Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 621 – 627

Procedia Computer Science

The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)

Towards Identifying Performance Anomalies

Haroon Malik[a]*, Elhadi M. Shakshuki[b]

[a]*Weisberg Division of Computer Science, Marshall University, WV, USA*
[b]*Jodery School of Compueter Science, Acadia, Univeristy, NS, Canada*

Abstract

Large-scale-software systems (LSSs) are composed of hundreds of subsystems that interact with each other in an unforeseen and complex ways. The operators of these LSSs strictly monitor thousands of metrics (performance counters) to quickly identify performance anomalies before a catastrophe. The existing monitoring tools and methodologies have not kept in pace with the rapid growth and inherit complexity of these LSSs; hence are ineffective in assisting practitioners to effectively pinpoint performance anomalies. We propose a methodology that uses entropy analysis to assist practitioners/operators of LSSs in quickly detecting underlying anomalies in the system. Our performance tests conducted on an open source benchmark system reveal that the proposed methodology is robust in pinpointing anomalies, do not require any domain knowledge to operate, and avoid information overload on practitioners.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Performance counter; Large scale systems; Datacenter; Performance

1. Introduction

Today's large scale systems (LSSs) such as Facebook, Google, Amazon and many other datacenters comprise hundreds or thousands of machines running complex software applications that require high availability and responsiveness. They provide composite services, support a large user base and handle complex business demands.

* Corresponding author. Tel.: +1-304-696-5655.
E-mail address: malikh@marshall.edu

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.04.140

In line with Lehman's laws of continuing change and increasing complexity[1], the periodic monitoring of such LSS has become more critical and challenging than before since processing is spread across hundreds of subsystems and millions of hardware nodes (and users). These LSSs must be carefully monitored for performance anomalies before a serious harm is done[2-4]. A performance anomaly is an unexpected situation that causes system to deviate from abiding its Server Level Agreements (SLAs)[5-8]. Its symptoms include, but not limited to delayed response time, increases latency, decreased throughput and in cases, system hanging, freezing and crashing under heavy workload; usually introduced into the system by operator errors, hardware software failures, resource over-/under-provisioning, and unexpected interaction between geographically distributed system components[6].

LSSs are usually service oriented systems and generate revenue by providing composite services to a large user base. Any discrepancy in their performance can cause huge monetary losses. For example, an hour-long PayPal outage due to periodic maintenance may have prevented up to $7.2 million in customer transactions[9]. Similarly, an overloading of Google Server resulted into thousands of accounts being inaccessible for several days, worst many contents of many of the many of the clients were lost. Therefore, the operator of these LSSs continuously monitor their system to identify performance anomalies so a fix can be made quickly[10-13].

2. Current State of Practice

In LSSs, the current practice of discovering performance anomalies is centered on three major approaches:

2.1. Reactive Approach

Reactive techniques are used to set thresholds for observed performance counters (e.g., CPU utilization, disk I/O, memory consumption and network traffic) and raise alarms when these thresholds are violated. In LSSs, such as data centers and cloud providers, hosting multitenant application, the workload volume can be un-predictive. Using static thresholds, may lead to false alarms, thereby wasting analyst's time. Moreover, reactive approach is inadequate for understanding the performance changes resulting from application updates.

2.2. Proactive Approach

This category includes techniques for continuous evaluation of a system behavior by comparing it against baselines or statistical models. LSSs are continuously evolving and baseline rarely exist. Furthermore, there is an overhead involved in keeping the performance models up-to-date since continuous training of models on the performance data is required to keep them abreast with the dynamic and evolving behavior of LSSs.

2.3. Rule of Thumbs

In this category, analysts use a few of the important performance counters known to them from past practice and domain gurus, among thousands that are collected, during the performance monitoring process. They usually perform manual ad-hoc checks such as conducting simple correlation tests and producing plots for visual inspections. For example, up-ward trend for the memory usage, throughout, is a good indicator of a memory leak.

3. Proposed Methodology

We believe the current practice of identifying performance anomalies in not effective since it can take hours of manual analysis and still analyst may miss performance anomalies that are not associated with 'rule of thumbs'. Towards this end, we proposed a methodology based on Shannon Entropy; which intuitively provides a measure of the uncertainty remaining in the system after an observation has been made[2]. The entropy of a continuous random variable X with probability density function p(x), is given by: