# Using Dynatrace Monitoring Data for Generating Performance Models of Java EE Applications

Felix Willnecker
Andreas Brunnert
fortiss GmbH
Guerickestr. 25
80805 München, Germany
{willnecker,brunnert}
@fortiss.org

Wolfgang Gottesheim
Compuware Austria GmbH
Freistädter Str. 313
4040 Linz, Austria
wolfgang.gottesheim
@dynatrace.com

Helmut Krcmar
Technische Universität
München
Boltzmannstr. 3
85748 Garching, Germany
krcmar@in.tum.de

## ABSTRACT

Performance models assist capacity management and planning for large-scale enterprise applications by predicting their performance for different workloads and hardware environments. Manually creating these models often outweighs their benefits. Automatic performance model generators have been introduced to facilitate the model creation. These generators often use custom monitoring solutions to generate the required input data for the model creation. In contrast, standardized application performance management (APM) solutions are used in industry to control performance metrics for productive systems. This work presents the integration of industry standard APM solutions with a performance model generation framework. We apply the integration concepts using the APM solution Dynatrace and a performance model generation framework for Palladio Component Models (PCM).

## Categories and Subject Descriptors

C.4 [**Performance of Systems**]: measurement techniques, modeling techniques

## General Terms

Measurement, Performance, Prediction

## Keywords

Load Testing; Performance Evaluation; Application Performance Management

## 1. INTRODUCTION

Performance of large-scale enterprise applications (EA) is a critical quality requirement [3]. Application providers and data center operators tend to over-provision capacity to ensure that performance goals are met [11]. This is due to a lack of tool support for predicting the required capacity of a software system for expected workloads [5]. Performance models and corresponding model solvers or simulation engines can enhance current capacity estimations and therefore increase the utilization of hardware and reduce costs for application operations [2, 7].

The effort of manually creating such performance models often outweighs their benefits [8]. Automatic model generators have been introduced to reduce this effort [4, 2]. These approaches rely on monitoring data from running systems to extract the performance models. Such generated models can be used as input for a simulation engine or an analytical solver to predict the resource utilization, throughput and response time for different workloads and hardware environments.

Monitoring data for the generation of performance models is gathered by either custom solutions or tools from the scientific community [4, 13]. On the other hand, monitoring of large-scale EAs are state of the art technology in practice [9]. Companies use the gathered monitoring data to detect and resolve performance problems in productive environments [10]. This work presents an extension of our existing performance model generation framework to work with industry standard Application Performance Management (APM) solutions. We extend the Performance Management Work Tools (PMWT[1]) model generator to create Palladio Component Models (PCM) based on data collected by the Dynatrace[2] APM solution [1, 12, 6, 4].

## 2. AUTOMATIC PERFORMANCE MODEL GENERATION FRAMEWORK

In order to use the Dynatrace APM solution we extend the model generation framework presented in [4] and shown in figure 1. This framework uses a custom agent that collects the monitoring data from a running Java EE application. The monitoring data is then processed and aggregated either as comma-separated value (CSV) files and imported into a database or as Managed Beans (MBeans). The aggregated data is input for the model generation. The result is a performance model compliant with the PCM meta-model. The extension proposed in this work allows to use data extracted by standard monitoring frameworks exemplified by a Dynatrace agent for the purpose of performance model

[1]http://pmw.fortiss.org/
[2]http://www.dynatrace.com/



*ICPE'15,* Jan. 31–Feb. 4, 2015, Austin, TX, USA.
ACM 978-1-4503-3248-4/15/01.
http://dx.doi.org/10.1145/2668930.2688061.

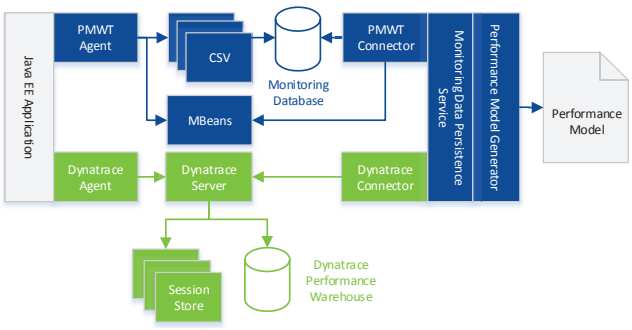


**Figure 1: PMWT Performance Model Generation Framework**

generation. This agent is attached using runtime options without changes to the instrumented application system's source code. The agent forwards collected data to the Dynatrace server, where detailed traces about method calls and error states are stored in session files for further analysis. Performance metrics derived from these traces are stored in a performance warehouse, and these metrics are typically used by operation engineers as data provider for monitoring dashboards. We extract data from both sources using an extension to our model generation framework called Dynatrace connector.

The Dynatrace connector leverages the representational state transfer (REST) interface of the Dynatrace server to extract detailed monitoring data. This REST interfaces provides, among others, call traces for instrumented operations including their resource demands. The Dynatrace connector is an extension of the monitoring data persistence service that is used by the model generator to access data from different sources. The model generator creates a performance model conforming to the PCM meta-model based on the traces and their average resource demands. The resulting models can then be used for the existing simulation engines and analytical solvers that exist for PCM models [12].

## 3. CONCLUSION & FUTURE WORK

This work proposed an integration of an industry APM solution with a performance model generation framework. Different input formats and levels of granularity can be processed. The extension shows that the generator and its interface are generally applicable and other APM solutions as generator input are possible. As the Dynatrace solution is in widespread use, the monitoring technology is tested more intensive than custom solutions and in varied operation environments. The generated model can be used to simulate different workloads and therefore enhance the Dynatrace solution with capacity planning capabilities.

As a next step we will further extend an existing prototype for the integration and evaluate it in a case study comparing the results using our PMWT agent and the Dynatrace agent. For the evaluation, we will extract models from a running SPECjEnterprise2010 instance using the two existing data collection approaches. Afterwards, the resulting models are used to predict the utilization, throughput and response time for an increased number of users. The prediction results are compared with measurement for similar workloads on the SPECjEnterprise2010 instance.

## 4. REFERENCES


[1] S. Becker, H. Koziolek, and R. Reussner. The palladio component model for model-driven performance prediction. *Journal of Systems and Software*, 82(1):3–22, 2009. Special Issue: Software Performance - Modeling and Analysis.

[2] F. Brosig, N. Huber, and S. Kounev. Modeling parameter and context dependencies in online architecture-level performance models. In *Proceedings of the 15th ACM SIGSOFT Symposium on Component Based Software Engineering*, CBSE '12, pages 3–12, New York, NY, USA, 2012. ACM.

[3] A. Brunnert, C. Vögele, A. Danciu, M. Pfaff, M. Mayer, and H. Krcmar. Performance management work. *Business & Information Systems Engineering*, 6(3):177–179, 2014.

[4] A. Brunnert, C. Vögele, and H. Krcmar. Automatic performance model generation for java enterprise edition (ee) applications. In M. Balsamo, W. Knottenbelt, and A. Marin, editors, *Computer Performance Engineering, 10th European Workshop on Performance Engineering*, volume 8168 of *Lecture Notes in Computer Science*, pages 74–88. Springer Berlin Heidelberg, 2013.

[5] A. Brunnert, K. Wischer, and H. Krcmar. Using architecture-level performance models as resource profiles for enterprise applications. In *Proceedings of the 10th International ACM Sigsoft Conference on Quality of Software Architectures*, QoSA '14, pages 53–62, New York, NY, USA, 2014. ACM.

[6] B. Greifeneder. Method and system for processing application performance data ouside of monitored applications to limit overhead caused by monitoring, June 2011. US Patent 7,957,934.

[7] L. Grinshpan. *Solving Enterprise Applications Performance Puzzles: Queuing Models to the Rescue*. John Wiley & Sons, 1st edition, 2012.

[8] S. Kounev. Performance modeling and evaluation of distributed component-based systems using queueing petri nets. *IEEE Transactions on Software Engineering*, 32(7):486–502, 2006.

[9] J. Kowall and W. Cappelli. Magic quadrant for application performance monitoring. *Gartner Research ID G*, 232180, 2012.

[10] H. Koziolek. Performance evaluation of component-based software systems: A survey. *Performance Evaluation*, 67(8):634 – 658, 2010. Special Issue on Software and Performance.

[11] M. Pawlish, A. Varde, and S. Robila. Analyzing utilization rates in data centers for optimizing energy management. In *Green Computing Conference (IGCC), 2012 International*, pages 1–6, June 2012.

[12] R. Reussner, S. Becker, E. Burger, J. Happe, M. Hauck, A. Koziolek, H. Koziolek, K. Krogmann, and M. Kuperberg. The palladio component model. *Journal of Systems and Software*, 82(1):3 – 22, 2009.

[13] A. van Hoorn, J. Waller, and W. Hasselbring. Kieker: A framework for application performance monitoring and dynamic software analysis. In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*, ICPE '12, pages 247–248, New York, NY, USA, 2012. ACM.