From: **IWST 2017** iwst2017@easychair.org
Subject: IWST 2017 notification for paper 9
Date: July 23, 2017 at 6:24 AM
To: Alexandre Bergel alexandre.bergel@me.com

Dear Alexandre

Congratulation, your paper Accurate VM profiler for the Cog VM has been accepted for IWST17. Please read carefully the reviews bellow. We ask that you pay close attention to these reviews in preparing the camera-ready version of your work, in addition to carefully proof-checking it for typos and other minor problems. You will receive a subsequent email containing directions for submitting the final camera-ready copy (CRC) of your paper. We expect the camera-ready version for the 25th of August.

For your paper to be published in the IWST17 conference proceedings, at least one of the authors of the paper must register for the ESUG conference and confirm that she/he will present the paper in person.

The conference schedule will be shared after the 15th of August. The authors of a full paper will have a slot of 20 minutes for presenting there work. The authors of a paper up to 6 pages will have a slot of 15 minutes.

See you in Maribor!
Sincerely,

Anne and Jannik

----------------------- REVIEW 1 ---------------------
PAPER: 9
TITLE: Accurate VM profiler for the Cog VM
AUTHORS: Sophie Kaleba, Clement Bera, Alexandre Bergel and Stéphane Ducasse

----------- Overall evaluation -----------
With a bit of cleanup, this paper's points will be more clear and its assertions will be able to shine to their full strength. However, please note that the related work section needs a bit more Google. Some technical questions remain, such as what happens if native functions are evicted from the generated code zone while the profiler is running. The paper should at least state this problem is a current limitation, or how this situation is handled, or whether something needs to be done --- this would be an item for the future work section.

See full review in the attached PDF (please print all pages).<This review contains an attachment, see the file review_1.pdf attached to this letter.>

----------------------- REVIEW 2 ---------------------
PAPER: 9
TITLE: Accurate VM profiler for the Cog VM
AUTHORS: Sophie Kaleba, Clement Bera, Alexandre Bergel and Stéphane Ducasse

----------- Overall evaluation -----------
This interesting paper addresses the problem of code profiling in presence of the JIT-enabled VM.
In this context, the JIT produces long inlined optimized native functions that makes profiling reports less meaningful.
The paper clearly describes existing tools in the Smalltalk/Cog ecosystem such as the VMProfiler that has been ported to Pharo.
It then presents an extension of the Cog VM to generate more "accurate" profiling reports.

I suggest that the authors remove the word "accurate".
IMHO, you are not improving the accuracy but the granularity of profiling reports.

I think that the title and the beginning of the paper should be revised to better explain what you did.
I suggest to focus on this idea of *granularity* of profiling reports.
Currently VMProfiler is at the method level while you propose an approach to support sub-methods profiling reports i.e. at the bytecode level
Please consider a new title such as: A Bytecode level profiler for the Cog VM

Regarding the structure of the paper, it is not clear if Section 2 is a contribution or not when reading the first time.
Rename sec 2 and better announce it at the end of the introduction

Rename sec 2 and better announce it at the end of the introduction.

Rename sec 3. something like: Our JIT-aware VMProfiler

5.1/
But standard profilers are not JIT aware so they give a coarse grain report that is not very helpful in many situations

5.2/
Please better explain the comparison of your approach to JVM TI one.
Can JVM TI be used to generate profiling reports JITed methods and sub-methods elements?

MINOR REMARKS

"However, the VM profiler, unlike ..." -> "However, the VMProfiler, unlike ..." (use the same name as before in the text)

Enilmoparts is not the reversed word of Trampoline it should be Enilopmart

Figure 7 is highly interesting.
Please put in bold interesting percentage to help the reader i.e 57.95% 12.00%

regarding the time spent in the integer loop (bpc 76->78), it would be good to explain to which instructions this loop corresponds to in the Fibonacci benchmark.

4/
"is decomposed in several bytecode ranges (in this case 8):" -> "is decomposed in 8 bytecode ranges:"

----------------------- REVIEW 3 ---------------------
PAPER: 9
TITLE: Accurate VM profiler for the Cog VM
AUTHORS: Sophie Kaleba, Clement Bera, Alexandre Bergel and Stéphane Ducasse

----------- Overall evaluation -----------
The authors detect a problem in the current implementation of the profiling tool VMProfiler, that is, it is unable to provide accurate information on the activity of native functions, resulting from a JIT compiler operation. VMProfiler most recent implementation addresses this problem, and the way it does so is what is explained with detail in the manuscript.

The paper is well structured, the problem well motivated and explicitly stated in page 4 (How to provide accurate profiling information in large native functions profiled?) and the proposal is detailed enough to be easy to follow by an expert in VM implementation. I'm not so sure a not-so-expert would be able to follow the paper, but the topic is difficult and I can not suggest an alternative nor do it better.

I only found some errors in the spelling of some names in the references [Bn12] and [BnRR11], where there is a space + n where should be an ñ.

review_1.pdf