Chapter 11
Cloud Load Balancing and Reinforcement Learning

Abdelghafour Harraz  
Mohammed V University, Morocco

Mostapha Zbakh  
Mohammed V University, Morocco

ABSTRACT

Artificial Intelligence allows to create engines that are able to explore, learn environments and therefore create policies that permit to control them in real time with no human intervention. It can be applied, through its Reinforcement Learning techniques component, using frameworks such as temporal differences, State-Action-Reward-State-Action (SARSA), Q Learning to name a few, to systems that are be perceived as a Markov Decision Process, this opens door in front of applying Reinforcement Learning to Cloud Load Balancing to be able to dispatch load dynamically to a given Cloud System. The authors will describe different techniques that can used to implement a Reinforcement Learning based engine in a cloud system.

INTRODUCTION

Cloud Computing is the new paradigm in which all nodes having access to the internet are able to communicate together through high speed networks, this creates an internet of things, where the need of managing the communication between servers and clients emerges as a critical problematic in order to avoid congestion and out of service states. Cloud Load Balancing takes on that mission by creating
an additional intelligence layer on top of the internet of things to regulate, watch, control and in some cases possibly plan on the resources available and their ability to handle transported traffic.

Cloud load balancing and load balancers in general are classified into two main types, static and dynamic load balancers. Static load balancers handle traffic and load with no special knowledge of the inner characteristics of the traffic being transported, they don’t change their behaviour based on runtime in terms of response time, network latency, pick times, resources availability in virtual machines, or other cloud system maintenance operations such as hot or off line virtual machines relocation between data centres, upsizing or downsizing of machines.

Static load balancers are simple to implement, and are not heavy consumer of the resources of the machine in which they sit, they make fast decisions therefore they don’t introduce significant latency to the processing chain, they are in most cases transparent in term of adding more service time to the queueing mechanism, but they also hold very little intelligence and they might miss on accurate decision actions in regards of the system they are load balancing.

Dynamic load balancers in the other hand are built with the ability to watch load changes, keep track of it, escalade info to a master node that is responsible of making the load balancing decision, or keeping a local vector of changes in each node, and then decide what is the most accurate actions for the next load balancing iteration. Dynamic load balancers are complex to implement, and are heavy consumer of the resource of the machine in which they sit. Communication between nodes might suffer from network and inter-connexion issues, and in the case when a master node is implemented, this one is a single point of failure and need to be load balanced itself.

In this perspective, the choice of the load balancer to use is a question of what type of traffic is being handled by the cloud system, and how variant it is, to be able to architect the system efficiently, and give a deep insight on how it will behave in the future to be able to plan and control it.

Making the right decision or taking the most accurate action in a given situation regarding a system requires having full knowledge about that system. If we suppose we have knowledge about arrival rate and average service time of a given cloud, we can model it as a queue theory problem and therefore compute and draw a mathematical model of the system to be able to control it in real time to reduce reaching overloaded state with a given certainty.

Since in most cases the system does not expose these kind of information:

- Arrival distribution is hidden by the type of requests made to the cloud and no benchmark can give an exact form of its moments.
- Exit distribution is completely controlled by the current available resources in the cloud.
www.igi-global.com/chapter/cloud-computing-technologies-for-green-enterprises/189368?camid=4v1a

Risk Management in the Cloud and Cloud Outages
www.igi-global.com/chapter/risk-management-in-the-cloud-and-cloud-outages/119929?camid=4v1a