

## Research of Resource Scheduling Strategy in Cloud Computing

<sup>1</sup>Ying Gao, <sup>1,2</sup>Guang Yang, <sup>3</sup>Yanglin Ma, <sup>4</sup>Mu Lei and <sup>5</sup>Jiajie Duan

<sup>1, 2, 3</sup> College of Computer Science and Engineering, South China University of Technology (SCUT) Guangzhou 510006, China, Ph. /Fax: 39380281

<sup>4</sup> Huanggang Dong Road, Tiaoqiao District, JiNan, Shandong Province,

<sup>5</sup> YunNan Electric Power Test & Research Institute Group CO., LTD. China

<sup>1</sup>gaoying@scut.edu.cn, <sup>2</sup>969041723@qq.com,

<sup>3</sup>191562160@qq.com, <sup>4</sup>sjjy\_66@vip.163.com, <sup>5</sup>86797485@qq.com

### Abstract

*To solve the cloud computing resource scheduling problem in IaaS platform, a scheduling model based on ant colony algorithm was proposed. In this model, pheromone changes dynamically according to the best route searched by ants. This model automatically updates pheromones and guides ants to search the global best route. Experiment results show that the proposed model is of better ability in energy consumption in the IaaS cloud computing platform.*

**Keywords:** Cloud computing, Resource scheduling, Ant colony optimization

## 1. Introduction

Cloud computing is a computing model which provides dynamic, scalable virtual resources as a service through the Internet. According to the definition of the NIST, cloud computing service model can be divided into three kinds which are referred to as SPI, namely SaaS, PaaS and IaaS [1]. Infrastructure as a service (IaaS) delivers hardware (including servers, storage devices and network, etc.) and software (including the virtualization of operating system, the file system, etc.) as a service [2]. The difference between IaaS and traditional hosting is that IaaS is able to allocate resources dynamically according to the requirements of users. Compared to the platform as a service (PaaS), the main job of IaaS service provider is to deploy, manage and dynamically allocate virtual machines [3].

It requires a lot of energy to run large-scale clusters and data centers in cloud computing platform. And the Energy consumption has become the primary problem of IT companies [4]. For example, in 2006, data centers consumed about 45 billion kilowatt hours of electricity, which equivalent to 1.5 percent of the entire U.S. electricity consumption and the energy consumption is rising at a rate of 18% per annum. So it is critical that the IaaS service providers should do their best to save resource consumption while ensuring the quality of services.

Traditional method of energy conservation and emissions reduction is to reduce the utilization rate of resources by adjusting the frequency and CPU voltage of the host. However, the SPEC experiments show that resource utilization cannot be effectively reduced by adjusting the CPU clock speed and voltage of host. The reason is that other devices (such as hard drives and network card) of the system's power consumption cannot be affected by adjusting CPU clock speed and voltage of the host. The most effective way is to shut down the host. Thanks to the rapidly development of the virtual machine technology, the virtual machine can be migrated without attracting the user's attention. Through virtual machine migration technology, virtual machines in low-load hosts can be migrated out and shut down, thus saves energy consumption. When resetting and

migrating the virtual machines, two factors should be considered: the virtual machines need to be migrated and the target hosts.

An energy-aware heuristic algorithm called EnaCloud was proposed in Literature. The algorithm grades virtual machines, and uses the method of inserting and swapping out to deploy the virtual machines, so as to achieve the energy saving effect. However, migration of large number of virtual machines is needed during the deployment of virtual machines in this model, which affect its performance. An ant colony algorithm with performance perception, proposed in literature [5], was used in batch deployment of the virtual machines. The migration of virtual machines on low-load hosts is not being considered. Therefore the model is not suitable for use in the environment where virtual machines change frequently. The resource scheduling strategy is divided into four parts in literature [6], and corresponding scheduling algorithm is designed for each part. However the algorithms need a lot of manual work, which is not suitable for large-scale clusters.

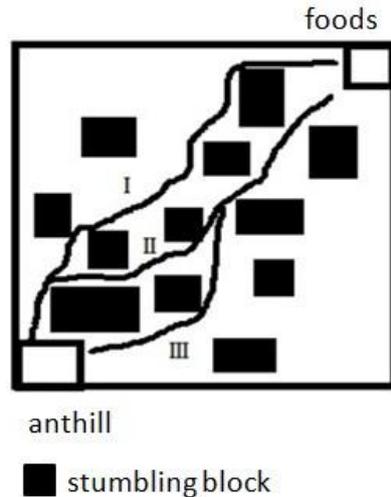
According to the characteristics of migration and reset of virtual machines in cloud platform, an ant colony algorithm based on fluctuation range was proposed, which uses the guided nature of pheromone in ant colony algorithm. The fluctuation range is calculated through the difference between the optimal path and suboptimal path, and the result is used to explore the optimal path for subsequent colony. In this paper, the improved ant colony algorithm will be introduced first, and then the algorithm is applied to the resource scheduling in cloud platform. Finally, the experiments show that the improved ant colony algorithm can save resource consumption and improve the utilization of cloud platforms.

## 2. The Improved Ant Colony Algorithm

Before introducing the resource scheduling algorithm in detail, the proposed improved ant colony algorithm is introduced. And then, the algorithm is used in the IaaS platform. Virtual machines are migrated to the appropriate hosts, reducing resource consumption.

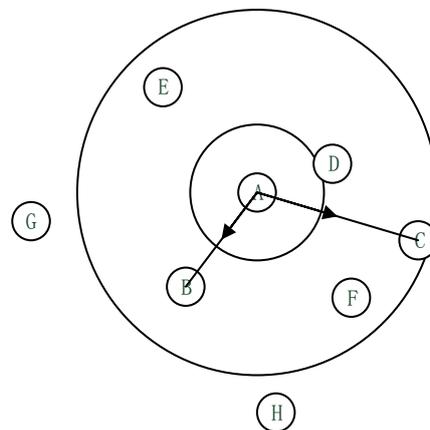
As a biological optimization algorithm, ant colony algorithm has positive feedback, heuristic search features [7]. In every search of the ants, an optimal path and a suboptimal path can be found [8]. Use the optimal path as the axis, spread around, until spread to the suboptimal path, a diffusion range can be found. The range is called fluctuation range. In this range, the probability that there exists a path better than the current optimal path is rather high. Using the fluctuation range to guide subsequent search of the ants will help to improve the ability of searching the optimal solution.

To explain it better, Figure 1 is used to illustrate the principle of the improved ant colony algorithm. Assuming that the ants embark from the anthill, the goal is the foods. And assuming that the optimal path the ants have found is I, suboptimal path is III. If the scope between the optimal path I and suboptimal path III is added to the search range, a path (path II) better than I can be found.



**Figure 1. Ant Foraging Routes**

Extend to the TSP problem. When an ant searches paths, assuming that the ant is now in city A, in the current optimal path, city A's next goal is to city B, city B's next goal is to city H. And in sub-optimal path, city A's next goal is to city C. For simplicity, the path which is better than the current optimal path is called the more optimal path in this paper, and the optimal path currently found is called the current optimal path. Use point A as the center of the circle, use distance AC to make a ring called outer ring, and use  $AC - 2 * (AC - AB)$  as distance to make a ring called inner ring, as shown in Figure 2. The inner ring and outer ring make the concentric ring. The probability that a better path exists in the concentric ring is rather high, so cities in the concentric rings should be a priority to search. The concentric ring is called fluctuation range in this paper; means that the more optimal path is searched fluctuate around the current optimal path.



**Figure 2. The Fluctuation Range**

The Improved ant colony algorithm guides subsequent search of the ant colony by modifying the update method of the pheromones. After every iteration in the ant colony, the current optimal path and suboptimal path will be recorded, and the fluctuation range will be calculated based on the current optimal path and suboptimal path, and then the cities in fluctuation range will be added to the pheromones, which will be used to guide the subsequent search of the ant colony. Given that the current optimal path and suboptimal path are constantly changing,

only the pheromones in the fluctuation range of the current optimal path will be added, pheromones added previously will not be retained, thus the convergence of the algorithm will not be affected.

First, find the fluctuation range. After iteration, every city has a certain length of fluctuations distance. The distance between city A and its next city in the optimal path is denoted by  $d_1$ , and the distance between city A and its next city in the suboptimal path is denoted by  $d_2$ . Then city A's fluctuation distance is the difference between  $d_1$  and  $d_2$ . The distance between city i and city j is denoted by  $d_{ij}$ , the number of cities is denoted by CN, the current optimal path is denoted by bsr,  $bsr_1$  is the first city to be visited in bsr, the suboptimal path is denoted by btr,  $btr_1$  is the first city to be visited in btr. Then the fluctuation distance of city i is:

$$wd(i) = \left| d_{bsr_k bsr_{k+1}} - d_{btr_j btr_{j+1}} \right| \quad (1)$$

Among them,  $bsr_k = btr_j = i$ . If  $i + 1 > CN$ , then  $bsr_{i+1} = bsr_1$ , else if  $j+1 > CN$ ,  $btr_{j+1} = btr_1$ .

Next, find cities in the fluctuation distance of city i. The distance between city i and j is denoted by  $d_{ij}$ , and the distance between city k and its next city in bsr is denoted by  $d_{bsr_k bsr_{k+1}}$ . Then the difference between  $d_{ij}$  and  $d_{bsr_k bsr_{k+1}}$  is:

$$dd(i, j) = \left| d_{ij} - d_{bsr_k bsr_{k+1}} \right| \quad (2)$$

Among them,  $bsr_k = i$ , and if  $k+1 > CN$ , then  $bsr_{k+1} = bsr_1$ .

If  $dd(i, j) \leq wd(i)$ , it means that city j is in the fluctuation distance of city i. For cities in the fluctuation distance of i, fluctuation pheromone will be used to guide the subsequent search by the ant colony. The fluctuation pheromone is defined as:

$$w\tau(bsr_k, j) = \left( 1 - \frac{dd(bsr_k, j)}{wd(bsr_k)} \right) * \frac{Q}{L_h} \quad (3)$$

Among them,  $dd(bsr_k, j) \leq wd(bsr_k)$ ; If  $dd(bsr_k, j) \leq wd(bsr_k)$ , then  $dd(bsr_k, j) / wd(bsr_k) = 0$ ;  $L_h$  is the length of the h-th iteration optimal path. That is, for city j which in the fluctuation distance of  $bsr_k$ , the weights of the fluctuation pheromone will be increased according to dd's value of city j. The smaller the dd's value is, the more fluctuation pheromone will be added.

Given that the current optimal path and suboptimal path is constantly changing, only the pheromone in the fluctuation range of the current optimal path will be increased, pheromone added previously will not be retained, thus the convergence of the algorithm will not be affected. To eliminate this effect,  $\tau'$  is used to record the pheromone matrix that pheromone is not increased.

$$\tau'_{ij}(t+1) = (1 - \rho) * \tau'_{ij}(t) + \Delta\tau_{ij}(t, t+1) \quad (4)$$

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^m \Delta\tau_{ij}^k(t, t+1) \quad (5)$$

$$\Delta\tau_{ij}(t, t+1) = \sum_{k=1}^m \Delta\tau_{ij}^k(t, t+1) \quad (6)$$

$$\Delta\tau_{ij}^k(t, t+1) = \begin{cases} \frac{Q}{L_k} & k \text{ Ant pass through the path}(i, j) \\ 0 & \text{else} \end{cases} \quad (7)$$

Pheromone is updated according to the following formula:

$$\tau_{ij}(t+1) = \tau'_{ij}(t+1) + e * w\tau \quad (8)$$

Equation (7) shows that fluctuation pheromone affects only the next iteration; the subsequent iterations will not be affected.

The improved ant colony algorithm is described as follows:

(1) Initialize the parameters. Set the number of ant's  $m$ , the number of iterations  $N=0$ , the maximum number of iterations  $N_{max}$ , empty the taboo list  $tabu$ , put the  $m$  ants into  $n$  cities, initialize the pheromone matrix  $\tau_{ij}(0) = C$ , where  $C$  is a constant. Set the weight of fluctuation pheromone  $e$ , the increment of pheromone  $\Delta\tau_{ij}=0$ , the stimulating factor of pheromone  $\alpha$  and the expected heuristic factor  $\beta$ .

(2) Iteration number is denoted by  $N$ , then  $N \leftarrow N+1$ ,  $k \leftarrow 0$ .

(3) The ant number is denoted by  $k$ .  $k \leftarrow k+1$ . If  $k > m$ , then jump to step (8).

(4) Put the city in which the ant current lives into the taboo list  $tabu$ .

(5) The ants calculate the probability according to the probability of state transition, then select the next city and move on using roulette.

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t)\eta_{ij}^\beta(t)}{\sum_{s \in allowed_k} \tau_{ij}^\alpha(t)\eta_{ij}^\beta(t)}, & j \in allowed_k \\ 0 & otherwise \end{cases} \quad (9)$$

(6) Put the city in which the ant current lives into the taboo list, and move the ant to the new city.

(7) If all  $n$  cities have been traversed, the current optimal path and suboptimal path will be updated according to the taboo list, then jump to step (3); if else, jump to step (4).

(8) Update the pheromones in each path according to formula (4), (5) and (6).

(9) If the end condition is met, that is, if the number of cycles  $N > N_{max}$ , end the loop and output the calculation result, otherwise empty taboo list and jump to the first (2) step.

### 3. Description of the Virtual Machine Reset and Migration Algorithm

The reset of the virtual machine has been described in detail before. In this section, the improved ant colony algorithm described in previous section will be used to solve the reset and migration problem of the virtual machines.

Set a threshold value  $TS$ , and add all of the hosts of which the load is lower than the threshold  $TS$  to the candidate host migration list. Order the candidate host migration list according to the load from small to large. The goal is to bring the virtual machines that on the candidate migration host to be running on the hosts that don't need to migrate. Now the virtual machine list and host list can be got. If the host list has inadequate resources to migrate the virtual machines, remove the host that has the largest load from the candidate host migration list, re-construct the virtual machine list and the host list until all virtual machines can be migrated to the host.

In order to standardize the comparisons between the host and the virtual machine to be migrated, the hosts and virtual machine need to be normalized. The normalization algorithm of hosts is defined as  $H'_{ip} = \frac{H_{ip} - H_p^{min}}{H_p^{max} - H_p^{min}}$ , where  $H_p^{min}$  is the minimum value of similar performance parameter  $p$  of the hosts, and  $H_p^{max}$  is the maximum value of the similar performance parameter  $p$  of the hosts,  $H_{ip}$  is the performance of current host when it is free. Similarly, the normalization algorithm of virtual machine is defined as  $V'_{ip} = \frac{V_{ip} - V_p^{min}}{V_p^{max} - V_p^{min}}$ , where  $V_p^{min}$  is the minimum value of similar performance parameter  $p$  of the virtual machines,  $V_p^{max}$  is the maximum value of the similar performance parameter  $p$  of the virtual machines,  $V_{ip}$  is virtual machine's requirements on performance parameter  $p$ . For simplicity, if the maximum and the minimum performance value is the same, the normalized value is 1.

The Euclidean distance will be used as a measure standard of the degree to which the virtual machine matches the host, that is, the shorter the Euclidean distance is, the better the host meets the virtual machine's performance requirements, the lower the cost is needed to deploy the virtual machine to the host [9]. The Euclidean distance between the

requirements of the virtual machine need to be deployed and the performance of the free

host is defined as 
$$D_{ij} = \sqrt{\sum_{p=1}^k \frac{(H'_{ip} - V'_{ip})^2}{k}}$$
.

Except the degree to which the host matches the virtual machine, the costs of migration should also be considered. Current common virtual machine migration methods are simply copy, lazy copy and pre-replication [10]. Virtual machine migration time is mainly determined by the memory size of the virtual machine. The formula  $COST_i = \frac{m_i}{b_j}$  is used to measure the migration cost in this paper. In the formula,  $m_i$  is the memory size of the virtual machine,  $b_j$  is the minimum bandwidth of the host where the virtual machine lives and the target host. And the normalized parameter is  $COST'_{ij} = \frac{COST_{ij} - COST^{min}}{COST^{max} - COST^{min}}$ .

Therefore, the total distance of migration is as follows:

$$E_{ij} = D_{ij} * COST'_{ij} \tag{9}$$

The goal is to minimize the total distance of migration.

Specific steps are as follows:

- (1) Initialize the number of iterations  $\leftarrow 0$ , the virtual machine list *vm*list of which the virtual machine need to be reset, and the host list *hot*list.
- (2) According to the virtual machine list *vm*list and host list *hot*list, calculate the Euclidean distance of virtual machines and hosts, and calculate the migration cost. Calculate the distance matrix according to the formula (9).
- (3) Initialize the ant's number  $k \leftarrow 0$ .
- (4) An Ant selects a host from the *hot*list according to formula (8); the host will be used to deploy virtual machine later. If the host does not meet the deployment requirements of virtual machine, the host should not be considered. In order to avoid migrate multiple virtual machines to one host at a time; the host will be put into the *taboo* list.
- (5) If *vm*list is not empty, jump to step (3).
- (6) Update the current optimal path and suboptimal path according to the *taboo* list. If  $k < MAX\_ANT$ ,  $k \leftarrow k + 1$ , and jump to step (3).
- (7)  $t \leftarrow t + 1$ .
- (8) Update the pheromone according to the formula (4), (5), (6), (7). When the end condition  $t > MAX$  is reached, record the optimal path and deploy the virtual machines that in the *vm*list to the hosts in order according to the optimal path. Else jump to step (3).

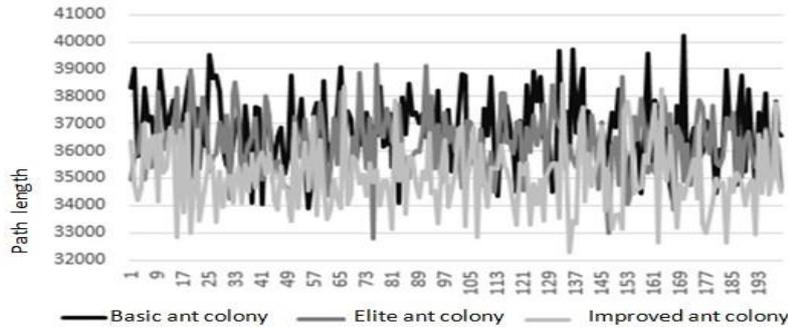
#### 4. Simulation Results and Analysis

First, test the improved ant colony algorithm on the TSP problem, then the performance of the algorithm will be compared, and finally, simulate the algorithm using the CloudSim platform.

In order to verify the validity of the improved ant colony algorithm, the improved ant colony algorithm will be used to solve the TSP problem. TSPLIB48 algorithm will be adopted as a contrast experiment. Table 1 shows the experiment results of the improved ant colony algorithm. In the experiment, the number of ants is 20, and the experiment was run 200 times. Figure 3 is a comparison of the colony experiments that obtain the path. The elite impact factor of the elite colony algorithm is 0.5.

**Table 1. The Results of Ant Colony Algorithm**

	longest route	minimum route	average route	iterationmean
Basic ant colony	40207	33911	36750	76.225
Elite ant colony	39151	32785	36228	109.085
Improved ant colony	38451	32307	35055	136.865



**Figure 3. A Comparison of the Colony Experiments that Obtain the Path**

The experiment result shows that the basic ant colony algorithm is able to obtain the optimal result using the smallest number of iterations, but it is easy to fall into the local optimal solution. The elite ant colony algorithm is able to provide the optimal solution. And the improved ant colony algorithm is easier than the first two algorithms to obtain the global optimal solution, and not easy to fall into the local optimal solution. It should be noted that, due to the addition of fluctuation pheromones, convergence speed of the improved ant colony algorithm is lower.

The table below is the results of 200 times ant colony experiments using different fluctuation impact factors. The table shows that the convergence speed of the algorithm can be improved by choosing suitable fluctuation impact factor.

**Table 2. Ant Colony Experiments Using Different Fluctuation Impact Factor**

e	longest route	minimum route	average route	iteration mean
0.2 5	38748	31716	35023	137.21
0.5	38451	32307	35055	136.865
0.7 5	38720	32676	35055	132.84

The following experiments are used to verify that the improved ant colony algorithm can be used for resource scheduling of cloud computing platform. The experiments use monitoring data [11] of PlanetLabCoMon project in March 3, 2011 as the data source. PlanetLab is a globally distributed computer cluster, mainly used for testing virtual machine deployment strategies. In the experiments, safety factor of 1.5 and a combination of maximum correlation and interquartile algorithm, safety factor of 1.5 and a combination of random selection algorithm and interquartile algorithm, safety factor of 1.5 and a combination of interquartile and minimizing usage algorithm, safety factor of 1.2 and a combination of maximum correlation and local regression algorithm,

safety factor of 1.5 and a combination of minimizing usage and local regression algorithm, and the scheduling algorithm described in this paper using threshold 0.2,0.3,0.4 are tested to compare their energy consumption. Table 3 shows that the use of the resource scheduling algorithm described in this paper can effectively save the energy consumption.

**Table 3. Energy Consumption Comparison of Scheduling Algorithms**

resource scheduling strategies	energy consumption ( kWh )
iqr_mc_1.5	178.8
iqr_rs_1.5	181.43
iqr_mu_1.5	204.22
lr_mc_1.2	164.82
lr_mu_1.2	191
ant_0.2	114.32
ant_0.3	128.09
ant_0.4	117.8

## 5. Conclusion

The improved ant colony algorithm proposed in this paper can effectively improve the ant colony algorithm's ability to explore the best solution by modifying the pheromone update method. And the algorithm is suitable for resource scheduling in IaaS platform. Experiments show that this algorithm can effectively save the resource consumption in cloud platform. Therefore, the proposed resource scheduling strategy is good enough to be applied to the cloud environment.

## Acknowledgements

The research is supported by: National Natural Science Foundation of China No. 51204071; Science and Technology Planning Project of Guangdong Province (2012B010100019); Ministry of Education University-Industry-Research Project of Guangdong Province (2010B090400535); The Fundamental Research Funds for the Central Universities (2013ZZ0047).

## References

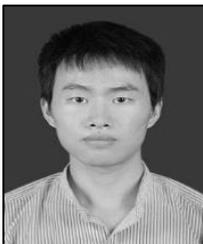
- [1] C.-Y. Tseng, K.-Y. Liu, L.-T. Lee, "Enhance the Performance of Virtual Machines by Using Cluster Computing Architecture", *TELKOMNIKA Indonesian Journal of Electrical Engineering*, (2013), vol. 11, no. 5, pp. 2553-2558.
- [2] S. Bhardwaj, L. Jain and S. Jain, "Cloud computing: A study of infrastructure as a service (IAAS)", *International Journal of engineering and information Technology*, (2010), vol. 2, no. 1, pp. 60-63.
- [3] S. Bhardwaj, L. Jain and S. Jain, "Cloud computing: a study of infrastructure as a service (IAAS)", *International Journal of Engineering and Information Technology*, (2010), vol. 2, no. 1, pp. 60-63.
- [4] A. Beloglazov, J. Abawajy and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing", *Future Generation Computer Systems*, (2012), vol. 28, no. 5, pp. 755-768.
- [5] Y. Xing, M. Zitang and S. Le, "Research on extended ant colony optimization based virtual machine deployment in infrastructure clouds", *Computer Science*, (2012), vol. 39, no. 9, pp. 33-37.
- [6] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers", *Concurrency and Computation: Practice and Experience*, (2012), vol. 24, no. 13, pp. 1397-1420.

- [7] H. Lourenco and D. Serra, "Adaptive search heuristics for the generalized assignment problem" *Math ware & soft computing*, (2002), vol. 2, no. 3, pp. 209-234.
- [8] C. Qi, "Vehicle Routing Optimization in Logistics Distribution Using Hybrid Ant Colony Algorithm", *TELKOMNIKA Indonesian Journal of Electrical Engineering*, (2013), vol. 11, no. 9, pp. 5308-5315.
- [9] S. Campanoni, G. Agosta, S. Reghizzi, "A highly flexible, parallel virtual machine", design and experience of ILDJIT, *Software: Practice and Experience*, (2010), vol. 40, no. 2, pp. 177-207.
- [10] J. SMITH, "A survey of process migration mechanisms", *ACM SIGOPS Operating Systems Review*, (1988), vol. 22, no. 3, pp. 28-40.
- [11] K. Park and V. Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab", *ACM SIGOPS Operating Systems Review*, (2006), vol. 40, no. 1, pp. 65-74.

## Authors



**Ying Gao**, Born on January, 1974, Hunan Province, and PRC  
Current position, grades: PhD, professor  
Research interests: Computational Method, Cloud Computing, Software Development Environment and Integrated Information, Software Engineering



**Guang Yang**, Born on April, 1988, Anhui Province, PRC  
Current position, grades: graduate student  
University studies: School of Computer Science, South China University of Technology  
Scientific interest: Cloud Computing, Computational Method, Software Development.



**Yanglin Ma**, Born on February, 1988, Anhui Province, PRC  
Current position, grades: graduate student, R&D Engineer  
Scientific interest: Cloud Computing, Computational Method, Software Development



**Jiajie Duan**, Born on September, 1986, Yunan Province, PRC  
Current position, grades: software engineer  
Scientific interests: Software Development, Software Engineering



**Lei Mu**, Born on February 28, 1968, Shandong Province, PRC  
Current position, grades: senior engineer  
Scientific interests: Software Development, Intelligent Traffic System

