Analysis of Application Performance and Its Change via Representative Application Signatures*

Ningfang Mi
College of William and Mary
Williamsburg, VA 23187, USA
ningfang@cs.wm.edu

Ludmila Cherkasova, Kivanc Ozonat, Julie Symons
Hewlett-Packard Labs
Palo Alto, CA 94304, USA
{lucy.cherkasova, kivanc.ozonat, julie.symons}@hp.com

Evgenia Smirni
College of William and Mary
Williamsburg, VA 23187, USA
esmini@cs.wm.edu

***Abstract*—Application servers are a core component of a multi-tier architecture that has become the industry standard for building scalable client-server applications. A client communicates with a service deployed as a multi-tier application via request-reply transactions. A typical server reply consists of the web page dynamically generated by the application server. The application server may issue multiple database calls while preparing the reply. Understanding the cascading effects of the various tasks that are sprung by a single request-reply transaction is a challenging task. Furthermore, significantly shortened time between new software releases further exacerbates the problem of thoroughly evaluating the performance of an updated application. 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.**

In this work, we propose a new approach based on an *application signature* that enables a quick performance comparison of the new application signature against the old one, while the application continues its execution in the production environment. The application signature is built based on new concepts that are introduced here, namely the *transaction latency profiles* and *transaction signatures*. These become instrumental for creating an application signature that accurately reflects important performance characteristics. We show that such an application signature is representative and stable under different workload characteristics. We also show that application signatures are robust as they effectively capture changes in transaction times that result from software updates. Application signatures provide a simple and powerful solution that can further be used for efficient capacity planning, anomaly detection, and provisioning of multi-tier applications in rapidly evolving IT environments.

I. Introduction

Fundamental to the design of reliable enterprise applications is an understanding of the performance characteristics of the service under different workload conditions and over time. In multi-tier systems, frequent calls to application servers and databases place a heavy load on resources and may cause throughput bottlenecks and high server-side processing latency. 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".

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

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 the performance issues and the observed application behavior in existing production systems. First, an existing production system can experience a very different workload compared to the one that has been used in its testing environment. Second, 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 looses productivity and experiences increased operating costs.

Automated tools for understanding application behavior and its changes during the application development 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 avoiding loss of productivity by the timely diagnosis of essential performance changes in application performance.

Nowadays, a new generation of monitoring tools, both commercial and research prototypes, provides useful insights into transaction activity tracking and latency breakdown across different components in multi-tier systems. Some of them concentrate on measuring end-to-end latencies observed by the clients [9], [16], [5], [13], [14]. Typically, they provide a latency breakdown into network and server related portions. While these tools are useful for understanding the client network related latencies and improving overall client experience by introducing a geographically distributed solution at the network level, this approach does not offer sufficient insights in the server-side latency as it does not provide a latency breakdown into application and database related portions.

Another group of tools focuses on measuring server-side latencies [2], [12], [10], [7], [15] using different levels of transaction tracking that are useful for "drill-down" performance analysis and modeling. Unfortunately, such monitoring tools