PAPER • **OPEN ACCESS**

Research on motif discovery algorithm in network based on MapReduce

To cite this article: Zheng Liu and Qian Zhang 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **490** 042026

IOP Conf. Series: Materials Science and Engineering **490** (2019) 042026 doi:10.1088/1757-899X/490/4/042026

Research on motif discovery algorithm in network based on MapReduce

Zheng Liu[1,*], Qian Zhang[2]

[1]College of Computer Science and Engineering, Northeastern University, China

[2]Advanced Product Division, Neusoft Corporation, China

*Corresponding author e-mail: liuzheng@mail.neu.edu.cn

Abstract. The motif discovery algorithms in network are always serial algorithms that run on single machine, which leads to lower efficiency and cannot meet the demands for discovering motifs in large-scale networks. In response to this situation, a parallel motif discovery algorithm is presented based on a parallel programming model called MapReduce in this paper. The algorithm is realized on Hadoop to discover the motifs in software networks with different scale. From the application test, the efficiency, speedup and expandability of the parallel algorithm are analysed and verified.

1. Introduction

In 2002, R.Milo et al. found some connected subgraphs appear much more frequently in real network than in random network and then called them as network motifs [1]. We can analyze and research on the structure feature and its forming mechanism of a complex system from its inner structure by network motifs. It will help the researchers understand the evolution mechanism of complex system structure in terms of local structure [2]. Recently, motifs are widely applied in the study of complex system and complex network [3], such as protein network, gene transcription network and neural network in life science area, social network and scientists network in social science area, etc.

After the theory of network motifs being presented, research on motifs attracted much attention from researchers of different research areas, and the motif discovery algorithm is always their research focus. It is a complex process that discover motifs in networks, including subgraphs statistics and subgraph isomorphism in random networks. Especially the process of dealing with subgraph isomorphism is NP problem [4]. Since R.Milo presented the motif discovery algorithm Mfinder based on edge expansion in 2002 [5], many motif discovery algorithms are presented and verified. But most of them are serial algorithms running on single machine, and the improvement of their efficiency is limit, while few parallel algorithm such as statistics parallel algorithm in single network presented by Wang et al. in 2005 [6] and Grochow algorithm presented by Schatz et al. in 2008 have limit scope of application that lead to poor practicability. According to the problems above, we present a parallel motif discovery algorithm based on MapReduce Model, which is a parallel programming model, and apply it to the analysis on software network structure. The application of this algorithm will lay a foundation for research on the relationship between micro structure features and macro structure features of software system in terms of motifs in the future.

2. Parallel programming model

MapReduce is a distributed parallel programming model presented by Google, and it is also a correlation realization of algorithm handling and generating very large scale data set. Using MapReduce model, it can realize automatic parallel processing and distributed computing on a large number of common PC by calling interface easily. The basic principle of MapReduce model is that user define two functions: Map and Reduce, among which, Map is used to map the input data, that means divide the data according to some rules, and Reduce is used to reduce the middle results, that means merge these data. The specific computing process including seven steps as following.

1) The input data is split into M data sections by MapReduce library firstly, and the size of data section can be assigned by user. Then create a large number of user program copies in cluster.
2) Among these program copies, there are two kinds of program: one master and several workers. Master is in charge of task allocation and management. According to heartbeats detection, master allocate the Map tasks and Reduce tasks to the worker that is idle.
3) The workers allocated with Map task load correlative data sections from cluster, parse out the key/value pair from these sections and pass the key/value pair to Map. Map function process and output the middle key/value pair, and then cache them into the worker's memory.
4) The key/value pairs cached are divided into R partitions by partition function and written into local disc periodically. The storage location is passed back to master, and master pass the location to Reduce worker.
5) After receiving the storage location, reduce worker use RPC to read the cache data from host disc of Map worker, then sort the key value and merge the data with same key value.
6) Reduce worker pass the key value of the middle data it processing and the correlative middle value set to the Reduce function defined by user. Reduce function process these data and append the output results to the output file of the partition it belongs to.
7) After all the Map tasks and Reduce tasks are completed, master wake up the user program, and the calling of MapReduce is returned. After all tasks completed, the results are output to R output files (each Reduce task generates an output file).

3. Motif discovery algorithm based on MapReduce

3.1 Workflow of motifs discovery algorithm

Generally, we don't know what kind of motifs are in the network we research on, so there are two steps during motif discovering. Firstly, search all subgraphs with size k in network as candidate for motifs, then find motifs from these subgraphs according to some decision conditions. Because there will be a large number of isomorphic subgraphs among all the size k subgraphs, we classify the subgraphs by isomorphism after building the standard label for each subgraph, and count the frequency of each subgraph appearing. The subgraph set classified by isomorphism is motif candidate set and this process is called statistics of subgraph in original network. After finding the motif candidate sets, we calculate the significance of each candidate subgraph in each candidate set according to its frequency. Next, do the same steps for several random networks generating by randomizing the original network. Finally we select the subgraph as network motif whose significance value meets the motif decision conditions.

3.2 Division of parallel tasks

ESU algorithm is a traditional subgraph discovery algorithm. Its basic idea is: expand from each node in network independently using recursive method to constitute a subgraph, and this process won't finish until the number of nodes in expanded subgraph reaches k [7, 8]. It is found that the main bottleneck during motif discovery is statistics of subgraph which occupy 95% of the whole testing time according to the analysis on traditional serial motif discovery algorithm. So it is very important to make the statistics tasks parallel for improve the efficiency of motif discovery.

The executing process of ESU algorithm can be described as a recursive tree in which the root represents the entrance of algorithm, the non-leaf nodes represent a recursion and leaf nodes represent all the subgraph found. It can be seen from the recursive tree that each subtree representing a recursion

process is independent and the nodes in it is unique in the whole tree. We can continue the following recursion from one node without its recursion information before. So the discovery tasks for subgraph can be paralleled.

If there are R random networks for subgraph discovery and statistics, there are R task units denoted by *U*. The task of discovering subgraph in each random network is set as a sub problem of motif discovery. The key problem of parallel motif discovery algorithm is dividing different task units into several task sets and executing in parallel. Here we divide the units by round-robin strategy. Set the number of task sets is M, then the No. of task unit divided into task set *i* is $i + j * M$. Here *i*=0,…,*M*, and *j*=0,…,*R*/*M*+1. Then load these task sets to parallel computing platform and realize the motif discovering in parallel.

3.3 Design of MRESU algorithm

3.3.1 Design of Map function. For Map function, the data it process is a task set and the units in set are its main process object. There are three steps in Map executing:

Step 1: get a task unit No. U_i in a task set, then generate a random network according to original network and name it as U_i. Delete No. U_i from task set.

Step 2: use ESU algorithm for motif discovering in this random network.

Step 3: classify the isomorphic subgraphs according to standard label, then count the frequency of the isomorphic subgraph appearing in the random network.

3.3.2 *Design of Reduce function.* Motif is a connected subgraph in real network which meet some special rules. These special rules including Z-score, P-value and appearing frequency are decision conditions for motif. P-value is used to measure the degree of divergence between metric samples and hypothesis with the value from 0 to 1. The P-value is smaller, the reason of hypothesis invalid is more compelling. P-value is also called significance level because it is believed the experimental result have statistics significance when P-value is small. Z-score indicates the significance degree of subgraph, that is the quantization of the higher frequency the subgraph appearing in real network than in a set of random networks, which is a necessary condition for deciding the subgraph as motif. Reduce function is used to compute Z-score and P-value of each kinds of subgraph by the output of Map function, and its specific steps are as follows:

Step 1: get the frequency of subgraph *i* appearing in original network;

Step 2: get the frequency of subgraph *i* appearing in each random network;

Step 3: compute the P-value of subgraph *i*;

Step 4: compute the Z-score of subpraph *i*.

4. Realization and analysis of algorithm

4.1 Distributed parallel computing platform-Hadoop

Hadoop is an open source distributed parallel computing platform based on MapReduce, and it is also an efficient, reliable and expandable software architecture described using master/slave [9, 10]. Here master is responsible for tasks delivering and scheduling and slave is responsible for tasks executing. A MapReduce job divides the input data into several independent data segments. When the tasks is starting, Map function loads input data in the form of key/value pairs and handles them, then output the middle results. MapReduce frame orders the middle results according to key value by Shuffle procedure and provides the ordered results to reduce function for further process.

The Hadoop cluster is composed with nine processors in this paper, including one master and eight slaves.

4.2 Analysis on algorithm applied in software network

There are simple structures composed with several connected classes or interfaces presenting obvious significance, which are called motifs in software network. Usually they are the network topology of design patterns or function components commonly used. We choose four open source software

systems and abstract their network topology, the number of nodes in which is from hundreds to tens of thousands. After serial motif discovery algorithm and MRESU algorithm being applied in these four software networks separately, we compare these two algorithms about their performance time and efficiency. It is shown in Fig.1 and Fig.2.

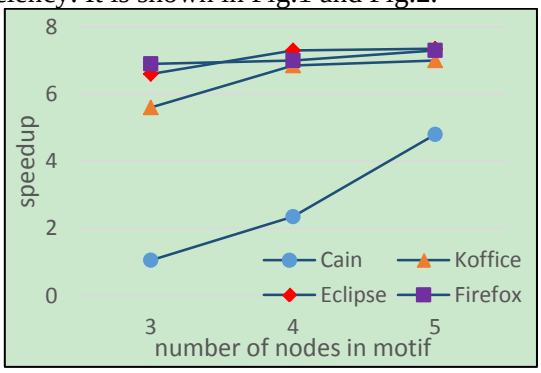

Figure.1 Speedup of MRESU algorithm

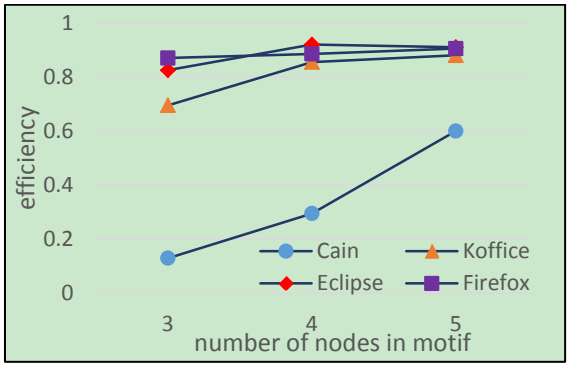

Figure.2 Efficiency of MRESU algorithm

The number of random network generating in the process of motif discovering R is set to 100 in the test, and the number of nodes k in motif is set to 3 then added to 5 generally. It can be seen that the performance speed and efficiency is improved greatly by using MRESU algorithm, especially when the scale of network is larger, the advantages of parallel algorithm are more obvious. Furthermore, we search motifs with the k value is 4 in Eclipse network and verify the expansibility of MRESU by adding the number of slave nodes in Hadoop cluster. The result is shown in Table 1.

Table 1. Experimental results of Hadoop of different sizes

	Single computer	m=2	m=3	m=4	m=6	m=8
Running time(min)	152.537	84.089	55.691	41.450	27.307	20.912
Speedup		1.814	2.739	3.680	5.586	7.294
Efficiency		0.907	0.913	0.920	0.931	0.912

It can be seen that along with the number of slave nodes being added, the performance time of motif discovery decline and the speedup increase obviously, which verify MRESU have good expansibility.

5. Conclusion

Researching on motif is one of the commonly used means in complex system and complex network area, of which motif discovery is the foundation of motif research. It is a complex process that discovering motifs in network. Many traditional motif discovery algorithms are serial algorithm based on single computer and the improvement of discovering efficiency is limit, while a few parallel algorithm can only be used in limited areas. So according to these problems, a parallel motif discovery algorithm base on MapReduce is presented in this paper. This algorithm is applied in motif discovery in software network and realized on Hadoop cluster. It is found the performance speed and efficiency of MRESU algorithm is improved greatly by analyzing and comparing with serial algorithm according to the experimental results. Especially, when the scale of network is larger, the advantage is more obviously. Meanwhile, MRESU has good expansibility and can be applied to different networks with different scale.

References

[1] Milo R, Shen-Orr R, Itzkovitz S, et al. Network Motifs: Simple Building Blocks of Complex Networks [J], Science, 2002, 298: 824-827.
[2] R.James Taylor, et al. Network motif analysis of a multi-mode genetic-interaction network [J], Genome Biology, 2007, 8:R160.
[3] Prill R J, Iglesias P A, Levchenko A. Dynamic Properties of Network Motifs Contribute to Biological Network Organization [J], PLos Biology, 2005, 3(11): 1881-1892.
[4] B. McKay. Practical graph isomorphism [J], Congressus Numerantium, 1981, 30:45-87.

[5] N.Kashtan, S.Itzkovitz, R.Milo. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs [J], bioinformatics, 2004, 1746-1758.
[6] Tie Wang, Jeffrey W. A parallel algorithm for extracting transcription regulatory network motifs [A], In Proceedings of the IEEE International Symposium on Bioinformatics and Bioengineering [C], 2005, 193-200.
[7] Sebastian Wernicke. A Faster Algorithm for Detecting Network Motifs [A], Proceedings of 5th WABI-05 [C], 2005, 165-177.
[8] Sebastian Wernicke. Efficient Detection of Network Motifs [J], IEEE/ACM transactions on computational biology and bioinformatics, 2006, 347-359.
[9] Tom White. Hadoop: The Definitive Guide [M], O'Reilly Media, 2012, 1-316.
[10] Weikuan Yu, et.al. Design and Evaluation of Network-Levitated Merge for Hadoop Acceleration [J], IEEE Transaction on Parallel and Distributed Systems, 2014, 602-611.