Enhancing DSCP simulator to realistic than simplistic scenarios

Aman Mangal (100050015)
Amit Panghal (100050016)

October 7, 2013

Contents

1 Introduction

Dynamic Server Consolidation Problem (DSCP) aims at effective management of resources (CPU, Memory etc.) in a data center environment by incorporating workload dynamics of the system. The goal is to minimize the cost of data center operations by minimizing number of physical machines required for given workloads of set of applications using virtualization techniques while still meeting SLA requirements. The static allocation of virtual machines running these applications to a set of physical machines is not enough. Thus, various dynamic online algorithms have been devised to minimize the total cost. These algorithm are studied and tested on simulators since real test is very costly.

Most of the simulators are designed specifically for a particular DSCP algorithm and are bound to simplistic assumptions made by them. There is no way to introduce experimental randomness without separating the algorithm from test environment. Some of assumptions made by most of the algorithms, hence their simulators are :

- CPU utilization of a physical machine as a parameter for SLA violation instead of reponse time of the application or Virtual Machine
- Migration time independent of utilization of resources
- All migrations happen at the same time
- Phase duration is assumed to be constant
- constant workload for a given phase with no randomness

2 Keywords

SLA = Service Level Agreement
DSCP = Dynamic Server Consolidation Problem
Phase = least unit of interval after which the measurements (CPU/MEM utilization, check for SLA violation etc) are retaken

3 Problem Statement

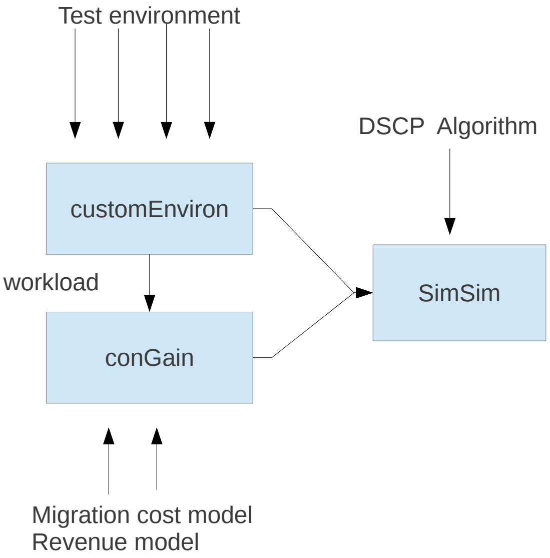
As mentioned above algorithm-specific DSCP simulators inherit the simplicity of their algorithm and are limited to a very simple test scenarios which

may be far away from reality. To combat these problems, we propose a simulator framework, SimCon that separates the test environment form the DSCP algorithm environment, enabling us to introduce randomness into the experiment.
This framework will consist of 3 major divisions

1. **customEnviron**:
   Given a workload, say number of requests vs time for each application, output a simplistic workload that can be input to second divison, conGain.

2. **conGain**:
   given the output workload from **customEnviron** and an allocation(vm to pm map) by algorithm, checks for any SLA violations and calculates the profit/loss for a duration. It will have a migration cost and a revenue model.

3. **simSim**:
   simulates a given DSCP algorithm online using conGain and customEnviron.



References

[1] 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, 2008

[2] G. Khanna, K. Beaty, G. Kaur, and A. Kochut, "Application performance management in virtualized server environments," in Network Operations and Management Symposium, 2006