EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

Bulk-bin-packing based migration management of reserved virtual machine requests for green cloud computing

Saikishor Jangiti[1], Subramaniyaswamy V[2] and Shankar Sriram V. S.[1,*]

[1]Center for Information Super Highway (CISH),
[2]School of Computing,
SASTRA Deemed University, Thanjavur, Tamilnadu, India.

Abstract

The dynamic consolidation of Virtual Machines (VMs) into a minimum number of Physical Machines (PMs) is a key energy-efficient practice in a cloud data centre, to reduce the running PMs and save electricity costs. We proposed a migration based VM consolidation approach for reserved requests. Real Dataset EC2 was used in the simulation experiments. The proposed BBPMM has demonstrated the elastic capability of adjusting the running PMs and it reduced 38% of running PMs in a reservation transition period.

Keywords: VM Consolidation, reserved instances, standard instances, bulk-bin-packing, cloud computing, energy-efficiency.

Received on 07 May 2019, accepted on 29 May 2019, published on 23 July 2019

doi: 10.4108/eai.11-6-2019.159605

[*]Corresponding author. Email:sriram@it.sastra.edu, jsaikishor@gmail.com

1. Introduction

The poor resource utilisation rates in data centres motivated towards server consolidation, using an existing technology called virtualisation, to improve utilisation rates by realising multiple virtual servers out of a single physical server. The power consumption of servers is high, and even when idle they consume 50% of their maximum power consumption [1]. In addition to the increase in resource utilisation, server consolidation resulted in economic and environmental benefits like low resource provisioning cost and reduction of harmful carbon footprints by reducing the required number of physical servers. The virtual servers can be migrated dynamically from physical servers [2].

Server consolidation also motivated virtual resource provision over the Internet called cloud computing (infrastructure as a service). Parts of the Physical Machine (PM) resources are provided to customers as Virtual Machines (VMs) according to their requirements. The scenario led to a new resource allocation problem called efficient VM initial placement in Cloud Data Centers (CDCs). In a CDC hosting homogenous PMs, the efficiency of a VM initial placement method is quantified by the number of PMs used to host the VM requests (the less the PMs used, the more efficient the placement method is). Also, the PMs should not overload to minimize the energy cost of operating the data centre [3]. Along with placement efficiency, processing speed and scalability to handle a large number of VM requests makes the placement method tractable. Automated resource allocation tools like OpenStack[2] use bin-packing heuristics for VM initial placement. Efficient initial placement can also minimise VM migration, i.e., reallocation of VMs to another PMs.

2. State of the art VM initial placement and VM reallocation methods

VM initial placement problem is NP-Hard, and no exact algorithm produces an optimal solution in a reasonable time. A state of the art literature review on VM initial placement and VM reallocation approaches is presented in Table 1 and Table 2 respectively; approximate algorithms like FFD, several other greedy heuristics, meta-heuristics and machine learning approaches are proposed to minimise the energy consumption and resource wastage.

Microsoft Virtual machine manager internally uses the geometric bin-packing heuristics proposed by Panigrahy et al. [3] called Weighted Dot Product (WDP), and Norm-Based Greedy (NBG). These geometric heuristics consumes a large amount of time while handling a huge number of requests, but demonstrated a good placement efficiency compare to the other scalable heuristics used in their study called FFDSum, FFDProd and FFDExpSum. A modified best fit decreasing (MBFD) algorithm was proposed by Beloglazov et al. in [4], which carries out a continuous search of each VM and PM combination to find the minimum power consuming combinations on the whole to reduce the CDC energy consumption. Zhang & Ansari [5] proposed bin-packing heuristics for heterogeneous VM placement, by considering heterogeneous physical machines in the cloud data centre. Canali et al. proposed an automatic technique to group VMs into classes based on their behaviour [6].

GGA proposed by Xu and Fortes is a multi-objective Grouping Genetic Algorithm for VM Placement (VMP). GGA explore the best permutations of VMs which consumes less energy when placed on PMs [7] with an objective to minimise the power consumption and resource wastage. Later the placement efficiency of GGA was dominated by VMPACS [8], a VM placement based on ant colony system optimisation with the same objectives focused by GGA. COFFGA [9], a genetic algorithm uses First-Fit bin-packing heuristic as a fitness function and can produce optimal solutions.

However, algorithms like VMPACS consume extra energy during the placement optimisation in the form of VM migrations. A similar VM migration based VM placement is introduced in [10] with a modified use of Particle Swarm Optimization (PSO), the modification is to update the particle positions with an energy-aware local fitness first strategy to minimise the energy consumption in CDC. Another meta-heuristic algorithm, Artificial Bee Colony (ABC) is used by Jiang et al. in [11] for VM migration based energy minimisation practices in CDCs. Similarly, biogeographybased optimisation too applied to

[2] https://www.openstack.org/

minimise multiple resources (CPU, RAM, Disk Storage and Network bandwidth) wastage [12].

Table 1. State of the art review of VM placement approaches.

Authors	Algorithm(s)	Focused on minimising
Panigrahy et al.	FFDSum, FFDProd, Weighted Dot Product, Norm-based greedy [3]	Physical Machines (PMs) used
Beloglazov et al.	Modified best fit decreasing (MBFD) [4]	Energy consumption and SLA violation
Zhang & Ansari	FFD-DRR DRR-BinFill [5]	Total estimated energy consumption by PMs
Canali et al.	Class-based placement [6]	PMs used
Xu and Fortes	GGA [7]	Total resource wastage, power consumption and thermal dissipation costs
Huda et al.	COFFGA [9]	PMs used
Gao et al.	VMPACS [8],	Power consumption and resource wastage
Wang et al.	PSO [10]	Energy consumption
Jiang et al.	Artificial Bee Colony [11]	Total estimated energy consumption by PMs
Zheng et al.	Biogeography-based optimisation [12]	Multiple Resource wastage (CPU, RAM, Disk Storage, Network Bandwidth)
Jangiti et al.	FFD-Aggregated Rank [13]	PMs used
Jangiti et al.	ENSEMBLE-HIDE-SPADE [14]	PMs used

Table 2. State of the art review of VM reallocation approaches

Authors	Approach and Description	Focused on minimising
Hermenier et al.	**Bin repacking** - VM consolidation by scheduling VM migrations based on new resource and placement requirements [15]	Minimizing average migration completion time
Chen et al.	**Effective VM Resizing** - PM consolidation by formulating as a stochastic bin- packing problem [16]	Server capacity and allowed server overflow probability
Masson et al.	**Iterated local search** - To maximise the resource usage in PMs [17]	Resource capacity constraints
Gao et al.	**VMPACS** does consolidation to minimise the number of PMs based on Ant colony optimisation [8]	Resource wastage
Mishra & Bellur	**Bal_pack** is a probabilistic model on resource shortfall threshold based on bin-packing model [18]	PMs used and SLA violations
Shojafar et al.	**Joint dynamic Lyapunov based scheduler,** a modified best fit decreasing in-packing model [19]	CPU Usage and Network Bandwidth

3. Preliminaries

A CDC offering reserved and standard VM instances receive similar type of requests in a large number for a reservation period. The pigeon-hole principle [20] states that at least one standard type of VM instance is repeated (requested more than once by one or more users), when the number of item types (m) is less than number of items (n). A bin-packing outline called bulk-bin-packing is proposed in our recent work [21] and is described here, since it is a base for the current proposed Bulk-Bin-Packing based Migration

Management (BBPMM) of reserved virtual machine requests for green cloud computing.

3.1. Bulk-Bin-Packing (BBP)

The bin-packing heuristic BBP will speed up the packing of huge number of items of few item types. BBP use bulk numbers based on Equation 1.

Definition 1- Bulk number [21]**:** Bulk number represents the number of more bins that can be filled with a similar packing that of current bin.

$$\boldsymbol{BulkNumber} = \boldsymbol{minimum}\left(\frac{\boldsymbol{remaining_i}}{\boldsymbol{packed_i}}\right),\quad \boldsymbol{1 \le i \le m\ \&\ packed_i \ne 0} \tag{1}$$

Definition 2 - Bulk-bin-packing [21]**:** Given $\boldsymbol{n}$ items of $\boldsymbol{m}$ dissimilar types in the form of $(\boldsymbol{item_i}, \boldsymbol{count_i})$ ordered pairs, where $\boldsymbol{i = 1..m}$ and $\sum_{i=1}^{m} \boldsymbol{count_i} = \boldsymbol{n}$. The items are to be packed into minimum bins of capacity $\boldsymbol{C}$. The bulk-bin-packing solution will be in the form of ordered pairs $\{(\boldsymbol{BulkNumber_1}, \boldsymbol{binConfig_1}), (\boldsymbol{BulkNumber_2}, \boldsymbol{binConfig_2}), \ldots\ldots, (\boldsymbol{BulkNumber_k}, \boldsymbol{binConfig_k})\}$, where $\boldsymbol{binConfig_{1\ldots k}}$ are $\boldsymbol{k}$ dissimilar packing configurations and $\boldsymbol{k \le 2 * m}$.

4. Bulk-bin-packing based Migration Management (BBPMM) of reserved VM requests

Based on the revenue and reservation models, VM instances offered by an IaaS CSP can be categorized as (i) Reserved, (ii) On-demand and (iii) Spot Instances. Customers can book reserved VMs in advance if resource quantities are well-known, otherwise on-demand instances at a premium cost. To improve profits, the CSP offers the unutilized resources in the running PMs as spot instances [22] and can be taken back at short notice if required. This paper proposes a split in the management of reserved and on-demand VM requests with different placement and migration controllers since their incoming and outgoing behaviour is different. Assume a batch of reserved VM requests is ready for placement at every reservation time unit T. Let T_i be the start time of i^{th} cycle of incoming reserved VM requests for placement. A request reserved for k time units and start time T_i needs resources allocated till T_{i+k}, where as on-demand requests may leave any time. The PMs are divided into two logical pools. All the PMs, other than those hosting reserved VM instances, can service on-demand requests. The resource gaps in the reserved logical pool PMs can be used by on-demand as well as spot instances up to the next reservation period. The logical view of the considered VM placement and migration management is depicted in Figure 1. The VM placement controller initially maps incoming VM requests to a minimum number of PMs. The VM migration controller reallocates VMs to other PMs with an objective to reduce the number of PMs powered on from time to time. Our recent contribution MCBVP [21] is highly useful for reserved VM initial placement, since requests are pooled for placement as bulk requests. Also, there is a chance of getting bulk on-demand requests at times. MCBVP can be used for on-demand request initial placement too. When the amount of on-demand requests are below the threshold level, the well known best-fit decreasing can be used.

4.1. Migration management of reserved VM requests

The on-demand VM requests need to be frequently consolidated compared to reserved VM requests. Several VM reallocation methods [8,17], [12,23] deal with the consolidation of on-demand VM requests alone. A gap was identified in the existing literature that none of the VM reallocation models is focused on BBPMM. A new migration management model is proposed for reserved VM requests. The incoming reserved requests are first accumulated and placed in PMs at regular reservation intervals. For every reservation period, according to variations in the number of hosting requests, the amount of running PMs needs to be adjusted in the reserved logical pool. The VM migrations are relatively less in this pool compared to the on-demand pool, and there is more possibility of efficient energy management. The VM migration management can be seen in two cases as shown in Figure 2. In **Algorithm 1,** step 2 calls *fillGaps(T)*, to fill the resource gaps. A resource gap is the resources occupied by VMs that are leaving at time T is mapped with same type of requests incoming at time T. Case -1: All the resource gaps are filled completely and new VM requests alone remain for placement. MCBVP is used only to place the new requests. Case 2: If both gaps and requests are remaining, all those VMs, which are currently in the PMs containing resource gaps, are also considered as new requests. Using MCBVP, new placement is generated for those existing requests along with the new VM requests.

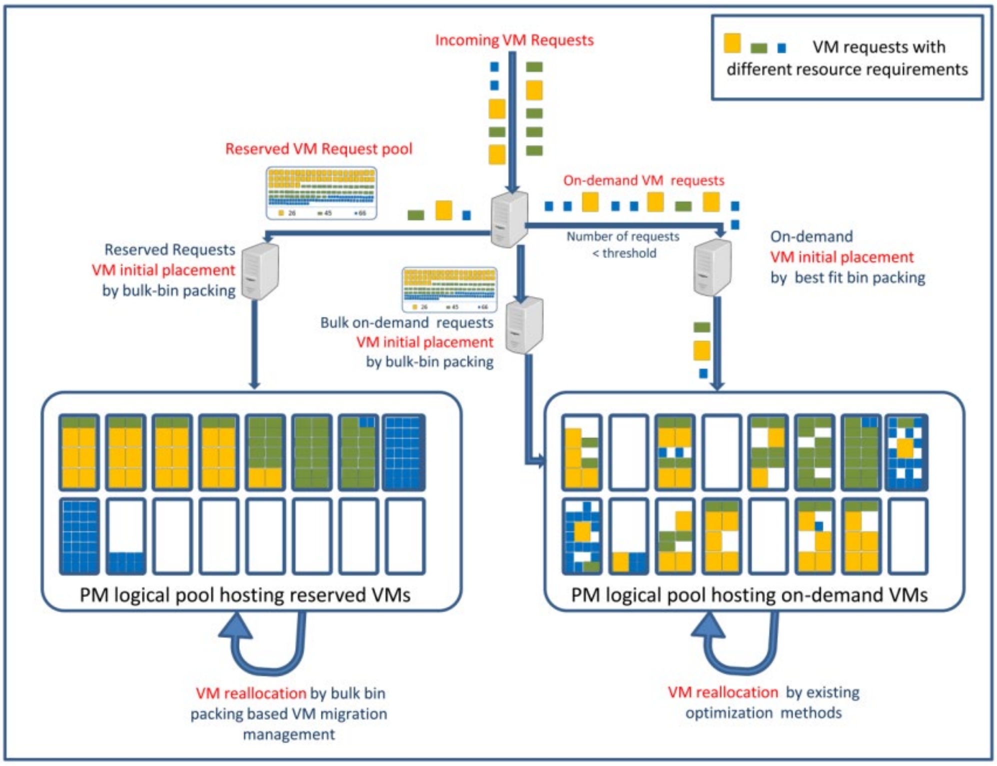

Figure 1. Logical View of a VM initial placement and migration manager

Algorithm 1: Bulk-Bin-Packing based Migration Management (BBPMM)

$\boldsymbol{Repeat\ at\ each\ reservation\ time\ T}$
  $\boldsymbol{for\ each\ PM\ hosting\ VMs\ leaving\ at\ T}$ do
    $\boldsymbol{T_{PMlive}} = \boldsymbol{PM.remainingLife}()$
    $\boldsymbol{for\ each\ VM_{PM}\ leaving\ current\ PM\ at\ T\ do}$
      $\boldsymbol{for\ each\ VM} \in \boldsymbol{VM_{requests}\ do}$
        if $\boldsymbol{VM.type} = \boldsymbol{VM_{PM}.type}$ & $\boldsymbol{T_{res}} \leq \boldsymbol{T_{PMlive}}$
          $\boldsymbol{PM.place}(\boldsymbol{VM, T, T_{res}})$
      $\boldsymbol{end\ for}$
    $\boldsymbol{end\ for}$
  $\boldsymbol{end\ for}$
  $\boldsymbol{if}(\boldsymbol{Gaps_{filled}})$
    $\boldsymbol{call\ MCBVP}(\boldsymbol{VM_{requests}})$
  $\boldsymbol{else}$ //if gaps remaining
    $\boldsymbol{for\ each\ PM} \in \boldsymbol{PM_{Gaps}\ do}$ // add all VMs in PMs with
      $\boldsymbol{for\ each\ VM} \in \boldsymbol{PM\ do}$ //gaps to the set $\boldsymbol{VM_{requests}}$
        $\boldsymbol{VM_{requests}} \leftarrow \boldsymbol{VM_{requests}} \cup \boldsymbol{VM}$
      $\boldsymbol{end\ for}$
    $\boldsymbol{end\ for}$
    $\boldsymbol{call\ MCBVP}(\boldsymbol{VM_{requests}})$
  $\boldsymbol{end\ if}$
$\boldsymbol{end\ repeat}$

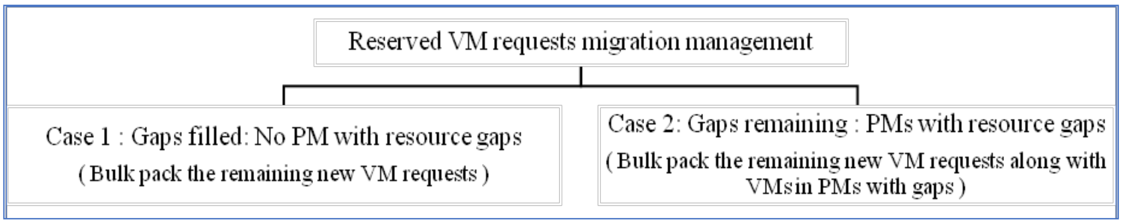

Figure 2. Cases in BBPMM

5. Simulation Experiments and Result Analysis

MCBVP and BBPMM were implemented in Python environment and the simulation experimentation was carried out in a Windows 10 operating system with Intel® Core ™ i7 processor @ 2.5 GHz and 4 GB RAM. Real Dataset – EC2 was used to demonstrate the migration management of reserved VM requests.

5.1. Dataset Description

We consider the Real Dataset – EC2 [12] inspired by Amazon EC2. Seventeen instances are available,of which compute-optimised and memory-optimised are five each, and the remaining seven are general purpose VM instances as shown in Table 3.

5.2. Results & Discussion

The simulation experiment is to test the proposed BBPMM (Algorithm 1). A random amount of all 17 types of VMs were supplied for placement in the initial reservation period T_0. The outcome of BBPMM, PMs used and the number of VMs hosted of each type during the sucessive ten reservation periods were as listed in Table 4. The VM requests supplied for placement at each reservation period is listed in Table 5. The automatic adjustment of PMs used at each reservation period and total VMs hosted by them is depicted in Figure 3.

Table 3. Standard VM instances from Real Dataset – EC2, their resource ratio and sizes

VM instance Specs						PM Specs	
id	model	Instance name	Cores (Size)	RAM (in GB)	RAM/Cores (RpC)	logical processors	RAM (in GB)
1	General Purpose	t2.micro	1	1	1	24	32
2		t2.small	1	2	2		
3		t2.medium	2	4	2		
4		m3.medium	1	3.75	3.75	40	128
5		m3.large	2	7.5	3.75		
6		m3.xlarge	4	15	3.75		
7		m3.2xlarge	8	30	3.75		
8	Compute optimized	c3.large	2	3.75	1.875	80	128
9		c3.xlarge	4	7.5	1.875		
10		c3.2xlarge	8	15	1.875		
11		c3.4xlarge	16	30	1.875		
12		c3.8xlarge	32	60	1.875		
13	Memory optimized	r3.large	2	15.25	7.625	80	512
14		r3.xlarge	4	30.5	7.625		
15		r3.2xlarge	8	61	7.625		
16		r3.4xlarge	16	122	7.625		
17		r3.8xlarge	32	244	7.625		

Table 4. Counts of total VM requests placed and PMs used by them at different reservation periods

Reservation period	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
Total VMs	7640091	6066295	6677950	7752188	12021222	10125176	16209807	25358139	40661396	31538381	31038843
PMs used	2265604	1567596	1849979	2204560	3559888	2999712	4699524	7427368	12438887	7658111	7616390

Table 5. VM request count of each type at different reservation periods

VM instance type	Reservation Period and Number of VMs placed of each type										
	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
t2.micro	366587	510643	466839	440391	538082	524530	830100	1232717	1878460	1988145	1937385
t2.small	184293	266966	178539	183054	259660	312540	562098	1038993	1690012	1785455	1738233
t2.medium	575595	643947	655350	693435	968025	956916	1321434	2355507	4515417	4514801	4531386
m3.medium	269854	247784	292616	309206	230109	310345	542089	988615	1289172	1291742	1277024
m3.large	637260	278355	162049	181717	245059	248522	449799	742122	763942	762608	766957
m3.xlarge	160458	154880	209227	146592	250976	429060	756179	814345	1233213	1230722	658831
m3.2xlarge	399046	502442	704569	805655	1348251	146270	115658	107159	165565	175615	180730
c3.large	283160	176485	146586	150423	251755	276550	537816	1057496	1056618	1055735	1119153
c3.xlarge	897334	387971	406908	413628	574927	542173	802750	977712	1523366	1520108	1578116
c3.2xlarge	161193	274495	378196	599003	900989	287056	307513	554479	971724	971366	971807
c3.4xlarge	617104	437612	305742	478532	719904	716759	1264548	1949242	3581426	3579239	3609748
c3.8xlarge	725419	320196	448663	449741	608732	640808	689845	1114689	1652737	1657518	1661556
r3.large	607297	257255	211705	277242	482660	482881	484125	929050	1285378	1286233	1282864
r3.xlarge	231075	229359	247280	412692	594569	542735	930222	1780393	2085169	2094060	2095092
r3.2xlarge	527994	487634	828300	917665	1690304	1571390	2983784	3254070	5307624	5306512	5311044
r3.4xlarge	710122	782291	950846	1180854	2164599	1970404	3304683	5937238	11021434	1922068	1924797
r3.8xlarge	286300	107983	84535	112359	192620	166236	327163	524311	640138	396453	394119

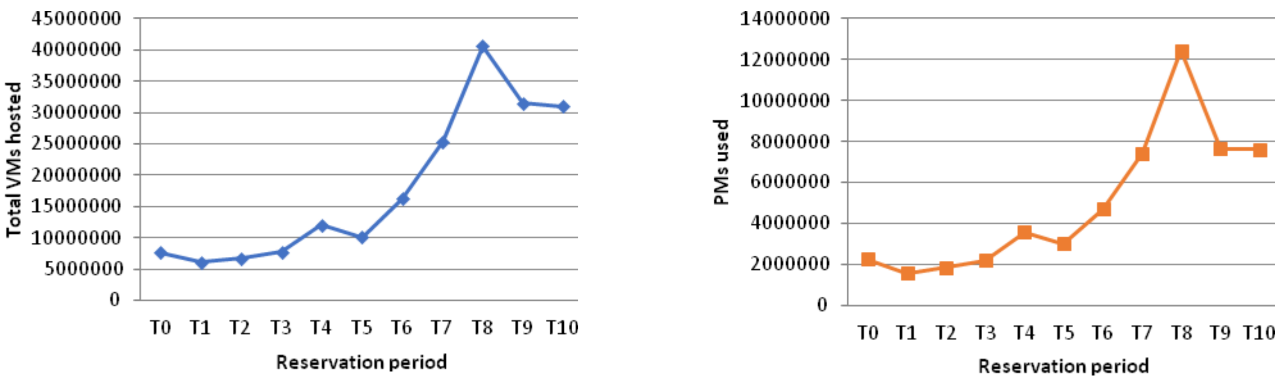

Figure 3. The outcome of BBPMM, PMs used and total VMs hosted at different reservation periods.

6. Conclusion and Future Work

As part of energy efficient management of CDCs, VM consolidation has secured a major concern in reducing CSP operational expenses by minimising the PMs used. In this paper, we presented a heuristic approach BBPMM based on our earlier contributions, MCBVP and BBP as part of energy efficient management of CDCs. At each reservation period, the problem is formulated as Vector Bin-Packing (VBP), a combinatorial optimisation and NP-hard problem.

The Real-time Amazon EC2 dataset was used in the simulation experiments for the demonstration of elastic usage of hosts. The experimental results show that BBPMM adjusted the PMs used in an elastic manner for each reservation period. The elastic reduction of PMs arises upon VMs exit is a green computing effort demonstrated by the proposed BBPMM. The transition period T_8 to T_9 provided the major opportunity of reducing the running PMs by 38% and BBPMM has successfully utilized it.

Acknowledgements.

The authors thank the Department of Science and Technology for their financial support (SR/FST/ETI-349/2013) under Fund for Improvement of S&T Infrastructure in Universities and Higher Educational Institutions.

References

[1] L.A. Barroso, U. Hölzle, The case for energy-proportional computing, Computer (Long. Beach. Calif). 40 (2007) 33–37. doi:10.1109/MC.2007.443.

[2] S. Jangiti, V.S.S. Sriram, R. Logesh, The role of cloud computing infrastructure elasticity in energy efficient management of datacenters, in: 2017 IEEE Int. Conf. Power, Control. Signals Instrum. Eng., IEEE, 2017: pp. 758–763. doi:10.1109/ICPCSI.2017.8391816.

[3] R. Panigrahy, K. Talwar, L. Uyeda, U. Wieder, Heuristics for Vector Bin Packing, Res. Microsoft. (2011) 1–14. doi:10.1109/GreenCom-CPSCom.2010.137.

[4] A. Beloglazov, J. Abawajy, R. Buyya, Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing, 28 (2012) 755–768. doi:10.1016/j.future.2011.04.017.

[5] Y. Zhang, N. Ansari, Heterogeneity aware dominant resource assistant heuristics for virtual machine consolidation, GLOBECOM - IEEE Glob. Telecommun. Conf. (2013) 1297–1302. doi:10.1109/GLOCOM.2013.6831253.

[6] C. Canali, R. Lancellotti, Scalable and automatic virtual machines placement based on behavioral similarities, Computing. 99 (2017) 575–595. doi:10.1007/s00607-016-0498-5.

[7] J. Xu, J.A.B. Fortes, Multi-objective virtual machine placement in virtualized data center environments, in: Proc. - 2010 IEEE/ACM Int. Conf. Green Comput. Commun. GreenCom 2010, 2010 IEEE/ACM Int. Conf. Cyber, Phys. Soc. Comput. CPSCom 2010, 2010: pp. 179–188. doi:10.1109/GreenCom-CPSCom.2010.137.

[8] Y. Gao, H. Guan, Z. Qi, Y. Hou, L. Liu, A multi-objective ant colony system algorithm for virtual machine placement in cloud computing, J. Comput. Syst. Sci. 79 (2013) 1230–1242. doi:10.1016/j.jcss.2013.02.004.

[9] H. Hallawi, J. Mehnen, H. He, Multi-Capacity Combinatorial Ordering GA in Application to Cloud resources allocation and efficient virtual machines consolidation, Futur. Gener. Comput. Syst. 69 (2017) 1–10. doi:10.1016/j.future.2016.10.025.

[10] S. Wang, Z. Liu, Z. Zheng, Q. Sun, F. Yang, Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers, in: Proc. Int. Conf. Parallel Distrib. Syst. - ICPADS, 2013: pp. 102–109. doi:10.1109/ICPADS.2013.26.

[11] J. Jiang, Y. Feng, J. Zhao, K. Li, DataABC: A fast ABC based energy-efficient live VM consolidation policy with data-intensive energy evaluation model, Futur. Gener. Comput. Syst. 74 (2017) 132–141. doi:10.1016/j.future.2016.05.013.

[12] Q. Zheng, R. Li, X. Li, N. Shah, J. Zhang, F. Tian, K.M. Chao, J. Li, Virtual machine consolidated placement based on multi-objective biogeography-based optimization, Futur. Gener. Comput. Syst. 54 (2016) 95–122. doi:10.1016/j.future.2015.02.010.

[13] S. Jangiti, E. Sri Ram, V.S. Shankar Sriram, Aggregated Rank in First-Fit-Decreasing for Green Cloud Computing, in: Springer, Singapore, 2019: pp. 545–555. doi:10.1007/978-981-13-0617-4_53.

[14] S. Jangiti, E. Sri Ram, L. Ravi, V.S.S. Sriram, Scalable hybrid and ensemble heuristics for economic virtual resource allocation in cloud and fog cyber-physical systems, J. Intell. Fuzzy Syst. Preprint (2019) 1–11. doi:10.3233/JIFS-179004.

[15] F. Hermenier, S. Demassey, X. Lorca, M. Chen, H. Zhang, Y.Y. Su, X. Wang, G. Jiang, K. Yoshihira, Bin repacking scheduling in virtualized datacenters, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 6876 LNCS (2011) 594–601. doi:10.1109/INM.2011.5990564.

[16] M. Chen, H. Zhang, Y.Y. Su, X. Wang, G. Jiang, K. Yoshihira, Effective VM sizing in virtualized data centers, Proc. 12th IFIP/IEEE Int. Symp. Integr. Netw. Manag. IM 2011. (2011) 594–601. doi:10.1109/INM.2011.5990564.

[17] R. Masson, T. Vidal, J. Michallet, P.H.V. Penna, V. Petrucci, A. Subramanian, H. Dubedout, An iterated local search heuristic for multi-capacity bin packing and machine reassignment problems, Expert Syst. Appl. 40 (2013) 5266–5275. doi:10.1016/j.eswa.2013.03.037.

[18] M. Mishra, U. Bellur, Whither tightness of packing? The case for stable VM placement, IEEE Trans. Cloud Comput. PP (2015) 481–494. doi:10.1109/TCC.2014.2378756.

[19] M. Shojafar, C. Canali, R. Lancellotti, E. Baccarelli, Minimizing Computing-plus-Communication Energy Consumptions in Virtualized Networked Data Centers, (2016) 1–8.

[20] W.A. Trybulec, Pigeon hole principle, J. Formaliz. Math. 2 (1990) 0. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.73.8514&rep=rep1&type=pdf (accessed February 21, 2017).

[21] S. Jangiti, S. Sriram. V.S., Scalable and direct vector bin-packing heuristic based on residual resource ratios for virtual machine placement in cloud data centers, Comput. Electr. Eng. (2018). doi:10.1016/j.compeleceng.2018.03.029.

[22] P. Sharma, S. Lee, T. Guo, D. Irwin, P. Shenoy, SpotCheck: Designing a Derivative IaaS Cloud on the Spot Market, in: ACM, New York, NY, USA, 2015: pp. 16:1–16:15. doi:10.1145/2741948.2741953.

[23] F. Hermenier, S. Demassey, X. Lorca, Bin repacking scheduling in virtualized datacenters, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 6876 LNCS (2011) 27–41. doi:10.1007/978-3-642-23786-7_5.