A Novel Coalitional Game-Theoretic Approach for Energy-Aware Dynamic VM Consolidation in Heterogeneous Cloud Datacenters

Xuan Xiao[1], Yunni Xia[1(✉)], Feng Zeng[4], Wanbo Zheng[2(✉)], Xiaoning Sun[1], Qinglan Peng[1], Yu Guo[3], and Xin Luo[5]

[1] Software Theory and Technology Chongqing Key Lab, Chongqing University, Chongqing 400044, China
xiayunni@hotmail.com
[2] Data Science Research Center, Faculty of Science, Kunming University of Science and Technology, Kunming 650500, China
zwanbo2001@163.com
[3] School of Public Administration, Sichuan University, Chengdu 610064, China
[4] Discovery Technology (shenzhen) limited, Shenzhen, China
[5] Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China

Abstract. Server consolidation technique plays an important role in energy management and load-balancing of cloud computing systems. Dynamic virtual machine (VM) consolidation is a promising consolidation approach in this direction, which aims at using least active physical machines (PMs) through appropriately migrating VMs to reduce resource consumption. The resulting optimization problem is well-acknowledged to be NP-hard optimization problems. In this paper, we propose a novel merge-and-split-based coalitional game-theoretic approach for VM consolidation in heterogeneous clouds. The proposed approach first partitions PMs into different groups based on their load levels, then employs a coalitional-game-based VM consolidation algorithm (CGMS) in choosing members from such groups to form effective coalitions, performs VM migrations among the coalition members to maximize the payoff of every coalition, and close PMs with low energy-efficiency. Experimental results based on multiple cases clearly demonstrate that our proposed approach outperforms traditional ones in terms of energy-saving and level of load fairness.

Keywords: Energy-aware · Dynamic VM consolidation · Load fairness · Merge-split method · Coalitional game · Heterogeneous clouds

1 Introduction

Nowadays, cloud computing is becoming an increasingly popular computational paradigm featured by the ability to provide elastic services over the internet

J. Miller et al. (Eds.): ICWS 2019, LNCS 11512, pp. 95–109, 2019.
https://doi.org/10.1007/978-3-030-23499-7_7

for a huge number of global users. With the rapid growth of cloud services, cloud infrastructures and their supporting datacenters are becoming increasingly complex, energy-requiring, and expensive with varying resource configurations and heterogeneous architectural setups. According to [1], electricity demand for world-wide datacenters is expected to increase by over 66% over the period of 2011–2035. Hence, resource and energy management become major concerns of both cloud providers and users. However, today's datacenters are still limited in ways of effectiveness of energy efficiency and energy management strategies. Among various energy management and energy saving technologies, dynamic VM consolidation is one of the most effective ones. Consolidation refers to the live migration operations of VMs between hosts with slight performance loss [2]. The aim of dynamic VM consolidation is to reduce the energy consumption of consolidation activities through live migration of VMs instead of static or planned ones. It is capable of turning idle active PMs into sleeping mode for energy saving. This technique considerably improves resource utilization and energy efficiency. In this work, we propose a novel energy-aware and merge-and-split-based coalitional game-theoretic approach for dynamic VM consolidation for heterogeneous cloud with varying resource configurations. The proposed approach involves multiple steps: (1) dividing PMs into three groups based on their workloads, (2) performing a coalitional game to improve the utilization, (3) letting PMs compete with each other and forming coalitions by using merge and split operations. To validate our proposed approach, we conduct extensive simulative studies based on multiple cases and show that our approach clearly outperforms traditional ones in terms of energy-saving and load fairness.

2 Literature Review

2.1 VM Consolidation Algorithms for Energy Management

Recently, considerable research efforts have been paid to the VM consolidation and related energy performance optimization problems. Related methods fall into two major categories, namely the dynamic server provisioning methods and dynamic VM consolidation ones [9]. The latter refers to the technique of reallocating VMs using live migration according to their real-time resource demand and switching idle hosts to the sleep mode. Various consolidation methods are heuristic-based or meta-heuristic-based. E.g., Buyya *et al.* [3] proposed a consolidation mechanism using two fixed threshold values calculated based on processors' utilization rates. He *et al.* [4] proposed an local-regression-based algorithm featured by a combination of local regression algorithm with the minimum-migration-time VM selection policy. Huang *et al.* [5] proposed a M-Convex VM consolidation method based on the semi-quasi M-convex optimization framework, which is capable of adaptively adapting its solutions according to the optimization objectives. Murtazaev *et al.* [6] developed the Sercon framework and considered an all-or-none migration strategy, where all the VMs in one active PM are tentatively migrated to other active PMs. If the migration is successful, a new placement scheme with a reduced number of active PMs is performed.

The above operation is iterated until no improvement can be made. Farahnakian *et al.* [7] used an online optimization metaheuristic algorithm called Ant-Colony-System to find near-optimal solutions for dynamic consolidations and showed that their proposed approach achieved good energy savings while meeting quality-of-service(QoS) constraints. They defined a multi-objective function which considers both the number of dormant PMs and the number of migrations. Wu *et al.* [8] proposed an improved-group-genetic-algorithm-based VM consolidation method to optimize trade-off between migration costs and energy consumption in heterogeneous clouds. Zhang *et al.* [9] presented a heterogeneity-aware resource monitoring and management system that is capable of performing dynamic capacity provision in heterogeneous datacenters. Duan *et al.* [10] proposed an improved ant-colony algorithm for energy-efficiency optimization by leveraging a prediction model based on fractal mathematics and a scheduler based on an improved ant colony algorithm.

2.2 Game-Theoretic Scheduling in Cloud

Recently, it is shown that game theory models and related methodologies can be effective in dealing with multi-constraint-multi-task scheduling and planning problems. Game-theoretic algorithms are featured by low time-complexity in comparison with heuristics, and thus can be highly suitable for scheduling and managing time-critical cloud systems. Extensive efforts were paid in this direction. E.g., Guo *et al.* [12] used a cooperative game model to guide VM consolidation with load and energy constraints, which is tested in a homogeneous cloud environment. Paul [13] proposed an uncooperative game-theoretic algorithm for dynamic VM consolidation problem in cloud computing. Xue *et al.* [14] used a coalitional game model to schedule the tasks in cloud. They proposed the merge-and-split-based mechanism to reduce the cost of tasks execution and increase the profit of cloud resource providers. Guazzone *et al.* [15] devise an algorithm, based on cooperative game theory that allows a set of cloud providers to cooperatively set up their federations in such a way that their individual profit is increased with respect to the case in which they work in isolation. A careful investigation into above contributions suggests that they are still limited in several ways: (1) most existing works considered energy-reduction and migration-cost-saving as objectives. However, the tradeoff between load fairness and energy-saving in heterogeneous clouds was less studied [20]; (2) various works aimed at closing as many PMs as possible in optimizing energy efficiency. However, it can be misleading and problematic to do so due to the fact that PMs in heterogeneous clouds are with varying energy-consumption characteristics and turning off fewer energy-requiring PMs may be more attractable than turing off more energy-saving ones. and (3) various existing works address cloud heterogeneity by considering heterogeneous PMs and VMs while ignoring the heterogeneity of workloads. However, it should be noted that in reality workloads can be heterogeneous as well [16,17]. Our proposed method therefore aims at appropriately addressing the above issues and overcoming related limitations.

3 System Model

As widely acknowledged [3,4], the power consumption of a PM, $P(u)$, is mainly decided by its resource utilization u according to (1). In (1), P_{max} denotes the energy-consumed by a fully-loaded PM, and α denotes the proportion of idle time of a PM.

$$P(u) = \alpha P_{max} + u(1-\alpha)P_{max} \tag{1}$$

According to [3], α is usually around 0.7. Note that the utilization of a CPU can be time-varying, we thus use $u(t)$ instead of u in (2). The total energy consumed, denoted by ξ, can be estimated through an integration form as (2), where t_0 denotes the starting time, and T the period during which a PM is running.

$$\xi = \int_{t_0}^{t_0+T} P(u(t))\, dt \tag{2}$$

It is assumed a datacenter has m types of heterogeneous machines, t_s is the time that the VM consolidation starts, and t_e is the time that VM consolidation ends. f_k is the energy consumed by a PM of type k per unit time. Let b_k denotes the energy consumed by all the machines of type k per unit time before consolidation. We have:

$$b_k = n_k * \int_{t_s-T}^{t_s} f_k \tag{3}$$

where n_k denotes the number of machines of the k^{th} type. Let a_k denotes the energy consumed by all the machines of type k per unit time after consolidation and it can be similarly calculated as:

$$a_k = n_k * \int_{t_e}^{t_e+T} f_k \tag{4}$$

Next, we should consider the energy consumed by VM migrations in a consolidation process. h represents the energy consumed by migration. In this paper, we adopt the function of migration-cost proposed by [12]. It is caculated by (5).

$$h = \int_{t_s}^{t_e} \Delta P_s(t)\, dt + \int_{t_s}^{t_e} \Delta P_d(t)\, dt + q \tag{5}$$

where $\int_{t_s}^{t_e} \Delta P_s(t)$ and $\int_{t_s}^{t_e} \Delta P_d(t)$ are the increased energy consumption of the source and destination PM respectively. q is the increased energy consumption as a result of turning on a PM, which is a constant value. If we do not need to turn on a new PM as the destination PM, when a VM is migrated, then $q = 0$. Based on the above assumptions and configurations, the problem we are interested in can thus be formulated in (6).

$$\begin{aligned} &Max \quad S = \int_{t_s}^{t_e} \textstyle\sum_{k=1}^{m}(b_k - a_k) - h \\ &s.t. \sum_{j=1}^{m} d_{ij} = 1, j = 1, 2, 3 \ldots, u_j > 0 \end{aligned} \tag{6}$$

where d_{ij} is a boolean variable to indicate whether the i^{th} VM is placed on the j^{th} PM. If the i^{th} VM is placed on the j^{th} PM, then let $d_{ij} = 1$; otherwise, $d_{ij} = 0$. u_j is the utlization of PM_j, and PM_j shouldn't be an empty PM. S denotes the energy saved by the VM consolidation approach. The above formulation aims at maximizing the energy saved by the VM consolidation approach, i.e., energy saved by consolidation with the constraints that every VM can only be placed on one PM and there is no idle PMs.

4 The Coalitional Game-Theoretic Approach

According to [21,22], a coalitional game Γ consists of two essential elements as shown in (7): (1) a set of players $N = \{1, 2 \ldots\}$, in this paper, PMs are modelled as players; (2) a characteristic value v that specifies the value created by different subsets of the players. i.e., the payoff of a coalition C. Here maximizing the payoff $v(C)$ means maximizing the energy-efficiency of a coalition.

$$\Gamma = (N, v) \tag{7}$$

Players of the game choose to join or not to join a coalition by deciding whether more energy-saving could be achieved. To facilitate the handling of the coalitional game over coalitions of PMs, we first partition PMs into three groups, i.e., E, H, and L, which contains PMs with extrahigh load, high load and low load respectively, according to two load thresholds, i.e., t_1 and t_2:

$$t_1 = Q_1, \ t_2 = Q_3 \tag{8}$$

where t_1 equals Q_1, which denotes the first quartile of the workloads placed on all PMs. t_2 equals Q_3, which denotes the third quartile of the workloads placed on all PMs. In our proposed algorithm, the merge-and-split-based coalitional games are performed to maximize v of any coalition, i.e., payoff, as shown in (9). We define the utilization of a coalition as v which equals the average utilization of PMs in the coalition C except the PMs with extrahigh load.

$$\begin{aligned} &Max \ \ v \\ &v = \frac{1}{n} \sum_{j=1}^{n} u_j \\ &s.t. \ \ 0 < u_j \leq x_j, \ \mathrm{PM}_j \notin E, \ \mathrm{PM}_j \in C \end{aligned} \tag{9}$$

where u_j denotes the real-time utilization of PM_j. x_j is the maximum utilization permitted of PM_j. n is the number of PMs in the coalition except the PMs with extrahigh load. In a coalitional game, the merge operation refers to grouping multiple PMs into a single coalition. The split operation works in the opposite direction, where workload from an extra-highly-loaded PM is distributed through multiple PMs. Only on condition that the payoff v, i.e. the energy-efficiency of a coalition is higher than the average one of all coalition members when

they are running individually, the PMs are merged to form a coalition. (10)-(a)/(b)/(c)/(d) denote the precondition for the merge of an extra-highly-loaded PM and a lowly-loaded PM, the split of an extra-highly-loaded PM, the merge of lowly-load PMs, and the merge of PMs with high load, respectively.

$$\begin{aligned}
&(a)\ \forall PM_j \in E, PM_i \in L, C = \{PM_i, PM_j\} \\
&v(C) > mean(u_j, u_i) \\
&(b)\ \forall PM_j \in E, u_j < v(C), \\
&C = \{PM_i, PM_k\}, PM_i, PM_k \in L/H \\
&(c)\ \forall PM_j, PM_i \in L, C = \{PM_i, PM_j\} \\
&v(C) > mean(u_j, u_i) \\
&(d)\ \forall PM_j, PM_i \in H, C = \{PM_i, PM_j\} \\
&v(C) > mean(u_j, u_i)
\end{aligned} \tag{10}$$

where u_i denotes the utilization of PM_i. Note that the operations enabled by the (a)(b)(c)(d) preconditions happen with the alphabetic order of these preconditions to ensure that PMs with extrahigh/low load are handled before those with high load. The steps of the above operations are implemented through Algorithm 1. Figure 1(a) illustrates a typical example of three kinds of merge operations. As can be seen, VM_{1-5} are on an extra-highly-loaded PM while VM_{25} is on a PM with low load, according to the algorithm, the two PMs are thus merged in a coalition and then form two highly-loaded PMs. VM_{29-30} are

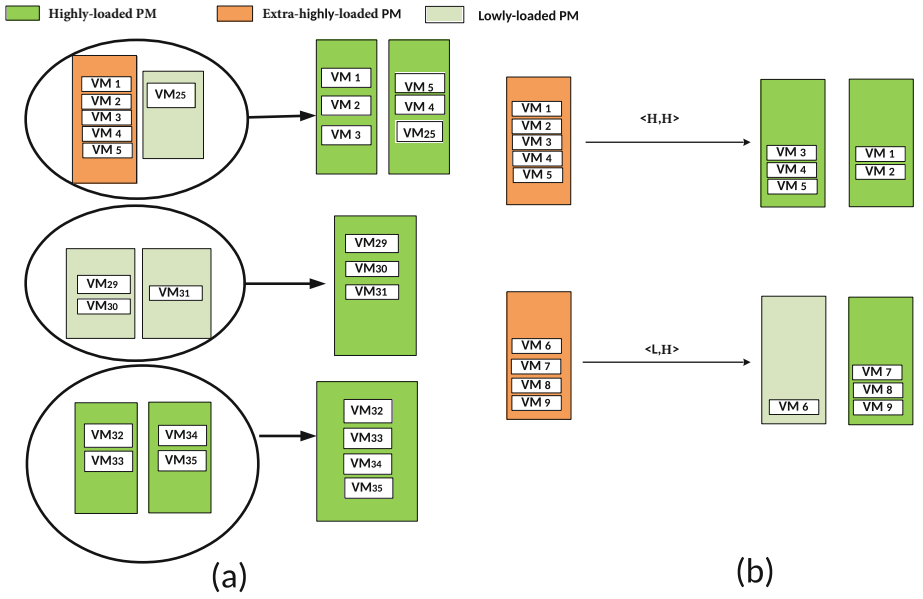


Fig. 1. Merge-and-split-based method of coalition formation

Algorithm 1. CGMS(E, H, L) algorithm

```
Input: E,H,L
Output: Updated E,H,L
Step 1:
for each PM_j in E do
    for each PM_i in L do
        if PM_i, PM_j can be merged according to (10)-a then
            migrate the source VM from PM_j to PM_i ;
        end
    end
end
Step 2:
for each PM_j in E do
    if PM_j can be split according to (10)-b then
        migrate the source VM from PM_j to a new PM;
    end
end
Step 3:
for each PM_j, PM_i in L do
    if PM_i, PM_j can be merged according to (10)-c then
        migrate the source VM from PM_j to PM_i;
        if PM_j is empty then
            close PM_j
        end
    end
end
Step 4:
for each PM_j, PM_i in H do
    if PM_j, PM_i can be merged according to (10)-d then
        migrate the source VM from PM_j to PM_i;
        if PM_j is empty then
            close PM_j
        end
    end
end
```

on a lowly-loaded PM while VM_{31} is on another PM with low load, the two lowly-loaded PMs are thus merged in a coalition and then form a PM with high load. VM_{32-33} are on a PM with high load while VM_{34-35} are on another PM with high load, the two highly-loaded PMs are thus merged in a coalition and then form a PM with high load. In Fig. 1(b), only extra-highly-loaded PMs undergo split operations. As can be seen, VM_{1-6} are on an extra-highly-loaded PM. This PM is thus splitted to two PMs with high load, which contain $VM_{3,4,5}$ and $VM_{1,2}$ respectively. After the game, numbers of extra-highly-loaded and lowly-loaded PMs are reduced while that of the PMs with high load is increased, thereby consolidating tasks into a reasonable number of PMs while avoiding both

waste of resources caused by idle PMs and potential performance degradations of extra-highly-loaded PMs. The aim of the coalitional game is thus to finally form a PM group G that contains PMs which are working in a high-efficiency state for saving energy.

$$G = \{PM_j \mid PM_j \in H \wedge u_j <= x_j\} \tag{11}$$

The coalition can be gradually formed by using Algorithm 1. Note that in lines 5, 12, 18, 27 in the pseudo codes stipulate that the resulting load of the destination PM is still subject to the load constraint, i.e., a PM should not be extra-highly-loaded. We consider d as the measure of load fairness.

$$d = (n_E + n_L)/n_H \tag{12}$$

where n_E, n_L, n_H are the number of PMs in E, L, and H, respectively. According to (12), a lower d indicates better load fairness. In this work, we consider load fairness [16,17,20] as an important metric and the optimization algorithm aims at fairly distributing workloads among PMs to aviod hotspots.

5 An Illustrative Example of CGMS

Example Analysis. We consider the example shown in Fig. 3 as an illustrative example of the effect of the merge-and-split process: a datacenter contains multiple PMs, whose indexes are shown in the X-label. The workload of each PM is based on CoMon workload traces [18] collected from 10 days during march and April 2011, which is collected from roughly 400–450 active PlanetLab nodes every 5 min within 10 days. Every PM contains 4 VMs with varying workloads as shown in Fig. 2. According to the workload data and (8), t_1 and t_2 are set as 20 and 60, respectively. As shown in Fig. 3, $L/H/E$ groups are marked blue/green/red. During the process, lowly-loaded and extra-highly-loaded PMs are turned into PMs with high load. The new PMs in H are marked by purple in Fig. 3(c)(d). The new PMs in L are marked by black in Fig. 3(c). As can be seen in Fig. 4(a), H is enlarged while E and L shrink. Thus, the overall energy efficiency is optimized

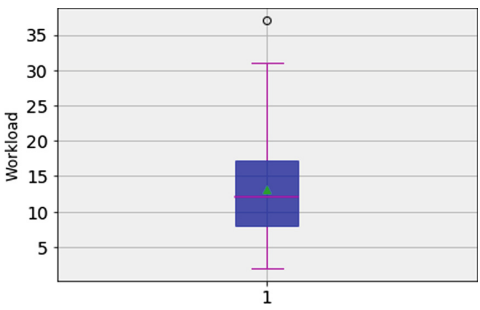


Fig. 2. VM workload used in example

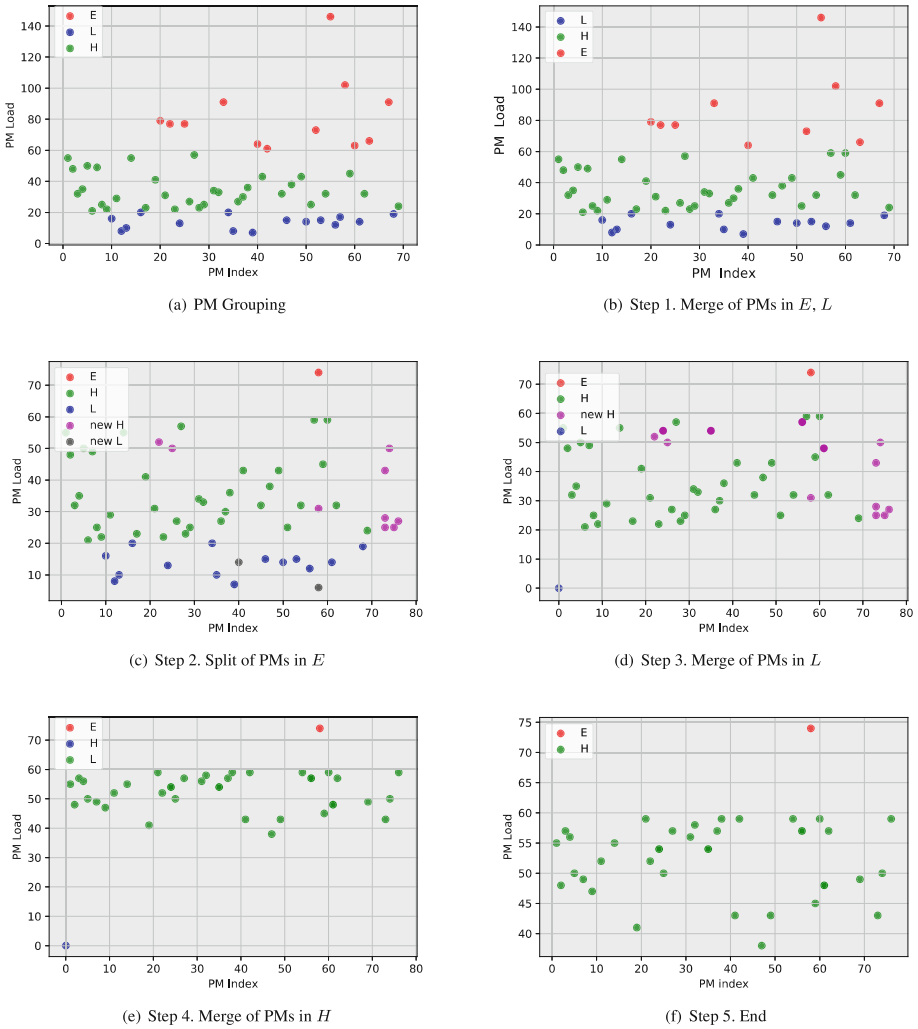
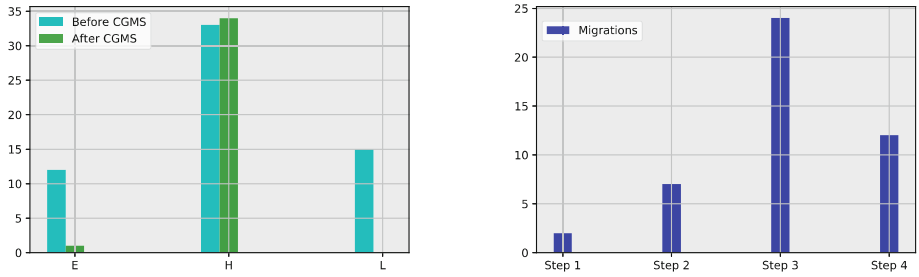


Fig. 3. An example of CGMS (Color figure online)

while the workload constraint for PMs is kept. Finally, number of migrations of every step is shown in Fig. 4(b). It is obvious that if a datacenter contains a lot of lowly-loaded PMs, a great number of VM migrations is required.

Time Complexity Analysis. The overall computational complexity of our approach can be analyzed by examing the group, merge, and split operations. In our algorithm, assuming that the number of PMs is g, the group operation's time complexity is $O(g)$. In step 1 assume the number of extra-highly-loaded PMs is y, number of lowly-loaded PMs is z, thus in step 1 the time complexity



(a) Change of number of PMs in E, H, L

(b) Migrations of every step in CGMS

Fig. 4. Example analysis

is $O(yz)$. In step 2, assume the number of involved extra-highly-loaded PMs is w, as only extra-highly-loaded PMs are involved in this step, thus the time complexity is $O(w)$. Assume number of lowly-loaded PMs involved in step 3 is r, number of PMs with high load involved in step 4 is s. Thus, we can figure out that the time complexity of step 3 and 4 is $O(s+r)$. Finally, the time complexity is $O(g + yz + w + s + r)$ totally.

6 Simulation and Evaluation

To validate our work, we implement a python-based VM consolidation system simulator, apply the algorithm in managing multiple heterogeneous PMs as given in Table 1. The energy consumption of each PM type is based on the Energy-Star-List [19]. Table 2 shows the VM types. The workload of each VM is based on CoMon workload traces. We consider VM load level of three scenarios: S1, S2, S3, plotted in Fig. 5. Each case is tested for 100 trials. Our proposed algorithm CGMS is compared with baseline approaches: Sercon(server consolidation) [6],

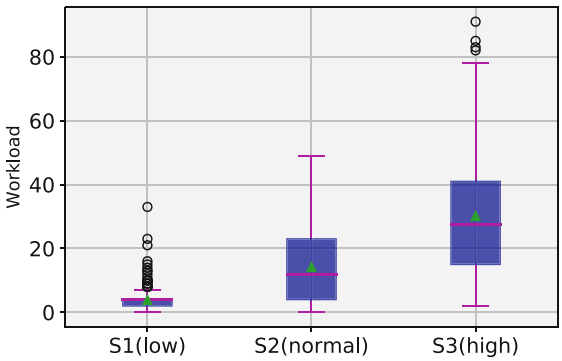


Fig. 5. Workload used in experiments

IGGA (improved group genetic algorithm) [8], and CGHO [12] (cooperative game in homogeneous cloud). Sercon is kind of improved greedy method to decrease the energy-cost and migration-cost, which inherits some properties of First-Fit and Best-Fit. Sercon used a migration threshold to control the migration efficiency. IGGA is kind of metaheuristic method using an improved genetic algorithm for VM consolidation. CGHO is another cooperative-game-theoretic algorithm tested in a homogeneous environment.

Table 1. VM configurition

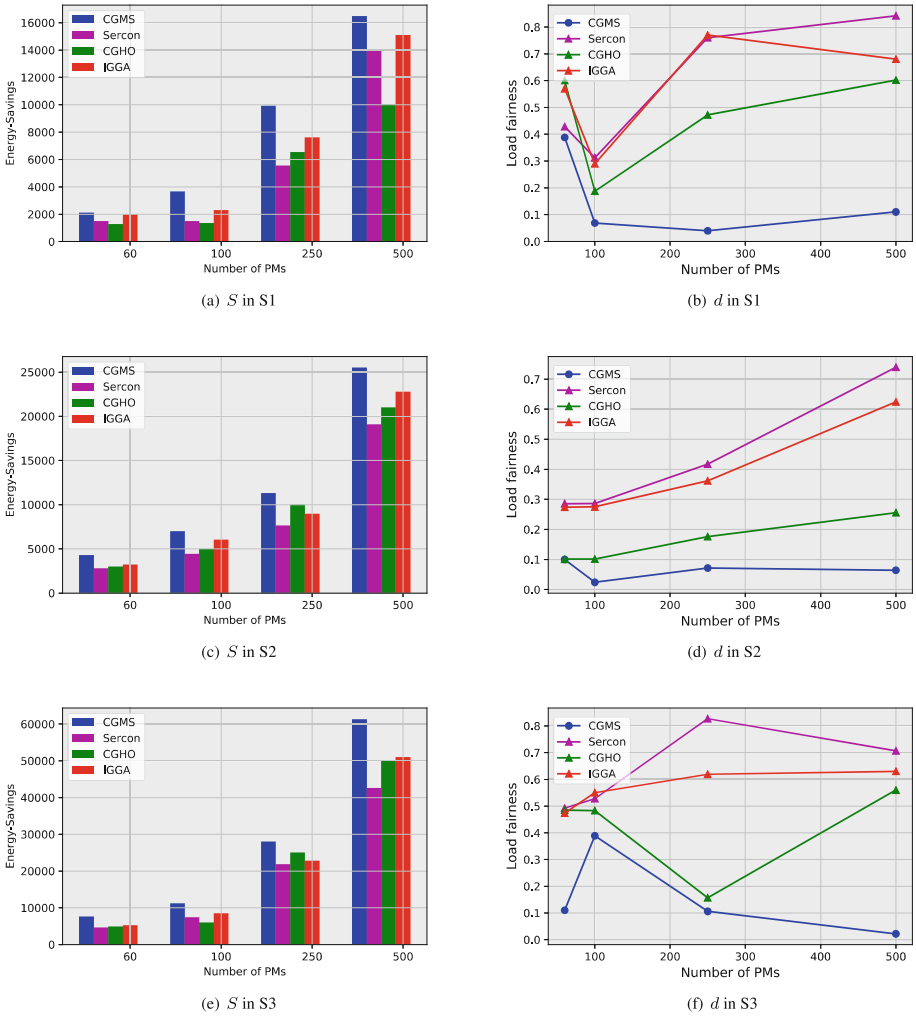
VM instance	Memory	CPU
Micro	613 M	500 MIPS
Small	1.7 GB	1000 MIPS
Large	3.75 GB	2500 MIPS
Extra large	4 GB	2500 MIPS

Table 2. PM configurition

PM instance	DELL R515	HP DL380G8	HP DL585G7
Memory	16 GB	32 GB	64 GB
Idle power	213 W	109 W	258 W
Peak power	420 W	276 W	396 W

Energy-Saving and Load Fairness. We first evaluate the energy-saving, i.e., S modelled in (6), and load fairness, i.e., d in (12), between CGMS and baseline algorithms. As shown in Fig. 6(a) (c) (e), when the number of PMs ranges from 60 to 500, our method achieves higher energy-saving (32.30% higher than Sercon in three scenarios on average; 20.03% higher than CGHO on average; and 14.28% higher than IGGA on average). The energy-saving increases with the number of PMs and outperforms baseline ones as well. As shown in Fig. 6(b)(d)(f), CGMS achieves better load fairness (85.71% lower than Sercon in three scenarios on average; 42.02% lower than CGHO on average; and 70.32% lower than IGGA on average) in all scenarios with varying PM numbers.

Computational Cost. Fig. 7 depicts the required runtime of each approach. With increase of N, the runtime of CGMS and CGHO increase slowly. Sercon is the fastest one, due to the characteristic of greedy heuristic algorithm. IGGA is a meta-heuristic algorithm. Its runtime rises smoothly with the number of PMs going up. As a result, CGMS keeps a relatively low cost, which is acceptable for most datacenters in different scales.



(a) S in S1 (b) d in S1

(c) S in S2 (d) d in S2

(e) S in S3 (f) d in S3

Fig. 6. Algorithm comparison in S1, S2, S3

The Number of Migrations. As shown in Table 3, we clearly see that Sercon achieves the least number of migrations in most cases, because it employs a greedy strategy in deciding when and which to migrate. However, it achieves the worst energy-saving. In contrast, CGMS achieves the second-least migrations (13.90% lower than CGHO on average; and 8.82% lower than IGGA on average) while clearly outperforms Sercon in term of energy-saving.

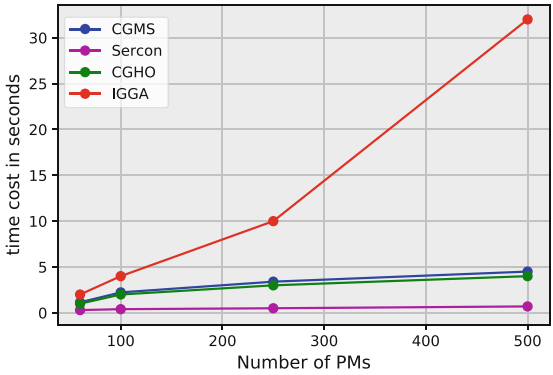


Fig. 7. Computation time of each approach

Table 3. Migrations in S1, S2, S3

Scenarios	Number of PMs	CGMS	IGGA	Sercon	CGHO
S1	60	34.6	37.5	30.1	40.2
	100	167.2	193.5	100.1	192.4
	250	322.2	405.2	226.3	365.0
	500	630.0	721.0	461.4	621.5
S2	60	16.8	15.3	15.6	24.9
	100	45.4	52.1	39.1	57.4
	250	169.0	185.6	137.8	190.4
	500	154.9	194.0	190.2	255.1
S3	60	18.3	16.1	12.0	30.4
	100	37.0	36.1	29.8	50.2
	250	151.8	170.5	155.3	181.8
	500	246.0	275.3	262.2	256.8

7 Conclusion and Future Work

In this work, we present a coalitional game approach for optimizing the energy efficiency of VM consolidation in heterogeneous cloud datacenters. The experiments results demonstrate that our approach clearly outperforms traditional approaches in terms of energy-saving and load-balancing. The following issues should be addressed as future work: (1) reducing migrations, number of migrations is expected to be optimized for a better level. (2) fault tolerance, it is promising to develop the fault tolerant mechanism based on our approach.

Acknowledgment. This work is supported in part by the International Joint Project funded jointly by the Royal Society of the UK and the National Natural Science Foundation of China under grant 61611130209, National Science Foundations of China

under grants Nos.61472051/61702060, the Science Foundation of Chongqing under No. cstc2017jcyjA1276, China Postdoctoral Science Foundation No. 2015m570770, Chongqing Postdoctoral Science special Foundation No. Xm2015078, and Universities' Sci-tech Achievements Transformation Project of Chongqing No. KJZH17104, Chongqing grand R&D projects Nos. cstc2017zdcy-zdyf0120 and cstc2017rgzn-zdyf0118. Yunni Xia and Wanbo Zheng are the corresponding authors of this work.

References

1. World Energy Outlook 2013 Fact Sheet. http://www.tp-ontrol.hu/index.php/TP_Toolbox
2. Varasteh, A., Goudarzi, M.: Server consolidation techniques in virtualized datacenters: a survey. IEEE Syst. J. **11**(2), 772–783 (2017)
3. Beloglazov, A., Buyya, R.: Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud datacenters. In: Proceedings of the 8th International Workshop on middleware for Grids, Clouds and e-Science ACM, pp. 1–6 (2010)
4. Beloglazov, A., Buyya, R.: Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud datacenters. Concurr. Comput. Pract. Exp. **24**, 1397–1420 (2012)
5. Huang, Z., Tsang, D.H.K.: M-Convex VM consolidation: towards a better VM workload consolidation. IEEE Transact. Cloud Comput. **4**, 415–428 (2016)
6. Murtazaev, A., Oh, S.: Sercon: server consolidation algorithm using live migration of virtual machines for green computing. IETE Techn. Rev. **28**(3), 212–231 (2011)
7. Farahnakian, F., et al.: Using ant colony system to consolidate VMs for green cloud computing. IEEE Transact. Serv. Comput. **8**, 187–198 (2015)
8. Wu, Q., Ishikawa, F., Zhu, Q., Xia, Y.: Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters. IEEE Transact. Serv. Comput. **1**(1), 99 (2016)
9. Zhang, Q., Zhani, M.F., Boutaba, R., Hellerstein, J.L.: Dynamic heterogeneity-aware resource provisioning in the cloud. IEEE Transact. Cloud Comput. **2**, 14–28 (2015)
10. Duan, H., et al.: Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems. Future Gener. Comput. Syst. **74**, 142–150 (2017)
11. Bharathi, P.D., Prakash, P., Kiran, M.V.K.: Energy efficient strategy for task allocation and VM consolidation in cloud environment. In: 2017 Innovations in Power and Advanced Computing Technologies, i-PACT 2017, pp. 1–6, January 2017
12. Guo, L., et al.: A game based consolidation method of virtual machines in cloud datacenters with energy and load constraints. IEEE Access. **6**, 4664–4676 (2018)
13. Paul, A.K., Sahoo, B.: Dynamic virtual machine placement in cloud computing. Indian J. Sci. Technol. 9(29) (2015)
14. Xue, F., Wu, Z.: Cloud tasks coalitional game scheduling based on merge and split mechanism. Comput. Eng. Des. (2018)
15. Guazzone, M., Anglano, C., Sereno, M.: A game-theoretic approach to coalition formation in green cloud federations. In: IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing IEEE, pp. 618–625 (2014)
16. Guo, M., Guan, Q., Ke, W.: Optimal scheduling of VMs in queueing cloud computing systems with a heterogeneous workload. IEEE Access. **6**, 15178–15191 (2018)

17. Ruiu, P., et al.: Workload management for power efficiency in heterogeneous datacenters. In: Proceedings - 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2016, pp. 23–30 (2016)
18. Park, K., Pai, V.S.: CoMon: a mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Oper. Syst. Rev. **40**(1), 65–74 (2006)
19. Energy Star Computer Server Qualified Product List. https://www.energystar.gov/ia/products/prod_lists/enterprise_servers_p_prod_list.xls
20. Xu, M., Tian, W., Buyya, R.: A survey on load balancing algorithms for virtual machines placement in cloud computing. Concur. Comput. Pract. Exp. **29**(1), e4123 (2016)
21. Myerson, R.B.: Game Theory, Analysis of Conflict. Harvard University Press, Cambridge (1991)
22. Saad, W., Han, Z., Debbah, M., Hjorungnes, A., Basar, T.: Coalitional game theory for communication networks: a tutorial. IEEE Sig. Process. Mag. **26**(5), 77–97 (2009)