Impact of Load Prediction Accuracy on Server Consolidation for Virtualized Resources

Cássio A. P. S. Alkmin* and Daniel de A. Cordeiro*
*Institute of Mathematics and Statistics
University of Sao Paulo
{cassiop, danielc}@ime.usp.br

Marco A. S. Netto† and Marcos D. Assunção†
†IBM Research
{mstelmar, marcosda}@br.ibm.com

***Abstract*—Virtualisation has become a key technology for simplifying resource management and reducing energy costs in large data centres. One of the challenges faced by data centres is to decide when, how, and which virtual machines (VMs) have to be consolidated into a single physical server. Several server consolidation strategies have been proposed and most of them share VM load prediction as a fundamental component. Therefore, a remaining question is to determine how accurate the load prediction needs to be in order to have the benefits of server consolidation. This paper evaluates X server consolidation strategies and the impact of load prediction accuracy on the number of VM migrations, energy savings, and SLA violations. We performed experiments using Google cluster workloads. The results indicate that X strategies require an accuracy level of Y% and the other K strategies require %A of accuracy.**

I. Introduction

One of the highest costs of current data centres is energy consumption. Over the last years, several data centres have started to use virtualisation to optimise resource management and reduce energy consumption. The benefits come from the fact that virtualisation allows several servers to be hosted on the same physical machines – concept known as *server consolidation*.

Numerous solutions have been proposed to know when, how, and which servers need to be consolidated [1]. These solutions basically consist in measuring the current load of the virtual machines, predicting the future load, and deciding whether virtual machines need to be resized or moved to another physical machine at a given time. A key open question is to know how heavily current server consolidation solutions rely load prediction.

This paper presents a study to quantify the load prediction accuracy necessary for a set of existing server consolidation solutions. This is particularly important as data centres usually cannot have precise information on future load, specially because customers process workloads from various areas. Therefore, the contributions of this paper are: (i) a detailed study to show the dependency between existing server consolidation algorithms and future load predictions; (ii) a methodology to perform this type of study; and (iii) results of such dependencies using workloads of a Google data centre.

March 8, 2013

II. Background and Related Work

Several server consolidation solutions are available in the literature. Their main framework consists in monitoring the virtual servers, predicting their future resource consumption, and deciding which action should be taken to optimise a given metric. Number of physical servers, number of virtual machine migrations, energy consumption, and server availability are examples of optimisation metrics. The solutions vary in aspects such as frequency when actions take place, whether they are proactive by using a prediction method or reactive by verifying resource consumption values, and costs for resizing and moving virtual machines.

One server consolidation solution is proposed by Khanna et al. [2], which consists in a dynamic management algorithm that is triggered when a physical server becomes overloaded or underloaded. The main goals of their algorithm are to: i) guarantee that SLAs are not violated; ii) minimise migration cost; iii) optimise the residual capacity of the system; and iv) minimise the number of physical servers used. Bobroff et al. [3] proposed and evaluated a dynamic server consolidation algorithm to reduce the amount of required capacity and the rate of SLA violations. The algorithm uses historical data to forecast future demand and relies on periodic executions to minimise the number of physical servers to support the virtual machines.

Speitkamp and Bichler [4], [5] described linear programming formulations for the static and dynamic server consolidation problems. They also designed extension constraints for limiting the number of virtual machines in a physical server, guaranteeing that some virtual machines are assigned to different physical servers, mapping virtual machines to a specific set of physical servers that contain some unique attribute, and limiting the total number of migrations for dynamic consolidation. In addition, they proposed an LP-relaxation based heuristic for minimising the cost of solving the linear programming formulations. Mehta and Neogi [6] introduced the ReCon tool, which aims at recommending dynamic server consolidation in multi-cluster data centres. ReCon considers static and dynamic costs of physical servers, the costs of VM migration, and the historical resource consumption data from the existing environment in order to provide an optimal dynamic plan of VMs to physical server mapping over time. Similarly, Verma et al. [7] developed the pMapper architecture and a set of server consolidation algorithms for heterogeneous virtualized resources. The algorithms take into account power and migration costs and the performance benefit

when consolidating applications into physical servers.

Wood et al. [8] developed the Sandpiper system for monitoring and detecting hotspots, and remapping/reconfiguring VMs whenever necessary. In order to choose which VMs to migrate, Sandpiper sorts them using a volume-to-size ratio (VSR), which is a metric based on CPU, network, and memory loads. Sandpiper tries to migrate the most loaded VM from an overloaded physical server to one with sufficient spare capacity.

Most of these solutions rely on monitoring and predicting future load. Server consolidation strategies may not be able to produce optimised actions if future load prediction is not accurate. For instance, Figure 1 illustrates a scenario with one physical machine that hosts two virtual machines. One physical machine is off. The prediction system detects that one of the virtual machines will require more computing power. As both virtual machines do not fit into the same physical machine, the management system switches on the second physical machine. If the virtual machine really requires more resources, it is moved to the second physical machine. If the prediction was not correct and the virtual machine's resource demand did not change, it remains on the first physical machine, thus wasting energy as the second physical machine was switched on. Given this scenario, the question investigated in this paper is therefore "What is the impact of load prediction accuracy on existing server consolidation solutions?".

[9]

III. Server Consolidation Strategies

This section describes server consolidation strategies that are evaluated in this paper. We also discuss possible effects of inaccurate load predictions.

A. Basic strategies

We call basic strategies those heuristics that are usually considered to solve bin packing problems. Given a set of virtual machines and a set of available physical machines:

- **First-Fit:** each virtual machine is allocated to the first physical machine that is able to provide the resources demanded by the virtual machine;
- **Best-Fit:** each virtual machine is allocated to the physical machine that can meet the virtual machine's demand and whose allocation results in the smallest residual capacity; and
- **Worst-Fit:** each virtual machine is allocated to the physical machine that can meet the virtual machine's demand and whose allocation results in the biggest residual capacity.

The algorithms search first for an already used physical machine, but if none of them could host the virtual machine, so one unused physical machine is selected (by the same constraints) to be "powered up" and to host the virtual machine.

In the simulations, we used the "decreasing" version of these strategies, i. e., First-Fit Decreasing, Best-Fit Decreasing and Worst-Fit Decreasing. In these versions, one step is added before the ordinary algorithm be executed: the virtual machines are decreasingly ordered (in a lexicographical order).

Migration Control Strategies: While the virtual machines are running, their resources' demands can change or not. A migration control strategy is used to decide when a virtual machine should be migrated. Ferreto et al. [1] proposed a migration control strategy based on virtual machines' demands changes: if a virtual machine's demands change, so the virtual machine should be migrated. We added one more constraint in Ferreto et al.'s strategy and evaluated the following situation too: if a virtual machine's demands change in a way that can overload the physical machine, so the virtual machine should be migrated.

B. Khanna et al. [2] strategy

Khanna et al. proposed algorithms trying "*to resolve SLA violations by reallocating VMs to PMs, as needed*". The proposed algorithm is reactive and is triggered to achieve two objectives: to resolve (or avoid) SLA violations and to turn off machines with low utilization.

Before explaining the algorithm, we have to set some terms. Each virtual machine (VM) was represented as a d-dimensional vector where each dimension represents one of the resources, and the resource utilization of VM_i at time t is represented by $U_i(t)$. Each physical machine, PM_j, has a fixed capacity C_j in a d-dimensional space. The $L_j(t)$ vector is the sum of $U_k(t)$, for all VM_k allocated in the PM_j at time t, and $R_j(t)$ is the residual capacity vector of the j^{th} physical machine at time t. In each i^{th} dimension, the associated resource has a *knee* value n_i, chosen as a superior limit of this resource utilization (if the utilization of this resource in one physical machine extrapolates this value, probably a SLA violation will occur).

! This dynamic algorithm runs at discrete time instances $t_0, t_1, \ldots$ to perform reallocation when triggered via a resource threshold violation alert. !

An initial allocation state is provided, and the d resources are monitored for each virtual and physical machine. The algorithm to migrate VMs from PM_i is triggered when a resource utilization in the physical machine PM_i reaches its *knee* value, or when the PM_i utilization falls below a low mark (all VMs allocated there are migrated in this case).

1) ! Select the VM from PM_i with the lowest utilization and move it to a physical machine which has the least residual capacity big enough to hold this VM !;
    a) If no physical machine has big enough residual capacity to hold this VM, then instantiate a new physical machine and allocate the VM to that machine.
2) If the SLA constraints aren't satisfied yet, repeat the first step until the constraints are satisfied;

No strategy was showed to allocate new virtual machines, so we decided to use the same strategy presented to choose whose physical machine would host a virtual machine that has to be migrated.

C. Strategy 3

IV. Evaluation

We evaluated the heuristics described in Section III considering different accuracy level of load prediction. We developed

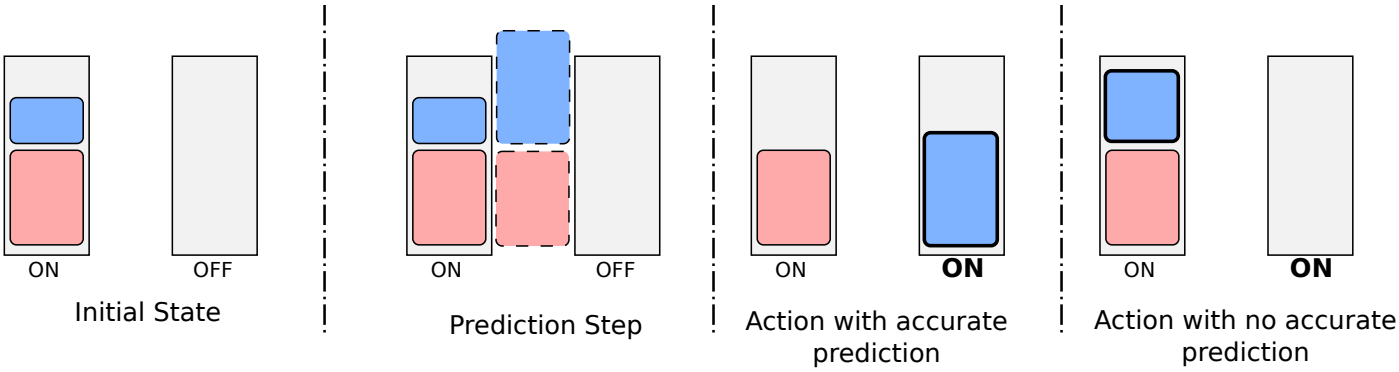

Fig. 1. Illustrative scenario where one physical machine hosts two virtual machines. One physical machine is off. If the prediction is correct, the second physical machine is switched on and the virtual machine is moved to it. If the prediction is not correct, the second machine is switched on but no virtual machine is moved to it, thus wasting energy.

a simulator to perform the experiments and utilised workloads from a Google data centre. This section presents the workloads, metrics, and result analysis.

A. Experiment Setup

We performed the evaluation using a Google cluster trace[1] [10]. This trace represents executions of tasks on a multi-purpose cluster composed of approximately 12 thousand machines.

B. Result Analysis

V. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank CAPES for financial support of research project.

References

[1] T. C. Ferreto, M. A. S. Netto, R. N. Calheiros, and C. A. F. De Rose, "Server consolidation with migration control for virtualized data centers," *Future Generation Computer Systems*, vol. 27, no. 8, pp. 1027–1034, 2011.

[2] G. Khanna, K. Beaty, G. Kar, and A. Kochut, "Application performance management in virtualized server environments," in *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*. IEEE, 2006, pp. 373–381.

[3] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic placement of virtual machines for managing sla violations," in *Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on*. IEEE, 2007, pp. 119–128.

[4] M. Bichler, T. Setzer, and B. Speitkamp, "Capacity planning for virtualized servers," in *Workshop on Information Technologies and Systems (WITS), Milwaukee, Wisconsin, USA*, 2006.

[5] B. Speitkamp and M. Bichler, "A mathematical programming approach for server consolidation problems in virtualized data centers," *Services Computing, IEEE Transactions on*, vol. 3, no. 4, pp. 266–278, 2010.

[6] S. Mehta and A. Neogi, "Recon: A tool to recommend dynamic server consolidation in multi-cluster data centers," in *Network Operations and Management Symposium, 2008. NOMS 2008. IEEE*. IEEE, 2008, pp. 363–370.

[7] A. Verma, P. Ahuja, and A. Neogi, "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, pp. 243–264.

[8] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Sandpiper: Black-box and gray-box resource management for virtual machines," *Computer Networks*, vol. 53, no. 17, pp. 2923–2938, 2009.

[9] M. A. S. Netto, C. Vecchiola, M. Kirley, C. A. Varela, and R. Buyya, "Use of run time predictions for automatic co-allocation of multi-cluster resources for iterative parallel applications," *Journal of Parallel and Distributed Computing*, vol. 71, no. 10, pp. 1388–1399, 2011.

[10] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proceedings of the Third ACM Symposium on Cloud Computing*. ACM, 2012, p. 7.

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