International Journal of Hybrid Information Technology
Vol. 9, No.11 (2016), pp. 137-146
http://dx.doi.org/10.14257/ijhit.2016.9.11.12

The Multi-objective VM Resource Scheduling by Using an Improved PSO Algorithm in the Cloud Data Center

Shengjun Xue[1], Jingyi Chen[2] and Mengying Li[3]

[1,2,3] *Nanjing University of Information Science & Technology, School of Computer and Software, Nanjing, China,Nanjing University of Information Science & Technology, Jiangsu Engineering Center of Network Monitoring, Nanjing, China*
[1] sjxue@163.com, [2] cjy_1225@163.com, [3] lmy3612@163.com

Abstract

With the exponential increase of the cloud business volume, Data center occurs load imbalance caused by some physical machine inefficiency due to the diversity of users requirements. Therefore the cloud datacenter need an appropriate algorithm to balance the PMs load and ensure the resource utilization in the cloud datacenter. The paper defines and formulates the problem parameters and proposes a Multi-objective Discrete Particle Swarm Optimization (MDPSO) to schedule the resources to the VMs requests according to the requirements. The simulation shows that the MDPSO algorithm not only guarantees the resource utilization, but also insures the PMs Load balance.

***Keywords**: cloud datacenter, resource, schedule, Multi-objective, Load balance*

1. Introduction

Due to the rate of business volume growth in the cloud service, the load in the cloud datacenter has been increased quickly. How to make good use of resource is the key point to reduce the unnecessary overhead. In the cloud datacenter, VT (Virtualization Technology) has been widely used. This technology expands the capacity of the hardware. Meanwhile the application programs won't impact to each other when they are running in the separate space in the same PM. VT also support the live migration, viz, when the PM overload, the VM could be move to another PM which is in the low load state . In general, VT not only enable the business attain the cost reduction but also the system reliability and scalability.

Cloud datacenter is the central of the resource. The resource mainly refers to the PM (Physical Machine), VM (Virtual Machine), the storage, the network, *etc*. Datacenter manage all the equipment. It should respond all the requests from the VMs. Different VMs have different priority, technology objects. Some of them request for the search service. Some are request for the computational or the storage resources. Therefore, an efficient resource scheduling is necessary for the properly work in the datacenter. The scheduler will find the resource in the datacenter according to the requests quality when the client requests for a service. The scheduler returns response to the client. Scheduler is required to give the response in the specified time. At the same time, the datacenter also need to consider the load balance, the utilization and the operating costs. The resource allocation mechanism determines the efficiency of the used resources and guarantees the Quality of Service (QoS) provided to the users. Hence, the resource scheduling mechanisms are considered as the critical technology in the cloud datacenter.

To satisfy multiple objectives, the scheduler has a strategy to schedule the resource. It's impossible to meet all the requests. So the manager determines the optimizing objects and execution policy when the resource is insufficient. The optimal placement of VMs within a data center network is a NP-hard problem. When large amount of VM data is migrated

ISSN: 1738-9968 IJHIT

from one physical host to another, The VM placement surely affects the data center network performance through the network[1][2][3]. The VM placement problems including the VM deployment and the dynamic VM resource management Here, the paper aim to achieve two objectives in our algorithm:

- VM deployment: Load balance (LB). To balance the resource utilization. This goal improves the datacenter performance. The rise of the cluster's processing capacity could enhance the user experience (UE) and the system reliability.
- Satisfy Multi-objects: Using MDPSO algorithm to find Pareto optimal solution to enhance the Resource Utilization and ensure the load balance.

The paper presents the design and implementation an improved PSO (Particle Swarm Optimization) deployment algorithm. We make the following contributions.

The paper proposes a new improved PSO algorithm which applied in the VM deployment to achieve the load balance and high resource utilization.

The paper tries to solve the multi-objective problem in the cloud datacenter based on a heuristic algorithm.

The paper simulates the algorithm and verifies the effectiveness of algorithm.

The rest of the paper is organized as follows. Section 2 presents the basic VM resource scheduling strategy and algorithm. Section 3 describes the details of the improved VM scheduling algorithm. Section 4 present the simulation and experiment results. Section 5 concludes and put forward the future research direction.

2. Related Work

Currently, there are several resources scheduling research direction: 1. The virtual machine placement [4]. 2. Data center energy consumption problem. 3. The economics of the cloud resource scheduling problem [5]. The strategy and algorithm of scheduling roughly divide into two categories: 1. high priorities of performance. 2. high priorities of cost. The traditional resource scheduling algorithm including Round Robin algorithm, Weighted Round Robin algorithm [6], Weighted Least-Connection Scheduling *etc*.

In terms of resource scheduling, domestic and overseas have done a lot of research work. The resources which Amazon provides to users can be divided into 8 classes. Users apply for rent according to their own business requirements. Data center scheduling algorithm give the feedback to the users based on the user characteristics, the resource type to find the right data center resources. Amazon's cloud resource scheduling combine some kind of strategies such as cost priority, meet the demand of different users, load balancing *etc*. [7][8][9]. BM's cloud computing platform built on the basis of virtual computing resources. IBM Tivoli completes the deployment of resources and cancellation of scheduled operations. IBM deployment management software Tivoli provisioning manager is responsible for monitoring the former provides IT resource health. Its scheduling policy is priority of performance and meeting the requirements of users[10]. Because of commercial confidentiality, Most of the technology is not known outside. The emergence of some open source cloud computing systems provides researchers with some excellent platform of cloud resource scheduling (Eucalyptus [11]). [12] Uses ucalyptus as the platform discuss the resource scheduling. The scheduling algorithm reads the log information which recorded in the load balancer in the real-time. The algorithm calculates the virtual machine average response time to determine the current system load. [13] systematic introduced the details of program and technology of HP data center cost model and become the important reference of cost model design. Most of the load balancing algorithm learn the traditional idea of load balancing Web server or server cluster and did not take into account the optimization of multiple objectives.

3. The Details of the Multi-Objective Algorithm

3.1. The Definition of the Problem Parameters

This subsection defines the structure of the datacenter. The structure quantifies the cloud datacenter resource for easy experimental comparison. So in this paper, the resource refers in particular to CPU, memory, the bandwidth. The paper simplifies the datacenter framework and presented the simplification PM cluster and VM cluster in the Figure 1.

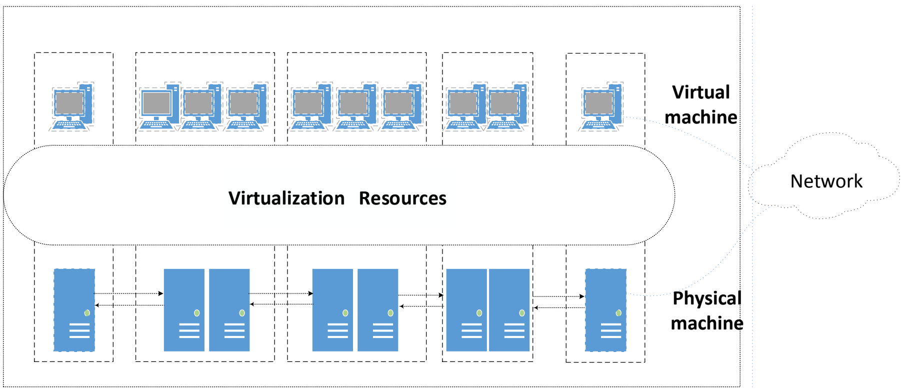

Figure 1. The Simplification PM and VM Cluster

The paper has some definitions about the problem parameter:

Definition 1: *PM set* α : $\{PM_i|0\leq i\leq n\}$, i denotes the number of the PMs.

Definition 2: VM set λ :$\{VM_j|0\leq j\leq n\}$, j denotes the number of the VMs.

Definition 3: *PM resource sets* φ: $\{PM_{ir}|0\leq r\leq n\}$, r denotes the categories of the virtualization resource.

Definition 4 : *PM Utilization* :

$$U_{pm_{ir}} = allocated_{pm_{ir}} / total_{pm_{ir}} \tag{1}$$

Which $allocated_{pm_{ir}}$ denote the used PM resource and the $total_{pm_{ir}}$ denote the total PM resource.

Definition 5: The PM average Utilization Uavg:

$$U_{avg} = \frac{\sum_{i=1}^{n} U_{pm_{ir}}}{n} \tag{2}$$

n denote the number of the PM.

Definition 6: The Specific Unbalance of resource r:

$$\mu_r = \frac{1}{n} \sum_{r=1}^{m} \frac{|U_{pm_{ir}} - U_{avg}|}{i} \tag{3}$$

Definition 7: The Specific Unbalance of Physical machine set:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} \sum_{r=1}^{m} \frac{|U_{pm_{ir}} - U_{avg}|}{i} \tag{4}$$

m denotes the categories of the resource and n denote the number of the PM.

The constraints condition are as following:

Constraint 1：The physical machines and virtual machine mapping is a one-to-many relationship. Which means one VM can only be placed on one PM and one PM can have many VMs. viz: $\sum_{i=1}^{n} D_{ij} = 1, j = [1,2,3\cdots m]$

Constraint 2: The VM resource requirement cannot exceed the PM resources capacity. Viz: $\sum_{i=1}^{n} VM_{ij} \le PM_j$, *i* denote the number of the VM request and j denote the categories of the requirement resource.

Here the definition 5 formulates the VM placement problem.

Definition 5: A VM request set $\lambda = \{VM_j|0\le j\le n\}$, and a set of PMs $\alpha = \{PM_i|0\le i\le n\}$, find out a placement solution, make the system in the most balanced load and meanwhile guarantee high utilization. *i.e*.

$$\begin{cases} \min \mu(0 < \mu < 1) \\ \max \dfrac{\sum U_{pm_{ir}}}{n} (0 < U_{pm_{ir}} < 1) \end{cases} \quad (5)$$

The problem is a NP-hardness problem, the tradeoff between the load balance and the Utilization should be considered.

3.2. The Improved PSO Scheduling Algorithm

PSO (Particle Swarm Optimization) algorithm is an efficient algorithm that have been used to optimize the different function [14][15][16]. It is easily implemented and usually converges faster than the Genetic Algorithm [17]. In the PSO algorithm, every individual can estimate the fitness value of the position by some rules and remember the best position of all the current finding position. Every individual is called “particle”. Every particle is a potential solution. For example: In a D dimensional objective searching space, every particle is seen as a dot. Assuming that the swarm is comprised of m particle and m cannot be too large or will affect the computational speed and convergence of the algorithm.

Supposing $x_i = (x_{i1}, x_{i2}, \cdots x_{iD})$ is *i*th particle's $(i = 1,2,\cdots m)$ D dimensional position vector $v_i = (v_{i1}, v_{i2}, \cdots, v_{id}, \cdots v_{iD})$ is the *i*th particle flight velocity. $p_i = (p_{i1}, p_{i2}, \cdots, p_{id}, \cdots, p_{iD})$ is noted as the best position by which it has ever visited. $p_g = (p_{g1}, p_{g2}, \cdots, p_{gd}, \cdots, p_{gD})$ is noted as the best position by which the swarm have ever visited. In basic PSO model, the particles are manipulated according to the equations (5):

$$\begin{aligned} v_{id}^{k+1} &= \omega v_{id}^{k} + c_1 \xi \left(p_{id}^{k} - x_{id}^{k} \right) + c_2 \eta \left(p_{gd}^{k} - x_{id}^{k} \right), \\ x_{id}^{k+1} &= x_{id}^{k} + \gamma v_{id}^{k+1} \end{aligned} \quad (6)$$

Where k expresses the kth iteration, c1 and c2 c are positive constants, called the cognitive coefficient and social coefficient respectively; ξ and η are random numbers uniformly distributed in the range (0~1). The parameter w is called inertia weight to balance the global and local search ability. γ is an operator to limit the velocity.

Here the algorithm supposed to use integer coding to define the particle's position, *e.g.* xi= (1, 3, 10, 2, 3, 2, 5) is one of the feasible solution. The algorithm firstly numbered the VMs and the PMs. In the feasible solution, “1” denote the first virtual machine would be placed in the first physical machine, “3” denote the second virtual machine would be placed in the third physical machine and by this analogy. But the basic PSO algorithm aimed at continuous value, so the algorithm uses a new Discrete PSO (DPSO) algorithm to solve the problem [18].

Set the particle's velocity $v_i = (v_{i1}, v_{i2}, \cdots, v_{id}, \cdots v_{iD})$, which $-v_{\max} \le v_i \le v_{\max}$, $|v_{\max}| = 4$, the sigmoid function to certify the probability of the VM_i will be placed into the PMj[19]. The sigmoid function is defined as following:

$$s = Sigmoid(v_{ik}) = \frac{1}{1 + \exp(-v_{ik})} \tag{7}$$

The particle movement equations are defined as following[18]:

$$\begin{aligned} V(t+1) &= \omega \otimes s(V(t)) \oplus R_1 \otimes [P_b(t) - X(t)] \oplus R_2 \otimes [P_g(t) - X(t)] \\ X(t+1) &= X(t) \oplus V(t+1) \end{aligned} \tag{8}$$

$\otimes$ denotes that counting the value of the product over half as one and disregarding the rest. The $\oplus$ is defined as following: $x \oplus y \begin{cases} x, if\ x = y \\ x, if\ s(x) > s(y) \\ y, else \end{cases}$. "–"denotes as following:

$$x - y \begin{cases} x, & if\ x = y \\ |x - y|, else & \end{cases}$$

3.3. Multi-objective DPSO VM Scheduling

To satisfy the resource utilization and load balance at the same time, the algorithm need to solve the multi-objective problem. The paper selects choose the Pareto optimal solution to solve the multi-objective optimization problem[20]. The algorithm guides the flight of the particles direction through every objective function together in Multi-Objective Optimization Problem to make particles fall into the Pareto optimal solution set. The pseudo-code algorithm is presented in Algorithm 1.

Algorithm1: The optimal solution evaluation selection algorithm

```
Input: A given population size N, every particle's position and velocity X_i, V_i. Output: the objective function
For i=1 to N,
   Fitness 1[i]=f1(X[i]);
   Fitness 2[i]=f2(X[i]);
   pBest[1,i] ← f1(x);
   pBest[2,i] ← f2 (x);
   gBest[1,i] ← f1(x);
   gBest[2,i] ← f2 (x);
   gBest = Average(gBest[1], gBest[2]);
   dgBest = Distance (gBest[1], gBest[2]);
For i=1 to N,
   dpBest[i] = Distance (pBest[1,i], pBest[2,i]);
For i=1 to N,
   if( dpBest[i]< dgBest)
      pBest[i] = Randselect( pBest[1,i], pBest[2,i])//random selection
   else
     pBest[i] = Average( pBest[1,i], pBest[2,i])//Assessment and selection
END
```

The Multi-objective DPSO VM scheduling algorithm is applied to the cloud environment resource scheduling model; set (5) equation as objective function, the pseudo-code for the high utilization and load balance of the VMs is presented in Algorithm 2. The algorithm change the particle's update formula and solve the multi-objective problem.

Algorithm2: Multi-objective DPSO VM scheduling algorithm

---

1 **Input**: A given population size, every particle's position and velocity *Xi* , *Vi*. **Output**: the objective function
2 choose the threshold value ε and maximum number of iterations N_{max}
3 Initialize the particle's position, Using integer form to define the position $Z_i^{(0)}=(Z_{i1},Z_{i2},\ldots Z_{in})$, which n is the PM's number. i is the particle's number.
4 Initialize the velocity of each particle $v_i^{(0)}=(v_{i1},v_{i2}\ldots v_{in})$
5 Calculate the particle's multi-objective fitness function (5), get the particle's fitness value $Z_{i1}^{(0)}$, $Z_{i2}^{(0)}$ denotes as $\tilde{D}^{(0)}$, $\tilde{A}^{(0)}$.
6 $p_{i1}^{(0)}=Z_{i1}^{(0)}$, $p_{i2}^{(0)}=Z_{i2}^{(0)}$. According to the $\tilde{D}^{(0)}=\min\{\tilde{D}_1^{(0)},\tilde{D}_2^{(0)},\cdots\tilde{D}_m^{(0)}\}$, $\tilde{A}^{(0)}=\max\{\tilde{A}_1^{(0)},\tilde{A}_2^{(0)},\cdots\tilde{A}_m^{(0)}\}$, find the global optimum p_{g1}^0, p_{g2}^0.
7 $k=0$
8 $k \leftarrow k+1$
9 Calculate fitness value of z_i, denote as $\tilde{D}_i^{(k)}$, $\tilde{A}_i^{(k)}$.
10 $\tilde{D}^{(k)}=\min\{\tilde{D}_1^{(k)},\tilde{D}_2^{(k)},\cdots\tilde{D}_m^{(k)}\}$, $\tilde{A}^{(k)}=\max\{\tilde{A}_1^{(k)},\tilde{A}_2^{(k)},\cdots\tilde{A}_m^{(k)}\}$
11 According to (7) equation to update the $p_{i1}^{(k)}$, $p_{i2}^{(k)}$ and p_{g1}^k, p_{g2}^k.
12 Calculate the global optimum p_g = *Average*(p_{g1}^k, p_{g2}^k);The Euler distance dp_g =*Distance*(p_{g1}^k, p_{g2}^k).
13 Calculate the Euler distance of every particle's local optimum dp_g^k.
14 ***if*** ($dp_g^k < dp_g$)
15 Choose the $p_{i1}^{(k)}$, $p_{i2}^{(k)}$ by the random selection.
16 ***else***
17 $p_i^{(k)}$ = *Average*($p_{i1}^{(k)}$, $p_{i2}^{(k)}$)
18 According to the (7) update $v_i^{(k)}, z_i^{(k)}$ by using $p_i^{(k)}$, p_g.
19 ***if*** $k > N_{max}$
20 END
21 ***else***
22 jump to **8**

---

4. Simulation Results

The simulation utilized CloudSim simulator for performance evaluation of the algorithm. The experiment modified the related methods and properties of VirtualMachine, VMAllocationPolicy, Datacenter class in the CloudSim. Using the Ant tool recompile CloudSim platform. The experience firstly random generate 50 PMs, and 100VMs. The 50 PMs are homogeneous. The PMs Set is as following:

Table 1.1. Physical Machine Parameter List

CPU/number	Computing Power/MIPS	Memory/GB	network bandwidth
16	3000	16	1000

The VMs types are as following Table:

Table 1.2. The Virtual Machine Application Type

CPU/number	Computing Power/MIPS	Memory/GB	network bandwidth
1	200	1	100
2	200	2	100
4	200	3.75	200
8	200	7.5	400

Every type has 25 requests and stochastic arrived. The population size is 30. The maximum iterations is 30.The learning factor is 0.5 and 0.5. The simulation compares the greedy algorithm (GA), the traditional PSO algorithm, and multi-objective DPSO algorithm (MDPSO). The simulation gets the results as Figure 2.

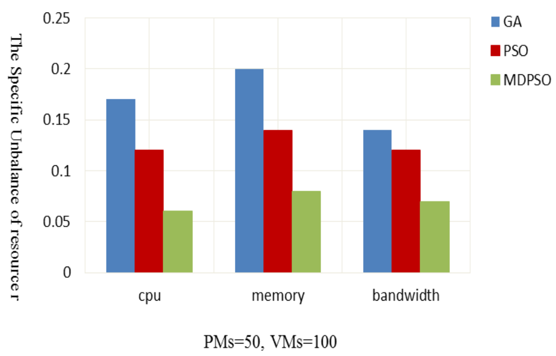

Figure 2. The Specific Unbalance of Different Resource by Three Algorithms

The results show that MDPSO algorithm makes the specific unbalance value minimum.

The simulation also compares the first 20 PMs utilization, the unit is percentage. In the Figures 3, 4, 5, we could find that the MDPSO not only guarantee the resource utilization, but also insure the PMs Load balance. Thereby the effectiveness of the new method is verified.

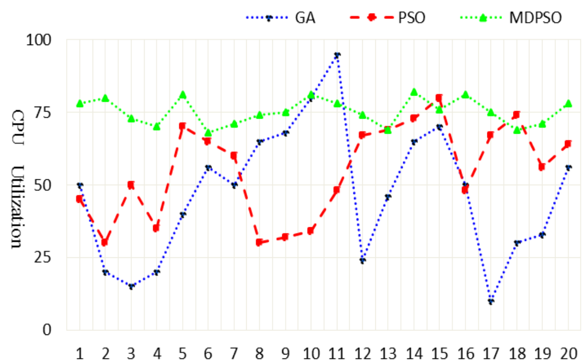

Figure 3. The CPU Utilization of Three Algorithms

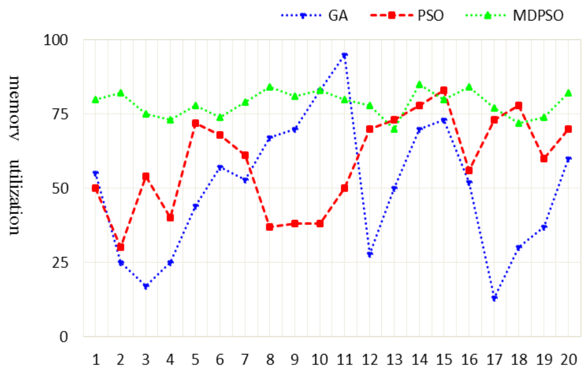

Figure 4. The Memory Utilization of Three Algorithms

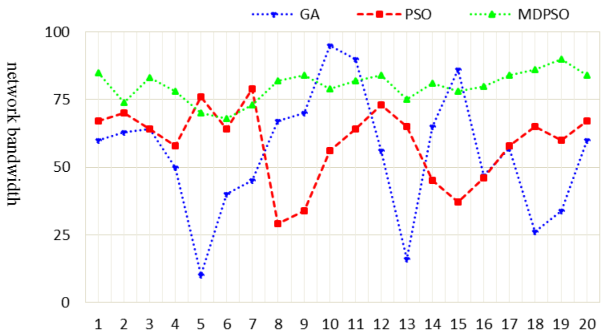

Figure 5. The Network Bandwidth Utilization of Three Algorithms

5. Conclusion

The paper proposed a Multi-objective DPSO algorithm to solve the VM resource allocation problem. We firstly discretize the particles value, and using a new particle movement equation to adapt to the cloud computing environment, and solve the multi-objective problem. In the paper, we choose the high utilization and load balance to be the objects, and two objects are incompatible. We utilize heuristic algorithm to find a Pareto optimal solution to solve the NP-hard problem in the cloud environment. The simulation experiments verify the effectiveness of the algorithm.

The research work is planned to be followed by the development of the cloud datacenter resource scheduler that supports the better efficient resource utilization for different objects. The future work we will focus on the implementation of the algorithm in the real cloud environment and the energy-aware of the VM scheduling to reduce the cost and implement the green cloud computing.

Acknowledgments

This work is partly supported by National Natural Science Foundation of China(Grand Nos.41275116), Jiangsu Economic and Information Technology Commission project of China (Grand Nos.{2011} 1178) and Open Fund Project of Jiangsu Engineering Center of Network Monitoring (Grand Nos.KJR1309).

References

[1] X. Meng, V. Pappas and L. Zhang, "Improving the scalability of datacenter networks with traffic-aware virtual machine placement", In: INFOCOM, 2010 Proceedings IEEE, **(2010)**, pp. 1–9.
[2] J. Shuja, K. Bilal and S A Madani, "Data center energy efficient resource scheduling", Cluster Computing, **(2014)**, pp. 1-13.
[3] W. Voorsluys, J. Broberg, S. Venugopal and R.Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation", In: Proceedings of the 1st International Conference on Cloud Computing. CloudCom '09, Springer-Verlag, Berlin, Heidelberg, **(2009)**, pp. 254–265
[4] P. Kokkinos and E. Varvarigos, "A framework for providing hard delay guarantees and user fairness in grid computing", Future Generation Computer Systems, vol. 25, no. 6, **(2009)**, pp. 674–686.
[5] D. Ergu, G. Kou, Y. Peng, Y. Shi and Y. Shi, "The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment", The Journal of Supercomputing, vol. 64, no. 3, **(2013)**, pp. 835-848.
[6] L. A. Barroso and U. Hölzle, "The datacenter as a computer: An introduction to thedesign of warehouse-scale machines", Synthesis Lectures on Computer Architecture, vol. 4, no. 1, **(2009)**, pp. 1-108.
[7] S. Ostermann, A. Iosup, N. Yigitbasi, R., Prodan, T. Fahringer and D. Epema, "A performance analysis of EC2 cloud computing services for scientific computing", In Cloud computing, Springer Berlin Heidelberg, pp. 115-131, **(2010)**.
[8] J. Liu, H. C. Kit, M. Hamdi and C. Y. Tsui, "Stable round-robin scheduling algorithms for high-performance input queued switches", High Performance Interconnects, **(2002)**, pp. 43-51.
[9] H. Hussain, S. U. R. Malik, A. Hameed, S. U. Khan, G. Bickler, N. Min-Allah and A. Rayes, "A survey on resource allocation in high performance distributed computing systems", Parallel Computing, vol. 39, no. 11, **(2013)**, pp. 709-736.
[10] M. A. Vouk, "Cloud Computing–Issues, Research and Implementations", Journal of Computing and Information Technology, no. 4, **(2008)**, pp. 235-246.
[11] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff and D. Zagorodnov, "The eucalyptus open-source cloud-computing system. In Cluster Computing and the Grid, 2009", CCGRID'09. 9th IEEE/ACM International Symposium on, IEEE, **(2009)**, pp. 124-131.
[12] E. Caron, F. Desprez, D. Loureiro and A. Muresan, "Cloud computing resource management through a grid middleware: A case study with DIET and eucalyptus", In Cloud Computing, 2009. CLOUD'09. IEEE International Conference on, IEEE, **(2009)**, pp. 151-154
[13] HP.HP datacenter. http://h20195.www2.hp.com/v2/GetPDF.aspx/4AA1-8079ENW.pdf.
[14] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", Proceedings of the sixth international symposium on micro machine and human science, vol. 1, **(1995)**, pp. 39-43.
[15] R. C. Eberhart and Y. Shi, "Evolving artificial neural networks", Proceedings of the International Conference on Neural Networks and Brain. **(1998)**, pp. 84-89.
[16] R. Eberhart, P. Simpson and R. Dobbins, "Computational intelligence PC tools", Academic Press Professional, Inc., **(1996)**.
[17] J. Kennedy, J. F. Kennedy and R. C. Eberhart, "Swarm intelligence", Morgan Kaufmann, **(2001)**.
[18] R. Zihui, "A discrete particle swarm optimization for solving multiple knapsack problems." Natural Computation", 2009. ICNC'09. Fifth International Conference on, IEEE, **(2009)**, vol. 3.
[19] Kennedy, James and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm", Systems, Man, and Cybernetics, (1997), Computational Cybernetics and Simulation, 1997 IEEE International Conference on, IEEE, vol. 5, **(1997)**.
[20] L. B. Zhang, C. G. Zhou, M. Ma and X. H Liu, "Solutions of Multi-Objective Optimization Problems Based on Particle Swarm Optimization", Journal of computer research and development, vol. 7, no. 41, **(2004)**, pp. 7.

Authors

Shengjun Xue, he earned his Ph.D. and MS degrees in computer application technology at Wuhan University of Technology in 2002. From 2007, he is a postdoctoral researcher at Purdue University. His major field of study is computer network, intelligent transportation and cloud computing. He has been worked in Nanjing University of Information Science& Technology since 2007. He has published more than 90papers in journals and conferences, and he has written 6 books. Now he is researching the cloud computing. Prof. Xue is the member of High Performance Computing and has served as the president of the international conference, such as RSETE, CSSS.

Jingyi Chen, she was born in Dongtai,China, in 1991. She is a master in Nanjing University of Information Science & Technology. Her research interests include cloud computing and meteorology.

Mengying Li, she was born in Nanjing, China, in 1992. She is a master in Nanjing University of Information Science& Technology. Her research interests include cloud computing and cloud scheduling.

