2017 IEEE 10th International Conference on Cloud Computing

How to Supercharge the Amazon T2: Observations and Suggestions

Feng Yan[1], Lihua Ren[2], Daniel J. Dubois[3,4], Giuliano Casale[3], Jiawei Wen[2], and Evgenia Smirni[2]

[1]University of Nevada, Reno, Reno, NV, USA, fyan@unr.edu
[2]College of William and Mary, Williamsburg, VA, USA, {lren01,jwen01,esmirni}@cs.wm.edu
[3]Imperial College London, London, UK, {daniel.dubois,g.casale}@imperial.ac.uk
[4]Northeastern University, Boston, MA, USA, d.dubois@northeastern.edu

***Abstract*—Cloud service providers adopt a credit system to allow users to obtain periods of performance bursts without additional cost. For example, the Amazon EC2 T2 instance offers low baseline performance and the capability to achieve short periods of high performance using CPU credits. Once a T2 instance is created and assigned some initial credits, while its CPU utilization is above the baseline threshold, there is a transient period where performance is boosted and the assigned CPU credits are used. After all credits are used, the maximum achievable performance drops to baseline. Credits accrue periodically, when the instance utilization is below the baseline threshold. This paper proposes a methodology to increase the performance benefits of T2 by seamlessly extending the duration of the transient period while maintaining high performance. This extension of the high performance transient period is combined with proactive migration to further take advantage of the initially assigned credits. We conduct experiments to demonstrate the benefits of this methodology for both single-tier and multi-tier applications.**

***Keywords*-busrtable performance; transient period; proactive migration; credits; T2 instance;**

I. Introduction

Cloud computing [1], [2] has become nowadays a well established paradigm [3], [4]. Many cloud platforms, such as Amazon EC2 [5], Google Compute Engine [6], and Microsoft Azure [7], offer flexible cloud computing services, enabling cloud users to quickly launch jobs by requesting the desired amount of resources through the Internet without maintaining their own hardware, paying only for the need-based resources.

As a widely-used cloud platform, Amazon EC2 has been extensively studied [8], [9], [10], [11], [12], [13], [14]. These studies range from comparing the performance of EC2's offerings to other cloud platforms, to the peculiarities of specific instance offerings of EC2, and to performance comparisons when applications (single-tier and multi-tier) are migrated from a traditional data-center environment to Amazon EC2.

The focus on this paper is on the most opportune usage of Amazon EC2's T2 instance types. T2 instances are initially allocated CPU credits, which can be spent to allow bursts of performance above the baseline, and periodically accrued then the load is under the baseline. These instances are designed for applications that do not pose consistent demands to CPU resources, for example, web services characterized by short bursts of heavy load. In this work we focus not only on making the most efficient use of T2 instances for this type of cloud services, but also for cloud services that require non-bursty, i.e., consistent CPU demands.

We consider two types of benchmarks: single tier benchmarks with different intensities in CPU demands and TPC-W, a multi-tier application, that is a well accepted transactional web benchmark that simulates a business oriented transactional web server [15]. For the single-tier benchmarks, we show that by using `cpulimit` [16] it is possible to increase the initial transient time period where CPU is used well-above its baseline T2 performance. It is possible to extend this transient period of high performance up to five times at the beginning of the instance launch. For such cloud services, it is possible to estimate the duration of this transient period where performance is sustained at a much higher level than baseline and estimate when CPU throttling stabilizes and periodic bursty allocation starts. For the multi-tier case, we also observe that the use of `cpulimit` can be tremendously beneficial for user-perceived performance, but the duration of the transient, high performance period cannot be easily predicted. Yet, in both cases we show that for applications that execute for a short term (up to several hours for the case of TPC-W), it is possible to achieve superior performance than the baseline one offered by T2. Beyond the transient period, where the CPU is throttled by the burstiness mechanism, the improvements of `cpulimit` are small. This makes us consider proactive migration, i.e., enable a checkpoint/restart mechanism by launching a new instance when the steady-state low performance period starts. Previous work [17] illustrates how to launch several *t2.micro* instances across different locations to avoid the CPU throttling penalty for long-duration jobs, but its major shortcoming remains the migration frequency, which can result in significant performance penalties. In this paper we show that `cpulimit` can significantly reduce the migration frequency of long-running jobs on T2. Our experiments demonstrate that we can effectively reduce the migration frequency up to 80% and that the proposed proactive approach enables a seamless migration scheme.

II. Background

An Amazon EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure. AWS offers a host of different types of instances, each optimized for

2159-6190/17 $31.00 © 2017 IEEE
DOI 10.1109/CLOUD.2017.43