Cloud resource scheduling research based on intelligent computing

Xianquan Zeng*

School of Information Engineering, Xuchang University, Xuchang City, Henan Province, China, 461000

Received 1 October 2014, www.cmnt.lv

Abstract

Load balancing is an important problem of cloud computing. Due to the particularity of cloud computing, it puts forward higher requirements for load balancing. Thus a kind of load balance scheduling model of cloud computing based on improved ant colony algorithm is put forward. A new pheromone update strategy is proposed for the traditional ant colony algorithm. Through the improvement strategy, load balance scheduling algorithm of cloud computing resources based on ant colony algorithm is more in line with the characteristics and requirements of cloud computing. The experiment result shows that performance of load balancing degree and time efficiency of the proposed algorithm are better than the performance of genetic algorithm and traditional ant colony algorithm.

Keywords: cloud resource scheduling, ant colony algorithm, load balancing degree time span

1 Introduction

Cloud Computing as a new business model and calculation model, it is changing the traditional pattern of network services, and at the same time it is also changing the way that people use the computer network and computer [1, 2]. Cloud computing can integrate the storage resources, computing resources and software services, which can be provided to customers in a cheaper and high speed way to realize the separation of resources management and use. Because computing centre itself has large scale under the cloud computing environment, as well as the possible heterogeneity among computing resources, cloud computing has the phenomenon of uneven load easily, which seriously affects the overall performance and the user experience of the cloud computing system [3, 4].

In the cloud computing, tasks and resources scheduling have important effect on the overall performance. When performing massive parallel tasks, load unbalance of the nodes is easy to occur, making the performance of the whole cloud computing system degradation and inefficiency [5-8]. An Approach to optimized resource scheduling algorithm for open-source cloud systems was proposed by Hai, Z [9]. Environment-conscious scheduling method on distributed cloud-oriented data centres was proposed by Garg [10]. A game theoretic formulation of the service provisioning problem in cloud systems was proposed by Ardagna [11]. Some intelligent computing algorithms are used in resources scheduling of cloud computing [12]. A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation was proposed by Zhang, Z [13]. A penalty-based genetic algorithm for the composite SaaS placement problem in the cloud was proposed by Zeratul [14]. A model of virtual resource scheduling in cloud computing and its solution using EDAs was given by Jianfeng Zhao [15]. A distributed QoS-Constraint task scheduling scheme in cloud computing environment was given by Bing Li [16]. A novel fault-tolerant scheduling algorithm with high reliability in cloud computing systems was written by Yun Ling [17]. Study on vehicle scheduling algorithms for logistics operations based on cloud computing and neural network was given by Jie XIE [18]. Research on regional electronic health records data centre based on cloud computing was proposed by Jun Liang [19]. Trust-drive minimizes cost within deadline algorithm for instance-intensive workflows in cloud computing environment was given by Li Daoguo [20]. An Improved PSO based task scheduling algorithm for cloud storage system was given by Wang Juan [21]. A semantics constrained net based on CPN for cloud workflow modelling was given by Zhu jingyi [22]. Introducing new services in cloud computing environment was given by Anirban Kundu [23]. A Group tracing and filtering tree for reset DDOS in cloud was given by Lin Fan [24]. Resource sharing models and heuristic load balancing methods for grid scheduling problems was given by Wanneng Shu [25]. Study on dynamic resource scheduling of collaborative product development Process was given by Li Yingzi [26]. High-precision GM (1,1) model based on genetic algorithm optimization was given by Young'un Yang [27]. There are many other intelligent algorithms [28-32], which can be used in cloud resource scheduling. Support sector machine based particle swarm optimization localization algorithm in WSN was written by Tao Tang. Energy-aware and revenue-enhancing combinatorial scheduling in virtualized of cloud datacentre was proposed by Zhiming Wang.

* ***Corresponding author's*** e-mail: xianquanzeng@hotmail.com

The paper is organized as follows. In the next section, scheduling model of cloud computing is proposed. In Section 3, scheduling scheme based on ant colony algorithm is proposed. In Section 4, in order to test the performance of proposed scheme, experiment is carried out and the proposed algorithm is compared it with the other two algorithms. Finally, some conclusions are given.

2 Scheduling model of cloud computing

In cloud computing, the system is usually divided into three layer structure as shown in Figure 1, which includes the user layer, the virtual layer and physical layer. Although there are many models of cloud computing, any hierarchical model is based on the three layer structure and scheduling model is built on the three layer structure.

The user layer [7-9] is the window that cloud computing shows services to the user and users apply for services and it is mainly responsible for collecting application from the users and submitting tasks to the virtual layer. Virtual layer represents all the visible resources, and it is responsible for the specific requirements of the user application, such as operating system of virtual machine, software service, etc. Virtual machine is responsible for receiving tasks submitted by the user. The tasks are queued and the tasks satisfying the operating conditions are sent to the physical layer. The physical layer is the actual server cluster. According to different requirements of the performance of user, one server may be running with multiple virtual machines. Server is responsible for performing tasks and submitting the results of the execution to the virtual machine. Virtual layer and physical layer are separated in general, the virtual machine does not correspond to a physical machine really, and the submitted task by virtual machine will be dealt with by a suitable physical machine.

Load model for virtual machine is as follows. Because custom-made service is different, the obtained processing ability, memory size, and width of bandwidth are different [11]. Here we only take the CPU load as definition of the load, and ignore the influence of main memory and bandwidth. Storage resource is relatively cheap according to user's customized resources, and insufficient memory is rare to turn out. Even if insufficient memory turns out, this problem can be alleviated by virtual memory, and the consumption is data transfer time between parts of main memory. The appearance of insufficient memory is not frequent, so this part of the time can be ignored. Bandwidth resource is set according to the user's custom. The total bandwidth resource of cloud computing platform is fixed. It is a small probability event that all users use their bandwidth resource at the same time to send or upload information. This problem can be solved by encouraging users to use bandwidth resources in the period except high peak, and by means of time-sharing service strategy.

The physical machine load is represented by the CPU occupancy rate. All the tasks are executed on the physical machine at last. The ultimate goal of load balancing is to achieve the effective use of physical machine to improve throughput, and reduce the average task execution time. We set two thresholds for the physical machine load. When the physical machine load is less than 20%, the physical machine is identified as in light load condition. When it is higher than 80%, it is considered to be in a state of overload. For nodes under a state of overload, pheromone model is adjusted to let overload node no longer receive new tasks. It does not send out load balance application, but reserves buffer. When nodes load is higher than 90%, the node will send out application of load balance. The object of this treatment is to reduce system overhead caused by load transfer. When load of overload node declines naturally, load migration does not occur.

3 Scheduling scheme based on ant colony algorithm

Ant colony algorithm is a kind of centralized scheduling algorithm, which requires the main node to collect load information of system to update pheromone table for the resource scheduling of new tasks. The use of ant colony algorithm can make the system scalable. When the system needs to increase capacity to join a new computing cluster, we just need to set the pheromone model without modifying algorithm itself. Ant colony algorithm can improve the performance of cluster system to some extent. Because it is the centralized algorithm, when the cluster is too big, the system information maintenance will consume large amounts of time and system resources, and scheduling of ant colony algorithm is more suitable for small cluster system. The distributed system can be divided into several subsystems, and the subsystem realizes centralized control. At the same time, main nodes of subsystems are used to collect load information of subsystem, and the main nodes of subsystems are used to realize the load balance of the whole system.

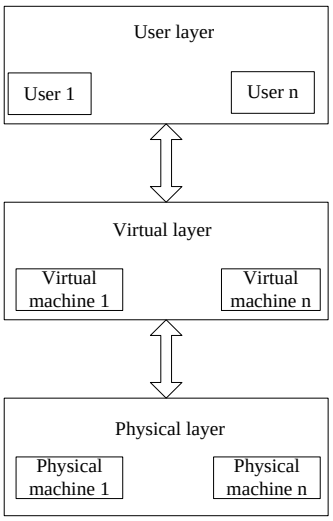

FIGURE 1 Architecture of cloud computing

The algorithm consists of the pheromone update and ant colony scheduling. The pheromone model is the basis of ant colony algorithm, which is mainly responsible for

the supervision of physical machine load state. Ants tend to the node with large pheromone, so that low load nodes can be given high pheromone values to guide the ant to tend to nodes with low load. Then load is guided to the nodes with low load. When there are scheduling demands in the system and node load is too high, ant colony algorithm is called for scheduling of system resources.

Cloud computing resources are discrete and unbalanced, and the real-time resource information is also changing. Ant pheromone model assign pheromone value to the point, and the size of the pheromone can describe real-time information of discrete resource point. A large number of information maintenance for discrete resources will inevitably need to consume large amounts of system resources. In order to reduce the burden of system and improve the rate and efficiency of scheduling, here we use a combination of global update and local update to update pheromone model.

Considering load change of the general node is gentle, load obvious change mainly occurs when a mission is scheduled to the node or a task is completed to exit. After a task is completed, local pheromone update is carried out for this node, until node load is relatively stable. After a long period, the entire pheromone model is updated globally. Every local update corresponds to only one node, thus system consumption is small, but it can improve the authenticity of the pheromone model. The object of global update is to avoid distortion of pheromone model, which is caused by state accumulation of unscheduled nodes in a long period of time. In order to avoid premature phenomenon, an improved ant colony algorithm is proposed. m represents the number of ants in the ant colony algorithm, d_{ij} represents the distance between node i and node j. $\eta_{ij} = \frac{1}{d_{ij}}$ represents visibility of path (i, j). $\tau_{ij}(t)$ represents amount of information of path (i, j) at time t. p_{ij}^k represents state transition probability of ant k between node i and node j.

$$p_{ij}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s \in allowed_k} \tau_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)}, & s \in allowed_k \\ 0, & otherwise \end{cases},$$

$allowed_k = \{0,1,\ldots,n-1\} - tabu_k$ represents the node that ant k selects in the next step. a represents information stimulating factor and β represents expected stimulating factor. $tabu_k$ represents the experienced node number of ant k at time t. It is not allowed that the ant passes the node of $tabu_k$ in one cycle. $tabu_k$ is empty, when one cycle is completed. Pheromone update for each path is as follows.

$$\tau_{ij}(t+n) = (1-\rho)\cdot\tau_{ij}(t) + \Delta\tau_{ij},$$

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^k.$$

In order to avoid premature phenomenon, τ_{ij} is limited to $[t_{\min}, t_{\max}]$, which is modified by the ant with the shortest path in one cycle. Modification strategy is carried out after pheromone update. The shortest path $\tau_{ij}(t)$ is determined by the following Equation.

$$\tau_{ij}(t+n) = \begin{cases} t_{\min}, & \tau_{ij}(t) \le t_{\min} \\ \tau_{ij}(t), & t_{\min} \le \tau_{ij}(t) \le t_{\max} \\ t_{\max} & \tau_{ij}(t) > t_{\max} \end{cases},$$

$$t_{\max(t)} = \frac{1}{1-\rho}\cdot\frac{1}{J^{opt(t-1)}}\cdot m,$$

$$t_{\min(t)} = \frac{t_{\max(t)}(1-\sqrt[n]{pbest})}{(avg-1)\sqrt[n]{pbest}}.$$

$avg = n/2$, and n represents the number of node. J^{opt} represents global optimal solution. $pbest \in (0,1)$ is one control parameter which adjusts pheromone dynamically.

Pseudo code of load balancing scheduling algorithm based on improved ant colony algorithm is as follows.

Set parameters, and initialize pheromone distribution.
While (application exists and the average load is less than 70%) do.
For (the ants in the ant colony) do
While (it does not meet the stop condition) do
Choose one node according to selection probability;
End of while
End of for
If (the scheduling is successful, meaning overload of the objective node does not occur after scheduling) do
Execute the task or migrate the virtual machine
Wait (10s);
Update pheromone according to node load found by ant, which is local update.
End if
End of while

Scheduling algorithm will be closed when system load is more than 70% and the system do not accept scheduling application any longer. In the selection probability, if the pheromone of a node is 0, selected probability of this node is 0, meaning that this node is not involved in scheduling. The pheromone update is as follows.

While (it meets local update condition) do // scheduling is successful
Wait (10s);
Update pheromone of the node;
End of While
If (it meets global update condition)//update cycle
For (node with concentrated resources) do

Global pheromone is updated;
End of for
End of if

4 Experiment and analysis

C++ program is written to simulate the algorithm under the environment of Visual Studio2010. Program runs on a Intel(R) Core(TM) 2 DuoT6670 processor of 2.20 GHz, and runs under the Windows 7 operating system. Here we only verify load balancing degree of the algorithm under the environment of heterogeneous multiple tasks and multiple service nodes. l represents load balancing degree, j represents the number of resources, x_j represents load value of resource n_j, and q represents average value of resource load.

$$l = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(x_j - q)^2},$$

$$q = \frac{1}{n}\sum_{j=1}^{n} x_j .$$

Time span is $T = T_C - T_s$, which includes scheduling time and execution time of the task. T_C represents end time of the task and T_S represents starting time of the task. There are 200 numbers of heterogeneous resource nodes and 9 groups of heterogeneous tasks. The span of which is 100. Heterogeneous degree of resource nodes and tasks are 10, which are randomly generated from 1 to 10. The greater the value of heterogeneous degree, the stronger the heterogeneity of nodes or tasks. A task of heterogeneous degree of 5 represents that the size of task is 5 unit task, and a resource node of heterogeneous degree of 5 represents that the processing capacity of this node is 5 unit task per unit time, meaning it can handle five unit task at the same time.

TABLE 1 Load balancing degree comparison

	200	300	400	500	600	700	800
GA	40.68	46.39	52.52	58.81	65.63	70.39	78.7
AC	22.26	30.78	37.82	50.23	59.1	64.28	72.67
IAC	21.15	28.26	35.01	47.34	56.06	61.91	68.51

TABLE 2 Time span comparison

	200	300	400	500	600	700	800
GA	1086	1297	1508	1685	1807	2042	2265
AC	889	1102	1350	1470	1601	1720	1810
IAC	869	1005	1267	1355	1437	1536	1726

Table 1 shows load balance degree of genetic algorithm, ant colony algorithm and improved ant colony algorithm. It can be seen that load balancing degree of the improved ant colony algorithm is better than genetic algorithm and traditional ant colony algorithm. Table 2 shows time span of genetic algorithm, ant colony algorithm and improved ant colony algorithm. Load balancing scheduling of genetic algorithm needs a process of learning step by step, the load balancing degree is worse than ant colony algorithm. Besides, the improved ant colony algorithm has faster time span than genetic algorithm and common ant colony algorithm.

Another experiment is done to test the performance of improved ant colony algorithm. The traditional ant colony algorithm and the improved ant colony algorithm are used in data clustering. The adopted data set is Iris data set, which has three kinds of data. The initial effect of Iris data set is shown in Figure 2 and three kind of colour are used to represent three kinds of data.

Clustering result of traditional ant colony is shown in Figure 3 and clustering result of traditional ant colony is shown in Figure 4. The horizon axis represents column and the vertical axis represents row. It can be seen that the clustering effect of proposed ant colony algorithm is better than the traditional ant colony algorithm.

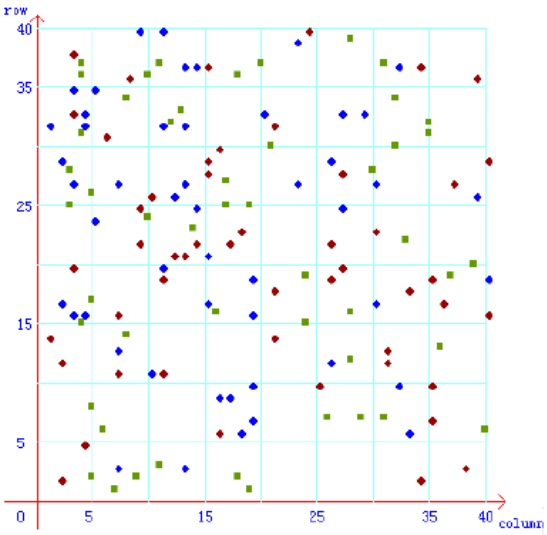

FIGURE 2 The initial effect of Iris data set

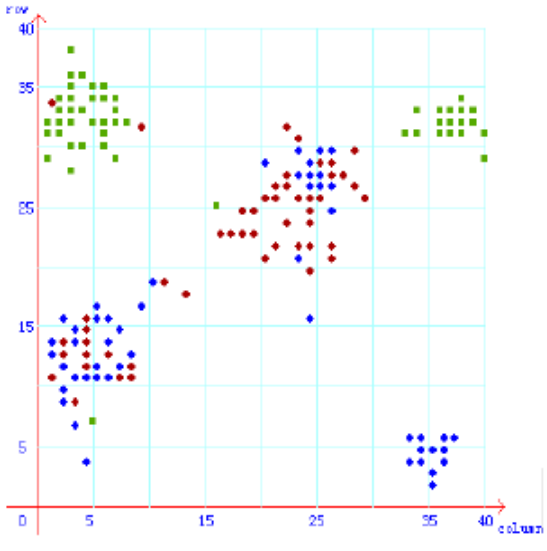

FIGURE 3 Clustering result of traditional ant colony

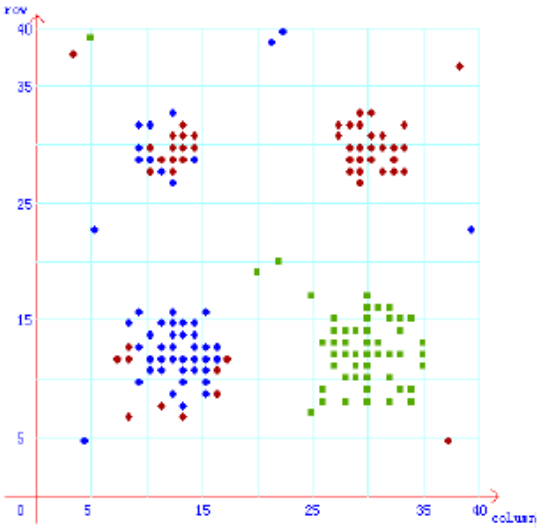

FIGURE 4 Clustering result of improved ant colony

5 Conclusions

According to the characteristics of cloud computing, combining with the characteristics of ant colony algorithm and operational mechanism, load balance scheduling model of cloud computing is proposed based on improved ant colony algorithm. Verification experiment is designed in the paper, showing that the load balance performance and efficiency of the proposed algorithm is better than genetic algorithm and traditional ant colony algorithm.

References

[1] Buyyaa R, Yeo Acs, Venugopals S 2009 Cloud computing and emerging IT platforms:vision, hype, and reality for delivering computing as the 5th utility *Future Generation Computer Systems* **25**(8) 599-616

[2] Vaquero LM, Rodero-Merino L, Caceres J 2008 A break in the clouds: towards a cloud definition. *SIGCOMM Computer Communication Review* **39**(1) 50-5

[3] Buyya R, Chee Shin Y, Venugopal S 2008 Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities *10th IEEE Conference on High Performance Computing&Communications* 5-13

[4] Kim K H, Beloglazov A, Buyya R 2011 Power-aware provisioning of virtual machines for real-time Cloud services *Concurrency and Computation: Practice and Experience* **23**(13) 1491-505

[5] Wang L, Laszewski G, Dayal J 2010 Towards Energy Aware Scheduling for Precedence Constrained Parallel Tasks in a Cluster with DVFS *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing* 368-77

[6] Von Laszewski G, Lizhe W, Younge A J, et al 2009 Power-aware scheduling of virtual machines in DVFS-enabled clusters *IEEE International Conference on Cluster Computing and Workshops* 1-10

[7] Kejiang Y, Xiaohong J, Dawei H,et al 2011 Live Migration of Multiple Virtual Machines with Resource Reservation in Cloud Computing Environments *IEEE International Conference on Cloud Computing (CLOUD)* 267-74

[8] Hien Nguyen V, Tran F D, Menaud J M 2010 Performance and Power Management for Cloud Infrastructures *IEEE 3rd International Conference on Cloud Computing (CLOUD)* 329-36

[9] Hai Z, Kun T, Xuejie Z 2010 An Approach to Optimized Resource Scheduling Algorithm for Open-Source Cloud Systems *Fifth Annual China Grid Conference (ChinaGrid)* 124-9

[10] Garg S K, Yeo C S, Anandasivam A, et al 2011 Environment-conscious scheduling of HPC applications on distributed Cloud-oriented data centers *Journal of Parallel Distributed Computing* **71**(6) 732-49

[11] Ardagna D, Panicucci B, Passacantando M 2011 A game theoretic formulation of the service provisioning problem in cloud systems *Proceedings of the 20th international conference on World wide web* 177-86

[12] Zeratul Izzah Mohd Yusoh, Maolin Tang 2012 Composite SaaS Placement and Resource Optimization in Cloud Computing using Evolutionary Algorithms *IEEE Fifth International Conference on Cloud Computing* 590-7

[13] Zhang Z, Zhang X 2010 A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation *2nd International Conference on Industrial Mechatronics and Automation (ICIMA)* 240-3

[14] Zeratul Izzah Mohd Yusoh, Maolin Tang 2010 A penalty-based genetic algorithm for the composite SaaS placement problem in the cloud *IEEE World Congress on Computational Intelligence* 600-7

[15] Zhao J, Zeng W, Liu M, Li G 2012 A model of Virtual Resource Scheduling in Cloud Computing and Its Solution using EDAs *JDCTA: International Journal of Digital Content Technology and its Applications* **6**(4) 102-13

[16] Bing L, Song A M, Song J 2012 A Distributed QoS-Constraint Task Scheduling Scheme in Cloud Computing Environment: Model and Algorithm *AISS: Advances in Information Sciences and Service Sciences* **4**(5) 283-91

[17] Liang Y, Ouyang Y, Luo Z 2012 A Novel Fault-tolerant Scheduling Algorithm with High Reliability in Cloud Computing Systems *JCIT: Journal of Convergence Information Technology* **7**(15) 107-15

[18] Xie J 2012 Study on Vehicle Scheduling Algorithms for Logistics Operations Based on Cloud Computing and Neural Network *AISS: Advances in Information Sciences and Service Sciences* **4**(9) 190-6

[19] Liang J, Sun TX, Xue MF, Ji Z, Zhang LZ, Li BL 2012 Research on Regional Electronic Health Records Data Center Based on Cloud Computing *IJACT: International Journal of Advancements in Computing Technology* **4**(11) 389-97

[20] Li D, Wang M, Zhao R 2012 Trust-drive Minimize Cost Within Deadline Algorithm For Instance-Intensive Workflows in Cloud Computing Environment *JCIT: Journal of Convergence Information Technology* **7**(13) 412-9

[21] Wang J, Li F, Chen A 2012 An Improved PSO based Task Scheduling Algorithm for Cloud Storage System *AISS: Advances in Information Sciences and Service Sciences* **4**(18) 465-71

[22] Zhu J 2011 A Semantics Constrained Net based on CPN for Cloud Workflow Modelling *IJACT: International Journal of Advancements in Computing Technology* **3**(7) 31-7

[23] Kundu Anirban, Banerjee Chandan, Priya Saha 2010 Introducing New Services in Cloud Computing Environment *JDCTA: International Journal of Digital Content Technology and its Applications* **4**(5) 143-52

[24] Lin F, Zeng W, Jiang Y, Li J, Liang Q 2010 A Group Tracing and Filtering Tree for REST DDos in Cloud *JDCTA: International Journal of Digital Content Technology and its Applications* **4**(9) 212-24

[25] Shu W, Ding L, Wang S 2012 Resource Sharing Models and Heuristic Load Balancing Methods for Grid Scheduling Problems *IJACT: International Journal of Advancements in Computing Technology* **4**(9) 315-22

[26] Li Y, Li C, Zhang S, Wang A 2012 Study on Dynamic Resource Scheduling of Collaborative Product Development Process *IJACT:*

International Journal of Advancements in Computing Technology **4**(19) 671-80
[27] Yang Y 2012 High-precision GM (1,1) Model Based on Genetic Algorithm Optimization *AISS: Advances in Information Sciences and Service Sciences* **4**(7) 223-30
[28] Liu J, Wang J, Zheng Y, Yao Y, Liu Z 2012 Annealing genetic algorithm for protein folding simulations in the 3D HP model *JDCTA: International Journal of Digital Content Technology and its Applications* **6**(9) 219-26
[29] Shen D, Li Y, Wei B, Xia X 2012 Adaptive Forking Multipopulation Differential Evolution Algorithm for Multimodal Optimization *JCIT: Journal of Convergence Information Technology* **7**(5) 57-65
[30] Guo Z-F 2012 A Novel Gravitation Search Algorithm and Different Evolution for Global Optimization *IJACT: International Journal of Advancements in Computing Technology* **4**(13) 261-8
[31] Liu Y, Zhang L, Zhao Y 2012 Calculation of Static Voltage Stability Margin Based on Continuous Power Flow and Improved Differential Evolution Algorithm *JDCTA: International Journal of Digital Content Technology and its Applications* **6**(17) 86-95
[32] Tang T, Guo Q, Yang M 2012 Support Vector Machine Based Particle Swarm Optimization Localization Algorithm in WSN *JCIT: Journal of Convergence Information Technology* **7**(1) 497-503
[33] Wang Z, Shuang K, Yang L, Yang F 2012 Energy-aware and revenue-enhancing Combinatorial Scheduling in Virtualized of Cloud Datacenter *JCIT: Journal of Convergence Information Technology* **7**(1) 62-70

Author

Xianquan Zeng, 1970.11, Xuchang County, Henan Province, China.

Current position, grades: the associate professor of School of Information Engineering, Xuchang University, China.
University studies: MSc in Computer Science and Technology from Xidian University in China.
Scientific interest: pervasive computing, software engineer.
Publications: more than 10 papers.
Experience: teaching experience of 20 years, 8 scientific research projects.