On Modelling Virtual Machine Consolidation to Pseudo-Boolean Constraints

Bruno Cesar Ribas[1,3], Rubens Massayuki Suguimoto[2], Razer A.N.R. Montaño[1], Fabiano Silva[1], Luis de Bona[2], and Marcos Castilho[1]

[1] LIAMF - Laboratório de Inteligência Artificial e Métodos Formais
[2] LARSIS - Laboratório de Redes e Sistemas Distribuídos
Federal University of Paraná
[3] Universidade Tecnológica Federal do Paraná - Campus Pato Branco
{ribas,rubensm,razer,fabiano,bona,marcos}@inf.ufpr.br

Abstract. Cloud Computing is a new paradigm of distributed computing that offers virtualized resources and services over the Internet. To offer Infrastructure-as-a-Service (IaaS) many Cloud providers uses a large data center which usage ranges 5% to 10% of capacity in average. In order to improve Cloud data center management and resources usage a Virtual Machine (VM) consolidation technique can be applied to increase workloads and save energy. Using VM consolidation, we introduce an artificial intelligence consolidation based in Pseudo-Boolean (PB) Constraints to find a optimal consolidation. To evaluate our PB consolidation approach we used the DInf-UFPR and Google Cluster scenario and the formulas are solved with two state-of-the-art solvers.

Keywords: pseudo-Boolean, satisfiability, optimization, cloud, consolidation.

1 Introduction

Cloud Computing is a new paradigm of distributed computing that offers virtualized resources and services over the Internet [1,7]. Using Cloud Computing it is possible to offer a pool of easily usable and accessible virtualized resources. These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs [16].

One of the service model offered by Clouds is Infrastructure-as-a-Service (IaaS) in which virtualized resource are provided as virtual machine (VM). With VMs, users obtain a personalized and isolated execution environment to execute applications. A VM also uses virtualized resources such virtual CPU, virtual RAM, virtual network and virtual storage devices.

Many Cloud providers use a large data center in order to offer IaaS. Data centers contains a huge amount of physical resources (server, disks, wired networks). Unfortunately, most of large data center usage ranges from 5% to 10% of capacity on average. In order to maximize the resources utilization by virtualized resources, a IaaS Cloud

provider can apply server consolidation technique [4,11,17] for VM reallocation on physical servers. This consolidation is also denoted as VM Consolidation.

A server consolidation can increase workloads on servers from 50% to 85% where they can operate more energy efficiently [5] and, in some cases, a consolidation can save 75% of energy [3]. Reallocating virtualized resources allow to shutdown physical servers, reducing cooling costs, headcount, hardware management and energy consumption costs.

To maximize Cloud data center usage, an optimal VM consolidation has been topic of research in Cloud Computing. There are works [2,4,11,17] that uses *Linear Programming* formulation or distributed algorithms to guarantee the optimal resource utilization. Different from these approaches we introduce an artificial intelligence approach based on Pseudo-Boolean (PB) [13] formulation to solve the optimization problem. We perform experiments using DInf-UFPR datacenter and Google Cluster to evaluate our approach based on real scenarios.

In Sect. 2 we present related works to consolidation in Clouds. Section 3 describes the Pseudo-Boolean formulation. In Sect. 4 we evaluate the proposed approach using data from real scenario. Finally, in Sect. 5 we present a conclusion and future works.

2   Related Works

Advances in virtualization technology allowed migration of VMs or entire virtual execution environment across physical resources. It also allowed a VM consolidation which has been investigated with different aspects [3,12,15] such performance of VM, energy consumption, costs of resource and costs of migration. Optimal VM consolidation has been explored and solved using *Linear Programming* formulation [2,4] and Distributed Algorithms [11] approaches.

Marzolla *et al.* [11] presents a gossip-based algorithm called *V-Man*. Each physical server (host) run *V-Man* with an *Active* and *Passive* threads. *Active* threads request a new allocation to each neighbor sending to them the number of VMs running. The *Passive* thread receives the number of VMs, calculate and decide if current node will pull or push the VMs to requested node. The algorithm iterate and quickly converge to an optimal consolidation, maximizing the number of idle hosts.

Ferreto *et. al.* [4] presents a *Linear Programming* formulation and add constraints to control VM migration on VM consolidation process. The migration control constraints uses CPU and memory to avoid worst performance when migration occurs.

Bossche *et. al.* [2] propose and analyze a *Binary Integer Programming* (BIP) formulation of cost-optimal computation to schedule VMs in Hybrid Clouds. The formulation uses CPU and memory constraints and the optimization is solved by *Linear Programming*.

Different from above approaches, we introduce an artificial intelligence solution based on *Pseudo-Boolean* formulation to solve the problem of optimal VM consolidation.

3   Pseudo-Boolean Optimization

A Pseudo-Boolean function in a straightforward definition is a function that maps Boolean values to a real number. The term pseudo-Boolean is given to these functions that are not

Boolean but remains very close to Boolean functions [8,10,13]. In a *Pseudo-Boolean* (PB) formula, variables have Boolean domains and constraints, know as PB constraints [13], are linear inequalities with integral coefficients. In *PB Optimization*, a cost function is added to a PB formula.

PB functions are a very rich subject of study since numerous problems can be expressed as the problem of optimizing the value of a PB function. PB constraints offer a more expressive and natural way to express constraints than clauses and yet, this formalism remains close enough to the *Satisfiability* (SAT) [8,10] problem to benefit from the recent advances in SAT solving.

Simultaneously, PB solvers benefit from the huge experience in *Integer Linear Programming* (ILP) and, more specifically, *0-1 programming*. This is particularly true when optimization problems are considered. Inference rules allow to solve problems polynomially when encoded with PB constraints while resolution of the problem encoded with clauses requires an exponential number of steps. PB constraints appear as a compromise between the expressive power of the formalism used to represent a problem and the difficulty to solve the problem in that formalism [13].

In this work we use PB constraints instead of raw Boolean because each Boolean variable has an integer coefficient that maps the structure of the servers and VMs in terms of processing power (CPU) and memory (RAM). With this construction there is no need to transform the formula into a CNF since PB can represent all that is necessary.

We take advantage of *PB optimization* [13] that are implemented on PB solvers, where we create one more PB constraint. This constraint does not have the inequality to express the upper bound of the constraint but is set as an objective constraint to the solver to find the minimal value that this constraint can assume while respecting all other constraints.

A detailed description of modern SAT solver, maximum satisfiability and Pseudo-Boolean optimization can be found, respectively in [8,10,13].

3.1 PB Formulation to Optimal VM Consolidation

The goal of our problem is to deploy K VMs $\{vm_1 \ldots vm_K\}$ inside N hardwares $\{hw_1 \ldots hw_N\}$ while minimizing the total number of active hardwares. Each VM vm_i has an associated needs such as number of VCPU and amount of VRAM needed while each physical hardware hw_j has an amount of available resources, number of CPU and available RAM.

In order to create the PB Constraints each hardware consists of two variables, one that relates hw_i to the amount of RAM hw_i^{ram} and one that relates to the amount of CPU hw_i^{proc}. Per hardware, a VM has 2 variables, one to relate the VM vm_j required amount of VRAM vm_j^{ram} to the hardware hw_i amount of RAM hw_i^{ram}, denoted as $vm_j^{ram \cdot hw_i}$. The another variable relate the required VCPU vm_j^{proc} to the amount of CPU available hw_i^{proc}, denoted as $vm_j^{proc \cdot hw_i}$. The total amount of VM variables is $2 \times N$ variables.

Our main objective is to minimize the amount of active hardware. This constraint is defined in 1. Each hw_i is a Boolean variable that represents one hardware that, when *True*, represents that hw_i is powered on and powered off otherwise.

$$minimize : \sum_{i=1}^{N} hw_i \tag{1}$$

To guarantee that the necessary amount of hardware is active we include two more constraints that implies that the amount of usable RAM and CPU must be equal or greater than the sum of resources needed by VM. These constraints are defined at 2 and 3, respectively.

$$\sum_{i=1}^{N} RAM_{hw_i} \cdot hw_i^{ram} \geq \sum_{j=1}^{K} RAM_{vm_j} \cdot vm_j^{ram} \tag{2}$$

$$\sum_{i=1}^{N} PROC_{hw_i} \cdot hw_i^{proc} \geq \sum_{j=1}^{K} PROC_{vm_j} \cdot vm_j^{proc} \tag{3}$$

To limit the upper bound of hardwares, we add two constraints per host that limit:

Available RAM per Hardware: This constraint dictates that the sum of needed ram of virtual machines must not exceed the total amount of ram available on the hardware, and it is illustrated in constraint 4;

Available CPU per Hardware: This constraint dictates that the sum of VCPU must not exceed available CPU, and it is illustrated in constraint 5.

$$\forall \, hw_i^{ram} \in hw_N^{ram} \left(\sum_{j=1}^{K} RAM_{vm_j} \cdot vm_j^{ram \cdot hw_i} \leq RAM_{hw_i} \right) \tag{4}$$

$$\forall \, hw_i^{proc} \in hw_N^{proc} \left(\sum_{j=1}^{K} PROC_{vm_j} \cdot vm_j^{proc \cdot hw_i} \leq PROC_{hw_i} \right) \tag{5}$$

Finally we add one constraint per VM to guarantees that the VM is running in exactly one hardware. These constraints can be seen on constraint 6.

$$\forall \, vm_i \in vm_K \left(\sum_{j=1}^{N} vm_i^{proc \cdot hw_j} \cdot vm_i^{ram \cdot hw_j} \cdot hw_j^{proc} \cdot hw_j^{ram} = 1 \right) \tag{6}$$

With this model we have $(2 \times N + 2 \times N \times K)$ variables and $(2 + 2 \times N + K)$ constraints with one more constraint to minimize in our PB formula. It is possible to get these amounts because it is a non-linear formula since constraint 6 has a sum of four multiplication.

Note that additional constraints, such as requiring minimal latency between VM, minimal guarantee of bandwidth, migration costs and others will add additional complexity to the problem and are left for future works.

4   Experiments

For the implementation and evaluation of the PB Constraints, we wrote a simple program that reads the amount of physical hardware followed by its amount of RAM and CPU, the amount of VM and its requirements of virtual memory (VRAM) and virtual processing power (VCPU), and solved the formula using open source PB solver/optimizer *Sat4j-PB* [6] and *BSOLO* [9].

We use two workloads to perform our PB consolidation approach. The first is the datacenter of Informatic Departament of Federal University of Paraná (DInf-UFPR), which are used to deploy VMs to offer services and execution environments for experiments of researches and students. The second is the Google Cluster Data project which has traces about machines and tasks running in Google servers. Tasks have resource requirements as well as VMs.

To evaluate both workloads we used the First-Fit and Round-Robin approaches to allocate the VMs on resources to compare with our PB optimal solution. With Round-Robin we expect to find the worst case and with First-Fit a medium case of consolidation.

We also used a *subset of workloads* to see the progress on the use of different amount of VM or tasks. A *subset of workload* is the larger subset of VMs or tasks which sum of VCPU requirements does not exceed σ percent of sum of physical servers CPU capacities. In this experiment we assume σ equals to 25%, 50% and 75%.

4.1   Better Use of DInf-UFPR Datacenter

In DInf-UFPR Datacenter we separated a set of physical server and VMs totalizing 9 servers and 22 VMs. The configuration are as follows on Table 1. The number of CPU and VCPU is given by the amount of processing cores and RAM and VRAM is given by amount of memory in Gigabytes.

To evaluate our approach in this scenario, we took the subset of VMs present in Table 2. The table shows information about subsets with respective sum of VCPU, sum of VRAM and amount of VMs.

As a result, Table 3 show the execution time, in seconds, of PB solvers for current scenario with above subsets workload. The table also shows respective amount of variables and amount of PB constraints generated from formula. Figure 1 presents the number of active servers for each subset. Each subset was executed using Round-Robin, First-Fit and PB consolidation with Sat4j-PB and BSOLO solver.

The results obtained in DInf-UFPR scenario show that PB optimal consolidation has a better result of First-fit, but it is very close to optimal due to little amount of servers. As expected, Round-Robin presents the worst-case of consolidation.

4.2   Google Cluster Data Project

Google Cluster Data [1] is a Google project to intend for the distribution of data about workloads running on Google Cluster. The workloads contains data traces about *12k* machines describing events and resource capacity of each server. The traces also describes around *132k* tasks workloads with respective resource requirements.

---

[1] http://code.google.com/p/googleclusterdata/

Table 1. Hardwares and VM description for DInf-UFPR scenario

(a) Hardware description.

(b) VMs desciptions.

Host	RAM	CPU
hw1	30	4
hw2	18	4
hw3	10	8
hw6	10	8
hw5	30	4
prd3b	125	32
prd3d	125	32
prd3c	125	32
tesla1	62	16
SUM	535	140

VM	VRAM	VCPU	VM	VRAM	CPU
planetmon	12	4	db	2	1
vc3-blanche	8	4	devel	4	2
alt	10	8	salinas	5	2
dalmore	10	8	vc3-colombard	8	2
mumm	10	8	vc3-educacional	2	2
priorat	5	8	vc3-newcastle	4	2
talisker	32	8	vc3-qef1	2	2
bowmore	20	12	vc3-qef2	2	2
alt-marcadle	80	16	vc3-qef3	2	2
alt-murphy	93	24	vc3-qef4	2	2
caporal	18	4	alt-guinness	120	32
			SUM	451	155

Table 2. Workload subsets with σ equals to 25%, 50% and 75% and respective sums of VRAM, VCPU and amount of VMs for DInf-UFPR scenario

Workload Percent	$\sum$ VRAM	$\sum$ VCPU	Amount of VMs
25%	51	23	11
50%	81	39	14
75%	138	71	18

Table 3. Variables and constraints generated and execution time for DInf-UFPR scenario using BSOLO and Sat4j-PB solvers

Formula	Variables	Constraints	BSOLO	Sat4j-PB
hw9-vm25p	108	25	**0.004**	**0.101**
hw9-vm50p	198	30	**0.004**	**0.109**
hw9-vm75p	288	35	**0.004**	**0.118**

Due to the long period to perform PB consolidation using all *12k* machines and *132k* workloads we selected five subset of machines. The size of each subset are **32**, **64**, **128**, **256**, **512** machines. For each size of subset machines, we used the above *subset of workload* to perform experiments. Table 4 shows the amount of resources used to evaluate PB consolidation and others allocation approaches. Values of CPU and RAM are normalized in a scale relative to the largest capacity of the resource on any machine in the period of trace. The value of the largest capacity is *1.0*.

As a result, Table 5 shows time results for the set of formulas explained above. For each instance was given a time limit of 7200 seconds. When the solver run out of time limit and did not found any solution it is show a Time Limit Exceeded (TLE). If the solver caught a Segmentation Fault signal a Runtime Error (RTE) is thrown as a result.
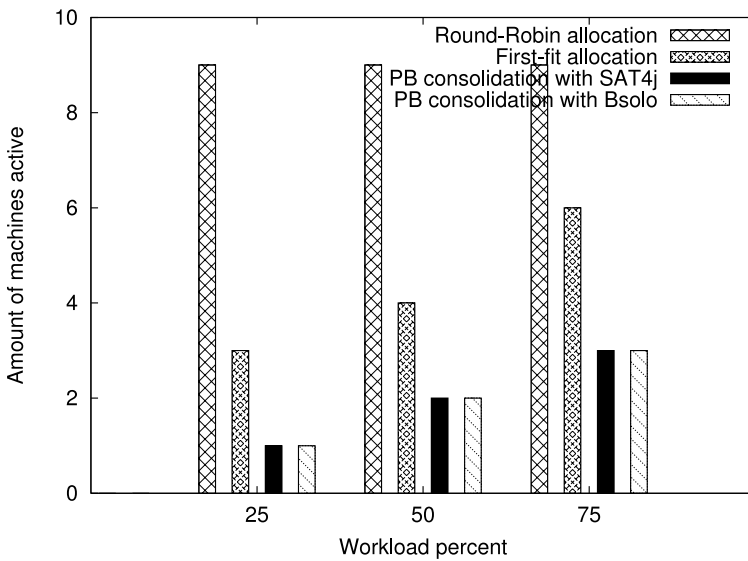
Fig. 1. Number of active hardware for each approach for DInf-UFPR scenario

Table 4. Workload subsets for each subset of machines. The workload has a σ equals to 25%, 50% and 75% and respective sum of VRAM, VCPU and amount of tasks for Google Cluster scenario.

#Machines	RAM	CPU	Workload %	$\sum$ VRAM	$\sum$ VCPU	#Tasks
32	14.9813	17.0000	25%	3.7375	4.3475	98
32	14.9813	17.0000	50%	5.7048	8.5640	173
32	14.9813	17.0000	75%	9.5204	12.7674	278
64	32.2117	34.5000	25%	5.7281	8.6389	174
64	32.2117	34.5000	50%	13.8382	17.2724	371
64	32.2117	34.5000	75%	19.3733	25.8826	559
128	61.8284	68.0000	25%	13.5025	17.0473	368
128	61.8284	68.0000	50%	26.3261	34.3367	713
128	61.8284	68.0000	75%	39.0425	51.0215	1048
256	121.5035	134.5000	25%	26.2943	33.9555	712
256	121.5035	134.5000	50%	49.0585	67.2507	1407
256	121.5035	134.5000	75%	75.6842	10.08777	2119
512	246.7420	275.2500	25%	50.9854	68.8945	1432
512	246.7420	275.2500	50%	100.1324	137.8664	2771
512	246.7420	275.2500	75%	206.4426	148.0852	4035

Figures 2a, 2b, and 2c respectively shows the result of amount actives machines for 32, 64, 128 and 256 subset of machines. For each subset, we perform the Round-Robin, First-Fit and PB consolidation approaches using Sat4j-PB and BSOLO solvers.

Unfortunately none of the tested solvers were able to find a satisfiable assignment for the larger formulas such subsets of 512 machines and 256 machines and only two

Table 5. Execution time per instance for BSOLO and Sat4j-PB solver. Time Limit was set to 7200s and TLE represents when Time Limit was Exceeded and RTE is for RunTime Error.

Formula	Variables	Constraints	BSOLO	Sat4j-PB
hw32-vm25p	6336	164	7242.75	**305.277**
hw32-vm50p	11136	239	7198.01	7204.971
hw32-vm75p	17856	344	7237.44	**6417.293**
hw64-vm25p	22400	304	7198.02	7227.192
hw64-vm50p	47616	501	7198.02	7243.419
hw64-vm75p	71680	689	7198.19	7243.385
hw128-vm25p	94464	626	TLE	7244.51
hw128-vm50p	182784	971	TLE	7244.46
hw128-vm75p	268544	1306	TLE	7243.678
hw256-vm25p	365056	1226	TLE	TLE
hw256-vm50p	720896	1921	RTE	TLE
hw256-vm75p	1085440	2633	RTE	TLE
hw512-vm25p	1467392	2458	RTE	TLE
hw512-vm50p	2838528	3797	RTE	TLE
hw512-vm75p	4132864	5061	RTE	TLE
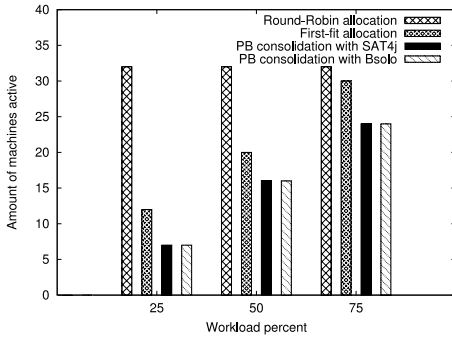
(a) Result for 32 machines.
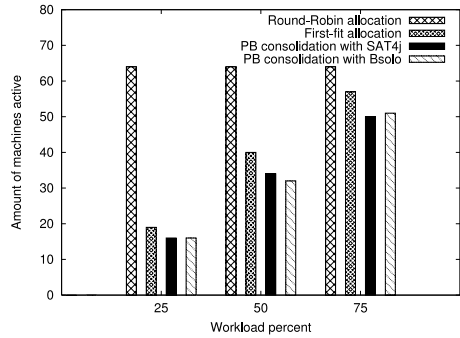
(b) Result for 64 machines.
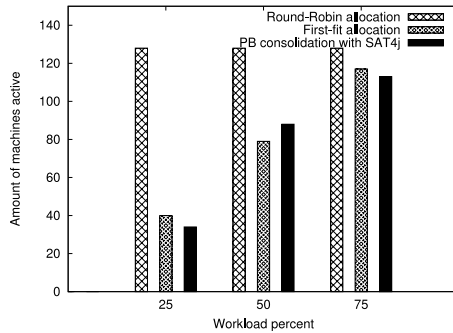
(c) Result for 128 machines.

Fig. 2. Number of active machines using Round-Robin, First-Fit and PB consolidation with Sat4J-PB and BSOLO solvers for Google Cluster scenario

instances reached optimum objective assignment. A non optimum solution can be easily identified in test case of 128 machines with 50% load where in Fig. 2c the First-Fit algorithm were able to optimize better than the PB Solver. Table 5 shows that the biggest formulas tested solver were not even able to find one satisfiable assignment to the formula, as can be identified as RTE and TLE. The RTE has many possibilities of errors caused in the solver execution, and it discussion is out of the scope of this work. The TLE means that it took too much time to find any satisfiable assignment with 7200 seconds time limit.

With the present result, we can confirm the VM consolidation by PB formulation approach is a valid formulation. When the Cloud has only a few resources, both physical and virtual, state-of-the-art solvers can prove optimal consolidation very fast. Within larger instances, PB solvers could not find the optimal, and in most of the cases they do not found any consolidation.

5   Conclusions

This paper presented a VM consolidation model using a artificial intelligence based on Pseudo-Boolean (PB) Constraints. A PB Constraints can be used to optimize costs, i.e minimizing the amount of active hardware. With a PB approach it is easily add extra restrictions to VM consolidation that would not be easily done with a First-fit or Round-Robin algorithms.

Unfortunately, follow experimental results, PB solvers were not able to solve the formulas of a huge test scenario such as Google Cluster. Also the benefit of running time was not as good as others approaches such First-fit algorithm.

Despite the fact tested solvers were not powerful enough to complete all formulas in a practical time we can use these formulas as a good benchmark to improve PB solvers.

We are interested in going on investigating some important research direction. First, we want to extend our solution and implement it inside a Cloud Management System (i.e. OpenNebula [14]) as an optimizer module. After we are interested to add some important restrictions such as network dependency of VMs and create classes of VMs to make better use of network interfaces of hosts.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: Above the Clouds: A Berkeley View of Cloud Computing. Tech. rep., EECS Department, University of California, Berkeley (2009)
2. Bossche, R., Vanmechelen, K., Broeckhove, J.: Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads. In: 3rd IEEE International Conference on Cloud Computing (CLOUD 2010), pp. 228–235. IEEE Computer Society (2010)
3. Corradi, A., Fanelli, M., Foschini, L.: Increasing Cloud Power Efficiency through Consolidation Techniques. In: 2011 IEEE Symposium on Computers and Communications (ISCC 2011), pp. 129–134. IEEE Computer Society (2011)
4. Ferreto, T.C., Netto, M.A.S., Calheiros, R.N., De Rose, C.A.F.: Server Consolidation with Migration Control for Virtualized Data Centers. Journal of Future Generation Computer Systems 27(8), 1027–1034 (2011)

5. Harmon, R., Auseklis, N.: Sustainable IT Services – Assessing the Impact of Green Comput-
   ing Practices. In: 2009 Portland International Conference on Management of Engineering &
   Technology (PICMET 2009), pp. 1707–1717. IEEE Computer Society (2009)
6. Le Berre, D.: SAT4j – A Reasoning Engine in Java Based on the SATisfiability Problem,
   `http://www.sat4j.org`
7. Leavitt, N.: Is Cloud Computing Really Ready for Prime Time? Journal of Computer 42(1),
   15–20 (2009)
8. Li, C.M., Manyà, F.: MaxSAT, Hard and Soft Constraints. In: Biere, A., Heule, M.J.H., van
   Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability, vol. 185, ch. 19, pp. 613–631. IOS
   Press (2009)
9. Manquinho, V.: BSOLO – A Solver for Pseudo-Boolean Constraints,
   `http://sat.inesc-id.pt/~vmm/research/`
10. Marques-Silva, J.P., Lynce, I., Malik, S.: Conflict-Driven Clause Learning SAT Solvers.
    In: Biere, A., Heule, M.J.H., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability,
    vol. 185, ch. 4, pp. 131–153. IOS Press (2009)
11. Marzolla, M., Babaoglu, O., Panzieri, F.: Server Consolidation in Clouds through Gossiping.
    In: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks
    (WoWMoM 2011), pp. 1–6. IEEE Computer Society (June 2011)
12. Mehta, S., Neogi, A.: ReCon: A Tool to Recommend Dynamic Server Consolidation in
    Multi-cluster Data Centers. In: 2008 IEEE Network Operations and Management Sympo-
    sium (NOMS 2008), pp. 363–370. IEEE Computer Society (2008)
13. Roussel, O., Manquinho, V.: Pseudo-Boolean and Cardinality Constraints. In: Biere, A.,
    Heule, M.J.H., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability, vol. 185, ch.
    22, pp. 695–733. IOS Press (2009)
14. Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: Virtual Infrastructure Management
    in Private and Hybrid Clouds. IEEE Internet Computing 13(5), 14–22 (2009)
15. Umeno, H., Parayno, C., Teramoto, K., Kawano, M., Inamasu, H., Enoki, S., Kiyama, M.,
    Aoyama, T., Fukunaga, T.: Performance Evaluation on Server Consolidation using Virtual
    Machines. In: 2006 SICE-ICASE International Joint Conference (SICE-ICCAS 2006), pp.
    2730–2734. IEEE Computer Society (2006)
16. Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M.: A Break in the Clouds –
    Towards a Cloud Definition. Journal of ACM SIGCOMM Computer Communication Re-
    view 39(1), 50–55 (2008)
17. Vogels, W.: Beyond Server Consolidation. Journal of ACM Queue 6(1), 20 (2008)