International Journal of Grid and Distributed Computing
Vol. 11, No. 3 (2018), pp.125-138
http://dx.doi.org/10.14257/ijgdc.2018.11.3.11

Energy and Utility-based Algorithm for Scheduling Virtual Machine in Could Computing

Zhu Wei[1] and Zhuang Yi[2]

[1,2]College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China,
[1]Jiangsu Automation Research Institute, Lianyungang 222006, China,
[1]zhuweipaper@126.com, [2]zy16@nuaa.edu.cn

Abstract

One of the pressing issues that the cloud data center needs to solve is how to improve its utility while reducing the energy consumption in the data center. This paper presents a virtual machine scheduling model based on multi-objective optimization VMSA-EU, which aims to minimize energy consumption and maximize the practicality of the data center. And proposed a virtual machine scheduling algorithm based on NSGAII to solve the model. Simulation results show that our models and algorithms can reduce energy consumption while improving utility. Compared with other similar existing algorithms, the algorithm has made some progress in execution time and scheduling results.

***Keywords**: cloud computing; energy; utility; virtual machine scheduling; multi-objective optimization*

1. Introduction

"Cloud" is the virtual computing resource which can realize self-management and self-maintenance, such as some large server clusters which including computing servers, storage servers, and bandwidth. Cloud computing takes all computing resources together and realizes automatic by software without human involvement. Cloud computing enables service providers to reduce costs and implement innovation[1] by eliminating the need to focus on technical implementation details by focusing on their own business. The goal of cloud computing is to deliver all kinds of IT resources to users as services via Internet. Virtualization technology enables the abstract and unified management of IT resources, which plays a very important role in the management and solution of large data centers and is the most important technical basis for supporting cloud computing.

2. Related Work

Virtual resources scheduling problem has been a hot research spot in numerous studies of virtualization. Early research focused mainly on how to ensure application performance while improving the utility of datacenter. At present, energy consumption and cost of data center have become two important aspects. Cloud computing service providers now mainly adopt some of the basic strategies of resource scheduling to schedule virtual resources. For example, EC2(Elastic Compute Cloud) uses the most basic rotation scheduling strategy for virtual machine scheduling to implement load balancing of the data center, but the energy consumption is not taken into consideration. HP schedules virtual machines in FCFS (first come, first serve)[2] mode to achieve the fairness of

Received (January 8, 2018), Review Result (March 3, 2018), Accepted (March 5, 2018)

ISSN: 2005-4262 IJGDC

virtual machine scheduling. These virtual machine scheduling strategies are simple and easy to realize. However, with the increasing scale of the cloud data center, these strategies cannot meet the demand.

In recent years, researchers have used new methods to solve the scheduling problems of complex virtual machines, such as heuristic algorithms and multi-objective optimization algorithms. In order to solve the load balancing problem in cloud data center, Teng F uses the game theory model for virtual machine scheduling[3]. Virtual machine scheduling algorithms based on game theory can better ensure load balancing without considering the utility and energy consumption of the data center, while ignoring the QoS requirements of cloud users. Sun D *et al.* propose a cloud virtual resources optimization scheduling algorithm based on immune clone[4]. This algorithm quantifies the utility and preference of users. From the perspective of multi-user QoS requirements, this algorithm is used as an objective function to solve the problem using multi-objective optimization algorithm, and finally a better virtual resource scheduling solution is obtained. Although the algorithm considers the QoS requirements of users from multiple perspectives, it does not consider the energy consumption of the data center and does not provide the corresponding scheduling strategy to reduce the energy consumption. For the problem of virtual machine placement in large scale cloud data center, Xiao designs a virtual machine scheduling framework based on heuristic algorithm[5], which can greatly reduce virtual machine migration. However, the framework ignores system load balancing and user application performance issues. In order to reduce energy consumption of data center, Li Xu proposes a virtual machine scheduling strategy based on multi-objective optimization[6], which models the virtual machine scheduling problem as a multiple optimization problem, and uses multi-objective optimization algorithm to solve problem. But the algorithm only considers energy consumption from the perspective of physical server activation quantity, and it hasn’t taken the cost of data center and quality of service into account. Paper [7] proposes a virtual machine algorithm based on multi-objective optimization, but the algorithm does not consider the diversity of virtual resources, and it only completes one experiment to verify the validity of the algorithm without conducting more experiments to compare. Considering the data transmission time and communications between subtasks, Fu Xiong proposes a heuristic optimal virtual machine placement algorithm[8]. Related simulations demonstrate that this algorithm can reduce the completion time of user tasks and ensure the feasibility and effectiveness of the overall network performance of applications when running in a cloud computing environment. However, this algorithm has not considered the increasing complexity of the initialization time and execution time, which have a great impact on the completion time of an application.

Furthermore, foreign scholars have a lot of research in the cloud computing scheduling. To solve the problem of high energy consumption in data center, Jang J W puts forward a virtual machine scheduling algorithm based on energy consumption awareness[9]. The algorithm uses the server consolidation technology to migrate the virtual machines which run on the physical machine with low load, and then shut it down to achieve the purpose of saving energy. However, the algorithm ignores that the data center’s load imbalance will affect application performance. In order to solve the resource competition among multi-tenant, H Hong proposed an optimal resource allocation strategy based on utility function[10]. By defining the evaluation function and the cost function of resource configuration among tenant, a utility optimization model is established. Liu J *et al.* propose a general solution based on multi-objective scheduling[11], which can execute Scientific Workflows in a multisite cloud while reducing execution time and monetary costs. The solution consists of a multi-objective cost model including execution time and monetary, a single site virtual machine provisioning approach (SSVP), which generates VM provisioning plans for fragment execution, and a multisite scheduling approach, ActGreedy, which allows for considering stored data constraints while reducing the cost based on the multi-objective cost model. Abdullahi M *et al.* propose a Discrete Symbiotic Organism Search(DSOS) algorithm[12] for optimal scheduling of tasks on cloud resources. It utilizes Symbiotic Organism Search(SOS) to schedule the cloud computing tasks, which is a metaheuristic optimization technique, in order to minimize makespan and increase system utilization. Simulation results reveal that DSOS converges faster when the search gets larger which makes it suitable for large-scale scheduling problems. However, it hasn't taken the energy consumption in the cloud data center into account. For the efficient scheduling problem in cloud computing environment, Gan Q *et al.* introduced the weighted least square support vector machine[13] to reflect the completion time and cost of cloud computing tasks and obtained the robust estimation through limited observation.

In summary, there are two problems in currently virtual machine scheduling policies. On the one hand, the existing virtual machine scheduling strategies in the cloud computing are simple and easy to implement, but they are not suitable for growing large scale cloud data center. On the other hand, the existing resource scheduling strategies based on energy consumption and cost just consider from one single aspect, such as the cost, energy consumption or users' QoS requirements.

Hence, in this paper, we expand two-thirds of paper[7], taking virtual resources from three-dimensions(CPU, ram, hard disk)to four-dimensions(CPU, ram, bandwidth, hard disk), and containing more experiments. We establish the virtual machine scheduling model VMSA-EU, give full consideration to reduce the energy consumption and improve the utility of datacenter. And we also propose a virtual machine scheduling algorithm based on NSGAII to solve the model.

3. Multi-objective Optimization Problem

The multi-objective optimization problem has been one of the main form of scientific research and engineering application performance problems. Multi-objective optimization problem is proposed first by the French economist V. Pareto when studying economic balance, introducing and promoting the Pareto optimal solutions. Each target in the multi-objective optimization problem is called the child targets. The mutual influence between each target and function makes that the multi-objective optimization is not only satisfy each sub-targets optimization conditions, and should satisfy the constraints of the relationship between sub goal conditions. Because the relationship between the child targets are often complex, sometimes even contradictory, multi-objective optimization problem is essentially deal with the uncertainty of child targets constraints. The mathematical model of multi-objective optimization consists of decision variables, objective function and constraint conditions. The application domain of multi-objective optimization is different, its mathematical description is also different, including general multi-objective optimization, dynamic multi-objective optimization and uncertain multi-objective optimization *etc.*

General multi-objective optimization mathematical description is as follows:

$$\begin{aligned} & min(max)\ \ y = f(x) = [f_1(x), f_2(x), \cdots, f_n(x)] \\ & s.t.\ \ n = 1, 2, \cdots, K \\ & \quad e(x) = [e_1(x), e_2(x), \cdots, e_k(x)] \le 0 \\ & \quad h(x) = [h_1(x), h_2(x), \cdots, h_m(x)] = 0 \\ & \quad where \qquad x = [x_1, x_2, \cdots, x_d, \cdots, x_N] \\ & \qquad\qquad\qquad x_{d\_\min} \le x_d \le x_{d\_\max} (d = 1, 2, \cdots, N) \end{aligned} \tag{1}$$

Where, x is N-dimension decision variables, y is objective function, K is the total number of objective functions. And $f_n(x)$ is the nth objective function. $e(x)$ are inequality constraints, $h(x)$ are equality constraints. The constraints determine feasible region of decision variables, and x_{d_min}, x_{d_max} is lower bound and upper bound of the decision variable x_d.

4. Virtual Machine Scheduling Model

In this paper, the virtual machine scheduling is seen as a classic packing problem (BPP), and the bin packing problem is to put some objects with various sizes into a defined space in order to obtain the specified optimal benefit. Virtual machine scheduling is that properly loading virtual machines into physical machines and different physical machines represent different boxes, virtual machines are the goods which are to be packed, available resources of physical machine represent the size of the box, and resources used by the virtual machine express the size of goods. Considering four virtual resources memory, CPU, bandwidth and disk, the virtual machine scheduling model VMSA-EU is modeled as a four-dimension bin packing problem. And then the four-dimension bin packing problem is converted to a multi-objective optimization problem. Therefore, we design a multi-objective optimization algorithm based on NSGAII to solve the problem.

4.1. Related Definitions and Concepts

Definition 1: In Our model VMSA-EU, vector $PM = \{PM_1, PM_2, ..., PM_n\}$ is all the PMs (physical machines) in data center, vector $VM = \{VM_1, VM_2, ..., VM_m\}$ is all the VMs(virtual machines), where $PMi = \{PM_{ci}, PM_{ri}, PM_{hi}, PM_{ni}, \rho\max_i, PM_{costi}\}$ respectively represent processing power PM_{ci}, memory capacity , network bandwidth and capacity of the hard disk, maximum power consumption and cost of; $VM_j = \{VM_{ci}, VM_{ri}, VM_{hi}, VM_{ni}, R_j\}$ respectively represent CPU resources, memory resources, hard disk resources and network bandwidth required for VM, and the utility of, and $R_j = \{R_{j1}, R_{j2}, \cdots, R_{jn}\}$, R_{ji} is the utility generated by running on. The objective functions and constraint conditions of virtual machine scheduling model VMSA-EU will be described in the following two sections.

4.2. Constraints

To allocate virtual resources efficiently and reasonably and gain the most utility of datacenter is the goal of virtual machine scheduling. And how to map VMs to PMs is a key problem. We use a $m \times n$ matrix $VP = \{VP_{ji}\}$ to show the mapping relation between VMs and PMs, where $i = (1,2,\cdots,n)$, $j = (1,2,\cdots,m)$, n is the number of PMs in datacenter, m is the number of VMs in datacenter, $VP_{ji} \in \{0,1\}$. If $VP_{ji} = 1$ it shows that VM_j is allocated running on PM_i, otherwise if $VP_{ji} = 0$ it indicates that VM_j is not allocated running on PM_i.

When scheduling VMs in VMSA-EU, the following constraints must be satisfied.

$$\sum_{i=1}^{n} VP_{ji} \le VP_{ji}, \forall j \in \{1,2,...,m\} \quad (2)$$

$$\sum_{j=1}^{m} VM_{cj} \times VP_{ji} \le PM_{ci}, 1 \le i \le n \quad (3)$$

$$\sum_{j=1}^{m} VM_{rj} \times VP_{ji} \le PM_{ri}, 1 \le i \le n \quad (4)$$

$$\sum_{j=1}^{m} VM_{hj} \times VP_{ji} \le PM_{hi}, 1 \le i \le n \quad (5)$$

$$\sum_{j=1}^{m} VM_{nj} \times VP_{ni} \le PM_{ni}, 1 \le i \le n \tag{6}$$

Where formula (2) shows that any VM can only be mapped to one PM. Formula (3), (4), (5) and (6) indicate that the total virtual resources allocated to VMs must not exceed the resources capacity of PMs.

4.3. Objective Functions

The problems of energy consumption and utility in a cloud computing data center have become the two problems thought most when cloud service provides build a cloud platform. In this paper, we design the following two objective functions according.

(1) Minimize energy consumption

Based on the fact that the energy consumption of PM has a linear relation with CPU utilization, paper [14] raises a model of energy consumption of PM shown in formula (7)[14].

$$E_n = \sum_{i=1}^{n} ((\rho \max_i - \rho \min_i) \times T_{cpui} + \rho \min_i) \tag{7}$$

We put forward a new model based on paper [14] as shown in formula (12). We can turn off some PMs those there are no VMs running to reduce energy consumption.

$$E_n = \sum_{i=1}^{n} ((\rho \max_i - \rho \min_i) \times T_{cpui} + \rho \min_i) \times C_i \tag{8}$$

Where, E_n is energy consumption of datacenter per unit time, which use watt (W) said; n is the number of PMs; ρmax_i is maximum power consumption of PM_i when PM_i's CPU utilization is at its peak, namely the power consumption when CPU utilization is 100%; ρmin_i is minimum power consumption of PM_i when PM_i is turned off or its CPU utilization is less than 1%; We can know that $\rho min_i \approx \rho max_i \times 0.01$ from paper [14] T_{cpui} is CPU utilization of PM_i, and according to related definitions in section 4.2.1 and section 4.2.2 we can draw that $Tcpui = (\sum_{j=1}^{m} VM_{cj} \times VP_{ji}) / PM_{ci}$; $C_i \in \{0,1\}$ shows whether the PM is active. If $C_i = 1$ then PM_i is active, namely there are some VMs running on PM_i , and if $C_i = 0$ then PM_i is turned off.

(2) Maximize utility

The utility of cloud computing datacenter is the difference between the utility of all VMs and the cost of VMs, and the calculation method is shown in formula (9).

$$R_n = \sum_{j=1}^{m} (\sum_{i=1}^{n} R_{ji} \times VP_{ji}) - \sum_{i=1}^{n} PM_{\cos ti} \times C_i \tag{9}$$

In formula (9), n is the number of VMs, m is the number of VMs. R_n is the utility of data center; R_{ji} is the utility generated by VM_j running on PM_i and $PM_{\cos ti}$ is the cost of PM_i. $C_i \in \{0,1\}$ indicates where the physical machine is active. If $C_i = 1$ then physical machine PM_i is active, namely there are some virtual machines running on PM_i, and if $C_i = 0$ then PM_i is turned off. Its calculation method is shown in (10).

$$C_i = \begin{cases} 1 & \sum_{j=1}^{m} VP_{ji} \ge 1 \\ 0 & \sum_{j=1}^{m} VP_{ji} = 0 \end{cases} \tag{10}$$

Hence, the virtual machines scheduling model based on multi-objective optimization VMSA-EU can be described as:

$$\begin{aligned}
&\min: E_n=\sum_{i=1}^{n}(((\rho_{\max i}-0.01*\rho_{\max i})\times\frac{\sum_{j=1}^{m}VM_{ci}\times VP_{ji}}{PM_{ci}}+0.01*\rho_{\max i})\times C_i)\\
&\max: R_n=\sum_{j=1}^{m}(\sum_{i=1}^{n}R_{ji}\times VP_{ji})-\sum_{i=1}^{n}(PM_{costi}\times C_i)\\
&s.t.\\
&\qquad \sum_{j=1}^{n}VP_{ji}\le 1,\quad \forall j\in\{1,2,...,m\}\\
&\qquad \sum_{j=1}^{m}VM_{cj}\times VP_{ji}\le PM_{ci},\quad 1\le i\le n\\
&\qquad \sum_{j=1}^{m}VM_{rj}\times VP_{ji}\le PM_{ri},\quad 1\le i\le n\\
&\qquad \sum_{j=1}^{m}VM_{hj}\times VP_{ji}\le PM_{hi},\quad 1\le i\le n\\
&\qquad \sum_{j=1}^{m}VM_{nj}\times VP_{ni}\le PM_{ni},\quad 1\le i\le n
\end{aligned} \tag{11}$$

5. Virtual Machine Scheduling Algorithm based on NSGAII

NSGAII is an improved non-dominated sorting in genetic algorithms based on NSGA[15]. NSGAII introduces elitism in non-dominated sort and adopts crowding distance to reduce the complexity of algorithm and maintain diversity. And it is also not affected by initial population. Hence, we use a multi-objective optimization algorithm with constraints based on NSGAII to solve our model VMSA-EU.

5.1. NSGAII Algorithm

The NSGAII algorithm is one of the multi-objective optimization algorithm and can handle arbitrary multi-objective optimization problem and get the uniform distribution of Pareto optimal solutions. And the algorithm has very good effect on finding Pareto optimal solutions and keeping the population diversity. The NSGAII algorithm has the following features:

(1) Using non-dominated sorting: The population is divided into different levels on the basis of individuals in the population of dominant degree. Let all non-dominated individuals in current population be the same level which are set to 1, and remove it from the population; And then find new non-dominated individuals which are set to 2; Repeat it until all individuals in population are set to the corresponding level.

(2) Calculate crowd distance to guarantee the diversity of population: estimate the crowded degree among the individuals through calculating distance individual adjacent between two individuals' objective function value of the.

(3) Put forward the elite strategy to prevent losing excellent individual: merge parent population and progeny population, and then do crossover mutation.

In order to evaluate the advantages and disadvantages between any two individuals in a population, the NSGAII puts forward the comparison operation principles: If an individual's non-dominated sorting level is low, the individual is better; if two individuals belong to the same level, then whose crowded distance is larger, who is better. The NSGAII algorithm basic steps are as follows:

(1) Generate an initial population Pt which size is N randomly within the solution space, and then decode the individuals in the population and calculate value of objective function;
(2) Do non-dominated sort for initial population, and then make a selection, crossover and mutation to get new population Qt;

(3) Elite strategy: merge Pt and Qt to get population Rt, and do non-dominated sort and calculate crowded distance for Rt. Choose N of the best individuals to form a new population Pt+1, and then make a selection, crossover and mutation to get new population Qt+1;
(4) Terminating condition: Whether to set the maximum number of iterations? If so, stop operation, output the Pareto solution of multi-objective optimization, otherwise go to step (3) to continue the elite strategy iteration.

5.2. Coding Method

Paper [16] uses a packet coding method as shown in Figure.1, where V represents virtual machines, H expresses physical machines and C is the scheduling solution of virtual machines. This paper improves this coding method and designs a new coding method as shown in Figure.2. We define an m-dimensional vector $C = \{C_1, C_2, \cdots, C_m\}$ to express the scheduling solution. Where $C_j \in$ □ ;$0 \le C_j \le n$, n is the number of physical machines, m is the number of virtual machines, and C_j shows which physical machine is running the virtual machine V_j. If $C_j = 0$, then the virtual machine V_j is not allocated to a physical machine, so the utility of V_j is 0 by formula (9).

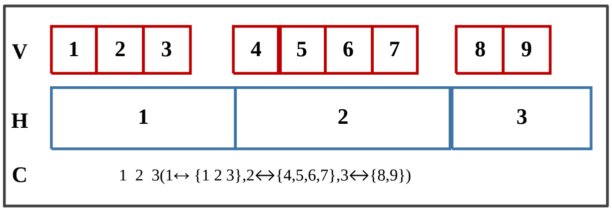

Figure 1. Coding Method in Paper [17]

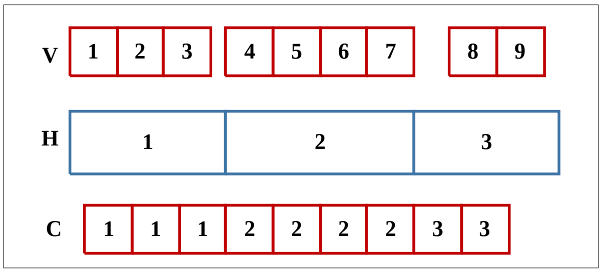

Figure 2. Coding Method in This Paper

5.3. Determine Initial Population

The NSGAII algorithm is quite to initial population's dependence, and chooses initial population randomly, so it makes the gap of population superiority big and has a direct effect on optimization results. As a result, we firstly choose a series of m-dimensional vector defined in section 4.3.2 randomly to determine initial population, and then optimize the population by formula (7)~(10) to remove those population unsatisfied constraints. Finally, each population C can only mapped to a feasible solution VP defined in section 4.2.2. Such as the population $C = \{1,1,1,2,2,2,2,3,3\}$ in Figure.2 can be represented by the feasible solution VP shown in formula (16). So that each population C can be converted into a binary encoding, and binary coding has a higher efficiency than the decimal encoding in the process of searching. And the result of optimize has better robustness on probability of mutation and crossover.

$$Solution\ VP = \begin{Bmatrix} 100 \\ 100 \\ 100 \\ 010 \\ 010 \\ 010 \\ 010 \\ 001 \\ 000 \end{Bmatrix} \quad (12)$$

It should be noted that:(1) If $C_j = i, i \neq 0$ then $VP_{ji} = 1$ and $VP_{ji}^{'} = 0(\forall i' \neq i)$, namely virtual machine VM_j is running on physical machine PM_i ; (2) If $C_j = 0$ then, $VP_{ji} = 0(\forall i \in \{1, 2, ..., n\})$, that is virtual machine VM_j is not allocated to any physical machine.

5.4. Fast Non-Dominated Sort

The NSGAII algorithm need to grade for the pros and cons of the individuals in the population before select operation, and the merits of the individual is judged by its objective function value. The algorithm needs to save two variables: (1) the number of solutions which dominate the solution n_p ; (2) a set of solutions Sp which the solution p dominates. The pseudo code of fast non-dominated sort is shown in Figure 3:

Fast non-dominated sort

```
Input: population P before fast non-dominated sort
Output: population P after fast non-dominated sort
Steps:
def fast_nondominated_sort( P ):
    F = [ ]
    for p in P:
        Sp = [ ]
        np = 0
        for q in P:
            if p > q:          //If p dominates q then include q in Sp
                Sp.append( q )
            else if p < q:     //If p is dominated by q then increment np
                np += 1
        if np == 0:
            p_rank = 1         //If no solution dominates p then p is a member
                               of the first front
        F1.append( p )
    F.append( F1 )
    i = 0
    while F[i]:
        Q = [ ]
        for p in F[i]:
            for q in Sp:
                nq -= 1
                if nq == 0:
                    q_rank = i+2
                    Q.append( q )
        F.append( Q )
        i += 1
```

Figure 3. Fast non-Dominated Sort

5.5. Selection, Crossover and Mutation

We need to do selection, crossover and mutation operation for population after fast non-dominated sort to produce a new offspring population for the next round of iterations.

(1) Selection

The selection operation is for that we can choose the optimal individual as parent populations when do crossover and mutation operation. Each individual in the population can get two attributes after non-dominated sorting and crowded distance calculation: non-dominated sorting rank and crowding distance. This paper we use the tournament selection operator, namely randomly select two individuals and compare the value of rank and distance, and then select a better individual. When the rank of individual i is less than the rank of the individual j, or when the rank of individual i is equal to the rank of the individual j and the distance of individual i is bigger than the individual j. Then the individual i is better than that of the individual j. The size of tournament selection operator in this paper is 2, and the select operation steps are as follows: (1) Select two individuals in a population, and compare the non-dominated sorting level and crowded distance after non-dominated sorting, then choose the optimal individual heredity to the next generation. (2) Repeat the above steps M times, get M individuals in the next generation population.

(2) Crossover and Mutation

Crossover operation of genetic algorithm is designed according to the nature of biological evolution process, and biological evolution in nature generates new chromosomes through crossover and mutation and chromosome recombination. Therefore, crossover operation of genetic algorithm also generates new individual by exchange genes between individuals. We exchange genes between two pairs of individual at a certain probability to generate two new individual. When the probability of crossover operation Pc is high, the genetic algorithm has stronger ability to generate new individual, that is to say, the genetic algorithm has stronger ability to explore new solution space. But when the crossover probability is too high, it will damage the optimal property of individuals a certain extent; When the probability of crossover operation Pc is lower, the genetic algorithm are less able to explore new solution space, but at the same time the possibility of destroyed optimal property of individuals is relatively smaller. Several kinds of crossover operator commonly used for binary coding or the real number encoding are the single point of intersection, two-point crossover and multipoint crossover and uniform crossover, *etc*. Genetic recombination is done between individuals from the parent population at a certain probability in crossover operation. It can generate a large number of new individuals in progeny populations, increase the probability of the emergence of the best individual, and also improve the global search ability; and the mutation operator is to keep the diversity of the individuals in the population by changing the individuals' genes, to overcome the premature convergence in the process of global search. Cooperate with each other using crossover and mutation operators can our algorithm have a good local and global search performance.

In this paper, we adopts multi-point crossover operator and binary mutation operator for crossover and mutation operation because we use binary encoding method. We random set up some multiple intersection in the two individuals to mutation, and then exchange genes between the intersection. We invert the binary bit at a certain mutation probability, namely if an original binary bit is 0 in one individual coding, the mutation operation set it to 1; If the original binary bit is 1, then set it to 0.

5.6. Processes of Virtual Machine Scheduling Algorithm based on NSGAII

In our virtual machine scheduling algorithm, the first step is doing chromosome encoding, and generating an initial population P0 which satisfy the constraint conditions randomly. And then choose better individuals from P0 as parent population to do selection, crossover and mutation operation to generate new offspring population Q0 after doing non-dominated sort and crowding distance for P0. Next, combine P0 and Q0 to do non-dominated sort and crowding distance, and choose better individuals as new offspring population to do next iteration until reach the number of iterations. Finally, we get the optimal solution.

The pseudo code of the algorithm is shown in Figure 4:

Virtual machine scheduling algorithm based on NSGAII

```
Input：number of population N，iterations gen
Output：decision variable P
Steps：
1)Initialize  P0              //Initialize the population P0
2)P0' = Constraint(P0)      //repair the initial population by (7)~(10) to make
                              initial population meet constraint conditions
3)evaluate_objective(P0')  //Calculate the objective function value
4)P0'' = non_domination(P0')     //non-dominated sort
5)for i=1:gen          //iterations
6)      Pt=tournament_selection(P0'')  //select parent populations
7)      Qt=genetic_operator(Pt) //Crossover and mutation
8)      Qt'=Constraint(Qt)
8)      Rt=(Pt) U(Qt')      //merge the parent population and progeny
                              populations
9)      Rt' =evaluate_objective(Rt)
9)      Rt''=non_domination(Rt' )
10)      Pt+1=replace_chromosome(Rt'')//select
11) end
```

Figure 4. The Pseudo Code of the Virtual Machine Scheduling Algorithm

6. Simulation Experiment and Analysis

In order to verify the feasibility of our model VMSA-EU and the virtual machine scheduling algorithm based on NSGAII, we perform two experiments which take minimizing energy consumption and maximizing utility of cloud computing datacenter as goals. The aim of experiment 1 is to verify whether the virtual machine scheduling algorithm based on NSGAII we designed can solve the virtual machine scheduling problem effectively. Experiment 2 shows that our algorithm can give better optimal solutions compared with other similar model. We take the opposite of objective function 2 the utility of datacenter in experiments, because of that algorithms adopted by our paper all compare the minimum of objective functions.

In Experiment 1, we conduct experiment 30 times and then average the results. Table 1 gives the configuration of experiment 1. Figure 5 gives the results of experiment 1, and the value of energy is shown in the horizontal coordinate, the opposite of utility value is shown in the vertical coordinate. From Figure 5, we can see that the optimal solutions we get are almost on a surface. Therefore, the virtual machine scheduling algorithm based on NSGAII better finds out the optimal solution of virtual machine scheduling problem, maximizes utility and minimizes energy consumption in datacenter. Because C is likely to be $C=\{0,0,\cdots,0\}$, namely all the virtual machines are unallocated on any physical server. So there happens that the values of both object functions are all 0.

Table 1. Parameters in Experiment 1

population size	iterations	crossover probability	mutation probability
200	1000	0.9	0.1

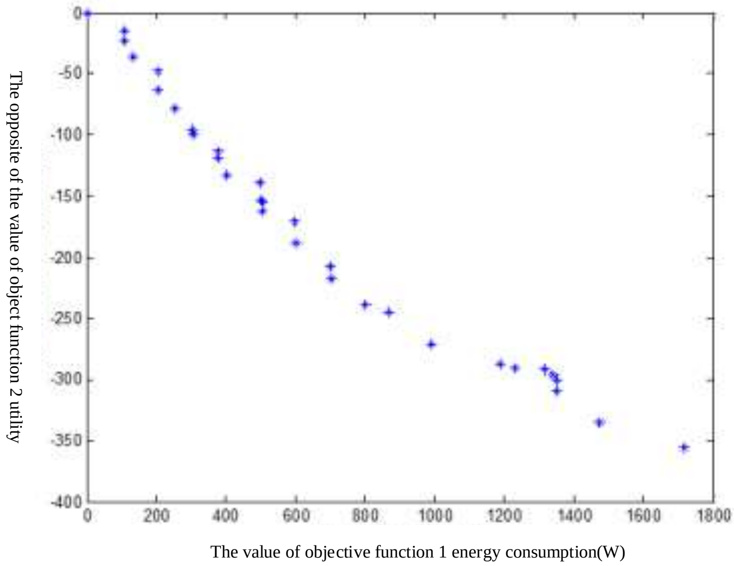

Figure 5. The Results of Experiment 1

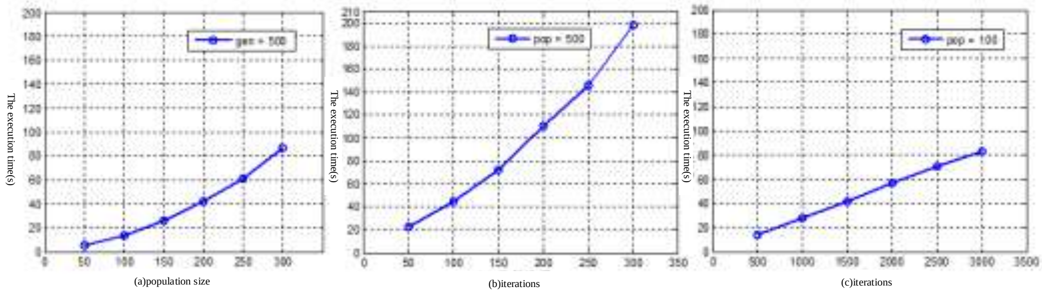

Figure 6. Execution Time of Algorithm with Different Parameters

The execution time of the virtual machine scheduling algorithm based on NSGAII with different parameters is shown in Figure.6. Figure.6 (a) shows the execution time when the population size is different while the number of iterations is same (gen=500); Figure.6 (b) and Figure.6 (c) show the execution when the number of iterations is different while the population size is same. From Figure.6 (b), we can see that the execution time when the population size is 500 has a larger increase than the execution time when the population size is 100. Thus we can be seen that the population size has a greater influence on the execution time of the algorithm.

In experiment 2, we compare our proposed algorithm with another multi-objective optimization algorithm SPEA2[18]. The configuration of experiment 2 is shown in Table 2. Figure.7 and Figure.8 gives experimental results and execution time, respectively.

Table 2. Parameters in Experiment 1

Algorithm	Population size	Iterations	Crossover probability	Mutation probability
NSGA2	200	1000	0.9	0.1
SPEA2	200	1000	0.9	0.1

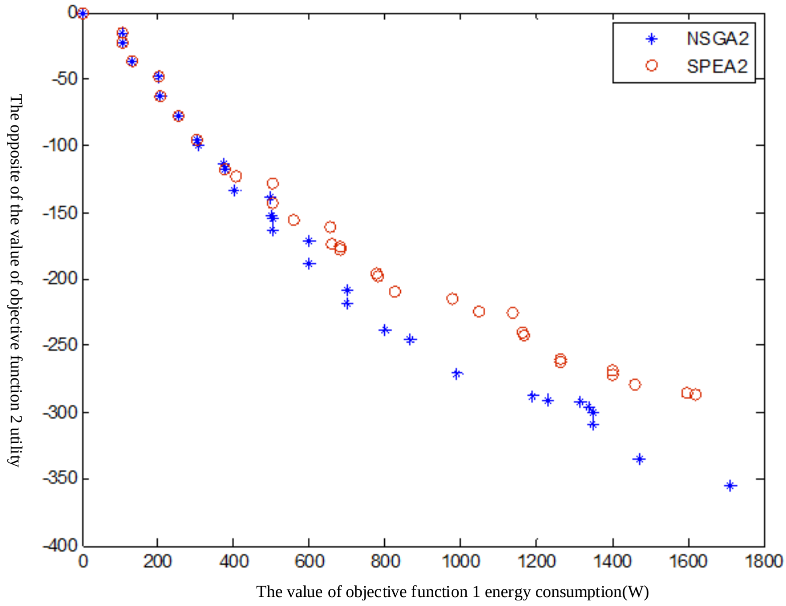

Figure 7. Comparison with Algorithm SPEA2

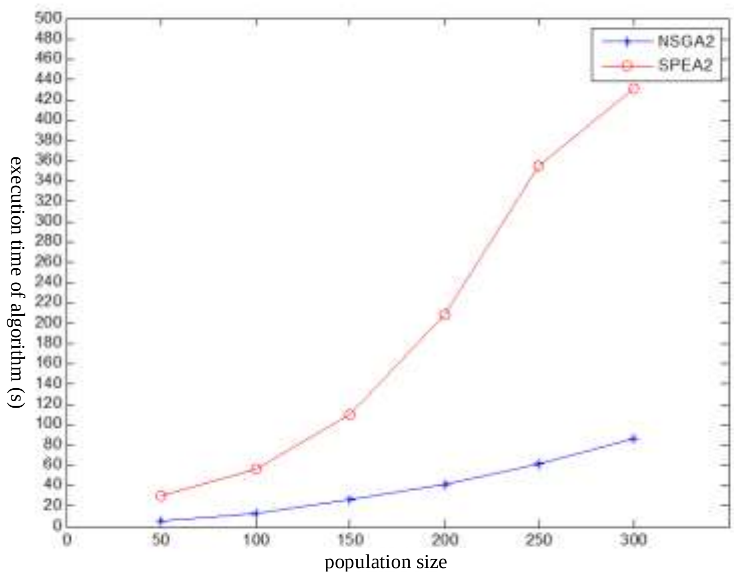

Figure 8. Comparison between the Execution Time

From Figure.7 and Figure.8, we can see that our algorithm has better performances both on results and executing speed than SPEA2.

7. Conclusion

Taking into minimizing energy consumption and maximizing utility of datacenter, we give out a virtual machine scheduling model VMSA-EU based on energy and utility, and we also put forward a virtual machine scheduling algorithm based on NSGAII to solve our model. The experimental results show that our model and algorithm can reduce energy consumption while improving utility. However, there are many factors that influence virtual machine scheduling in the real cloud computing environment. Therefore, we will take more actuals factors into our future work.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (General Program) under Grant No. 61572253, *etc.* 13th Five-Year Plan *etc.* Equipment Pre-Research Projects Fund under Grant No. 61402420101HK02001, Aviation Science Fund under No. 61202351.

References

[1] H Stevens and C Pettey, "Gartner says cloud computing will be as influential as e-business", Gartner Newsroom, Online Ed, **(2008)**.
[2] R. Dukaric and M. B. Juric, "Towards a unified taxonomy and architecture of cloud frameworks", Future Generation Computer Systems, vol 29, no. 5, **(2013)**, pp. 1196-1210.
[3] F. Teng, "A new game theoretical resource allocation algorithm for cloud computing",, International Conference on Advances in Grid and Pervasive Computing, **(2010)**, pp. 321-330.
[4] D. Sun, G. Chang and F. Li, "Optimizing multi-dimensional QoS cloud resource scheduling by immune clonal with preference", Acta Electronica Sinica, vol 39, no. 8, **(2011)**, pp. 1824-1831.
[5] L. I. Qiang, "B University, Beijing, *etc.*Adaptive Management and Multi-Objective Optimization for Virtual Machine Placement in Cloud Computing", Chinese Journal of Computers, vol 34, no.12, **(2011)**, pp. 2253-2264.
[6] Y Guo and S Mao, "CRP Algorithms Based Resource Provisioning Policy for Cloud Computing", Computer Technology & Development, **(2012)**.
[7] X. U. Li, Z. B. Zen and C. Yao, "Study on virtual resource allocation optimization in cloud computing environment", Journal on Communications, **(2012)**.
[8] J. R. Zhu, Y. Zhuang and J. Li, "Virtual Machines Scheduling Algorithm Based on Multi-Objective Optimization in Cloud Computing", Advanced Materials Research, vol 1046, **(2014)**, pp. 508-511.
[9] X. Fu, Y. Cang and X. Zhu, "Scheduling Method of Data-Intensive Applications in Cloud Computing Environments", Mathematical Problems in Engineering,vol 2015, no.4, **(2015)**, pp. 1-8.
[10] H. Hong, "Utility Optimization of Multi-tenant in Divisible Cloud Resource Allocation", International Journal of Grid and Distribution Computing, vol 10, no.2, **(2017)**, pp. 63-71.
[11] J. W. Jang, M. Jeon and H. S. Kim, "Energy Reduction in Consolidated Servers through Memory-Aware Virtual Machine Scheduling", IEEE Transactions on Computers, vol 60, no.4, **(2011)**, pp. 552-564.
[12] J. Liu, E. Pacitti and P. Valduriez, "Multi-objective scheduling of Scientific Workflows in multisite clouds", Future Generation Computer Systems, vol 63, **(2016)**, pp. 76-95.
[13] Q. Gan and J. H. Zheng, "A New Algorithm to Improve Efficiency of Resource Scheduling in Clouding Computing Based on Extended Support Vector Machine", International Journal of Grid and Distribution Computing, vol 9, no.3, **(2016)**, pp. 125-134.
[14] M. Abdullahi, M. A. Ngadi and S. I. M. Abdulhamid, "Symbiotic Organism Search optimization based task scheduling in cloud computing environment", Future Generation Computer Systems, vol 56, **(2016)**, pp. 640-650.
[15] Z. Wang, K. Shuang and L. Yang, "Energy-aware and revenue-enhancing Combinatorial Scheduling in Virtualized of Cloud Datacenter", Journal of Convergence Information Technology, vol 7, no.1, **(2012)**.
[16] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms", MIT Press, **(1994)**, pp.221-248.
[17] J. Xu and J. a B Fortes, "Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments", Green Computing and Communications, **(2010)**, pp. 179-188.
[18] M. Laumann, "SPEA2 : Improving the Strength Pareto Evolutionary Algorithm", Technical Report, **(2001)**.

Authors

Zhu Wei, he received the B.S. degree from Northeastern University in 1996 and his M.S. degree in Computer Applications Technology from Huazhong University of Science & Technology in 2004. Currently, He is the research professor of Jiangsu Automation Research Institute, Lianyungang, China. His research interests include distribute systems, embedded systems, cloud computing and real-time control systems.

Yi Zhuang, she was born in 1956. She graduated from the Department of Computer Science, Nanjing University of Aeronautics and Astronautics in 1981.Now she is a professor and Ph. D. supervisor of the College of Computer Science and Technology at Nanjing University of Aeronautics and Astronautics. Her research interests include network and distributed computing, information security and dependable computing.

