

# A Systematic Mapping Study on High-level Language Virtual Machines

Vinicius H. S. Durelli, Katia R. Felizardo, and Marcio E. Delamaro  
Computer Systems Department  
University of São Paulo  
São Carlos, SP, Brazil  
{durelli,katiarf,delamaro}@icmc.usp.br

## ABSTRACT

**Background:** There is a large body of literature on research in virtual machine for high-level languages, i.e., high-level language virtual machines (HLL VMs). Despite being a well-established research area, there are no studies focusing on characterizing the sorts of research that have been conducted and shedding light on most investigated subjects as well as subjects requiring further research.

**Objectives:** To conduct a systematic mapping study of the literature describing research into HLL VM.

**Research method:** We undertook a systematic mapping study of the literature based upon searching of major electronic databases.

**Results:** 128 papers have been selected and classified by their contribution, employed HLL VM implementation, type and date of publication.

**Conclusions:** The majority of the selected studies concentrate on improvements for boosting performance, introducing better garbage collection capabilities, and adapting HLL VMs or their core components to meet the requirements for embedded platforms. Furthermore, from examining the selected studies we have found that Java virtual machine (JVM) implementations are by far the most employed within academic settings. Among them, Jikes Research Virtual Machine is the most-widely used.

## Categories and Subject Descriptors

D.3.4 [Programming Languages]: Processors—*Virtual Machines, Interpreters, Run-time Environments, Intermediate Languages*

## General Terms

Experimentation, Garbage Collection, Languages

## Keywords

Systematic Mapping Study, High-level Language Virtual Machine, Evidence

## 1. INTRODUCTION

Virtual machines (hereafter abbreviated to VMs) have been designed, built, and investigated by operating system developers, language and compiler designers, and hardware developers. There is a broad array of VMs, each of them with its unique characteristics, implementation, and goals. In the context of this paper we are particularly interested in VMs geared towards supporting high-level languages, i.e., high-level language VMs (HLL VMs) [7].

HLL VMs have been playing an important role as a mechanism for implementing programming languages for more than thirty years. A great deal of the contemporary high-level languages have their execution environment based upon HLL VMs and due to the Software Engineering (SE) benefits provided by these managed execution environments, they are used on different platforms ranging from web servers to embedded systems. Despite being a well-established and researched technology, to the best of our knowledge there are no comprehensive studies focusing on an overview of this research area and its most investigated subjects. In order to fill in such a gap we have conducted a systematic mapping study (henceforth we will term this a mapping study for brevity) of existing research into HLL VMs. Systematic mapping is a methodology that involves searching the literature to ascertain the nature, extent, and quantity of published research papers (which are referred to as *primary studies*) on a particular area of interest [5]. Mapping studies aggregate and categorize primary studies, thereby yielding a synthesized view of the research area being considered.

This paper outlines the mapping study we have undertaken in order to classify and categorize current evidence on HLL VMs. The following contributions are made: we identify (i) which areas in HLL VM research have been most subjected to investigation, (ii) which areas require further research, (iii) the relevant publication forums, and (iv) which HLL VM implementations are the most widely used within the academic community. Moreover, we also present a visual summary of the results in the form of a bubble plot, i.e., the map.

We focus on describing the results of our mapping study and the essential elements of the research protocol we have devised in advance for conducting it. The remainder of this paper is organized as follows. Section 2 outlines the major elements of the devised research protocol, thereby describing how the mapping study was conducted. Section 3 presents the results of our mapping study and answers its research questions, Section 4 discusses threats to validity, and conclusions are drawn in Section 5.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

VMIL'10, 17-OCT-2010, Reno, USA

Copyright 2010 ACM 978-1-4503-0545-7/10/10 ...\$10.00.

## 2. THE MAPPING STUDY PROCESS

The process we have applied to conduct the mapping study herein described is detailed by Petersen et al. [5]. According to them, its essential steps are: (i) definition of research questions, (ii) conducting the search for relevant primary studies, (iii) *screening* of papers, (iv) *keywording* of abstracts, and (v) data extraction and mapping.

Research questions must embody the mapping study purpose. Hence, given that we set out to determine which HLL VM features are the most investigated and improved by the research community along with the most used implementations in such context, our two research questions reflect this purpose as follows.

**RQ<sub>1</sub>:** which functionalities/features/characteristics of HLL VMs have been most investigated?

**RQ<sub>2</sub>:** which are the mainstream HLL VM implementations within the academic community?

The search for primary studies involved defining both the search string and the electronic databases to be used. The string we used for searching is composed of a combination of the following keywords and acronyms: virtual machine, VM, high-level language virtual machine, and HLL VM. These search terms were chosen from a trial with a candidate set. It is worth mentioning that we selected quite general keywords in order not to narrow the mapping study scope.

The search encompassed electronic databases deemed as the most relevant scientific sources [3] and thus likely to contain important primary studies. We used the search string on the following electronic databases: ACM Digital Library<sup>1</sup>, EngineeringVillage<sup>2</sup>, IEEE Xplore<sup>3</sup>, Springer Lecture Notes in Computer Science (LNCS)<sup>4</sup>, and ScienceDirect<sup>5</sup>. During the search conduction, no limits were placed on date of publication.

Screening aims at determining which primary studies are relevant to answer our research questions, thus in this step all retrieved primary studies were evaluated. To this end, we applied a set of inclusion and exclusion criteria to each retrieved study. The inclusion criteria devised and applied are:

- if several papers reported similar studies, only the most recent was selected;
- papers describing more than one study had each study individually evaluated;
- it has to describe at least a prototypical implementation of the proposed improvement, thereby mentioning the HLL VM implementation that was modified.

and the following exclusion criteria:

- papers that do not present studies pertaining to HLL VMs, e.g., papers describing research on system VMs [7];

- studies describing the introduction of improvements that consist in solely modifying the intermediate language of the HLL VM under consideration;
- studies whose proposed enhancements do not imply in making changes to the underlying HLL VM, e.g., papers describing features implemented atop HLL VMs;
- studies whose target HLL VM is either a co-designed (e.g., composed of both software and hardware portions) or an entirely implemented in hardware HLL VM;
- technical reports, documents that are available in the form of either abstracts or presentations (i.e., elements of “grey” literature), and secondary literature reviews (i.e., systematic literature reviews [4] and mapping studies).

An initial figure of 142 candidate papers was obtained after applying the inclusion and exclusion criteria based only upon title and abstract. This initial set was reduced to 136 after filtering for duplicates. Close examination spotted two journal papers that are extended versions of earlier conference papers, thus the latter were excluded leaving a set of 134 papers. Three papers were excluded because they described studies with no running prototype of the proposed enhancement. After going over introductions and conclusions, we ended up with a final set of 128 primary studies as shown in Table 1. Nevertheless, it is important to mention that other parts, besides introductions and conclusions, quite often had to be read to find out which HLL VM implementation was used.

**Table 1: Papers retrieved from each electronic database, total of candidate studies, and the final set.**

Electronic Database	Number
ACM Digital Library	1554
EngineeringVillage	1395
IEEE Xplore	309
Springer LNCS	640
ScienceDirect	1123
<b>Total</b>	5021
<b>Candidates</b>	142
<b>Final set</b>	128

We applied a *keywording* strategy [5] aimed at devising our own classification scheme and categories for the selected primary studies. By applying such strategy, initially, abstracts are read for the purpose of finding keywords and concepts that reflect their contribution. Subsequently, these keywords and concepts are combined together to produce a general understanding regarding the nature and contribution of the research. Eventually, the final set of keywords is used to define representative categories. The classification scheme gradually evolves toward its final version as new categories are added, merged, or split up. At the end of the depicted strategy, 14 categories have been obtained.

Each included primary study was assigned to one or more categories. As for the 14 categories we have devised in the

<sup>1</sup><http://portal.acm.org/>

<sup>2</sup><http://www.engineeringvillage.org/>

<sup>3</sup><http://ieeexplore.ieee.org/>

<sup>4</sup><http://www.springer.com/lncs>

<sup>5</sup><http://www.sciencedirect.com/>

*keywording* step, they group together primary studies whose research range from improvements for boosting HLL VMs performance to enhancements geared towards introducing functionality that attenuates the run-time overhead associated with certain testing activities. Given this wide variety of categories, in order to provide an overview of the sort of research grouped in them, each one of the resulting categories is outlined as follows:

**Optimization:** this category includes primary studies concerned with optimizations that have been proposed and implemented to improve the run-time performance of HLL VMs. The proposed optimizations described by studies in this category range from improvements to interpreters and just-in-time (JIT) compilers to architectural modifications for mitigating the computational cost of certain language constructs.

**Garbage Collection (GC):** in this category are included studies that report on techniques for tracking and reclaiming memory no longer reachable from the running application.

**Debugging:** this category comprises studies focusing on introducing functionality for supporting debugging activities into HLL VMs.

**Memory Leak Tolerance (MLT):** this category includes primary studies that present prototype implementations for effectively tolerating memory leaks.

**New Language Construct (NLC):** studies grouped in this category detail modifications needed to support new language constructs.

**Profiling:** in this category are included primary studies concerned with both collecting and exploiting profiling information in the context of HLL VMs.

**Aspect-Oriented Programming (AOP):** this category contains primary studies focusing on providing support for features specific to AOP languages.

**Embedded System (ES):** this category contains primary studies aimed at adapting HLL VMs to some constraints imposed by ESs, e.g., stringent energy consumption and low memory footprint.

**Security:** this category is composed of primary studies concerned with either enhancing or introducing new security features.

**Real-Time:** real-time applications have timely requirements that depend upon the adaptability provided by a real-time HLL VM. Studies in this category fall into this context, reporting on either the design, architecture, and implementation of real-time HLL VMs or improvements to extant real-time HLL VMs.

**Distributed Computing (DC):** primary studies emphasizing support for distributed computing are grouped in this category.

**Fault Tolerance (FT):** in this category are included primary studies concerned with designing and implementing fault-tolerant HLL VMs.

**Resource Sharing among HLL VMs (RSVM):** primary studies exploring resource sharing among HLL VMs are grouped in this category.

**Testing:** primary studies belonging to this category piggyback on HLL VMs's control over execution to provide functionality that attenuates the run-time overhead associated with several testing activities.

Figure 1 shows the frequency of primary studies by category. As previously mentioned, certain primary studies were grouped in more than one category, affecting the frequency

count; the sum of the frequencies shown in Figure 1 (153) is greater than the total of selected studies presented in Table 1 (128).

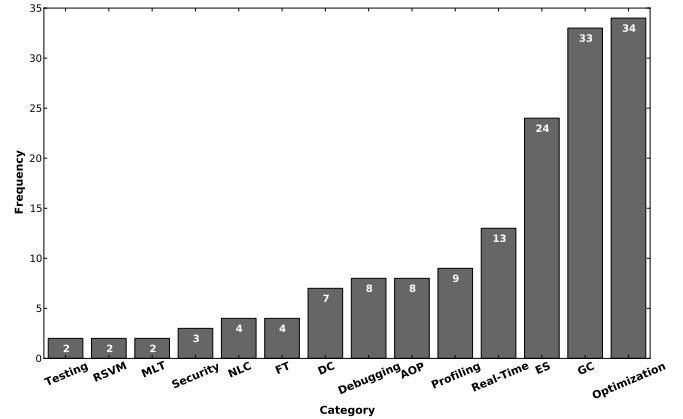


Figure 1: Frequency of studies in each category.

### 3. ANALYSIS

The focus of this section is to present the broad overview of research within HLL VM we have acquired after classifying and categorizing primary studies. We also use information drawn from this overview to answer our mapping study's research questions.

It is fairly evident from observing Figure 1 that optimization, GC, and ES are by far the categories containing more studies. Hence, the answer to  $RQ_1$  is that the functionalities most investigated are concerned with boosting performance, effectively managing memory, and adapting HLL VMs or their core components for resource-constrained environments. In contrast, the categories with less studies are security, MLT, RSVM, and testing. Thus, it is argued that optimization, GC, and ES are evidence *clusters* (i.e., where there may be scope for more complete literature reviews to be undertaken), whereas security, MLT, RSVM, and testing can be regarded as *gaps* (i.e., where new or better primary studies are required). Furthermore, in the time span of the last 11 years, great attention has been devoted to optimization, GC, and ES. As shown in Figure 2, a year-wise distribution of these categories reveals an increase in publication over time, most notably from 2003 to 2008. Our results show that there was an accentuated increase in the number of publications belonging to the ES category in 2005 and a plateau occurred from 2006 to 2008. Conversely, certain categories, e.g., MLT, RSVM, and testing, show no increase in the number of publications.

The majority of selected primary studies are published by ACM Digital Library, as shown in Figure 3. The other electronic databases, EngineeringVillage, Springer LNCS, and IEEE Xplore, had 38, 16, and 12 selected studies, respectively. As previously described in Table 1, we also searched through ScienceDirect, however, since it was the last electronic database to be examined, at that point all relevant studies it returned had already been selected from another electronic database.

As for the publication types, we have selected primary studies that have been published in conferences, journals,

symposia, books, and workshops. The majority of the primary studies are conference papers (46), and only 8 primary studies are workshop publications. Figure 4 gives further detail on the classification of primary studies according to their publication type.

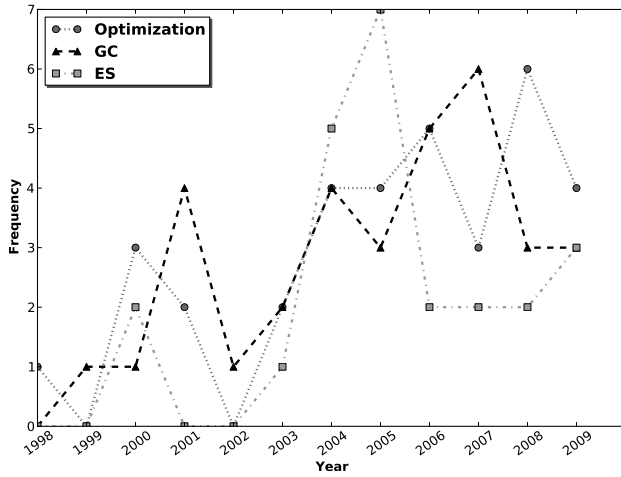
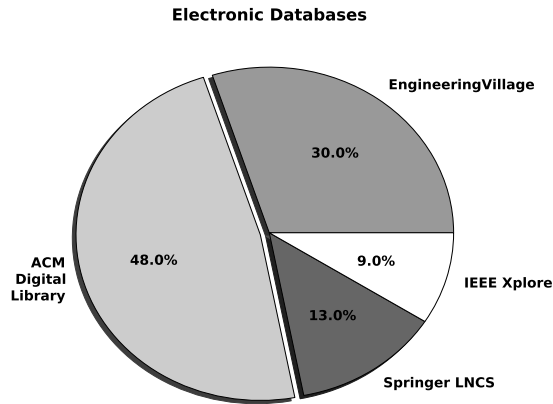


Figure 2: Year-wise distribution of publications on the most investigated categories.

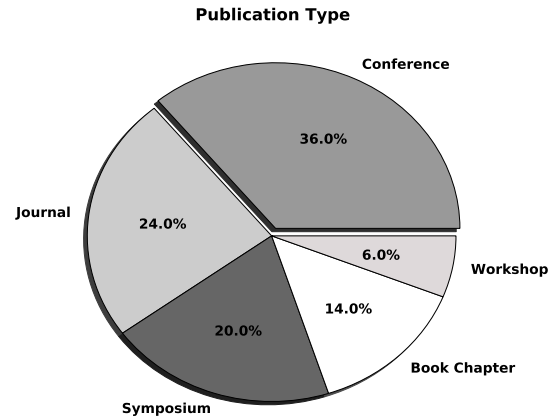


Electronic Database	Number
ACM Digital Library	62
EngineeringVillage	38
Springer LNCS	16
IEEE Xplore	12
ScienceDirect	0

Figure 3: Distribution of primary studies by electronic database.

Based upon the information we have acquired, it was also possible to identify prominent publication forums for research papers in HLL VMs. The most notable medium for reporting research results is the LNCS series, from the final set of 128 studies, 18 of them have been published in an

LNCS book (of which 16 were retrieved from Springer itself and 2 from EngineeringVillage). Another important publication forum is the Conference on Object-oriented Programming, Systems, Languages and Applications (OOPSLA)<sup>6</sup>; 15 studies have been selected from it. Three other major publication forums also worth mentioning are the International Symposium on Memory Management (ISMM), International Symposium on Principles and Practice of Programming in Java (PPPJ), and International Symposium on Code Generation and Optimization; 5 primary studies have been selected from each of them. Venues whose goal is to be a research forum specialized in HLL VMs or in the broad area of virtualization, e.g., Virtual Execution Environments (VEE), have not had many publications selected from them. For instance, only 2 of the selected primary studies have been published in VEE.



Publication Type	Number
Conference	46
Journal	31
Symposium	25
Book Chapter	18
Workshop	8

Figure 4: Distribution of primary studies by publication type.

### 3.1 Map

Instead of using frequency tables we have decided to produce a bubble plot to report the frequencies and distribution of the selected studies according to their categories and publication date, thereby providing a map of research in HLL VM. Bubble plots are essentially two x-y scatter plots with bubbles in category intersections. The size of each bubble is determined by the number of primary studies that have been classified as belonging to the pair of categories corresponding to the bubble coordinates. This visual summary provides a bird's-eye view that enables one to pinpoint which categories have been emphasized in past research along with gaps and opportunities for future research. Our resulting map is shown in Figure 5.

<sup>6</sup>As of 2010, OOPSLA is undergoing a transformation into SPLASH.

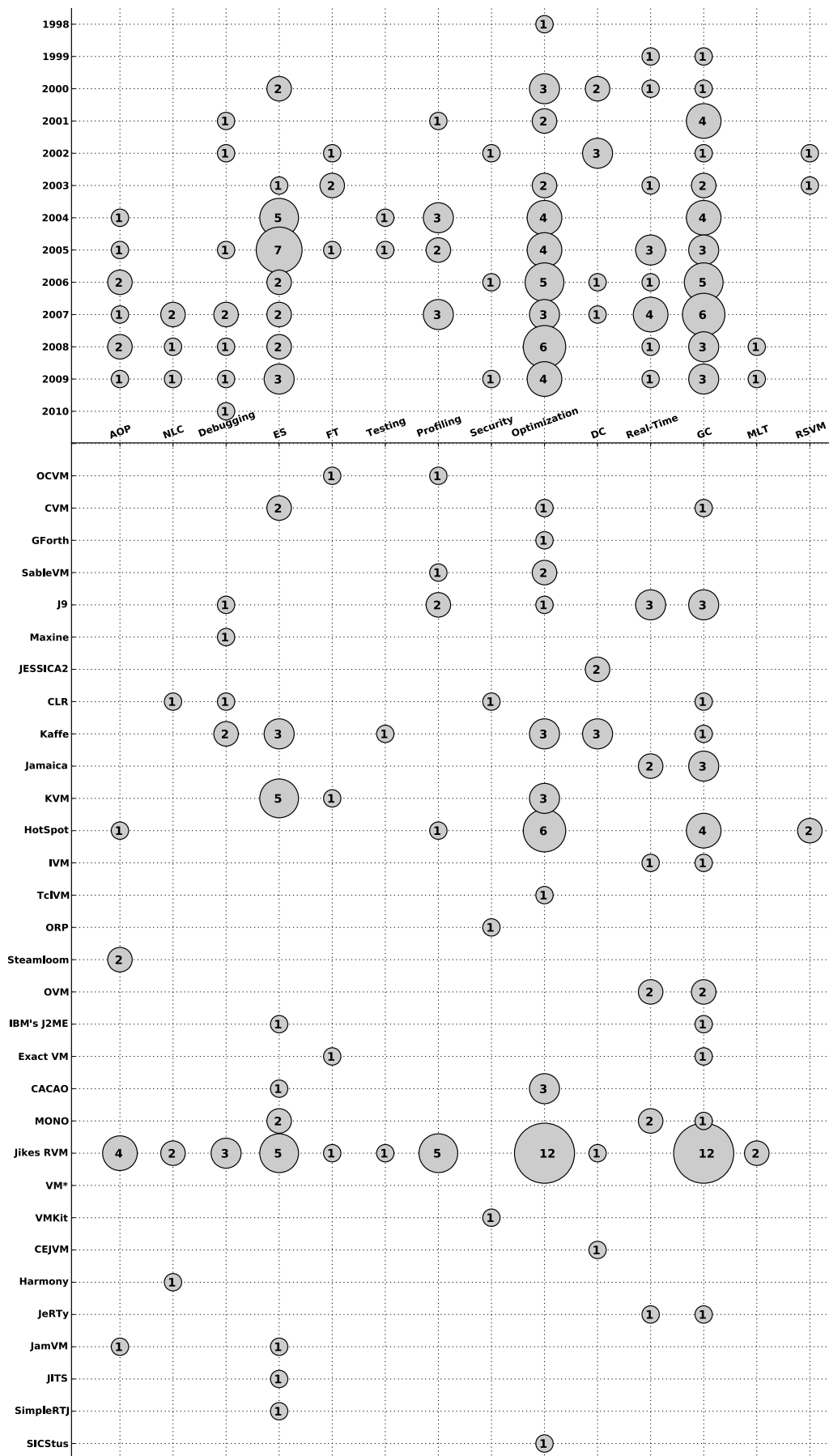


Figure 5: Map containing the distribution of HLL VM research by category, implementation employed, and year of publication.

As shown in Figure 5 the facets we have used for organizing the map are category, implementation employed, and year of publication. It is evident from observing the figure that most of the selected studies have used JVM implementations as their target platform.

More precisely, from the 31 HLL VM implementations mentioned, 24 were JVM implementations. Among them, the most-widely used JVM implementations are Jikes RVM, HotSpot, Kaffe, J9, and K Virtual Machine (KVM). Jikes RVM was used in at least one study belonging to each category, apart from security, real-time, and RSVM. In addition, most studies in the optimization and GC categories adopted Jikes RVM for implementing their improvements. Based upon this information, we argue that the answer to **RQ<sub>2</sub>** is that JVM implementations are the HLL VMs most used within academic settings. Moreover, Jikes RVM is by far the most used JVM implementation.

#### 4. THREATS TO VALIDITY

Aimed at ensuring an unbiased selection process we defined research questions in advance and devised inclusion and exclusion criteria we believe are detailed enough to provide an assessment of how the final set of primary studies was obtained. However, we cannot rule out threats from a quality assessment perspective, for we simply selected studies without assigning any scores. In addition, we wanted to be as inclusive as possible, thus no limits were placed on date of publication and we avoided imposing many restrictions on primary study selection since we wanted a broad overview of the research area.

Given that a limited set of search engines was used, it is rather possible we have missed some papers. Thus, another threat consists in whether we have properly identified and selected all relevant publications. Nevertheless, this threat was mitigated by selecting search engines which have been regarded as the most relevant scientific sources [3] and therefore prone to contain the majority of the important studies.

Whether our resulting classification scheme and categories are coherent also represents a threat to validity. As emphasized by Pretorius and Budgen [6], one of the problems of mapping studies is that the best way to classify and categorize the outcomes can only be determined when these are known. In addition, the fact that certain studies were grouped in more than one category may represent a potential threat to frequency counts and statistics in this mapping study.

#### 5. CONCLUDING REMARKS

The novelty of this paper lies in the methodology we have applied to conduct the search for studies. Only few papers documenting systematic mapping studies have hitherto been published [1, 2, 6], yet none of them cover research into HLL VMs.

Our mapping study reveals that the majority of research into HLL VMs focuses on optimizing these execution environments, improving their memory management capabilities, and tailoring them to resource-constrained settings. The results also point out that most of these studies have adopted JVM implementations as their target HLL VM. Among them, the most-widely used in these studies are Jikes RVM, HotSpot, Kaffe, J9, and KVM.

As for the publication types, the majority of the studies

are conference publications. OOPSLA is the conference with more selected studies. However, the publication forum with more selected studies is the LNCS series.

Another contribution of this paper is the map we have created. By observing such a map it is possible to ascertain the extent and form of literature related to HLL VMs, thereby identifying which categories have been emphasized in past research, gaps, and possibilities for future research. Furthermore, it provides additional insight into the frequencies of publication over time.

We have confined our analysis mainly to the extent of the evidence available, rather than the content. Thus, as a longer-term future work, we intend to carry out systematic reviews in order to pinpoint the state of evidence in the most prominent categories.

#### 6. ACKNOWLEDGMENTS

The authors would like to thank the financial support provided by FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico). We also thank Fabiano Ferrari, Paulo Nardi, and Rodrigo Araujo for their proof-reading of this paper.

#### References

- [1] W. Afzal, R. Torkar, and R. Feldt. A Systematic Review of Search-based Testing for Non-functional System Properties. *Information and Software Technology*, 51(6): 957–976, 2009.
- [2] J. Bailey, D. Budgen, M. Turner, B. Kitchenham, P. Brereton, and S. Linkman. Evidence Relating to Object-Oriented Software Design: A Survey. In *1st International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 482–484. IEEE Computer Society, 2007.
- [3] T. Dybå, T. Dingsøy, and G. K. Hanssen. Applying Systematic Reviews to Diverse Study Types: An Experience Report. In *1st International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 225–234. IEEE Computer Society, 2007.
- [4] B. Kitchenham. Procedures for Performing Systematic Reviews. *Joint Technical Report TR/SE-0401 (Keele) – 0400011T.1 (NICTA)*, Software Engineering Group - Department of Computer Science, Keele University and Empirical Software Engineering - National ICT Australia Ltd, Keele/Staffs-UK and Eversleigh-Australia, 2004.
- [5] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. Systematic Mapping Studies in Software Engineering. *12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, pages 71–80, 2008.
- [6] R. Pretorius and D. Budgen. A Mapping Study on Empirical Evidence Related to the Models and Forms used in the UML. In *2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 342–344. ACM, 2008.
- [7] J. Smith and R. Nair. The Architecture of Virtual Machines. *Computer*, 38(5):32–38, 2005.