Intelligent Transportation Control System Based on Cloud Computing

Lei Yu, Hongke Xu, Xin Hu

Department of Electrical Engineering, School Electronic and Control Engineering, Chang'an University, Xi'an, Shaanxi, 710075, China

Abstract
For the past few years, phenomena such as traffic congestion and traffic accidents in big cities have occurred one after another, greatly and adversely affecting the people's lives. Intelligent transportation system is perceived as an optimum solution to solve this question. This paper delves into several common route guidance algorithms, analyzes respective characteristics, improves and parallelizes ant colony algorithm on the basis of these characteristics, and designs and implements based on Map Reduce under Hadoop. Faced with specific application of transportation system and based on Map Reduce model and distributed data management technology, this paper delves into transportation network data management methods, analyzes transportation data by using VISSIM and designs the transportation network data structure. By abstractly establishing the transportation network data model catering to ant colony algorithm, the parallel ant colony system on the basis of cloud computing is implemented. This experiment has demonstrated good actual value the intelligent transportation system possesses.
Key Words: INTELLIGENT TRANSPORTATION, CLOUD COMPUTING, TRANSPORTATION NETWORK

1. Introduction

The target of intelligent transportation system is to create a more secure and effective transportation and reduce the influence on environment exerted by traffic congestion and pollution by utilizing the most advanced technologies. In transportation, computer, information, communication, System Science & Engineering and other circles, people intensively and deeply delve into theories and practices of intelligent transportation system at present, which has a quite bright development prospect. RGS [1] (Route Guidance System) is a major aspect of ITS. On the basis of the computer, electronic, communication, network and other modern technologies, it may enable each terminal device to automatically display the vehicle's current location, road traffic condition and transportation network map by adopting the electronic transportation map, global positioning system (GPS), computer and other advanced communication techniques. Besides, users may also inquire optimal path and real-time transportation information as per the required origin and destination for guidance. This system is featured by the consideration integrating people, vehicles and roads. By guiding the travel behaviors of users, it improves the traffic conditions of road network, prevents the occurrence of the traffic congestion, reduces the vehicle detention time and eventually achieves the reasonable distribution of traffic flow on each road section in the traffic network.

The labeling algorithm is quite important in numerous shortest path query algorithms. It can be resolved into label setting (LS) and label correcting (LC) [2] in accordance with different label node processing strategies. As first proposed by Dijkstra, a Dutch mathematician, in 1959, LC algorithm is a greedy algorithm with most comprehensive theory and widest application. And many subsequent LS algorithms are improved on the basis of Dijkstra algorithm or implemented in different ways, such as using heap structure, barrel structure and sorting connection table. LC algorithm adopts various heuristics principles represented by the Pallottino algorithm [3], Bellman-Ford-Moore algorithm, ant colony algorithm and topological sorting algorithm [4]. Some foreign scholars have deeply delved into and studied the design and implementation of the parallel in the shortest path algorithm. The methods to realize parallel of label setting LS algorithm and label correcting LC algorithm are thus gradually set up. At the same time, we also test and compare the performance and efficiency of these algorithms. At present, many parallel shortest path algorithms are designed for the corresponding serial algorithm with eminence in algorithms proposed by Tseng [5], R C Paige, C P Kruskal [6], Habbal [7] and etc. Kruskal and Paige proposed a parallel synchronized method to achieve the Dijkstra algorithm. However, they presented a train of thought to solve the problem only and no concrete implementation was given. Crauser put forward a theoretical method for the parallel of Dijkstra algorithm. He conducted the parallel implementation by several steps. Similarly, he also didn't test and actually applied it in any specific experiments, so the specific effect should be further examined. Narayanan delved deeply into the Floydal algorithm and presented two parallel implementation methods based on data. He compared these two methods, but neither compared them with serial algorithm nor gave out the relevant speed-up ratio. M.Randall [8], from Bond University Australia, raised the parallel ant solution when dealing with the TSP problem and tested performance of parallel ant solution in ant colony algorithm by selecting the TSPLIB problem.

Currently, people have carried out increasingly in-depth studies on cloud computing, more complicated traffic guidance system and greater transportation data, the cloud computing have been eventually and gradually applied in the transportation field. The parallel design implementations of the foregoing shortest path algorithm are all based on OpenMP or MPI, which have already not been suitable for applied requirements. For this reason, this paper delves into the transportation network path algorithm based on cloud computing, implements the parallel ant colony algorithm based on cloud computing and designs the intelligent transportation control system based on cloud computing.

2. Parallel Ant Colony Algorithm Based on Cloud Computing

When solving the TSP problem, the node of next city selected by the basic ant colony algorithm may meet the requirement unless it appeared in taboo list. The intersection nodes are connected by actual road sections and there's no loopback in route guidance, which means that users cannot encounter the same intersection node. However, if the ant colony transfer is only guided by the pheromone and heuristic information in basic ant colony algorithm, the loopback are easily formed in the path and larger non-linear coefficients in the path are selected, namely invalid path, which fails to perfectly meet the user requirements. To solve the questions stated above, this paper introduces the state transition rule containing directional function. A new heuristic factor acting as the heuristic factor of an ant may be created in accordance with the distance between a point and target point.

2.1 State Transition Rule

Assume that an ant is at the node of (x_0, y_0) and there are 4 points we can select in the next step and these 5 points are L_0 (original node), L_1, L_2, L_3, L_4 away from the target point, respectively. If we select one as an alternative node, the charge of distance away from the target point is $\Delta L_i = L_i - L_0$; the reference point charge of define distance is $H = \sum |\Delta L_n|$ and define direction function is L_r, For example, the direction function L_{ri} is:

$$L_{ri} = \frac{H - \Delta L_i}{\sum |H - \Delta L_n|} \quad (1)$$

Where n is the next node that can be selected. The direction function enables an ant have the global awareness capacity apart from the pheromone, makes the whole ant colony more intelligent and let them search out a global optimal solution in fewer reproductive generations. The selection strategy after introducing direction function:

$$P_i^k(t)=\begin{cases}\dfrac{\dfrac{\tau_{0i}^{a}(t)\eta_{0i}^{\beta}(t)}{\sum_{s\in allowedk}\tau_{0s}^{a}(t)\eta_{0i}^{\beta}(t)}+k0L_{ri}}{1+k_0} & i\in allowedk\\ 0 & \text{others}\end{cases}$$

(2) Where, k_0 is the weight coefficient of direction function in the transfer probability function and $k_0 \geq 0$ is an appropriate value selected by the test. We should find out an equilibrium point between global searching capability and heuristic factor to make the ant colony converge to the global optimal solution and prevent it converge to the local optimal solution.

2.2 Improve the Pheromone Updating Rule

The path selection is not determined by a single condition, but to comprehensively consider various influences, such as overall driving time, overall path length, entire expenses, extent of road congestion and the number of intersections running through. These conditions have a certain effect on the number of information content in paths passed. In consideration of multi objectives and attributes characteristics represented when selecting paths, and to be closer to the real environment, we add several limit constraints in the pheromone updating. The function f is defined and its expression is shown as below:

$$f=\omega_1 f_1+\omega_2 f_2+\omega_3 f_3+\omega_4 f_4+\omega_5 f_5 \quad (3)$$

f_1, f_2, f_3, f_4, f_5 refer to the entire expenses, overall driving time, road traffic capacity, road security and the number of intersections, respectively. The expressions are as follows:

$$f_1=\sum_{i\in V}\sum_{j\in V} mf_{ij}+k\sum_{i\in V}\sum_{j\in V} md_{ij}$$

$$f_2=\begin{cases}\sum_{i\in V}\sum_{j\in V} md_{ij}t_{ij} & \text{if } \sum_{i\in V}\sum_{j\in V} md_{ij}t_{ij}\leq T\\ 0 & \text{others}\end{cases}$$

$$f_3=\begin{cases}\sum_{i\in V}\sum_{j\in V} mr_{ij} & \text{if } \sum_{i\in V}\sum_{j\in V} mr_{ij}\leq R\\ 0 & \text{others}\end{cases}$$

$$f_4=\begin{cases}\sum_{i\in V}\sum_{j\in V} ms_{ij} & \text{if } \sum_{i\in V}\sum_{j\in V} ms_{ij}\leq S\\ 0 & \text{others}\end{cases}$$

$$f_5=\sum_{i\in V}\sum_{j\in V} mp_{ij} \quad (4)$$

$\omega_1,\omega_2,\omega_3,\omega_4,\omega_5$ are signed weight coefficients, V refers to a set of node and the change of set of node $\Delta\tau_{ij}^{k}$ is:

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{f} & \text{If the kth ant passes through the road section ij in this circuit}\\ 0 & \text{Otherwise}\end{cases} \quad (5)$$

3. Design of Intelligent Transportation Parallel Algorithm Based on Cloud Computing

There are some common parallel algorithm strategies are used currently, for instance, the resolve-and-rule, path doubling technique, tree contraction technique and iterative improvement technique. The resolve-and-rule is to resolve a large complex problem into p simple sub-problems approximately equal, to assign these sub-problems to different processors for solution and to combine these solving results by different processors to obtain the solution of the original problem. This method is not just to solve a single problem, but is a basic thought of parallel algorithm design. It can be further resolved into two aspects, functional resolution and data resolution. The functional resolution is to resolve the problem into different sub-functions according to the functions should be finished and assign each sub-function to different processors for implementation, requiring a lower coupling among sub-functions. The data resolution is to resolve the large-scale data and each processor should process the assigned data by using the same method. The parallel strategy of ant colony algorithm under the cloud computing environment implemented in this paper is to parallel the independent ant colony and its concrete manifestation is shown as below. To store the transportation data, pheromone data and parameters data of the parallel ant colony system in DDB HBase on Hadoop platform; To assign a certain amount of ant colony to each Map, read the transportation, pheromone and parameters data of each ant from the HBase, select paths according to the transition probability formula, construct feasible solutions and finally output each ant's feasible solution as value; To obtain the current optimal solution of each Map output, update and put the global pheromone data into HBase; To conduct several MapReduce iterations to eventually obtain the optimal solution.

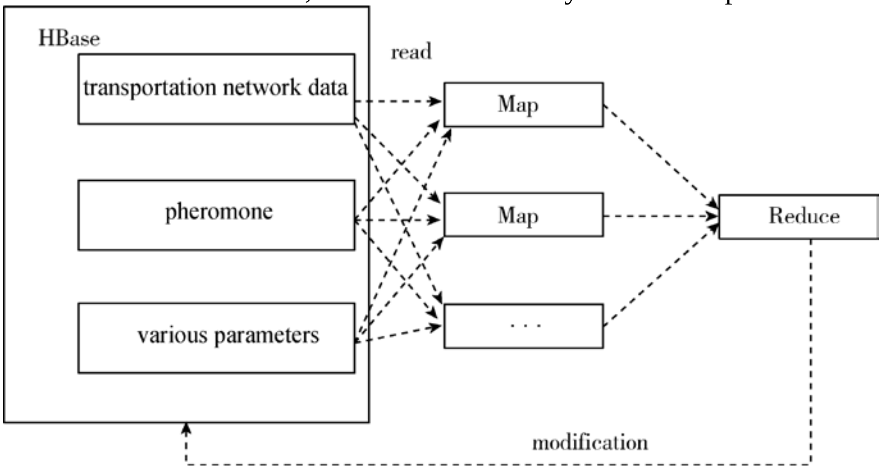

Figure 1. MapReduce Process of Ant Colony Algorithm

The detailed implementation process is as follows:

(1) In Map period, to configure MapReduce parameters, read the maximum number of iterations NC max in MapReduce process from the DDB HBase and start the MapReduce process.

(2) To read various parameters data from the DDB HBase, initialize the pheromone heuristic factor, distance information heuristic factor as well as starting and end point.

(3) To initialize the ant colony of each Map, place each ant at the starting point, read the next node information can be chosen in this node from HBase, read pheromone and weighted distance information between this node and the next node information can be chosen, select next node according to the state transition probability formula $P_{ij}^{k}(t)$.The rest can be done in same manner until we reached the end point and output the path and path length selected by each ant as value.

(4) In Reduce period, read the parameter configuration data from HBase and initialize the pheromone evaporation factor.

(5) To calculate the current shortest path and write it in the HBase according to the path and path length selected by each ant output in the Map period; To update global pheromone in accordance with pheromone updating rule and also write the updated pheromone in the HBase.

4. Intelligent Transportation System on the Basis of Cloud Computing and Experimental Data Analysis

4.1 Transportation Network Data Structure

As previously mentioned, the transportation system is a large and complex system. When carrying out the path query in the transportation network, the optimal path got merely by considering each node and path length is far from meeting people's needs. Factors such as various road traffic infrastructures (including signs, marked lines and signal lamps, etc), traffic flow and traffic accidents that may occur from time to time in the transportation network would exert certain influences. As a result, all foregoing factors should be taken into consideration when designing the path query algorithm of the route guidance system. The processing of transportation network data is the basis to analyze the path query algorithm of the route guidance system. This paper analyzes VISSIM developed by PTV Company from Germany. The first step is to abstract the displayed urban road network and establish corresponding network model, where the network model refers to the network diagram. The next step is to analyze the shortest path problem in the urban road network through the network diagram and form spatial data corresponded and attribute data associated at the same time. There are generally several basic elements in the road network:

Peak (node) is marked as V: road intersection or end point. Edge is marked as E/arc is marked as A: we call the road section between two peaks as edge and if the direction of road section is regulated, we call it arc. The weight of edge or arc is marked as W: we can choose functions of different or certain road section attribute and regard the road section length and road section congestion extent as the weight of corresponding edge or arc. After regulating the peak, edge (arc) and its weight, the road network is abstracted as a weighted undirected graph or weighted directed graph. In this way, querying the shortest path between two places has turned into querying shortest path in the graph. Under normal circumstances, the size of the traffic flow in two directions of the road section is inconsistent and the weight calculating of shortest path also has something to do with the traffic flow, so we should use the directed graph to represent the transportation network.

On the whole, the entire transportation network can be represented by a directed graph $G(V,E,W)$. Where V is a set of peaks, $V=\{V_i \mid i=1,2,\ldots,n\}$; E is a set of edges, $E=\{e(V_i,V_j) \mid V_i,V_j \in V\}$; W is a set of weights, $W=\{W(V_i,V_j) \mid V_i,V_j \in V\}$.

As a simple road network abstracted on the basis of the foregoing principles, Figure 2 can determine the shortest path between two places by turning it into the shortest path between two nodes in the graph.

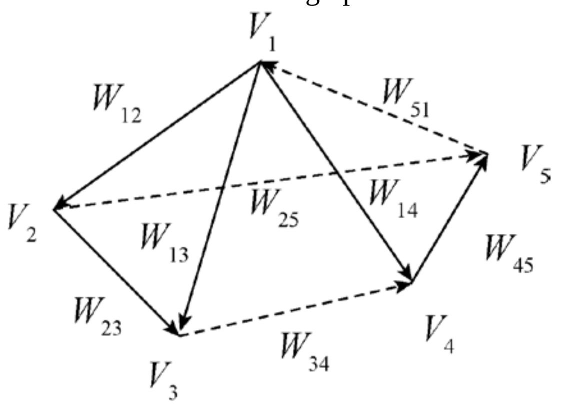

Figure 2. Abstract Graph of Road Network

In the traffic guidance system, determining dynamic driving time of road section is the most important problem to be solved. As the most important parameter to describe the road traffic status, the driving time during the road section may directly reflect the road congestion situation and be deemed as the basis of path query. The driving time of road section is in relation to the traffic capacity, signal control, road section length, interaction between vehicles and other factors. In order to avoid making the problem to be rather complex, this paper only consider the traffic flow and signal control. The total time T_k to pass the road section mainly includes three parts: driving time T_k^r, queue waiting time T_k^q and time to pass the interactions T_k^c.

4.2 Calculation of Road Section Driving Time

We can obtain the transportation data required for calculating the road section driving time by various transportation facility testing tools, such as loop coil detector, radar and supersonic ultrasonic detector, of which the loop coil detector has the highest use frequency. Based on the lane loop coil buried underground, the loop coil detector gets the traffic flow, time occupancy ratio and vehicle velocity through changes caused by coils and vehicles existing on the coils. The time occupancy ratio of the road section refers to the ratio between the total time all vehicles uses to pass a certain road section and estimated total time.

$$\theta_k^t = \frac{1}{r}\sum_{i=1}^{N} t_i \qquad (6)$$

This paper uses the ratio between the number of vehicles and time occupancy ratio to estimate the average driving time of the road section. Where Le represents the sum of average vehicle length and detector length, then the number of vehicle is $S_k^t = L_k \cdot K_k^t = \theta_k^t \cdot L_k / L_e$

4.3 Queue Waiting Time

According to the queue theory, the average time consumed when queuing is:

$$d = 1/(u-\lambda) \qquad (7)$$

4.4 Time to Pass the Interactions

According to the traffic flow theory, the user service time in the queuing system is the rate of service and the time for each vehicle to pass the interactions is:

$$T_k^c = 1 \big/ g_k^t \cdot C_k \qquad (8)$$

The data on cloud computing platform Hadoop is stored in the DFS HDFS and DDB HBase. Hence, the data obtained from transportation supervision should be stored in DDB HBase after getting the data type that can be processed by the query algorithm with the foregoing methods. In practice, the data in transportation system is constantly changing and only real-time updating the data in HBase can ensure the validity of the search results by query algorithm.

This paper achieves the parallel ant colony system based on the cloud computing, deals with the transportation network data and provides the connector stored in HBase. This paper implements type of Map by inheriting type of MapReduce and Mapper. There are four parameter types to designate types of load key, input value, output key and output value, respectively. Both the input and output keys are a long integer offset and both the input and output values are a line of text. Read related parameters from HBase first, initialize each ant in the ant colony, then construct feasible solution by using ant related methods and eventually output each feasible solution. The structure of Map type is shown in Figure 3:

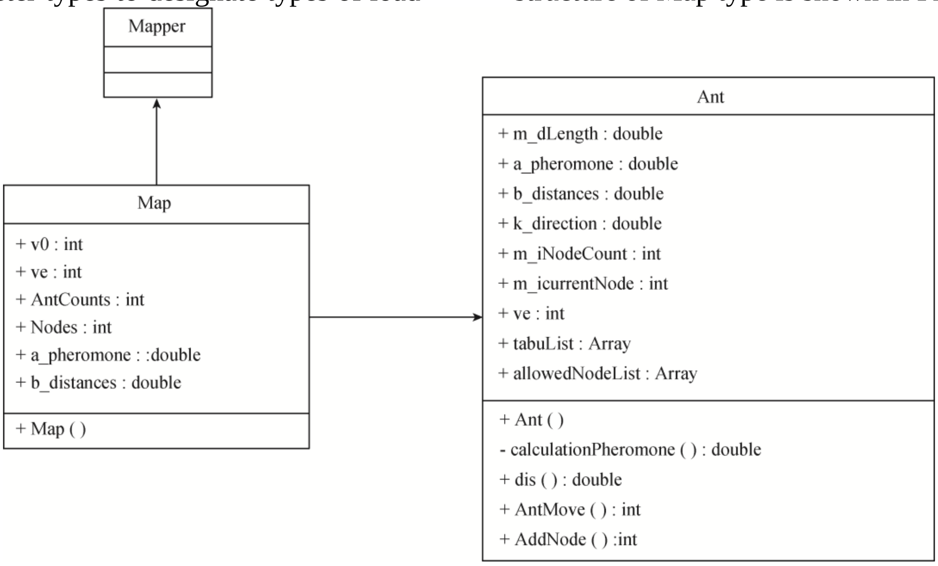

Figure 3. Structure Diagram of Map Type

This paper implements the algorithm by adopting Java language on Eclipse and the experimental environment is a cluster system comprised by 10 Lenovo PCs with Intel ® Core ™Duo 2.99GHz, SD 2GB and 300GB Hard Disk. Each PC is equipped with Ubuntu 12.04 OS and Hadoop cloud computing platform.

The parameters of parallel ant colony algorithm set in this paper are $\alpha = 1, \beta = 4, \rho = 0.4, NC_{\max} = 10, Q = 100,$ and the total number of ants is 1200. We set the number of Map as 1~10 to conduct experiment and the run results of parallel ant colony algorithm are shown in Figure 4. As shown in the figure, the runtime of improved ant colony algorithm is obviously less than the runtime of traditional ant colony algorithm.

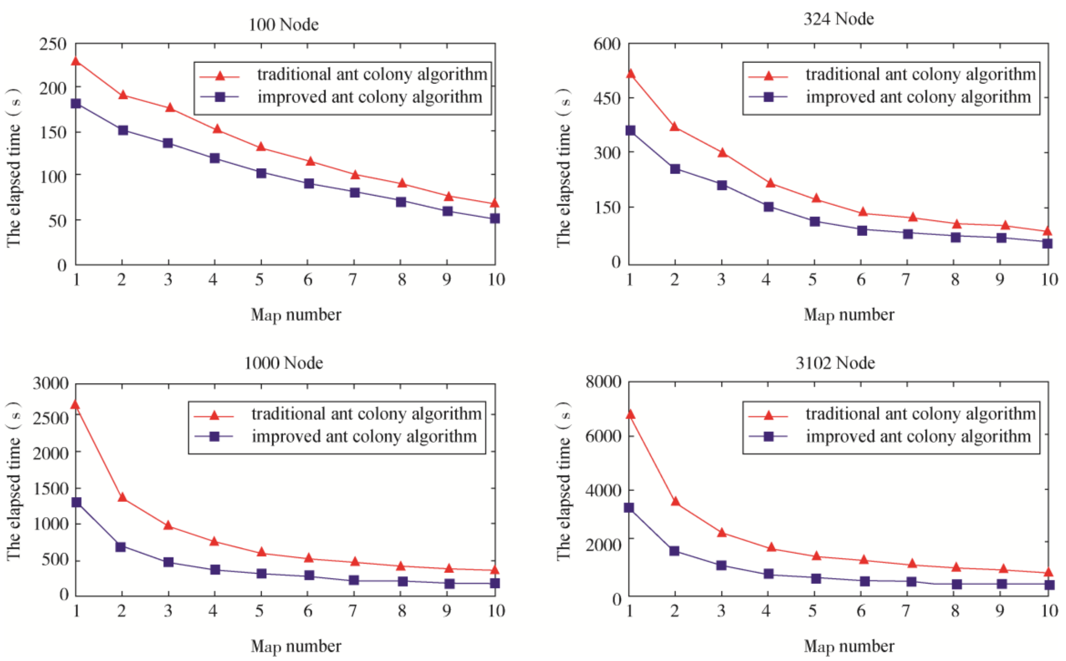

Figure 4. Run Results of Parallel Ant Colony Algorithm Based on Cloud Computing

When the number of Map is 1, algorithm conducts serial running. Therefore, as shown in Figure 5, when the number of nodes is 100, the runtime has no obvious change and more nodes lead to more obvious acceleration effect.

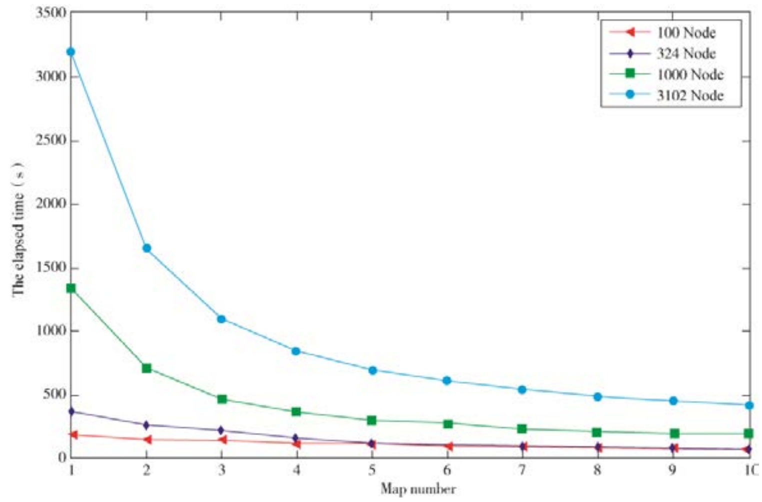

Figure 5. Run Results of Improved Parallel Ant Colony Algorithm Based on Cloud Computing

By computational formula S = serial algorithm runtime/parallel algorithm runtime, we can obtain speed-up ratio of parallel algorithm and quantitatively analyze algorithm efficiency. If the number of Map is 1, the parallel algorithm runtime equals to serial algorithm runtime and the obtained speed-up ratio is shown in Figure 6. Similarly, the speed-up ratio is relatively small when the number of nodes is 100. More nodes shall lead to larger speed-up ratio, and when the nodes reach to a certain scale, the speed-up ratio has no obvious change.

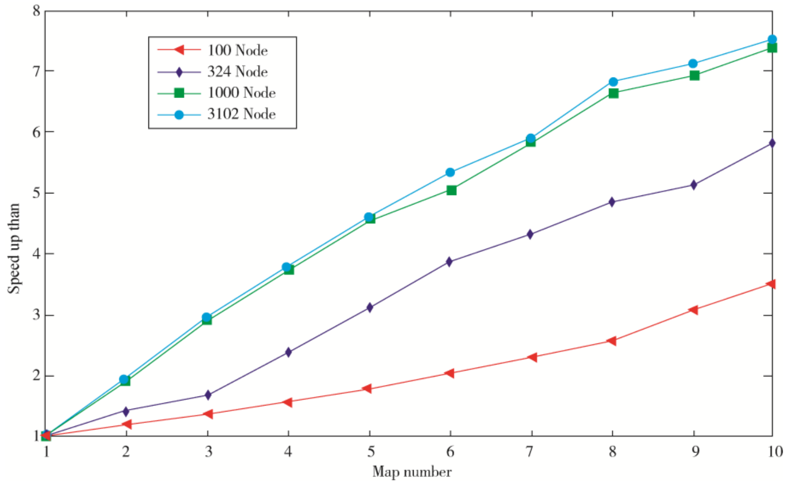

Figure 6. Speed-Up Ratio of Runtime of Improved Parallel Ant Colony Algorithm

5. Conclusion

In order to improve the massive data processing capacity, people have delved in-depth studies on cloud computing technology and successfully applied it into transportation field. By delving into the transportation network path query algorithm based on cloud computing, this paper has designed and implemented the parallel ant colony algorithm based on MapReduce on cloud computing platform Hadoop and created the intelligent transportation control system. The experimental results have demonstrated that the algorithm is efficient.

6. Acknowledgments

This paper is a science and technology key project of social development in Shaanxi Province(NO.:2013B0320181); and a scientific research launching project of overseas students returning to China(NO.: 2014G1321039); and a project of natural science funds in Shaanxi Province and a basic scientific research project of central colleges and universities (NO.:2013jm8017).

References

1. Zhaosheng Y, Lianyu C. (2010) Study Progress of Dynamic Route Guidance System. *Highway Transportation Technology*, 2(1): p.p. 35-38.
2. Michelle H., Valerie T., David B. (2001) Implementing Parallel Shortest Path for Parallel Transportation Applications. *Parallel Computing*, 27(12): p.p.1537-1568.
3. Pallottino S. (2014) *Shortest Path Methods: Complexity, Interrelation and New Propositions. Network*, Wiley Interscience: New York.
4. Derg A.V., Radzik T.A. (2013) Heuristic Improvement of the Bellman-Ford Algorithm. *Applied Mathematics Letters* 6(3), p.p.3-6.
5. Eve D. P., Bertsekas J. N. (2010) Partially asynchronous Parallel Algorithms for Network Flow and Other Problems. *SIAM Journal on Control and Optimization*. 28(3), p.p.678-710.
6. Rpaige C., KRUSKAL P., (2005) Parallel Algorithms for Shortest Path Problems. *Proceeding of International Conference on Parallel Processing*. New York, USA, p.p.14~20.
7. Mahabbal H., Koutsopoulos S. (2014) Solution Algorithm for the all-pairs shortest path problem on Massively Parallel Computer Architectures. *Transportation Science*, 28(3), p.p.:273-290.
8. Randall M. (2012) A Parallel Implementation of Ant Colony Optimization. *Journal of Parallel and Distributed Computing*, 40(6), p.p. 1421-1432.