Hytrace: A Hybrid Approach to Performance Bug Diagnosis in Production Cloud Infrastructures

Ting Dai, Daniel Dean[†], Peipei Wang, Xiaohui Gu, Shan Lu[‡]
North Carolina State University, *{tdai, pwang7, xgu}@ncsu.edu*
[†]InsightFinder Inc., *daniel@insightfinder.com*
[‡]University of Chicago, *shanlu@cs.uchicago.edu*

ABSTRACT

Server applications running inside production cloud infrastructures are prone to various performance problems (e.g., software hang, performance slow down). When those problems occur, developers often have little clue to diagnose those problems. We present HyTrace, a novel *hybrid* approach to diagnosing performance problems in production cloud infrastructures. HyTrace combines rule-based static analysis and runtime inference techniques to achieve higher bug localization accuracy than pure-static and pure-dynamic approaches for performance bugs. HyTrace does not require source code and can be applied to both compiled and interpreted programs such as C/C++ and Java. We conduct experiments using real performance bugs from seven commonly used server applications. The results show that our approach can significantly improve the performance bug diagnosis accuracy compared to existing diagnosis techniques.

CCS CONCEPTS

• **Computer systems organization → Cloud computing**; • **Software and its engineering → Automated static analysis**; **Dynamic analysis**; **Software performance**;

KEYWORDS

Hybrid analysis, performance bug diagnosis

1 INTRODUCTION

Cloud computing infrastructures have become increasingly popular by allowing users to access computing resources in a cost-effective way. However, when performance problems (e.g., software hang, performance slowdown) occur in production cloud infrastructures, it is notoriously difficult to diagnose because the developers often have little diagnostic information (e.g., no error log or core dump) to localize the fault.

Previous work on performance bugs can be broadly classified into two groups: 1) *static analysis schemes* [3–6] that detect bugs by searching specific performance anti-patterns in software, such as inefficient call sequences or loop patterns; and 2) *dynamic runtime analysis schemes* [1, 2] that closely monitor runtime application behaviors to infer root causes of performance problems. In contrast, dynamic approach can target the specific problem that has occurred in the production environment. However, it needs to perform monitoring on production systems, inevitably imposing overhead. To avoid excessive runtime overhead, previous research proposed performance diagnosis based on system-level metrics or events that can be easily collected with low overhead, such as CPU utilization, free memory, system calls, and performance-counter events [1, 2]. Unfortunately, without knowledge about program semantics, those dynamic techniques suffer from both false positives and false negatives too [1, 2].

We developed HyTrace, a novel *hybrid* performance bug diagnosis scheme for production cloud infrastructures. Our technique does not require application source code and imposes little overhead, which makes it practical for the production cloud environment. Our idea is to construct a static anti-pattern detector (Hytrace-static) and a dynamic abnormal behavior detector (Hytrace-dynamic) that each individually provides *high coverage* maybe at the expense of precisions. When combining such schemes, the high coverage will naturally be retained and the lost precision fortunately can be regained as most false alarms will not be reported by both schemes that conduct diagnosis from different perspectives.

We successfully reproduced 14 out of the 133 performance bugs [1]. Our results show that HyTrace significantly improves the *accuracy*, with the true root-cause' ranking improved from top 31 to top 6 (on average) for diagnosing 13 out of 14 *reproduced* performance bugs compared to existing pure-dynamic analysis tools (PerfScope [2]). None of these bugs can be covered by traditional pure static checkers that target on general software bugs (Infer [3], Findbugs [6]) or specific types of loop inefficiency bugs (Caramel [5]). Moreover, HyTrace-static improves the *coverage* by at least 69% for diagnosing 133 performance bugs compared to Infer, Findbugs and Caramel. HyTrace is light-weight: imposing less than 3% overhead to production systems and localizing suspicious functions for complex server applications with millions lines of code within tens of minutes.

REFERENCES

[1] J. Arulraj, P. Chang, G. Jin, and S. Lu. 2013. Production-run software failure diagnosis via hardware performance counters. In *ASPLOS*.
[2] Daniel J Dean, Hiep Nguyen, Xiaohui Gu, Hui Zhang, Junghwan Rhee, Nipun Arora, and Geoff Jiang. 2014. PerfScope: Practical Online Server Performance Bug Inference in Production Cloud Computing Infrastructures. In *SOCC*.
[3] Facebook. 2017. Infer. (2017). http://fbinfer.com
[4] G. Jin, L. Song, X. Shi, J. Scherpelz, and S. Lu. 2012. Understanding and detecting real-world performance bugs. In *PLDI*.
[5] Adrian Nistor, P. Chang, Cosmin Radoi, and Shan Lu. 2015. Caramel: Detecting and Fixing Performance Problems That Have Non-intrusive Fixes. In *ICSE*.
[6] University of Maryland. 2017. Findbugs. (2017). http://findbugs.sourceforge.net

SoCC '17, September 24–27, 2017, Santa Clara, CA, USA

ACM ISBN 978-1-4503-5028-0/17/09.
https://doi.org/10.1145/3127479.3132562

[1]This benchmark suite is used by previous performance diagnosis work.