Truthful Strategy and Resource Integration for Multi-tenant Data Center Demand Response

Youshi Wang[1,4], Fa Zhang[2], and Zhiyong Liu[1,3](✉)

[1] Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{wangyoushi,zyliu}@ict.ac.cn
[2] Key Laboratory of Intelligent Information Processing, ICT, CAS, Beijing, China
zhangfa@ict.ac.cn
[3] State Key Laboratory for Computer Architecture, ICT, CAS, Beijing, China
[4] University of Chinese Academy of Sciences, Beijing, China

Abstract. Data centers' demand response (DR) program has been paid more and more attention recently. As an important component of data centers, multi-tenant data centers (also called "colocation") play a significant role in the demand response, especially in the emergency demand response (EDR). In this paper, we focus on how the colocation can perform better in the EDR program. We formulate the "uncoordinated relationship" in the colocation which is the key problem affecting energy efficiency, and propose a reward system to motivate tenants to join the EDR program, and a truthful strategy is developed to ensure the authenticity of tenants' information. For achieving the overall coordination, we integrate tenants' resources to increase the colocation's resource utilization and optimize the whole colocation's energy efficiency, then devise two algorithms to solve the actual resource migration and integration problem. We analyze the complexity of allocation model and two algorithms. Experimental results show that our solution is practical and efficient.

Keywords: Colocation · Emergency demand response · Uncoordinated relationship · Truthful strategy design · Algorithm analysis

1 Introduction

Data center has become an integral part of our everyday lives. As the scale of the network is continuously extending, more and more data centers have been established, and consume huge amount of energy. The latest data suggests that only in 2013 the power consumption of data centers in the United States has increased to 91 billion kilowatt-hour and the prospective consumption in 2020 will reach 138 billion kilowatt-hour [1]. Because of the huge energy consumption, the demand response program becomes more important in date centers. Especially, the emergency demand response, in some extreme cases such as extreme weather or natural disaster, is for protecting the secure of State Grid by reducing

J. Wang and C. Yap (Eds.): FAW 2015, LNCS 9130, pp. 259–270, 2015.
DOI: 10.1007/978-3-319-19647-3_24

the energy consumption. As one of the major sources of energy consumption, the control of data centers' energy consumption plays an extremely important role in the emergency demand response.

Generally, the data center can be classified into three types: private data center, colocation data center and public cloud computing data center. By the data from National Resource Defense Council (NRDC), the private data center is still the major part, accounting for 53 % of the total, although, the colocation has developed rapidly in recent years and has occupied for about 40 % of the total. NRDC estimates that in many small office-based organizations with on-premise server rooms, as much as 30 % of their total electricity use may be directed toward powering and cooling servers running 24 h a day, even when performing little or no work. The energy wasted by colocation annually is equivalent to the output of seven medium-sized coal-fired power plants [2].

However, the colocation's business model is different from other data centers: the colocation operator is responsible for energy supply and cooling, and all servers are controlled by tenants. Due to losing the direct control of servers, colocation becomes very weak for the energy consumption management. We call the dispersion of management as "uncoordinated relationship" and it has become a serious problem for colocation's energy-saving. In the colocation, because tenants pay the bill based on their peak energy, they don't care the energy consumption and always keep servers working on high performance. Moreover, due to tenants' independence and the obstacles of communication between the operator and tenants, even if each tenant wants to optimize his own energy consumption independently, it is very difficult to get the global energy consumption optimization by each tenant's independent optimization.

Statistic results from large private data centers such as Google and Amazon [3], show that rack units spend about 80 % of their time using less than 65 % of their peak power, and for the whole cluster, it never runs above 72 % of its aggregate peak power. Data analysis from "Hotmail", "MSR" and "Wikipedia" shows that the average utilization of tenant's server is only 15 % [4]. So low resource utilization has become a critical problem need to be addressed in the colocation.

The energy-saving problem of the data center has been researched widely, such as Dynamic Voltage and Frequency Scaling (DVFS) [5], energy-efficient topologies ([6–8]), virtualization of computer resources ([9–12]), and traffic engineering ([13–17]). However, almost all of these techniques work on a hypothesis that data centers are collaborative, that is, under the uncoordinated situation like the colocation, they can't be used directly to solve the energy consumption problem.

Facing the problem of the "uncoordinated relationship" and low resource utilization in the colocation, we propose a framework by combining resource integration technology and economic incentives. Firstly, we design the reward system to encourage tenants to join the EDR program actively. We evaluate each tenant's resource migration cost and then return the benefits of energy-saving based on each tenant's cost. Secondly, for bids of tenants who join the EDR, we propose a truthful strategy to ensure the authenticity of tenants' information. Then we develop two algorithms to integrate tenants' resources, and reduce the whole colocation's energy consumption by increasing resource utilization. Using the integration of tenants' resources, we achieve the unified dispatch management in the colocation, and strengthen the cooperation of the whole colocation.

To summarize, our contributions in this paper are three-fold:

- We formulate the "uncoordinated relationship" problem in the colocation which is the main reason for the colocation's energy inefficiency problem.
- To solve the colocation's "uncoordinated relationship", we propose the reward system by combining the economic incentive and the resource migration method, then design a truthful strategy to ensure the authenticity of information from tenants.
- We prove the resource integration model is an NP-complete problem. For solve it, we develop Priority-based Resource Allocation Algorithm (P-RAA) and Dynamic Resource Allocation Algorithm (D-RAA), and give the time complexity of two algorithms.

The rest of paper is organized as follows. We first propose the reward system and the truthful strategy in Sect. 2. In Sect. 3, we analyze the complexity of the resource integration model, then propose P-RAA and D-RAA to solve it and analyze their time complexity. In Sect. 4, we show and analyze our simulation experimental results. Finally, we summarize and conclude in Sect. 5.

2 Reward System and Truthful Strategy Design

In this section, we first study how to encourage tenants to join the EDR program actively, and propose a reward system to solve the amotivational problem. Then, we design a truthful strategy to ensure the authenticity of tenants' information including the resource requests and energy-saving targets.

2.1 Rewards System

Firstly, the cost of tenants can be divided into two parts, the switch loss and the performance degradation. The main source of switch loss is turning servers into sleep/off mode and bringing them back to normal operation [18]. Because the switch loss is independent of other external factors, we can use a fixed value η to denote it in one server. Then, we will focus on how to measure the degradation of performance.

When allocating resources based on tenants' requests, a bad situation is that real-time resource demand is beyond the actual load. From the view of the computer system, when actual load is beyond capacity limitation, redundant traffic will queue and wait. We use λ_i and μ_i to denote the arrive rate and processing rate of tenant i's traffic and use λ_t and μ_t to denote the actual arrive rate and actual processing rate. The theory rate is equivalent to each tenant's resource request, meanwhile, every tenant would get resources based on his request regardless of the actual traffic arrive rate, so the processing rate can be fixed, expressed as $\mu_i = \mu_t$. Under the actual situation, λ_t is fluctuating so we use λ_i as standard to estimate λ_t. An M/M/1 queue model is used to calculate the theory delay: $d_i = \frac{1}{\mu_i - \lambda_i}$ for tenant i. Tenants' actual delay could be divided into two parts by the actual traffic arrive rate:

$$\begin{cases} d_1 = \frac{1}{\mu_i - \lambda_i} \,|_t \; t \in T(\lambda_t \leq \lambda_i) \\ d_2 = \frac{1}{\mu_i - \lambda_t} \,|_t \; t \in T(\lambda_t > \lambda_i) \end{cases} \tag{1}$$

We set d_1 as benchmark. Because μ_i has been fixed by each tenant's request, the main influence on delay d_1 is traffic arrive rate λ_i. So delay d_1 can be expressed as a function only about λ_i:

$$d_1 = \frac{1}{\mu_i - \lambda_i} \Leftrightarrow \lambda_i {=} 1 \times d_{base} \tag{2}$$

where d_{base} denotes the unit delay benchmark. For d_2, we have:

$$d_2 = (\lambda_t/\lambda_i) d_{base} \tag{3}$$

Delay d_1 is determined by tenants' resource requests, so it is set as a part of performance degeneration. The total performance degeneration cost can be expressed:

$$\begin{gathered} d_i{}' = \textstyle\int_{t_1} d_2 = \int_{t_1} \frac{\lambda_t}{\lambda_i} d_{base} \\ t_1 \in T(\lambda_t > \lambda_i) \end{gathered} \tag{4}$$

For λ_t, we adopt the gaussian distribution (limiting the data within a certain range) to simulate its fluctuating. In the actual simulation, we restrict interval length within 2σ (σ denotes the standard deviation). This can ensure that the simulation data is closer to the truth.

Synthesizing the switch loss and the performance degradation, we can get a total delay expression:

$$\begin{gathered} d_{i\_cost} = m_i \times \eta + \textstyle\int_{t_1} \frac{\lambda_t}{\lambda_i} d_{base} \\ t_1 \in T(\lambda_t > \lambda_i) \end{gathered} \tag{5}$$

where m_i is the number of tenant i's servers.

The reward function, which is used to motivate tenants to join the EDR program, is based on tenant cost and energy-saving revenue. We can express the reward function as:

$$Rd_i = \frac{d_{i\_cost}}{\sum\limits_{i=1}^{M'} d_{i\_cost}} \times \sum_{i=1}^{M'} (\gamma \times pr_i) \tag{6}$$

where γ is determined by operators to measure the tenant's energy-saving revenue based on the actual energy consumption target pr_i and M' denotes the set of tenants selected.

2.2 Truthful Strategy

Since tenants lack of enough economic incentives to save energy, and they seldom or never cooperate with each other, operators need to exploit a pattern to enhance the synergy between tenants. Therefore, we encourage tenants to submit their resource demands in the form of bids and integrate tenants' requests by the operator. The bid includes two aspects: resource request $r_{i\_j,k}(j \in (1,R), k \in (1,m_i))$ (R denotes the kinds of resource) and energy-saving target pr_i. Here $r_{i\_j,k}$ means the jth resource demand of tenant i's kth server. We get an equivalent relation between the resource request $r_{i\_j,k}$ and the arrive rate λ_i and the processing rate μ_i:

$$\begin{cases} \mu_i = f_1(r_{i\_j,k}) \\ \lambda_i = f_2(r_{i\_j,k}) \end{cases} \tag{7}$$

f_1 and f_2 are both proportional functions.

Next, we talk about how to ensure the authenticity of tenants' bids. We assume that each tenant is "rational-economic man", which means that the tenant always maximizes his own benefit. At the same time, this assumption also ensures that tenants wouldn't lie purposely without any benefit. In order to maximize interests, tenants trend to get more resources and more earnings. So we need not only to avoid lying about the resource demand, but also to get a truthful energy-saving target as a constraint for resource integration. Migrating fewer resources to save more energy is an optimal solution for decreasing the overall energy consumption. We use a weight function to express this relationship:

$$\omega_i = \frac{pr_i}{\sum_{k=1}^{m_i}\sum_{j=1}^{R} r_{i\_j,k}} \tag{8}$$

From the view of operators, they would like to accept bids whose weight ω_i is higher, and from the view of tenants, they want to get higher priority by submitting higher energy conservation goals or requesting less resource. We use an optimal model, showed in next section, to calculate tenants' priorities, and apply the weight coefficient to adjust the objective function of the model. Less resource demand, more cost reduction. But the resource demand should have a lower bound to maintain the system stability and quality of service. So for every tenant, submitting his real demand is more conducive to get higher priority. However, if tenants set an excessive energy-saving goal than the actual energy-saving for getting higher priority, it may cause resource integration scheme inefficiency even losing efficacy. For avoiding this situation, we firstly define a penalty function to increase the cost of lying, and then we propose an offline algorithm P-RAA to allocate the resource based on tenants priorities, and an on-line algorithm to allocate the resource when some tenants lie about their demands.

When finishing tenant i's resource migration, the operator can get his actual energy-saving pr_{i_true}. By it, we can get the penalty function:

$$\begin{gathered} f(\beta) = |\beta(pr_i - (1+p_{c1})pr_{i\_true}), 0|_{\max} \\ \beta \propto \frac{pr_i - (1+p_{c1})pr_{i\_true}}{pr_{i\_true}} \end{gathered} \tag{9}$$

where p_{c1} is the percentage of maximum permissible error which is used to judge whether the error is factitious between the actual and the expected energy-saving, and β is the punishment coefficient which is used to decide gradations of punishment when tenants submitted inveracious energy-saving targets.

3 Model Analysis and Algorithm Design

Using prr to denote the whole energy-saving demand, we can get the energy-saving constrain:

$$\begin{gathered}\sum_{i=1}^{M} x_i pr_i \geq prr \\ x_i \in (0,1)\end{gathered} \tag{10}$$

where x_i denotes whether tenant i is selected.

Combined with the capacity constrain, we propose the selection model to determine whose bids can be accepted to join the EDR program:

$$\begin{gathered}(P_1) \min \sum_{i=1}^{M} x_i (m_i \times \eta + \int_{t_1} \frac{\lambda_t}{\lambda_i} d_{base}) \frac{1}{\omega_i} \\ s.t. \\ \sum_{i=1}^{M} x_i pr_i \geq prr \\ \sum_{i=1}^{M} (x_i \sum_{k=1}^{m_i} \sum_{j=1}^{R} r_{i\text{-}j,k}) \leq \bar{C} \\ x_i \in (0,1) \\ t_1 \in T(\lambda_t > \lambda_i)\end{gathered} \tag{11}$$

where $\overline{C}$ is the operator's capacity, M is the number of tenants. This is a mixed integer linear problem. By (P_1), we can get a selection result vector $\bar{X} = (x_1, x_2, ..., x_M)$. For ensuring that (P_1) has a solution, we need to make a hypothesis: we can find the solution based on two constraints including the energy-saving constraint and the resource capacity constraint in (P_1). This hypothesis is critical to our objective of designing the resource integration algorithm. After getting $\overline{X}$, the next objective is to design algorithms for optimizing the resources allocation which will be discussed in Subsect. 3.2.

3.1 Complexity Analysis

We now analyze the computational complexity of the optimizing resource allocation problem. Here, this problem is taken analogous to 0–1 multi-knapsack problem. A recognized result is that this model is also NP-complete, and based on this result, we propose the following Lemma 1. 0–1 multi-knapsack problem is an extension of 0–1 knapsack problem that considers multi-knapsacks rather than one knapsack, which is known to be NP-complete.

Lemma 1. *Finding the optimal solution of allocation problem based on the selection model is an NP-complete problem.*

Proof. We assume that existing a polynomial time algorithm can solve the optimality of allocation problem. The optimizing model can be seen as a process of finding minimum n, and this process includes limited attempts to find the optimal result. For every attempt, we set a value for n and then judge n severs whether can hold all requests, and we call it a subproblem. In this subproblem, there are $N = (M \times m_i)$ resource requests, and for each request, the resource demand is $r_{i\_k} = \sum_{j\in(1,R)} r_{i\_j,k}$, which is indivisible. The resource integration model is to find an optimal solution which can use minimum servers to hold all selected tenants' requests. It can be divided into multiple subproblems, and each of them is to judge whether all requests can be holden when the number of servers is fixed. We see every request as an item and the resource demand just is the item's weight. Meanwhile, operator's servers are seen as n knapsacks whose capacity are $(k_1, k_2, \ldots\ldots, k_n)$. When each request's value is 1, the resource integration model can be expressed as:

$$
\begin{aligned}
(P_2) \qquad & \max \sum_{i=1}^{n}\sum_{j=1}^{m} z_{ij}v_j \\
s.t.\ & \\
& \sum_{j=1}^{m} z_{ij}w_j \leq k_i \ 1 \leq i \leq n \\
& \sum_{i=1}^{n} z_{ij} \leq 1 \ 1 \leq j \leq m \\
& z_{ij} \in (0,1)
\end{aligned}
$$

where z_{ij} denotes whether item j is put into knapsack i, and v_j is the value of item j. This is a 0–1 multi-knapsack problem. According to the assumption, the resource integration problem can be solved within polynomial time, the subproblem should also be solved within polynomial time. We get the contradictory result $P = NP$, so the original assumption is false. Therefore, Lemma 1 is true. □

3.2 The Off-line and On-line Algorithms

For solving the resource allocation problem, we adopt the greedy algorithm to develop an offline algorithm (Priority-based Resource Allocation Algorithm (P-RAA), see Algorithm 1 and an online algorithm (Dynamic Resource Allocation Algorithm (D-RAA), see Algorithm 2. When all tenants provide their truthful demands, P-RAA can solve the resource allocation problem offline. We divide this algorithm into three parts. Firstly, we initialize the tab of each resource request and its time complexity is $O(N)$. Then in (P_1), we calculate each tenant's value based on the objective function and sort all tenants' value, and the corresponding time complexity is $O(N + M\log_2 M)$. Finally, we use the greedy algorithm to allocate resources and its time complexity is $O(M'^2)$. So the total time complexity of P-RAA is within $O(M'^2 + 2N + M\log_2 M)$, and this shows that P-RAA is a polynomial time algorithm.

A specific situation must be considered when the actual energy-saving is below the energy-saving demand prr. Under this situation, we need to accept more bids from remaining tenants. For these tenants, we adopt the sequence

Algorithm 1. Priority-based Resource Allocation Algorithm (P-RAA)

for all $i \in M$ and $j \in m_i$ **do**
  Initialize parameter $flag_{ij} = 0$; as the tab of each resource request
end for
By (P_1), choose s set of tenants who can join in the resource migration process and get the vector $\overline{X} = (\overline{x_1}, ..., \overline{x_{M'}})$
Using the greedy algorithm with suitable greed factor
for all $i \in \overline{X}$ **do**
  Allocate tenant $i's$ resources to the servers based on the greedy algorithm
  $flag_{ij} = 1$; set tenant i's resource request j as unavailable
end for
Output: The solution of all resource requests allocation

from Algorithm 1 and then use the greedy algorithm to choose bids. Based on the rank, D-RAA is designed as an online algorithm for accepting bids dynamically. Because accepting more bids may need more operator's servers, so our greed aim is to use fewer servers to satisfy energy-saving constrain. Firstly, we consider to fill in the servers which has been used in the offline algorithm, and then we would use more severs to handle remaining resources. We assume that n' tenants are selected. We don't consider the time complexity of sorting, because it has been consisted in Algorithm 1. So the time complexity of D-RAA can be expressed as $O(n' \times n)$.

Algorithm 2. Dynamic Resource Allocation Algorithm (D-RAA)

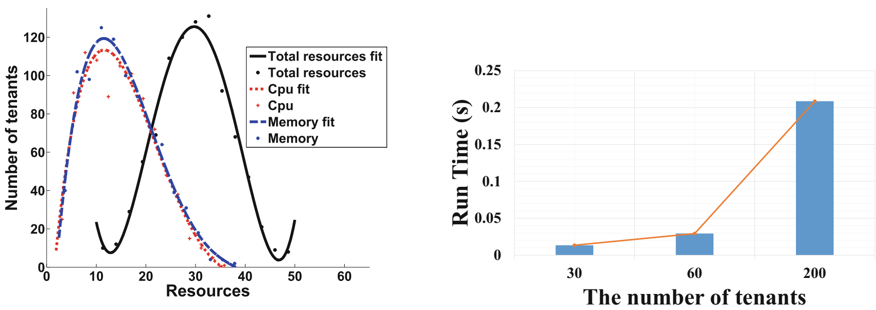
Input: The resources allocation information from **P-RAA** and the servers used is $K = (k_1, k_2, \ldots\ldots, k_n)$
Sort tenant $i (\forall i \in M'')$ based on Algorithm 1; M'' represents a set of tenants whose bids aren't accepted
Check K and get each server's resource information
while Energy conservation constraints are not satisfied **do**
  if i's bid can be put into K **then**
    Using Greedy strategy to fill K
  else
    Put i's bid into a new server in order
  end if
end while
Output: $\overline{x_i} (\forall i \in M'')$

Using algorithms P-RAA and D-RAA, we achieve a robust model for resource integration. Combined with the truthful strategy, operators can use the information to achieve the overall coordination.

4 Experimental Results

We evaluate the performance of our strategy and algorithms using the simulation data. In this work, we generate the resource demand of each tenant following Gaussian distribution under three different resource utilizations (15 %, 20 % and 30 %), and implement two strategies with different greed factors. From the experiments, we want to obtain two objectives: (1) looking for the better greed factor (Line 5 in Algorithm 1) to solve the multi-knapsack problem, (2) evaluating the performance of our algorithms in energy-saving, and showing the percentage of energy-saving of our solution in different conditions.

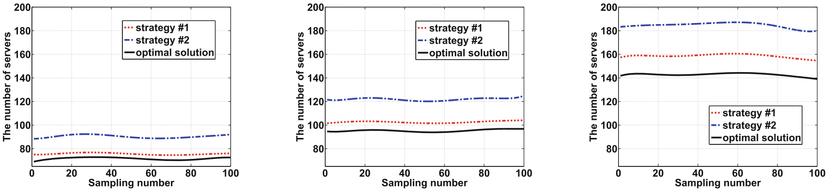
Firstly, Based on three different resource utilizations, we use the Gaussian distribution to simulate the actual total resource demand of each tenant. The main reason why choose the Gaussian distribution is that we need the characteristic of fluctuating around the mean. Since the Gaussian distribution is unbounded, we limit the value interval for avoiding negative number. Then we adopt the uniform distribution to divide each tenant's total resource demand for getting the CPU and memory demands. Figure 1(a) shows the characters of the data distribution under the resource utilization of 15 %. Then we express the feasibility of our solution. Figure 1(b) shows the running time of our solution in cases of 30, 60, 200 tenants. As shown in Fig. 1(b), the running time will be increased with the growth of tenants, but even when tenants reach to 200, the running time of our solution is only 0.2 second. For the EDR program, the response time is requested to be smallest possible, so our solution is fast enough for responding the EDR program.



(a) (b)

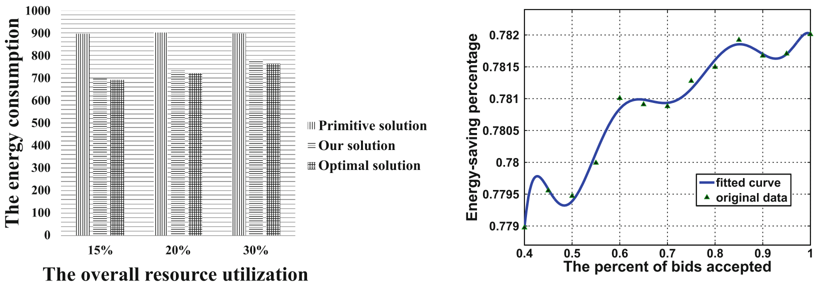
Fig. 1. (a) is the data distribution graph for the resource utilization of 15 %, (b) is the actual running time of the whole solution

Next, we verify the effect of different greed factors on performance of energy-saving. We use two strategies with different greed factors, one is to consider the size of resource requirements as the major factor, as shown with the strategy 1 in Fig. 2, the other tends to make resource requests to meet the resource supply structure of operators, as shown in strategy 2 in Fig. 2. Assuming that the resource is continuous and separable, the optimal solution can be got based on traditional linear programming with an absolute lower bound. Figure 2 shows that strategy 1 is better than strategy 2, and always under different resource utilizations, and very close to the optimal result.



(a) (b) (c)

Fig. 2. Comparison of two kinds of greedy algorithms and the optimal solution based on three factors. (a), (b) and (c) signify the different resource utilization 15 %, 20 % and 30 %.



(a) (b)

Fig. 3. Energy-saving figure: (a) is the comparison of energy-saving effect in three conditions; (b) shows the relationship between energy-saving and bids accepted

To further analyze the performance of our solution, we compare the energy consumption of our algorithm with the optimal solution, as shown in Fig. 3. In the Fig. 3(a), the actual energy consumption is very close to the optimal energy consumption, under different resource utilizations. However, how much the energy is saved depends on the accepted number of tenants' bids. Figure 3(b) describes the relationship between the accepted number of bids and the percent of energy-saving. We can find that, with the resource utilization increasing from 40 % to 100 %, our solution can get about 78 % average energy-saving. Also with the increasing of resources migrated, the energy-saving ratio takes on a growth trend but is not obvious, the reason is that more resources migrated need to take up more servers from operators. In this work, 78 % average energy-saving is relatively stable, which suggests that we can obtain an efficient energy-saving solution by our algorithms.

5 Conclusion

Because of high energy consumption, the colocation plays an irreplaceable role in the EDR program. From its characteristics, we discover that the "uncoordinated relationship" and the low resource utilization problems are the keys for improving the colocation's energy efficiency. In this paper, we design a reward system to encourage tenants to submit their resource demands and energy-saving targets. The reward system firstly evaluate each tenant's migration cost and then return the benefits of energy-saving based on the cost. For ensure the authenticity of tenants' bids, we propose a truthful strategy including the design of the weight parameter and the build of the penalty function. Then we integrate all tenants' resource demands by two algorithms: an off-line algorithm P-RAA and an on-line algorithm D-RAA. By resource integration, we achieve the unified dispatch management in the colocation, and reduce the colocation's energy consumption by improving the overall resource utilization. Finally, we analyze the complexity of the resource integration model, and show the specific time complexity expression for P-RAA and D-RAA. Experimental results show that our solution is effective on energy-saving in the colocation and fast enough for responding the EDR program.

Acknowledgement. This research was supported in part by the National Natural Science Foundation of China (Grant No. 61221062 and 61202059).

References

1. Multi-tenant data centers need to play bigger energy efficiency role (2014). http://www.datacenterknowledge.com/archives/2014/08/26/data-center-energy-efficiency-role
2. Is cloud computing always greener? (2012). http://www.nrdc.org/energy/files/cloud-computing-efficiency-IB.pdf
3. Barroso, L.A., Hölzle, U.: The datacenter as a computer: an introduction to the design of warehouse-scale machines. Synth. Lect. Comput. Archit. **4**(1), 1–108 (2009)
4. Ren, S., Islam, M.: Colocation demand response: why do i turn off my servers. In: ICAC (2014)
5. Semeraro, G., Magklis, G., Balasubramonian, R., Albonesi, D.H., Dwarkadas, S., Scott, M.L.: Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling. In: Proceedings of the Eighth International Symposium on High-Performance Computer Architecture, pp. 29–40. IEEE (2002)
6. Hou, C., Zhang, F., Anta, A.F., Wang, L., Liu, Z.: A hop-by-hop energy efficient distributed routing scheme. ACM SIGMETRICS Perform. Eval. Rev. **41**(3), 101–106 (2014)

7. Abts, D., Marty, M.R., Wells, P.M., Klausler, P., Liu, H.: Energy proportional datacenter networks. ACM SIGARCH Comput. Architect. News **38**(3), 338–347 (2010). ACM
8. Huang, L., Jia, Q., Wang, X., Yang, S., Li, B.: Pcube: improving power efficiency in data center networks. In: 2011 IEEE International Conference on Cloud Computing (CLOUD), pp. 65–72. IEEE (2011)
9. Jin, X., Zhang, F., Hu, S., Liu, Z.: Risk management for virtual machines consolidation in data centers. In: 2013 IEEE Global Communications Conference (GLOBECOM), pp. 2872–2878. IEEE (2013)
10. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. ACM SIGOPS Oper. Syst. Rev. **37**(5), 164–177 (2003)
11. Beloglazov, A., Buyya, R.; Energy efficient resource management in virtualized cloud data centers. In: Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 826–831. IEEE Computer Society (2010)
12. Mukherjee, T., Banerjee, A., Varsamopoulos, G., Gupta, S.K., Rungta, S.: Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers. Comput. Netw. **53**(17), 2888–2904 (2009)
13. Wang, L., Zhang, F., Arjona Aroca, J., Vasilakos, A.V., Zheng, K., Hou, C., Li, D., Liu, Z.: Greendcn: a general framework for achieving energy efficiency in data center networks. IEEE J. Sel. Areas Commun. **32**(1), 4–15 (2014)
14. Wang, X., Yao, Y., Wang, X., Lu, K., Cao, Q.: Carpo: correlation-aware power optimization in data center networks. In: INFOCOM: 2012 Proceedings IEEE, pp. 1125–1133. IEEE (2012)
15. Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S., McKeown, N.: Elastictree: saving energy in data center networks. In: NSDI, vol. 10, pp. 249–264 (2010)
16. Shang, Y., Li, D., Xu, M.: Energy-aware routing in data center network. In: Proceedings of the First ACM SIGCOMM Workshop on Green Networking, pp. 1–8. ACM (2010)
17. Zhang, Y., Ansari, N.: Hero: hierarchical energy optimization for data center networks. In: 2012 IEEE International Conference on Communications (ICC), pp. 2924–2928. IEEE (2012)
18. Lin, M., Wierman, A., Andrew, L.L., Thereska, E.: Dynamic right-sizing for power-proportional data centers. IEEE/ACM Trans. Network. (TON) **21**(5), 1378–1391 (2013)