



<b>Title</b>	Leverage of extended information to enhance the performance of JEE systems
<b>Authors(s)</b>	Portillo Dominguez, Andres Omar, Murphy, John, O'Sullivan, P. (Patrick)
<b>Publication date</b>	2012-10-30
<b>Publication information</b>	Portillo Dominguez, Andres Omar, John Murphy, and P. (Patrick) O'Sullivan. "Leverage of Extended Information to Enhance the Performance of JEE Systems," October 30, 2012.
<b>Conference details</b>	The IT&T 11th International Conference on Information Technology and Telecommunication 2012 , Cork, Ireland, 29-30 October 2012
<b>Item record/more information</b>	<a href="http://hdl.handle.net/10197/9219">http://hdl.handle.net/10197/9219</a>

Downloaded 2026-05-01 03:57:32

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd\_oa)



© Some rights reserved. For more information

# Leverage of Extended Information to Enhance the Performance of JEE Systems

Andrés Omar Portillo Domínguez<sup>1</sup>, John Murphy<sup>1</sup>, Patrick O’Sullivan<sup>2</sup>

<sup>1</sup> Lero - The Irish Software Engineering Research Centre, Performance Engineering Laboratory, UCD School of Computer Science and Informatics, University College Dublin, Ireland  
andres.portillo-dominguez@ucdconnect.ie, j.murphy@ucd.ie

<sup>2</sup> IBM Software Group, Dublin, Ireland  
patosullivan@ie.ibm.com

## Abstract

This paper offers an overview of the performance engineering field, including some of its latest challenges. Then, it briefly describes the research area of enhancing the performance of JEE systems through leveraging its “Extended Information” and some recent investigation trends in that front. Finally some future research ideas are presented.

**Keywords:** Performance Engineering, Performance Enhancement, JEE Systems

## 1 Performance Engineering – Overview

Performance will always be a key non-functional requirement, as it plays a major role in the successful adoption of any software product. This is even more important on enterprise-level applications, where issues might not only affect the users’ experience, but also have a considerable financial impact.

The latest trends in the information technology world, like Service Oriented Architecture (SOA) and Cloud Computing, have augmented the complexity of distributed applications and brought new challenges. For example, applications in the Cloud are commonly installed in huge server farms composed by hundreds or even thousands of nodes, while previously they used to be restricted to relatively small clusters of servers. Similarly SOA has broken barriers between previously unconnected technologies and allowed their integration. Increased complexity of the applications and a steady growth in the volume of transactions have exposed new areas of potential failure points in applications further complicating testing, monitoring and performance tuning.

The rapid adoption of these new paradigms has brought up the natural need of questioning if and how the existing software engineering techniques and methodologies might be applicable to this new technological landscape, along with the identification of what adjustments and improvements would be necessary to make them work effectively [1].

## 2 Leverage of Extended Information to Enhance Performance in JEE

Considering the broad universe of variables that could impact the performance of an application (especially in complex distributed environments such as Cloud Computing), an area of interest to focus on is the identification, analysis and understanding of information that is relevant in a high-quality measurement and diagnosis of performance of those systems. Input data might come from multiple sources, some are internal to the inspected environment (i.e. Java dumps or logs from possible different nodes), while others are external (i.e. bug history during development or historical performance baseline in a production environment). This data consolidation can be seen as “Extended Information”.

In this context, JEE applications represent an attractive study field. Java is a predominant technology at the enterprise level and results in this area could potentially be applied in wide range of systems.

## 3 What has been done to study the performance of JEE systems?

Most of the recent research in this field can be categorized into the following trends:

- *Intrusive vs. Non-intrusive Monitoring:* Intrusive monitoring is now the most commonly used approach to gather the data to be analyzed (i.e. Eclipse TPTP<sup>1</sup> or IBM Tivoli<sup>2</sup>). Intrusive monitoring

---

<sup>1</sup> Eclipse Test & Performance Tools Platform, <http://www.eclipse.org/tptp>

is used to obtain as much information as possible about the observed system through instrumentation and monitoring agents. On the contrary, non-intrusive techniques look for information that can be freely obtained from the environment without the need of modifying it (i.e. Oracle. HPROF<sup>3</sup> or IBM WAIT<sup>4</sup>).

- *Hot spot vs. Idle-Time Analysis*: Strategies also differ in the type of information they use to identify performance issues. The school of thought that dominates is the study of the Hot Spots [2,3], where it is assumed that they are bottleneck candidates because the program spends a high percentage of its execution time there. On the contrary, other techniques analyze the idle-time [4] because problems often manifest as idle-time in multi-tier applications.

A common objective among different approaches is the consolidation of the available information to facilitate its interpretation and analysis which usually include diverse levels of abstraction and visual representations. Regardless of the degree of “friendliness” that those views provide, it is common that a certain level of expertise is expected from people to understand the results and perform a deeper analysis.

## 4 Future Research ideas

Future work in the field could revolve around the following points:

- To analyze the different types of information which can be obtained from a Java application environment (i.e. Javadumps or logs) and identify what knowledge can be acquired from it.
- To complement the results of point “a” by identifying what other information sources are highly desirable to improve the quality of a performance analysis (i.e. bug history in software development or monitoring reports during an application production life). To research alternatives to integrate those sources and means to exploit the knowledge that can be inferred from them.
- To investigate how the integrated results of points “a” and “b” could fit into:
  - A Software Development Life Cycle, so issues can be uncovered as early as possible.
  - Support and Maintenance processes, so that the execution of activities such as root cause analysis of issues or impact analysis of enhancements could be facilitated.
- Finally, all the previous points could be iterative and support the broader goal of **generating tools that simplify the analysis and enhancement of performance in JEE applications** by offering expert guidance about their possible causes and solutions.

## 5 Conclusion

Complexity of JEE systems keeps increasing due to the adoption of SOA paradigm and a shift towards Cloud-based deployments. This makes test and performance-related activities very challenging. “Extended Information” about application environment could be leveraged in building more intelligent tools that would facilitate the performance optimization of that type of applications.

## Acknowledgements

This work was supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre ([www.lero.ie](http://www.lero.ie))

## References

- [1] Murphy, J. (2011) *Performance Engineering for Cloud Computing*, Computer Performance Engineering, LNCS 6977 1-9
- [2] Wen, Y., Zhao, J., Chen, H., Zhang, J.(2011) *HotSpotInsight: A Java Application Introspection Platform Based on JVM*. International Conference on Instrumentation, Measurement, Computer, Communication and Control
- [3] Mytkowicz, T., Diwan, A., Hauswirth M. and Sweeney, P.(2010) *Evaluating the accuracy of Java profilers*. ACM SIGPLAN conference on Programming language design and implementation
- [4] Altman, E., Arnold, M., Fink, S and Mitchell, N. (2010) *Performance analysis of idle programs*. International Conference on Object Oriented Programming Systems, Languages and Applications

---

<sup>2</sup> IBM Tivoli Monitoring, <http://www-01.ibm.com/software/tivoli/products/monitor/>

<sup>3</sup> Oracle HPROF, <http://java.sun.com/developer/technicalArticles/Programming/HPROF.html>

<sup>4</sup> IBM WAIT, <https://wait.ibm.com/>