol 2, Issue 3, May-June 2012.
- [3] Qi Cao, Zhi-bo Wei and Wen-Mao Gong, "An optimization algorithm for task scheduling based on activity based costing in cloud computing", Wuhan University, China 978-1-4244-2902-8/09/\$25.00 2009 IEEE.
- [4] Hong Sun, Shi-ping Chen, Chen Jin and Kai Guo, "Research and simulation of task scheduling algorithm in cloud computing" University of Shanghai, China, July 25, 2013.
- [5] Ram Kumar Sharma and Nagesh, Sharma, "A Dynamic optimization algorithm for task scheduling in cloud computing with resource utilization,"IJSET@2013 volume no.2, Issue no 10, pp: 62-68, year 2013.
- [6] Nithiapidary Muthuvelu, Junyang Liu, Nay Lin Soe, Srikumar Venugopal, Anthony Sulistio and Rajkumar Buyya, "A Dynamic job grouping-based scheduling for deploying applications with fine-grained tasks on global grids" The University of Melbourne, Australia ICT building, 111 Barry Street, Carlton VIC 3053, 2005.
- [7] Herald Killapi, Eva Sitaridi and Manolis M. Tsangaris, "Schedule optimization for data processing flows on the Cloud", University of Athens, copyright 2011.
- [8] Vandana Choudhary, Saurabh Kacker, Tanupriya Choudhary and Vasudha Vashisht, "An Approach to improve task scheduling in a decentralized cloud computing environment", IJCTA| jan-feb 2012.
- [9] Chandrashekar S.Pawar and Rajnikant B. Wagh, "Priority based dynamic resource allocation in Cloud computing", Shirpur, India, 2011.
- [10] S.V Parikh et al., "Double Level Priority Based Task Scheduling with Energy Awareness in Cloud Computing",

- Journal of Information, Knowledge and Research in Information Technology, ISSN: 0975 – 6698| NOV 12 TO OCT 13 | VOLUME – 02, ISSUE – 02
- [11] Dr.V.Vaithyanathan et al., “An Efficient TPD Scheduling Algorithm for Cloud Environment”, International Journal of Engineering and Technology (IJET), ISSN: 0975-4024, Vol 5 No 3 Jun-Jul 2013.
- [12] Rep. UCB/EECS, vol. 28, 2009. J. Geelan, "Twenty-one experts define cloud computing," Cloud Computing Journal, vol. 2009, pp. 1-5, 2009.
- [13] R. Mikkilineni and V. Sarathy. (2009). Cloud Computing and the Lessons from the Past in 18th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE'09, 2009.
- [14] S. Ma, "A Review on Cloud Computing Development," Journal of Networks, vol. 7, no.2, pp. 305-310, 2012.
- [15] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, vol. 53, no.6, 2009.
- [16] I. Foster, Z. Yong, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," 2008, pp. 1-10.
- [17] CloudComputing[Online]: http://en.wikipedia.org/wiki/cloud_computing.
- [18] A. Fox and R. Griffith, "Above the clouds: A Berkeley view of cloud computing," Dept. Electrical Eng. and Compute. Sciences, University of California, Berkeley.
- [19] S. Singh and K. Kant, "Greedy grid scheduling algorithm in dynamic job submission environment," in Emerging Trends in Electrical and Computer Technology (ICETECT), 2011 International Conference on, 2011, pp. 933-936.
- [20] R. N. Calheiros, et al., "CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services," Arxiv preprint arXiv: 0903.2525, 2009.
- [21] C T Lin*1 et al., “Comparative Based Analysis of Scheduling Algorithms for Resource Management in Cloud Computing”, *1 Department of Mechanical Engineering University of California, Davis, California, Vol.-1, Issue-1, July 2013Deborah Estrin, Lewis Girod, Greg Pottie, and Mani Srivastava. Instrumenting the world with wireless sensor networks. In International Conference on Acoustics, Speech, and Signal Processing, 2001.
- [22] Gaganpreet Kaur Sehdev, Anil Kumar, "Performance Evaluation of Power Aware VM Consolidation using Live Migration", IJCNIS, vol.7, no.2, pp. 67-76, 2015.DOI: 10.5815/ijcnis.2015.02.08.
- [23] Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines", Proceedings of the 2nd Symposium on Networked Systems Design and Implementation (NSDI'05), 2005.
- [24] Feng-Seng Chu, Kwang-Cheng Chen and Chen-Mou Cheng, "Toward Green Cloud Computing", ACM, 2011.
- [25] Sumit Goyal, "Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review", IJCNIS, vol.6, no.3, pp.20-29, 2014. DOI: 10.5815/ijcnis.2014.03.03.
- [26] Y. Jadeja and K. Modi, "Cloud computing - concepts, architecture and challenges", In Computing, Electronics and Electrical Technologies (ICCEET), pp. 877-880, March 2012.
- [27] Bhushan Lal Sahu and Rajesh Tiwari, "A Comprehensive Study on Cloud Computing", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 9, September 2012.
- [28] Sarojadevi K and Jeevitha R, "Uncloud the Cloud of Cloud Computing" <http://www.caesjournals.org/splupload/IJCAES-AISC-2011-174.pdf>.
- [29] Shengmei Luo, Zhaoji Lin, Xiaohua Chen, Zhuolin Yang and Jianyong Chen "Virtualization security for cloud computing service," International Conference on Cloud and Service Computing, pp.174-179, 2011.
- [30] J E Smith and Ravi Nayar, "Introduction to virtual Machines", Elsevier Science, Nov 14, 2004.
- [31] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges", Internet Services and Applications-Springer, pp.7-18, 2010.
- [32] M. B. Mollah, K. R. Islam, and S. S. Islam "Next generation of computing through cloud computing technology" CCECE, page 1-6, IEEE, 2012.
- [33] Shyam Patidar, Dheeraj Rane and Pritesh Jain, "A Survey Paper on Cloud Computing", Second International Conference on Advanced Computing & Communication Technologies- IEEE, 2012.
- [34] Rich Lee and B. Jeng, "Load Balancing Tactics in Cloud", International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery- IEEE, pp. 447 - 454, Oct. 2011.
- [35] N. Meghanathan, "A Survey on the Communication Protocols and Security in Cognitive Radio Networks," *International Journal of Communication Networks and Information Security*, (IJCNIS), vol. 5, no. 1, pp.19-37, April2013.
- [36] Pushtikant Malviya and Shailendra Singh, "A Study about Green Computing", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 6, June 2013.
- [37] Mrs .Sharmila Shinde, Mrs. Simantini Nalawade, Mr .Ajay Nalawade, "Green Computing: Go Green and Save Energy", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, July 2013.
- [38] S. Pearson, —Taking account of privacy when designing cloud computing services, In: IEEE ICSE Workshop on Software Engineering Challenges of Cloud Computing, (pp. 44-52), May 2009.
- [39] S. Yu, C. Wang, K. Ren and W. Lou, —Achieving secure, scalable, and fine-grained data access control in cloud computing, in: Proceedings of IEEE INFOCOM, (pp. 1-9), March 2010.
- [40] H. Takabi, J.B. Joshi and G.J. Ahn, —Security and privacy challenges in cloud computing environments, *I Security & Privacy*, IEEE, 8(6), 24- 31, 2010.
- [41] T. Dillon, C. Wu and E. Chang, —Cloud computing: Issues and challenges, In: Proceedings of IEEE 24th International Conference on Advanced Information Networking and Applications, (pp. 27- 33), April, 2010.
- [42] G. Boss, P. Malladi, D. Quan, L. Legregni and H. Hall, —Cloud computing, *IBM white paper*, 321, 224-231, 2007.
- [43] J.W. Rittinghouse and J.F. Ransome, —Cloud computing: implementation, management, and security, CRC press, 2009.
- [44] W. Kim, —Cloud Computing: Today and Tomorrow, *I Journal of Object Technology*, 8(1), 65-72, 2009.
- [45] Y. Chen, V. Paxson and R.H. Katz, —What’s new about cloud computing security, University of California, Berkeley Report No. UCB/EECS-2010-5 January, 20(2010).
- [46] D. Durkee, —Why cloud computing will never be 28 Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review Copyright © 2014 MECS

- I.J. Computer Network and Information Security, 2014, 3, 20-29 free. Queue, 8(4), 20, 2010.
- [47] D. G. Feng, M. Zhang, Y. Zhang and Z. Xu, —Study on cloud computing security, Journal of Software, 22(1), 71-83, 2011.
- [48] S. Pearson and A. Charlesworth, — Accountability as a way forward for privacy protection in the cloud, In: Cloud computing (pp. 131-144). Springer Berlin Heidelberg, 2009.
- [49] D. Zisis and D. Lekkas, —Addressing cloud computing security issues, Future Generation Computer Systems, 28(3), 583-592, 2012.
- [50] K. Popovic and Z. Hocenski, —Cloud computing security issues and challenges, In: Proceedings of IEEE 33rd International Convention on MIPRO, (pp. 344-349) May, 2010.
- [51] W. Jansen and T. Grance, —Guidelines on security and privacy in public cloud computing, NIST special publication 800-144, 2011.
- [52] S. Srinivasan and R.B. Krishnan, — Data property analyzer for information storage in cloud, in: Proceedings of IEEE International Conference on Pattern Recognition, Informatics and Medical Engineering (pp. 443-446), February, 2013.
- [53] S. Sawhney, H. Puri and H.V. Rietschote, U.S. Patent No. 8, 370, 312. Washington, DC: U.S. Patent and Trademark Office, 2013.

Kurukshetra University, Kurukshetra Haryana (India).She is having expertise and research working experience in field of Distributed Networks, Grid Computing, Cloud computing, Evolutionary algorithms like Genetic algorithms and heuristic optimization techniques like Particle Swarm Optimization, and Ant Colony optimization etc..



Dr. R. K. Chauhan is working as professor in the Department of computer science & Applications in Kurukshetra University, Kurukshetra Haryana (India). He is having twenty four years of teaching experience. He is having expertise and research working experience in field of Data Mining Relational data base, Big Data, Cloud Computing. He has published more than eighty research papers.



Dr. Ramesh Kait is working as Assistant professor in the Department of computer science & Applications in Kurukshetra University, Kurukshetra Haryana (India). He is having fourteen years of teaching experience. He is having expertise and research working experience in field of Wireless adhoc Networks, Mobile Computing, Wireless Sensor Networks, Security in Computing, Computer Networks, Networks Security and Vehicular adhoc Networks.

Authors' Profiles



Ms. Deepika Saxena has completed her M.Tech in CSE from Kurukshetra University Kurukshetra. Recently, she is working as Head of Department and Assistant professor in the Department of computer science at Dayanand Mahila Mahavidyalaya, girls college under

How to cite this paper: Deepika Saxena, R.K. Chauhan, Ramesh Kait,"Dynamic Fair Priority Optimization Task Scheduling Algorithm in Cloud Computing: Concepts and Implementations", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.2, pp.41-48, 2016.DOI: 10.5815/ijcnis.2016.02.05