Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm

Jing Liu*[1], Xing-Guo Luo[2], Xing-Ming Zhang[3], Fan Zhang[4] and Bai-Nan Li[5]

[1,2,3,4,5] **National Digital Switching System Engineering & Technology Research Center, Zhengzhou 450002, China**

Abstract
Cloud computing is an emerging high performance computing environment with a large scale, heterogeneous collection of autonomous systems and flexible computational architecture. To improve the overall performance of cloud computing, with the deadline constraint, a task scheduling model is established for reducing the system power consumption of cloud computing and improving the profit of service providers. For the scheduling model, a solving method based on multi-objective genetic algorithm (MO-GA) is designed and the research is focused on encoding rules, crossover operators, selection operators and the method of sorting Pareto solutions. Based on open source cloud computing simulation platform CloudSim, compared to existing scheduling algorithms, the results show that the proposed algorithm can obtain a better solution, and it provides a balance for the performance of multiple objects.
Keywords: *Task Scheduling, Cloud Computing, Multi-Objective Genetic Algorithm, CloudSim.*

1. Introduction

Cloud computing, the long-held dream of "computing as a utility", is emerging as a new paradigm of large-scale distributed computing driven by economies of scale, in which a pool of highly scalable, heterogeneous, virtualized, and configurable and reconfigurable computing resources (e.g., networks, storage, computing units, applications, data) can be rapidly provisioned and released with minimal management effort in the data centers [1-6]. Economically, the main appeal of cloud computing is that customers only use what they need, and only pay for what they actually use. Resources are available to be accessed from the cloud at any given time, and from any location via the internet [7]. However, data centers use a significant and growing portion of energy, an average data center consumes as much energy as 25,000 households. Therefore, energy-aware computing is crucial for cloud computing systems that consume considerable amount of energy.

The resource demands for different jobs fluctuate over time. Job scheduling system, which efficiently allocates resources to required tasks under the constraint of the Service Level Agreements (SLAs), is a fundamental issue in achieving high performance in cloud computing and of great significance for improving resource load balance, security, reliability and reducing energy consumption of the whole system. However, it is a big challenging problem for efficient scheduling algorithm design and implementation in cloud computing environment.

To reduce the energy consumption, Pinheiro et al. propose a model for minimization of power consumption in a heterogeneous cluster of computing nodes serving multiple web-applications, which periodically monitors the load of resources and makes decisions on switching nodes on/off to minimize the overall power consumption [8]; Raghavendra et al. combine five different power management policies and explore the problem in terms of control theory, but the system fails to support variable SLAs for different applications [9]; Lee et al. propose two algorithms based on pricing model, using processor sharing in order to balance between profit and resource utilization [10]; Gang et al. propose a linear programming driven genetic algorithm, aiming to establish the best scheduler in a utility grid by minimizing the combined costs of all users in a coordinated way [11].

All of the above mentioned methods consider the profit or the energy in their study, but do not the relationship between them. To overcome the deficiencies of the above algorithms, in this paper, we first establish a macroscopic scheduling model with cognition and decision components for the cloud computing, which considers both the requirements of different jobs and the circumstances of computing infrastructure, then propose a job scheduling algorithm based on Multi-Objective Genetic Algorithm (MO-GA), taking into account of the energy consumption and the profits of the service providers, and providing a dynamic selection mechanism of the most suitable scheduling scheme for users according to the real-time requirements; at last, we take some experiments to validate our design and compare our MO-GA based scheduling model to the traditional ones.

2. Job Scheduling Model

In cloud computing, service requests have heterogeneous resource demands because some services may be CPU-intensive whereas others are I/O-intensive. Cloud resources need to be allocated not only to satisfy Quality of Service (QoS) requirements specified by users via SLAs, but also to reduce energy usage and improve the profits of the service providers.

2.1 Model Architecture

Fig 1 shows the functional architecture of the scheduling model we have established, the detail functions of the main components are introduced as follows:

Request cognition component should be fully aware of the special needs for different businesses, which may include the computing, storage and communication requirements for computing, arrival law and concurrent conditions, security and privacy requirements, QoS of the service and so on;

Service decomposition component decomposes the service request into different level of granularities with different processor preferences. In the next procedure, the task manager will analyze the resource requirements of each granularity, and mapping it on to optimal processors to reach a effective solution.

Task manager is responsible for task status management (start, stop, cancel…), determining the scheduling sequence and resource assignment for the requests and allocating suitable resources to each job under the help of the scheduling algorithm.

Resource cognition component plays the role of managing the available resources, monitoring the performances of resources, dynamic optimization of scheduling strategy and error notification.

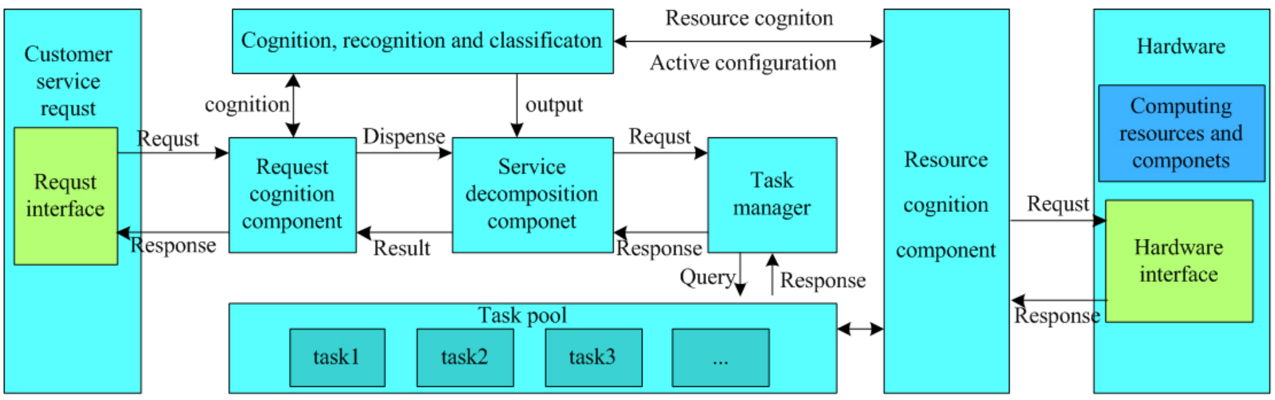

Fig. 1. Functional Architecture of Job Scheduling Model for Cloud Computing

2.2 Problem Formulation

In our model, a cloud application is considered as a collection of work items or jobs that carry out a complex computing task by using cloud resources, and the set $A = (a_1, a_2, \ldots, a_M)$ is a batch of applications arrived in a period. During the scheduling process, the client submits a service request for application $a_i (1 \le i \le M)$, with the resource requirements represented as a triplet (t_i, n_i, d_i), where, t_i represents the reservation time of the application for virtual machines (VMs), which are the virtualized computing elements in cloud computing by means of virtualization technology, n_i for the number of VMs needed for a_i and d_i for the deadline after what the application will be considered to be failed. The problem need to solve for this algorithm is how to schedule these M applications to the given N clouds under the constraints and make the objective function optimal. Where, the N clouds distributed in different geographical areas around the world are usually heterogeneous, while in a cloud, all the VMs are considered homogeneous with the virtualization techniques.

2.3 Objective Function

Suppose application a_i is scheduled to execute on cloud C_j, and p_j represents the Power of each VM in C_j, then, the energy consumption for execution of a_i is given by:

$$E_{ij} = p_j n_i t_i \tag{1}$$

And the profit of the service provider is:

$$R_{ij} = n_i t_i pr - co_{ij} \tag{2}$$

where, the pr is the price unit charged by provider for application a_i, and co_{ij} is the cost of the provider for executing the application a_i.

Combing Eq. (1) and (2), the objective functions can be written as follows:

$$\min E = \min(\sum_{i=1}^{M}\sum_{j=1}^{N} E_{ij}) \quad (3)$$

$$\max R = \max(\sum_{i=1}^{M}\sum_{j=1}^{N} R_{ij}) \quad (4)$$

where, E and R is the total energy consumption and profit for the execution of M application on N clouds respectively.

2.4 Constraints

The constraints are listed as follows:

(1) The application a_i has to be finished before the deadline d_i, otherwise, the schedule is considered to be failed;

(2) Each application can be allocated to only one cloud.

3. MO-GA Scheduling Algorithm

3.1 Encoding Rule

Each schedule is expressed as a 2 by M matrix, where, M is the length of the chromosome. The first row of the matrix represents the requested applications, and second of the matrix is the corresponding number of the cloud where the application is executed. Fig. 2 shows an example of scheduling result, in which, application 2 is assigned to cloud 0, and application 1 is allocated to cloud 5.

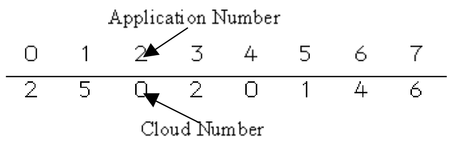

Fig. 2 Encoding example of a Scheduling

According to the above rule, we can see that each application can only be assigned to one cloud, while a cloud may be able to process several applications.

3.2 Population Initialization

The population initialization affects the quality of the future generations, and is an important step in the whole algorithm. In this paper, this step is conducted by combing the random and greedy initialization methods. Owning to the greedy initiation method, the scheduler rejects the applications not meeting the deadline constraint which may cause the whole scheduling failed. This kind of initialization method helps add variety to the initial population and avoid biasing the search of MO-GA.

3.2 Genetic Algorithm

Genetic algorithm is a search heuristic that mimics the process of natural evolution based on a population of candidate solutions. It is routinely used to generate useful solutions to optimization and problems. In the process of evolution, a modification is performed by those operators on each individual. Each chromosome represents a scheduling result, and an evaluation operator (fitness) is called to evaluate the offspring.

(1) Individual Evaluation
In this paper, the fitness is deduced from the energy consumption and profits of the service providers. Only the solutions with the best rank after the evaluation of the fitness function are stored in the Pareto archive which contains the different non-dominated solutions generated through the generations.

(2) Selection operation
The selection operation is based on tournament operator of k individuals, with two strategies: elitism and crowding. The elitism strategy makes use of the individuals in Pareto archive and selects the best ones according to the non-dominated concept to the next generations, allowing the convergence of the evolution process. Crowding strategy takes advantage of crowding distance to estimate the intensity of surrounding solutions and remove the solutions which were too crowded by ranking the crowding distance of each individual. The crowding distance is defined as the circumference of the rectangle defined by its left and right neighbors, and infinity if there is no neighbor.

(3) Crossover Operation
The crossover operator uses two individuals s_1, s_2 to generate two new individuals $s_1^{'}, s_2^{'}$. For individual s_1, first, the operator randomly generates two integers i, j, where, $1 \le i \le j \le N$; then, copies the tasks in s_1 before i and after j to $s_1^{'}$, and maps the tasks between i and j to a temporary individual s_1^m according to the tasks allocation result in s_2; finally, copies the tasks in s_1^m to corresponding place in $s_1^{'}$, as shown in Fig. 3. The individual $s_2^{'}$ is generated using the same method.

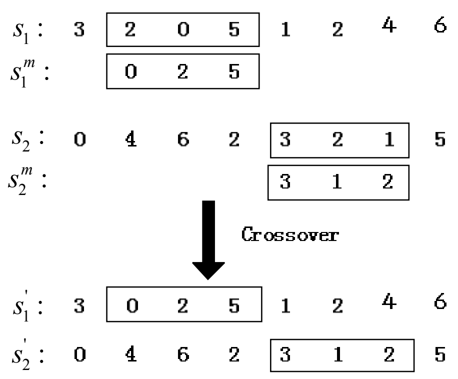

Fig. 3 The crossover operation mechanism

(4) Mutation Operation
The mutation operation chooses two tasks in a individual randomly, and swaps their allocation position to generate a new individual.

3.3 Optimal Selection in Pareto Archive

The results of MO-GA algorithm are a set of Pareto solutions, providing a wide range of possible options, while reducing the efficiency of scheduling process. In practice, users sometimes need to adjust the degree of preference for a particular objective dynamically. This step provides an approach to pick up an optimal solution among the external Pareto archive according to the current requirement. A two dimensional vector is introduced to represent the weighting for a particular objective, whose direction points to the most favorable solution.

Fig. 4 shows an example with 3 two-dimensional vectors, where, p_1 - p_5 represent the external archive of after the MO-GA algorithm, $v_1 = (0,1)$, $v_2 = (\sqrt{2}/2, \sqrt{2}/2)$ and $v_3 = (1,0)$ represents three kind of requirements respectively. For example, p_1 is the optimal solution for vector v_1, and p_3 for v_1, p_5 for v_1.

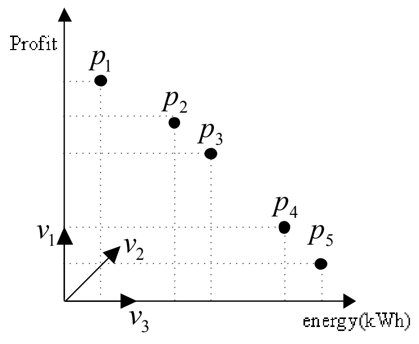

Fig. 4. The schematic diagram of optimal selection

3.4 Implementation Steps

Based on the above ,the implementation steps of this algorithm is listed following:
(1) Initial the population by greedy and random methods;
(2) Modify the individual during the evolution process of the MO-GA algorithm according to the operators indicated in Sec. 3.2, and store the results to external Pareto archive;
(3)Select the optimal solution according to the vector and implement the scheduling result to distributed cloud federation.

4. Simulations and analysis

In order to validate the effectiveness of the proposed algorithm, simulation experiments based on the platform of CloudSim were performed.

4.1 Experimental Settings

The parameter settings in this experiment are as follows

Table 1: Characteristics of the VMs in Each Cloud

NO.	*Power (kW)*	*Frequency(GHz)*	*Memory (GB)*	*Amount*
1	0.28	1.6	1	250
2	0.44	1.8	1	200
3	0.54	2.0	2	150
4	1.04	2.4	2	100
5	1.59	2.6	4	80
6	2.31	2.8	4	60
7	2.43	3.0	4	30
8	3.45	3.2	8	20

(1) Cloud federation parameter. Table 1 shows the characteristics of the VMs which compose the cloud federation in CloudSim. There are eight clouds in this experiment, and the VMs in the same cloud are homogeneous.
(2) Application settings. The request consists of 10000 applications, and the task arrival rates in our experiments are low, medium and high(the high rate has ten times more applications arrival during the same period of time than the medium, so does the medium to low). The length of each task obeys [500, 8000] bimodal distribution, and each has a crest at 2000 and 6000 and near, which is in MI(mega-instructions).
(3) The population size is 20, number of generations is 1000, crossover and mutation rate is 0.95 and 0.1

respectively. The users should pay ¥2 for each VM per hour, and the provider should pay ¥0.5 per kWh for electricity.

4.2 Performance Evaluation

We conduct several experiments and with different parameters of our algorithm. Comparison between our algorithm, maximum applications scheduling algorithm and random scheduling algorithm is listed as following. The maximum applications scheduling algorithm aims to maximize the number of scheduled applications, while the random scheduling algorithm randomly assigns the applications to the cloud. The results of each experiments of the MO-GA have been deduced from 30 independent runs. Table 2 shows the comparison of the MO-GA algorithm with the selection vector $v = (\sqrt{2}/2, \sqrt{2}/2)$ to the other two algorithms, according to the different arrival rates.

Table 2: Experimental comparison for three algorithms

Algorithm	*Arrival Rate*	*Energy (kWh)*	*Profit (¥)*	*Failed Applications*
MO-GA	Low	2340	2988	91
	Medium	2506	2927	130
	High	3875	2846.5	214
Maximum applicatio -ns	Low	4213	2826	105
	Medium	3950	2811.5	159
	High	3904	2796	281
Random	Low	1979	893	2703
	Medium	624.8	251.7	8211
	High	8.2	5.6	9635

We can conclude from the Table 2 that:

(1) Compared to the other algorithms, the MO-GA based scheduling method can obtain a higher profit, while consume lower energy. When the arrival rate is low, the MO-GA methods consumes 44.46% less energy, and obtains 5.73% higher profit, but the improvement reduces with the growth of the arrival rate;
(2) The more the application rate is high, the worse are the results and the higher the number of failed applications is;
(3) There is little difference between the profit and failed applications in MO-GA and maximum applications scheduling algorithm, while the failed applications varies a lot between the two algorithm;
(4) The random scheduling algorithm considers nothing of the resource requirements of the applications and the computing ability of the VMs, thus, poor results are obtained.
(5) With the increase of the arrival rate, more applications are rejected. This is because that all the cloud becomes saturated and busy at the high arrival rate, with no ability to accept the new arrival applications.

To investigate the influence of the selection vector to the MO-GA scheduling result, we conduct an experiment and compare the results of the MO-GA algorithm with three different vectors. The results are listed in Table 3

Table 3: Experimental comparison for three selection vectors

Vector	*Arrival Rate*	*Energy (kWh)*	*Profit (¥)*	*Failed Applications*
$v_1 = (0,1)$	Low	2647	3036.7	93
	Medium	2730	2964.8	131
	High	4070	2856.2	249
$v_2 = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$	Low	2340	2988	91
	Medium	2506	2927	130
	High	3875	2846.5	214
$v_3 = (1,0)$	Low	2334	2980	91
	Medium	2481	2931.4	125
	High	3819	2854	213

From the results, we can see that:
(1) The vector orientations that favor a specific objective obtain a significant improvement on this objective, especially at the low and medium arrival rates.
(2) v_2 corresponds to the scheduling favorite no objective, and the result is also average compared to the other two situations.

4. Conclusions

In this paper, we have established a scheduling model for cloud computing based on MO-GA algorithm to minimize energy consumption and maximize the profit of service provides under the constraint of deadlines. We first propose a job scheduling architecture under the environment of cloud computing, which contains several components to analyze the application, and allocate the suitable resources to the applications to improve the effectiveness and efficiency of the computing; then, the MO-GA based scheduling algorithm is proposed, at last, several experiments are conducted to validate our scheduling models

Acknowledgments

This work was financially supported by the National High-Tech Research and Development Plan of China (2009AA012201), Key Project of Shanghai Science and Technology Commission (08dz501600).

References

[1] Armbrust M, Fox A, Griffith R, Joseph A D, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A and Stoica I, “A view of cloud computing”, Communications of the ACM, Vol. 53, No. 4, 2010, pp. 50-58.

[2] Nidhi Jain Kansal and Inderveer Chana, “Cloud. Load Balancing Techniques : A Step Towards Green Computing”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No. 1, 2012, pp. 238-246.

[3] Iosup, A., Ostermann, S., Yigitbasi, M.N., Prodan, R., Fahringer, T. and Epema, D.H.J, “Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing”, IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 6, 2011, pp. 931-945.

[4] Arunadevi.M and R.S.D Wahidabanub, “Design of Power Efficient Schema for Energy Optimization in Data Center With Massive Task Execution Using DVFS”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, 2012, pp. 407-414.

[5] Almutairi, A., Sarfraz M., Basalamah S., Aref W. and Ghafoor A, “A Distributed Access Control Architecture for Cloud Computing”, IEEE Software Vol. 29, No. 2, 2012, pp. 36-44.

[6] Junaid Qayyum, Faheem Khan, Muhammad LaL, Fayyaz Gul, Muhammad Sohaib and Fahad Masood, “Implementing and Managing framework for PaaS in Cloud Computing”, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No. 3, 2011, pp. 474-479.

[7] Sanjeev Narayan Bal, “Clouds for Different Services”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, 2012, pp. 273-277.

[8] E. Pinheiro, R. Bianchini, E.V. Carrera and T. Heath, “Load balancing and unbalancing for power and performance in cluster-based systems”, in Proceedings of the Workshop on Compilers and Operating Systems for Low Power, 2001, pp. 182-195.

[9] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang and X. Zhu, “No ‘‘power’’ struggles: coordinated multi-level power management for the data center”, SIGARCH Computer Architecture News, Vol. 36, No. 1, 2008, pp. 48-59.

[10] Lee Y.C., Wang, C., Zomaya, A.Y. and Zhou B.B., “Profit-driven service request scheduling in clouds”, In: Cluster, Cloud and Grid Computing (CCGRID), 2010, pp. 15-24.

[11] Garg, S.K., Konugurthi P. and Buyya R, “A linear programming driven genetic algorithm for meta-scheduling on utility grids”, International Journal of Parallel Emergent and Distributed Systems, Vol. 26, No. 6, 2011, pp. 493-517.

Jing Liu is a Ph.D. student at National Digital Switching System Engineering & Technology Research Center, China. He has completed his Bachelor’s and Master’s degrees in Information Engineering University, Zhengzhou, Henan province. He is actively involved in research on resource management in virtualized data centers for Cloud computing, job scheduling and PSS.

Xing-guo Luo is Professor of National Digital Switching System Engineering & Technology Research Center, China. His interests include cloud computing and wireless communication.

Xing-ming Zhang is Professor of National Digital Switching System Engineering & Technology Research Center, China. His interests include cloud computing and Network on a Chip.

Fan Zhang is Lecturer of National Digital Switching System Engineering & Technology Research Center, China. His interests include cloud computing and Network on a Chip.

Bai-nan Li is a Ph.D. student at National Digital Switching System Engineering & Technology Research Center, China. His interests include cloud computing and resource allocation.