International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

Web auto configuration for n-tier in VM based Dynamic Environment by Reinforcement Learning Approach

Keyur P Prajapati , Akash Dave

Asst.Prof, MBIT, New V.V.Nagar

Abstract: Web system is power with optimize configuration parameters. For the Dynamic web system configuration is the crucial part. As on today for better web services availability, configuration is the very important parameter. For the web server and Application server hundred of config parameters are there for the better deployment and service availability [3].

In this paper we are going to describe the methodology of reinforcement learning which will how optimize all the parameters with best results in terms of web usability.

Introduction

Web systems for Apache and Tomcat bases applications normally contain a large number of parameters; their settings are crucial to systems performance and service availability. Manual configuration based on operator's experience is a lethargic and error-prone task. Current studies notify that more than 60% root causes of Internet service outages was due to false configuration caused by administrator mistakes. The configuration challenge is due to a number of reasons. First is the scalability of system and complexity which introduce more and more configurable parameters to a level beyond the capacity of an average-skilled operator. For example, both of the Apache server and Tomcat server have more than a hundred configurable [8] parameters to set for different tuning environments. In a multi-component system, the interaction [2] between the components makes performance tuning of the parameters even harder.

Past studies devoted to autonomic configuration of web systems. Most of them focused on performance parameters tuning for dynamic traffic in static environments. Their optimization approaches are very hard applicable to online setting of the parameters in VM-based dynamic platforms due to their different level complexity. There were a few control approaches targeted at online tuning in response to changing workload [5]. They were largely limited to tuning of single Max Client parameter because of the inherent workload balancing complexity

In this paper, we propose a reinforcement learning approach, namely RAC, for automatic configuration of multi-tier web systems in VM-based dynamic environments. Reinforcement learning is a process of taking actions based on previous results. For a web system, it's possible configurations form a state space. We define actions as reconfiguration [1] of the parameters. Reinforcement learning is intended to determine appropriate actions at every step to maximize the highest reward. Recent studies showed the feasibility of RL approaches in resource allocation [7] power management job scheduling in grid and self-optimizing memory controller and auto tuning config parameters. To best of our knowledge, the RAC approach should be the first one in the application of the RL principle to automatic configuration of web systems.

RL will definitely help the system admin for configuring the files at once. Rest at all it will apply its own algorithms based on past rewards and finally optimize the whole system at its own best shown in fig [1].

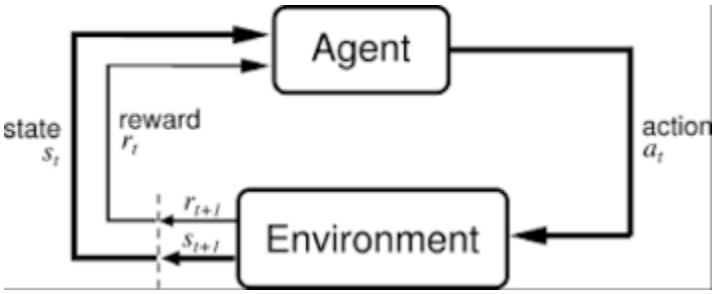

Fig 1.

Challenges in Website Configuration

1) Match Configuration to Workload:

   Based on dynamic environment lot of configurations is needed to be set to achieve performance hike. To optimize web server and application server's config parameters, one agent (watcher) would be there and it has to take its own decision to apply result and dispatch them to achieve good result in terms of performance [4].
2) Match Configuration to Dynamic VM Environments:

   Now in days VM are quietly use to host web sites and web applications. Possibly more than one OS are to be hosted in VM [6] and so we can manage resources in between (like storage and Memory).
   Our main aim is how to manage resources in between two or more OS through VM.

   If we are talking about single OS in VM and also it can communicate to host OS and manage resources but it toughest job to deal with OS and it looking like to impossible task to play with OS internals as shown in fig [2].

   So finally here is the representation of Single VM with two guest OS and Web Servers. To deal with other OS in same VM and borrow or send its resources to achieve high throughput and law latency.

- Proposed research work meet the requirement of dynamic web movement with large set of databases with stop-the-world hike.
- Architecture that enables auto config implementations for the context of web.
- Proposed work fit in short-pauses [5] with low latency and high throughput.
- Provide some of the performance metrics which will help others in testing these implementations in different environments.

Reinforcement Learning Approach to Auto configuration

1) Parameter Selection and Auto-configuration

   In VM with existing execution of web, RL agent will decide parameters to be set and deploy it.
   RL agent's crucial task is to decide the list of parameters is to be optimized. Because of inter related configuration parameters, to load auto scalar is also the responsibility of RL agent.

2) RL-based Decision Making

Reinforcement learning is a process of learning through interactions with an external environment (or the web system in this paper). The reconfiguration process is typically formulated as a finite Markov decision process (MDP), which consists of a set of states and several actions for each state.

State Space: For the online auto-configuration task, we define a state as possible system configuration. For the selective group of n parameters, we represent a state by a vector in the form as: si = (Para1, Para2, · · ·, Paran).

Action Set : We define three basic actions: increase, decrease, and keep associated with each parameter. We use a vector ai to represent an action on parameter i. Each element itself is a 3-element vector, indicating taken/not taken (1/0) of three actions. For example, the following
notation represents an increase action on parameter i: a increase i = (· · · , Parai(1, 0, 0), Paran(0, 0, 0)).

Immediate Reward. The immediate reward should correctly reflect the system performance. The immediate reward r at time interval t is defined as rt = SLA − perft, where SLA is a reference time predefined in Service Level Agreement, and perf is measured response time. For a given SLA, a lower response time returns a positive reward to the agent; otherwise the agent will receive a negative penalty.

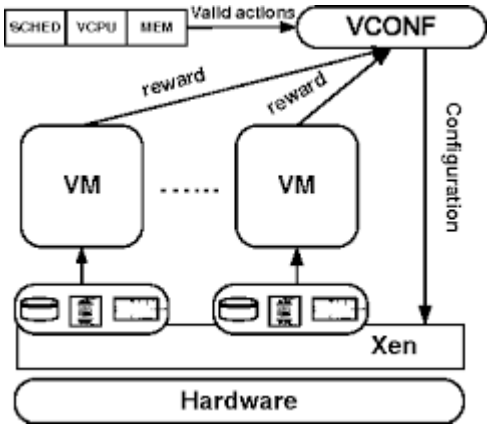

Fig 2

Effect of Exploration

How to balance exploration and exploitation is one of the challenges in online RL algorithms. Insufficient explorations would result in suboptimal configurations while too much exploration would incur prohibitive performance degradation. Effect of the exploration rate in RAC performance is very important part. Two types of explorations were considered: in batch training and in online learning.

Conclusion

In this paper, we propose a reinforcement learning approach, namely RAC, towards automatic configurations of multi-tier web systems in VM-based dynamic environment. To avoid initial learning overhead, we equip the RL algorithm with efficient heuristic initialization policies. Experiments
in a multi-tier web system showed that RAC is applicable to online system configuration adaptation in the presence of variations in both workload and VM resources. It is able to direct the web system to a near-optimal configuration within less than 25 trial-and-error iterations.

References:

1 J. Rao, X. Bu, C.-Z. Xu, L. Wang, and G. Yin. Vconf:a reinforcement learning approach to virtual machine autoconfiguration. In *ICAC*, 2009.

2 A. A. Soror, U. F. Minhas, A. Aboulnaga, K. Salem, P. Kokosielis, and S. Kamath. Automatic virtual machine configuration for database workloads. In *SIGMOD Conference*, 2008.

3 G. Tesauro. Online resource allocation using decompositional reinforcement learning. In *AAAI*, 2005.

4 J. Wildstrom, P. Stone, and E. Witchel. Carve: A cognitive agent for resource value estimation. In *ICAC*, 2008.

5 Naskos, Athanasios, Anastasios Gounaris, and Spyros Sioutas. "Cloud elasticity: a survey." *International Workshop on Algorithmic Aspects of Cloud Computing*. Springer, Cham, 2015.

6 Wei, Jianbin, and Cheng-Zhong Xu. "eqos: Provisioning of client-perceived end-to-end qos guarantees in web servers." *IEEE Transactions on Computers* 12 (2014): 1543-1556.

7 Gutierrez, Felipe. "Web Applications with Spring Boot." *Pro Spring Boot 2*. Apress, Berkeley, CA, 2019. 87-125.

8 Peinado, Ignacio, and Manuel Ortega-Moral. "Making web pages and applications accessible automatically using browser extensions and apps." In *International Conference on Universal Access in Human-Computer Interaction*, pp. 58-69. Springer, Cham, 2014.