仮想マシン組替えに対する厳密及びハイブリッドアルゴリズム

Exact and Hybrid Algorithms for Virtual Machine Reassignment

福永アレックス[*1] 網代育大[*2]
Alex Fukunaga Yasuhiro Ajiro

[*1] 東京工業大学グローバルエッジ研究院
Global Edge Institute, Tokyo Institute of Technology

[*2] NEC サービスプラットフォーム研究所
Service Platforms Research Laboratories, NEC Corporation

In modern data centers, virtual machines that provide various services are assigned to physical servers. Due to fluctuations in demand, it is sometimes necessary to migrate virtual machines between physical servers in order to eliminate overloading of computing resources such as CPU, memory, and bandwidth. The multidimensional VM reassignment is the problem of reassigning virtual machines to eliminate overloading while minimizing the number of virtual machines that are moved. We consider an exact search algorithm for finding the optimal solution to the VM reassignment problem, as well as a hybrid approach which uses heuristics to generate upper bounds for an exact algorithm.

1. Introduction

Server consolidation is the process of using virtual machine technology to consolidate multiple servers onto a set of fewer servers. [Vogels 08]. There is currently great demand in the IT industry for server consolidation products and solutions due to the significant improvement in energy efficiency and reduction in “server sprawl” (reduced hardware and datacenter space requirements) which can be achieved by server consolidation.

The server consolidation problem can be modeled as of vector packing problem. Consider a set of n virtual machines (VMs), where each VM has associated with it a d-dimensional demand vector, where each dimension corresponds to a resource type (e.g., CPU, RAM, network bandwidth). Given a set of m physical servers, each with a d-dimensional capacity vector, the server consolidation problem is the problem of assigning the n VMs to the m physical servers, such that each VM is assigned to exactly one physical server, and for every physical server, for each resource dimension, the sum of the demands for the resource is within the capacity of the physical server.

Suppose that due to changes in demand, the current resource requirements of the VMs differ from the initial projections when the initial server assignments were made, resulting in some servers being overloaded. It is possible to reassign virtual machines among the physical servers in order to rebalance the loads. However, migration of a virtual machine between servers incurs costs (direct costs include system administration costs, intangible costs are the costs of possible downtime for a service). The *Min-Cost Virtual Machine Reassignment Problem* (VMRP) is the problem of finding a new assignment of jobs to servers such that (1) no server is overloaded, and (2) the number of jobs that are moved between severs is minimized. In this paper, we consider algorithms for the VMRP. We propose an exact algorithm based on the IDA* search algorithm. We also investigated hybrid algorithms that combine the exact algorithms with heuristic algorithms.

2. An Exact Algorithm for the VM Reassignment Problem

The Min-Cost VM Reassignment Problem is a generalization of vector packing, which is, in turn a generalization of bin packing, which is NP-complete. Therefore, the VMRP is NP-complete. One possible approach is to modify an existing vector packing algorithm. However, this is difficult because successful vector packing branch-and-bound algorithms such as [Caprara 01] rely on techniques such a column generation and lower bounds, which, if applied straightforwardly, can prune the optimal VMRP solution. Thus, we developed an exact algorithms specifically designed to solve the VMRP, which is based on an algorithm for 1D bin packing repair proposed in [Fukunaga 08].

We search a space where each node represents a partially committed assignment of variables to values, and edges between the nodes represent a commitment of a variable to some value. For each variable, we represent its current value, as well as whether a commitment has been made to the value. A variable x is *committed* to value v at node N if x is assigned to v at N and every descendant of N, and *uncommitted* otherwise. At the root node of this search space, the variables are assigned the values of the initial assignment, and all variables are uncommitted. Finding the shortest path from the initial assignment to a goal state yields an optimal solution to the VMRP. The shortest path can be found using a tree search algorithm such as depth-first branch-and-bound or the IDA* algorithm [Korf 85]. We use a standard most-constrained variable ordering and a simple, random value ordering for all combinations of C-space, D-space, DFBNB, and IDA*.

Our search algorithm for the VMRP is the basic search strategy described above, where the lower bound (admissi-

---

連絡先: 福永アレックス、東京工業大学　東京都目黒区大岡山
2-12-1 S-18 fukunaga@is.titech.ac.jp

ble heuristic) applied at each node (for both branch-and-bound and IDA* search strategies) is computed as follows: For each oversubscribed server B, for each resource dimension, sort the uncommitted VMs assigned to B in non-decreasing order of demand, and count the number of items that must be removed from B in this order until the demand no longer exceeds capacity. We take the maximum of this number over the resource dimensions, then sum over all bins, which gives us our lower bound.

2.1 A Hybrid, Exact Algorithm

A weakness IDA* is that when problem instances become too difficult to solve optimally within a given amount of time, IDA* is useless because it only generates nodes which are guaranteed to have a cost no greater than the optimal solution, and can only return the optimal solution. For large problems, this can mean that IDA* spends all of its time looking for only the optimal solution, and when time runs out, IDA* can only return "failure"

In contrast, branch-and-bound offers "anytime" behavior, and can return solutions of improving quality, depending on the amount of time allocated. This allows branch-and-bound to be used as an approximation algorithm for difficult problems. On the other hand, branch-and-bound has the problem that is very easy for depth-first search to make a "mistake" at a shallow node in the search tree, and end up spending an enormous amount of time searching parts of the search space which do not contain good solutions. In fact, we have found that it can be extremely difficult for branch-and-bound to find any feasible solution to the VMRP.

Previous work has investigated the use of an iterative, heuristic algorithms for the VMRP which can find reasonable solutions quickly [Ajiro 08]. We have implemented a *hybrid*, exact algorithm for the VMRP, which first uses a heuristic algorithm (a relatively short run of a genetic algorithm based on the heuristic in [Ajiro 08]) to try to quickly compute a reasonable upper bound U. We then run the branch-and-bound described above with a starting upper bound of U. While the heuristic is not guaranteed to find an optimal solution, it is hoped that the heuristic can find a good upper bound which accelerates the branch-and-bound search.

3. Experimental Results

Here, we summarize the results of a preliminary, experimental evaluation of IDA*, branch-and-bound, and hybrid exact algorithms for the VMRP. We generated random VMRP problems instances, with $n \in \{8, 16, 20, 32, 40, 64\}$ virtual machines, and each VM had 2 resource dimensions (to model, for example, CPU load and RAM usage). Instance sets with various relative resource demands were generated, as well as various correlations between the demands on the two resources. In addition, we considered various demand patterns, based on [Ajiro 08]. This resulted in 385 different classes of problems on which IDA*, branch-and-bound, and the hybrid algorithm. For each class of problem, we generated 5 instances (total of 1925 instances). We executed each of the three algorithms on each of the instances, with a time limit of 2 minutes per instance per algorithm.

Although there is insufficient space to list the full results, the following is a summary of the main results:

- For problems with up to 20 items, IDA* found an optimal solution within seconds, and usually much faster than the branch-and-bound and hybrid algorithms.

- For problems with 32 or more items, IDA* was unable to find optimal solutions within the given time limit.

- For problems with 32 or more items, the hybrid algorithm always outperformed branch-and-bound, returning a better solution within the time limit. In general, using a heuristic to compute an upper bound significantly improved upon branch-and-bound, except on trivial problem instances where all algorithms found a solution immediately, in which case there is no advantage to using the hybrid algorithm.

Thus, our study indicates that for small problem instances, IDA* is the preferred algorithm, while for larger problem instances, the current IDA* algorithm can not return any solution, so the hybrid algorithm is the algorithm of choice.

4. Conclusions

We have proposed exact and a hybrid algorithms for the VMRP. Our experimental results show that problems with up to 20 VMs can be exactly solved consistently, while for larger problems the hybrid algorithm performed best. While this work focused on simply minimizing the number of VMs that are reassigned, the basic search strategy in our algorithms does not depend on having a unit cost function. Future work will investigate additional cost metrics which consider the characteristics of the VM.

参考文献

[Ajiro 08] Ajiro, Y.: Recombining virtual machines to autonomically adapt to load changes, in *Proc. 22nd Annual Conference of the Japanese Society for Artificial Intelligence* (2008)

[Caprara 01] Caprara, A. and Toth, P.: Lower bounds and algorithms for the 2-dimensional vector packing problem, *Discrete Applied Mathematics*, Vol. 111, pp. 231–262 (2001)

[Fukunaga 08] Fukunaga, A.: Search algorithms for minimal cost repair problems, in *Proc. Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Systems* (2008)

[Korf 85] Korf, R.: Depth-first iterative-deepening: an optimal admissible tree search, *Artificial Intelligence*, Vol. 27, No. 1, pp. 97–109 (1985)

[Vogels 08] Vogels, W.: Beyond server consolidation, *ACM Queue*, Vol. 6, No. 1 (2008)