A dynamic resource chain task unloading method based on improved greedy algorithm

Hu Bo¹,a, Tian Xiaolei²,b, Yang Chao³,c, Jiang Wei⁴,d, and Dong Shi Dan Jie⁵,e
Chen Minghui⁶,f, Chen Wu⁷,g, Chen Shu⁸,h, and Weng Zhangjun⁹,i

¹ State Grid Liaoning Electric Power Co., Ltd, Shenyang, Liaoning, China
² State Grid Liaoning Electric Power Co., Ltd, Shenyang, Liaoning, China
³ Information communication branch of State Grid Liaoning Electric Power Co., Ltd, Shenyang, Liaoning, China
⁴ State Grid Corporation of China, Beijing, China
⁵ State Grid Information and Communication Industry Group Co., Ltd, Shenyang, Liaoning, China
⁶ State Grid Xintong Yili Technology Co., Ltd, Shenyang, Liaoning, China
⁷ State Grid Xintong Yili Technology Co., Ltd, Shenyang, Liaoning, China
⁸ State Grid Xintong Yili Technology Co., Ltd, Shenyang, Liaoning, China
⁹ State Grid Xintong Yili Technology Co., Ltd, Shenyang, Liaoning, China

aemail: hubo@ln.sgcc.com.cn, bemail: tianxl@ln.sgcc.com,
cemail: yangchaoneu@sina.com, demail: wei-jiang@sgcc.com.cn,
eemail: dongshidanjie@sgitg.sgcc.com.cn, femail: chenminghui@sgitg.sgcc.com.cn,
gemail: chenwu@sgitg.sgcc.com.cn, hemail: chenshu1@sgitg.sgcc.com.cn,
iemail: wengzhangjun@sgitg.sgcc.com.cn

Abstract. The computing network based on cloud computing is becoming more and more complex. How to effectively unload the computing tasks is the key problem to improve the service ability of the computing network. In this paper, a mathematical model of task unloading problem in computing network is established, and an optimal solution based on improved greedy algorithm is proposed, which can quickly find the way of resource chain task dynamic unloading and realize task fast allocation and calculation. In this paper, the search space for greedy algorithms is too large and there is a lack of support for HPC LAN. Therefore, the greedy algorithm search is improved from one layer to two layers, which are respectively searched by the strongly connected graph and strongly connected graph of the virtual graph, which can adapt to the computing network with complex structure, and has friendly characteristics for high-performance computing LAN.
1. Introduction

Internet of things (IOT) is an important development of computer field in the future. Various industrial networks, such as telecommunication network and power network, have gradually evolved into a multi-layer IOT structure based on IOT architecture, which is "perception layer + network layer + platform layer + application layer". The computing network based on Internet can be divided into infrastructure as a service (IAAs), platform as a service (PAAS) and software as a service (SaaS). For resource management, it can be be divided into storage resources, computing resources and network bandwidth resources.[4]

Facing the new Internet of things architecture, how to provide platform services to users is the function. On the basis of realizing the management and control of terminal intelligent devices, IOT system also provides terminal devices with edge computing capability based on computing network, so that edge terminals with ordinary performance can use powerful IOT system to realize network computing function.

By providing users with virtualized, standardized and choreographed computing resources, the Internet of things architecture transforms the network bandwidth, storage space and computing power in the whole network into resource sharing pool, and dynamically unloads the computing tasks of edge nodes to the resource chain of the network through an effective end-user resource chain task dynamic unloading method, so as to help every user. The terminal not only realizes powerful network computing, but also realizes the effective management and scheduling of the whole network resources.

Edge node computing tasks usually need to apply to the computing network for a series of resources to meet its computing requirements. Through the arrangement of computing network, these resources form a resource chain and are provided to the edge nodes. Generally, a computing network can be abstracted as a directed connectivity graph. The resource discovery and allocation problem in a directed connectivity graph is a typical high latitude NP problem.[1]

In [2], a multi-user oriented serial task dynamic unloading strategy MSTDOS is proposed to solve the problem of task unloading under resource constraints of edge server. The strategy takes application completion time and terminal energy consumption as evaluation indexes, and uses the first come first service principle to make an approximately optimal unloading decision.

In paper [3], the unified allocation of bandwidth resources and computing resources in task offloading is comprehensively considered, and a model to minimize the total delay of task execution is established with the unified constraint of network resource chain. A heuristic algorithm based on greedy strategy is designed to solve the resource allocation problem. In particular, this paper also considers the load balancing problem of multi cell computing, solves the problem that the computing capacity of hot area does not match the user's computing needs, and optimizes the task unloading of the whole network.

In order to solve the problem of limited resources of mobile terminal, this paper proposes a task unloading and resource allocation algorithm based on edge end cooperation. With the goal of maximizing the total benefit of task completion, a fast task unloading algorithm is realized by using dynamic programming algorithm.

This paper proposes a dynamic resource chain task unloading method based on improved greedy algorithm. According to the characteristics of the industry Internet of things, it reduces the search space of greedy algorithm, and improves the search for the high-performance computing regional network which is usually configured in the industry Internet of things. By introducing the virtual node search and the user-defined search algorithm in the regional network, it can well adapt to the structure Complex computing network is a simple, efficient and flexible task unloading algorithm.

2. Resource chain task unloading model

Edge computing makes the Internet of things computing is no longer completed by a single machine, but by the whole computing network. In the Internet of things, a variety of connected terminals will apply for different computing tasks. For example, the face recognition terminal will apply for graphics and image recognition tasks; the bill terminal will apply for data processing tasks, etc.
In the Internet of things computing network, different tasks are completed by different computing nodes. A task can also be divided into several subtasks, and different subtasks are completed by different computing nodes. Sufficient resources must be allocated for the completion of subtasks. If the front and back connected subtasks execute the required resources, the chain is called resource chain. Subtasks can be migrated between different computing nodes. The overall completion time of a task depends on the calculation time of its subtasks and the migration time of its subtasks between the calculation nodes.

The emergence of the computing network based on the Internet of things makes the data collected from the edge of the computing network, and the computing network completes complex computing functions, which greatly reduces the performance requirements of the edge nodes, and makes it possible for a large number of nodes to access to the Internet of things and transform into powerful intelligent computing nodes.[6-8]

In the computing network, the user's computing task needs must be efficiently unloaded to the computing network, in order to complete the user's computing needs. The general requirements of task offloading include: (1) offloading tasks to the computing network as quickly as possible.(2) After the resource offloads the task to the computing network, it can meet the user's computing requirements as quickly as possible.

The task unloading model of computing network can be constructed by using directed graph. Suppose a directed graph, which represents each computing node in the computing network, and an arc represents the cost of task execution migration between nodes. Suppose there is a computing node in the computing network, a task can be divided into independent sub tasks. There is a context between subtasks, that is to say, subtasks must be completed before they can be carried out. Suppose that the processing time of a given task on a standard computing node in a computing network is. Compared with the standard node, the computing power of other non-standard nodes is expressed by computing power coefficient. Therefore, the processing time of a subtask on a computing node with computing power coefficient is.

Therefore, the model of task unloading in computing network is as follows

Constraint condition: where, (1)
Objective function: (2)

In the above formula, it represents whether the calculation node is selected to perform tasks, and its values are 0 and 1. 0 represents not selected, and 1 represents selected. Represents the time that subtasks migrate from node to node, that is, the arc length in the graph.

Therefore, the task unloading problem is to find a solution vector which satisfies the above constraints and maximizes the objective function. The solution vector represents the unloading path of the task, and the resources on the unloading path node are also the resource chain needed to complete the task.

3. Task unloading based on improved greedy algorithm

The basic idea of greedy algorithm is to get the solution of a problem through a series of choices. Greedy algorithm is the most greedy choice in the current state. Greedy algorithm starts from the initial solution, uses the method of constructing the optimal solution step by step to complete all the tasks unloading, and each step produces a component of tuple solution vector. In the process of selecting the solution components, adding new components, the resulting partial solutions do not violate the constraint conditions of feasible solutions.[9,10]

```c
Algorithm: findunloadpath (g, n)
/*
The algorithm finds a feasible task unloading path in the directed graph G;
G is a directed weighted graph describing the computing network;
N is the number of nodes in G.
*/
Begin
```
UnloadPath={};
While (not solution (unloadpath)) // task unload path not found
{
  =Select (g); // make greedy selection in input graph G
  If (feasible (unloadpath,)) // judge whether it is feasible to add a node to the unload
  path
  Unloadpath = unloadpath +; // add the component to the task unload path
  ;
}
Return unloaded path; // returns the task unload path
End

The greedy strategy in this algorithm finds all the connecting nodes, makes the computing resources of
the nodes match and larger than the computing resources needed by the subtasks, and makes the arc
shortest. Two improvements are made in this algorithm: (1) the computation time of subtasks at nodes
is converted to the arc length of migration, which can simplify the use of greedy algorithm and avoid
the use of complex transformation. (2) In the computing network, in order to avoid the fragmentation of
calculating node resources, the greedy policy setting needs to be improved. Suppose that the standard
calculating resource is, the allocated computing resource must be, where.

In computing networks, there are usually computing area networks with higher network connectivity.
Generally, the characteristic of strong computing area network is that any two nodes can be the same
and have high bandwidth. This way is similar to the strongly connected graph in graph. A strongly
connected graph is a directed graph G in which any two different vertices are reachable to each other.
In order to improve the efficiency of offloading tasks in the computing network, the strongly connected
graph can be dynamically reduced to a node. This node uses its own allocation method to schedule tasks.
Therefore, in the algorithm in the above table, it is modified that the graph is the new graph after the
strong connected subgraph of the original graph is transformed into a virtual node. It uses self-defined
scheduling algorithm to select new nodes in strongly connected subgraph.

The improved algorithm no longer searches the next node in the whole graph G greedily, but searches
the node in the new strong connectivity transformation graph first. If the strong connectivity subgraph
is found, the final node is found in the strong connectivity subgraph. The improved method in this paper
provides great flexibility for resource chain task dynamic unloading: (1) in the whole computing network,
if several computing nodes constitute a high-performance computing cluster, the algorithm can better
support task centralized unloading in this high-performance computing cluster. (2) High performance
cluster can use self-defined method to select computing nodes, which can effectively support various
scheduling algorithms in the cluster, and has no overall impact on the whole resource chain task dynamic
unloading algorithm. This method of two-level scheduling or two-level unloading enhances the
flexibility of dynamic unloading of resource chain tasks.

4. Algorithm application
As shown in Figure 1, it is a computing network with 20 nodes. The connection between nodes in the
whole network is shown in the figure. Now there are three tasks, waiting for task unloading. Then each
task will use the resource chain task dynamic unloading method based on improved greedy algorithm to
unload the task.
In specific applications, the computing power of each node in the whole computing network is expressed by the ratio of the computing power of each node to that of the standard node. The computing power of all nodes changes dynamically with the load of computing tasks. The dynamic computing power value is put on the scheduling nodes in the network. In the algorithm, \( \text{select}(g) \) can select the node that best meets the calculation requirements of the current subtask. In the application of the algorithm, we should consider the following situations: (1) if the bandwidth of the network has the characteristics of dynamic change, then we can refresh the weight of the whole graph dynamically. (2) For unknown computing tasks, if the end time is longer, the computing power of the node will be lower, so as to avoid assigning tasks again and improve the completion time of the whole task. 

5. Conclusion

With the development of cloud computing and Internet of things, edge computing based on computing network has become more and more popular. In this paper, a mathematical model of task unloading problem in computing network is established, and an optimal solution based on improved greedy algorithm is proposed, which can quickly find the way of resource chain task dynamic unloading and realize task fast allocation and calculation.

The method proposed in this paper is simple, efficient and flexible. It can be well adapted to complex computing networks and is friendly to high performance computing local area networks. For complex computing networks, such as the local area networks (LANs) with high bandwidth resources and computing power deployed in the computing network, it can automatically adapt to the internal resource allocation algorithm of the LAN without adjusting the task unloading algorithm of the whole network.

In addition, the method proposed in this paper has the advantages of simplicity, which is suitable for deployment to the edge nodes with weak computing power, so as to endow the edge nodes with strong network computing power.

Acknowledgement

State Grid Corporation of China headquarters science and technology project funding.

References

[1] Zhang Wenwen, Du Yongwen, Zhang Xiquan. Lightweight task offloading optimization for multi-user mobile edge computing [J]. Minicomputer systems, 2020, 41 (10): 2056-2062.

[2] Liu Wei, Huang Yu Cheng, Du Wei, et al. Resource constrained serial task offloading strategy in mobile edge computing [J]. Acta Sinica Sinica, 2020, v.31 (06): 309-328.
[3] Zhao Hongyu. Research on computing offload in resource constrained mobile edge computing system [D]. 2019.

[4] Zhang Shuhua, Tong Jie, Zhang Zhen, et al. Research on edge computing technology for smart sensing of energy Internet [J]. Power information and communication technology, 2020, 18 (4): 42-50DOI:10.16543/j.2095-641x.electric.power.ict.2020.04.005.

[5] Wu Liuqing, Zhu Xiaorong. Joint optimization algorithm for task offloading resource allocation based on edge end collaboration [J]. Telecom Science, 2020, (3). 42 ~ 52.

[6] Yao Hui Juan, Geng Liang. Next generation network architecture for computing network convergence[J]. Telecom Science, 2019, 35(9):38-43DOI:10. 11959/j.issn.1000-0801.2019212.

[7] Cao Chang, Zhang Shuai, Liu Ying, et al. Computing power network scheduling technology based on the cooperation of communication cloud and bearer network [J]. Telecom Science. 2020, (7). 55-62doi:10.11959/j.issn.1000-0801.2020197.

[8] Cai Yueping, Li Tianchi. Research on ubiquitous deterministic network for computing power matching scheduling [J]. Information and communication technology. 2020, (4). 9-15.

[9] Yang Wujun, Hao Kai. Cloud computing task scheduling based on greedy improved algorithm [J]. Sensors and Microsystems, 2016,35 (12): 143-145DOI:10.13873/J. 1000-9787 (2016) 12-0143-03.

[10] Wang Dewen, Liu Xiaomeng. Resource scheduling of cloud computing platform based on improved particle swarm optimization [J]. Computer application research, 2015, (11): 3230-32343246DOI:10.3969/j.issn.1001-3695.2015.11.006.