Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2020.010407
ISSN 2790-0916 Vol. 1, Issue 4: 49-56

Resource Allocation of Distributed System Based on Self-organizing Clustering Algorithm

KhasheiSiuki Abbas*

KEMRI Wellcome Trust Res Programme, Kilifi, Kenya

**corresponding author*

Keywords: Self Organization, Clustering Algorithm, Distributed System, Resource Allocation

Abstract: With the development of network physical system, the distributed resource allocation strategy combined with physical system has been paid more and more attention, and many valuable research results have been obtained. This paper mainly studies the resource allocation of distributed system based on self-organizing clustering algorithm. This paper first introduces K-means clustering algorithm, and then proposes an intelligent scheduling algorithm based on K-means clustering algorithm. Through the simulation experiment, we can know that the resource allocation algorithm proposed in this paper is effective.

1. Introduction

With the rapid development of digital system, communication network and sensor technology, network system has become an important system in production and life, and has been applied to many fields, such as sensor network, smart grid and robot network. The solution of optimization problems in network systems has always been a key research topic in the field of network control [1-2]. Recently, in order to solve the optimization problems in large-scale network systems, the distributed optimization method based on multi-agent system design has attracted many scholars' interest. This paper focuses on a kind of important optimization problem in network system resource allocation problem. The goal of resource allocation problem is to find a strategy to minimize the sum of all individual cost functions, and the obtained strategy needs to meet the global constraints of the entire network and the individual capacity constraints. At present, there are many algorithms for solving resource allocation problems. One common feature of these algorithms is that a control center is required to coordinate the calculation of the entire network. Therefore, such algorithms are generally called centralized algorithms [3-4]. These algorithms first let the control center collect the data of all individuals, then find the optimal decision-making strategy, and finally send the optimal strategy to the individuals in the network. As more and more individuals are included in the network, the scale of optimization problem becomes larger and larger, which makes the control center need to undertake huge computational tasks. In addition, the equipment failure, external disturbance, time delay and other factors in the system make the information data transmitted by the individual to the control center inaccurate, which will have a great impact on the calculated results [5]. In order to overcome these weaknesses of centralized algorithms, it is necessary to study new algorithms for solving resource allocation problems in networks.

Resource allocation is a basic problem in many fields such as social science and engineering practice. Currently, there are three main research directions [6]. One is a completely centralized resource allocation strategy. In this strategy, there is a "central processor" to collect the information of the entire network system, including the cost function of a single individual and the entire network, the performance constraints of each individual, the supply and demand of the entire network, and make a unified decision based on the collected information, and then distribute the decision results to each individual to complete the entire resource allocation process. The second is a decentralized control strategy based on the master-slave structure. There is also a master node in this strategy. However, unlike the fully centralized control strategy, the slave nodes in this strategy also have certain storage and computing capabilities. The slave nodes can perform simple local calculations using the local information they can obtain, but still need to feed back the calculation results to the master node, Then the master node makes the resource allocation decision and sends the decision result to the slave node, so as to complete the whole resource allocation process [8-9]. Scholars have proposed a third research method based on the idea of distribution, that is, distributed optimal resource allocation strategy. In the distributed resource allocation problem, the self-agent has independent decision-making ability and a variety of local cost functions, and can realize the optimal resource allocation of the entire network through mutual communication with the neighbor self-agent [10].

Compared with the classical distributed consistency problem, the purpose of distributed resource allocation problem is no longer to drive all agents to achieve the consistency of decision. Instead, considering the resource constraints of each agent, multiple agents work together to solve an optimal decision, thus minimizing the cost function of the entire network. Therefore, designing an appropriate distributed resource allocation strategy is the key to accomplish this task.

2. Resource Allocation and Scheduling of Distributed System based on Clustering Algorithm

2.1. K-means Clustering Algorithm

K-means algorithm is the most typical algorithm in the partition method, and it is also the most used algorithm today. It makes the elements in the data set move between different class clusters through multiple iterations until they reach the appropriate class cluster [11-12]. K in the k-means algorithm is the number of class clusters formed by the target. Its size needs to be defined by the user. During the calculation of the algorithm, the data similarity within the class cluster and the data similarity between the class clusters will be continuously calculated. The calculation method of the similarity is the average value of all data in the class cluster [13]. The execution process of K-means algorithm is as follows.

1) K data points are randomly selected from the original data set as the initial clustering center.

2) Traverse all data objects in the data set, and classify each data object to the nearest one according to Euclidean distance

The central point of is located in the class cluster. After the process is completed, each data object in the set has its own class cluster.

3) Traverse all clusters, calculate the average value of data in each cluster, and take it as the new cluster center.

4) Go back to step (2) and continue to execute. When the new center point does not change or the change range is small, the whole cluster

The process ends.

In this process, an objective function is usually defined, which ends when the objective function is minimized in the iteration process. The objective function usually adopts the square error criterion function, and its formula is shown in (1).

$$E = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \| x_i^{(j)} - m_j \| \quad (1)$$

Where, $x^{(j)}_j$ represents the ith data in the jth class cluster, NJ represents the number of data objects in the class cluster CJ, and MJ represents the center point of the class cluster CJ. The k-means algorithm is applicable to clustering of data sets with compact distribution and obvious distinction between clusters. The k-means algorithm is oriented to numerical feature types and can only be used when the average value of clusters can be calculated. This limits its scope of application. However, the data we face today are still mostly numerical data, A few character type or other non numeric data features are usually converted into numeric type by means of one hot and stringindex [14]. The k-means algorithm requires the user to specify the K value and determine K initial clustering centers in advance. When facing an unknown data set, the K value cannot be accurately defined. Because noise and outliers have a great impact on the k-means algorithm, the method of randomly determining the K value and K initial clustering centers will have the possibility that the results are locally optimal [15].

2.2. Intelligent Scheduling Algorithm Based on K-means Clustering Algorithm

(1) Distributed computing framework

As an important component of Hadoop, MapReduce is a computing framework that provides developers with distributed programming. The MapReduce framework can divide the operation of large batch data into several small tasks and distribute them to each node in the cluster for execution. Finally, the output results of each node are summarized to obtain the final results [16]. The MapReduce framework abstracts the complex calculation process into two functions: map and reduce. The output of map function will be used as the input of reduce function. The idea of divide and conquer is adopted to decompose large tasks into multiple small tasks, and the results of all small tasks are collected to obtain the final results [17-18].

The parallel computing model based on map and reduce is shown in Figure 1.

(2) Algorithm design

CAA algorithm is an idea and method to select an appropriate model to guide the task scheduling of new tasks according to the K-means clustering model and the similarity between the newly submitted jobs and the historical data. In general, it combines the characteristics of the distributed computing framework, that is, log data, to train an appropriate clustering model for the historical data. For any distributed cluster, it will continuously process the jobs submitted by users every day, When we only focus on the tasks in the submitted jobs and their required computing resources, as well as the available computing nodes and their available computing resources in the current cluster, the jobs submitted at any time will have a high similarity with the historical data in the cluster. After the model training is successful, we will consider the resource availability matrix and the task resource demand matrix in the current cluster for the new tasks submitted to find a suitable model to schedule their tasks.

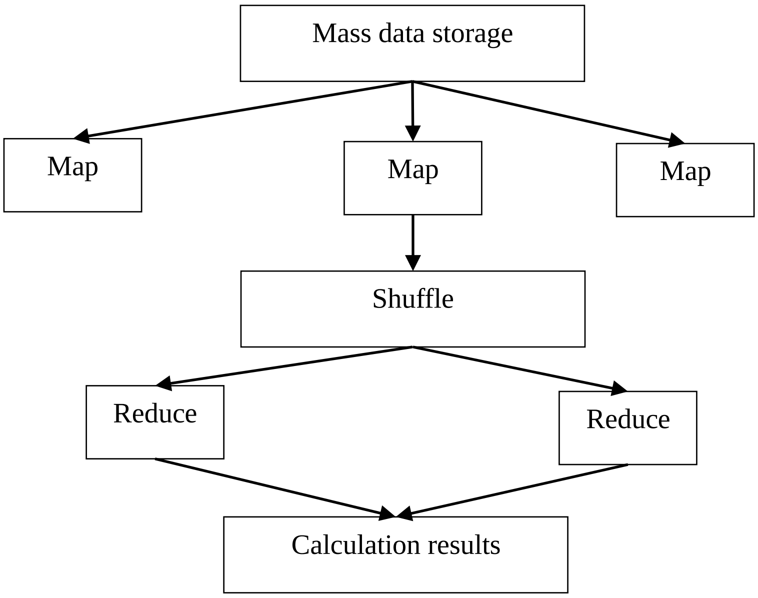

Figure 1. Parallel computing model based on map and reduce

According to the log files of the resource management modules such as HDFS and yarn, collect the nature of the processed tasks, data block access, completion time and other information, build the corresponding resource requirement matrix requirements, and the historical resource information of the computing nodes used to process these tasks at that time, and build the resource availability matrix resources, The improved hkmbm clustering algorithm and data Agnes algorithm are applied to cluster the resource demand matrix and the resource availability matrix. The cluster number k is simulated by traversal. After the optimal K value is selected according to the algorithm, the combination relationship between the cluster property and K is determined to prepare appropriate reference information for subsequent guidance of new work;

After a user submits a new job, information is collected for different distributed computing frameworks. In particular, in Hadoop cluster, there are different monitoring processes under different operating systems. Then, the monitoring process is deployed on the management node to read the current available computing resources of the node in real time, such as the availability of the central processor, the utilization of memory and ram, and the number of cores available to the central processor.

For example, in the Linux system, the above available computing resource information of the current node can be calculated according to the monitoring of the information in the system files in the Linux. The monitoring process is deployed on each work node to write the real-time information of the node into a specific report information file. Every time heartbeat establishes communication between the work node and the management node, the real-time information of the node is read from the file, and the information is sent to the management node together to build the cluster resource availability matrix resourcenew and task resource requirement matrix requiresnew at this time;

Use Euclidean distance to calculate the similarity between the new matrix and the matrix in the existing reference information:

$$Similarity = \frac{1}{Dist(\mathrm{Re}\,quires_{new}, \mathrm{Re}\,quires) + Dist(\mathrm{Re}\,sources_{new}, \mathrm{Re}\,sources)} \quad (2)$$

Apply algorithm 1 or 3 to perform simple clustering with reference to the K value in the backup information with the largest degree of similarity. Similarly, according to the order of Euclidean distance, a one-to-one mapping relationship is adopted for the task cluster and the computing node cluster after clustering;

After the mapping relationship is determined, the corresponding task is allocated to the designated computing nodes one-to-one according to the order of submission time by using the traditional FIFO mode on the premise of giving priority to data locality, and the task processing sequence t of each computing node is returned.

3. Algorithm simulation experiment

3.1. Experimental Platform

This paper uses the deep learning cluster of our high-performance computing center as our testing platform. The cluster has 6 GPU cards, including four node cluster and two node cluster. The configuration is shown in Table 1.

Table 1. Training job model settings

GPU model	VRAM	Quantity
Nvidia Tesla P40	24GB	2
Nvidia RTX2080TI	11GB	4

In this paper, NVIDIA teslap40 is used as our main test environment to obtain the interference matrix, memory usage curve and perform training tasks.

3.2. Comparison Method

Job set type: we use the simulator to automatically generate three types of job sets, and each type contains 100 DL jobs.

(1) Average: randomly select 100 jobs in the job;

(2) High collision c-high: operations with high collision probability of more than 50%;

(3) High interference i-high: operations with high interference value of more than 50%;

This paper uses two other algorithms to test three different workloads. Analyze and experiment the resource usage and the delay of job execution time respectively.

4. Experimental Result

4.1. Resource Usage

Table 2. Comparison of resource usage of three algorithms

	Algorithm 1	Algorithm 2	CAA
average	43	48	53
C-HIGH	50	49	59
I-HIGH	41	58	56

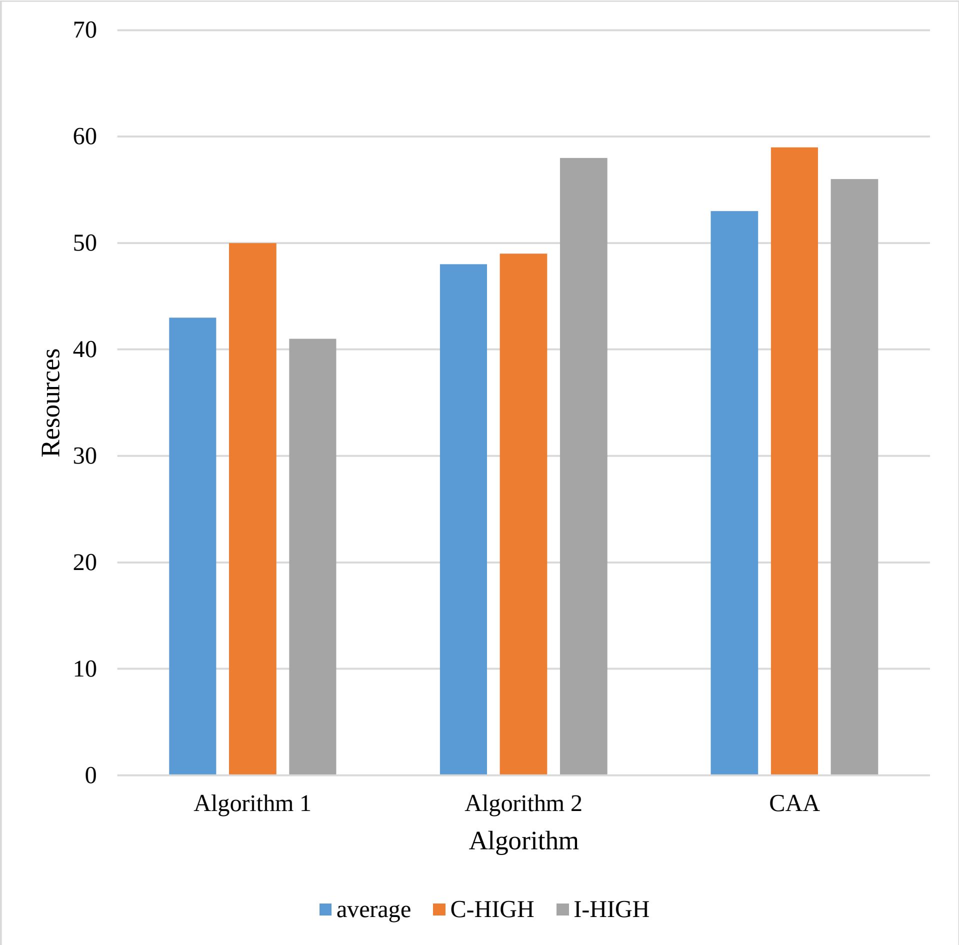

Figure 2. Static resource usage

As shown in Table 2 and Fig. 2, the analysis of the resource utilization results of the comparison algorithm is as follows: compared with the basic algorithm that only considers the remaining resources of GPU, the remaining three algorithms proposed in this paper all increase the use of equipment resources. It is not difficult to see that the number of resources used when i-greedy and c-greedy algorithms are respectively used for high interference job sets and high collision probability job sets. In comparison, the CAA algorithm has a good improvement on the numerical analysis of the resource usage of these three job sets.

4.2. Operation Execution Time

As shown in Fig. 3, in all cases, the average cooperative parallel job execution time is longer than the independent execution time. All three workload types perform well. The reason why the heuristic algorithm has been significantly improved is that we finally avoided the most difficult jobs to arrange, thus reducing the jobs with the most serious performance degradation. Therefore, the average performance degradation of the overall job set is kept high, and the performance is improved regardless of the job set.

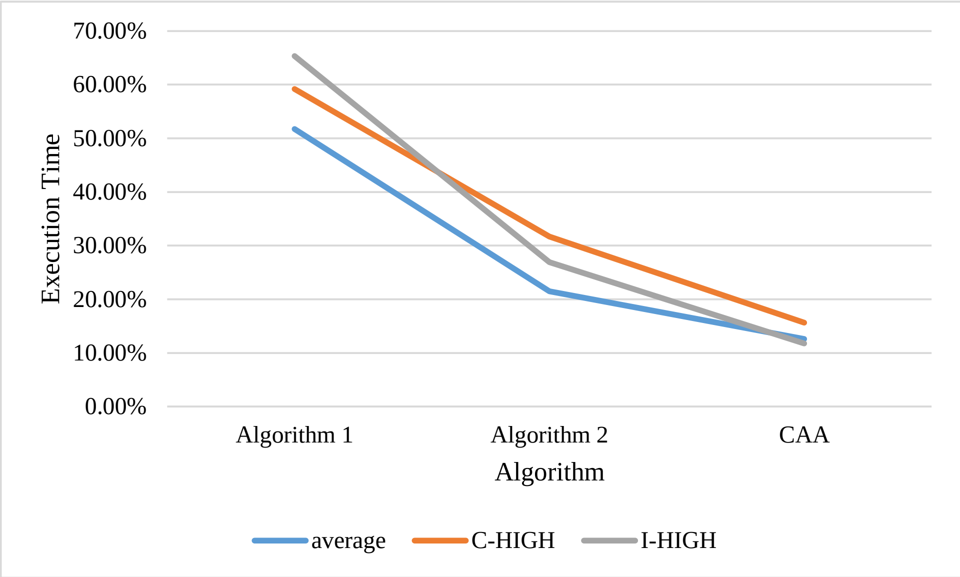

Figure 3. Deep learning job performance

5. Conclusion

With the development of information technology, sensor technology and control theory, the scale of resource allocation problem in the network becomes larger and larger, which makes the traditional centralized scheduling method may not be feasible to solve the resource allocation problem in the future network. In addition, the centralized algorithm requires the control center to collect the parameters of all devices and then process them. This information interaction mode is easy to cause the leakage of device privacy data. In order to meet these new challenges, this paper studies the distributed resource allocation in distributed systems. The distributed resource scheduling based on clustering algorithm proposed in this paper realizes static resource allocation and dynamic resource load adjustment strategies, and achieves good results.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Kim C , Kim H , Jung H . IoT distributed processing system based on resource allocation algorithm. International Journal of Applied Engineering Research, 2017, 12(24):15050-15054.

[2] Sangaiah, Hosseinabadi, Shareh, et al. IoT Resource Allocation and Optimization Based on Heuristic Algorithm. Sensors, 2020, 20(2):539.
[3] Sharma S , Singh B . Overlapping Coalition-Based Resource and Power Allocation for Enhanced Performance of Underlaying D2D Communication. Arabian Journal for Science and Engineering, 2019, 44(3):2379-2388.
[4] Marden, Jason. The Role of Information in Distributed Resource Allocation. IEEE Transactions on Control of Network Systems, 2017, PP(3):1-1.
[5] Nguyen D T , Long B L , Bhargava V K . A Market-Based Framework for Multi-Resource Allocation in Fog Computing. IEEE/ACM Transactions on Networking, 2019, PP(99):1-14.
[6] Gs A , Rui W B , Xfw A , et al. Distributed algorithm for resource allocation problems under persistent attacks - ScienceDirect. Journal of the Franklin Institute, 2020, 357(10):6241-6256.
[7] Halabian H . Distributed Resource Allocation Optimization in 5G Virtualized Networks. IEEE Journal on Selected Areas in Communications, 2019, PP(3):1-1.
[8] Afrin M , Jin J , Rahman A , et al. Multi-objective resource allocation for Edge Cloud based robotic workflow in smart factory. Future Generation Computer Systems, 2019, 97(AUG.):119-130.
[9] Kaur S , Ghumman M . A Review On Dynamic Resource Allocation Based On Lease Types In Cloud Environment. International Journal Of Computers & Technology, 2017, 16(1):7581-7585.
[10] Ahmadi B , Ceylan O , Ozdemir A . Distributed energy resource allocation using multi-objective grasshopper optimization algorithm. Electric Power Systems Research, 2020, 201(3):107564.
[11] Panadero J , Armas J D , Serra X , et al. Multi criteria biased randomized method for resource allocation in distributed systems: Application in a volunteer computing system. Future Generation Computer Systems, 2017, 82(MAY):29-40.
[12] Saghiri A M , Meybodi M R . A distributed adaptive landmark clustering algorithm based on mOverlay and learning automata for topology mismatch problem in unstructured peer‐to‐peer networks. International Journal of Communication Systems, 2017, 30(3):e2977.1-e2977.22.
[13] Saihanqiqige. Application Research of English Scoring Based on TF-IDF Clustering Algorithm. IOP Conference Series: Materials Science and Engineering, 2020, 750(1):012215 (6pp).
[14] Kim W , Yoon W . Joint beam forming and resource allocation for WPCN with distributed massive MIMO system. International Journal of Applied Engineering Research, 2017, 12(20):10073-10076.
[15] Lim N , Majumdar S , Ashwood-Smith P . MRCP-RM: A Technique for Resource Allocation and Scheduling of MapReduce Jobs with Deadlines. IEEE Transactions on Parallel & Distributed Systems, 2017, PP(5):1-1.
[16] Sangaiah, Hosseinabadi, Shareh, et al. IoT Resource Allocation and Optimization Based on Heuristic Algorithm. Sensors, 2020, 20(2):539.
[17] Tim D . A novel approach on the provision of distributed energy resource capabilities. it - Information Technology, 2017, 59(1):5-11.
[18] Ahmadi B , Ceylan O , Ozdemir A . Distributed energy resource allocation using multi-objective grasshopper optimization algorithm. Electric Power Systems Research, 2020, 201(3):107564.