International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:3, 2011

A Generic Approach to Achieve Optimal Server Consolidation by Using Existing Servers in Virtualized Data Center

Siyuan Jing, Kun She

***Abstract*—**Virtualization-based server consolidation has been proven to be an ideal technique to solve the server sprawl problem by consolidating multiple virtualized servers onto a few physical servers leading to improved resource utilization and return on investment. In this paper, we solve this problem by using existing servers, which are heterogeneous and diversely preferred by IT managers. Five practical consolidation rules are introduced, and a decision model is proposed to optimally allocate source services to physical target servers while maximizing the average resource utilization and preference value. Our model can be regarded as a multi-objective multi-dimension bin-packing (MOMDBP) problem with constraints, which is strongly NP-hard. An improved grouping generic algorithm (GGA) is introduced for the problem. Extensive simulations were performed and the results are given.

***Keywords*—**GGA-based Heuristics, Preference, Real-world Constraints, Resource Utilization, Server Consolidation

I. Introduction

NOWADAYS, most of the IT service providers chronically host single services or applications on dedicated physical servers in their data centers because of the complex resource requirement of deployed services and the desire to provision for peak demand. This leads to a phenomenon that size of the data centers grows quickly, but the average server utilization is typically very low (The Gartner Group estimated that the utilization of servers in data centers is less than 20 percent [1].). As a consequence, it incurs high investment and operational costs (e.g., for management, maintenance, energy consumption) but decreasing return of investment to IT service providers. However, maximizing profit is the intrinsic characteristic to business organizations. Therefore, they have to find techniques to solve this problem. There is another fact which must be emphasized that the Environmental Protection Agency (EPA), in its August 2007 report to the US Congress, affirmed that data centers consumed about 61 billion kilowatt-hours (kWh) in 2006, roughly 1.5 percent of total U.S. electricity consumption, for a total electricity cost of about $4.5 billion [2]. High energy consumption not only translates to high energy cost, but also high carbon an emission which is not environmentally sustainable.

Virtualization-based server consolidation is an ideal technique to solve the mentioned-above server sprawl problem [11], a situation in which multiple underutilized server take up more space and consume more energy than can be justified by their workload. The most important benefit of such technique to our work is that it can reduce required physical servers and other infrastructure to easier system management, increase resource utilization and thereby reduce overall hardware and operational cost [3], [22].

Generally, there are two ways to achieve such a consolidation. One is that the IT manager wants to switch to a new technology of hardware, such as a new generation of blade servers, and therefore purchases some new servers for consolidation. The other is that IT manager wants to use existing servers for consolidation. This paper focuses on the latter. The difference is that the existing servers generally are heterogeneous, e.g., different technique architecture and resource capacity. Moreover, these servers are diversely preferred by IT manager in terms of their technique architecture, operational cost and running time, and so on. Therefore, when modeling this problem, we not only need to maximize the resource utilization, but also consider the IT manager's preference. We formulate the server consolidation problem (SCP) as a MOMDBP with constraints.

Contribution synopsis The development of the decision model and the optimization technique for the SCP consists of the following:

1) Considering some real-world constraints (e.g., the cooperative relationship and conflict among services, hardware technique requirements of services), five rules for server consolidation are proposed. These principles can make solution much more practical and to be a good guideline to IT managers when they consolidate servers.
2) The problem is formulated to maximize the average resource utilization of selected servers and the average preference value of selected servers. The latter is the first time considered in server consolidation problem.
3) Considering the peak time of each service is in different time section, a workload timesharing analyzing method is used in modeling.
4) An improved GGA is introduced to solve the problem. Extensive experiments are performed to evaluate the proposed model.

The rest of the paper is organized as follows: A brief discussion of related work is presented in Section II. The background knowledge of Virtualization-based server consolidation is analyzed and five rules of consolidation are

Siyuan Jing is with the Department of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China (e-mail: jingsiyuan_628@126.com).

Kun She., is with the Department of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China (e-mail:kunshe@126.com).

given in Section III. Meanwhile, a system model for *SCP* and the formulating problem are introduced. An optimization technique, i.e. GGA-based heuristic, is proposed in section IV. Section V gives the results of simulation with the discussion. In Section we draw conclusions.

II. Related Works

Capacity planning has been a central issue in computer science for a long time [17], [22]. Server consolidation can be regarded as a new extensive capacity planning problem [11]. A traditional analytical approach to support capacity planning is the queuing theory. For example, in [12], the authors propose a utility analytic model for Internet-oriented server consolidation in VM-based data centers, modeling the interaction between server arrival requests with several QoS requirements, and capability flowing amongst concurrent services, based on the queuing theory.

Another way to this problem is combinatorial optimization theory. In [6], the authors propose two energy-conscious task consolidation heuristics aiming to maximize resource utilization and explicitly take into account both active and idle energy consumption. In [21], the authors solve server consolidation problem by combinatorial optimization theory. In [4] and [5], the authors model the server consolidation problem as a vector packing problem with conflicts and tires to minimize the number of servers used for hosting applications within data center and maximize the packing efficiency of the server utilized.

III. Problem Formulation

A. Virtualization

Virtualization is a technique for hiding the physical characteristics of computing resources from the way in which other systems, applications, or end users interact with those resources. Software such as VMWare [8] (or Xen [9] or Microsoft Virtual Server [18]) can transform or virtualize the hardware resources of an x86-based computer—including the CPU, RAM, hard disk, and network controller—to create a fully functional virtual machine that runs its own operating system and applications just like a “real” computer. Note that virtual machines can cover most x86 operating systems (i.e. Windows, Linux, or Solaris x86). Multiple virtual machines share hardware resources without interference so that a single computer can run several operating systems and applications at the same time [19].

B. Workload of Service

The common way of capacity planning is to analyze workload of each service and reserve certain amounts of IT resource, e.g. CPU, RAM, bandwidth, for them [7], [10]. For this purpose, CPU capacity may be measured in SAPS or HP Computons, memory in Gigabyte, and bandwidth in Megabits per second. All of these data are usually logged by some monitor software (e.g., Tivoli) every five minutes or every hour.

Practically, we can easily find that the peak demand of IT resource of each service is in different time sections. Given two services LDAP and OA as an example (See figure 1), we can find that the peak demand of LDAP is around 9:00 am, for most of staff access the system and begin their daily work at this time. But at other time, the resource demand of LDAP is low. Instead, the peak demand of OA is during the working time. Therefore, if we allocate the two services to a same physical server, there will be no decreasing QoS for both of them, i.e., it’s a feasible solution of consolidation for them.

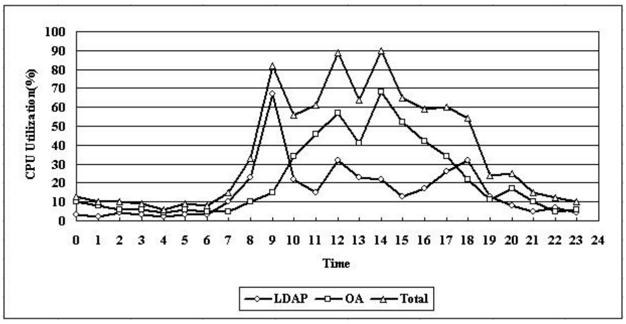

Fig. 1 The comparison of resource demand between different services

C. Preference of IT managers

An acceptable assumption is that IT managers have their own preference for the existing servers. For example, managers want to use servers, whose running time is relative short, to consolidate and the rest are used for the backup, or they want to use servers whose technique is relative advanced. Therefore, when modeling, we not only need to maximize the resource utilization of servers, but also consider the IT manager’s preference.

An item *x* can be represented by a set of features or preference granules $x=\{\Theta_1,\Theta_2,\cdots,\Theta_k\}$ [25] [26], profile is then defined by a set of preference values for each preference granules, $G=\{pref(\Theta_1),pref(\Theta_2),\cdots,pref(\Theta_k)\}$ .The $pref(\Theta_1)$ represents how much a user likes a given preference granule and is computed from the user record. Furthermore, the formula above can be denoted as the follow:

$$pref(x)=\sum_{\Theta_i\in x} pref(\Theta_i)$$

By this way, we can easily study the preference of IT manager to existing servers and apply it into server consolidation problem formulating. Due to this work does not focus on how to study preference; we will not introduce detailed method of it.

D. Five Rules for Consolidation

During the process of consolidation, there are many factors must be taken into account. We analyze such factors and propose five rules in server consolidation.

1) Required IT resource of each service must be satisfied

QoS is the most important evaluating criteria which must be ensured in consolidating. QoS, for the most part, relies on the IT resources (e.g., CPU, RAM, Bandwidth, etc.) assigned to service [16]. Therefore, the IT resource requirement of each service must be satisfied. For the proposed time timesharing analyzing method (See section 3.2), it can be said that the sum of workload of all services (i.e., allocated to a same server) in each interval can not exceed the resource capacity of target server.

2) If a set of services have features which can enhance QoS, they must be allocated to the same target server

To our knowledge, some services, just like web server and database, have frequent interaction. If we deploy them on a server, the time spend on data transmission would be reduced, in another words, the QoS is enhanced. Therefore, if a set of services have such features, they would better be allocated to the same target server. If the sum of workload of the services exceeds the capacity of target server, i.e. violating rule 1, we could analyze the correlation of services and divide them into small subsets.

3) If a set of services are conflicting, they must be allocated to different target servers

Some services have the same special technique requirement (e.g., web servers need port 80 for HTTP) or security strategy of some services are conflicting, such technical conflicts must be avoided when making a solution, i.e., they must be allocated to different target servers.

4) If a type of server has technique features which meet the helpful requirement of a service, the service must be pre-allocated to such type servers

Some services have helpful technique requirements of servers (e.g., x86 architecture or OS). Such requirements must be satisfied by pre-allocating the services to the server which meets has the technique features.

5) If a type of server has technique features which conflict with a service, the service is forbidden to allocated to such type servers

Some services have rejective technique requirements of servers, such requirements must be avoided when consolidating servers, i.e., they must be forbidden to be allocated to such type servers.

If all rules are satisfied, we say it's a feasible solution of server consolidation.

E. The Model and Problem Formulating

The system model contains a collection of (target physical) servers and a collection of (source) services.

Servers. Suppose that there are a set of heterogeneous servers used for service allocation, $P=\{p_1,p_2,\cdots,p_M\}$, where p_j is the *jth* server. Each server is characterized by 1) has a certain capacity r_{jk} of resource $k\in K, K=\{k_1,k_2,\cdots k_{n_K}\}$ (e.g., CPU, RAM, Bandwidth, etc.); 2) has a potential preference value $pref\left(p_j\right)$; 3) has a set of features $A\left(p_j\right)=\{\Theta_1,\Theta_2,\cdots,\Theta_m\}$.

Services. Suppose that there are a set of services, $S=\{s_1,s_2,\cdots,s_N\}$, where s_i is the *ith* service. Each service is characterized by 1) has resources requirement u_{itk}, where t is the time section, $t\in T, T=\{t_1,t_2,\cdots,t_{n_T}\}$ and k is the resource $k\in K, K=\{k_1,k_2,\cdots k_{n_K}\}$. 2) The helpful requirement of server feature is $A\left(s_i^+\right)$, 3) the rejective technique features of server are $A\left(s_i^-\right)$. Moreover, $A\left(s_i^+\right)\cap A\left(s_i^-\right)=\varnothing$.

Preliminaries Now, suppose we are given a set of target servers P and a set of services S , and we need to allocate S to P . All rules of server consolidation must be satisfied (See section 3.4). We term such state as a feasible service to server allocation when each service $s_i\in S$ can be allocated onto at least one p_j subject to all of constrains associated with each service.

Our aim is to compute the optimal solution which can maximize the average resource utilization of selected target servers and the average preference value of selected target.

Mathematically, we can formulate the problem as follows:

$$\max\ f_r=\frac{\sum_{j=1}^{m}\left(\frac{1}{n_K n_T}\sum_{k=1}^{n_K}\sum_{t=1}^{n_T}\left(r_{jk}-\sum_{i=1}^{n}u_{itk}x_{ij}\right)\right)*y_j}{\sum_{j=1}^{m}y_j}$$

$$and \qquad (1)$$

$$\max\ f_p=\frac{\sum_{j=1}^{m}pref\left(p_j\right)*y_j}{\sum_{j=1}^{m}y_j}$$

Subject to

$$y_j, x_{ij}\in\{0,1\}, i=1,2,\cdots,n; j=1,2,\cdots,m \qquad (2)$$

$$\text{if } s_i\to p_j, \text{then } x_{ij}=1,\forall i,\forall j \qquad (3)$$

$$\text{if} \sum_{i=1}^{n}x_{ij}\neq 0, \text{then } y_j=1, \text{else } y_j=0,\forall j \qquad (4)$$

$$\sum_{j=1}^{n}x_{ij}=1,\forall i \qquad (5)$$

$$\sum_{i=1}^{n}u_{itk}x_{ij}\leq \alpha r_{jk}y_j,\forall j,\forall t,\forall k \qquad (6)$$

Here α is the predefined risk threshold.

The objective function maximizes the average resource utilization of selected target servers and the average preference value of selected target servers. Constraint (5) ensures that each service is allocated once and constraint (6) ensures that the aggregated resource requirement of allocated services in each time section is lower than the capacity of target server (i.e., consolidation rule 1).

Furthermore, other four consolidation rules can be formulated as:

$$\sum_{i \in Q_l} x_{ij} = \left|E_l\right|,\ Q_l = \left\{ i \middle| s_i \in E_l, \forall i \right\}, \forall j, \forall l \qquad (7)$$

Where E_l represents a subset of services which satisfy the consolidation rule 2. This constraint ensures that all of the services in subset will be allocated to a same target server.

$$\sum_{i \in Q_l} x_{ij} \le 1, Q_l = \left\{ i \middle| s_i \in C_l, \forall i \right\}, \forall j, \forall l \qquad (8)$$

Where C_l represents a subset of services which meet the consolidation rule 3. This constraint ensures that each selected target server can only contain one service in subset at most.

$$\sum_{j \in J_i^+} x_{ij} = 1,\ J_i^+ = \left\{ j \middle| A^+\left(s_i\right) \subseteq A\left(p_j\right), \forall j \right\}, \forall i \quad (9)$$

$$\sum_{j \in J_i^-} x_{ij} = 0,\ J_i^- = \left\{ j \middle| A^-\left(s_i\right) \cap A\left(p_j\right) \ne \varnothing, \forall j \right\}, \forall i \quad (10)$$

IV. Optimization Technique

Formulations (1) ~ (10) define multi-objective multi-dimension bin-packing problem (MOMDBP) with multiple real world constraints. This is complete NP hard [13][14]. The genetic algorithm (GA) has proven to be an effective optimization tool for a large number of complicated problems in combinational optimization [15], [27], [28]. To apply GA to a specific problem, a solution representation, decoding procedure, fitness function and population initializing method must be defined. Due to the classical Holland GA is hard to be directly applied to grouping problem, just like MOMDBP, for the binary encoding scheme in Holland GA translate solutions to bit strings which is not efficient to handle set values. *E. Falkenauer* proposed a Grouping Generic Algorithm (GGA) to solve this problem, by improving the encoding scheme which changes the structure of the simple chromosomes from item oriented to group oriented [24]. In this work, we follow this technique and propose our own fitness function and population initializing method. The details of GGA can be found in [24].

A. Fitness Function

The solution decoding procedure determines which servers are selected. Therefore, we can directly calculate the average resource utilization of the selected servers and the average preference value of selected servers by objective function (formulation (1)). Weighting method is used in the work to convert the optimization of multiple objectives into a single objective. The fitness function likes (11).

$$F = k_1 f_r + k_2 f_p \qquad (11)$$

B. Population Initializing

Authors in [24] generate the initial populations by randomly allocate service to server based on First Fit strategy. It's not fit to this work for the heterogeneous servers which include different resource capacity and preference value. Here, we also use weighting method to build a function to represent the score of each server (See formula (11)), including preference and resource capacity.

$$W\left(p_j\right) = c_1 * \frac{pref\left(p_j\right) - \min\limits_{i=1}^{M} pref\left(p_i\right)}{\max\limits_{i=1}^{M} pref\left(p_i\right) - \min\limits_{i=1}^{M} pref\left(p_i\right)} + c_2 * \frac{1}{n_K} * \sum_{k=1}^{n_K} \frac{r_{jk} - \min\limits_{j=1}^{M} r_{jk}}{\max\limits_{j=1}^{M} r_{jk} - \min\limits_{j=1}^{M} r_{jk}} \qquad (12)$$

Here parameters c_1, c_2 are the weight of preference and resource capacity, respectively.

Then we allocate service to server based on First Fit (FF) strategy where services are randomly selected and servers are selected from the best to the worst. Constraint checking is done at each stage of assignment to avoid violation of any of the constraints. This process continues till all the services are assigned to the servers. After the initialization of the object part, group part of the chromosome is constructed.

V. Simulation, Results and Discussion

In this section we will present the results from extensive performed simulations to evaluate the proposed model and GGA. At first, experimental setup will be introduced. Next, a comparison in computational results between GGA and B&B (Branch and Bounds) will be given. Finally, we will show the impact of time interval on solution quality.

A. Simulation Data

In our work, pseudo-random numbers are used as simulation data, just like that in [4], [5] and [21]. Compared with these existing models, there are two key differences, i.e., time sharing method and different objective function. Therefore, on the one hand, the simulation data used in our work must be time series. On the other hand, the simulation data must include preference value. Besides that, as we know, the coming tasks of a service follow the Poisson process and the shape is controlled by parameter , this knowledge can be used in data generating. The detailed process of data generating can be divided into two parts, i.e. service and servers, just as follow:

Part 1: Data of Service

Step 1. Uniform distribution is used for randomly generating parameter , the interval is set to [5, 35] (because the average value of resource utilization is 20% [1]).

Step 2. Based on the parameter , the 24 hour's workload of service is randomly generated by Poisson distribution. The interval is 1 hour. Furthermore, for reducing the computational complexity, only one dimension data is generated.

Remarks: this method can be applied to multi-dimension resource without any changes;

Part 2: Data of Server

Step 1. The resource capacity of server is generated by uniform distribution, the interval is set to [100, 150].

Step 2. The preference of a server is generated by uniform distribution, the interval is set to [0.1, 1].

Furthermore, besides the two parts, some technique constraints would be randomly added to the service and server respectively.

B. Simulation Design

The simulations were designed to answer two questions, i.e., solution quality and impact of the time interval on solution quality. It should provide IT managers with guidance in what methods and model parameters are appropriate for a consolidation task. In our experiments, we use the following treatment elements:

(1) Model
(2) GGA Heuristic Algorithm
(3) Resource Type (the dimension of problem)
(4) Services (different resource requirement and some technique constraints)
(5) Servers (different resource capacity and different preference value of IT managers, some technique constraints)
(6) Time interval (i.e., 1 hour interval versus 3 hour interval)

Besides, the simulations were performed on an experimental PC with AMD Phenom II×4 945 (3000MHZ) and 2GB DDR. Moreover, Matlab 2009a is used as simulation tool.

C. Parameter of GGA

(1) The size of population: 150;
(2) Generation number: 100;
(3) Crossover rate: 0.8;
(4) Mutation rate: 0.01;
(5) Parameter in fit function: $k_1 = 0.7, k_2 = 0.3$.

D. Computational Results

In [4] and [5], solution quality refers to ratio of consolidation, i.e., how many services are allocated to the selected homogeneous servers on average. It's not fit to our work for the different objective. In this work, the number of selected servers, average resource utilization and average preference value of selected servers were used to evaluate the solution quality. We compared it between GGA and B&B in different problem size, from 100 to 800. In the comparison, we set risk threshold and used the smallest interval, i.e., 1 hour interval. GGA was repeated 10 times with different starting population. The results are given in table 1. Furthermore, we calculated the average number of service allocated to per server, the average value of resource utilization and preference based on table 1. This is given in table 2.

In table 1, results are divided into 4 blocks according to the different constraint rate, which is calculated from the number of constraint divide by the number of service. In each block, the two methods are compared in eight problem sizes and in three domains, i.e., number of server, resource utilization and value of preference. Table 2 gives us much more intuitive results. In the results, we find the average preference value of GGA is lower than that of B&B when constraint rate is 0%. Factually, the fit value of GGA is higher. This is due to the weight of resource utilization is higher than preference. Except that, in other three groups of results, GGA performed better than B&B in all of three domains.

TABLE I
COMPARISON OF SOLUTION QUALITY FOR THE GGA AND B&B IN DIFFERENT PROBLEM SIZE

Number of Service	Constraint Rate = 0%						Constraint Rate = 5%					
	GGA			B&B			GGA			B&B		
	Number of Severs	Resource Utilization	Preference	Number of Severs	Resource Utilization	Preference	Number of Severs	Resource Utilization	Preference	Number of Severs	Resource Utilization	Preference
100	12	0.67	0.78	12	0.65	0.81	12	0.65	0.76	13	0.61	0.78
200	26	0.71	0.72	28	0.66	0.77	27	0.67	0.74	29	0.58	0.74
300	41	0.66	0.76	45	0.67	0.71	42	0.64	0.68	47	0.53	0.71
400	57	0.68	0.73	62	0.63	0.73	58	0.68	0.71	64	0.62	0.72
500	72	0.68	0.69	79	0.62	0.66	74	0.65	0.72	82	0.57	0.68
600	91	0.62	0.71	97	0.58	0.76	92	0.63	0.73	99	0.59	0.65

700	104	0.63	0.69	109	0.59	0.74	105	0.61	0.68	112	0.61	0.67
800	120	0.64	0.72	124	0.61	0.72	122	0.63	0.67	126	0.57	0.68
Number of Service	**Constraint Rate = 10%**						**Constraint Rate = 15%**					
	GGA			**B&B**			**GGA**			**B&B**		
	Number of Severs	**Resource Utilization**	**Preference**	**Number of Severs**	**Resource Utilization**	**Preference**	**Number of Severs**	**Resource Utilization**	**Preference**	**Number of Severs**	**Resource Utilization**	**Preference**
100	12	0.66	0.74	13	0.62	0.72	13	0.64	0.74	14	0.58	0.72
200	27	0.64	0.73	29	0.58	0.71	28	0.61	0.73	31	0.54	0.66
300	43	0.61	0.76	48	0.56	0.68	43	0.63	0.68	49	0.57	0.68
400	59	0.65	0.68	65	0.59	0.70	60	0.58	0.71	67	0.52	0.65
500	76	0.66	0.71	82	0.54	0.72	77	0.59	0.63	83	0.49	0.68
600	93	0.62	0.66	100	0.57	0.65	93	0.56	0.66	102	0.54	0.60
700	107	0.60	0.64	113	0.59	0.61	107	0.60	0.62	114	0.45	0.64
800	122	0.63	0.66	127	0.54	0.62	123	0.58	0.64	129	0.51	0.66

TABLE II
COMPARISON OF SOLUTION QUALITY FOR THE GGA AND B&B

	Constraint Rate = 0%			**Constraint Rate = 5%**			**Constraint Rate = 10%**			**Constraint Rate = 15%**		
	Avg. Num of Service per Server	**Avg. Resource Utilization**	**Avg. Preference**	**Avg. Num of Service per Server**	**Avg. Resource Utilization**	**Avg. Preference**	**Avg. Num of Service per Server**	**Avg. Resource Utilization**	**Avg. Preference**	**Avg. Num of Service per Server**	**Avg. Resource Utilization**	**Avg. Preference**
GGA	6.883	0.661	0.724	6.767	0.645	0.711	6.679	0.634	0.698	6.618	0.599	0.676
B&B	6.475	0.626	0.738	6.294	0.599	0.704	6.239	0.574	0.676	6.112	0.537	0.661

E. Influence of the Time Interval

Time interval is a crucial variable in our work. Different intervals will lead to different solution quality. This section gives the performed simulation to show the influence of the time interval. Parameters were the same as above. 1 hour, 4 hours, 8 hours and 24 hours interval were used in simulation. In different intervals, the peak value of each interval was used as the resource demand in this interval. Figure 2 and figure 3 show the result of comparison of different intervals in the average number of allocated service per server and the fit value. The results show that, the finer the time interval, the better the solution is. Note that the 24 hours interval reduces our model (without preference) to the model proposed in [5]. We can find that, on average, the allocated services per server in our model are 22% more than that in [5].

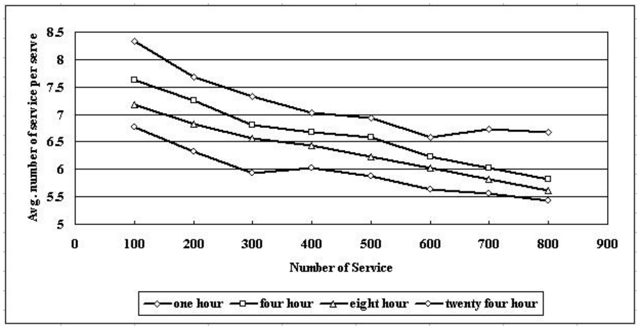

Fig. 2 Comparison of average number of allocated service per server with different time interval

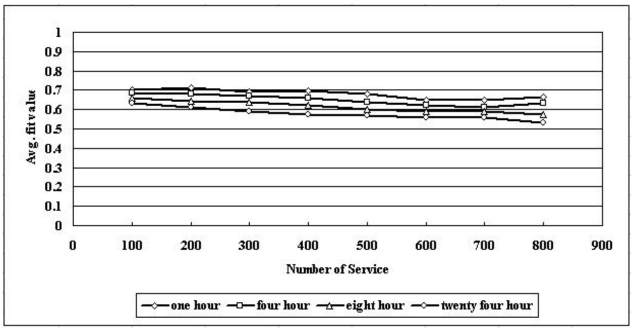

Fig. 3 Comparison of average fit value with different time interval

VI. Conclusion

This paper proposes a practical way to solve server consolidation using existing servers. Different from the existing works, we not only consider the different resource capacities and techniques, but also take into account another two important factors, i.e., servers are diversely preferred by IT manager and the peak time of each service is in different time section. The problem was formulated as a MOMDBP with constraints. An improved GGA was introduced for the problem. Extensive simulations explain that the proposed method can achieve a good solution.

Acknowledgement

This work was supported by the National High-tech Research and Development Program of China (Grant No. 2008AA04A107), the key fields Guangdong-Hongkong Technology Cooperation Funding (Grant No. 2009498B21).

References

[1] K. Parent, “Server Consolidation Improves IT’s Capacity Utilization.”, Court Square Data Group, 2005.

[2] J. Koomey, “Estimating total power consumption by servers in the US and the world.”, Final Report, 2007.

[3] GTSI White Paper, “Reducing Data Center’s Power and Energy Consumption: Saving Money and Go Green.”, 2008.

[4] R. Gupta, S. K. Bose, S. Sundarrajan, et al., “A two stage heuristic algorithm for solving the server consolidation problem with item-item and bin-item incompatibility constraints.”, In Proceedings of IEEE International Conference on Service Computing, Hawaii, USA, 2008, vol. 2, pp. 39-46.

[5] S. Agrawal, S. K. Bose, S. Sundarrajan, “Grouping genetic algorithm for solving the server consolidation problem with conflicts.”, In Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, Shanghai, China, 2009, pp. 1-8.

[6] Y. C. Lee, A. Y. Zomaya, “Energy efficient utilization of resources in cloud computing systems.”, Journal of Supercomputing, 2010.

[7] P. Padala, X. Y. Zhu, Z. K. Wang, et al., “Performance evaluation of virtualization technologies for server consolidation.”, HP Lab, 2007.

[8] VMWare White Paper, www.vmware.com

[9] P. Barham, B. Dragovic, K. Fraser, et al., “Xen and the Art Virtualization.”, In Proceedings of the 9th Symposium on Operating Systems Principles, 2003, pp. 164-177.

[10] A. Spellmann, K. Erickson, J. Reynolds, “Server consolidation using performance modeling.”, IT Professional, Vol. 5, Issue 5, pp. 31-36, 2003.

[11] K. Parent, “Server Consolidation Improves IT’s Capacity Utilization.”, Court Square Data Group, 2005.

[12] Y. Song, Y. W. Zhang, Y. Zh. Sun, “Utility analysis for Internet-oriented server consolidation in VM-based data centers.”, In Proceedings of IEEE International Conference on Cluster Computing and Workshop, 2009, pp. 1-10.

[13] C. Chekuri, S. Khanna, “On Multi-Dimensional Packing Problems.”, In Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms, 1999, pp. 185-194.

[14] M. R. Garey, D. S. Johnson, “Computer and Interactability: A Guide to the Theory of NP-Completeness”, W.H. Freeman and Company, 1979.

[15] A. L. Corcoran, R. L. Wainwright, “A generic algorithm for packing in three dimensions.”, In proceedings of the 1992 ACM/SIGAPP Symposium on Applied Computing, 1992, pp. 1021-1030.

[16] F. Benevenuto, C. Fernandes, “Performance Models for Virtualized Applications.”, ISPA 2006 Workshops, LNCS, pp. 427-439.

[17] D. Menasce, “Performance by Design: Computer Capacity Planning.”, Prentice Hall, 2004.

[18] Microsoft Virtual Server, Microsoft Corporation, http://www.microsoft.com/windowsserversystem/ virtualserver/

[19] G. Somani, S. Chandhary, “Application Performance Isolation in Virtualization.”, In Proceedings of IEEE International Conference on Cloud Computing, 2009, vol. 2, pp. 39-46.

[20] Gartner Research, “Server Consolidation: Benefits & Challenges”, 2002.

[21] Y. Aijiro, A. Tanaka, “A Combinational Optimization Algorithm for Server Consolidation.”, the 21st Annual Conference of Japanese Society for Artificial Intelligence, 2007.

[22] M. Bichler, “Capacity Planning for Virtualized Servers.”, 16th Workshop on Information Technologies and System, Milwaukee, USA, 2006.

[23] Liang Liu, Hao Wang, Xue Liu, et al., “GreenCloud: a new architecture for green data center”, In Proceedings of the 6th International conference industry session on Autonomic computing and communications industry session, 2009, pp. 29-38.

[24] E. Falkenauer, “A Hybrid Group Generic Algorithm for Bin Packing”, Journal of Heuristic, vol. 2 pp. 5-30, 2004.

[25] Sung Young Jung, Jeong-Hee Hong, Taek-Soo Kim, “A Statistic Model for User Preference”, IEEE Trans. on Knowledge and data engineering, vol.17, No. 6, 2005.

[26] S. Y. Jing, K. She, “A Rough Sets Approach to User Preference Modeling”, RSKT2010, LANI, Beijing, China, 2010.

[27] D. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning”, Reading, MA: Addison Wesley, 1989.

[28] T. Back, “Evolutionary Algorithms in Theory and Practice. Evolution Strategies”, Evolutionary Programming. Genetic Algorithms. Oxford University Press, 1996.