

Anomaly? Application Change? or Workload Change?

Towards Automated Detection of Application Performance Anomaly and Change*

Ludmila Cherkasova¹, Kivanc Ozonat¹, Ningfang Mi², Julie Symons¹, Evgenia Smirni²

¹ HPLabs, Palo Alto & ² College of William and Mary, Williamsburg

lucy.cherkasova@hp.com, kivanc.ozonat@hp.com, ningfang@cs.wm.edu, julie.symons@hp.com, esmirni@cs.wm.edu

Abstract

Automated tools for understanding application behavior and its changes during the application life-cycle are essential for many performance analysis and debugging tasks. Application performance issues have an immediate impact on customer experience and satisfaction. A sudden slowdown of enterprise-wide application can effect a large population of customers, lead to delayed projects and ultimately can result in company financial loss. We believe that online performance modeling should be a part of routine application monitoring. Early, informative warnings on significant changes in application performance should help service providers to timely identify and prevent performance problems and their negative impact on the service. We propose a novel framework for automated anomaly detection and application change analysis. It is based on integration of two complementary techniques: i) a regression-based transaction model that reflects a resource consumption model of the application, and ii) an application performance signature that provides a compact model of run-time behavior of the application. The proposed integrated framework provides a simple and powerful solution for anomaly detection and analysis of essential performance changes in application behavior. An additional benefit of the proposed approach is its simplicity: it is not intrusive and is based on monitoring data that is typically available in enterprise production environments.

1 Introduction

Today's IT and Services departments are faced with the difficult task of ensuring that enterprise business-critical applications are always available and provide adequate performance. As the complexity of IT systems increases, performance management becomes the largest and most difficult expense to control. We address the *problem* of efficiently diagnosing essential performance changes in application behavior in order to provide timely feedback to application designers and service providers. Typically, preliminary performance profiling of an application is done by using synthetic workloads or benchmarks which are created to reflect a "typical application behavior" for "typical client transactions". While such performance profiling can be useful at the initial stages of design and development of a future system, it may not be adequate for analysis of performance issues and observed application behavior in existing production systems. For one thing, an existing production system can experience a very different work-

load compared to the one that has been used in its testing environment. Secondly, frequent software releases and application updates make it difficult and challenging to perform a thorough and detailed performance evaluation of an updated application. When poorly performing code slips into production and an application responds slowly, the organization inevitably loses productivity and experiences increased operating costs.

Automated tools for understanding application behavior and its changes during the application life-cycle are essential for many performance analysis and debugging tasks. Yet, such tools are not readily available to application designers and service providers. The traditional *reactive* approach is to set thresholds for observed performance metrics and raise alarms when these thresholds are violated. This approach is not adequate for understanding the performance changes between application updates. Instead, a *pro-active* approach that is based on *continuous* application performance evaluation may assist enterprises in reducing loss of productivity by time-consuming diagnosis of essential performance changes in application performance.

With complexity of systems increasing and customer requirements for QoS growing, the research challenge is to design an *integrated framework of measurement and system modeling techniques* to support performance analysis of complex enterprise systems. Our goal is to design a framework that enables automated detection of application performance changes and provides useful classification of the possible root causes. There are a few causes that we aim to detect and classify:

- *Performance anomaly.* By performance anomaly we mean that the observed application behavior (e.g., current CPU utilization) can not be explained by the observed application workload (e.g., the type and volume of transactions processed by the application suggests a different level of CPU utilization). Typically, it might point to either some unrelated resource-intensive process that consumes system resources or some unexpected application behavior caused by not-fully debugged application code.
- *Application transaction performance change.* By transaction performance change we mean an essential change (increase or decrease) in transaction processing time, e.g., as a result of the latest application update. If the detected change indicates an increase of the transaction processing time then an alarm is raised to assess the amount of additional resources needed and provides the feedback to application designers on the detected change (e.g., is this change acceptable or expected?).

It is also important to distinguish between *performance anomaly* and *workload change*. A performance anomaly is indicative of abnormal situation that needs to be investigated and

*This work was completed in summer 2007 during N. Mis internship at HPLabs. E. Smirni is partially supported by NSF grants ITR-0428330 and CNS-0720699, and a gift from HPLabs.

