International Journal of Innovative
Computing, Information and Control
Volume **13**, Number **3**, June **2017**

ICIC International ©2017 ISSN 1349-4198
pp. 1021–1028

# LOAD BALANCING ALGORITHM BASED ON COMPUTATIONAL LOAD ENTROPY

Weibo Zhou[1,2], Yong Zhong[1] and Shiqing Nie[1,2]

[1]Chengdu Institute of Computer Applications
Chinese Academy of Science
No. 9, Sec. 4, Remin South Road, Chengdu 610041, P. R. China

[2]University of Chinese Academy of Science
No. 19A, Yuquan Rd., Shijingshan District, Beijing 100049, P. R. China
findz@qq.com

Received October 2016; revised February 2017

Abstract. *Under the distributed environment, load imbalance of cluster nodes usually causes many problems such as lower resource utilization and longer response time. According to the theory of entropy, we define computational load entropy in this study and further based on it, a load balancing algorithm (CLE) is proposed. The proposed algorithm has achieved the quantitative regulation of load balancing in distributed systems by means of the system load determination, single node determination, and load regulation. Through the comparison tests with the RR algorithm, the proposed algorithm is verified for its good balancing effect on the load of the computed tasks in distributed systems. Results also show that the proposed algorithm can effectively control the load imbalance of the cluster and improve the resource utilization of the cluster system.*
**Keywords:** Computation load entropy, Load balancing, Distributed system, Cluster system

1. **Introduction.** The study of load balancing is of vital importance in the model of distributed storage and computing. It mainly includes the storage load balancing and the computing load balancing. The former aims to solve the problems of the equilibrium distribution of storage resources while the latter is targeted to the equilibrium distribution of computing resources. A good load balancing scheme can make full use of the system resources, avoid the bottleneck of performances and the waste of computing resources, and effectively improve the overall performance of a distributed system.

There are generally two approaches of computing the load balancing, i.e., static load balancing and dynamic load balancing [1]. The static load balancing approach carries out the task allocation and the resource scheduling according to the prior condition. Its evaluation indicators are mostly based on the prior conditions of some physical properties such as CPU computing power, storage capacity, network bandwidth, and memory size. While the dynamic load balancing approach executes dynamic regulation through the real-time collection and analysis of the load of each node in the system.

In order to save costs, the equipment that most nodes use in a distributed system is of low reliability, which inevitably leads to frequent node failures. Such failures often result in the reality that the computing resources in the whole distributed system have great uncertainty which has brought a big challenge to the resource allocation and the task scheduling. Traditional static load balancing algorithm cannot meet the requirements and the dynamic load balancing theory has become a hot spot in the research of load balancing. The concept of the computational load entropy that is proposed in this paper

is an extension of the dynamic load balancing theory and is a quantitative analysis of the equilibrium distribution of computing resources in the whole cluster system from a macro point of view.

The main contribution of this study is to provide a new method to measure and balance the computation load. The organization of this paper is as follows: related works are presented in Section 2; the load balancing algorithm is defined in Section 3; the load balancing algorithm is described in Section 4; simulation tests are shown in Section 5; finally, conclusions are made in Section 6.

2. **Related Works.** The concept of entropy was put forward in 1854 by T. Clausius [2]. In the field of thermodynamics, entropy is a physical quantity denoting the chaos degree of a system. When the system reaches the equilibrium of thermodynamics, the entropy value is the largest. In information theory, entropy is used to measure the expectation of a random variable and it is also called information entropy which represents the amount of information loss of a signal in the process of transmission before the signal is received.

Essentially the load balancing in the distributed system can be understood as the internal resource equilibrium of the system. Through the analysis of the internal similarity between the thermodynamic system and the distributed system, [3] discussed the relationship between the equilibrium of load balancing and energy from the perspective of entropy. [4] constructed an entropy optimization model by using the maximum entropy and the entropy increase principle. [4] also put forth a dynamic weighted evaluation model for resources in the distributed system. [5] defined parallel computing load entropy and proved its rationality as measurement for the degree of system load balancing. [5] also proposed a load balancing algorithm for the homogenous cluster based on parallel computing load entropy. However, this algorithm did not consider the heterogeneity of the cluster and the communication delay of nodes in a large-scale cluster, thus it cannot effectively regulate and avoid the load imbalance.

By contrast, the proposed load balancing method focuses on the computing task and the node computing ability without considering the node homogeneity and heterogeneity. Thus it can deal with the load balancing of computing resources in the heterogenous cluster.

3. **Computational Load Entropy.** In thermodynamics heat always goes from the high temperature object spontaneously to the low temperature one and eventually reaches the thermal equilibrium state when the system reaches the maximum entropy [6]. In the field of informatics, the information entropy is usually used to measure the amount of information. If the higher the uncertainty of information is, the larger the amount of information is, the greater the information entropy becomes. If the lower the uncertainty of the information is, the less the amount of information is, the smaller the information entropy becomes [7]. Based on the theory of entropy and combined with the characteristics of dynamic load balancing in the distributed system, this research puts forward the concept and theory of computational load entropy and presents the related load balancing algorithm.

**Definition 3.1.** *The total number of jobs (N) is the cumulative number of jobs of all the nodes to be processed in the system within one cycle. Suppose that there are n nodes in a distributed system and the number of jobs of node i to be processed is $N_i$, and then the total number of jobs in the system is defined as*

$$N = \sum_{i=1}^{n} N_i \tag{1}$$

**Definition 3.2.** *The computing capacity (C) is the maximum number of jobs that can be processed per unit time by the system when operating at full capacity. Suppose that the computing capacity of node i is $C_i$, and then the computing capacity of the system is defined as*

$$C = \sum_{i=1}^{n} C_i \tag{2}$$

**Definition 3.3.** *The node computing ratio ($P_i$) is the ratio of the node computing capacity to the system computing capacity. The node computing ratio of node i is defined as*

$$P_i = \frac{C_i}{C} \tag{3}$$

**Definition 3.4.** *The node weighted loading ($L_i$) is the radio of the number of jobs of a node to be processed within one cycle to the node computing ratio of that node. The node weighted loading of node i is defined as*

$$L_i = \frac{N_i}{P_i} \tag{4}$$

**Definition 3.5.** *The ratio of node weighted loading ($L_{r_i}$) is the radio of the node weighted loading of a node to the system node weighted loading. Suppose the ratio of the node weighted loading of node i is $L_{r_i}$ which is defined as*

$$L_{r_i} = \frac{L_i}{\sum_{i=1}^{n} L_i} \tag{5}$$

The objective of load balancing algorithm is to distribute $N$ requested jobs to $n$ computing nodes. If the method of random distribution is taken, these $n$ nodes will have different degrees of loading, which will result in significant uncertainty. We need to find a way to measure the degree of loading that has such uncertainty. Based on the entropy theory, the definition of computational load entropy is given as follows.

**Definition 3.6.** *In distributed cluster systems, if there are n computing nodes, then the computational load entropy H of distributed systems is defined as*

$$H = -\sum_{i=1}^{n} (L_{r_i} * \log_2 L_{r_i}) \tag{6}$$

The computational load entropy is an extension of entropy theory in the field of system balancing and it has all the properties of entropy. According to the principle of entropy increasing and the principle of the maximum entropy, it can be proved that the entropy value of the computational load entropy always increases gradually from the minimum value $H_0$ of the most unbalanced state till to the maximum value $H_{\max}$ of the complete balancing state. The load balancing degree in the system can be simply and effectively measured by evaluating the entropy value in the computational load entropy.

4. **Load Balancing Algorithm.** The proposed load balancing algorithm determines the load balancing degree of the system according to the entropy value in the computational load entropy and the load regulation is carried out when the entropy value is less than a predefined threshold. The proposed algorithm has three components: system load determination, single node load determination, and load regulation.

4.1. **System load determination.** The system load determination is a key component in the proposed algorithm. If the result of the system load determination is balanced, there will be no need for the subsequent single node load determination and load regulation. Because performing the load regulation needs to exhaust a certain cluster resources along with large system overhead, a balanced threshold is set in the proposed method. When the entropy value $H$ in the computational load entropy is less than the threshold, it indicates that the load imbalance of the whole system has reached a certain degree and the load regulation needs to be carried out.

In a distributed system, the node computing ratio $P_i$ of each node can be determined based on the prior knowledge; thus, the process of computing the computational load entropy only needs to collect the number of jobs $N_i$ of each node. The determination process of the system load includes:

1) Set the balance factor $\alpha$ $(0 < \alpha < 1)$, hence the balance threshold $H_\alpha = \alpha H_{\max}$ and $0 < H_\alpha < H_{\max}$;

2) Collect the number of jobs $N_i$ of each node and compute the actual computational load entropy $H$ of the whole system according to Equation (6);

3) Determine whether the system is balanced or not. If $H >= H_\alpha$, it indicates that there is a good load balance in the system and it does not need to carry out the load regulation. If $H < H_\alpha$, it indicates that there is a high degree of load imbalance in the system and the load regulation needs to be performed.

4.2. **Single node load determination.** If the whole system is load imbalance, the load regulation needs to be performed. It also needs to compute which nodes have more load and which nodes have less load. Then an orderly queue of over loaded nodes and an orderly queue of empty loaded nodes are constructed according to the load condition of each node.

The determination process of the single node load includes:

1) Compute the weighted load of each node. The number of jobs $N_i$ of each node can be collected and the computing ratio $P_i$ of each node can be determined based on the prior knowledge. Hence the weighted load $L_i$ of each node can be quickly computed according to Equation (4);

2) Compute the weighted load of the whole system.

$$L = \frac{\sum_{i=1}^{n} N_i}{\sum_{i=1}^{n} P_i} = \frac{N}{1} = N \tag{7}$$

3) Construct the orderly queue of over loaded nodes and the orderly queue of empty loaded nodes. If $L_i > L$, it indicates that node $i$ is over loaded and insert $i$ into the over loaded queue according to the max heap sort order. If $L_i < L$, it indicates that node $i$ is empty loaded and insert $i$ into the empty loaded queue according to the min heap sort order.

4.3. **Load regulation.** The aim of load regulation is to adjust the jobs from the over loaded nodes to the nodes of less over loaded and makes balanced the load of the whole distributed cluster. Its detailed process is as follows.

1) Compute the number of jobs that are needed to be regulated from the over loaded nodes.

$$\Delta N_i = N_i - N * P_i \tag{8}$$

2) Compute the number of jobs that are needed to be regulated from the empty loaded nodes.

$$\Delta N_i' = N * P_i - N_i \tag{9}$$

TABLE 1. Algorithm of load regulation

**Algorithm**

```
//BigQ denotes the over loaded queue of the max heap.
  SmallQ denotes the empty loaded queue of the min heap.

1)  while(BigQ){
2)      Node(i) = BigQ[head++];
3)      ΔN_i = N_i − N * P_i;
4)      Node(j) = SmallQ[head++];
5)      ΔN_j = N * P_j − N_j;

6)      if(ΔN_i > ΔN_j){
7)        ΔN_i = ΔN_i − ΔN_j;
8)        Insert Node(i) into BigQ;
9)        }
10)      else{
11)            ΔN_j = ΔN_j − ΔN_i
12)            Insert Node(j) into SmallQ;
13)         }
14) }
```

3) According to the number of the jobs to be regulated, regulate these jobs orderly and repeatedly. The detailed algorithm is presented in Table 1.

Aiming to balance the load, the proposed load balancing algorithm regulates the load among each node in the system by means of the above three steps of system load determination, single node load determination, and load regulation. Such dynamic load balancing mechanism that first considers the whole and then the parts can control the balancing degree of the computational load among the nodes and maximize the utilization of the system resources.

5. **Tests and Analysis.** To verify the feasibility and effectiveness of the proposed algorithm, we build a CloudSim [8] cloud computing environment and compare our algorithm with the Round Robin (RR) scheduling algorithm. The RR scheduling algorithm which is used as the algorithm of task scheduling in the CloudSim is one of the most commonly used polling scheduling algorithms. In addition, the algorithm in [5] is designed for the homogeneous cluster and not for the heterogeneous cluster; thus, it is not selected for comparison with our proposed algorithm.

5.1. **Test 1.** In this test, we fix the scale of the cluster and increase the job numbers. The virtual machine size is set to 8 nodes. The simulation test is first performed according to the Round Robin (RR) algorithm. Next, the bindCloudletToVm() method in the DatacenterBroker class is rewritten by the proposed load balancing algorithm (CLE) based on the computational load entropy, and take the number of the jobs as the parameter to test the completion time of all the jobs. Test results are shown in Figure 1.

In Figure 1, when the number of jobs is less, there is no obvious difference in the performance of both the RR algorithm and the proposed CLE algorithm. Due to the load determination and regulation in the proposed algorithm, the average job response time is slightly longer than the RR. When the number of jobs increases, due to the lack of load balancing strategy in the RR algorithm, there is a significant rise in the average job

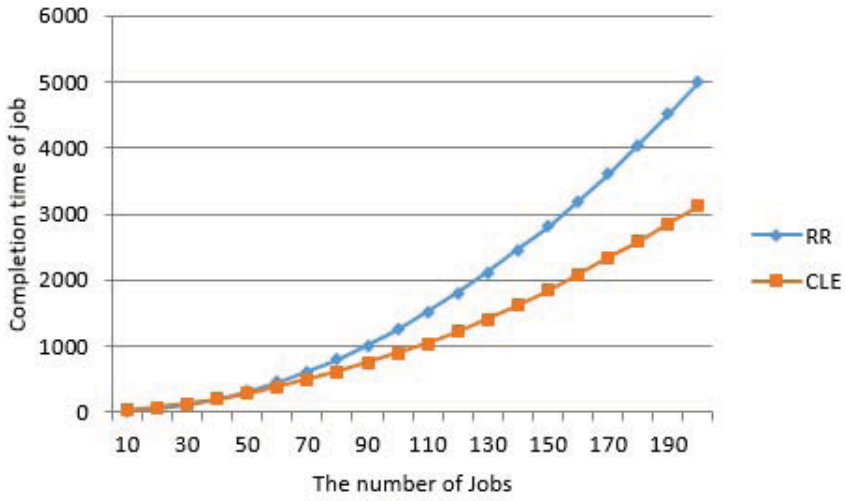


FIGURE 1. Job completion time of 8 nodes

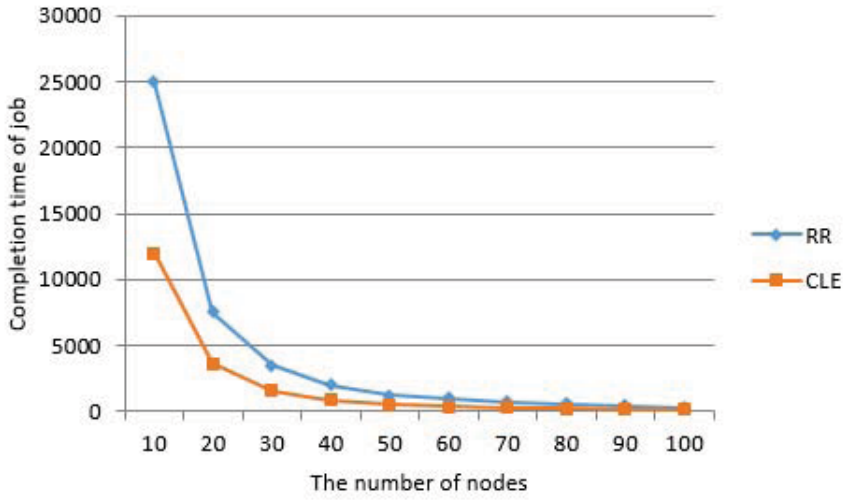


FIGURE 2. Completion time of 500 jobs

response time in the RR. However, the proposed CLE algorithm can properly regulate the system load and greatly reduce the average job response time. When the number of jobs reaches 200, about 40% job completion time is saved in the CLE algorithm.

5.2. **Test 2.** In this test, the job numbers are set to 500 and the number of the nodes is increased gradually. Performances of both algorithms are tested and the results are shown in Figure 2.

In Figure 2, when the computing nodes are less, there are relatively more jobs, the computing nodes are of heavy load, so the job completion time is longer. Because the CLE algorithm can dynamically regulate the cluster load and make full use of the computing ability of the cluster, the CLE algorithm has a shorter job completion time than the RR algorithm. With the increase of the cluster computing nodes, the load on each computing node gradually reduces and the job completion time is significantly shorter. However, due to the load balancing in the CLE algorithm, the CLE algorithm still has a shorter job completion time than the RR.

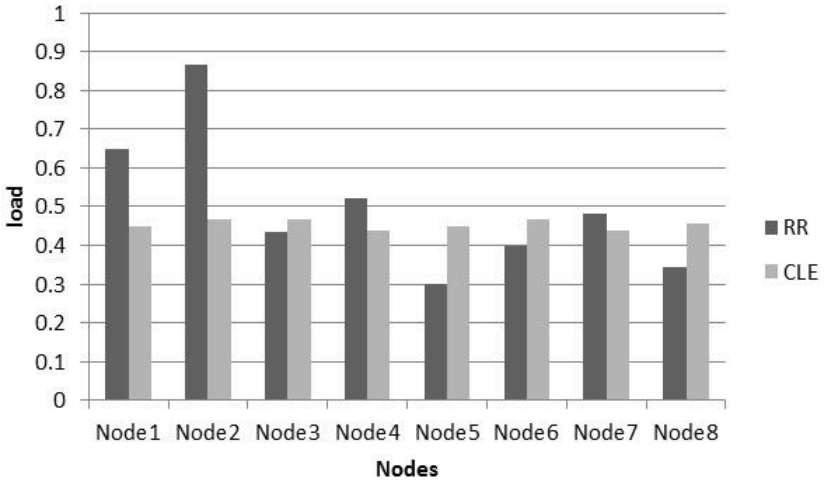


FIGURE 3. Load comparison

5.3. **Test 3.** In this test, we fix the scale of both the cluster and the number of jobs, and the computing capacity of each node is modified. The virtual machine size is set to 8 nodes, the number of jobs is fixed as 100, and the Mips is set as $\{200, 150, 300, 250, 400, 300, 250, 350\}$. The load condition of both algorithms is tested and results are shown in Figure 3.

It can be noticed in Figure 3 that under the condition of 8 nodes and 100 jobs, each node in the RR algorithm is not balanced because of its scheduling principle of sequential distribution, while the proposed CLE algorithm fully considers the computing capacity of each node and makes each node load balanced. This is also the main reason that the job completion time of the CLE is shorter in Test 1 and Test 2.

Through the above three tests, it can be fully proved that the proposed CLE algorithm has a good ability of load balancing in the scheduling process of the task computing.

6. **Conclusions.** Regarding the problems of low system resource utilization and long job response time resulted from the load imbalance in the distributed cluster system, based on the theory of entropy, this paper proposes the concept of computational load entropy and presents the load balancing algorithm according to the computational load entropy. The proposed algorithm has achieved the quantitative regulation of load balancing in distributed systems by means of the system load determination, single node determination, and load regulation. Through the comparison tests with the RR algorithm, the proposed algorithm is verified for its good balancing effect on the load of the computed tasks in distributed systems. Results also show that the proposed algorithm can effectively control the load imbalance of the cluster and improve the resource utilization of the cluster system. The current focus of this research is on the balance of the computing resources and the future research could be carried out from the perspective of integrating the computing resources, storage resources, and communication resources.

**Acknowledgment.** This work is partially supported by the Sci. & Tec. Support Plan of Sichuan Province under the grant No. 2014GZ0013 and the Sci. & Tec. Innovation Talent Project of Sichuan Province under the grant No. 2014-063. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

## REFERENCES

[1] W. Yao, H. Pan and C. Gao, A method based on the combination of dynamic and static load balancing strategy in distributed rendering systems, *Journal of Computational Information Systems*, vol.10, no.4, pp.1759-1766, 2014.

[2] F. F. Camacho, N. U. Lugo and H. C. Martínez, The concept of entropy, from its origins to teachers, *Revista Mexicana De Fisica*, vol.61, no.2, pp.69-80, 2015.

[3] J. Y. Dong et al., The entropy value of load balance algorithm about cloud computer cluster, *Journal of Chengdu University of Information Technology*, vol.25, no.6, pp.580-583, 2010.

[4] L. Y. Zuo, Z. B. Cao and S. B. Dong, Virtual resource evaluation model based on entropy optimized and dynamic weighted in cloud computing, *Journal of Software*, vol.24, no.8, pp.1937-1946, 2013.

[5] H. Y. Sun et al., A load balancing algorithm based on parallel computing entropy in HPC, *Journal of Shenzhen University Science and Engineering*, vol.1, pp.64-68, 2007.

[6] X. T. Cheng et al., Entransy balance for the closed system undergoing thermodynamic processes, *International Journal of Heat & Mass Transfer*, vol.60, no.1, pp.180-187, 2013.

[7] D. V. Alexis, The expectation value of the entropy of a digital message, *Open Systems & Information Dynamics*, vol.9, no.2, pp.97-113, 2002.

[8] R. N. Calheiros et al., CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Software Practice & Experience*, vol.41, no.1, pp.23-50, 2011.