International Journal of Engineering & Technology, 7 (2.19) (2018) 70-74

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Hybrid AFSS-SHC Optimization Methodology for Balancing the Load Efficiently in Cloud Environment

Aswini. J[1], N. Malarvizhi[2], Anitha. K[3]

[1]*Research Scholar,*
Department Of Computer Science And Engineering,
Meenakshi Academy Of Higher Education And Research,
Chennai – 600 069, Tamil Nadu, India.
And
[1]*Assistant Professor,*
Department Of Information Technology,
Jawahar Engineering College,
Chennai – 600093, Tamil Nadu, India.
[2]*Professor & Head,*
Department Of Computer Science And Engineering, School Of Computing,
Vel Tech Rangarajan Dr. Sagunthala R&D Institute Of Science And Technology,
Chennai-600062, Tamil Nadu, India.
[3]*Assistant Professor,*
Department Of Computer Science And Engineering,
R.M.K Engineering College,
Chennai-601206, Tamil Nadu, India.
** Aswini.Jayaraman@Gmail.Com*

Abstract

Cloud computing helps to share data and provide many resources to users. Users pay only for those resources as much they used. Rapid increase in load to these cloud framework cannot be predicted. Load balancing is one of the issues in cloud computing that distributes the workload to the nodes in such a way no node is overloaded or under - loaded. Load balancing is a main challenge in cloud environment. In this work, scheduling algorithm is applied for load balancing by considering the cost of task execution and make span. This scheduling algorithm efficiently maps task to available nodes in cloud and it is beneficial to user and service provider. Load balancing segregates assignments of tasks among all available virtual machines from datacenters. Assignment of tasks to virtual machines can be done with minimum delay. To enhance the make span, resource utilization, our proposed framework utilizes AFSS-SHC load balancing strategy. A metaheuristics swarm intelligence algorithm which is NP-hard have been suggested to balance load across devices. The algorithms taken into account are-HEFT,PSO and PSO-HC. The proposed methodology AFSS-SHC optimized the task scheduling. Random tasks have been taken for this purpose and simulated to show that the proposed methodology works efficiently to reduce the make span of tasks to reduce the cost.

Keywords: *Cloud Computing, Load Balancing, Swarm intelligence, Virtual Machines and Make span.*

1. Introduction

The invention of cloud computing has carried with it several favorable circumstances for the deployment of extensive data in the real world. Cloud computing is an Internet-based computing, in which data is shared to devices on request. The framework involves fundamental attributes such as load balancing, elasticity, on-demand self service, measured service and broad network access. Load balancing is a main phase in cloud environment. It is the process of circulating the aggregate workload to the individual frameworks in a distributed network to utilize the resources effectively and to enhance the execution [1].

Algorithms of Load balancing are delegated static and dynamic in light of the present condition of the framework [2]. A dynamic load balancing algorithm will not think about the earlier state or conduct of the framework. It relies upon the present behavior of the framework where as static load balancing algorithms does not rely on the current state of the framework. Previous knowledge of the framework is necessary.

Dynamic load balancing methodologies are further disseminated as distributed and non distributed load balancing algorithms. However, the fundamental objective of these algorithms is to enhance the performance, to keep up framework stability and to prevent loss of information in case of failure of the system [3]. In dynamic load balancing methodologies, job allocation is done for the processors during the runtime. Based on the upcoming data gathered the coordinator allocates new processes to the slaves. The

basic things to consider while developing such calculation are estimation of load, connection of load, quality of different system, execution of structure, cooperation between the hubs, idea of work to be traded, determination of hubs etc [4][5].

In general, load balancing problem is a NP-hard problem [6–9]. Hence, numerous heuristic methods have projected to put on a moderately optimal solution. Notwithstanding, the weakness of heuristic techniques is that they wind up stuck in local optima. To unravel this issue, the metaheuristics method is utilized to tackle the balancing problem [10]. The transformative calculation, a predominant metaheuristics technique, can research and make utilization of potential arrangements in the pursuit condition. Be that as it may, this looking capacity sets aside a long time to converge the individuals to the optimal point. Freshly, population-based methods such as Particle Swarm Optimization (PSO), are combined with local search algorithms such as hill climbing to quicken the capacity of search [11].

To amplify the capability to get away from area in the look into environment where the appropriate arrangement isn't discovered , the varieties of the PSO [12, 13] have been formed. In addition, a different swarm intelligence algorithm named Fish School Search algorithm (FSS), presents an exceptionally momentous element called volitive operator which will be extremely appropriate for dynamic condition. PSO converges quicker compared to FSS. Therefore the volitive operator of FSS can be incorporated with the PSO to overcome the limitation of PSO and to improve the execution of the PSO. Thus in this paper a hybrid methodology VPSO-SHC is proposed.

The sequence in which the paper is structured is as three more sections. In Section 2, the load balancing issue and its model are displayed. Section 3 depicts the exploratory outcomes and execution investigations of the proposed hybrid methodology. At last, section 4 confers the conclusion and future work.

2. Proposed Work

This section gives the background of PSO, FSS and Volitive PSO. Then the details of Stochastic Hill-Climbing method for local optimization is described and finally hybrid AFSS-SHC phases are depicted in more detail.

2.1 PSO (Particle Swarm Optimization)

Particle Swarm Optimization (PSO) is a optimization methodology motivated by conduct of flock of birds. Kennedy and Eberhart [18] initially proposed this algorithm and is extensively used to resolve issues in optimization. The standard method comprises a swarm of particles, in which every particle comprise a location inside the exploring area $\vec{s_i}$ and every location correspond to a key. In accordance to present velocity $\vec{v_i}$ the particles fly over entire exploration area probing for the finest key. ($\overrightarrow{P_{besti}}$) signify best location of particle. ($\overrightarrow{G_{best}}$) gives best location of entire swarm during exploration. Equation 1 finds velocity of a particle i according to Shi and Eberhart [19].

$$v_i(t+1) = w\vec{v_i}(t) + r_1c_1\left[\overrightarrow{P_{best_i}} - \vec{s_i}(t)\right] + r_2c_2\left[\overrightarrow{G_{best_i}} - \vec{s_i}(t)\right] \quad (1)$$

In the above equation $r1$ and $r2$ represent random numbers interim [0, 1]. Weight (w) controls authority of velocity attained previously that maintains constant behavior of the process. The parameter $c1$ represents cognitive constant and $c2$ is social acceleration constant. These constants weights the power of the particle and also gives the information obtained from neighbor of the particle. Equation 2 updates the location of every particle based on its velocity.

$$\vec{s_i}(t+1) = \vec{s_i}(t) + \vec{v_i}(t+1) \quad (2)$$

Two scenarios named global and local define the neighborhood of the particles. The global scenario called G_{best} obtains direct information from other particles. The local scenario called L_{best} collaborates information only with two neighbors. This action is used to control the premature attraction of all particles.

2.2 FSS (Fish School Search)

The Fish School Search (FSS) is a methodology that depend on extroverted performance of fish in the ocean. Initially Bastos-Filho *et al* presented this methodology. According to this approach, every fish signify a key for the problem. The accomplishment of a fish through the exploration procedure is designated by its mass. There are named operators in FSS, that are implemented during every iteration for each fish belonging to the school: $Step_{individ}$ represent individual movement. It is used for narrow search; feeding updates the mass of the fish. It represents rate of triumph or stoppage through the exploration procedure; collective-instinctive behavior directs all fish moves, headed for a follow-on; collective-volitive behavior coordinate exploration granularity. In the proposed work since dynamic environments are dealt, feeding and collective-volitive movement operators were utilized to construct proposed hybrid methodology.

2.2.1 Feeding Operator

This operator decides the mass variation of fish at every cycle. It should be observed that a fish either increment or diminish its mass individually according to achievement or disappointment amid the exploration procedure. The mass of the fish is assessed by Equation 3:

$$M_i(t+1) = M_i(t) + \frac{\Delta f_i}{\max(|\Delta f|)}, \quad (3)$$

Where $M_i(t)$ represent the mass of fish *i*, Δf_i gives the disparity amid the updated location and the present location of fish, $max(|\Delta f|)$ designate the complete value of highest fitness deviation amongst entire fish. M_{scale} confines the maximum mass of fish. The mass of fish can diverge in the interval $[1, w_{scale}]$. Its initial value is assigned to $\frac{M_{scale}}{2}$.

2.2.2 Collective-Volitive Movement Operator

The granularity of the exploration operated by the fish school is controlled by this operator. At whatever time the whole school acquires considerable results, the operator performs the approximation of fish planning to hasten the convergence in the direction of a superior area. In contrast, the operator makes the fishes to move away from the school's barycenter and thus fishes have more probability to flee from a local minimum. The withdrawal of the fish can be assessed as:

$$\overrightarrow{B(t)} = \overrightarrow{B(t)} = \frac{\sum_{i=1}^{N} \vec{s_i}(t) M_i(t)}{\sum_{i=1}^{N} \vec{s_i}(t)} \quad (4)$$

The withdrawal of fish is done by,

$$\vec{s_i}(t+1) = \vec{s_i}(t) \pm step_{vol} r_1 \frac{\vec{s_i}(t) - \vec{B}(t)}{d(\vec{s_i}(t), \vec{B}(t))} \quad (5)$$

here $r1$ is a random number in the interim [0, 1]. The Euclidean distance amid the particle i and the core point of mass is evaluated by $d(\vec{s_i}, \vec{B})$. The $step_{vol}$ controls step size of fish which is bounded by parameters $step_{vol\text{-}min}$ and $step_{vol\text{-}max}$ and decreases linearly from $step_{vol\text{-}max}$ to $step_{vol\text{-}min}$ during iterations of the algorithm.

2.3 Augmented Fish School Search (AFSS)

This section gives the detail of the methodology, called AFSS, which is a combined version of the FSS and the PSO. The combined version include two FSS operators namely feeding and the collective-volitive movement into the PSO. In the AFSS, each particle is assigned with a weight and is called weighted particle, where the weight shows collective-volitive movement. This results in growth or reduction of the school. The $step_{vol}$ diminishes using equation (6). ($decay_{vol}$) indicates volitive step decay rate that should be in the interim [0, 100].

$$step_{vol}(t+1) = step_{vol}(t) \frac{100 - decay_{vol}}{100} \quad (6)$$

Whenever there is a change in the environment, the $step_{vol}$ is reinitialized to $step_{vol\text{-}max}$. The AFSS pseudo code is shown in algorithm 3.

2.4 Hybrid AFSS- SHC algorithm

The hybrid methodology starts by randomly initializing the population using AFSS. Then, Heterogeneous Earliest Finish Time (HEFT) [14] is utilized to analyze and rank each particle which is portrayed in Algorithm 2. Generally convergence rate of result will be low in evolutionary approaches. This is due to the cooperation of the algorithms with the local search. An alternative of Hill Climbing methodology called Stochastic Hill Climbing (SHC) [15] is one among better approaches for resolving these sort of optimization tribulations. This algorithm is just a loop which progressively moves towards the upward path called uphill. The algorithm will stop once the ”peak” is reached. Here no neighbor has a elevated value. This variation picks indiscriminately amongst the uphill moves and the likelihood of choice could fluctuate with the sheerness of uphill move.
Accordingly it maps assignments to an arrangement of assignments by rolling out minor improvements to the existing assignment. To enhance the assessment score of the state every component of the set is assessed by a few criteria that intends to draw nearer to a ample assignment. The best component of the set is made as the following assignment. This fundamental activity is rehashed until either an answer is found or a ceasing criteria is met. Along these lines it has two essential systems in particular candidate generator and an assessment criteria. The former maps one arrangement possibility to a set of conceivable successors and the latter ranks each valid assignment, such that enhancing the assessment prompts to valid arrangements. Therefore, in the proposed work, so as to enhance the arrangements locally, subsequent to utilizing the AFSS methodology, Hill Climbing strategy is utilized [16].

The proposed work is tested repetitively till the termination condition is met. The process stops if the fitness of the particle is constant for continuous ten generations or it reaches fifty generations. The primary work of the proposed methodology is presented in Algorithm 1.

Algorithm 1: Main Function of AFSS-Stochastic Hill Climbing Algorithm

Input

Parameters of AFSS-SHC;
Parameters of task assignment;

Result

Scheduled task.

1. Initialize a population
2. Do steps 3,4,5,6 till the termination criteria is satisfied
3. Implement Algorithm 2 for assigning subtasks to processors and to perform fitness evaluation;
4. Select particles by Roulette Wheel selection
5. Implement Algorithm 3 to execute the AFSS algorithm phase
6. Call Algorithm 4 to perform a local search using Stochastic Hill Climbing algorithm
7. Return an optimized particle

Algorithm 2 Subtask assignments to resources

Input:

The present particles

Result:

Make span

Enqueue subtasks in priority queue
for each particle in queue and is not empty **do**
choose the subtask t_i from queue;
for every p_k in the processor set do
Calculate EFT(t_i, p_k) value using suitable HEFT scheduling;
Allocate subtask t_i to the processor p_k that minimizes EFT(t_i, p_k);
end for
Dequeue t_i from the priority queue;
end for
return make span=AFT(t_{exit}).

Algorithm 3 AFSS pseudo code

while *condition for termination is not satisfied* **do**
for each *particle*
Find the fitness of the particle;
Find $\overrightarrow{P_{best}}$ and $\overrightarrow{L_{best}}$
end
if *change in the environment is detected* **then**
Initialize $step_{vol}$;
end
for each *particle*
Do updation of velocity and the particle position;
Calculate the fitness of particle;
end
Run the feeding operator;
Run the collective-volitive movement operator;
for each *particle of the swarm* **do**
Calculate $\overrightarrow{P_{best}}$ and $\overrightarrow{L_{best}}$
end
Upgrade $step_{vol}$ and w;
end

Algorithm 4 Stochastic Hill-Climbing Algorithm

Stage 1: Maintain a list table of Virtual Machine servers and the condition of the VIRTUAL MACHINE BSY/AVAIL At the begin all VIRTUAL MACHINEs are accessible.
Stage 2: Another task turn up in the cloud.
Stage 3: Create question for the following assignment.
Stage 4: Create a VIRTUAL MACHINE id arbitrarily.

Stage 5: Parse the distribution table from to get the status of the specific VIRTUAL MACHINE. On the off chance that the VIRTUAL MACHINE is discovered unallocated:

Stage 5a: Return the identification of VIRTUAL MACHINE.

Stage 5b: Send the demand to the VIRTUAL MACHINE recognized by that id.

Stage 5c: Update the designation table appropriately. In the event that the VIRTUAL MACHINE is observed to be assigned.

Stage 5d: Use an irregular capacity to created an arbitrary VIRTUAL MACHINE.

Stage 5e: Select the VIRTUAL MACHINE for designation to the activity with a likelihood to such an extent that this VIRTUAL MACHINE will have the capacity to deal with the activity productively.

Stage 5f: Keep record of execution of the VIRTUAL MACHINE in the event that it doesn't perform as indicated by desire (cost esteem) diminish its likelihood for task in next emphasis.

Stage 5g: Update the allotment table as needs be.

Stage 6: When the VIRTUAL MACHINE wraps up the demand, and the reaction cloudlet is gotten. Produce a notice of VIRTUAL MACHINE de-designation..

Stage 7: Continue from Step 2 for next designation.

3. Experimental Results

This section analyze the behavior of the proposed methodology by utilizing make spans. The simulation of the proposed hybrid AFSS-SHC methodology is simulated in diverse systems. The resource powers are diverse in the execution frameworks. The attained results are compared with the PSO [17], PSO-HC by involving the proposed algorithm.

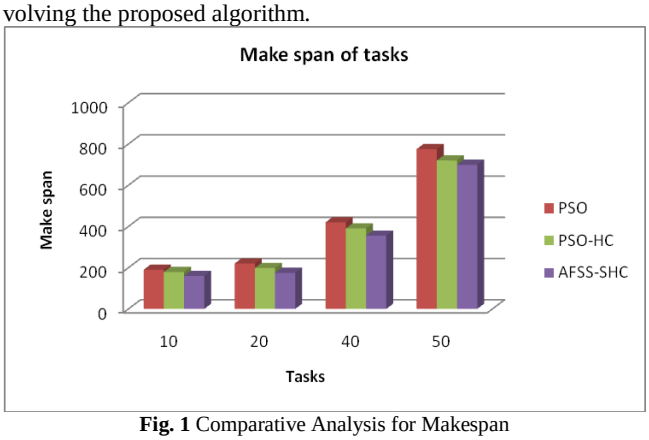

Fig. 1 Comparative Analysis for Makespan

The CloudSim environment is utilized to test the efficiency of the proposed methodology. Fig. 1 portrays make spans of 50 diverse tasks with 10 subtasks acquired from scheduling algorithms in the course of PSO, PSO-HC and the AFSS-SHC methodology. The simulation results attained designate that the proposed AFSS-SHC methodology performs better than these two methodologies in terms of make span, and acquire much time in contrast to PSO. The methodologies are executed using 50 subtasks with 50 different randomly generated tasks. The attained outcome points that the anticipated methodology performs better than PSO and PSO-HC methodologies with regard to make spans.

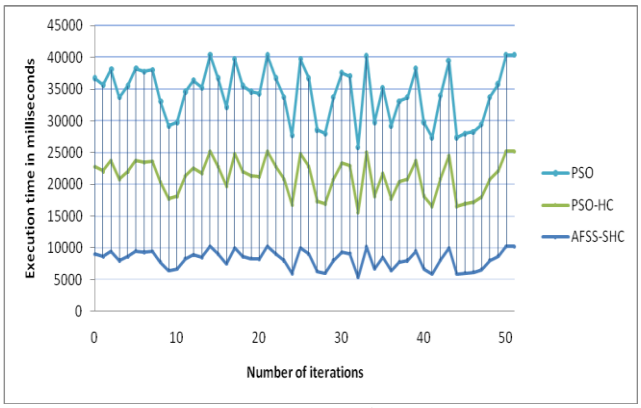

Fig. 2 Comparative Analysis for Execution time

The fig. 2 shows the time of execution for PSO, PSO-HC and AFSS-SHC methodologies. The obtained results shows that AFSS-SHC methodology is better than existing methodologies. It is evident from the outcomes that AFSS-SHC methodology optimizes the existing methodologies.

4. Conclusion

The hybrid AFSS-SHC methodology for task scheduling in cloud computing environment is proposed in this paper. This methodology utilizes AFSS in combination with Stochastic Hill-Climbing methodology, to assign subtasks to offered resources. As per the outcomes attained on random tasks of different dimensions the proposed hybrid methodology is extensively efficient than the PSO and PSO-HC methodologies with regard to make spans. As a future work, the proposed methodology can be tested using more parameters such as fault recovery by taking into consideration the load balancing of the resources.

References

[1]. R. W. Lucky, 'Cloud computing', IEEE Journal of Spectrum, Vol. 46, No. 5, May 2009.

[2]. Uddalak Chatterjee , 'A Study on Efficient Load Balancing Algorithms in Cloud Computing Environment', International Journal of Current Engineering and Technology, Vol.3, No.5 , December 2013.

[3]. Klaithem , Nader, Mariam and Jameela ,' A Survey of Load Balancing in Cloud Computing :Challenges and Algorithms', 2012 IEEE Second Symposium on Network Cloud Computing and Applications

[4]. Ali M. Alakeel, 'A Guide to Dynamic Load Balancing in Distributed Computer Systems', IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[5]. Sandeep, Sarabjit and Meenakshi, 'Performance Analysis of Load Balancing algorithms' , World Academy of Science, Engineering and Technology 2008.

[6]. Negar and Nima, A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments, 2017 The Korean Institute of Communications Information Sciences.

[7]. M. Ashouraie and N. Jafari Navimipour, 'Priority-based task scheduling on heterogeneous resources in the expert cloud', Kybernetes 44, 1455–1471, 2015.

[8]. P. Azad, J.N. Navimipour, An energy-aware task scheduling in cloud computing using a hybrid cultural and ant colony optimization algorithm, Int. J. Cloud Appl. Comput. 7 (2017).

[9]. N.J. Navimipour, F.S. Milani, Task scheduling in the cloud computing based on the cuckoo search algorithm, Int. J. Model. Optim. 5 (2015) 44.

[10]. M. Akbari, H. Rashidi, A multi-objectives scheduling algorithm based on cuckoo optimization for task allocation problem at compile time in heterogeneous systems, Expert Syst. Appl. 60 (2016) 234–248.

[11]. S. Padmavathi, S. MohitGolchha, A. SeeniMohamed, Memetic algorithm based task scheduling using probabilistic local search, in: B.K. Panigrahi, S. Das, P.N. Suganthan, P.K. Nanda (Eds.), Swarm, Evolutionary, and Memetic Computing: Third International Con-

ference, SEMCCO 2012, Bhubaneswar, India, December 20–22, Proceeding, Springer, Berlin, Heidelberg, 2012, pp. 224–231.
[12]. Blackwell and Bentley,' Dynamic Search with Charged Swarms. In: Proceedings of the Genetic and Evolutionary Computation Conference', pp. 19–26, 2002.
[13]. Carlisle and Dozier, ' Applying the particle swarm optimizer to non-stationary environments' , Thesis, Auburn University, Australia, 2002.
[14]. H. Topcuoglu, Hariri, Min-You, 'Performance-effective and low complexity task scheduling for heterogeneous computing', IEEE Trans. Parallel Distributed Systems, 260–274, 2002.
[15]. Russell and Norvig, 'Artificial intelligence: A modern approach 3/e', Pearson Publication, ISBN-10:136042597, 2010.
[16]. Keshanchi and Jafari Navimipour, 'Priority-based task scheduling on cloud computing environment using a memetic algorithm', Circuits System Computing, 2015.
[17]. Kumari, Raja, Shanthini, 'A hybrid approach of genetic algorithm and multi objective pso task scheduling in cloud computing', Asian J. Res. Soc. Sci. Human, 1260–1271, 2017.