IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-2 E-ISSN: 2347-2693

Hybrid Task Scheduling Algorithm Based on ANT Colony Optimization and Particle Swarm Optimization for Cloud Environment

D. Gupta[1*], H.J.S. Sidhu[2]

[1*]CSE, Desh Bhagat University, Mandi Gobindgarh, Punjab, India
[2]CSE, Desh Bhagat University, Mandi Gobindgarh, Punjab, India

*Corresponding Author: dineshgupta@ptu.ac.in

Available online at: www.ijcseonline.org

Received: 31/Jan//2018, Revised: 06/Feb2018, Accepted: 20/Feb/2018, Published: 28/Feb/2018

***Abstract*:** Cloud computing refers to many different types of services and applications being delivered over the internet cloud. Cloud load balancing is the process of distributing workloads across multiple computing resources. Load balancing is an optimization problem and goal of any optimization is to either minimize effort or to maximize benefit. The effort or the benefit can be usually expressed as a function of certain design variables. Hence, optimization is the process of finding the conditions that give the maximum or the minimum value of a function. Load balancing is a problem where you try to minimize value of parameters like Makespan time, Response Time, etc. and increase the utilization of cloud resources. Metaheuristic algorithms are a natural solution to the problem of load balancing in cloud. But these algorithms as such do not provide a complete solution. This paper proposes a hybrid of Particle Swarm optimization and Ant Colony optimization for load balancing of tasks on cloud resources.

Keywords: ACO, PSO, VM, SJF, IAAS, PAAS, SAAS, Data Centre, Cloud Computing, DI.

I. Introduction

Cloud computing provides on-demand access to shared pool of resources over internet in a self service, dynamically scalable and metered manner. Cloud computing is a new field for much research is required across a broad array of topics. One of the important research issues which need to be focused for its efficient performance is scheduling. The goal of scheduling is to map tasks to appropriate resources that optimize one or more objectives. Scheduling in cloud computing belongs to a category of problems known as NP-hard problem due to large solution space and thus it takes a long time to find an optimal solution. In cloud environment, it is preferable to find suboptimal solution, but in short period of time. Metaheuristic based techniques have been proved to achieve near optimal solutions within reasonable time for such problems**.** Section 2 gives a brief background to the problem of task scheduling of cloud resources and work that has already been in this field. Section 3 discusses the methodology for the proposed algorithm. Brief overview of the tools used and results of the implementation are given in section 4. Section 5 concludes the outcome of the paper and discusses the scope of future work based on the outcome of this paper.

II. Background and Related work

2.1 Background

Cloud computing uses a virtualized software model, enabling the sharing of physical services, storage, and networking capabilities. It allows for the provision of services based on current demand requirements. It provides support for access to resources across the internet from a broad range of devices such as PCs, laptops, and mobile devices, using standards-based APIs (for example, ones based on HTTP) [13]. Users are charged on the basis of pay-per-use model as cloud uses metering for managing and optimizing the service and to provide reporting and billing information. Cloud load balancing is an optimization problem for distributing workloads across multiple computing resources. Optimization is the process of finding the conditions that give the optimal or near-optimal maximum or the minimum value of a function with respect to some goals. Formally it can be said that optimization problem is to find an $x^* \in X$ for which

$f(x^*) \geq f(x)$ for all $x \in X$ (maximization problem)

$f(x^*) \leq f(x)$ for all $x \in X$ (minimization problem)

where:

X is a set of feasible solutions $x \in X$

$f : X \rightarrow R$ is an evaluation function that assigns a real value to every element x of the search. [10].

Ant Colony Optimization (ACO) meta-heuristic is inspired by the behavior of real ants. The idea is to find the shortest path between their colonies and a source of food. While walking amid their colony and the food source, ants leave pheromones on the ways they move. The pheromone intensity on the passages increases with the number of ants passing through and drops with the evaporation of pheromone. As the time goes on, smaller paths draw more pheromone and thus, pheromone intensity helps ants to recognize smaller paths to the food source [11]. ACO algorithm can help us in solving the cloud computing resource management and job scheduling problem. Particle Swarm Optimization (PSO) is a swarm-based intelligence algorithm influenced by the social behavior of animals such as a flock of birds finding a food source or a school of fish protecting themselves from a predator [12]. A particle in PSO is analogous to a bird or fish flying through a search (problem) space and the movement of each particle is co-ordinated by a velocity which has both magnitude and direction. Position of each particle, i.e. assignment of cloud resource to a task at any instance of time is influenced by its best position and the

position of the best particle in a problem space.

2.2 Related work

Cloud task scheduling problem relates to scheduling of tasks on resources subject to some constraints to optimize some objective function. Scheduling allows optimal allocation of resources among given tasks in a finite time to achieve desired quality of service. The aim is to build a schedule that specifies when and on which resource each task will be executed [1]. Cloud task scheduling policy based on Ant Colony Optimization (ACO) algorithm outperforms FCFS and RR algorithms. But the ACO fails to improve the imbalance factor beyond a factor [2]. Goal Oriented Task Scheduling (GOTS) schemes give service providers a fair chance to apply approach and schedule the tasks and resources that can generate maximum possible economic gains, while using least resource provisioning [3]. An extensive survey and comparative analysis of various scheduling algorithms for cloud and grid environments based on three popular metaheuristic techniques: Ant colony optimization (ACO), genetic algorithm (GA) and Particle Swarm optimization (PSO) is given in [4]. Metaheuristic techniques are usually slower than deterministic algorithms and the generated solutions may not be optimal, most of the research done is towards improving the convergence speed and quality of the solution. These issues have been undertaken by modifying the transition operator, pre-processing the input population or taking hybrid approach in metaheuristic techniques [4]. Overview of different Bio inspired algorithm for tackling various challenges faced in Cloud Computing Resource management environment is given in [5]. Bio inspired algorithm plays very important role in distributed, data mining, power system, economics, robotics, information security, control system, image processing etc. There are great opportunities of exploring or enhancing this field algorithm with the help of innovative ideas or thoughts. Since this field of Bio inspired algorithm bridge a knowledge bond between different communities like computer science, biology, economics, artificial intelligence etc [5]. [6] presents an overview of significant advances made in the emerging field of nature-inspired computing (NIC) with a focus on the physics- and biology based approaches and algorithms. A parallel development in the past two decades has been the emergence of the field of computational intelligence (CI) consisting primarily of the three fields of neural networks, evolutionary computing and fuzzy logic. It is observed that NIC and CI intersect. Some researchers have argued that swarm intelligent provide computation intelligence. A Platform-as-a-Service model to build an HPC cloud setup is proposed in [7]. The key goals for the architecture design is to include features like on-demand provisioning both for hardware as well as HPC runtime environment for the cloud user and at the same time ensure that the HPC applications do not suffer virtualization overheads. Scheduling algorithms based on artificial bee colony are discussed in [8]. These algorithms exploit the smartness of bees' behavior to reach an near optimal assignment of tasks to resources. An analytical comparison has also been done to briefly show the characteristics and applications of bee colony technique for scheduling in distributed computing area. An Autonomous Agent Based Load Balancing Algorithm (A2LB) which provides dynamic load balancing for cloud environment is proposed in [9]. The proposed mechanism provides proactive load calculation of VM in a DC and load agent is responsible for searching an unloaded VM whenever the load of a VM reaches to its threshold value.

III. Methodology

Algorithm

Input:

CloudletList: List of all clouldlets received.

VmList: List of all VM's

Main_Function Hybrid_TS

// initially schedule a small no. of incoming tasks using PSO [number of tasks can be fixed or decided on the basis of percentage of total tasks]. In the proposed work, cloudlets = 2 times the size of VM list are initially passed to PSO scheduling algorithm

1. PSOCloudletList = CloudletList [1: Sizeof (VmList)* 2]
2. Call PSO_TS (PSOCloudletList, VmList)
3. ACOCloudletList = CloudletList – PSOCloudletList

// Scheduling based ACO algorithm

4. temp_List_of_Cloudlet = null, temp_ACO_List_of_Cloudlet = ACOCloudletList and n= size of VMs list

5. while temp_ACO_List_of_Cloudlet not empty

if (size of temp_ACO_List_of_Cloudlet greater than n)

Transfer the first arrived n Cloudlets from temp_List_of_Cloudlet and put them on temp_List_of_Cloudlet

else

Transfer all Cloudlets from temp_List_of_Cloudlet and put them on temp_List_of_Cloudlet

end If

6. Call ACO_TS (temp_ACO_List_of_Cloudlet, VmList)
7. temp_ACO_List_of_Cloudlet = temp_ACO_List_of_Cloudlet - temp_List_of_Cloudlet
8. **Compute the degree of imbalance factor between the VM's.**
9. If the degree of imbalance factor is greater than threshold value then
   a. For each VM in the VM List, compute the length of all cloudlets assigned to each VM in the VM List.
   b. Sort of VM's on the basis of length and create a list UL_VmList of all underloaded VM's and List of Cloudlets OL_VM_CloudletList of Cloudlets in the execution list of overloaded VM's
   c. Call PSO_TS (OL_VM_CloudletList, UL_VmList) and transfer Cloudlets from overloaded loaded VM to Least loaded VM in the sorted list.

End Do

Function PSO_TS (CloudletList, VmList)

1. Set particle dimension as equal to the size of ready tasks T.
2. Initialize particles position randomly from PC = 1.....j and velocity vi randomly.
3. For each particle, calculate its fitness value.
4. If the fitness value is better than the previous best pbest, set the current fitness value as the

   new pbest.

5. Perform Steps 3 and 4 for all particles and select the best particle as gbest.
6. For all particles, calculate velocity and update their positions.
7. If the stopping criteria or maximum iteration is not satisfied, repeat from Step 3 & 4.

Function ACO_TS (CloudletList, VmList)

1. Initialize pheromone value for each path between tasks and resources, set optimal solution to NULL and place m ants on random resources.
2. Repeat for each ant
a. Put the starting resource of first task in tabu list and all other tasks in allowed list.
b. Based on the probability function or transition rule, select the resource for all remaining tasks in the allowed list.
3. Compute fitness of all ants which in this case is Makespan time.
4. Replace the optimal solution with the ant's solution having best fitness value if its value of better than previous optimal solution.
5. Update both local and global pheromone.
6. Stop when the termination condition is met and print the optimal solution.

Degree of imbalance (DI) can be computed using equation 1 & 2. It measures the imbalance between VMs [2].

$$T_i = \frac{TL\_Tasks}{Pe\_num_j * Pe\_mips_j}$$

$$DI = \frac{T_{max} - T_{min}}{T_{avg}}$$

TL_Tasks refers to the length of all tasks assigned to VM_i. T_{max}, T_{min} and T_{avg} refer to the average of T_i among all VMs [2].

IV. Results

Net Beans IDE along with CloudSim libraries are used for implementing the task scheduling. Net Beans IDE is a free, open source, integrated development environment (IDE) that enables you to develop desktop, mobile and web applications. CloudSim is an extensible simulation toolkit or framework that enables modeling, simulation and experimentation of Cloud computing systems and application providing environments. Following is the parameter setting of CloudSim:

Table 1: Parameters Setting of Cloudsim

Entity Type	Parameters	Values
Task(Cloudlet)	Length of Task	20000-400000
	Total Number of Task	150-250
Virtual Machine	Total Number of VMs	8
	MIPS	1024-4096
	VM Memory (RAM)	128-512
	Bandwidth	500-2000
	Cloudlet Scheduler	Time_shared and Space_shared
	Number of PEs Requirement	2
Data Centre	Number of Datacenter	1
	Number of Host	3
	VM Scheduler	Time_shared and Space_shared

No. of Cloudlets	Ant Colony Optimization (ACO)	Particle Swarm Optimization (PSO)	Autonomous agent load balance (A2LB)	Hybrid Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO)
150	270	250	240	180
175	309	300	285	220
200	325	335	310	240
225	355	370	335	260
250	403	425	375	295

Table 2: Makespan Time for ACO and PSO Task scheduling with fixed VMs = 8

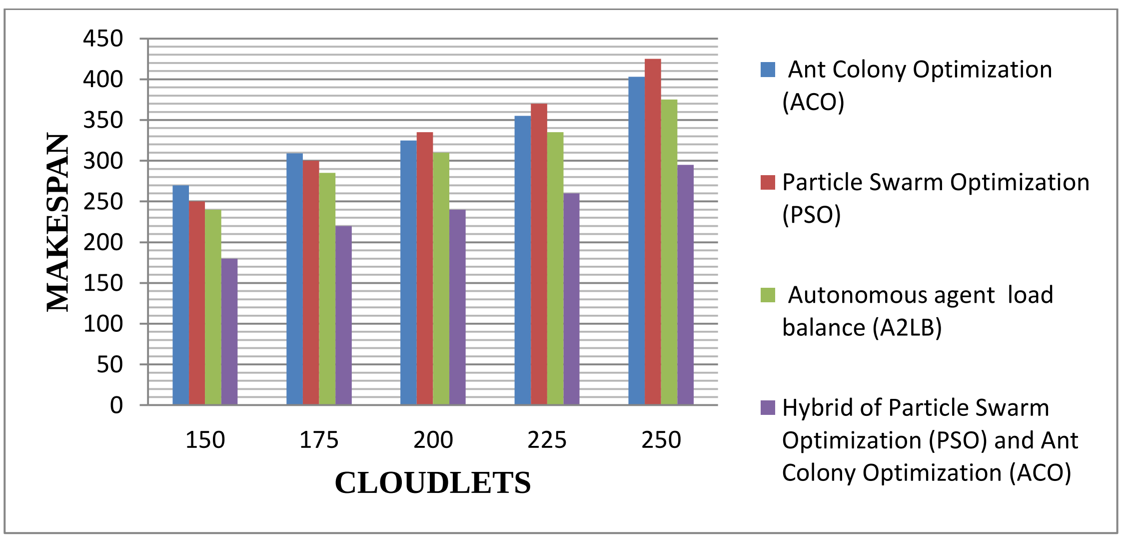

Fig.1: Makespan Time for ACO and PSO Task scheduling with fixed VMs = 8

V. Conclusion

Scheduling of tasks on cloud resources is a large and complex problem for which the use of Metaheuristic algorithms is a natural solution. But these Metaheuristic algorithms only provide a near optimal solution in lesser time. There is scope for improvement in the results. In this paper, a hybrid of two popular Metaheuristic algorithms (Particle Swarm optimization & Ant Colony optimization) has been proposed. The idea is to take advantage of fast computation of PSO to initially schedule small number of tasks using PSO and to schedule the remaining tasks using ACO. Also in case the imbalance factor breaches the threshold, the tasks from over-loaded VMs be transferred to under-loaded VMs using PSO again. It is clear that the hybrid solution provides much better result than the 3 other algorithms. In future resource awareness can also be added to this hybrid algorithm to ensure fair treatment that can result in even better results.

Also, the comparison in performance of scheduling algorithms can be done using addition task scheduling parameters like response time, flow time, etc.

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] Medhat Tawfeek, Ashraf El-Sisi, Arabi Keshk , Fawzy Torkey, "Cloud Task Scheduling Based on Ant Colony Optimization" , The International Arab Journal of Information Technology, Vol. 12, No. 2, March 2015.

[3] Jyoti Thaman, Manpreet Singh, "Current Perspective in Task Scheduling Techniques in cloud Computing: A Review", International Journal in Foundations of Computer Science & Technology (IJFCST) Vol.6, No.1, January 2016.

[4] Mala Kalra, Sarbjeet singh, "a review of metaheuristic scheduling techniques in cloud computing", available online 18 august 2015.

[5] Gurtej Singh, Amritpal kaur, "Bio Inspired Algorithms: An Efficient Approach for Resource Scheduling in Cloud Computing", International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 10, April 2015.

[6] Nazmul Siddique, Hojjat Adeli, "Nature Inspired Computing: An Overview and Some Future Directions", Published online: 30 November 2015.

[7] Pratima Dhuldhule, J. Lakshmi, S. K. Nandy, "High Performance Computing Cloud - a Platform-as-a-Service Perspective" , 2015 International Conference on Cloud Computing and Big Data.

[8] Elaheh Hallaj, Elaheh Hallaj, "Study and Analysis of Task Scheduling Algorithms in Clouds Based on Artificial Bee Colony", Second International Congress on Technology, Communication and Knowledge (ICTCK 2015) November, 11-12, 2015 - Mashhad Branch, Islamic Azad University, Mashhad, Iran.

[9] Aarti Singh, Dimple Juneja, Manisha Malhotra, "Autonomous Agent Based Load Balancing Algorithm in Cloud Computing", International Conference on Advanced Computing Technologies and Applications (ICACTA-2015), pp. 832-841.

[10] F. Rothlauf, "Design of Modern Heuristics Principles and Application", Springer. Verlag Berlin Heidelberg, pp. 07-36, 2011. ISBN 978-3-540-72961.

[11] Dorigo M, Stu¨ tzle T. Ant colony optimization. MIT Press; 2004.

[12] J. Kennedy and R. Eberhart. Particle swarm optimization In IEEE International Conference on Neural Networks, volume 4, pages 1942–1948, 1995.

[13] V. Kale, "Big Data Computing: A Guide for Business and Technology", CRC Press, US, pp. 177-203, ISBN 9781498715331.

Authors Profile

Mr. Dinesh Gupta, did his M. Tech in IT from Department of CSE GNDU Amritsar, India. Currently he is pursuing his Ph. D in CSE from Desh Bhagat University. He has more than 8 years of experience in teaching. Currently he is working as Assistant Professor in department of CSE, IKGPTU, India. He has more than 8 publications in leading research Journal.

Dr. Harmaninder J. S. Sidhu is serving as Assis-tant Professor in Department of Computer Science and Applications, Desh Bhagat Uni-versity, Mandi Gobindgarh, Punjab, India. He is involved in academic activities for last 10 years. His interest areas include Open Source Software and Algorithm Analysis.