Distributed Autonomous Virtual Resource Management in Datacenters Using Finite-Markov Decision Process

Liuhua Chen, Haiying Shen and Karan Sapra
Department of Electrical and Computer Engineering, Clemson University
({liuhuac, shenh, ksapra}@clemson.edu)

Background and Motivation

To provide robust infrastructure as a service (IaaS), clouds currently perform load balancing by migrating virtual machines (VMs) from heavily loaded physical machines (PMs) to lightly loaded PMs. Previous load balancing methods have the following **disadvantages:**

1. A delay to achieve load balance;
2. No long-term load balance;
3. High overhead.

To overcome these problems, we propose a proactive Markov Decision Process (MDP)-based load balancing algorithm.

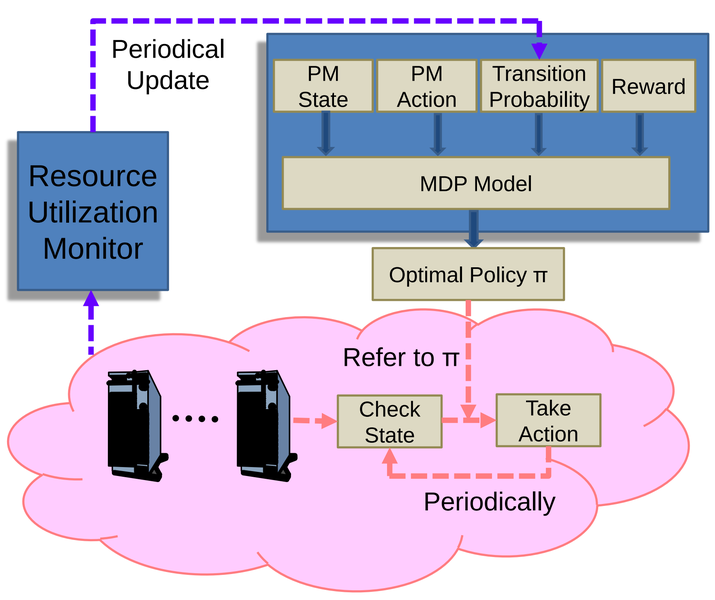

Figure 1 Overview of MDP-based load balancing.

Overview:
MDP classifies the resource utilization degree of a PM to states, and treats VM migrations as actions. MDP learns the probabilities of PM state transitions from traces, and computes the optimal polity that PMs should perform in order to achieve maximum rewards.

The optimal policy π is a mapping from states to actions. Following a policy π , every PM:

1. Determines its current state s;
2. Executes action π(s);
3. Periodically go to step 1.

Design Details

Challenges:

1. MDP components must be well designed for low overhead;
2. Transition probabilities in the MDP must be stable.

To handle challenge 1, our designed MDP intelligently uses a PM load state as a state and records the transitions between PM load states by moving out a VM in a specific load state.

To handle challenge 2, we carried out experiments to study the transition probability and show that the transition probability matrix remains stable.

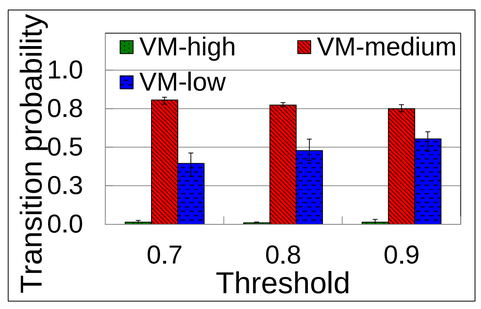

Figure 2 Probability of state transitions.

A PM-high has similar probabilities to transit to state medium under slightly varying threshold, when it migrates VM-high, VM-medium and VM-low.

MDP components:

1. **State**: classification of resource utilization of a PM.
2. **Action**: a migration of VM in a certain state (VM-State).
3. **Probability**: the probability that state s will transit to state s' after taking action a.
4. **Reward**: given after transition to state s' from state s by taking action a in order to encourage PMs to become lightly loaded.

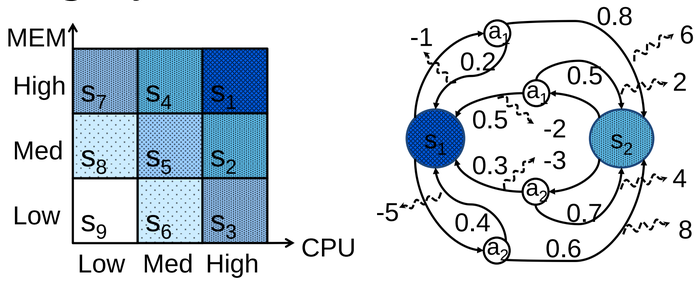

Figure 3 Example of a simple MDP.

Advantages of MDP-based load balancing:

1. Reduces SLA violations and achieves long-term load balance;
2. Reduces the overhead and delay;
3. Build one MDP used by all PMs.

Experimental Results

We conducted trace-driven experiments on CloudSim. We implemented two versions: MDP uses the MDP model for identifying VMs to migrate; and MDP* uses the model for both VM selection and PM selection. We compared them with Sandpiper [1] and CloudScale [2].

References:
[1] T. Wood, P. J. Shenoy, A. Venkataramani, and M. S. Yousif. Black-box and gray-box strategies for virtual machine migration. In Proc. of NSDI, 2007.
[2] Z. Shen, S. Subbiah, X. Gu, and J.Wilkes. CloudScale: Elastic resource scaling for multi-tenant cloud systems. In Proc. Of SOCC, 2011.

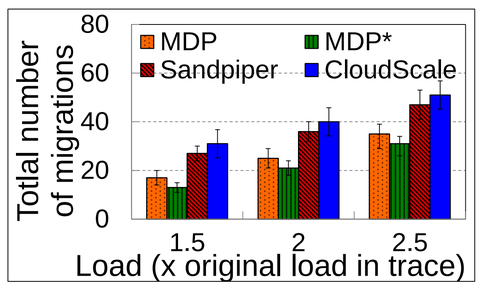

Figure 4 The number of VM migrations

Result: MDP*<MDP<Sandpi per<CloudScale

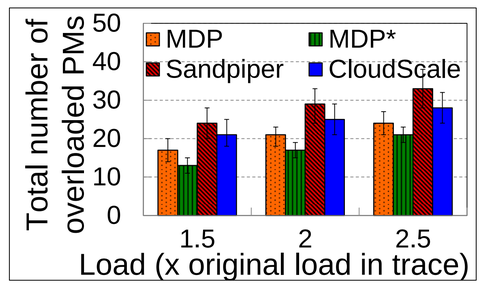

Figure 5 The number of overloaded PMs.

Result: MDP*<MDP<CloudScale<Sandpiper

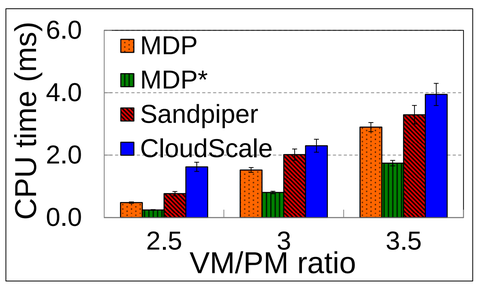

Figure 6 The CPU time consumption.

Result: MDP*<MDP<Sandpiper<CloudScale

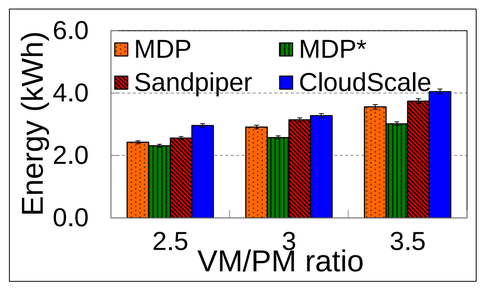

Figure 7 Energy consumption.

Result: MDP*<MDP<Sandpiper<CloudScale

Acknowledgments

U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006, CNS-1249603, Microsoft Research Faculty Fellowship 8300751.

Future Work

Make our algorithm fully distributed to increase its scalability.