Multi-objective Optimization Model of Virtual Resources Scheduling Under Cloud Computing and It's Solution

Jianfeng Zhao, Wenhua Zeng, Min Liu, Guangming Li
Cognitive Science Department
Xiamen University
Xiamen, China
zjfhi2008g@gmail.com, whzeng@xmu.edu.cn

Jianfeng Zhao, Wenhua Zeng, Min Liu, Guangming Li
Fujian Key Laboratory of the Brain-like Intelligent Systems(Xiamen University)
Xiamen University
Xiamen, China
liumin_xt@163.com, mingming2033@163.com

Abstract—**It's an basic requirement in cloud computing that scheduling virtual resources to physical resources with balance load, however, the simple scheduling methods can not meet this requirement. This paper proposed a virtual resources scheduling model and solved it by advanced Non-dominated Sorting Genetic Algorithm II (NSGA II). This model was evaluated by balance load, virtual resources and physical resources were abstracted a lot of nodes with attributes based on analyzing the flow of virtual resources scheduling. NSGA II was employed to address this model and a new tree sorting algorithms was adopted to improve the efficiency of NSGA II. In experiment, verified the correctness of this model. Comparing with Random algorithm, Static algorithm and Rank algorithm by a lot of experiments, at least 1.06 and at most 40.25 speed-up of balance degree can be obtained by NSGA II.**

Keywords-cloud computing; virutal resources; physical resources; resources scheduling; NSGA II

I. INTRODUCTION

Cloud computing [1] is used to describe larger scale computing centre based on virtual technology, for example Amazon Elastic Compute Cloud (EC2) [2] is a classical cloud computing products. However, the exact definition of cloud computing is nebulous. Vaquero given a definite of cloud computing after researching 20 defines, Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services) [3]. Whatever the definite of Cloud computing, virtualization [4] play a key role in cloud computing. So the researching about virtualization becomes a hot topic.

In all researches about virtualization, the scheduling of virtual resource is an important topic. The scheduling of virtual resource points scheduling virtual resources to physical resources with balance. At present, the main methods of scheduling resources are the simple methods, such as the EC2 uses proprietary policy; VMware initializes placement on CPU load and dynamic placement to balance average CPU or memory load and consolidate servers; Nimbus uses static-greedy and round-robin resource selection policy; OpenNebula initializes placement based on requirement/rank policies to prioritize those resources more suitable for the virtual machine (VM) using dynamic [5]. However, as the scale of cloud computing rapidly increasing, the scheduling of virtual resources becomes more complex, more scholars are abstracted by this topic. Iyer *et al.* investigate the shared resource contention problem for virtual machines [6]. Costanzo *et al.* provided an execution environment for running applications on top of interconnected infrastructures [7]. Tian *et al.* proposed failure rules aware node resource provision policies for heterogeneous services consolidated in cloud computing infrastructure [8].

In this paper, we proposed a scheduling virtual resources model in cloud computing and solved it employing advanced NSGA II. In this model, the resources including virtual or physical are abstracted the nodes with a lot of attributes, in this situation, the scheduling becomes a marching problem. More than, proposed three multi-objective functions based on balance distribution. In experiment, verified the feasible of this model and compared NSGA II with Random algorithm, Static algorithm and Rank algorithm which are used in different cloud products.

II. THE MODEL OF SCHEDULING VIRTUAL RESOURCES

Cloud computing is a larger scale computing center or at least based on larger scale computing center, the difference of cloud computing with the traditional computing center is virtualization. As shown in figure 1, there is a layer called virtualization layer in cloud computing between users and physical layer. The advantages of virtualization are:

- **Usability.** As usual, the physical devices can only be used by one person at a specified time, so one hand the utilization ratio of devices is low, on other hand, some requirements can not be serviced. Using virtualization, the physical device can be shared by many people at the same time. Besides this, every user can configure their own running environment without interference.
- **Safety.** In virtual environment, every user's environments are isolated with each other, so other

National Natural Science Foundation of China (Grant No. 60903129, 60975084, 30900328).

978-1-4577-1637-9/11/$26.00 ©2011 IEEE

user's environments will not be affected when a user's program collaps, in addition to the user's permission is limited strictly.

- **Moving.** It's easy for virtual resources to move to new environment when the hardware is destroyed or upgraded.

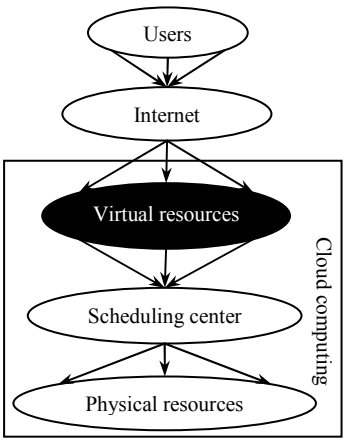

Figure 1. A cloud computer platform

All above advantages come from the independency of virtualization. When users send some requirements to cloud computing center, as shown in figure 1, the system generates some virtual resources as the respond instead of assigning to the concrete physical devices. Then the scheduling center will decide where the virtual machines should be assigned to. However, user's requirement is various and the physical devices are also diverse, therefore the scheduling problem become intractable. Inspired by the workflow scheduling in Grid environment, the resources can be abstracted to nodes with attributes. If the resources can be abstracted the nodes, then the scheduling process can be recognized as a nodes matching process. The formal describing is as following:

A virtual resource can be abstracted a node with a lot of attributes, therefore a node with attributes CPU, memory and bandwidth can be described formally as $v(c,m,b)$, where c is attribute CPU, m is attribute memory and b is bandwidth. Similarly the physical resource can be described using $p(c,m,b)$.

TABLE I. ABSTRACTED VIRTUAL RESOURCES AND PHYSICAL RESOURCES

	Virtual resources				Physical resources		
	cpu	***ram***	***bw***		***cpu***	***ram***	***bw***
v1	400	100	10	p1	1000	4000	100
v2	200	200	20	p2	500	3000	10
v3	1000	200	30	p3	2000	2000	50
v4	70	20	5	p4	200	500	20
v5	1500	2000	10	p5	600	600	50
v6	500	1000	60	p6	700	1000	100
v7	60	80	100				
v8	700	80	50				
v9	500	600	10				
v10	400	800	30				
total	5330	5080	325		5000	11100	330

Table 1 shows the abstracted virtual resources and physical resources, where include 10 virtual resources and 6 physical resources. Based on the above analyzing, this paper proposed a definition about the scheduling virtual resources.

Def. 1 Scheduling virtual resources is to find a solution which maps the virtual resources to physical resources. For virtual resources $v_i(c_{vi}, m_{vi}, b_{vi})$, $v_i = 1,2, \ldots ,m$ and physical resources $p_i(c_{pi}, m_{pi}, b_{pi})$, $p_i = 1, 2, \ldots ,n$, a scheduling is expressed by a serial including m physical resources $L(p_{i1},p_{i2},\ldots,p_{im})$, at this time, virtual resource c_{v1} is scheduled to physical resource p_{i1}, virtual resource c_{v2} is scheduled to physical resource p_{i2}, and so on.

Theorem 1 The problem of scheduling virtual resources is a hard problem using exhaustive algorithm to solve.

Proof If use exhaustive algorithm to solve the scheduling problem in cloud computing, then there are n choices for v_1, n choices for v_2, …, n choices for v_m, so there are n^m kind of rank for L, the time complexity is $O(n^m)$, depend on reference [9], this problem is hard problem.

III. THE SOLVING ALGORITHM OF SCHEDULING MODEL

As above analyses, it's a hard problem that solving the scheduling model. At present, the heuristic algorithms are known as the better algorithms to solve the hard problem. This paper takes heuristic algorithms to solve this problem. In heuristic algorithms, three parts are important: object functions, code and searching method. We will introduce those three parts separately.

A. Object functions

Load-balance is an important target for scheduling method in cloud computing, therefore inspired by the variance idea in mathematic, this paper quantify the resources load-balance using variance and proposed the object functions of different attributes.

For virtual resources $v_i(c_{vi}, m_{vi}, b_{vi})$, $v_i = 1,2, \ldots ,m$ and physical resources $p_i(c_{pi}, m_{pi}, b_{pi})$, $p_i = 1, 2, \ldots ,n$, a given solution is $(v_{a1}, v_{a2}, \ldots, v_{am})$, therefore:

- The balance degree of CPU

Set

$$\Gamma_{ctotal} = \frac{\sum_{i=1}^{m} c_{vi}}{\sum_{i}^{n} c_{pi}}$$

$$\Gamma_{ci} = \frac{\sum_{i=1}^{m} c_{va_i}}{c_{pi}} \quad \text{where,} \quad c_{va_i} = \begin{cases} c_{vi} & a_i = \text{i} \\ 0 & others \end{cases}$$

so, the balance degree of CPU is:

$$P_c = \sqrt{\sum_{i=1}^{m} (\Gamma_{ci} - \Gamma_{ctotal})^2}$$

Similarly,

- The balance degree of memory

$$P_m = \sqrt{\sum_{i=1}^{m} (\Gamma_{mi} - \Gamma_{mtotal})^2}$$

- The balance degree of bandwidth

$$P_b = \sqrt{\sum_{i=1}^{m} (\Gamma_{bi} - \Gamma_{btotal})^2}$$

B. Coding method

From the definition we know that the virtual resources scheduling is finding a scheduling serial $L(p_{i1},p_{i2},...,p_{im})$, so this paper uses the ID of physical resource as the code. Set the ID for virtual resources are $v_i=1, 2, ..., m$ and the ID for physical resources are $p_i=1, 2, ..., n$, so for the virtual resources ($v_1, v_2, ..., v_m$), the code is $(v_{a1}, v_{a2}, ..., v_{am})$, where, $v_{ai} \in \{p_i\}$, at this time, for virtual resource v_i, $i = 1, 2, ..., m$, it is scheduled to physical resource v_{ai}.

C. Searching method

Alg. 1 obtain non-dominated set

```
non-dominate set F travel_tree(Tree) {
        for (p=Tree; p!=null; p=p.left) {
                put p into the first non-dominated set F1;
        }
        put F1 into F;
        while(the number of individuals in F < half number of tree nodes){
                for(every individual p in Fi-1) {
                        for(pi = p.right; pi != null; pi=pi.left){
                                if(pi is not dominated by the
individuals in Fi){
                                        put pi into Fi;
                                }else{
                                        put pi into Left;
                                }
                        }
                        for(pi is in Left){
                                if(pi is not dominated by
individuals in Fi){
                                        put pi into Fi;
                                }
                                if(pi dominate anybody qi in Fi){
                                        put qi into Left;
                                }
                        }
                }
        }
}
```

This paper uses NSGA II [10] to search the best result. NSGA II is the classic algorithm in multi-objective optimization algorithms. The idea of NSGA II is similarly with Genetic Algorithm (GA), the difference is the selection. In NSGA II, the persons are chosen depending on the Pareto non-dominated sorting firstly and crowd degree secondly. Pareto non-dominated sorting is: set finding the lowest value of object functions, if for all objects exist A（B）<=B（A） and A not equal with B, then A(B) dominates B(A), otherwise, A and B is non-dominated. A lot of non-dominated solutions can be obtained through once running, however, it's time-consumer to solve the Pareto non-dominated solutions. For improving the performance of NSGA II, inspired by reference [11], this paper proposed a non-dominated solution set sorting algorithm to enhance the solving speed.

At first, sort people according with the first object function and give a ID for every individual, set 1, 2, …, n. Therefore, it's impossible that an individual be dominated by the individual after it. Then construct a binary tree employing binary tree sorting algorithm. At last, using algorithm 1 obtains the non-dominated sorting.

The whole flow chart of NSGA II is drawn in figure 2. At first, generate people P1 including n individuals and generate next generation people Q1 by selection, mutation; then, generate a binary tree on sorting PiUQi by employing binary sorting algorithm, travel binary tree using algorithm 1 to obtain a set and calculates the crowd degree on this set; at last, chose the individuals depending on the rule of non-dominated firstly and lower crowd degree, namely, if A dominates B, then chooses A, else A and B is non-dominated, then chooses the individual with lower crowd degree.

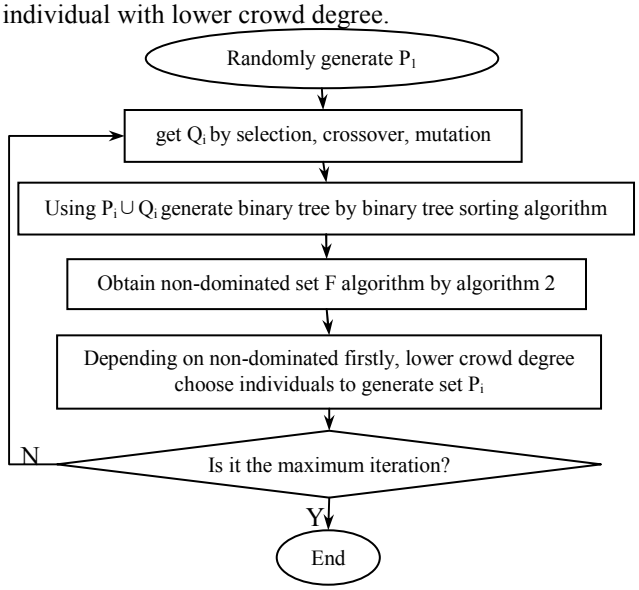

Figure 2. The whole flow chat of NSGA II algorithm

In flow chart, the selection is tournament selection and the crossover is two points crossover, that is, randomly chooses two individual and choose the better individual as the parent p_1, similarly chooses another individual as the parent p_2; then randomly generates two crossover points c_1, c_2 and exchanges the gene from c_1 to c_2 of p_1 with p_2, two new children are generated u_1, u_2. The mutation points mutating each gene of every individual based on mutation ration of Pc, namely, if the random number is lower than mutation ration of Pc, then the original gene is replaced by randomly generating a gene. The sorting method based on binary tree is employed, which is described in above paragraph. The crowd degree is calculated according with reference [10], that is, for each object function,

sorting the individuals depended on the value of object function, the first and last individual has the lowest crowd degree, others individual's crowd degree are the reciprocal of the distance of two neighbor individuals.

IV. EXPERIMENTS AND ANALYSIS

This experiment code is written in JAVA, the parameters are: people's number: 48, max-iteration: 1000, crossover factor: 0.9, mutation factor: 1/*l*, where *l* is the length of gene.

This paper compared NSGA II with other three algorithms mentioned in reference [5]: Random Algorithm, Static Algorithm and Rank Algorithm. For virtual resources set V, physical resources set P, three algorithms are briefly described as following:

- Random Algorithm

Randomly generate a scheduling solution $L(a_1, \dots, a_m)$

- Static Algorithm

The scheduling solution is $L(a_1, \dots, a_m)$, where, $a_i = i\%n$

- Rank Algorithm

The main idea of Rank Algorithm is: for a virtual resource, it will be scheduling to the closest physical resource. The pseudo-code is shown as following:

Alg. 2 Rank algorithm

```
Solution L Rank_sch(V,P) {
        L = ∅;
        for each vi belong to V {
                current=0;
                for each pi belong to P {
                        for each goals {
                                match_degree[i] += goals;
                        }
                        if (match_degree[current] > 0 &&
match_degree[i-1] > 0 && match_degree[current] > match_degree[i-1])
                                current = i;
                        else if (match_degree[current] < 0 &&
match_degree[i-1] < 0 && match_degree[current] < mathc_degree[i-1])
                                current = i;
                        else if (match_degree[current] < 0 &&
match_degree[i-1] > 0)
                                current = i;
                        end
                        L = L U current;
                }
        }
        return L;
}
```

A. Solve the example in this paper

One hand, we verify the correctness of model proposed in this paper, on the other hand, we compare NSGA II with other three algorithms by solving the example in section 2. Table 2 shows the solving results given by NSGA II. There are only 8 solutions are listed for the length limitation of paper.

From the table 2 we know that NSGA II can obtain a lot of non-dominated solutions by once running. Therefore this paper use four statics to choose the finally solution: CPU static, memory static, bandwidth static and balance static. In reality, users can choose different static based on their demands. CPU static is: choose the solution whose CPU loading is the most balance. Memory static and bandwidth static is similarly with CPU static, which memory and bandwidth is the first choice. Balance static is: choose the solution whose sum of object function value is minimal from all solutions. Table 3 lists the results with different statics. In specially, only one result can be given by other algorithms.

TABLE II. THE NON-DOMINATED SET OBTAINED BY NSGA II

Solutions	cpu balance degree	memory balance degree	bandwidth balance degree
2322001054	2.1768	0.5430	0.0486
2312552504	1.1316	1.6437	0.0486
5310522204	0.1605	1.2760	0.8411
0453130240	4.6868	0.2580	2.7494
1354230114	2.2351	0.1729	16.0825
1453120242	4.0816	0.2425	3.8281
0354201004	2.7826	0.3090	1.3767
0312052554	0.3886	1.0268	0.9052

TABLE III. COMPARED NSGA II WITH OTHER ALGORITHMS BY SOLVING THE EXAMPLE IN THIS PAPER

Algorithm	Static	Solution	CPU	Memory	Bandwidth
NSGA II	CPU strategy	2310252554	0.24	1.32	0.96
	memory strategy	1453230141	4.94	0.17	11.21
	bandwidth strategy	2312002054	1.07	0.89	0.05
	balance strategy	5312022054	0.51	0.82	0.17
Random	-	1320352225	2.24	2.23	3.28
Static	-	0123450123	7.60	1.00	7.84
Rank	-	3451132021	41.65	26.87	158.31

From the table 3 we know that different attributes are mutual influences, they are non-dominated. At the same time, users can choose different solution based on their preferences, however, other algorithms only give one solution. At the same time, table 3 tells us that the results of NSGA II are better than other algorithm whatever the statics.

Appendix 1 shows the concrete resource allocation result of the example in section 2 using different algorithms. Corresponding with table 3, there are 7 solution results and 21 resource allocation results due to each solution has 3 attributes. The row of CPU1 represents the CPU allocation situation of every physical node using the first solution in table 3, namely the solution of NSGA II with CPU strategy. Similarly, RAM1 and BW1 represent respectively the memory allocation result and bandwidth allocation result. The front 7 columns are the concrete value of allocation result, the end 7 columns are the percentage of the allocation resource with the physical resource on the physical nodes.

At first, the appendix tells us that the NSGA II can obtain different results depending on the different strategies. From

table I we know that Γ_{ctotal} = 1.066, Γ_{mtotal} = 0.46, Γ_{btotal} = 0.98. On the CPU strategy, the cpu distribution is most equal. It's similarly on others strategy. Another conclusion we can get is that NSGA II algorithm is better than other algorithms. For example, for the physical nodes p4, the cpu load is 1 using NSGA II, however, the Random algorithm is 3, the Static algorithm is 8.5 and the Rank algorithm is 2.35.

B. larger-scale data comparison

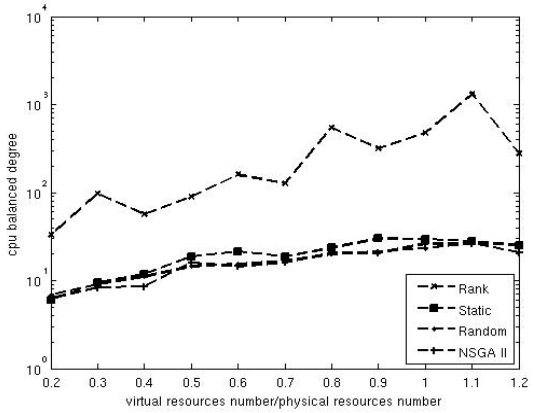

Figure 3. The variation of cpu balance degree as the ratio of virtual resources and physical resources

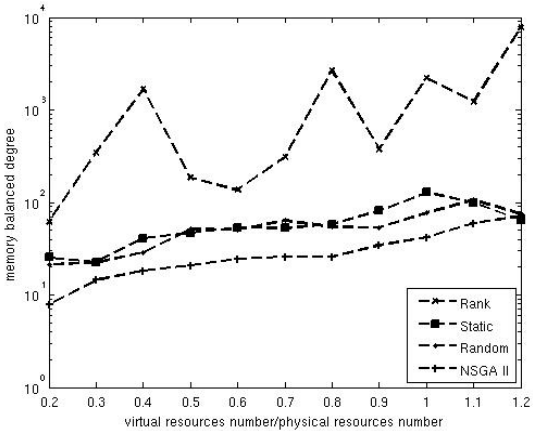

Figure 4. The variation of memory balance degree as the ratio of virtual resources and physical resources

For more detail comparing NSGA II with other algorithms, understanding the solutions of NSGA II with different resources, this paper solves the virtual resources scheduling problems with different ratio of virtual resources and physical resources. In experiment, randomly generates [100-1000] physical resources, then generates virtual resources depending on different ratio. Figure 3-5 shows the average results with balance static after running 100 times. Three figures display the CPU balance degree, memory balance degree and bandwidth balance degree.

From the figures we know that NSGA II is better than other algorithms, Random Algorithm and Static Algorithm are close each other, the Rank Algorithm is the worst. The reason is the Rank Algorithm schedules the virtual resource based on the matching rule, at this time, it's more possible that the virtual resources are assigned the same physical resources. In figure, the lowest speed-up is 1.06 in CPU balance degree and the largest speed-up is 40.25 in bandwidth.

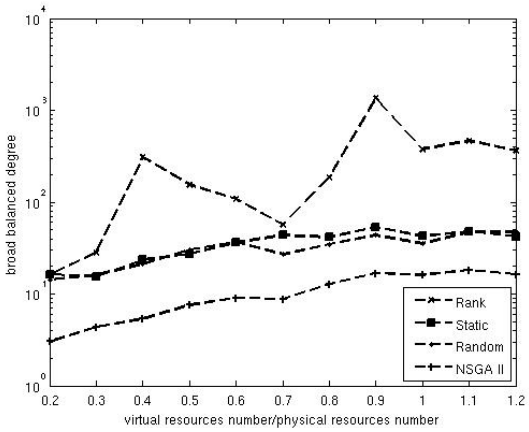

Figure 5. The variation of bandwidth balance degree as the ratio of virtual resources and physical resources

V. CONCLUSION

This paper proposed a model of virtual resources scheduling under cloud computing, in which the virtual resources and physical resources are abstracted a lot of nodes with attributes. The scheduling process of virtual resources is changed to a matching of virtual resources and physical resources. Using NSGA II to solve this model and improving the performance of NSGA II by introducing binary tree sorting. At last, this paper compares NSGA II with other three algorithms: Random Algorithm, Static Algorithm, Rank Algorithm. The experiment shows that NSGA II can be used to solve virtual resources scheduling problem and can obtain many choices by once running, users can choose different solutions based on different static. Comparing with other algorithms, the experiment results show that NSGA II is more suitable to schedule virtual resources.

ACKNOWLEDGMENT

This work was supported by the Fujian Key Laboratory of the Brain-like Intelligent Systems (Xiamen University), P.R.China.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, *et al*. "A view of cloud computing," Communications of the ACM, vol. 53, pp. 50-58, April 2010.

[2] Amazon. Amazon EC2. "http://www.amazon.com/ec2," 2011.

[3] L. M. Vaquero, L. Rodero-Merino, J. Caceres, M. Lindner. "A break in the clouds: towards a cloud definition," ACM Sigcomm, vol. 39, pp. 50-55, January 2009.

[4] M. Rosenblum, T. Carfinkel. "Virtual Machine Monitors: Current Technology and Future Trends Computer," Computer, vol. 38, pp. 39-47, May 2005.

[5] B. Sotomayor, R. S. Montero, I. M. Llorente, I. Foster. "Virtual Infrastructure Management in Private and Hybrid Clouds," Internet Computing IEEE, vol. 13, pp. 14-22, May 2009.

[6] R. Iyer, R. Illikkal, O. Tickoo, L. Zhao, P. Apparao, D. Newell. VM^3: Measuring, modeling and managing VM shared resources.. Computer Networks. vol. 53, pp. 2873–2887, Aguest 2009.

[7] A. d. Costanzo, M. D. d. Assunção, R. Buyya. "Harnessing Cloud Technologies for a Virtualized," Distributed Computing Infrastructure, vol. 13, pp. 24-33, Octobor 2009.

[8] G. Tian, D. Meng, J. Zhan. "Reliable Resource Provision Policy for Cloud Computing," Chinese Journal of computer, vol. 33, pp. 1859-1872, Octobor 2010.

[9] D. S. Hochbaum. "Approximation Algorithms for NP-Hard Problems," Boston, PWS Publishing Company, p. 23, 1997.

[10] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation. vol. 6, pp. 182-197, April 2002.

[11] C. Shi, Z. Yan, Z. Shi, L. Zhang. "A fast multi-objective evolutionary algorithm based on a tree structure," Applied Soft Computing, vol. 10, pp. 468–480, Feburary 2010.

Appendex 1 The concrete allocation result and load corresponding with the solutions in table III

	p1_v	p2_v	p3_v	p4_v	p5_v	p6_v	p1_%	p2_%	p3_%	p4_%	p5_%	p6_%
CPU1	1000	500	2300	200	700	630	1	1	1.15	1	1.17	0.9
RAM1	200	600	2900	200	80	1100	0.05	0.2	1.45	0.4	0.13	1.1
BW1	30	10	50	20	50	165	0.3	1	1	1	1	1.65
CPU2	1500	1600	400	1070	260	500	1.5	3.2	0.2	5.35	0.43	0.71
RAM2	2000	1180	800	220	280	600	0.5	0.39	0.4	0.44	0.47	0.6
BW2	10	120	30	35	120	10	0.1	12	0.6	1.75	2.4	0.1
CPU3	970	500	2900	200	700	60	0.97	1	1.45	1	1.17	0.086
RAM3	1820	600	2300	200	80	80	0.455	0.2	1.15	0.4	0.13	0.08
BW3	95	10	50	20	50	100	0.95	1	1	1	1	1
CPU4	900	500	2570	200	700	460	0.9	1	1.285	1	1.17	0.66
RAM4	1800	600	2220	200	80	180	0.45	0.2	1.11	0.4	0.13	0.18
BW4	90	10	45	20	50	110	0.9	1	0.9	1	1	1.1
CPU5	1000	400	2560	600	0	770	1	0.8	1.28	3	0	1.1
RAM5	200	100	3680	1000	0	100	0.05	0.03	1.84	2	0	0.1
BW5	30	10	180	50	0	55	0.3	1	3.6	2.5	0	0.55
CPU6	1900	700	560	1700	400	70	1.9	1.4	0.28	8.5	0.67	0.1
RAM6	2100	1200	680	280	800	20	0.525	0.4	0.34	0.56	1.33	0.02
BW6	20	80	110	80	30	5	0.2	8	2.2	4	0.6	0.05
CPU7	500	2100	1560	470	200	500	0.5	4.2	0.78	2.35	0.33	0.71
RAM7	1000	1080	2080	120	200	600	0.25	0.36	1.04	0.24	0.33	0.6
BW7	60	110	110	15	20	10	0.6	11	2.2	0.75	0.4	0.1