2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science

A parallel algorithm of path planning for DEM terrain data

Yanli Wang
School of Computer Science and Technology
Nanjing Normal University,
Nanjing, 210023, Jiangsu, China
e-mail: 2424461907@qq.com

Wanfeng Dou
School of Computer Science and Technology
Nanjing Normal University,
Nanjing, 210023, Jiangsu, China
e-mail:douwanfeng@njnu.edu.cn

***Abstract*—In this paper, a parallel algorithm of path *planning for DEM terrain data is proposed. First of* all, the raster grid DEM data are transformed into the network structure of graph theory based on the domain model in order to apply to the classical Dijkstra shortest path algorithm. Then, for the processing of large-scale network, a parallel algorithm of Dijkstra serial algorithm is implemented by MapReduce programming model. The experimental results show that our algorithm can archive a better efficiency in the execution time.**

***Keywords*- Dijstra;DEM;MapReuce;Parallelization**

I. INTRODUCTION

As a classic problem of graph theory, the shortest path problem has always been a research hotspot in operations research, computer science, terrain information science, transportation and other fields. Dijstra algorithm is a classical algorithm for solving single source shortest path problems. The development of modern mapping technology makes the delineation of geographical data more and more accurate, which makes the calculation time of the shortest path method based on three-dimensional terrain data, DEM data, is getting longer and longer. The parallel computing method has been used to solve the shortest path planning [1-2].

The shortest path on the terrain has two parallel types: data parallelization and task parallelization. Because of the high correlation with the internal of the Dijkstra algorithm, the data parallelization strategy is mainly adopted to improve the computation efficiency. MapReudce proposed by Google Corp. [3-4, 9], is adopted commonly as a parallel programming model of special structure for the parallel operation of massive data sets. Therefore, a parallel algorithm of shortest path on the DEM terrain data is design and implemented based on MapReduce mode. First this paper adopts the MapReduce to transform the three-dimensional DEM terrain data into the network from graph theory. Then, under the Hadoop cloud platform, the MapReduce programming model was used to design parallel Dijstra algorithm.

II. NETWORK TRANSFORMATION BASED ON RSG GRID DEM

A. The domain model of a regular grid

Definition: a network (weighting diagram) D can be defined as an even pair D= (V,E,W); Where, V is a set of non-empty vertices, and the fixed point number is N=|V|; E is the set of edges of V*V; W is the right of the side, which can represent the distance, time and cost between two points.

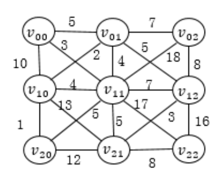
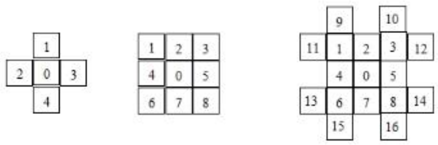


Figure1. an example of a network

There is an adjacency relationship in the rules grid DEM data, which can be used to form the relevant network (weighted graph) through domain patterns. According to the domain pattern, the node of grid DEM data is abstracted as the vertex in the network and the adjacent relation between the cell points is abstract as the arc in the network. The weight of each arc is calculated

978-1-5386-7445-1/18/$31.00 ©2018 IEEE
DOI 10.1109/DCABES.2018.00016

according to the adjacent relation. There are 4, 8 and 16 neighborhood modes, as shown in figure 2.



a) b) c)

Figure2. three neighborhood modes :a) 4 neighborhood mode; b) 8 neighborhood mode; c) 16 neighborhood mode.

B. Implementation of the conversion from RSG DEM grid to directed network

Generally speaking, the advantage of adopting an 8-neighborhood mode can not only reduce the complexity of problem and performance of computation but also guarantee the required accuracy. Therefore, we adopt the 8-neighborhood mode to realize the transformation between the RSG DEM to the network. The transformation process is given as follows.

- A grid cell in the RSG DEM data is abstracted into a vertex in the directed network, and the vertex set is S={v_0, v_1, ..., v_n}.
- Take the cell V_i in the set S according to the 8-neighborhood mode, the adjacency relation between V_i and its adjacent vertex is abstracted as an arc. If the adjacency vertex of the V_i exists, the weight value of the corresponding arc is calculated by means of the formula (3) and saved. If the adjacency vertex of the V_i does not exist, the weight value o is not preserved or set to ∞.
- let S=S - V_i, if the set S is empty, then end transformation procedure; Otherwise, return the step (2).

III. Terrain path shortest path algorithm

A. Dijkstra algorithm thought

This paper focuses on the shortest path problem of single source point of grid DEM data. By the network N and the starting point, the shortest path is solved from the starting point v to the network. Dijkstra algorithm [5] was proposed by Dijkstra in 1956. Dijkstra algorithm can find the shortest path between source points and nodes at the minimum cost of a graph with no negative weighted values [6]. The idea of serial Dijkstra algorithm is as follows.

- The nodes in the weighting diagram are divided into three types: unmarked nodes, temporary mark nodes and permanent marker nodes. First, the source point is initialized to a permanent node, and all other nodes are initialized to unmarked nodes.
- In the first search, the nodes adjacent to the permanent node are marked as temporary nodes.
- The node with the minimum weight value of the source point is searched from the temporary mark node list. Mark it as a permanent node and mark the node adjacent to the node as a temporary node and calculate its weight value.
- Until the linked list of the corresponding target node or temporary mark node is found to be empty, the algorithm ends.

The searching process of a shortest path according to Dijkstra (assuming that the source point is v_{00}) is shown in figure 3 when the network in figure 1 as an example.

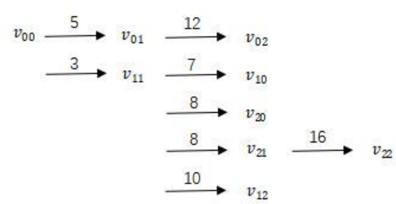


Figure3. the searching process of a shortest path

B. Breadth-First Search

Breadth First Search (BFS) [7-8] is one of the connected graph search strategy. For a vertex, first of all, the vertex is visited. Second, all non-access adjacency vertices from the vertex are found and accessed. And then, the adjacent nodes of these points are assessed in turn until all vertices are visited. Time complexity of BFS is

O(|V|+|E|).

Similarly, for the network in figure 1, the order in which the vertices are accessed according to breadth first is (assuming that the source point is v_{00} and the weighted value is 1) as follows.

$$v_{00} \rightarrow v_{10} \rightarrow v_{11} \rightarrow v_{01} \rightarrow v_{20} \rightarrow v_{21} \rightarrow v_{22} \rightarrow v_{12} \rightarrow v_{02}$$

IV. MAPREDUCE DESIGN AND IMPLEMENTATION OF DIJSTRA ALGORITHM

A. MapReduce Parallel programming model

Based on MapReduce, a complete graph structure is usually decomposed into several local substructures for parallel processing of each substructure. In the map phase, the adjacent points are generated <key,value> key value pairs, and after shuffle and sort, update node information in the reduce phase. The graph algorithm is usually an iterative process, which takes the output of the previous step as the next input, and is controlled by the extra driver.

In the shortest path algorithm based on network, Dijkstra algorithm is essentially a single-source serial algorithm. The core of the algorithm is to use the queue to save the global information weight value of the source point to other nodes. Each iteration is to find the node with the smallest weight value in the queue and update its value to the source point. For MapReduce, global sharing information is difficult to achieve. In this paper, the Dijstra algorithm based on breadth first search is proposed to solve the shortest path problem. The parallelization of Dijkstra algorithm based on MapReduce framework is as follows:

- In the map phase, read the input file in HDFS;If the value of the node distance is not MAX_VALUE, then the adjacency node of the node is traversed. the ID, weight, and ID of the adjacent node are output .the parent node and the vertex is set to the ID of the parent node of the adjacent contact.
- In the shuffle stage, the value part of the map result data with the same key value will be aggregated locally to reduce the network transmission and speed up the calculation.
- In the reduce stage, for the same ID, the weight value and the parent node ID should be updated if the weight is higher than the current weight value.
- If the distance of all nodes is not updated, the shortest path is gained. Otherwise, repeat the above step.

B. Data Storage

The data storage representation of the graph is generally two types: adjacency matrix and adjacency list. Among them, the adjacency matrix storage data is fast, but the storage space is large, so it is suitable for the storage of small data. For terrain data such a large amount of data, the adjacency list storage method is generally adopted. Using the MapReduce calculation model to implement Dijkstra algorithm to solve the shortest path, the adjacency list storage method used is shown in table 1.

TABLE I. THE ADJACENCY LIST STRUCTURE OF THE NETWORK

NodeID	**Distance**	**(AdjacentNode,Weight)**	**Parent Node**
v_{00}	0	$(v_{01},5),(v_{11},3),(v_{10},10)$	NULL
v_{01}	∞	$(v_{00},5),(v_{10},2),(v_{11},4),(v_{12}$	NULL
...	...	...	...
v_{21}	∞	$(v_{20},12),(v_{10},13),(v_{11},5),($	NULL
v_{22}	∞	$(v_{21},8),(v_{11},17),(v_{12},16)$	NULL

Among, NodeID is a node identification. Distance represents the distance weight value of the node to the source point, in which the distance from the source to the source point itself is 0. The tuple of (AdjacentNode, Weight) represents the adjacent node information, including the adjacent nodes of the node and the weight value of the path. ParentNode is the identity of the father node of the node.

C. Result Analysis

TABLE II. COMPARISON BETWEEN TWO ALGORITHMS ON A SMALLER DATASET

Node Number	Serial Dijkstra algorithm/s	Parallel Dijkstra algorithm /s
100	0.049	13.118
18204	0.864	14.216

The experimental environment is set up the processor with 3.20GHz and 4GB memory. In this paper, the serial Dijkstra algorithm is compared with the parallel Dijkstra algorithm in the pseudo-distributed Hadoop environment. The results show that the serial algorithm takes less time. The reason is that the parallel Dijkstra algorithm need to read and write repeatedly the data from the hard disk, but for the serial Dijkstra algorithm data is stored in the memory without too I/O time consumption with a smaller dataset. However, in the case of limited time, the serial Dijkstra will cause errors due to insufficient memory due to insufficient memory. The Dijkstra parallelization on MapReduce does not exist these problem.

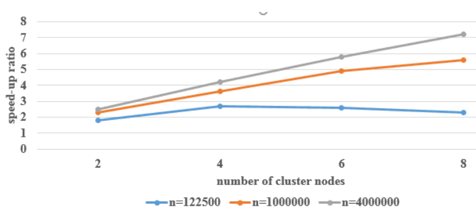


Figure4. the acceleration ratio

In the case of large data volume, the experimental cluster environment in this paper has 108 computing nodes. Each computing node has four processors each which is of 2.66GHz, 12GB Memory. In this paper, the comparison of the time acceleration ratio between different data volumes and different cluster nodes is carried out. It is found that in the case of large datasets the number of vertices is certain, and before the peak the acceleration ratio will become larger as the number of cluster nodes increases. But, after the peak, the acceleration ratio decreases as the number of cluster nodes increases. The reason is network throughput between cluster nodes. In the case of the number of nodes in the same cluster, the larger the data size, the acceleration ratio or the increasing trend. To sum up, the feasibility and superiority of Dijkstra parallelization algorithm are well represented in the case of mass graph data processing.

V. Summary

This paper mainly introduces the transformation of RSG grid DEM data through domain mode and realizes parallel data processing. In addition, according to the MapReduce programming model, the traditional serial Dijkstra algorithm is designed in parallel, and different datasets are tested on the Hadoop cluster to get the running time and efficiency. By comparing with the traditional serial Dijkstra, the performance of the parallel Dijkstra algorithm can achieve an evident improvement.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 41771411).

References

[1] Tremblay N, Florian M. Temporal shortest paths: Parallel computing implementations[J]. Parallel Computing, 2001, 27(12):1569-1609.

[2] Zhang D, Shou Y, Xu J. A mapreduce-based approach for shortest path problem in road networks[J]. Journal of Ambient Intelligence & Humanized Computing, 2018(4):1-9.

[3] Mmel R. Google's MapReduce programming model — Revisited[J]. Science of Computer Programming, 2008, 70(1):1-30.

[4] Atta F, Viglas S D, Niazi S. SAND Join — A skew handling join algorithm for Google's MapReduce framework[C]// Multitopic Conference. IEEE, 2011:170-175.

[5] Dijkstra E W. A Note on Two Problems in Connection with Graphs[J]. Numerische Mathematics, 1959, 1(1):269-271.

[6] Xu M H, Liu Y Q, Huang Q L, et al. An improved Dijkstra's shortest path algorithm for sparse network ☆[J]. Applied Mathematics & Computation, 2007, 185(1):247-254.

[7] Gazit H, Miller G L. An improved parallel algorithm that computes the BFS numbering of a directed graph [J]. Information Processing Letters, 1988, 28(2):61-65.

[8] Kurant M, Markopoulou A, Thiran P. On the bias of BFS (Breadth First Search)[C]// Teletraffic Congress. 2010:1-8.

[9] He B, Fang W, Luo Q, et al. Mars: A MapReduce Framework on graphics processors[C]// International Conference on Parallel Architectures and Compilation Techniques. IEEE, 2017:260-269.