International Journal of Grid and Distributed Computing
Vol. 9, No. 10 (2016), pp.221-230
http://dx.doi.org/10.14257/ijgdc.2016.9.10.20

Research on the Optimal Task Scheduling Algorithm Based on SDN Architecture

Zhe Li[1,2], Zhi-Long Deng[3] and Tian-Fan Zhang[1,2]

[1]*College of Technology, Hubei Engineering University, Xiao Gan 432000, China*
[2]*Department of Automatic Control, Northwestern Polytechnical University, Xi'an 710072, China*
[3]*Department of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China*
[1]*lizhe_hbeu@vip.163.com and* [2]*alitasoft@hotmail.com*

Abstract

A new task scheduling algorithm based on Hadoop is proposed to optimize scheduling of resources problems under the Distributed cloud-computing platforms. The core idea of the algorithm is full reference to the current network conditions and treat it as an important reference for system task scheduling, with the bandwidth management ability ,SDN architecture allows us to allocate bandwidth according to a time slot strategy, then according to the operation completed sooner or later the time to decide whether the task assigned to the local node or low load of non-local node .In this way, we will not only ensure the locality of the task from a global perspective , but also assign tasks in an optimal way to efficiently. In the end, we do experiments based on the scheduler to verify the quality of scheduling algorithm.

Keywords: *cloud computing; Hadoop; SDN; task scheduling; load equalization*

1. Introduction

With the development and progress of grid computing, service-oriented and virtualization technology, cloud computing [1] received strong support and promote in the theory and technology, gradually becoming the focus of attention and the future development trend of computing models.

However, the large-scale network of cloud computing, flexible change of network requirements, dynamically scalable server resources network configuration and the multi-service provider's own network interfaces commonly proposed demanding of the standardized interface and control [2] . But the traditional network architecture could not be well adapt ed to changes of these requirements [3] .As a new network architecture, software-defined networking(SDN) can be abstraction and simplification of the existing complex network system to decoupling the control layer and data layer, allows developers and general users to the network-oriented programming systems. Sandhya *et. al.,* [4] propose a method to improve JobTracker in the Hadoop based on SDN. Bailey *et. al.,*[5] using QoS queuing mechanisms to improve Hadoop priority based on OpenFlow which is a concrete implementation of the SDN.

In this paper, the cloud platform network architecture is designed and researched based on a kind of SDN, making the use of bandwidth's capabilities of management and control for SDN to be complement of cloud computing, resulting in the optimization of problems of task scheduling for Hadoop platform.

A rigorous definition of self-organization for dynamic systems was proposed by Anceaume *et. al.,* [6] Aroca *et. al.,* [7] propose an algorithm for the online scheduling problem and studied the competitiveness of proposed solution. Sheilchalishahi *et. al.,* [8]

ISSN: 2005-4262 IJGDC

proposed a solution for scheduling problem based on multi-capacity bin-packing algorithms, considering host selection and queuing based on multi-resource scheduling. Palmieri *et. al.*, [9] perform end-to-end path rerouting and VM migration to improve access to big data. Maheshwari *et. al.*, [10] proposes a multi-site workflow scheduling technique that uses performance models to predict the execution time on resources and dynamic probes to identify the achievable network throughput between sites.

2. Hadoop Task Scheduling Analysis

In Hadoop computing cluster, in order to avoid node going down, the data entered will be automatically backed up in multiple nodes. Before performing the calculation, Non-local task will carry out moving data , copying data required to perform tasks from the nodes storing the data to the nodes performing tasks, as a result of occupying a lot of network bandwidth resources, as shown in Figure 1:

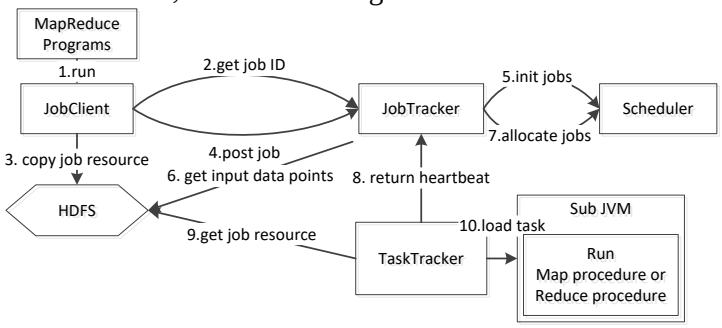

Figure 1. Basic Hadoop Task Scheduling

However, when the load above computing nodes among the Hadoop computing cluster is unbalanced, if the local data is still blindly protected and the task is given to the local node to be performed, which may lead to that high load node is assigned more task and the low-load node is not assigned tasks. Since the computing power of the individual computing node is limited, tasks on the high-load node cannot be immediately executed, the local task assigned to it must be performed after waiting for a long time, which greatly delaying the task's completion time. In this case, the computing tasks should be considered to be assigned to a low-load node, even on the condition of calculation delay caused by the network problem, task assigned to the non-local low-load node would get an earlier task completion time than task assigned to the non-local high-load node. In this paper, the cloud platform network architecture is designed and researched based on a kind of SDN, making the use of bandwidth's capabilities of management and control for SDN to be complement of cloud computing, resulting in the optimization of problems of task scheduling for Hadoop platform.

3. SHSA Task Scheduling Algorithm

3.1. Model of Job Completion Time

The ability of each node in the Hadoop cluster and the Hadoop cluster size are certain, distribution strategy influence the job completion time. Given input data in the Hadoop clusters distributed storage situation, how to distribute the task to make the completion time shortest has become a priority [11] . So Task scheduling problems of Hadoop system is converted to how to distribute the tasks for each node to ensure the job completion time shortest. Based on the above analysis, in order to better make a point of SHSA algorithm, this paper makes the following definition:

1) Hadoop job distributed is broken into m task, TK_i is defined as the job of the number I; The job of the input data was split into n, coexisting in n computing nodes, ND_j defines the computing node in j; SZ_i defined as the block size of input data.

2) Define $BW_{j,k}$ as the largest network bandwidth between ND_j and ND_k ; define $BW_{\gamma l}$ as real-time network bandwidth for the link.

3) If task TK_i is assigned to the node ND_j , the input data of TK_i could be stored in the node $ND_{dataSrc}$,when performing, the data could be needed to move the node ND_j from the node $ND_{dataSrc}$, define $TM_{i,j}$ as the data movement time; define $TP_{i,j}$ as the computing time for the task; define $TE_{i,j}$ as the actual execution time of the task.

Form (1) to (3), the formula (1) and (2) are given as bellow:

$$TM_{i,j} = SZ_i / BW_{dataSrc,j} \tag{1}$$

$$TE_{i,j} = TP_{i,j} + TM_{i,j} \tag{2}$$

4) When the node ND_j is performing, define the node is busy; if not performing, called idle state; define γI_j as the time form busy state to the idle state of the node ND_j ; When all the data have been calculated on the task TK_i , called the task completion time, define $\gamma C_{i,j}$ as the completion time for TK_i .

From 1) to 4), the Formula(3) is given as bellow:

$$\gamma C_{i,j} = TE_{i,j} + \gamma I_j \tag{3}$$

(5) For a task TK_i , a principle in which compute nodes are assigned are: to find such an available node in n compute nodes of the entire Hadoop computing clusters, which can provide the earliest task completion time. The node could be referred to the optimal node of the task [12] .

(6) For a job, only when the job of each task is completed, that could say the job is finished. From a global perspective point of view, the completion time of the entire job is the time the last task needs to be completed; so It is needed to find the slowest map or reduce task TK_i to obtain a first job completion time.

From (5) to (6), the Formula (4) is given as bellow:

$$\min\{\gamma C_{i,j} = \max \gamma C_{i,j} (1 \le i, i \le m, 1 \le j, j \le n)\} \tag{4}$$

Here the m refers to the number of task split, the number of n refers to the number of the nodes in the whole Hadoop cluster.

According to the above definition, considering the cluster bandwidth resources, especially in the presence of many non-local task, with the SDN network configuration and bandwidth management capabilities, the main ideas of SHSA algorithm is:

- Make occupancy period of the remaining bandwidth in each link divide into equal time slot and it like time-collocation of signal lamp [13] , defined $TS = \{TS_1, TS_2, \cdots, TS_k\}$, the duration of each time slot is an adjustable parameter, the parameter configuration depends on the specific network scenarios.
- The paper defines $SL_{\gamma l}$ as a residual bandwidth at a particular moment of specific link, if a map or reduce task TK_i in the time period (t_m, t_n) need to move data on a specific path, the task scheduler of Hadoop could use SDN controller assign the time slot related to the task, to ensure that all the link

bandwidth resources in the path could be reserved by TK_i at the beginning of time slot $TS_m, t_m \in TS_m$ to the end slot $TS_n, t_n \in TS_n$.

- Before task scheduling of Hadoop system, after the calculated data is carried out in the task TK_i with SDH bandwidth control capability, SDN controller could free the link, which will make link resources free for other data needed to move.

By means of bandwidth allocation mechanism based on slots, the user could make maximize use of network bandwidth capacity [14] , to reduce the time it takes to transfer the data between the data link, thereby enhancing the overall performance of computing clusters.

3.2. SHSA Task Scheduling Process

A job the user submits have tasks, Hadoop system computing cluster have available nodes for the job. Considering the Hadoop system could be Shared by multiple users at the same time, the available compute nodes may be less than the number of all the nodes of the whole Hadoop clusters. Based on the above consideration, the specific process of SHSA task scheduling algorithm is as shown in Figure 2.

```
for i = 1 to m
  A:Using SDN to get real-time bandwidth BW_rl and the
    corresponding link time interval idleresidue ratio SL_rl
  B:Looking for the ND_loc of available free time γI_loc for TK_i
  C:Looking for the ND_min now of available free time γI_min now for TK_i
  if ND_min now == ND_loc || γI_min now ≥ γI_loc then
    allocate TK_i t o ND_loc = ND_min now
  elseif ND_loc ≠ NULL && γI_min now < γI_loc then
    D:Using formula (1) and (3) to calculate γC_i,min now and the
      required bandwidth BW_i,min now
    ensure γC_i,min now < γC_i,loc
    if BW_i,min now ≤ BW_rl then
      allocate TK_i to the remote node ND_min now
      allocate SL_rl link,at the same time using formula (1) to
        calculate the timeslot number needed by TK_i
    end if
  end if
end for
return allocate all the tasks(m)
```

Figure 2. SHSA Task Scheduling Algorithm

3.3. The Design of the SHSA Task Scheduler

The SHSA scheduler module design is shown in Figure 3,which mainly divided into initialize the module, task allocation module, job queue management module, homework assigned pool and early allocation module. Among them, the Hadoop system provides interfaces for the task allocation module, initialization module and job queue management module, the SHSA scheduler implements all of these interfaces by the custom, in addition, the SHSA scheduler. The functions of each module are as follows:

1) Initialize the module: initializes jobs that the users submitted , and adds it to the job queue to wait for execution.

2) Task allocation module: Maintenances job queue to wait for execution, manages and schedules all of the Job-In-Progress which are mapped.

3) Job queue management module: The main module of SHSA scheduler for task scheduling, executes the main process of task assignment.

4) Homework assigned pool: Maintains a homework assigned pool, and stores the position of each node.

5) Early allocation module: Performs assignments pre-allocated, comprehensive consideration of local tasks and network bandwidth when the job is executed for the first time, assigns each task execution nodes and adds node queues into his homework assigned in the pool.

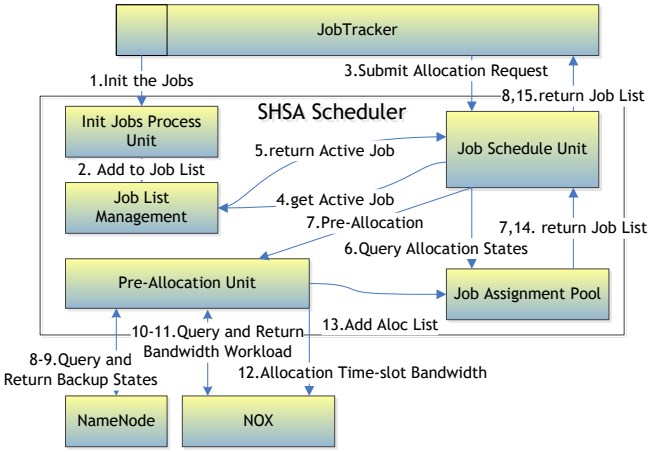

Figure 3. SHSA Scheduler Module and the Flow Chart of Scheduling

4. Experiment and Analysis of Results

4.1. Experimental Environment and Data

The test environment structure as shown in Figure 4, one Master node, four Slave nodes and SDN controller through the Open vSwitch connected to a LAN.

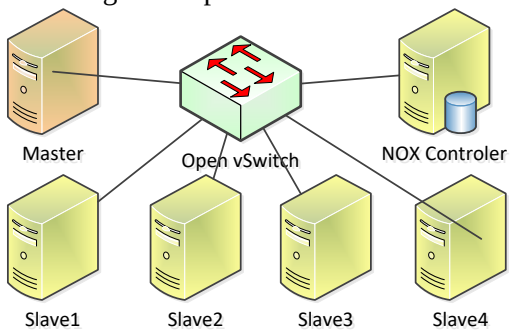

Figure 4. Hadoop Platform Deployment Environment Structure Based on SDN

The mater node and NOX controller is configured with Intel i5-4430 3GHz and equipped with 8GB memory; Slave nodes consists of two kinds of systems in heterogeneous environments, one of the categories is Intel i5-4430 3GHz with 4GB memory, the other one is NVIDIA TK1 system, 32bit quad core based on Cortex-A15 architecture with 2.3GHz, and carrying 192 core GPUs based on Kepler framework with CUDA.

Task scheduling algorithm is carried out to verify after build test platform，adopting the Sort of classic and Wordcount job to test the performance. Setting the five dimensions of the test samples, and two kinds of operation actual test results as shown in Table 1, and Table 2.

Table 1. Validation Data of Sort Assignments

Data Size	FIFO(s)				BAR(s)				SHSA(s)			
	MT	RT	JT	LR	MT	RT	JT	LR	MT	RT	JT	LR
100MB	36	51	71	0.5	34	50	67	0.25	33	43	55	0.5
250MB	64	68	117	0.5	44	87	110	0.667	50	57	91	0.667
500MB	70	119	168	0.5	79	107	155	0.5	71	102	144	0.5
1GB	146	207	323	0.625	124	196	285	0.563	148	227	262	0.563
5GB	909	1432	1859	0.638	932	1110	1632	0.71	805	1367	1572	0.663

Table 2. Validation Data Wordcount Assignments

Data Size	FIFO(s)				BAR(s)				SHSA(s)			
	MT	RT	JT	LR	MT	RT	JT	LR	MT	RT	JT	LR
100MB	35	64	78	0.333	35	57	78	0.333	36	63	78	0.333
250MB	59	109	156	0.5	66	105	146	0.667	52	94	128	0.667
500MB	137	230	269	0.833	119	194	259	0.667	97	186	229	0.583
1GB	129	291	311	0.571	125	260	305	0.762	122	255	298	0.762
5GB	637	1159	1396	0.752	659	1210	1377	0.752	487	1120	1302	0.752

(MT: the time Map task takes; RT: the time Reduce task takes; JT: the total completion time; LR: the amount of general tasks)

4.2. Contrast and Analysis of Job Completion Time

According to the test results and comparing completion time of the three schedulers, the result is shown in Figure 5, Figure 6:

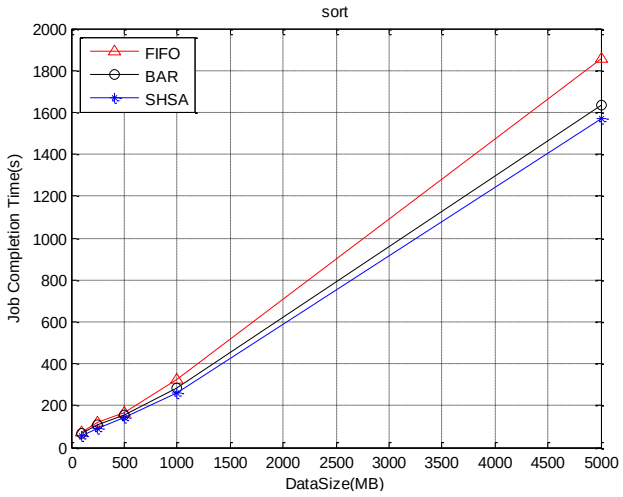

Figure 5. The Comparison Chart of Sort Job Completion Time

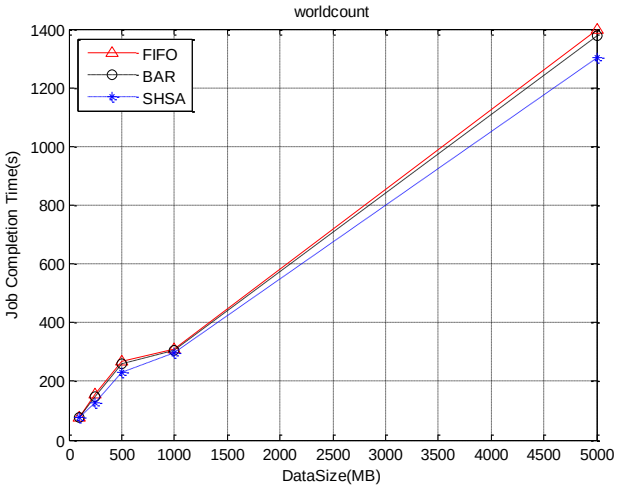

Figure 6. The Comparison Chart of Wordcount Job Completion Time

In the two categories, the Sort and Wordcount task, the time-consuming of all the tasks is rapidly rising trend .Compared to SHSA scheduler， FIFO and BAR tasks take shorter time. The main reason is that SHSA scheduler to task allocation from the global perspective, consider data factors, both local and link bandwidth for each task to find an optimal computing node, so as to achieve the whole job of global optimal scheduling

4.3. Contrast and Analysis of Local Data Rate

The Local data rate in the process of Sort and Wordcount test job execution, recording he number of the Local nodes and non-local nodes，according to the data obtained in Table 1 and Table 2, can draw out the homework of data rate of local line chart which are shown in Figure 7, and Figure 8:

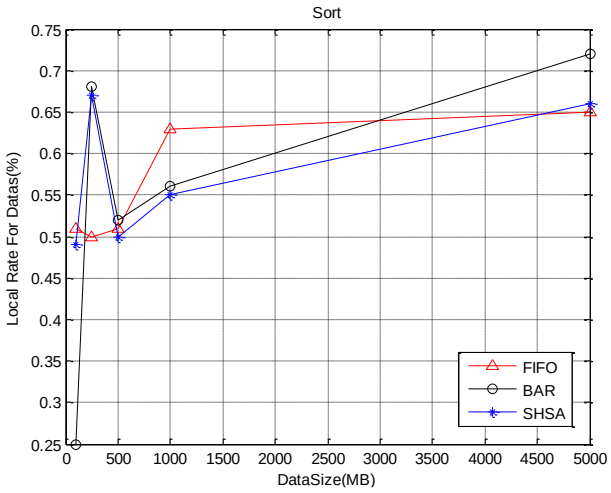

Figure 7. Local Rates of Sort Data

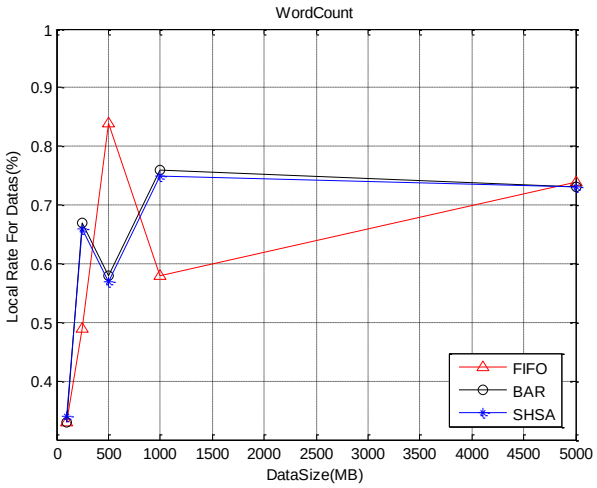

Figure 8. Local Rates of Wordcount Data

From Figure 7, and Figure 8, in most cases, the data rate is above 50%, it shows that local node is the optimal node, nodes with local data choose by task will have shorter completion time. But when operating data size is 500M, local rate of SHSA algorithm is 58.3%, lower than the FIFO and BAR algorithm, but Table 2 shows that the completion homework time only 229s, below the FIFO algorithm of the 269s and BAR in the 259s. Possible reasons for this phenomenon is that the local node task has queue length, and higher load, and network resources are idle, the migration of data to a local node does not need a long time. With this advantage the scheduler chose local node with more advantages for the processing of tasks, the total time is shorter.

5. Conclusion

In this paper, aiming at doing the task scheduling problem in large data platform, combined with the Hadoop open source big data system, a dynamic scheduling algorithm is designed and using global resource management ability of SDN, according to the time slot strategy to allocate bandwidth, and thus confirm the task allocation method, which can obtain higher global performance, better avoid falling into local optimum. On the cluster in build Hadoop test, with the Sort and Wordcount algorithm to test different sizes of data, Two groups of test results verified the SHEA ,FIFO and BAR algorithms which are proposed in this paper have better performance ,and also verify the quality of scheduling algorithm.

Acknowledgements

This work was supported by the Program of Science Foundation of Hubei Province (2014CFB576). and Zhejiang Province Natural Science Foundation (LY13F020023) and Science Foundation of Hubei Province (z2013016). and Scientific research program Funded By The public security Department of Hubei Province (hbst2014yycx03).

References

[1] Q. Li, Q. F. Hao, L. M. Xiao and Z. J. Li, “Adaptive Management adn Multi-Objective Optimization for Virtual Machine Placement In Cloud Computing”, Chinese Journal of Computers, vol. 34, no. 12, **(2011)**, pp. 2253-2264.

[2] L. X. XIAO, “Research on the Optimization of Enrollment Data resources Based on Cloud Computong Platform”, International Information and Engineering Technology Association, vol. 2, no. 2, **(2015)**, pp. 9-12.

[3] L. Hu, H. Jin and X. Liao, "Magnet: A novel scheduling policy for power reduction in cluster with virtual machines", Proceedings of the 2008 IEEE International Conference on Cluster Computing, Japan, **(2008)**, pp. 13-22,
[4] S. Narayan, S. Bailey, M. Greenway, R. Grossman, A. Heath, R. Powell and A. Daga, "Openflow enabled hadoop over local and wide area clusters", in Proceedings of the 2012 SC Companion: High Performance Computing, Networking, Storage and Analysis (SCC), Salt Lake City, Utah, USA, **(2012)** November, pp. 1625–1628.
[5] S. Narayan, S. Bailey and A. Daga, "Hadoop acceleration in an openflow-based cluster", in Proceedings of the 2012 SC Companion: High Performance Computing, Networking, Storage and Analysis(SCC), Salt Lake City, Utah, USA, **(2012)** November, pp. 535–538.
[6] A. Emmanuelle, D. Xavier, G. Maria and R. Matthieu, "Towards a theory of self-organization", Lecture Notes in Computer Science, v 3974 LNCS, p 191-205, 2006
[7] A. A. Jordi, F. A. Antonio, A. M. Miguel, T. Christopher and W. Lin, "Power-efficient assignment of virtual machines to physical machines", Future Generation Computer Systems, vol. 54, **(2016)** January 1, pp. 82-94.
[8] S. Mehdi, M. W. Richard, G. Lucio, V. P. J. Luis and G. Francesca, "A multi-dimensional job scheduling", Future Generation Computer Systems, vol. 54, **(2016)** January 1, pp. 123-131.
[9] F.Palmieri, U. Fiore, S. Ricciardi and A. Castiglione, "GRASP-based resource re-optimization for effective big data access in federated clouds", Future Generation Computer Systems, vol. 54, **(2016)** January, pp. 168-179.
[10] K. Maheshwari, E. S. Jung, Jiayuan Meng, Vitali Morozov, Venkatram Vishwanath and Rajkumar Kettimuthu, "Workflow performance improvement using model-based scheduling over multiple clusters and clouds", Future Generation Computer Systems, vol. 54, **(2016)** January, pp. 206-218.
[11] Y. Q. Fang, D. H. Tang and J. W. Ge, "Research on Schedule Strategy Based on Dynamic Migration of Virtual Machines in Cloud Environment", Microelectronics & Computer, vol. 29, no. 4, **(2012)**, pp. 45-48.
[12] Z. S. Yan, C. J. Zhang and M.Ying, "Study on regression analysis and prediction of time-series data stream using sliding windows", Journal of Natural Science of Heilongjiang University, vol. 23, no. 6, **(2006)**, pp. 865-867.
[13] Y. LI , D. D. WEI, Z. MU, Z. XIONG, Y. WANG and W. YIN, "Study of The Time-Collocation of Signal Lamp at Intersection", Mathematical Modelling of Engineering Problems, vol. 2, no. 1, **(2015)**, pp. 13-16.
[14] N. Mckeownn, T. Anderson and H. Balakrishna, "Open-Flow: Enabling Innovation in Campus Networks", ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, **(2008)**, pp. 69-75.

Authors

Zhe Li, was born on Jan, 1986, in Hanchuan City Hubei Province, PhD of Department of Automatic Control, Northwestern Polytechnical University（NWPU）Work at the Hubei Engineering University, lecturer.Major in: Consistency control of multi-agent in complex environment, Multi-Rate with multi-agent systems and Distributing control system. E-mail: lizhe_hbeu@vip.163.com