**Distributed Computing Over The Internet Project Proposal**

# Improved Reward Cost Utilities For Markov Decision Process Elastic Computing

Raymond Jacobson

## Background

Naskos, et al. propose, in *Dependable Horizontal Scaling Based On Probabilistic Model Checking*, a method to ensure accurate on-demand scaling of cloud computing systems. To promote dependable elastic policies, Naskos, et al. propose a formalized method to express horizontal scaling decisions with Markov Decision Processes (MDP). In the MDP model proposed, states are representative of the number of Virtual Machines, or VMs, ($s_{num\_vms}$) in a given elastic computing cluster. Actions are determined by 3 potential decisions given a current state: add (*add*), remove (*rem*), and no operation (*no-op*). A reward value is assigned to a particular state given by the formula:

$$r = \begin{cases} 1 + 1/vms\_num & if\ latency \leq x \\ 0 & if\ latency > x \end{cases}$$

where x is a maximum latency threshold.

Using log data from previous loads on the elastic cloud, the MDP is solved periodically and a transition from one state to another is followed put the system in a state where it properly handles the current workload.

## Proposal

The focus of this work is provide MDP based solutions for determining elastic policies with a more robust and accurate method to assignment of rewards. In Naskos, et al.'s work, rewards are purely assigned to states in the MDP and do not capture any differences that might occur between transitions themselves. In other words, transition probabilities in the MDP are not described. Furthermore, although the reward utility function described in Naskos, et al.'s work generally addresses the needs for a system to not be under or over provisioned, it doesn't precisely capture intricacies of latency, locality, or the time and fiscal impact of new virtual machine deployment or load balancing.

In this project, it is proposed that a novel and more accurate method for calculating the MDP rewards used for elastic policy making can be achieved by giving weight to a range of other learnable factors. The end goal of such work would be to *learn* an accurate reward utility function through methods of supervised learning, but this immediate project is intended as a step toward that direction.

## Previous Work

The work proposed by Naskos, et al. suggests a need for future work in assigning rewards to actions (state-to-state transitions) represented by $R(s_i, s_j)$ rather than purely state rewards ($Rs_i$). Although the paper acknowledges, the potential room for improvement, the formalization of the elastic policy making process, provides an extensible model for implementing new methodologies for reward utilities.

**Benefit Distributed Computing**

As internet usage continues to increase year over year, so does the amount of fluctuating workloads, which motivates the need for highly dependable elastic policies. Current state of the art elastic computing systems (Amazon EC2, for example) currently only employ rule-based policies. Dependable systems have been proposed for elastic policy making, but they are currently in their infanancy. A good reward utility function would provide elastic computing with highly dependable policy making ability, a necessity in the age of distributed computing over the internet.

**References**

[1] Athanasios Naskos, Emmanouela Stachtiari, Anastasios Gounaris, Panagiotis Katsaros, Dimitrios Tsoumakos, Ioannis Konstantinou, Spyros Sioutas, "Dependable Horizontal Scaling Based on Probabilistic Model Checking", CCGRID, 2015, 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid) 2015, pp. 31-40, doi:10.1109/CCGrid.2015.91