Laboratoire Samovar, UMR 5157
Équipe HP2, département d'informatique
Telecom SudParis

Gaël Thomas
Telecom SudParis, 9, rue Charles Fourrier, 91011 Evry
Tel : 06 10 39 31 17
Email : gael.thomas@telecom-sudparis.eu

Paris, 21 August 2017

Evaluation of the PhD thesis of Mr Clément Béra: "Sista: a Metacircular Architecture for Runtime Optimisation Persistence"

The PhD thesis of Clément Béra presents optimization techniques for managed runtimes, and especially an optimized just-in-time compiler for the Pharo runtime.

Managed runtimes are today widely used because they abstract away the code from the underlying hardware, which provides safety and portability. Technically, a managed runtime simulates an abstract machine with a virtual instruction set. At runtime, the managed runtime maps this virtual instruction set to the concrete instruction set provided by the hardware. In order to achieve performance, a managed runtime may compile on the fly and just-in-time the virtual instruction set to the concrete instruction set. As this compilation phase takes time, and as this time is taken during the run, compiling a code only becomes interesting if the code is executed often. For a cold code, i.e., a code rarely executed, the increased performance of the compiled code does not amortize the time taken to compile the code.

In his PhD thesis, Clément Béra proposes an optimizing just-in-time compiler. This compiler is based on the baseline compiler of the Pharo runtime, which quickly compiles the code, but produces an inefficient code. The compiler of Clément Béra detects the hot functions, gathers runtime information to drive the optimizations, and generate optimized versions of the hot functions. Implementing a just-in-time compiler is an impressive piece of work.

Clément Béra explores the frontiers of the just-in-time compilation techniques by addressing two original challenges. First, in order to decrease the warm-up time of an application, Clément Béra proposes a new architecture to make the optimization performed by the just-in-time compiler persistent from one run to another. This contribution is especially interesting for the smartphones, for which saving the energy consumed to optimize the code is a primary concern. In order to make the optimizations persistent, Clément Béra proposes to split the compiler in two parts. The first part optimizes the code, but, instead of generating concrete instructions as in traditional just-in-time compilers, the first part generates an optimized version of the virtual code. This optimized version can thus be reused from one run to another. The second part, which is based on the baseline Pharo compiler, does not optimize the code, but simply translates the optimized code to the concrete instruction set. Thanks to his proposal, Clément Béra is able to divide by up to 5 the execution time of applications, but without increasing the warm-up time of the applications since the optimizations remain persistent from one run to another.

For the second challenge, Clément Béra explores how the just-in-time compiler can optimize itself on the fly. For that purpose, Clément Béra proposes to write the optimizing compiler in the language executed by the Pharo runtime (Smalltalk). Then, Clément Béra proposes to use the optimizing compiler to optimize itself (metacircular architecture). Clément Béra highlights the limits of a metacircular architecture. First, Clément Béra shows that the compiler has to disable temporarily its optimization when it optimizes its own code, in order to avoid infinite optimization loops. Then, Clément Béra shows that the deoptimizer, which deoptimizes a function when a speculative invariant is broken during the run, cannot be optimized because deoptimizing the deoptimizer may lead to infinite deoptimization loop. For these two limitations, Clément Béra proposes original and interesting trade-offs. As a result, Clément Béra definitely proves that most of the compiler can be optimized by itself on the fly with a metacircular architecture.

The document, written in English, is 137 pages long and divided in 10 chapters. The document is very well written and structured. I really enjoyed reading the document.

Introduction and background. In the first chapter, Clément Béra presents the thesis and gives the context. This chapter is easy to read and clearly presents the contributions. In a second chapter, Clément Béra presents a state-of-the-art of just-in-time compilation techniques. The chapter presents the function-based and the meta-tracing architectures, along with metacircular optimizing techniques and techniques to make a managed runtime state persistent. This chapter is very clear and well documented. The chapter 3 presents the Pharo runtime and gives all the background required to understand the remainder of the document.

Contributions. The chapters 4 to 8 present the contributions and the performance evaluation of the contributions. The chapter 4 gives the overall architecture, the chapter 5 how the optimizing just-in-time compiler was implemented, the chapter 6 discusses the metacircular architecture, the chapter 7 explains how the runtime state becomes persistent and the chapter 8 presents a systematic evaluation. This set of chapters is particularly interesting and shows that Clément Béra has a very strong background in just-in-time compilation techniques. Clément Béra systematically illustrates with concrete examples his contributions, which makes the reading very clear. The content of the chapters 4 and 5 is relatively expected as they describe the just-in-time compiler, but these chapters are clearly written and show that Clément Béra made an outstanding implementation work. I really enjoyed the chapter 6, which discusses the limitations of a metacircular architecture. The problems of infinite recursions indentified by Clément Béra are very complex, and he was able to explain them very clearly. The solutions to prevent these problems are all motivated, and I don't see how we could have a better architecture. Congratulation. The chapter 7, which describes how the state becomes persistent, is clear and interesting. As I enjoy just-in-time compilation techniques, I would like to see even more implementation details (especially how the heap is saved/restored on a different architecture), but adding these details would probably make the document a little bit indigestible. Finally, the evaluation is sound and complete. Clément Béra evaluates 11 real applications and discusses all the results. I found this chapter very convincing. The evaluation shows that the optimizing just-in-time compiler largely increases the performance of the Pharo runtime and also shows that the just-in-time compiler can optimize a large set of applications.

Perspectives and conclusion. The chapter 9 presents the future works and the chapter 10 concludes the document. The future works range from engineering future works to research future works and, once again, shows that Clément Béra perfectly understand his subject.

Overall, during his thesis, Clément Béra has realized an impressive work to implement a new just-in-time compiler for the Pharo runtime. The PhD thesis presents the architecture of a metacircular just-in-time compiler and explores the limits of the approach. The PhD thesis also presents new optimization techniques, which make the optimizations persistent from one run to another. These two contributions go largely beyond the state-of-the-art and clearly show that Clément Béra is an excellent researcher in just-in-time compilation techniques. The document is easy to read and shows that Clément Béra has a perfect mastery of his research subject. I'm also very impressed by the large amount of work performed by Clément Béra to evaluate his contributions. Implementing an optimizing just-in-time compiler is very complex and error prone, and the difficulty is still increased for a metacircular just-in-time compiler. The just-in-time compiler will be integrated in the production Pharo runtime, which indubitably highlights the quality of the implementation. Clément Béra has also published his work in several international venues, especially in ISMM, which is considered as one of the top-level conference in memory management techniques.

For these reasons, I recommend that Clément Béra be awarded the PhD degree of Lille. And I can only congratulate Clément Béra for his outstanding work.

Gaël Thomas
Professeur, Telecom SudParis