Modified Particle Swarm Optimization based upon Task categorization in Cloud Environment

Neha Miglani, Gaurav Sharma

Abstract: Cloud Computing has become a spearhead in the field of industries and academia. As far as the IT Industry is concerned, it is pioneering the peculiar domains of clustering, virtualization and grid computing. Traditionally, the complex computation nowadays, demands abundance of resources and computing facilities to perform operational tasks. Cloud computing provides user a new wave in procuring available resources. To scale up the capacity, task scheduling has been emerged as one of the key features of Cloud Computing. Though it is considered as NP-Hard problem, yet numerous researchers and authors have tried to reap out the effective and implementable results for scheduling of tasks to different virtual machines. Meta-heuristic techniques have been embedded to obtain nearly optimal results in the previous studies, still loopholes are lying in the consideration of multiple QoS parameters. In this paper, PSO approach has been modified by manipulating parameters based on the QoS factors from the very initial stage. Instead of considering the population randomly, MIPS and Bandwidth factors have been inculcated to refine and adjust the parametric structure as well as for balancing the load more efficiently. The experimental setup shows that the proposed algorithm works fairly well in assigning the upcoming tasks, henceforth, resulting in reduction of execution time as well.

Index Terms: Cloud Computing, Load Balancing, Particle Swarm Optimization, Quality of Service, Task Scheduling

I. INTRODUCTION

Scheduling involves an optimal resource allocation in definite time duration to attain desired QoS values. The resource scheduling is done keeping in mind various constraints, eventually resulting in the optimized value of targeted objective function. The main concern is to allocate the upcoming task to available resources in such a manner that all resources are utilized efficiently and effectively [1]. Formally, task scheduling algorithm focuses more on efficiency in spite of how fairly the resources have been allocated, which further results in the over-utilization of resources having higher capability of computation [2]. This field has always remained an active area of research for researchers , whether it is in the terms of job scheduling in an operating system or it is the domain of distributed computing namely, grid computing, parallel computing or cloud computing or cluster computing concerning the field of task scheduling.

Amongst all the various types of distributed computing paradigm, cloud computing is the one which has become crucial and has become more or less the buzz word in the IT world. Numerous factors namely, computation power, cost savvy, storage space, security, increased collaboration, scalability and information sharing etc., contributed this domain to grab such attention.

Cloud Computing, being a buzzword in the market, is definitely leading to a new horizon for bringing the evolution and changes in several concepts such as IT management, virtualization, grid as well as distributed computing environment, resulting as the solution for scalability, reliability and security concerns of the users [33]. As depicted in Fig. 1, it is an Information Technology resource [34][35] comprising of applications, infrastructure, storage, services such as IaaS, PaaS, Saas and so on, those are not locally available, yet leveraged by an Enterprise over the Internet. This computing paradigm has dynamicity as well as it is an Internet-based service on an on-demand basis, nevertheless, different organisations have different needs, therefore, cloud is always subjected to various Quality of Service parameters [28-31] and the concept of Load Balancing [3][4].

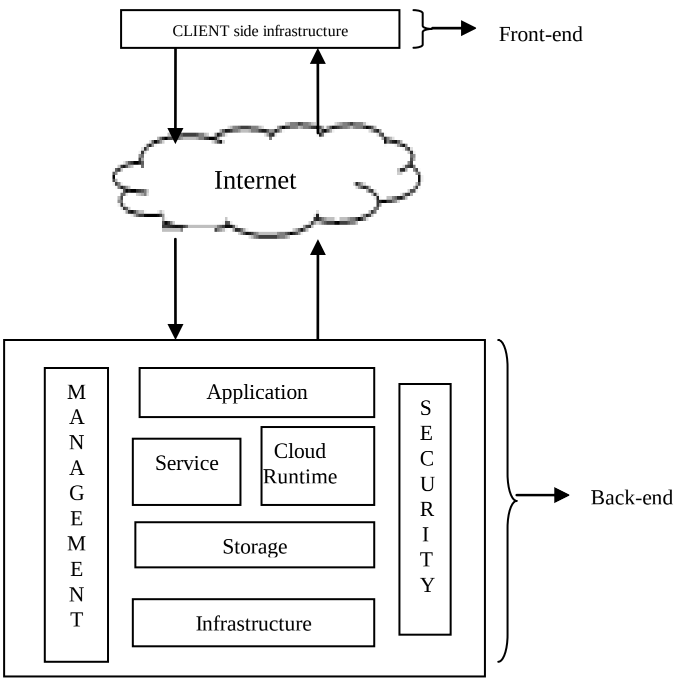

Fig.1: Cloud Computing Basic Architecture

Revised Manuscript Received on December 22, 2018.

Neha Miglani, Department of Computer Engineering, National Institute of Technology, Kurukshetra, Haryana, India.

Dr. Gaurav Sharma, Department of Computer Science and Engineering, Seth Jai Parkash Mukand Lal Institute of Engineering and Technology, Radaur, Kurukshetra University, Kurukshetra, Haryana, India

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Different users have different needs, such as, one user wants to get facilitated of low budget project, and another user wants to have higher bandwidth, yet another desires high processing speed and so on. To cater such varying needs, it is important to allocate all the respective jobs to the available resources in the best possible manner [5]. There are numerous challenges in the field of cloud computing as far as quality services are concerned; reason being cloud environment is somewhat a new business model to work upon. There are many issues related to quality, namely, performance issues, security issues, usability issues, reliability issues and management issues as well [32]. Though it is difficult to target all the issues at once, yet one can focus around the challenge as per the problem being considered to be optimized.

Although this problem of resource allocation has diverted many researchers yet it is difficult to reach to an optimum solution. There does not exist any algorithm or approach which can solve the problem in polynomial time, yet meta-heuristic techniques can be used to reach nearly optimal solution within reasonable time span [6].

In this paper, task scheduling approach has been proposed taking into consideration QoS Factors and Load Balancing using meta-heuristic technique, Particle Swarm Optimization [7]. Highlights of the work can be summarized as:

- Despite of considering the random initial population, tasks have been prioritized to well-utilize the available resource pool.
- Heuristic improvement in PSO algorithm is done by modifying the parameters considering different QoS factors.

Quality of Service is not a universal term, instead it is an application-specific term, which varies depending upon application in hand, such as packet delay is acceptable in e-mails whereas unacceptable as far as video conferencing is concerned. Thus far, the crucial parameters affecting QoS are- time, cost and reliability. As such, cost and time are inversely proportional to each-other; if a project can be allocated more funds, thus an advanced equipment and tools may be used, resulting in lesser time for its completion and vice-versa. This paper aims at balancing the cost and time factor by using job categorization, henceforth, resulting in improved reliability as well by reducing the failure rates. This has been done by allocating jobs to the virtual machines considering their priorities for bandwidth or MIPS rate. By allocation of resources as per user needs would definitely raise the QoS co-efficient values.

Structure of this paper is as follows: Section 2 is focussed on the discussion of related research work been done in the past for scheduling the tasks in hand. Section 3 illustrates the concept of Particle Swarm Optimization and its parametric structure. The proposed methodology has been elaborated in the Section 4, eventually leading to evaluation and simulation of proposed approach done in Section 5. Research is concluded and future scope for improvement of the work presented is highlighted in Section 6.

II. RELATED WORK

Cloud Computing is the way of delivering numerous computing services such as data recovery, security, database, software over the Internet, known as cloud[17][19]. The main problem lies in allocating available resources to various computing tasks. Although the problem is NP-Complete yet many researchers have focused on optimizing the resource allocation by utilizing different meta-heuristic approaches [13-16] [18].

Rodriguez and Buyya et. al [8] worked up on the task scheduling strategy while considering different constraints such as deadline and cost minimization using meta-heuristic optimization technique. Various parameters namely, heterogeneity of resources, dynamic variations and performance of Virtual Machines available were considered.

Pandey et al [11] illustrated a scheduling approach for cloud resources using Particle Swarm Optimization Technique while considering computation cost and data transmission cost. The proposed approach of cost saving using PSO was compared with existing Best Resource Selection (BRS) algorithm and it was analysed that PSO achieved far better results in comparison to BRS, along with an effective load balancing factor.

Cost factor always plays a crucial role in the cloud environment; due to high communication costs, it becomes difficult to apply task scheduling strategies in a large scale environment. For coping up with such factors, Sridhar et. al [12] formulated a hybrid Particle Swarm Optimization (PSO) resulting in improved average length of schedule designed and execution ratio of the task.

Solmaz et. al [20] considered and compared three heuristic approaches, Genetic Algorithm, PSO and modified PSO Algorithm to improvise the scheduling of tasks to different resources in cloud environment. All the three algorithms were targeted to yield an optimized scheduling of tasks to reduce the overall execution time.

Yang et. al [23] illustrated task scheduling algorithm based on QoS-DPSO by considering multi-dimensional QoS requirements. Berger model was introduced in the paper to make judgements regarding fairness of the allocated resources. They also improved PSO Algorithm by altering the parameters in a dynamic manner and using the discrete method of coding position.

Upadhyaya et. al [28] proposed an innovative QoS Model for defining the quality parameters for e-learning applications in cloud computing to bridge the gap between the users' expectations and services provided to them.

Mondal et. al [29] developed the load balancing approach in cloud computing environment along with the enhancement in Quality factors based upon the Fuzzy logic. Processor speed and load acted as an input to the approach and eventual attainment of balanced load was the output while taking into consideration the QoS parameters.

Abdallah et. al [30] focussed the Quality parameters for the Web services based on SaaS. In this paper, framework was also developed for the verification of whether the delivered services comply with the Service Level Agreement of cloud providers.

Awad et. al [24] developed mathematical model using Load Balancing Mutation Particle Swarm optimization (LBMPSO) based allocation and schedule for cloud computing taking into consideration the reliability and availability of the cloud environment.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The proposed approach was evaluated and then compared with the existing algorithms, random algorithm, standard PSO and LCFP algorithm and it was showed that considered factors resulted in better and optimized values

Different meta heuristic approaches namely, PSO [21], GA [10], ACO [22] exists for an optimum allocation of resources. Zhang et. al [9] compared the two approaches Genetic Algorithm and Particle Swarm Optimization and deduced that PSO obtained better results in comparison to GA in Grid Computing [26] [27]. Also, Quality of solution obtained and speed of convergence was also effective in case of PSO.

III. PARTICLE SWARM OPTIMIZATION APPROACH

The Particle Swarm Optimization is a meta-heuristic approach, which actually took its inspiration from the behaviour of animals and eventually worked well with optimization problems. This algorithm was originally developed by Dr. Kennedy and Eberhart in 1995 [25], and is analogous to the evolutionary computation techniques such as Genetic Algorithm. The concept lies in that it mimics the social behaviour of particles in terms of flocking and swarming phenomena. This approach simulates the behaviour of bird flock or group of fishes to reach the desired location. It works by considering the initial population (known as swarm) of random solutions (dubbed, particles). In initial stage, there is randomization considered for the generation of particles. Their movement in the initial space depends primarily on their current position and velocity vector, which further decides their personal best position (p_{best}) and global best position (g_{best}) for the entire population (swarm). After every next iteration, every individual particle updates their position as well as velocity vector to reach an optimum value. There would be a problem specific fitness value, in context of which all the particles make updation. For k^{th} iteration, particles would be updating its parameters based on the following relations:

$$X^i_{k+1} = X^i_k + V^i_{k+1} \quad (1)$$

Resulting in updated position of i^{th} particle after k^{th} iteration and the velocity with which the particle updates its position is:

$$V^k_{i+1} = w_k V^i_k + c_1 r_1 (p_{best}{}^i_k - X^i_k) + c_2 r_r (g_{best} - X^i_k) \quad (2)$$

In equation (2), w is inertia weight whose value should be chosen wisely; as if w chosen would be large, the particle might go out of population space, and if w is small, it would take longer to yield optimum results in a specified duration. c1, c2 are acceleration co-efficients, usually consists of value equals 2, r1, r2 are random numbers lying in the range [0, 1]. To provide consistency in the results, particle velocity is bounded with the limits, that is, $V_i \in [-V_{max}, +V_{max}]$ such that particle should not leave the bounded space.

IV. SOLUTION FRAMEWORK

A. Problem Definition: As far as Cloud data centers are concerned, working model can be categorized based upon the concept of Virtual Machines. VMs can further be split on the basis of task being computation-intensive, communication-intensive, cost factor, performance and numerous other parameters. Depending upon the requirement of the problem-domain for the proposed approach, the factors selected are: Communication-intensive jobs and Computation-intensive jobs. Assume there are n VMs available in the Cloud Data Center, namely {M_i, for i=1, 2, 3...n}. The three attributes used to represent the parameters are: {$M_{seq}M_{CI}, M_{CPI}$} where M_{seq} represents Virtual Machine sequence number, M_{CI} signifying the communication-intensive machines and M_{CPI} denoting the computation-intensive machines, respectively.

A. PSO-based Strategy Formulation

A.1 Particle Initialization

The standard PSO builds the initial population space randomly. In the proposed approach, the initial population is categorized on the basis of two factors:

i. Communication-intensive: Numerous upcoming tasks relies heavily on the need of high bandwidth, therefore, such tasks have been prioritized on the basis of bandwidth requirement.

ii. Computation-intensive: When the cloud user demands that the task should get executed in the minimum time duration, henceforth, demanding high processing speed.

In this manner, Virtual Machines have been categorized logically; as and when the new task comes, depending upon the needs of user, task moves to the TaskList of either communication-based VMs or Computation-based VMs. This way, the tasks categorization is achieved.

Algorithm: PSO-based Task Scheduler.
Input: TaskList(),Virtual Machines(M_i), Bandwidth, MIPS, Acceleration Co-efficient c_1,c_2, random numbers r_1,r_2, inertia weight w, newtask().
Output: Optimized task scheduling along with an improved execution time.

1. Initialize the upcoming newtask(), TaskList(), acceleration co-efficient c1, c2; inertia weight w and random numbers r1, r2, gbest and pbest.
2. Logically categorize the VMs (particles) on the basis of bandwidth (communication-intensive) and MIPS (computation-intensive).
3. For each upcoming newtask()
   If (priority(bandwidth)>priority(MIPS))
      Chosen_particle= Fitness (bandwidth)
   else
      Chosen_particle=Fitness(MIPS)
4. Submit the newtask() to the Chosen_particle.
5. //Make parameter updations
   i) update the position of respective particle based on (1).
   ii) update the Velocity Vector using (2).
6. Repeat steps 2-5 until the termination criterion is met.
7. Results so obtained are optimized.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Fitness Function

This paper is focussed on the reduction of total time spent for the execution of assigned job. Henceforth, the objective lies in the reduction of makespan time (Bandwidth-based/ MIPS-based), that is, for every individual machine, fetching the execution time for completion of all the jobs assigned to that machine. Smallest amongst all is the fitness value of that particle.

Algorithm: Fitness Function Calculation
Input: Virtual Machines, newtask(), Bandwidth, MIPS
Output: Category-specific fitness obtained.

Fitness (X)
1) For every individual particle from the logical category X
    - calculate fitness value (in terms of makespan time), p_{best} of particle i
    - if p_{best} (of i^{th} particle)<g_{best}
      update g_{best}.(for entire population for the category X)
2) return particle with lowest p_{best} value (which would be the g_{best} value for entire population of category X).

A.3 Updation of position and velocity vector

As the fitness function is oriented towards finding the smallest value of makespan, therefore, each Virtual Machine has its own personal best (p_{best}) and eventually, yielding the global best (g_{best}), by comparison of the smallest fitness values of all individual machines. Particle velocity has been updated using (2).

Position of Virtual machine can be updated considering updated velocity vector and previous position of that respective particle, and thus, can be modified using (1).

A.4 Termination criteria

There are no fixed rules or convergence criteria for PSO Approach; it all depends upon the problem specification. In this proposed technique, when all the upcoming tasks get exhausted, the process immediately comes to hault. This has been considered as the convergence criteria.

After the completion of the proposed technique, the upcoming tasks get allocated to the available Virtual Machines in an optimized manner and reduced execution time as well. Precisely, algorithm flow is described in Fig. 2.

V. EXPERIMENTAL SIMULATION

The proposed approach has been verified and validated on the Cloudsim Software. The proposed algorithm has been embedded in the Cloudsim layer itself. 10 Virtual Machines, each VM comprising memory 8G, CPU: Processor 3.20GHZ and 30 GB Hard Disk, have been considered for performance evaluation of the proposed approach. The key parameter considered to formulate efficiency measure is the makespan time for scheduling tasks on available resources.

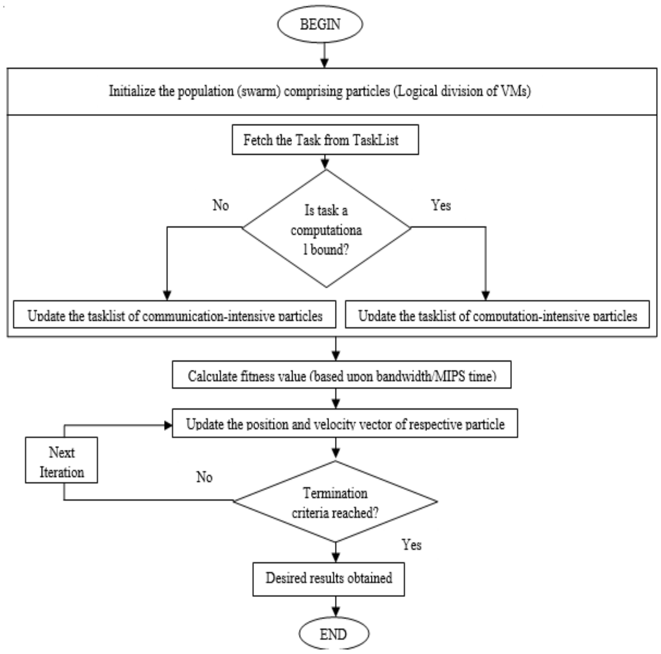

Fig 2: Flowchart for PSO-based Task- Scheduler

Based upon the varying needs of upcoming tasks, resource parameters are grounded on the basis of range, instead of precisely fixing the definite value as described in Table I along with the parameter specifications for the approach being used, PSO highlighted in Table II.

Table I: Particles and Cloudlet Specifications

Parameter	Specified Range	Parameter	Specified Range
Particles (Virtual Machines)		***Tasks***	
Number of VMs	10	*Length Range*	5000-14000
MIPS	300-800	*Number of tasks*	1000
Bandwidth	600-1200		

Table II: Evaluation Parameters for Proposed Algorithm

Parameter considered	Experimental Value
Population Size	50
Inertia weight, w	0.72
Acceleration co-efficient, c_1	2
Acceleration co-efficient, c_2	2
r_1	0.58
r_2	0.63
Maximum number of Iterations	150

After performing the maximum number of iterations, the results have been evaluated and stored with respect to the makespan time considered as an efficiency parameter. Besides considering the results of proposed approach; two approaches, namely, Min-Min Approach and Existing PSO have been taken into account for making a comparative analysis amongst the existing approaches and the proposed one.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table III: Comparative Analysis of Makespan Time for different Algorithms

Sequence No.	Algorithms	Number of Tasks	Execution time (Makespan)
1	Min-Min Algo	200	595
	Existing PSO		587
	Hybrid PSO		580.5
2	Min-Min Algo	400	1150
	Existing PSO		1098
	Hybrid PSO		990
3	Min-Min Algo	600	2080
	Existing PSO		2053
	Hybrid PSO		2019
4	Min-Min Algo	800	2845
	Existing PSO		2790
	Hybrid PSO		2716
5	Min-Min Algo	1000	3556
	Existing PSO		3491
	Hybrid PSO		3406

Table III. depicts the comparison of makespan time being calculated for the three approaches while fluctuating the number of tasks from 200 to 1000; and results are quite descriptive in order to make a conclusion that the proposed hybrid PSO outperformed the two approaches considered in such a manner that lower the execution time, higher is the efficiency and the same has been described in the fig. 3 in a graphical format.

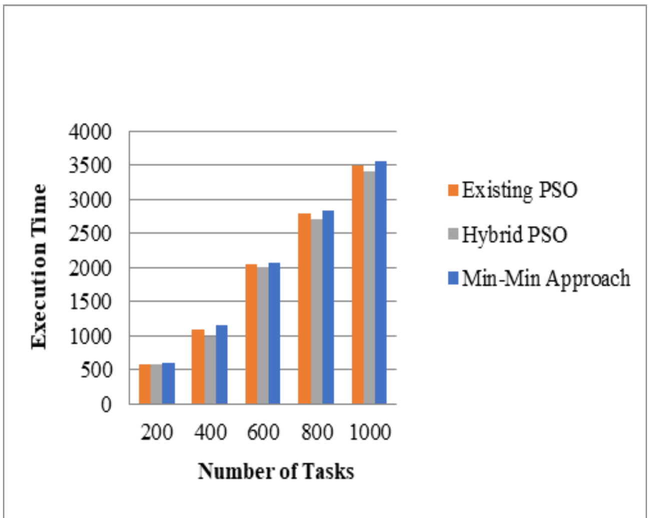

Fig.3: Experimental comparison for Execution Time for varying number of tasks

Alongwith the execution time, success ratio is also measured in order to affirm the integrity of proposed approach, which is provided in Table IV and Fig. 4.

Table IV: Comparative Analysis of Success Ratio for different Algorithms

Number of Tasks	Min-Min Approach	Existing PSO	Hybrid PSO (Proposed Approach)
200	0.87	0.9	0.94
400	0.85	0.88	0.91
600	0.86	0.86	0.9
800	0.83	0.85	0.86
1000	0.81	0.84	0.85

Altogether, it can be summarized that the proposed approach can effectively schedule the tasks and is able to provide higher success ration and well-optimized results.

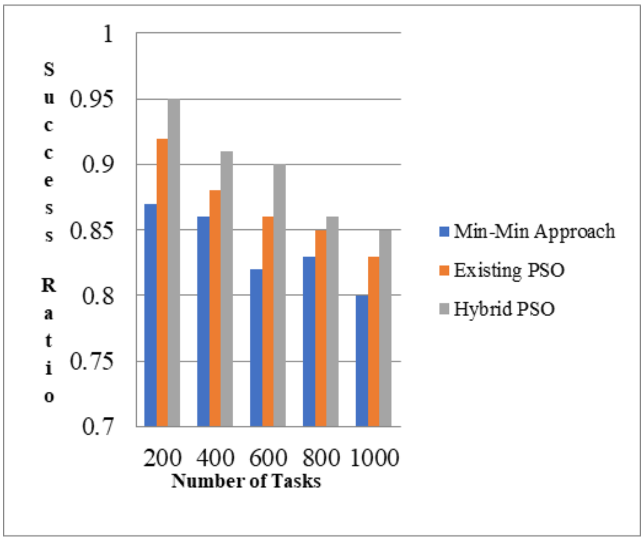

Fig. 4: Success Ratio for varying number of tasks

VI. CONCLUSION AND FUTURE SCOPE

The proposed approach has definitely helped overcoming the difficulties lying in scheduling of numerous tasks as well as balancing the load effectively. PSO is utilized by altering the algorithm by embedding QoS factors into it. In-spite of allocating tasks to VMs in random fashion, tasks were categorized based on their priorities of being communication or computation-intensive. Experimental results clearly visualized the efficiency and performance of proposed approach in terms of reduced makespan time as well as increased success ratio. Henceforth, it can be summarized that this approach is able to bring the optimized results in terms of execution time, bandwidth and processing speed, and reduction in cumulative cost factor as well.

The current approach may be extended and elaborated in future to bring out more efficient results. Though Bandwidth and MIPS has been considered in the approach, yet the scope of amortized results is there, that is, despite of taking the raw value, it can be normalized in context of collaborative value for the respective factor so that results obtained would be more precise and justified.

REFERENCES

1. Karger D, Stein C, Wein J. Scheduling Algorithms. “Algorithms and Theory of Computation Handbook: special topics and techniques”. Chapman & Hall/CRC; 2010.
2. M.A. Arfeen, K. Pawlikowski, A. Willig. “A Framework for Resource Allocation Strategies in Cloud Computing Environment”. Computer Software and Applications Conference Workshops (COMPSACW), IEEE 35th Annual, 2011, pp. 261 -266.
3. Salim Bitam, “Bees Life algorithms for job scheduling in cloud computing”, International Conference on computing and Information Technology, 2012.
4. Saeed Parsa and Reza Entezari-Maleki, “RASA: A New Grid Task Scheduling Algorithm”, International Journal of Digital Content Technology and its Applications, Vol.3, pp. 91-99, 2009.
5. Mala Kalra and Sarbjeet Singh, ”A review of metaheuristic scheduling techniques in cloud computing”, Egyptian Informatics Journal (2015) 16, 275-295.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

6. Talbi EG. “Metaheuristics: from Design to Implementation” Wiley; 2009.
7. Hemlata S. Urade and Prof. Rahila Patel, “Study and Analysis of Particle Swarm Optimization: A Review” 2nd National Conference on Information and Communication Technology (NCICT) 2011.
8. Rodriguez M A, Buyya R, "Deadline based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds [J]", IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp: 222-235, 2014.
9. Zhang L, Chen Y, Sun R, et al., "A Task Scheduling Algorithm based on PSO for Grid Computing [J]", International Journal of Computational Intelligence Research, vol. 4, no. 1, pp: 37-43, 2008.
10. Tirado J M, Higuero D, Isaila F, et al., "Predictive Data Grouping and Placement for Cloud-based Elastic Server Infrastructures [C]", Proceedings of the 2011 11th IEEE ACM International Symposium on Cluster, Cloud and Grid Computing, pp: 285-294, 2011.
11. Pandey, S., Wu, L., Guru, S. M., & Buyya, R., “A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments”, IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 400-40, 2010.
12. M.Sridhar and G. Rama Mohan Babu,” Hybrid Particle Swarm Optimization Scheduling for Cloud Computing”, International Advance Computing Conference (IACC), 978-1-4799-8047-5/15, 2015 IEEE.
13. HE Hua, XU Guangquan, PANG Shanchen, ZHAO Zenghua, “AMTS: Adaptive Multi-Objective Task Scheduling Strategy in Cloud Computing” China Communications, April, 2016.
14. Baliga J, Ayre R W A, Hinton K, et al. "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport [Jr, Proceedings of the IEEE, vol. 99, no. 1, pp: 149-167, 2011.
15. Kumar P, Verma A, "Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm [Jr, International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 5, pp: 111-114, 2012.
16. T. Chen, B. Zhang, X. Hao, Y. Dai, “Task scheduling in grid based on particle swarm optimization”, The Fifth International Symposium on Parallel and Distributed Computing, ISPDC '06. pp. 238-245, 2006.
17. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, “Cloud Computing and Emerging IT Platforms”, Vision, Hype, and Reality for Delivering Computing as the 5th Utility, Future Generation Computer Systems 25(6), 599–616 (2009).
18. Baomin Xu, Chunyan Zhao, Enzhao Hu, Bin Hu, “Job scheduling algorithm based on Berger model in cloud environment”, Advances in Engineering Software 42, PP. 419–425,2011.
19. M.A. Arfeen, K. Pawlikowski, A. Willig. “A Framework for Resource Allocation Strategies in Cloud Computing Environment. Computer Software and Applications”, Conference Workshops (COMPSACW), IEEE 35th Annual, 2011, pp. 261 -266.
20. Solmaz Abdi, Seyyed Ahmad Motamedi, and Saeed Sharifian, “Task Scheduling using Modified PSO Algorithm in Cloud Computing Environment”, International Conference on Machine Learning, Electrical and Mechanical Engineering (ICMLEME'2014) Jan. 8-9, 2014 Dubai (UAE).
21. Kennedy, J., “Particle swarm optimization”, In Encyclopedia of Machine Learning, Springer US, pp. 760-766, 2010.
22. B. K. Nanda, G. Das, "Ant colony optimization: a computational intelligence technique", *Int. J. Comput. Commmun. Technol*, vol. 2, no. 6, pp. 105-110, 2011.
23. Anqi Xu, Yang Yang, Zhenqiang Mi and Zenggang Xiong,” Task scheduling algorithm based on PSO in cloud environment”, DOI 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015, 978-1-4673-7211-4/15 IEEE.
24. A.I.Awad, N.A.El-Hefnawy, H.M.Abdel_kader, “Enhanced Particle Swarm Optimization For Task Scheduling In Cloud Computing Environments”, International Conference on Communication, Management and Information Technology (ICCMIT) Procedia Computer Science 65 (2015) 920 – 929.
25. James Kennedy and Russel Eberhart” Particle Swarm Intelligence”, IEEE 1995.
26. Rajesh K. Bawa and Gaurav Sharma, “Modified Min –Min Heuristic for Job Scheduling based on QoS in Grid Environment”. IEEE Conference on International Management in the Knowledge Economy, 19-20 December 2013.
27. Rajesh K. Bawa and Gaurav Sharma, "Reliable resource selection in grid environment”. International Journal of Grid Computing & Applications, March 2012, Volume 3, Number 1, pp. 1-10.
28. Upadhyaya Jolly and Ahuja Neelu Jyoti, “Quality of Service in Cloud Computing in Higher Education: A Critical Survey and Innovative Model” International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2017) 978-1-5090-3243-3/17 2017 IEEE.
29. Mondal Himadri Shekhar, Hasan Md. Tariq, Karmokar Taposh Kumar, Sarker Shamlendu, “Improving Quality of Service in cloud computing architecture using Fuzzy Logic” Proceedings of the 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), 978-1-5386-0869-2/17 2017 IEEE.
30. Abdallah A.Z.A. Ibrahim, Sebastien Varrette and Pascal Bouvry, “On Verifying and Assuring the Cloud SLA by Evaluating the Performance of SaaS Web Services across Multi-Cloud Providers” 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops 2325-6664/18 2018 IEEE DOI 10.1109/DSN-W.2018.00034.
31. Halabi, Talal, and Martine Bellaiche. "Evaluation and selection of Cloud security services based on Multi-Criteria Analysis MCA." Computing, Networking and Communications (ICNC), 2017 International Conference on. IEEE, 2017.
32. Shan Luo, Yanhui Zhou, “How to Guarantee the Cloud Services Quality” 978-1-4673-9904-3/16 2016lEEE.
33. Huimin Zhang, Xiaolong Yang, “Cloud Computing Architecture based-on SOA” 2012 Fifth International Symposium on Computational Intelligence and Design 978-0-7695-4811-1/12 IEEE
34. David S. Linthicum, Cloud Computing and SOA Convergence in Your Enterprise. RR Donnelley, Crawfordsville, Indiana. September 2009.
35. Niranjan Lal, Dr. S Qamar, Mrityunjay Singh” Internet-ware cloud computing Challenges” International Journal of Computer Science and Information Security (IJCSIS)- 1947-5500, Vol. 7, No. 3, pp. 206-210, 2010.