International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 1, Jan-Feb-2020, ISSN (Online): 2395-566X

Hybrid Teacher Learning and Particle Swarm Optimization Based Dynamic Load Balancing

Vinod Patidar, Dr. Shiv Sakti Shrivastava
Department of Computer Science & Engineering,
Rabindranath Tagore University Bhopal, MP, India

Abstract - : **A digital platform objective is to split information, estimation, and service evidently over a scalable system of nodes. While Cloud computing a service accumulates the information and dispersed resources in the release atmosphere. Consequently load balancing in this atmosphere is a key problem. This paper has resolve this issue by developing a dynamic algorithm which generate job sequence as per available resources. This paper has proposed a hybrid genetic algorithm where Particle Swarm Optimization PSO and Teacher Learning Optimization TLO algorithms operation were combined to generate low cost job sequences. Experiment was done on real dataset job sequences. Results were compared with existing algorithm on different evaluation parameters and it was obtained that proposed TLPSO algorithm values are better.**

Keywords- **Crop yield prediction, Data mining, machine learning, Vegetation Index..**

I. INTRODUCTION

Distributed computing gives power to an effective computing by centralized memory, processing, storage and bandwidth. It should ensure that, no single VM is stacked vigorously and moreover guarantee that some VMs don't stay inactive or potentially under stacked. Cloud computing has been increasing in massive where client can pay (as you use) for software and hardware. To enhance the reaction time of the client's submitted application or requests, all the accessible assets are utilized with the help of load calculations [1, 2].

Load regulating methods way to accelerate the implementation of operation by ejecting activities from over stacked VMs and conveying them to under stacked VMs, Load adjusting is the means to boost the implementation of similar and dispersed framework in the course of a reorganization of load amongst the processors or centers, as load balancing is one of the only hurdle recognized with dispersed computing, the load may be reliant upon CPU limit, memory storage, display stack and so on [3, 4].

Load balancing in clouds is a method that allocates the overload vibrant local workload consistently across all the nodes [5]. It is utilized to accomplish a high customer fulfillment and source deployment ratio, making sure that no single node is overwhelmed, hence improving the in general presentation of the PC. Accurate load balancing be able to assist in using the accessible resources optimally, thus diminishing the resource utilization. It in addition helps in applying fail-over, permitting scalability, evading bottlenecks and over-provisioning, lessening. Reply time etc [6-8]. So objective of load balancing is developing the presentation by balancing the load amongst these different resources (network links, central processing units, disk drives.) to attain best supply consumption, highest throughput, highest response time, and evading overload. To allocate load on diverse systems, diverse load balancing algorithms are utilized.

II. RELATED WORK

In [9] Fog processing can enhance the resource use proficiency of the user gadget, and tackle the issue about service balancing of the postponement sensitive applications. This paper inquires about on the structure of the fog processing, and receives Cloud Atomization Technology to transform physical nodes in various levels into virtual machine nodes. On this premise, this paper utilizes the graph separation hypothesis to assemble the fog processing's load balancing calculation in light of dynamic graph apportioning. The experiment results demonstrate that the structure of the fog registering after Cloud Atomization can build the system network adaptably, and dynamic load balancing instrument can adequately arrange system resources in addition by decreasing the utilization of node migration brought by system changes.

In [10] paper, demonstrate an algorithm to decrease the operational cost of cloud server organization with the assistance of fog gadgets, which can maintain a strategic distance from the income misfortune because of wide-zone arrange spread postponement and save system transfer speed while serving closer cloud clients. Since fog gadgets may not be claimed by a cloud specialist organization, they ought to be made up for serving the solicitations of cloud clients. When thinking about temperate pay, the ideal number of solicitations handled locally by each fog gadget ought to be chosen. Therefore, existing burden balancing plans created for cloud server organization can't be connected specifically and it is exceptionally important to update a cost-product input job balancing calculation for the cloud system. To accomplish the above point, authors initially detail a fog helped operational cost minimization issue for the cloud specialist co-op. At that point, authors plan a parallel and appropriated input job balancing calculation with low computational many-sided quality in light of proximal Jacobian rotating heading technique for multipliers.

In [11] the portrayal of computational resources and their ideal allotment among inhabitants with various prerequisites holds the way to actualizing compelling programming systems for such a worldview. To address this issue, an efficient structure for checking, investigating and enhancing system execution is proposed in this examination. In particular, a spiral premise work neural system is set up to change experiment jobs with conceptual depictions into particular resource prerequisites as far as their amounts and characteristics. Also, a novel numerical model is developed to speak to the perplexing resource allotment process in a multi-occupant figuring condition by considering need based inhabitant fulfillment, add up to computational cost and staggered input job adjust. To accomplish ideal resource assignment, an enhanced multi-objective hereditary algorithm is proposed in view of the elitist document and the K - implies approaches.

In [12] paper, authors saw that the homogeneous setup of jobs on heterogeneous nodes can be an essential source of load imbalance and in this way cause poor execution. Jobs ought to be allot in various designs to coordinate the capacities of heterogeneous nodes. To this end, authors proposed a self-versatile job tuning approach, Ant that naturally looks the ideal setups for singular undertakings running on various nodes. In a heterogeneous group, Ant first partitions nodes into various homogeneous sub-clusters in view of their equipment setups.

It at that point regards each sub-cluster as a homogeneous bunch and freely applies the self-tuning calculation to them. Insect at long last designs jobs with haphazardly chose setups and steadily enhances undertakings arrangements by replicating the arrangements from best performing assignments and disposing of poor performing arrangements. To quicken undertaking tuning and abstain from catching in nearby ideal, Ant utilizes hereditary calculation amid versatile job setup.

III. PROPOSED METHODOLOGY

In order to make a general model which work on various available data Indices a Teacher Learning Particle Swarm Optimization (TLPSO) genetic algorithm was used where data is classified without finding all possible combination. Here dynamic load was easily scheduled by this TLPSO where hybrid combination has increase the work efficiency.

1. Pre-Processing: As the dataset available for processing is present in different file format so, some pre-processing steps are required for the conversion of data into experimental environment. In this work data is inform of vectors of the job timing for different machine. So reading of vectors in string form and conversion of those string in proper numeric value is done in pre-processing. Collection of all vector is done in a single matrix is also done here.

2. TLPSO (Teacher Learning Particle Swarm Optimization): In this model Teachers Learning Particle Swarm Optimization Algorithm was used for assigning the incoming process to respected machine as per requirement. In this work genetic algorithm TLPSO is was proposed as this makes better load balancing.

3. Generate Population

Different combination of the entire jobs sequence were generate randomly in this step of proposed model. As paper has use TLPSO hybrid algorithm which is combination of Particle Swarm optimization with Teacher Learning approach. Hence population generation was done in this step for TLPSO algorithm, where each chromosome is a combination set of jobs in a unique sequence. So size of chromosome depends on number of jobs K, while population has m any number of chromosomes.

4. Gp = Random(K, m)----Eq. 4

Where m is number of chromosome in the population.

5. Fitness Function

Evlaution of any chromosome was done by this function where each chromosome having minimum makespan time is consider as best solution. So makeSpan time is estimate as shown in equation 2 which was used for finding the fitness value. This can be understand as let solution set Gp fitness value need to calculate. Than time taken by all machine to execute the each job in the batch is total MakeSpan time. So sum of all job execution time is term as the probable solution fitness value shown in equation (3).

J_{max}=Max_Execution_Time { J_1, J_2, J_3,………..J_n}---(2)

International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 1, Jan-Feb-2020, ISSN (Online): 2395-566X

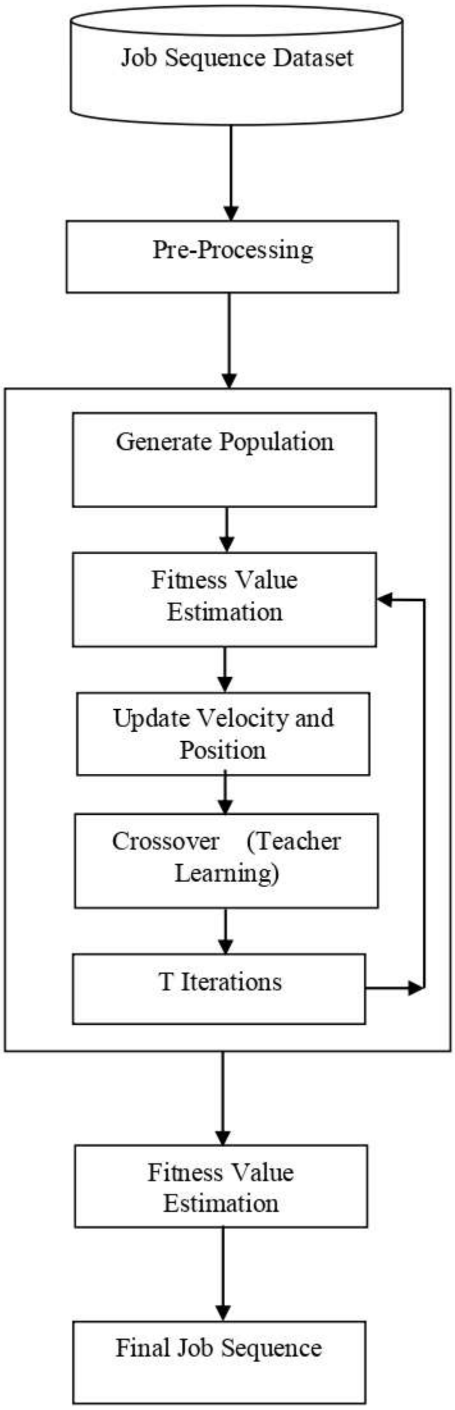

Fig.1. Block diagram of proposed hybrid model of load balancing.

6. Local and Global Best

Estimation of chromosome fitness value depends on makespan term value. So as per chromosome job sequences makespan value depends where minimum makespan chromosome act as best solution. Following algorithm find fitness value of chromosome:

Input: Gp, JET // JT: Job execution Time
Output: MakeSpan, Lbest // MakeSpan: Fitness Value
Loop I:m
$C_h \leftarrow G_p[m]$
MakeSpan[k,N] = Fitness(C_h, JET) // Rand value 0-1
EndLoop
Lbest ← Min(MakeSpan)

Out of all iterations minimum local best value act as global best. This Lbest and Gbest is for PSO algorithm, for first iteration both values are same.

7. Iteration Steps As TLPSO include Teacher and particle basic operation for population updation which iteratively call in this section. Such as PSO feature of velocity and position were update in each iteration. While Teach phase were involve for teacher learning. To merge both algorithm as per flow chart operation were explained below.

$$V_{i+1} = (V_i * t + (m - X_i) * r * (L_{best} - X_i) + r' * (G_{best} - X_i)$$

4. $X_{i+1} = X_i + *V_{i+1}$

Where r is random constant having value between 0-1.

Finally as per PSO algorithm velocity value were update as per intertia weight and constriction factor, once velocity getupdate than position feature of the chromosome were also update by Eq. In above equation V is velocity, X is position while t, r and r' are random number whose values ranchangege between 0-1. So as per X and V values crossover operation were performed.

8. Crossover (Teacher Phase): This phase was used for the crossover of the chromosomes by the single best solution from the population [13]. Here best solution act as a teacher and its selection is based on the minimum fitness value. In order to do crossover operation random position probable solution value is copied from the teacher chromosome and it was replaced to the non teacher chromosome. This improves the population quality. This can be understood as, let best solution is L_{best} than crossover operation.

5. $Gp[m] \leftarrow Crossover(G_p[m], G_p[L_{best}], r)$

6. where r=[1…n]Type equation here.

7.

9. Update Global Best

At the end of each iteration global best value were compared with current local best value. So when local best value chromosome value is better as compared to global best solution means if chromosome makespan value is less than global best value were update.

10. Experiment and Results

In order to conduct experiment and measure evaluation results MATLAB 2012a version software is use. This section of paper show experimental setup and results. The tests were performed on an 2.27 GHz Intel Core i3 machine, equipped with 4 GB of RAM, and running under Windows 7 Professional. Proposed algorithm was compared with SJFRL in [14].

11. Dataset

Table 1 Description of dataset with attributes [5].

Attributes	Set1	Set2
Number of Jobs	20	20
Number of Machines	10	5
Sample Size	20x10	20x5

Sample of number of jobs 15, number of machines 5, upper bound 1278 and lower bound 1232, processing times:
27 92 75 94 18 41 37 58 56 20 2 39 91 81 33 14 88 22 36 65
79 23 66 5 15 51 2 81 12 40 59 32 16 87 78 41 43 94 1 93
22 93 62 53 30 34 27 30 54 77 24 47 39 66 41 46 24 23 68 50
93 22 64 81 94 97 54 82 11 91 23 32 26 22 12 23 34 87 59 2
38 84 62 10 11 93 57 81 10 40 62 49 90 34 11 81 51 21 39 27

12.Evaluation Parameter

Makespan is defined as the time required for processing all thejobs or the maximum time required for completing a given set of jobs. Minimization of makespan ensures better utilization of the machines and leads to a high throughput [7].

$$J_{max}=\text{Max} \{ J_1, J_2, J_3, \ldots\ldots J_n\}$$

Total flowtime is defined as the sum of completion time of every job or total time taken by all the jobs. Totalflowtime of the schedule is computed using equation [8]:

$$F = \sum_{i}^{n} J_i$$

Completion Time variance is defined as the variance about the mean flowtime and iscomputed using equation [9]:

$$V = \frac{1}{n}\sum_{1}^{n}(J_i - \bar{F})^2$$

Where $\bar{F}$ is the mean flowtime.

Relative Percent Deviation (RPD)

$$RPD = \left[\frac{G - C^*}{C^*}\right] \times 100$$

where, G represents the global best solution obtained by the proposed algorithm for a given problem and C^*represents the upper bound value.

Results

Table 2 Makespan based comparison of load balancing algorithms.

Algorithms	Job Sequences	
	n= 20 Jobs, 5 machine	n= 20 Jobs, 10 machine
TLPSO	1301	1723
SJFRL	1672	1728

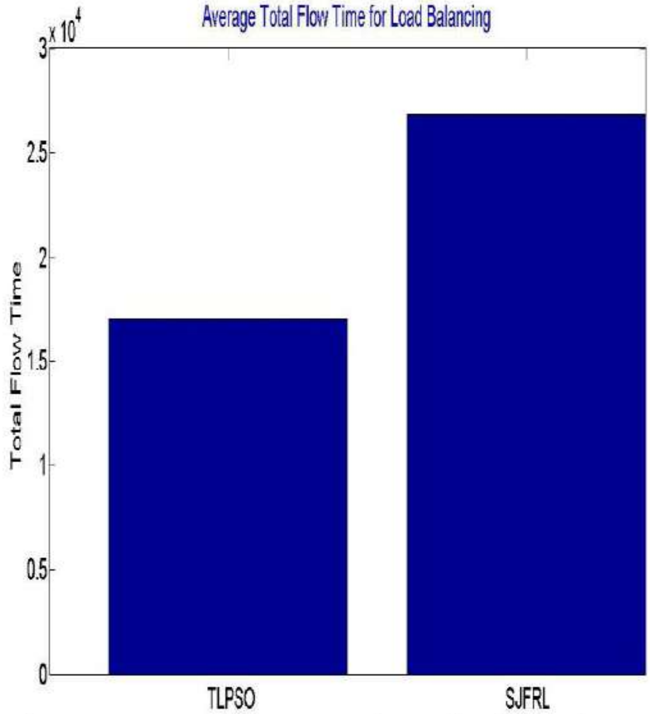

Fig. 2 Average Makespan comparison of load balancing algorithms.

Above table 2 and fig. 2 shows that proposed model of hybrid teacher learning and particle swarm optimization algorithm has reduce the makes span time of the different job scheduling sets. This reduction of work was done by improving PSO algorithm by crossover operation of teacher learning.

Table III: Total flow Time based comparison of load balancing algorithms.

Algorithms	Job Sequences	
	n= 20 Jobs, 5 machine	n= 20 Jobs, 10 machine

TLPSO	14932	19138
SJFRL	24566	29108

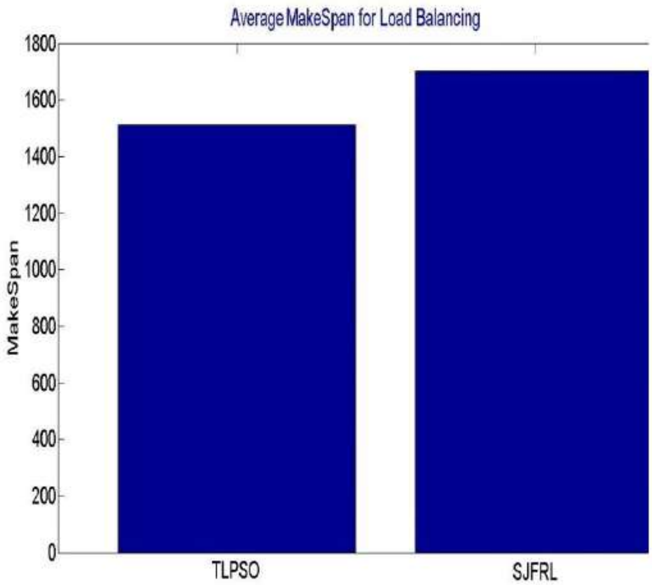

Fig. 3 Average total flow time comparison of load balancing algorithms.

Above table 3 and fig. 4 shows that proposed model of hybrid teacher learning and particle swarm optimization algorithm has reduce the Total flow Time of the different job scheduling sets. This reduction of work was done by improving PSO algorithm by crossover operation of teacher learning.

Table IV: RPD Makespan based comparison of load balancing algorithms.

	Job Sequences	
Algorithms	n= 20 Jobs, 5 machine	n= 20 Jobs, 10 machine
TLPSO	13.1304	8.91277
SJFRL	45.3913	9.22882

Above table 4 shows that proposed model of hybrid teacher learning and particle swarm optimization algorithm has reduce the RPD makes span time of the different job scheduling sets. This reduction of work was done by improving PSO algorithm by crossover operation of teacher learning.

Table V: RPD Total Flow Time based comparison of load balancing algorithms.

	Job Sequences	
Algorithms	n= 20 Jobs, 5 machine	n= 20 Jobs, 10 machine
TLPSO	4.41958	19.6125
SJFRL	71.7902	81.925

Above table 2=5 shows that proposed model of hybrid teacher learning and particle swarm optimization algorithm has reduce the RPD Total Flow time of the different job scheduling sets. This reduction of work was done by improving PSO algorithm by crossover operation of teacher learning.

IV. CONCLUSIONS

Load balancing of jobs with dynamic requirement of resources is always a great issue in field of computer science research. This paper has generate dynamic sequence of job as per its resource by using hybrid genetic algorithm. Use of PSO and Teacher Learning improve the work efficiency by reducing the makespan of the work. Modification in genetic algorithm of PSO by involving crossover operation as per teacher learning has increase the chromosome quality in each iteration. This paper has perform experiment on different set of jobs with different resource requirement. Result shows that proposed model has reduce the makespan by 11.05% as compared to existing method SJFRL in [14]. It was also obtained that proposed model has improved the Total Flow Time value by 36.52%. In future one can adopt other gentic algorithm with some neural network model for better learning of job sequence patterns.

REFERENCES

[1]. Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, Qiaozhu Mei "Rumor has it: Identifying Misinformation in Microblogs" Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1589–1599, Edinburgh, Scotland, UK, July 27–31, 2011
[2]. Tetsuro Takahashi, Nobuyuki Igata, "Rumor detection on twitter," SCIS-ISIS 2012, Kobe, Japan, November 20-24, 2012, IEEE.
[3]. Aditi Gupta and Ponnurangam Kumaraguru "Credibility Ranking of Tweets during High Impact Events,"ACM, 2012.
[4]. Sahana V P, Alwyn R Pias, Richa Shastri and Shweta Mandloi, "Automatic detection of Rumoured Tweets and finding its Origin," Intl. Conference on Computing and Network Communications (CoCoNet'15), Dec. 16-19, 2015, Trivandrum, India, Journal: IEEE.
[5]. Zhe Zhao, Paul Resnick, and Qiaozhu Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," In Proceedings of the 24th International Conference on World Wide Web. ACM, 1395–1405
[6]. Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen,Kam-FaiWong and Meeyoung Cha, "Detecting Rumors from Microblogs with Recurrent Neural Networks," Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), 2016.

[7]. Arkaitz Zubiaga, Maria Liakata And Rob Procter, “Exploiting Context For Rumour Detection In Social Media,” Springer, 2017

[8]. GORDON W. ALLPORT AND LEO POSTMAN, “AN ANALYSIS OF RUMOR,” Downloaded from http://poq.oxfordjournals.org/ at University of California, San Francisco on December 11, 2014

[9]. Zubiaga A, Liakata M, Procter R, Wong Sak Hoi G, Tolmie P (2016) Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads. PLoS ONE 11(3): e0150989. doi:10.1371/journal.pone.0150989

[10]. A. Friggeri, L. Adamic, D. Eckles, and J. Cheng, “Rumor Cascades,” Icwsm, pp. 101–110, 2014.

[11]. https://pdfs.semanticscholar.org/d051/ba0a904e4b4c45a2af145aa29b8490bbbc5c.pdf

[12]. https://blogs.msdn.microsoft.com/andreasderuiter/2015/02/09/performance-measures-in-azure-ml-accuracy-precision-recall-and-f1-score/.

[13]. Anuradha Purohit, Deepika Atre, Payal Jaswani, and Priyanshi Asawara, “Text Classification in Data Mining,” International Journal of Scientific and Research Publications, Volume 5, Issue 6, June 2015.

[14]. https://www.cse.iitk.ac.in/users/se367/10/presentation_12288888888ocal/Binary%20Classification.html.

[15]. Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata And Rob Procter, “Detection and Resolution of Rumours in Social Media:A Survey,” arXiv:1704.00656v3 [cs.CL] 3 Apr 2018.

[16]. Pooja Kewat , Roopesh Sharma , Upendra Singh , Ravikant Itare, “Support Vector Machines Through Financial Time Series Forecasting ”, Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of IEEE , 20-22 April 2017 ,pp. 1-7.

[17]. Yashika Mathur ,Pritesh Jain ,Upendra Singh, “Foremost Section Study And Kernel Support Vector Machine Through Brain Images Classifier ”, Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of IEEE , 20-22 April 2017 ,pp.1-4.

[18]. Vineeta Prakaulya ,Roopesh Sharma ,Upendra Singh, “Railway Passenger Forecasting Using Time Series Decomposition Model ”, Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of IEEE , 20-22 April 2017 ,pp.1-5.

[19]. Sonal Sable ,Ankita Porwal ,Upendra Singh , “Stock Price Prediction Using Genetic Algorithms And Evolution Strategies ”, Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of IEEE , 20-22 April 2017 ,pp.1-5.

[20]. Rohit Verma ,Pkumar Choure ,Upendra Singh , “Neural Networks Through Stock Market Data Prediction” , Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of IEEE , 20-22 April 2017 ,pp.1-6.

[21]. Dinesh Bhuriya ,Girish Kaushal ,Ashish Sharma, Upendra Singh ,“ Stock Market Predication Using A Linear Regression ”, Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of IEEE , 20-22 April 2017 ,pp. 1-4.

[22]. Ashish Sharma ,Dinesh Bhuriya ,Upendra Singh, “Survey Of Stock Market Prediction Using Machine Learning Approach” , Electronics, Communication and Aerospace Technology (ICECA), 2017 International conference of IEEE , 20-22 April 2017 ,pp.1-5.