Indian Journal of Science and Technology, Vol 8(23), DOI: 10.17485/ijst/2015/v8i23/70854, September 2015
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Using Harmony Search Algorithm for Load Balancing in Cloud Computing

O. Norouzpour[1*] and N. Jafarzadeh[2]

[1]Department of Engineering Computer Software Science, Qeshm International Branch, Islamic Azad University, Qeshm, Iran; omidnoroozpour@hotmail.com
[2]Department of Computer Science, South Branch, Islamic Azad University, Tehran, Iran; n.jafarzadeh@azad.ac.ir

Abstract

Background: In this paper, we have surveyed the load modification and the trust capability in the cloud computing data centers. **Method**: The components of harmony search Algorithm include the harmony memory, the pitch adjustment rate and the random selection. The harmony memory checks and selects the responses. HMCR (Harmony Memory Considering Rate) is used for showing the possibility of selection from the memory and the pitch adjustment rate is used for causing a new harmony. **Findings:** Standard deviation and the resource exploitation average are the most important criteria at the load balancing. In this paper, the proposed Algorithm has been compared with the Genetic Algorithm and the simulation results show that the proposed Algorithm has a better competence than the Genetic Algorithm as the HSD (Harmony Search Degertekin) rate of the proposed method is less than the HSD rate of Genetic Algorithm and the average of utilization from resources in the proposed method is more as well as with the increasing number of tasks, the performance of the algorithm is improved. **Application/Improvement:** Using this method of load balancing, the satisfaction of users is maximized, the time of response is minimized and the utilization of resources is increased.

Keywords: Cloud Computing, Dynamic Load Balancer, Genetic Algorithm, HSD Algorithm, Load Balancing

1. Introduction

Cloud computing has many advantages such as: establishment low price, usage-based pay and service flexibility which have made cloud computing more attractive. Cloud computing is facing many challenges as it is rather new. One of the challenges is the load balancing by autonomous computing. Load balancing is a process for redistribution of works among nodes in an equal way in distributed systems. It improves the efficiency of the works and the sources. Load balancing occurs when the amount of load in one place is heavier and other places are inactive. The Algorithm dynamic balance of charge has the important discussions such as: the balance and estimation of the load, comparing the levels of load and the performance index and stability. Because, the process encounters the decrease and increase in cloud environment every moment, it is necessary to use the dynamic Algorithm[2].

Implementation of load balance is a difficult task in cloud computing. In classic methods, obtaining the optimal response is time consuming, thus application of more initiative methods is recommended In this article, the Algorithm of the harmony search has been used that it acts dynamically and it is a new meta heuristic which it begins to search in the space of the issue solution by the generation from the rector solution in shape of the harmony memory, that it moves toward the optimal spaces with a possibility approach. In this part, we proceed to review many produced ways for the load balancing.

2. 1 Round – Robin

In this Algorithm, the first node is chosen accidently and then, the whole nodes appropriate in form of RR. This Algorithm will not be suitable for the cloud computing. Because, some of the cloud computing is loaded extremely and others aren't like this[1,3].

*Author for correspondence

2.2 Active Clustering Algorithm

As grouping, the nodes act alike. The processes including: A node starts the own credible neighbor nodes that is different from the previous node. After the interface node, makes a connection between a neighbor that is a like the initial node, then, the interface node, separates the connection of the initial node with itself. All of the above processes are repeated regularly. In this Algorithm, the performance of system increases as the resources increases, in result, the operating power ascends by using this useful resources. This Algorithm, reduces as the diversity of system increases[4].

2.3 Throttled Load Balancer

Throttled load balancer is a dynamic load balancing algorithm. In this algorithm, the client first requests the Load balancer to find a suitable Virtual machine to perform the required operation. In Cloud computing, there may be multiple instances of virtual machine. These virtual machines can be grouped based on the type of requests they can handle. Whenever a client sends a request, the load balancer will first look for that group, which can handle this request and allocate the process to the lightly loaded instance of that group[5,10].

3. Harmony Search Algorithm

Algorithm of the harmony search[8] is the simplest and the latest of Meta–heuristic approaches that take an inspiration from playing of simultaneous music band in searching the optimal answerable at the optimization issues. In other words, throe is a similarity between, finding an optimal solution in the complex problem and the performance of playing music. First time, Game offered this solution in 2001 – according to the logic of this Meta–heuristic approach, to try for having harmony in a playing music is similar, finding an optimal solution in the optimization problems. HS as genetics is a member of the vulnerary Algorithm. Based on harmony algorithm the presented timing has components:

Harmony memory, Sound volume regulation Random selection, Band Width (BW) and the level of Harmony memory determine whether new responses and harmonies can be acceptable or not. To utilize this memory more efficiently a parameter will be introduced to show the likely selection of harmony memory HMCR.

The second factor is the regulation of sound volume which has the parameters of likely regulation and volume change PAR. The latter parameters are used in creating a new harmony.

The improvement and reformations:

Degertekin, HMS, brought up a new idea in the harmony memory production in 2008 and, instead of the whole members of memory are produced randomly, the double amount of the random response produced and the numbers of HMS of the best, placed in the harmony memory. Mahdavi and the colleague applied the variable parameter instead of the fixed parameter[6,7].

4. The Suggestive Technique

Steps of the proposed Algorithm

Step 1: Every harmony is an array from resources that the length of assigned tasks for it is based on the window size, that, it perches to FCFS in line. The objective function is one of the main features at the Algorithm of the load balancing that the level of competence is based on the performance time for the transmission and the tasks to resources.

$$\text{Load}(\text{ri}) = \text{Current Load}(\text{ri}) + \sum_{i=0}^{n} \frac{\text{Tasklength}(j)}{pp(r_i)} + \sum_{j=1}^{n} \frac{\text{Osizeof}(j) + \text{lsizeof}(j)}{PW(r_i)}$$

Relation (1)

Where, ri the desired resource, current load () is the current load of each resource, task length gives the processing Requirement of each take, PP (ri) processing power of each resource, PW is the bandwidth between each resource with Scheduler, Isize and OSize offer the size of input and output files. Makes pan is a time for ending of the assigned final task to the resources that it has present in the Relation (2). In this relation N is the number of resources.

Make span = Max (load (ri)) is {1, 2, 3 ... n}

Relation (2)

Fundamentally, the maximum time for the completing task is in the whole processors. The rate of operation from each resource is the load proportion to the makes pan which has been shown in Relation (3).

U (ri) = load (ri)/make span

Relation (3)

The average of operation from each resource has been shown in Relation 4 that it as the effective criterion is a solution at the mount of competence.

$$\mathrm{Avg}(u)=\frac{\sum_1^n u(k)}{N}$$

Relation (4)

The amount of resource load dispersion the load average is one main factors for obtain competence, which is expressed as the standard deviation.

$$\sigma=\sqrt{\frac{\sum|_{=1}^{N}\left(\mathrm{loadi}-\overline{\mathrm{load}}\right)^2}{N}}$$

Relation (5)

The aim function is defined a function of the makes pan and it is the operation average from the resources and the standard deviation that it has been shown in the Relation (6). In this aim function, makes pan and the standard deviation have a reverse relation. If every characteristic increases, the competence of solution will reduce and the load isn't distributed balancing.

The aim function with the operation average from resources has a straight relation because; to increase of the operation average is an indicative for the proper load balancing. In this relation, we have multiplied three characters. Because make span is a great amount and its converse is a small amount that in compared to the operation average from resources, it is trifle. Then, if we add three-characters of make span amount, it won`t be much influence on the competence.

$$\mathrm{Fitness}=\frac{1}{\mathrm{Makespan}}\times\mathrm{AveU}\times\frac{1}{\sigma}$$

Relation (6)

Step 2: we have meant the parameters amount, BW, PAR, BW, in the proposed Algorithm BW, PAR, HMRC. The function from the solution competence at the previous generation that it is expressed in the Relation (7) if the best Harmony have had a better competence than the previous generation harmony, HMCR will increase, otherwise, HMCR will decrease and this can search in the whole solution space .

$$\mathrm{HMCR}(K)\begin{cases}\left\{\mathrm{HMCR}_{\min}+\left(\mathrm{HMCR}_{\max}-\mathrm{HMCR}_{\min}\right)\times\dfrac{f_{best}}{f_{k-1}} & f_{k-1}\geq f_{best}\\ \mathrm{HMCR}_{\max} & f_{k-1}<f_{bset}\end{cases}$$

$$\mathrm{PAR}(k)\begin{cases}\mathrm{PAR}_{\max}-\left(\mathrm{PAR}_{\max}-\mathrm{PAR}_{\min}\right)\times\dfrac{f_{best}}{f_{k-1}} & f_{k-1}\geq f_{best}\\ \mathrm{PAR}_{\min} & f_{k-1}<f_{bset}\end{cases}$$

$$\mathrm{bw}(k)\begin{cases}\mathrm{bw}_{\max}-\left(\mathrm{bw}_{\max}-\mathrm{bw}_{\min}\right)\times\dfrac{f_{best}}{f_{k-1}} & f_{k-1}\geq f_{best}\\ \mathrm{bw}_{\min} & f_{k-1}<f_{bset}\end{cases}$$

Relation (7)

Step 3: We produce a random response that is double HMS and place the number of HMS from the best HMS in Harmony memory.

Step 4: In this step, if the condition of HMCR realize, First, the player survey the element in the first two harmony, if two harmony be equal he will trust in his memory. And he places the element without changing in the New Harmony; otherwise he does the improvement approach. In case, the produced element doesn`t place at the high or low range, a number will produce with an even distribution by accident (Randomly). That it presents the results instead of the high or low limit of element. And it prevents from assigning of tasks to the first or final resource which aren`t the best resources necessarily.

Step 5: In fourth step. if the first and second harmony be similar directly we will simple from them, so there is the possibility of similar between the new harmony and the available harmony, so that, the produced harmony is compared to the available harmony in the memory and if they were similar, the leap will be done on it.

Step 6: If a resource has an extra load and the task is assigned to it, that process will be transferred to the resource that has the load lowest.

Step 7: If the competence of the New Harmony be better than the available worst harmony, this harmony will replace with the worst harmony. This process continues to realize one of the stop conditions. The first condition is the specified number and the second condition is the available best competence, in 1% percent of the total repetitions.

5. Simulation

In this part, the proposed harmony Algorithm is compared to the genetics Algorithm[9]. In this test, the standard deviation and the operation average of resources that are the important criteria are surveyed and the numbers

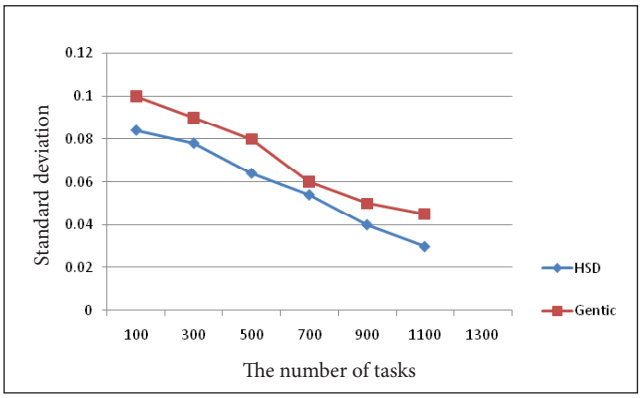

Figure 1. Comparison standard deviation genetic algorithm and HSD.

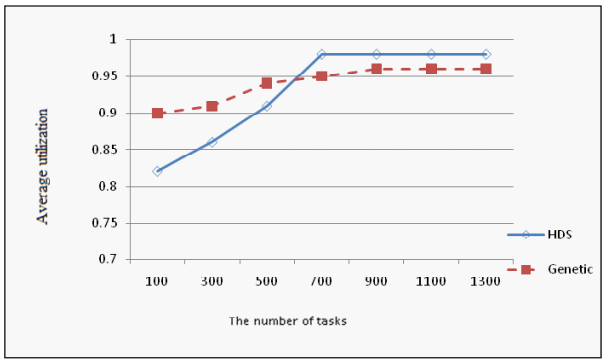

Figure 2. Comparison average utilization of genetic algorithm and HSD.

of tasks have changed from 100 to 1300. As you observe, whatever the numbers of tasks increase, the effectiveness of Algorithm will improve in Figure 1 and 2. In this test, the standard deviation of the proposed approach is fewer than the genetic Algorithm (Figure 1.) and the operation average of resources is more in the proposed approach (Figure 2).

6. Conclusion

The cloud computing is a new concept. With regard to the advantages and added value for the market and the users, this method is expanding rapidly. In order to fulfill the promises of cloud computing and also to make cloud computing users, trust it without concern, some parts of it should be studied. To fulfill this aim this paper attempts to investigate load balancing and assurance ability (disruption toleration) in cloud computing data center. In this project, it is used the search harmony Algorithm that, it dose the load balancing dynamically and also, it has an easy implementation of time and has less performance than other evolutionary thing Algorithms. And its characteristics are scored the diverse and, after producing of the new harmony, if harmony repeat frequently and has a good result, it will repeat again, otherwise, the changes is done on it, if the new solution is similar to the previous solution, the leap will apply, and, instead of, the whole members of memory are produced randomly, the double amount of the random response produced and the numbers of HMS of the best, placed in the harmony memory.

7. References

1. Moharana S, Ramesh RD, Powar D. Analysis of load balancers in cloud computing. IJCSE. 2013 May; 2(2):101–8.
2. Gupta R. Review on existing load balancing techniques of cloud computing. International Journal of Advanced Research in Computer Science and Software Engineering. 2014 Feb; 4(2):168–71.
3. Kumar R, Madhu Priya M, Shahu Chatrapati K. Effective distributed dynamic load balancing for the clouds. IJERT. 2013 Feb; 2(2):1–6.
4. Kansal NJ, Chana I. Cloud load balancing techniques: a step towards green computing. IJCSI International Journal of Computer Science Issues. 2012 Jan; 9(1):238–46.
5. Kokilavani T, Amalarethinam DIG. Load balanced min-min algorithm for static meta-task scheduling in grid computing. International Journal of Computer Applications. 2011 Apr; 20(2):43–9.
6. Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems. Mathematics and Computation. 2007 May; 188(2):1567–79.
7. Degertekin SO. Optimum design of steel frames using harmony search algorithm. Structural and Multidisciplinary Optimization. 2008 Oct; 36(4):393–401.
8. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: Harmony search. Simulation. 2001 Feb; 76(2):60–8.
9. Zomaya AY, Teh YH. Observations on using genetic algorithms for dynamic load-balancing. IEEE Transactions on Parallel and Distributed Systems. 2001 Sep; 12(9):899–911.
10. Sran N, Kaur N. Comparative analysis of existing load balancing techniques in cloud computing. International Journal of Engineering Science Invention 2013 Jan; 2(1):60–3.