Volume 8, No. 5, May – June 2017

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Hybrid Approach: ACO with Gravity Search to Reduce Makespan and Flowtime for Resource Allocation in Cloud Computing

Navjot kaur
Dept. Of Computer science and technology
Guru Nanak Dev University , Amritsar
Punjab, India

Anil kumar
Dept. Of Computer science and technology
Guru Nanak Dev University Amritsar
Punjab,India

Abstract: Cloud computing is becoming need of the hour and provides resources to the machines that needs it. Cost to the users is on the basis of pay per use. The machines requirement must be satisfied by cloud provider in order to be competitive in current environment. Resource allocation and execution of entire job is critical in this situation. Proposed literature works towards allocating resources and provides mechanism to search for the resources if not available. ACO with gravity search is proposed in this case. Ant colony optimization look for the available resources and if not available GSA is implemented to perform the job. Makespan and Flowtime is considerable reduced in this case.

Keywords: Cloud Computing, Cost, Makespan, Flowtime

I. INTRODUCTION

The proposed system deals with multihueristic scheduling problem. The jobs are to be located to resources and minimal possible cost is needed to be encountered. Hybrid approach of ant colony and gravity search is proposed through this paper. [1], [2]Makespan and flowtime is considerably reduced.

[3]Ant colony optimization (ACO) is a sort of reproduced transformative algorithm. It emulates ants' searching procedure to find the briefest way, exists together with the attributes of haphazardness and heuristic. It is connected effectively to tackle combinatorial optimization issues, for example, the TSP(Travelling salesman problem) issue, the occupation shop planning issue, and so forth. [3]In reasonable application, ACO has the restriction of effortlessly being caught into neighborhood ideal and long time to merge. We propose an enhanced ant colony optimization algorithm, comprising of presenting irregular variable and presenting elitist ants and also debilitated procedure. Arbitrary component gives a heading to seek inside the field of the ideal way. [4]Elitist ants and debilitated methodology reinforces the pheromone over the briefest way and debilitates the pheromone over the problematic way to diminish the aggregated effect.

[5]GSA is a heuristic stochastic swarm-based hunt calculation in the field of numerical enhancement, in light of the gravitational law and laws of movement. In the same way as other nature enlivened calculations, it needs refinements to amplify its execution in taking care of different sorts of issues. Furthermore to the issue encoding that occasionally can be a test, adjusting its parameters assume a huge part adjusting the seek time versus arrangement quality.[5] This calculation is generally later and not intensely investigated. Cell position is one of four continuous strides in physical configuration procedure of VLSI circuits, in particular: parceling, situation, directing and compaction. In the arrangement organize, the depiction of the physical format of the chip is presented, by allocating geometric directions to the cells. [6]The goal of the arrangement calculation is to discover a design that limits a taken a toll capacity, whose real part is the region, yet regularly includes the viewpoint proportion, to make the chip as near square as could reasonably be expected and subsequently increment the pass on yield.

Proposed system deals with hybridization of both ant and gravity search technique to produce least possible makespan and flowtime. Next section describes existing literature to prove worth of the study.

II. RELATED WORK

This segment gives the review of the innovation and the foundation of the considering space. That may help in understanding the earth, the issues and challenges and as of late created answers for the distributed computing area. The cloud makes it possible for you to get to your data from anyplace whenever. Methods of [3], [7]Resource Scheduling There are a rich measure of strategies are accessible for proficient asset planning some of them every now and again utilized strategies are talked about in this area.

[6]Particle Swarm Optimization (PSO) is a swarm-based insight algorithm affected by the social conduct of creatures, for example, a rush of flying creatures finds a nourishment source or a school offish shielding them from a predator.[8] A molecule in PSO is undifferentiated from a fledgling or fish-flying through a pursuit (issue) space. The development of every molecule is co-ordinate by a speed which has both greatness and heading. [9]Each molecule area at any example of time is affected by its best position and the position of the best molecule in an issue space

[10]Genetic algorithm is a technique for planning in which the errands are dole out assets as per individual arrangements, which advises about which asset is to be doled out to which, assignment. [11]Genetic Algorithm is base on the organic idea of populace era. [10]The primary terms utilized as a part of hereditary algorithms are Initial populace, wellness work, choice, hybrid, transformation.

[12]Receptacle 8pressing issues include the pressing of objects of given sizes into receptacles of given limit.[13] On account of one-dimensional receptacle pressing the span of

each protest is a genuine number in the vicinity of 0 and 1, and each canister is of same limit. It is required that the whole of the items stuffed into any given container may not surpass 1. [12]The issue of finding a pressing utilizing a base number of containers is known to be NP-hard.
[14]In priority based scheduling algorith.m is altered by the scheduling heuristic or executing most elevated priority undertaking with progress ahead of time by pre-empting best-exertion undertaking as done in.[15] Algorithm demonstrates the pseudo codes of Priority Based Scheduling Algorithm (PBSA).
Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 or Open Type fonts are preferred. Please embed symbol fonts, as well, for math, etc.

III. PROPOSED WORK

Cloud computing is a kind of Internet-based computing that gives shared PC handling assets and information to PCs and different gadgets on request. It is a model for empowering omnipresent, on-request access to a common pool of configurable computing assets (e.g., PC systems, servers, stockpiling, applications and services), which can be quickly provisioned and discharged with negligible administration exertion. [16], [17]Distributed computing and capacity arrangements furnish clients and undertakings with different abilities to store and process their information in either exclusive, or outsider information centers that might be situated a long way from the user–ranging in separation from over a city to over the world. [18]–[20]Distributed computing depends on sharing of assets to accomplish soundness and economy of scale, like an utility (like the power framework) over a power organize.
Proposed system is divided into phases. First phase required cloudlets to be loaded. These cloudlets are composed of set of jobs, which are further divided into task depending upon capacity of VM.

Table I. Processes along with processes Id , Burst Time and Arrival time

Process id	Burst time	Arrival time
1	*25*	*0*
2	*35*	*1*
3	*45*	*2*
4	*55*	*3*
5	*65*	*4*
6	*75*	*5*
7	*85*	*6*
8	*95*	*7*
9	*100*	*8*
10	*49*	*9*

Task division takes place as follows

$$Task_i = Jobs_i / VM_{capacity} \quad (1)$$

Equation 1: Task division in jobs

Let VM capacity is 10 then process 1 is divided into 3 tasks having burst time 10, 10 and 5. Process 2 is divided into 4 tasks having burst time 10,10,10 and 5. Process 3 is divided into 5 tasks having burst time 10,10,10,10 and 5 .Process 4 is divided into 6 tasks having burst time 10,10,10,10,10,and 5. Process 5 is divied into 7 tasks having burst time 10,10,10,10,10,10 and 5. Process 6 is divided into 8 tasks having burst time 10,10,10,10,10,10,10 and 5.Process 7 is divided into 9 tasks having burst time 10,10,10,10,10,10,10,10 and 5.Process 8 is divided into 10 tasks having burst time 10,10,10,10,10,10,10,10,10 and 5.Process 9 is divided into 10 tasks 10,10,10,10,10,10,10,10,10,10. Process 10 isdivided into 5 tasks having burst time 10,10,10,10 and 9. No of ants will be equal to no of task.

$$ants_i = task_i \quad (2)$$

Equation 2: Ants formation equation

After ants[8], [13], [21] are assigned, schdule formation process begins. Phenormone matrix is created by ants locating resources within VM. As resources are found, phenormone matrix contains 1. The ants when found resources, process begins to execute. In case resources are not located within virtual machines, gravity search algorithm executes. This algorithm locate resources and then ant colony optimization algorithm again executed. This process continues untill all the jobs are executed. In the end, makespan and flowtime is noted. The generation terminates as optimal makespan and flowtime is found out.

The algorithm for the G-ACO is listed as under

Algorithm G-ACO

1. Initialize Cloud
   Create Datacenter, VMs, along with capacity of VMs.
2. Load Jobs
   Jobs are loaded from extranal source, Jobs are divided into tasks depending upon VM capacity
   Tasks=Jobs/VM_Capacity
3. Execute ACO
   Locate resources
   If (Found(Resources))
   Ph[i]=1
   Else
   Execute GSA for locating resource
   End
4. Repeat step 3 untill ants locates resources
5. Execute tasks on VMs and calculate Makespan and flowtime

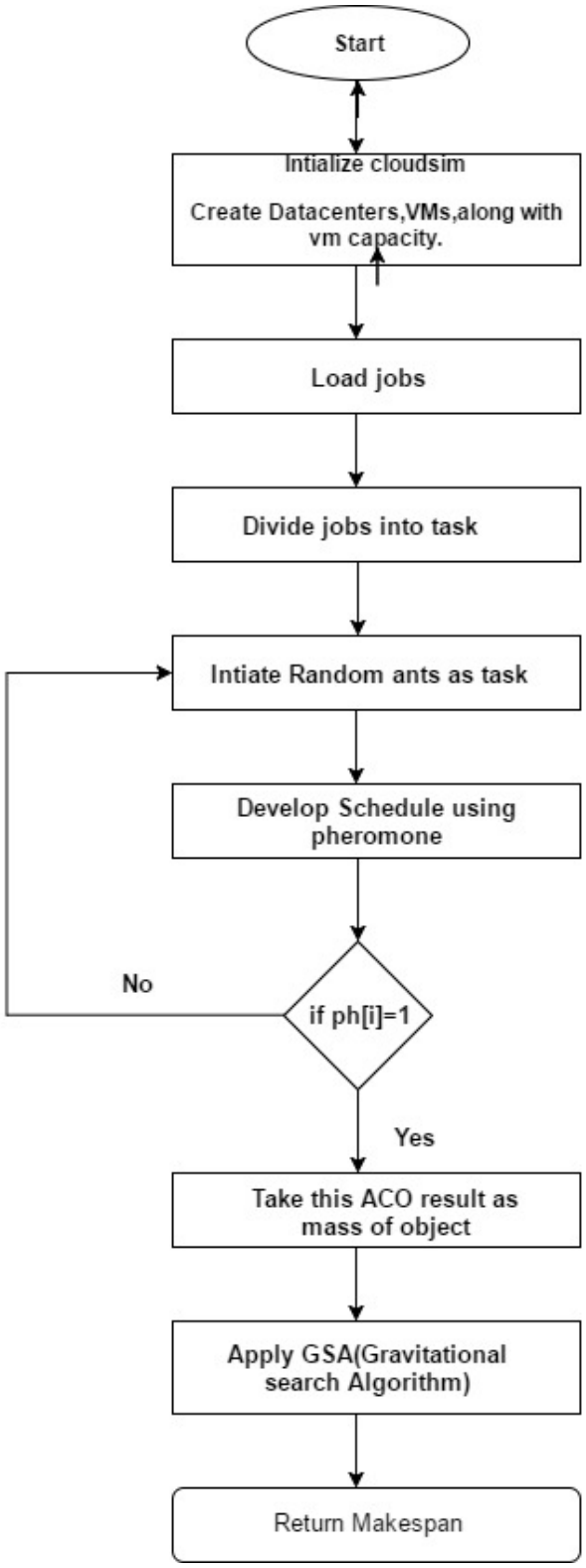

Figure1: Proposed Work

IV. PERFORMANCE ANALYSIS AND RESULTS

The performance analysis indicates compariosn of performance parameters with existing literature. Performance analysis is done in terms of makespan and flowtime. Result section is describe in terms of the following.

Table II. Makepsan of schedule

Process	*Existing*	*Proposed*
10	1440	500
20	1450	520
30	1460	540
40	1470	570
50	1480	600

As resources in terms of process arrives at the virtual machines existing system produced makespan in the capacity of 1400 whereas proposed approch makespan is in the capacity of 500. Hence significant difference in makespan is
observed.

Table III. Showing flow time of schedule

Process	*Existing*	*Proposed*
10	500	105
20	530	135
30	545	145
40	560	160
50	580	175
60	600	200

Figure 2: Plot of makespan of schedule

Flowtime is the time taken in order to process current job. As the process arrives at the VM in case of existing system considerable rise in flowtime is observed. The average energy

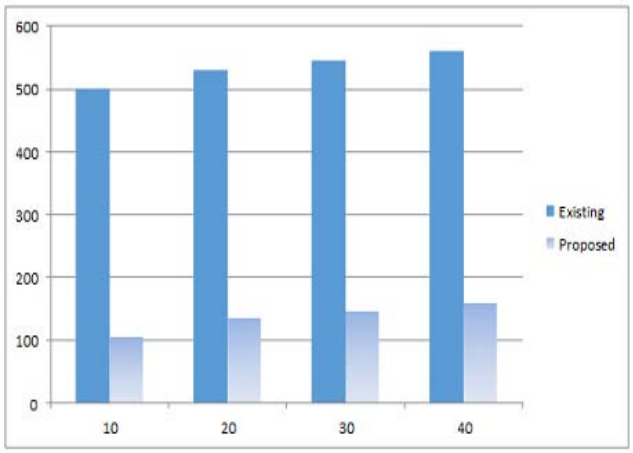

Figure 3: Showing flowtime of schedule

consumption in terms of flowtime is 550 where as in proposed system average energy consumption is 150.The flowtime assosiated with the given schedule is as under
The result indicates that proposed system with G-ACO produce better result as compared to existing technqiue without gravity search.

V. CONCLUSION AND FUTURE SCOPE

Ant colony optimization is used for multiheuristic problem solving. Multiheuristic indicates the applications having multiple objectives assosiated with them. Ant colony optimization fails as distance between resources increases.Result obtained in terms of flowtime and makspan is better as compared to existing system. Gravity search algorithm combined with ACO is used in oder to locate the resources and assign resources with ACO. Hybridication hence is the solution of the heavy tasks allocation to VMs. This literature utilizes gravity search as optimal searching criteria looking for the resource within VMs and allocation is accomplished with the help of ACO. Optimal result in terms of makespan and flowtime is obtained through the proposed literature proving worth of study.

In future ant colony and honey bee algorithm can be merged for further reducing makespan and flowtime assosiated with jobs.

VI. REFRENCES

[1] T. Kokilavani and D. I. G. Amalarethinam, “Reduced Makespan Task Scheduling Algorithm for Grid Computing,” vol. 9, no. 27, pp. 71–76, 2016.

[2] C. R. Reeves, “A genetic algorithm for flowshop sequencing,” *Comput. Oper. Res.*, vol. 22, no. 1, pp. 5–13, Jan. 1995.

[3] F. Lie, C. Huang, C. Wu, K. Chen, and H. Kuo, “Demonstration of ACO-Based Freeform Source for ArF Laser Immersion Lithography System,” pp. 6421–6428, 2017.

[4] H. Hu and H. Wang, “A Prediction- Based ACO Algorithm to Dynamic Tasks Scheduling in Cloud Environment,” pp. 2727–2732, 2016.

[5] Q. Fei, L. Jing, L. Jie, and Z. Bin, “A multi-subgroup hierarchical hybrid of central force optimization and gravity search algorithm,” *Proc. - 12th Int. Conf. Comput. Intell. Secur. CIS 2016*, pp. 138–141, 2017.

[6] B. J. Kwak, N. O. Song, B. Park, and Y. H. Kim, “Gravity search adaptive algorithm,” *5th Int. Conf. Signal Process. Commun. Syst. ICSPCS’2011 - Proc.*, 2011.

[7] R. N. S, G. N. Gopal, and S. G, “A novel scheme for authenticated secured de-duplication with identity based encryption in cloud,” *2016 Int. Conf. Inf. Sci.*, pp. 228–232, 2016.

[8] K. Mistry, L. Zhang, S. C. Neoh, C. P. Lim, and B. Fielding, “A Micro-GA Embedded PSO Feature Selection Approach to Intelligent Facial Emotion Recognition,” *IEEE Trans. Cybern.*, pp. 1–14, 2016.

[9] T. K. Maji and P. Acharjee, “Multiple solutions of optimal PMU placement using exponential binary PSO algorithm,” *12th IEEE Int. Conf. Electron. Energy, Environ. Commun. Comput. Control (E3-C3), INDICON 2015*, vol. 9994, no. c, 2016.

[10] D. Pradhan, S. Wang, S. Ali, T. Yue, and M. Liaaen, “CBGA-ES: A Cluster-Based Genetic Algorithm with Elitist Selection for Supporting Multi-Objective Test Optimization,” *2017 IEEE Int. Conf. Softw. Testing, Verif. Valid.*, pp. 367–378, 2017.

[11] M. Cuka, D. Elmazi, R. Obukata, K. Ozera, T. Oda, and L. Barolli, “An Integrated Intelligent System for IoT Device Selection and Placement in Opportunistic Networks Using Fuzzy Logic and Genetic Algorithm,” *2017 31st Int. Conf. Adv. Inf. Netw. Appl. Work.*, pp. 201–207, 2017.

[12] V. Luantangsrisuk, P. Songmuang, and R. Kongkachandra, “Automated Test Assembly with Minimum Redundant Questions Based on Bee Algorithm,” *2016 12th Int. Conf. Signal-Image Technol. Internet-Based Syst.*, pp. 652–656, 2016.

[13] W. Sa-ngiamvibool, “Optimal Fuzzy Logic Proportional Integral Derivative Controller Design by Bee Algorithm for Hydro-Thermal System,” *IEEE Trans. Ind. Informatics*, vol. 3203, no. c, pp. 1–1, 2017.

[14] D. Guo, C. Gao, W. Ni, and X. Hu, “Max-Flow Rate Priority Algorithm for Evacuation Route Planning,” *2016 IEEE First Int. Conf. Data Sci. Cybersp.*, pp. 275–283, 2016.

[15] H. P. Wdqj *et al.*, “3RZHU 3ULRULW \ $ OJRULWKP LQ $ SSOLFDWLRQ WR $ LUFUDIW,” pp. 5–9, 2016.

[16] A. Bouti and J. Keller, “Towards practical homomorphic encryption in cloud computing,” *Proc. - IEEE 4th Symp. Netw. Cloud Comput. Appl. NCCA 2015*, pp. 67–74, 2015.

[17] J. C. Chen, C. K. Wen, and P. Ting, “An efficient pilot design scheme for sparse channel estimation in OFDM systems,” *IEEE Commun. Lett.*, vol. 17, no. 7, pp. 1352–1355, 2013.

[18] S. Jayanthi and S. Babu, "Green Cloud Computing - Resource Utilization with Respect to SLA and Power Consumption," vol. 3, pp. 11–21, 2015.

[19] K. Sudhakar, "Redundant Sift Features Via Level Sets For Fast Copy Move Forgery Detection," vol. 1.

[20] A. A. Yassin, A. A. Hussain, and K. A.-A. Mutlaq, "Cloud authentication based on encryption of digital image using edge detection," *Int. Symp. Artif. Intell. Signal Process.*, pp. 1–6, 2015.

[21] A. V. Leonov, "Application of bee colony algorithm for FANET routing," *Int. Conf. Young Spec. Micro/Nanotechnologies Electron Devices, EDM*, vol. 2016–August, pp. 124–132, 2016.