Placement Algorithm for Virtual Machines

Chetan S
Jeffrey John Geevarghese
Karthik R

April 1, 2010

1 Introduction

The placement algorithm for Virtual Machines(VMs) in a data center allocates various resources such as memory, bandwidth, processing power, etc. from a physical machine (PM) to VMs such that the number of PMs used is minimized.

This problem can be viewed as a multi-dimensional packing problem. The resource requests of VMs is considered as d-dimensional vectors with non-negative entries (balls). The resource available at each PM is considered to be a d-dimensional vector with a magnitude of 1 along each dimension (bins). The goal is to minimize the number of bins such that for every bin the sum of the vectors placed in that bin is coordinate-wise no greater than the bin's vector. Thus, the resource allocation problem is an instance of the d-dimensional Vector Bin Packing problem (VBP).

For d = 1, the VBP is identical to the 1-dimensional Bin Packing problem.

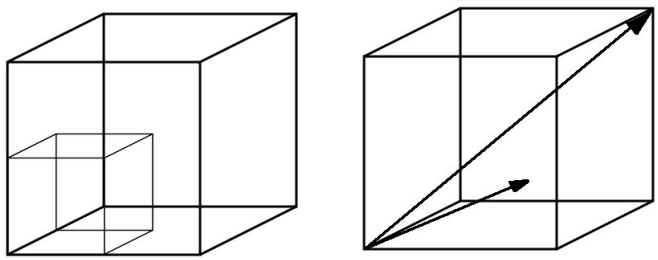

Figure 1: Bin-packing & Vector Bin Packing along 3-dimensions

2 Previous Work

One dimensional bin packing problem has been studied extensively. Fernandez de la Vega and Lueker [6] gave the first asymptotic polynomial-time approximation scheme (APTAS- See section 2.1). They put forward a rounding technique that allowed them to reduce the problem of packing large items to finding an optimum packing of just a constant number of items (at a cost of ϵ times OPT). Their algorithm was later improved by Karmarkar and Karp [7], to a $(1+log^2)$-OPT bound.

For 2-dimensional vector bin packing, Woeginger [9] proved that there is no APTAS. For higher dimensions, Fernandez de la Vega and Lueker [6] proposed a simple $(d+\epsilon)$-OPT algorithm, which extends the idea of 1-dimensional bin packing. Chekuri and Khanna [5] showed an $O(\log\ d)$-approximation algorithm that runs in polynomial time for fixed d. Bansal et al. [1] have recently improved this result, showing an $(\ln\ d + 1 + \epsilon)$-approximation algorithm for any $\epsilon \geq 0$

2.1 Asymptotic Polynomial-Time Approximation Scheme (APTAS)

Given a (deterministic) algorithm for the problem, we say that it has asymptotic approximation guarantee ρ if there exists a constant δ such that the value of the solution found by the algorithm is at most ρ.OPT(I) + δ for each instance I.

An APTAS is a family of polynomial-time algorithms such that, for each $\epsilon \geq 0$, there is a member of the family with asymptotic approximation guarantee $1 + \epsilon$ and $\delta \geq 0$.

3 Algorithms

The algorithms we present in this section are slight variants of the ones by Bansal et al. [1] and Chekuri & Khanna [5]. In this light, we present two approaches -

3.1 Reduction to the Set Cover problem

The vector bin packing problem can be reduced to the set cover problem. Given an instance I of the VBP, let S be the power set 2^I. We find a set S', a subset of S, such that each individual element of S' can be packed in a bin (a heuristic can be used for this). Thus, S' contains all the feasible bin packings. Now, we must pick elements(sets) from S' such that all VM vectors are covered using the least number of elements(sets). This is nothing but the Set Cover problem. The Set Cover problem has a 2-OPT algorithm (using LP-rounding technique).

The algorithm is as follows -

Algorithm $PackVectorsInBins(I)$

($*$ Packing the VM vectors into bins $*$)

Input: Set of VM vectors $\{p_1, p_2, ... p_n\}$; I

Output: Packed Bins and contents; B

1. $S = GeneratePowerSet(I)$
2. $S' = PackableInOneBin(S)$
3. $B = SolveSetCover(I, S')$
4. **return** B

The power set generation can be done in polynomial time. Some heuristics for obtaining the sets which are packable in one bin can be used (Heuristic can be made to run in polynomial time in the average case). Finally, the Set Cover problem can be solved using the standard LP formulation. Thus, the algorithm runs in polynomial time for most cases and the solution returned by the algorithm is atmost 2-OPT.

3.2 LP formulation

We use a variable x_{ij} to indicate if vector p_i is assigned to bin j. We guess the least number of bins m (can be done by binary search) for which the following LP is feasible. We also know that $m \leq$ OPT, because any OPT solution for the ILP would be a solution in the fractional LP as well. The constraints on x_{ij} are as follows -

$$\sum_j x_{ij} = 1 \qquad 1 \leq i \leq n \tag{1}$$

$$\sum_i p_i^k . x_{ij} \leq 1 \qquad 1 \leq j \leq m, 1 \leq k \leq d \tag{2}$$

$$x_{ij} \geq 1 \qquad 1 \leq i \leq n, 1 \leq j \leq m \tag{3}$$

Once we obtain a feasible solution, we can pack the vectors which have been assigned completely to the corresponding bins. The similar procedure can be repeated with fewer number of vectors remaining.

The algorithm is as follows -

Algorithm *PackVectorsInBins*(I)

($*$ Packing the VM vectors into bins $*$)

Input: Set of VM vectors {p_1, p_2, ... p_n}; I

Output: Packed Bins and contents; B

1. $S = I$
2. **while** ($S! = \phi$)
3. $P = FormulateJustFeasibleLP(S)$
4. $Q = LPSolve(P)$
5. **for** *every* j
6. **for** *every* i
7. **if** $x_{ij} = 1$
8. **do** *Pack vector i in bin j*
9. $S = S \setminus i$;

The algorithm can be proved to converge within a polynomial number of steps.

Theorem [5]. *Any basic feasible solution to the LP defined by Equations 1, 2 and 3 has atmost d.m vectors that are assigned fractionally to more than one machine.*

By the above theorem, we can see that the number of variables is brought down from $n.m$ to $d.m$, thus indicating that the number of VM vectors are strictly reducing in each iteration. Note that m also strictly decreases in each iteration.

References

[1] Nikhil Bansal, Alberto Caprara, and Maxim Sviridenko. Improved approximation algorithms for multidimensional bin packing problems. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 697–708, Washington, DC, USA, 2006. IEEE Computer Society.

[2] Nikhil Bansal, Jos R. Correa, Claire Kenyon, and Maxim Sviridenko. Bin packing in multiple dimensions: Inapproximability results and approximation schemes. *Math. Oper. Res.*, 31(1):31–49, 2006.

[3] Alberto Caprara. Packing 2-dimensional bins in harmony. In *FOCS '02: Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 490–499, Washington, DC, USA, 2002. IEEE Computer Society.

[4] Alberto Caprara and Paola Toth. Lower bounds and algorithms for the 2-dimensional vector packing problem. *Discrete Appl. Math.*, 111(3):231–262, 2001.

[5] Chandra Chekuri and Sanjeev Khanna. On multi-dimensional packing problems. In *SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, pages 185–194, Philadelphia, PA, USA, 1999. Society for Industrial and Applied Mathematics.

[6] Wenceslas Fernandez de la Vega and George S. Lueker. Bin packing can be solved within 1+epsilon in linear time. *Combinatorica*, 1(4):349–355, 1981.

[7] Narendra Karmarkar and Richard M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *SFCS '82: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 312–320, Washington, DC, USA, 1982. IEEE Computer Society.

[8] Hans Kellerer and Vladimir Kotov. An approximation algorithm with absolute worst-case performance ratio 2 for two-dimensional vector packing. *Operations Research Letters*, 31(1):35 – 41, 2003.

[9] Gerhard J. Woeginger. There is no asymptotic ptas for two-dimensional vector packing. *Inf. Process. Lett.*, 64(6):293–297, 1997.