Constraint Programming-Based Virtual Machines Placement Algorithm in Datacenter

Yonghong Yu, Yang Gao

Constraint Programming-Based Virtual Machines Placement Algorithm in Datacenter

Yonghong YU[1,2] and Yang GAO[1]

[1] State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210093, China
[2] College of Tongda, Nanjing University of Posts and Telecommunications, Nanjing 210003, China
{yuyh.nju@gmail.com,gaoy@nju.edu.cn}

Abstract. As underlying infrastructure of cloud computing platform, datacenter is seriously underutilized, however, its operating costs is high. In this paper, we implement virtual machines placement algorithm in CloudSim using constraint programming approach. We first formulate the problem of virtual machines placement in virtualized datacenters as a variant of multi-dimensions bin packing problem, and then exploit constraint solver to solve this problem with the objective of minimizing number of physical machines that host virtual machines. Finally, we compare different virtual placement algorithms for evaluating constraint programming-based virtual machine placement algorithm including the built-in virtual machine placement algorithm in CloudSim and FFD algorithm. The experimental results show that constraint programming-based virtual machines placement algorithm can efficiently reduce the number of physical machines to achieve the goal of reducing datacenter operating costs and improving resource utilization.

Keywords: Datacenter; Virtual Machine Placement; Constraint Programming; CloudSim

1 Introduction

As an emerging computing paradigm, Cloud Computing can provide users with on demand IT services and elastic computing platforms. Many institutes and companies recently focus on cloud computing technology. Some notable companies have designed and set up their cloud computing platforms respectively, such as Amazon EC2, IBM Blue Cloud, Microsoft Windows AZure, and Salesforce Sales Force. These cloud computing platforms offer customers storage, computing power, deploying environments and IT services at different layers. The development of cloud computing abilities is closely related to supports of underlying datacenters, which provide a powerful parallel computing power and massive data storage for cloud computing.

Operating costs of datacenters is rapidly increasing with scaling of datacenters. However, datacenters resources are seriously underutilized. Energy consumptions of IT devices in servers clusters is the largest contributor to operating

costs. It is estimated that in 2006 the total electricity consumptions of datacenters in US cost of about $4.5 billion. Furthermore, US energy consumption by datacenters could nearly doubled again in 2011 [1]. However, statistics in sports, e-commerce, and financial domains show the average server utilization varies between 11% and 50%[2]. The leading reason of underutilization is that administrator allocates hardware resources for hosting applications according to peak requirements of applications, although peak workloads of applications may not appear frequently. Furthermore, workloads of applications are dynamic, as well as there are stringent requirements for applications performance, such as applications throughput, response time, and cloud computing platform must meet applications SLA(Service Level Agreement). It is a big challenge for cloud datacenters to reduce operating costs and improve resource utilization while meet applications SLAs.

Server virtualization technology is an effective approach to improve the efficiency of resources and save energy while provides performance guarantees for applications. Using server virtualization technology, application runtime environment is encapsulated in a VM(Virtual Machine), and one or more virtual machines host on a PM(Physical Machine). Virtualization technology can provide performance isolation between VMs resided on the same PM, and avoid performance degradation caused by greedy or malicious applications . Moreover, it is helpful to improve resource utilization and save costs by multiplexing of datacenters resources across applications. Furthermore, virtualization technology enables VM live migration to handle dynamic workloads and meet application performance requirements.

However, adjusting the configuration of virtual machine and performing live migration manually obviously does not meet the needs of datacenters management. We need automate and intelligent approaches to address virtual machine management issue. Management of virtual machine consist of two phases: initial plan and runtime management. In initial plan stage, given a set of virtual machines and resource requirements of each virtual machine, as well as a set of physical servers, we need map each virtual machine onto physical machines. The goal of initial plan stage is to minimize the number of physical servers that host all virtual machines. In runtime management stage, we need to adjust configuration of virtual machines or perform live migration of virtual machine according to dynamic workloads. The purpose of runtime management is to minimize the number of virtual machines migration considered migration costs, while meet application SLAs. In this paper, we focus on the problem of virtual machine placement appeared in initialization plan stage of datacenters management with the goal of minimizing the number of physical machines. We first formulate the problem of virtual machine placement as a variant of multi-dimensions bin packing problem, and then exploit constraint solver to solve this problem with the objective of minimizing number of physical machines. Finally, we implement virtual machines placement algorithm using constraint programming technique in CloudSim[3]. Experimental comparison with other heuristic methods verifies the effectiveness of constraint programming-based approach.

The remainder of this paper is organized as follows. Section 2 describes related work. Then, Section 3 formulates virtual machine placement problem and describes in detail constraint programming-based virtual machines placement algorithm. Evaluation results are presented in Section 4. Finally,Section 5 concludes this paper and presents future work.

2 Related Work

Recently, several prototypes and methods have been proposed to address problems of virtual machine management by making a trade-off between applications performance, energy consumption and migration costs.

Verma et al.[4] have proposed pMapper, which is a power-aware application placement controller in heterogeneous server clusters. pMapper exploited heuristic to address virtual machine placement problem. Moreover, working set sizes of applications are considered while placing applications on physical machines. Wood et al.[5] presented Sandpiper, a system that automates the tasks of monitoring datacenters and detecting overloaded virtual machines, determining a new mapping relations between virtual machines and physical machines. Sandpiper implements a black-box approach which is fully OS- and application-agnostic as well as a gray-box approach that can leverage information collected from OS and application. To coordinate the actions taken by platform management and virtualization management in datacenters as well as support existing multivendor solutions, Kumar et al.[6] explored a practical coordination solution that loosely-couple platform and virtualization management in datacenters. In particularly, their solution designed a stabilizer component that prevent unnecessary virtual machine migrations by making a migration decision based on probability with which such a decision remains valid over certain duration in the future. In [7], Dhiman et al. present vGreen, a multitiered software system for energy-efficient virtual machine management in virtualized clusters environment. The most important feature of vGreen is that it implements policies for scheduling and power management of virtual machine using novel hierarchical metrics which capture energy consumption and performance characteristics of both virtual machines and physical machines. To handle local optimization problem of virtual machine placement in datacenters, Hermenier et al. investigated global optimization technique and proposed Entropy resource manager for homogeneous clusters, which utilizes constraint programming[8] to perform dynamic virtual machine consolidation with the aim of minimizing migration overhead. Our work draws upon the methods used in Entropy. On the contrary, we focus on virtual machine placement problem occurred in initial plan stage, and implement virtual machine placement algorithm based on constraint programming in CloudSim.

The above studies focus on configuration of virtual machine during runtime management of datacenters. Besides the above literatures, there're still some other research work similar to our work that is only concerned about initial plan task of datacenters. For example, Campegiani et al.[9] presented a genetic algorithm to search the optimal allocation strategy of virtual machines ,and designed

penalty function to deal with unfeasible solutions. Bellur et al.[10] proposed linear programming and quadratic programming techniques to solve the problem. Jing XU et al.[11] presented a two level control system to find the mapping of workloads to virtual machines and virtual machines to physical resources using an improved genetic algorithm with fuzzy multi-objective evaluation.

3 Constraint Programming-Based Virtual Machine Placement Algorithm

The problem of virtual machine placement can be formulated as a variant of multidimensional bin packing problem, which is combinatorial NP-hard problem. For this problem, several heuristics have been developed, such as FFD(First Fit Decreasing), BFF(Best First Fit), which can quickly provide sub-optimal solution. In this section, we first formulate the problem of virtual machine problem, and then describe in detail virtual machine algorithm based on constraint programming.

3.1 Formalization of Virtual Machine Problem

In virtual machine placement problem, virtual machines are viewed as boxes, where various resource requests of each virtual machine are considered as dimensions of box with non-negative values. Physical servers are considered as bins, where CPU, memory and bandwidth capacities are regarded as properties of box. The goal of virtual machine placement problem is to determine the minimum number of physical machines required by the set of virtual machines.

The problem of virtual machine placement in datacenters is defined as : given a set of virtual machines $VM = \{vm_1, vm_2, ..., vm_n\}$ and a set of physical machines $PM = \{pm_1, pm_2, ..., pm_m\}$, where each vm_i is a triplet $vm_i = (cpu_i, ram_i, bw_i), 1 \leq i \leq n$ denoted cpu,memory and bandwidth requirements of virtual machine respectively, each pm_j is also a triplet $pm_j = (cpu_j, ram_j, bw_j), 1 \leq j \leq m$ denoted resource capacity of physical machine. In addition, $x_{ij}, 1 \leq i \leq m, 1 \leq j \leq n$ and $y_i, 1 \leq i \leq m$ are decision variables, $x_{ij} = 1$ if and only if vm_j is mapped onto pm_i, $y_i = 1$ if pm_i is used to host virtual machine. The objective is to minimize $\sum_{i=1}^{m} y_i$ while finding all values of x_{ij} .

There are several implicit constraints in the above definition: (1) each virtual machine can be hosted on only one physical machine; (2) for each type of resource, the amounts of resource requests of virtual machines sharing the same physical machine are smaller or equal to capacity of physical machine hosting them; (3)the number of physical machines that host virtual machines are not more than m, $\sum_{j=i}^{m} y_i \leq m$.

3.2 Constraint Satisfaction Model of Virtual Machines Placement Problem

In this section, we describe how to tackle virtual machine placement problem in datacenters using constraint programming approach.

Constraint programming has been widely used in a variety of domains such as production planning, scheduling, timetabling and product configuration. The basic idea of constraint programming is that user formulates a real-world problem as a CSP(constraint satisfaction problem) and then a general purpose constraint solver calculates solution for it. Formally, a CSP is defined by a triplet $(Variables, Domains, Constraints)$, where $Constraints$ are the set of constraints being responsible for pruning the search space.

Combining the definition of virtual machine placement problem and the basic idea of constraint programming, the problem of virtual machine placement can by modeled by CSP in the following way:

Variables

- $X = \{x_{ij} | i \in [1, m], j \in [1, n]\}$ denote placement solution of virtual machine in datacenters.
- $Y = \{y_i | i \in [1, m]\}$, if the physical machine pm_i is used, then $y_i = 1$.
- $Load\_CPU = \{load\_CPU_i | i \in [1, m]\}$, $load\_CPU_i$ denotes the total of cpu requests of virtual machines hosted on pm_i.
- $Load\_RAM = \{load\_RAM_i | i \in [1, m]\}$, $load\_RAM_i$ denotes the total memory requests of virtual machines hosted on pm_i.
- $Load\_BW = \{load\_BW_i | i \in [1, m]\}$, $load\_BW_i$ denotes the total bandwidth requests of virtual machines hosted on pm_i.
- $solutionNum$ denotes the number of physical machines used in the solution for virtual machine placement problem.

Domains

- $\forall x_{ij} \in X, x_{ij} \in [0, 1]$
- $\forall y_j \in Y, y_j \in [0, 1]$
- $\forall load\_CPU_i \in Load\_CPU, load\_CPU_i \in [0, pm_i.cpu_i]$
- $\forall load\_RAM_i \in Load\_RAM, load\_RAM_i \in [0, pm_i.ram_i]$
- $\forall load\_BW_i \in Load\_BW, load\_BW_i \in [0, pm_i.bw_i]$
- $solutionNum \in [0, m]$

Constraints

- C1: the total of resource requests of virtual machines sharing the same physical machine are smaller or equal to resource capacity of physical machine hosting them.
  1. $\forall i, \sum_{j=1}^{n} x_{ij} \times vm_j.cpu_j \leq pm_i.cpu_i, i \in [1, m]$
  2. $\forall i, \sum_{j=1}^{n} x_{ij} \times vm_j.ram_j \leq pm_i.ram_i, i \in [1, m]$
  3. $\forall i, \sum_{j=1}^{n} x_{ij} \times vm_j.bw_j \leq pm_i.bw_i, i \in [1, m]$
- C2: assign $y_j = 1$ if there are some virtual machines packed in physical machine pm_j. In this paper ,we only consider cpu load of physical machine. $\forall y_j \in Y, y_j = 1 \Leftrightarrow load\_CPU_j > 0$
- C3: each virtual machine can be packed in only one physical machine. $\forall j, \sum_{i=1}^{m} x_{ij} = 1, j \in [1, n]$
- C4: the number of physical machines that host virtual machines are not more than m.
  $solutionNum = \sum_{i=1}^{m} y_i \leq m$

3.3 Implementation of Constraint Programming-Based Virtual Machine Algorithm

After formulating the virtual machine placement problem as CSP, we now choose a constraint solver to solve the problem.

Constraint programming is often realized in imperative programming by software library, such as Geocode, JaCoP and Choco. We choose Choco as our constraint solver that is compatible with CloudSim. The algorithm for constraint programming-based virtual machine placement as follows:

Algorithm 1 CP-Based Virtual Machine Placement Algorithm.

Input:

$List < PM > pmList$: the set of physical machines.

$List < VM > vmList$: the set of virtual machines .

Output:

$X[m][n]$: the allocation strategy for virtual machines.

1: Defining variables $X[m][n]$,$Y[m]$,$Load\_CPU[m]$,$Load\_RAM[m]$,$Load\_BW[m]$ and $XT[n][m]$,which is a transpose of X.
2: Creating model for CSP : $CPModelmodel = newCPModel()$
3: Initialing variables defined in (1).
4: // Adding constraint C1 in CPModel
5: **for** $i = 0; i < m$; $i++$ **do**
6: $model.addConstraint(eq(load\_cpu[i], scalar(vmList.cpu, X[i])))$;
7: $model.addConstraint(eq(load\_ram[i], scalar(vmList.ram, X[i])))$;
8: $model.addConstraint(eq(load\_bw[i], scalar(vmList.bw, X[i])))$;
9: **end for**
10: // Adding constraint C2 in CPModel
11: **for** $i = 0; i < m$; $i++$ **do**
12: $model.addConstraint(ifOnlyIf(eq(Y[i], 1), gt(load\_cpu[i], 0)))$;
13: **end for**
14: // Adding constraint C3 in CPModel
15: **for** $i = 0; i < n$; $i++$ **do**
16: $model.addConstraint(eq(1, sum(XT[i])))$;
17: **end for**
18: // Adding constraint C4 in CPModel
19: $model.addConstraint(eq(usedBin, sum(Y)))$;
20: $model.addConstraint(leq(usedBin, m))$;
21: //Creating constraint solver and reading CPModel
22: $Solvers = newCPSolver()$;
23: $s.read(model)$;
24: $s.minimize()$ and return $X[m][n]$;

In addition, we can set timeout for this algorithm to reduce the search time of constraint solver, and constraint solver will return local or global optimal solution until timeout expired.

4 Evaluations

To evaluate constraint programming-based virtual machine placement algorithm, we implement this algorithm in CloudSim and conduct two experiments to compare with FFD algorithm.

CloudSim is developed for simulation of cloud computing platform and support system modeling of datacenters, resource management and task scheduling. In CloudSim, the task of virtual machine placement is mainly completed by VmAllocationPolicySimple class which selects a physical machine with the most number of available process units to create virtual machine iteratively. The virtual machine placement policy will use up all physical machine as long as the number of virtual machines is greater than the number of physical machines in datacenter, and this motivated us to design a virtual machine placement policy to reduce the number of physical machines that can host all virtual machines.

We define four classes of virtual machines representing different types of virtual machines respectively. The characteristics of each type of virtual machine are described in Table 1.

Table 1. virtual machine types

VM category	*CPU(MIPS)*	*RAM(MB)*	*BW(Mbit/s)*
VM1	2500	870	100000
VM2	2000	1740	100000
VM3	1000	1740	100000
VM4	500	613	100000

Among four classes of virtual machine, VM1 represents Computing-intensive applications, VM2 and VM3 represents IO-intensive middle scale applications, VM4 represents small scale applications. There are two classes of physical server in datacenter. Their hardware parameters are described in Table 2.

Table 2. Hardware parameters of physical servers

PM category	*CPU(MIPS)*	*RAM(MB)*	*BW(Mbit/s)*
PM1	2*1860	4096	1000000
PM2	2*2660	4096	1000000

Throughout experiments we use a machine with a Intel(R) Core(TM)2 Duo CPU E7500@2.93 and 2GB memory installed CloudSim 3.0, Choco2.1.3 and JDK 1.7. The simulated datacenter consists of 50 physical machines, and each type of physical machine occupies half of the total. In order to verify constraint programming-based virtual machine placement algorithm that can efficiently reduce the number of physical machines required, we compare our approach

with the built-in virtual machine placement algorithm of CloudSim and FFD at different ratios of the number of virtual machines to physical machines, in which we set timeout = 1 second for constraint programming-based virtual machine placement algorithm. The experimental results are plot in Figure 1.

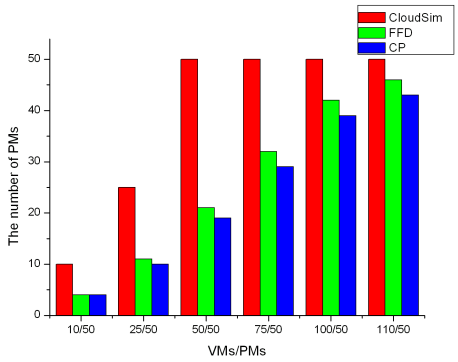

Fig. 1. The number of Physical machines used

The results show that FFD algorithm and constraint programming algorithms is obviously better than CloudSim built-in algorithm in term of the number of physical machines used to host all request virtual machine in the same situations. When the ratio of number of virtual machines to number of physical machines equal to 1, CloudSim built-in algorithm occupies all physical machines, and other two algorithms only need about 20 physical machines. Comparing FFD algorithm with constraint programming-based virtual machine placement algorithm, the differences between FFD algorithm and constraint programming approach is small when the ratios are relatively lower, whereas with the increase of ratios of number of virtual machine to the number of physical machines, constraint programming approach can reduce more number of physical machines than FFD algorithm. In addition, when tmeout =1,2,3,4,5 minutes for constraint programming algorithm, we find the same result as Figure 1. It indicates that constraint programming algorithm can compute approximately global optimization solution within timeout.

To analyze the overhead of constraint programming-based virtual machine placement algorithm, we run each algorithm 10 times and average running time of each algorithms at different ratios, and setting timeout=1 minute for constraint programming approach. The experimental results are shown in Table 3.

From Table 3, we can find that the performance of FFD algorithm is optimal and valid that FFD algorithm can provide a fast and but often non-optimal solution. When the ratio of number virtual machines to number physical machines

Table 3. Average running time of three algorithms

VM/PM	*CloudSim*	*FFD*	*CP*
10/50	5.48ms	1.58ms	47ms
20/50	8.72ms	1.68ms	timeout
50/50	11.56ms	2.53ms	timeout
75/50	13.25ms	3.08ms	timeout
100/50	15.38ms	3.87ms	timeout

is 10/50, constraint programming-based virtual machine placement algorithm is able to return solution within timeout. However, when the number of virtual machine increase, constraint solver have not search whole space until timeout expired and only return approximately global optimal solution.

Our experiments conclude that constraint programming-based virtual machine placement algorithm can find better solution than FFD algorithm and CloudSim built-in placement algorithm with the goal of minimizing the number of physical machines hosting all requested virtual machine within timeout. Although constraint programming approach suffers from relatively long running times, we argue that time performance of virtual machine placement algorithm based on constraint programming is also acceptable because initial configuration of virtual machines is a task of planning and normally takes a long time. In practice, it is needed to trade-off between minimizing the number of physical machines and performance.

5 Conclusion and Future Work

In this paper, we focus on the problem of virtual machine placement appeared in initial plan stage of datacenters management and implement virtual machine placement algorithm exploiting constraint programming approach in CloudSim. The objective is to cut datacenters operating costs and improve resource utilization by reducing the number of physical machines and using virtualization technology. In order to achieve this objective, we formulate the virtual machine placement problem as CSP, and then choose Choco to solve this problem. We experimentally evaluate our approach and compare it with the built-in virtual machine placement algorithm in CloudSim and FFD algorithm. Experimental results show that constraint programming-based virtual machine placement algorithm can efficiently reduce the number of physical servers compared with FFD algorithm and CloudSim built-in placement algorithm, although suffers from relatively long search times.

A salient feature of cloud computing is to provide on-demand resource allocation strategy for applications in order to handle dynamic workloads. As part of future work, we are planning to study how to apply reinforcement learning and virtual machine live migration technologies to intelligently deal with dynamic peak workloads.

Acknowledgments. We would like to acknowledge the support for this work from the National Science Foundation of China (Grant Nos.61035003, 61175042, 61021062), the National 973 Program of China (Grant No. 2009CB320702), the 973 Program of Jiangsu, China (Grant No. BK2011005) and Program for New Century Excellent Talents in University (Grant No. NCET-10-0476).

References

1. Brown, R., et al.: Report to congress on server and data center energy efficiency: Public law 109-431. (2008)
2. Dasgupta, G., Sharma, A., Verma, A., Neogi, A., Kothari, R.: Workload management for power efficiency in virtualized data centers. Communications of the ACM **54**(7) (2011) 131–141
3. Calheiros, R., Ranjan, R., Beloglazov, A., De Rose, C., Buyya, R.: Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience **41**(1) (2011) 23–50
4. Verma, A., Ahuja, P., Neogi, A.: pmapper: power and migration cost aware application placement in virtualized systems. In: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, Springer-Verlag New York, Inc. (2008) 243–264
5. Wood, T., Shenoy, P., Venkataramani, A., Yousif, M.: Sandpiper: Black-box and gray-box resource management for virtual machines. Comput. Netw. **53**(17) (December 2009) 2923–2938
6. Kumar, S., Talwar, V., Kumar, V., Ranganathan, P., Schwan, K.: vmanage: loosely coupled platform and virtualization management in data centers. In: Proceedings of the 6th international conference on Autonomic computing, ACM (2009) 127–136
7. Dhiman, G., Marchetti, G., Rosing, T.: vgreen: A system for energy-efficient management of virtual machines. ACM Transactions on Design Automation of Electronic Systems (TODAES) **16**(1) (2010) 6
8. Rossi, F., Van Beek, P., Walsh, T.: Handbook of constraint programming. Volume 35. Elsevier Science (2006)
9. Campegiani, P.: A genetic algorithm to solve the virtual machines resources allocation problem in multi-tier distributed systems. In: Second International Workshop on Virtualization Performance: Analysis, Characterization, and Tools (VPACT09), Boston, Massachusett. (2009)
10. Bellur, U., Rao, C., SD, M.: Optimal placement algorithms for virtual machines. Arxiv preprint arXiv:1011.5064 (2010)
11. Xu, J., Fortes, J.: Multi-objective virtual machine placement in virtualized data center environments. In: Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom), Ieee (2010) 179–188