International Journal of Advanced Technology in Engineering and Science
Vol. No.5, Issue No. 06, June 2017
www.ijates.com
ijates
ISSN 2348 - 7550

Task Scheduling Using Hybrid PSO in Cloud Based Environment

Anjanee Mourya[1],Bhupesh Kumar Dewangan[2]

Computer Science & Engineering, Chhatrapati Shivaji Institute of Technology Durg, (India)

Abstract: Cloud computing ensures access to shared resources and ordinary infrastructure, providing services on concern over a community for operations to fulfil changing industrial enterprise requirements. Scheduling may be a prominent activity that's finished during a cloud computing surroundings. To increase cloud computing work load performance, responsibilities programming is accomplished to induce most financial profit. In cloud, excessive communication cost prevents assignment schedulers from being distributed in immense scale assigned environments. This study proposes a hybrid improvement primarily based entirely on Particle Swarm improvement (PSO) and Genetic algorithmic program (GA) for programming in cloud environments.

***Keywords*: cloud computing, scheduling, particle swarm optimization (PSO), genetic algorithm (GA).**

I. INTRODUCTION

Cloud computing is a new computing generation to enhance the virtualized sources designed for quit clients in a dynamic environment so as to provide reliable and relied on service [1]. The provided services includes however not restrained to the opportunity of building application & special sever a offerings through net virtualization which is the essential approach which improve physical belongings utilization in cloud computing [2]. It lets in abstraction and isolation of underlying bodily assets and reduce required hardware device. establish an green load balanced algorithm should be proposed to make sure cloud computing have to be used correctly & powerfully which is a important reason of the delivery carrier. There is an actual want to make bigger new project scheduling algorithm to fulfil the virtualization ideas & demand. The intention of the assignment scheduling algorithm is international journal of Grid Distribution Computing to reap immoderate tool throughput, enhance the load balance and lessen the final touch period at the equal time as ensuing in the identical time meeting the task requirement with available virtualized sources. Constant with the undertaking scheduling, a set of the perfect wide variety of duties is to be scheduled to the virtual machines. Task scheduling over the Cloud Computing belongings are the most essential undertaking due to the fact the consumer will have to pay for aid use on the premise the time

Evolutionary algorithms are based on species origin. Examples are Particle Swarm Optimization (PSO) and Genetic algorithm (GA). PSO is a parallel evolutionary computation method and a heuristic seek method stimulated by way of biological populations swarming behaviour [4]. The usage of PSO ensures an excellent performance. GA is a search heuristic that mimics natural evolution. It is mechanically used to generate useful answers for optimization and search problems. GAs belongs to a larger class of evolutionary algorithms,

producing answers to optimization issues with techniques from herbal evolution like inheritance, selection, mutation, and crossover [5]. PSO algorithm has many blessings like easy realization, excessive flexibility, strong robustness, and scalability because of which it solves many combinational issues. But, its negative aspects are low convergence charge when fixing big scale optimization troubles and easily sinking into nearby optima due its randomness [6]. PSO is ideal in a preliminary section however while going via iterations convergence charge turns into low and particles lose variety. There is want for an algorithm to offset these troubles and so this study proposed a hybrid algorithm in which PSO combines with GA i.e. GAPSO algorithm ensuring higher effects because of the houses of each. The final sections of this paper are prepared as follows: section 2 opinions related works, section 3 explains the methodology. Section 4 discusses experimental consequences and section 5 concludes the work.

II. RELATED WORK

(Zhifeng Zhong.et al. 2016) [7] This paper introduces a greedy Particle Swarm Optimization (G&PSO) primarily based algorithm to remedy the assignment scheduling hassle. It makes use of a grasping algorithm to quickly remedy the initial particle cost of a particle swarm optimization algorithm derived from a virtual machine-based cloud platform. The archived experimental consequences show that the algorithm exhibits higher overall performance including a quicker convergence fee, more potent nearby and international search capabilities, and an extra balanced workload on each virtual device. Consequently, the G&PSO algorithm demonstrates stepped forward virtual system performance and useful resource utilization compared with the traditional particle swarm optimization algorithm.

(Xu. Panpan et al. 2015) [17] Genetic algorithm is widely utilized in optimization troubles for its outstanding global seek approach and fairly parallel processing approach; Simulated annealing algorithm can keep away from the quest process falling into local most useful. A hybrid genetic algorithm based on simulated annealing is designed via combining the benefits of genetic algorithm and simulated annealing algorithm. The numerical experiment represents the hybrid genetic algorithm may be applied to solve the characteristic optimization troubles effectively.

(I. C. Trelea 2003) [15] the particle swarm optimization algorithm is analyzed the use of preferred results from the dynamic machine principle. Graphical parameter selection pointers are derived. The exploration–exploitation tradeoffs is mentioned and illustrated. Examples of overall performance on benchmark capabilities superior to formerly posted effects are given.

Keng Mao Cho[9]This paper combines ant colony optimization and particle swarm optimization to resolve the VM scheduling trouble, with the result being called ant colony optimization with particle swarm (ACOPS). ACOPS uses ancient information is expecting the workload of latest enter requests to adapt to dynamic environments without extra task facts. ACOPS also rejects requests that cannot be satisfied before scheduling to reduce the computing time of the scheduling technique.

Al-maamari, F. Omara.[20] have proposed a venture scheduling algorithm for cloud computing based totally on a merge PSO algorithm with the Cuckoo seek (CS) algorithm, called (PSOCS), the task assigned to the virtual device, that goals to minimizes make span and the most useful resource usage. In

keeping with the work in a paper, the Particle Swarm Optimization has been involved to optimize the assignment scheduling hassle with focusing on minimizing the overall executing time.

III. METHODOLOGY

3.1 PSO Algorithm

The PSO algorithm first proposed by using the use of Eberhart and Kennedy in 1995, and its easy concept is based totally on a look at of birds foraging behaviour [8]. Therefore, PSO algorithm turns out to be inspired by the use of the behavioural trends of natural organization and in the end has been used to clear up and optimize problem. The critical purpose is to reduce the entire completion time and to reduce the load balance within the cloud computing machine

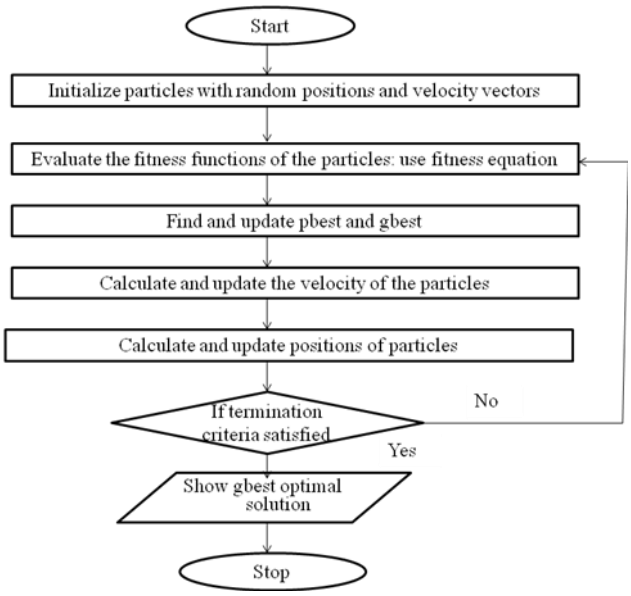

Fig 1 Proposed PSO algorithm

3.2 PSO algorithm flow chart- Figure contains the flow of the PSO algorithm from starting to end. It is starting from the initialization of particle

- S= size of swarm particle
- T= it is a number of tasks
- M= it represents number of virtual machine

And then it evaluates fitness value by using the fitness formula:

$$\text{fitness}(i) = \frac{1}{SFT}, 1 \leq i \leq S \qquad (1)$$

Where,

$$SFT = \max_{1 \leq m \leq M}\left(\sum_{n=1}^{k} VM(m,n)\right) \qquad (2)$$

- Fitness Function: The inverse of the total task completion time is used to represent the fitness function, which is a key parameter in cloud computing
- SFT represents the time needed to complete all the tasks;

- VM(m; n)represents the time for the n-th task to run on the m-th virtual machine, and
- K is the number of tasks distributed to this virtual machine.

Then find out the value of pbest and gbest that is local/particle best and global best, by using the equation (3), (4) and (5)

$$pbest_i(t+1) = \begin{cases} pbest_i(t), \text{if } f(p_i(t+1)) \le f(pbest_i(t)); \\ p_i(t+1), \text{if } f(p_i(t+1)) f(pbest_i(t)) \end{cases} \quad (3)$$

$$if(\max(pbest(t)) = getMax(f(pbest_1(t)), f(pbest_2(t)) \dots \dots f(pbest_s(t)) \quad (4)$$

$$gbest(t) = \begin{cases} \max(pbest(t)), \text{if } f(\max(pbest(t)) > f(gbest); \\ gbest, \qquad\qquad \text{else} \end{cases} \quad (5)$$

After finding the 2 high-quality values, the particle updates its speed and positions with following equation (6) and (7).

$$v_i(t+1) = \omega \times v_i(t) + c_1 \times Rand(\ ) \times (pbest_i(t) - p_i(t)) + c_2 \times Rand(\ ) \times (pbest_i(t+1) - p_i(t) \quad (6)$$

$$p_i(t+1) = p_i(t) + v_i(t) \quad (7)$$

- In the above formulas, t represents the number of iterations;
- ω = 0.8 is the inertia weight;
- C1 and c2 are learning factors, and generally c1= c2 =2.
- Rand () is a random value within [0, 1].
- $f(p_i(t))$= denotes particles fitness function
- $f(p_i(t+1))$ = denotes as fitness function during next iteration.

During the process of iteration, the position of the particle is limited to a specific range ($1 \le p_i(t) \le M$) then calculate its velocity then calculate and update position of particles, then terminate if criteria is satisfy and show the gbest optimal value. If criteria is not satisfies then again go back to second process and calculate the fitness value and execute the whole loop and repeat the process till optimise solution achieve

3.3 Genetic Algorithm

Genetic algorithms (GA) are search based totally at the ideas of natural desire and genetics. GAs is a subset of a much larger branch of computation referred to as evolutionary computation.

It does some distance make bigger through John Holland. In GAs, we have a pool or a population of feasible answer to the given problem. Those answers then go through crossover and mutation (like in natural genetics), producing new child, and the system is repeated over numerous generations. Each individual is assigned a

fitness value (based mostly on its intention characteristic cost), and the extra match people are given a higher danger to mate and yield more "greater fitness" particle.

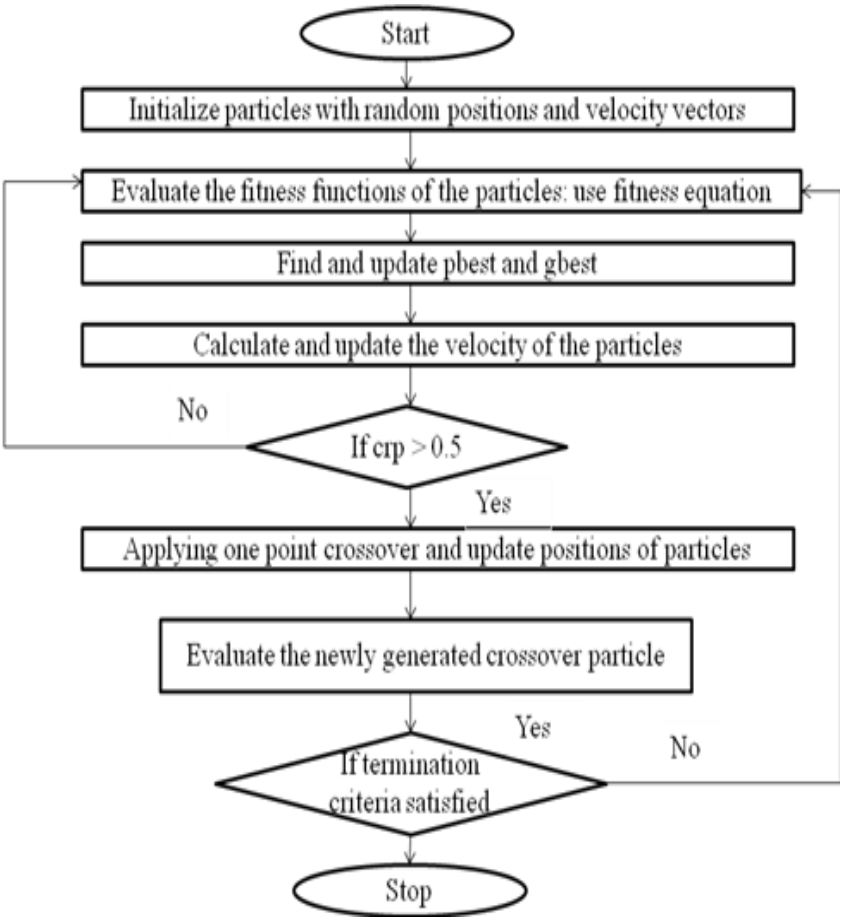

Fig 2 Proposed hybrid PSO&GA algorithm

3.4 Process of Hybrid PSO and Genetic Algorithm

The steps of Hybrid PSO and GA algorithm are as follows: [6]

Step 1 Initialization of the particle swarm. The position and velocity of the particles are first initialized.

Step 2 Calculate each particle's fitness function value using Formulas (1) and (2).

Step 3 Update the optimal. Update the individual and group optimal based on Formulas (3)–(5):

(1) Compare the value of the particle's fitness function to its individual optimal pbest, if the value of the particle's fitness function is better than pbest, and then replace the value of pbest with the current position of the particle.

(2) Compare the particle's fitness function value to its group optimal gbest, if the fitness function value of the particle is better than that of the initial solution calculated by the greedy algorithm, then reset the value of gbest with the particle's current position.

Step 4 Update the velocity and position of the particle using Formulas (6) and (7) respectively.

Step 5 Then use the genetic algorithm crossover function cr /random>0.5, it will perform a crossover operation till the random value is greater than 0.5.

Step 6 Evaluate the newly generated crossover particle to find out the best solution

Step 7 Stop conditions. The loop will return to Step2 until the stop conditions are met.

VI. SIMULATION AND ANALYSIS

To validate the feasibility and performance of the Hybrid PSO and Genetic algorithm in terms of scheduling potential in a cloud, we used the cloud computing simulation platform cloudsim [11], NetBeans 8.2 javase windows is used for the implementation. The computer architecture and operating system of the cloud data centre were x86 and windows7, respectively, where each virtual machine had a 1.2 GHz CPU, 4GB RAM, and 100GB hard drive

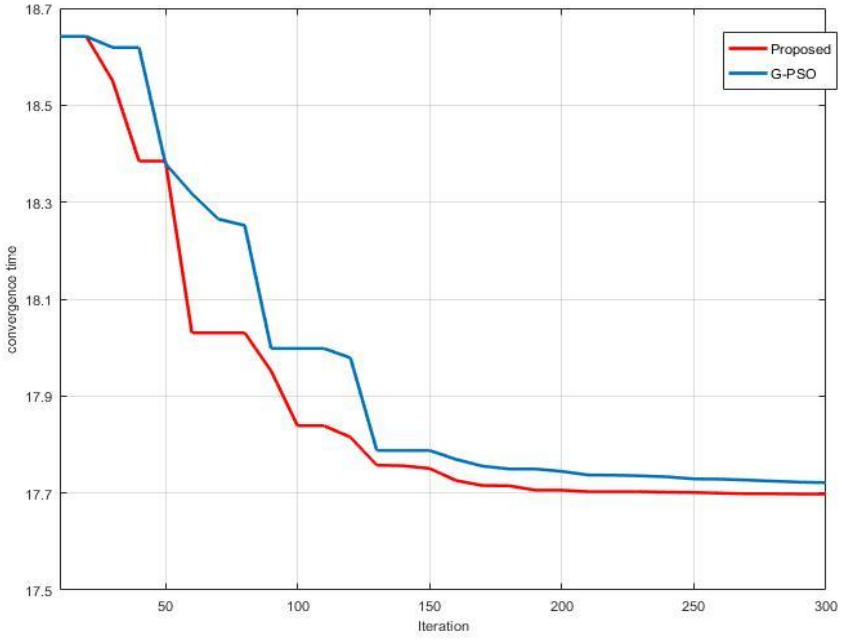

Fig 3 Completion time Vs iteration with 50 tasks

Figures 3 show the entire completing times for 5 virtual machines with 50 iterations. It is able to be seen from Fig 3. whilst the use of the proposed PSO&GA algorithm, the full finishing touch time for the assigned task became 10 s a whole lot less than using the G&PSO algorithm. similarly, the proposed PSO&GA algorithm had less new release, a faster convergence velocity, and much less randomness inside the strategies of optimization for small-or-massive scale venture scheduling. Fig 3 suggests that regardless of the truth that the entire task completion time of the proposed PSO&GA algorithm while scheduling a small-scale challenge is longer than the PSO algorithm on the preliminary stage of generation, the proposed algorithm has a shorter overall undertaking of entirety time. It additionally has a more potent potential for close by seek, because of this that it has, to some extent, conquered the lack of ability of the G&PSO algorithm with its inadequate close by seek

capability. Compared with the G&PSO algorithm, while scheduling a large-scale task, the proposed algorithm suggests a stronger functionality in the optimization procedure and has a higher scheduling effect.

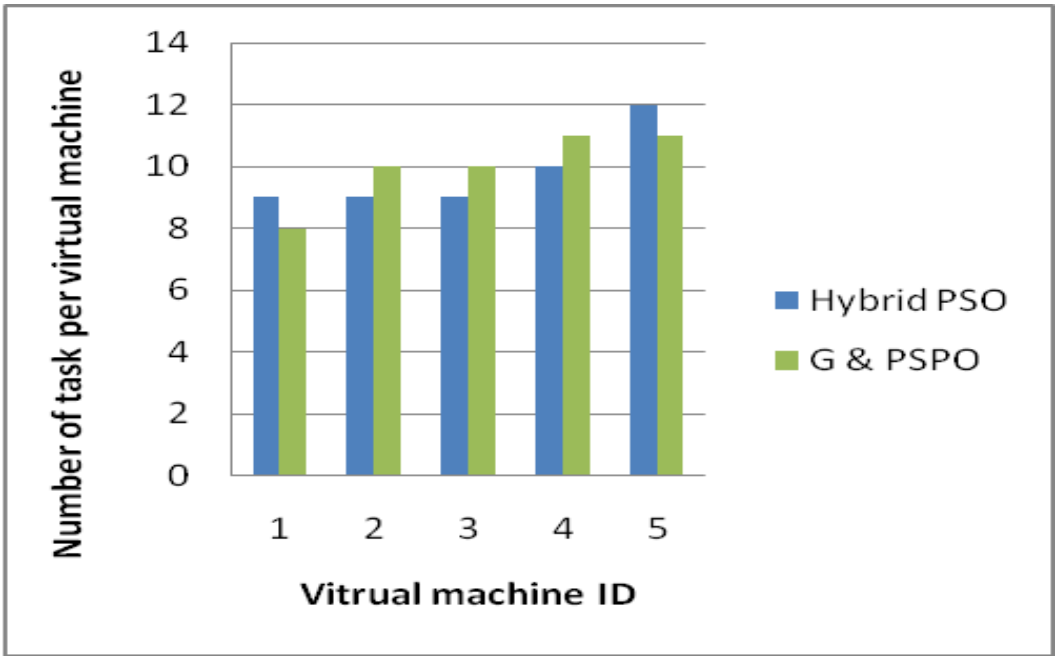

Fig 4 5 Virtual machine with 50 tasks

The wide variety of obligations assigned to each virtual system is proven in Fig.4 (50 tasks). In phrases of usage of the virtual machine aid, as shown in Fig. 4, whilst performing huge-or-small scale task scheduling, the number of obligations assigned to each virtual device is closer to the suggest cost when using the PSO&GA algorithm. This consequences in stepped forward usage of device assets and avoids a workload overload on the virtual machines.

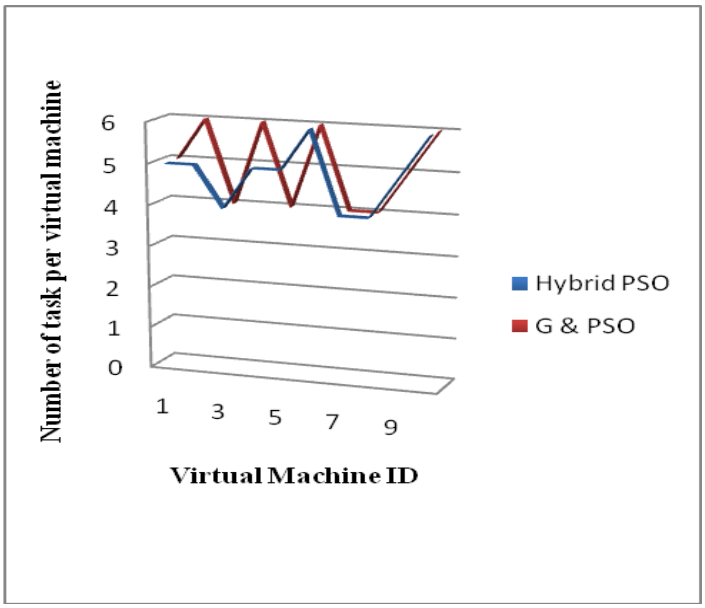

Fig 5, 10 virtual machine with 50 tasks

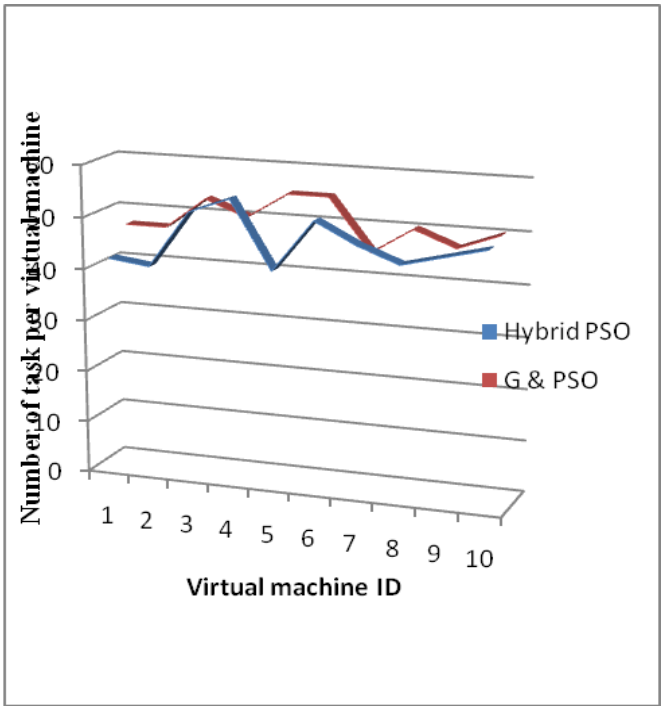

Fig 6, 10 virtual machine with 500 tasks

Figure 5 indicates that PSO&GA algorithm has better load balancing overall performance as compared with the G&PSO algorithm. usually, using more virtual machines does no longer imply acquiring a better result, as configuring every virtual device often consumes extra machine sources and eventually results in a increase in the basic system overall performance. because of the predicament of the physical hardware within the widespread host and community bandwidth, the number of virtual machines assigned to a single host ought to be set to no greater than 10 to obtain the first-class system overall performance. To similarly verify the performance of the proposed G&PSO algorithm of load balancing in virtual machines, the variety of virtual machines turned into improved from 5 to 10 and their processing competencies updated to 500, 600, 700, 800, 900, 1000, 550, 650, 750, and 850. The variety of duties remained unchanged, and those responsibilities are assigned to each virtual device proven in Figs. 5 and 6. In Figs.5 and 6, the simulation effects for each large-scale and small-scale mission scheduling are proven. When the usage of the proposed algorithm the number of tasks assigned to each virtual machine remains toward the mean cost and the device load remains balanced. In conclusion, the proposed G&PSO algorithm achieves the desires of shorter mission of completion time and a greater balanced virtual machine load; the comprehensive efficiency of the cloud computing platform has therefore been advanced.

VII. CONCLUSION

This paper aimed to clear up the task scheduling problems of virtual machines on a cloud platform, and the PSO&GA algorithm was proposed to decrease the same old final time and balance the workload in each virtual system. compared with the G&PSO algorithm, the hybrid PSO&GA algorithm has a faster convergence charge inside the early level of latest release, a stronger local seek functionality all through the later duration of era,

better global optimization average overall performance, and overcomes the incapacity of the traditional algorithm with an awful lot less randomness. On a cloud platform simulated with the aid of Cloudsim (facts centre disposes one server), the proposed algorithm now not only satisfactory reduces the overall task completion time, however additionally balances the system load and improves the complete performance of the whole cloud platform

VIII. REFERENCES

[1.] T. Dillon, C. Wu, and E. Chang, "Cloud computing: issues and challenges", in Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, (2010), pp. 27-33.

[2.] F. Baroncelli, B. Martini and P. Castoldi, "Network virtualization for cloud computing", annals of telecommunications-annales des telecommunications, vol. 65, (2010), pp. 713-721.

[3.] X. Wang, C. S. Yeo, R. Buyya And J. Su, "Optimizing The Makespan And Reliability For Workflow Applications With Reputation And A Look-Ahead Genetic Algorithm", Future Generation Computer Systems, Vol. 27, (2011), Pp. 1124-113.

[4.] Saeed Javanmardi, Mohammad Shojafar, Danilo Amendola, Nicola Cordeschi, Hongbo Liu, And Ajith Abraham, "Hybrid Job Scheduling Algorithm For Cloud Computing Environment", Springer verlag Berlin Heidelberg 2014

[5.] Chun-Yan Liu Cheng-Ming ZOU,Pei WU," A Task Scheduling Algorithm Based On Genetic Algorithm And Ant Colony Optimization In Cloud Computing" In 978-1-4799-4169-8/14 2014 IEEE

[6.] Zhifeng Zhong_, Kun Chen, Xiaojun Zhai, And Shuange Zhou," Virtual Machine-Based Task Scheduling Algorithm In A Cloud Computing Environment" In Issnll1007-0214ll07/09llpp660-667 Volume 21, Number 6, December 2016 Ieee

[7.] Panpan Xu, Shulin Sui, Zongjie Du Application Of Hybrid Genetic Algorithm Based On Simulated Annealing In Function Optimization World Academy Of Science, Engineering And Technology International Journal Of Mathematical, Computational, Physical, Electrical And Computer Engineering Vol:9, No:11, 2015

[8.] I. C.Trelea, The Particle Swarm Optimization Algorithm: Convergence Analysis And Parameter Selection, Information Processing Letters, Vol. 85, No. 6, Pp.317–325, 2003.

[9.] Keng-Mao Cho Pang-Wei Tsai Chun-Wei Tsai Chu-Sing Yang A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing Neural Comput & Applic DOI 10.1007/s00521-014-1804-9 Springer 2014

[10.] A. Al-Maamari And F. Omara, "Task Scheduling Using Hybrid Algorithm In Cloud Computing Environments", Iosr Journal Of Computer Engineering (Iosr-Jce), Vol. 17, (2015), Pp. 96-106.

[11.] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, CloudSim: A toolkit for modelling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *SoftwarePractice & Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[12.] Xu Gaug An adaptive parameter tuning of particle swarm optimization algorithm Elsevier Vol:219 Issue: 9 2013

[13.] S. Kaur And A. Verma, An Efficient Approach To Genetic Algorithm For Task Scheduling In Cloud Computing Environment, International Journal Of Information Technology & Computer Science, Vol. 4, No. 10, Pp. 74–79, 2012.

[14.] H. Liu, A Measurement Study of Server Utilization In Public Clouds, In IEEE Ninth International Conference On Dependable, Autonomic And Secure Computing, 2011, Pp.435–442.