

# Optimisation Optimiste des Langages de Programmation Dynamiques

Maxime Chevalier-Boisvert

Mars 2010

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Langages statiques et dynamiques . . . . .	2
1.2	Le langage JavaScript . . . . .	2
1.3	Interprètes et Machines Virtuelles . . . . .	2
1.4	Optimisation des langages dynamiques . . . . .	2
<b>2</b>	<b>État de l'art</b>	<b>3</b>
2.1	Le langage SELF . . . . .	3
2.2	L'univers de Java . . . . .	3
2.3	Techniques d'analyse et d'optimisation . . . . .	3
2.4	Systèmes métacirculaires . . . . .	3
2.5	JavaScript et navigateurs web . . . . .	4
<b>3</b>	<b>Objectifs visés</b>	<b>5</b>
3.1	Optimisation optimiste . . . . .	5
3.2	Implantation métacirculaire . . . . .	5
3.3	Tachyon comme plateforme de recherche . . . . .	5
3.4	Perspectives à long terme . . . . .	5
<b>4</b>	<b>Conclusion</b>	<b>6</b>
4.1	État actuel du projet . . . . .	6
4.2	Échéancier . . . . .	6
4.3	Langages de programmation du futur . . . . .	6

# Chapitre 1

## Introduction

### 1.1 Langages statiques et dynamiques

Différences sémantiques.

### 1.2 Le langage JavaScript

Pas Java!

Particularités du langage. Dynamique, fonctionnel, impératif, orienté-objet.

### 1.3 Interprètes et Machines Virtuelles

### 1.4 Optimisation des langages dynamiques

Évaluation partielle, extraction d'information. Analyse et profilage. Défis techniques.

## Chapitre 2

# État de l'art

### 2.1 Le langage SELF

SELF[1]

### 2.2 L'univers de Java

Beaucoup de recherche centrée autour de Java.

HotpathVM[2]

Java HotSpot[3]

### 2.3 Techniques d'analyse et d'optimisation

Procedure cloning[4]

ML run-time code generation[5]

Psyco JIT specialization[6]

IBM's adaptive optimization survey[7]

RATA[8]

### 2.4 Systèmes métacirculaires

Klein[9]

Jalapeno[10], now JikesRVM

Maxine[11]

PyPy[12]

PyPy Tracing JIT[13]

## 2.5 JavaScript et navigateurs web

JSMeter[14]

TraceMonkey[15]

Concurrent tracing JIT[16]

Tracing for the Web 3.0[17]

Mention Google V8 and explain some of its main design aspects, even though there are no publications from the Google V8 team.

## Chapitre 3

# Objectifs visés

### 3.1 Optimisation optimiste

Techniques d'optimisation innovative.

### 3.2 Implantation métacirculaire

### 3.3 Tachyon comme plateforme de recherche

### 3.4 Perspectives à long terme

## Chapitre 4

# Conclusion

4.1 État actuel du projet

4.2 Échéancier

4.3 Langages de programmation du futur

# Bibliographie

- [1] C. Chambers, D. Ungar, and E. Lee. An efficient implementation of SELF a dynamically-typed object- oriented language based on prototypes. *ACM SIGPLAN Notices*, 24(10) :49–70, 1989.
- [2] A. Gal, C.W. Probst, and M. Franz. HotpathVM : an effective JIT compiler for resource-constrained devices. In *Proceedings of the 2nd international conference on Virtual execution environments*, pages 144–153. ACM, 2006.
- [3] T. Kotzmann, C. Wimmer, H. M  
"ossenb  
"ock, T. Rodriguez, K. Russell, and D. Cox. Design of the Java HotSpot™ client compiler for Java 6. *ACM Transactions on Architecture and Code Optimization (TACO)*, 5(1) :1–32, 2008.
- [4] K.D. Cooper, M.W. Hall, and K. Kennedy. A methodology for procedure cloning. *Computer Languages*, pages 146–160, 1989.
- [5] P. Lee and M. Leone. Optimizing ML with run-time code generation. In *Proceedings of the ACM SIGPLAN 1996 conference on Programming language design and implementation*, pages 137–148. ACM, 1996.
- [6] A. Rigo. Representation-based just-in-time specialization and the psyco prototype for python. In *Proceedings of the 2004 ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation*, pages 15–26. ACM, 2004.
- [7] M. Arnold, S.J. Fink, D. Grove, M. Hind, and P.F. Sweeney. A survey of adaptive optimization in virtual machines. *Proceedings of the IEEE*, 93(2) :449–466, 2005.
- [8] F. Logozzo and H. Venter. RATA : Rapid Atomic Type Analysis by Abstract Interpretation–Application to JavaScript Optimization. In *Compiler Construction*, pages 66–83. Springer, 2010.
- [9] D. Ungar, A. Spitz, and A. Ausch. Constructing a metacircular virtual machine in an exploratory programming environment. In *Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 11–20. ACM, 2005.
- [10] B. Alpern, C.R. Attanasio, J.J. Barton, M.G. Burke, P. Cheng, J.D. Choi, A. Cocchi, S.J. Fink, D. Grove, M. Hind, et al. The Jalapeno virtual machine. *IBM Systems Journal*, 39(1) :211–238, 2010.



- [11] M. Bebenita, M. Chang, G. Wagner, A. Gal, C. Wimmer, and M. Franz. Trace-based compilation in execution environments without interpreters. In *Proceedings of the 8th International Conference on the Principles and Practice of Programming in Java*, pages 59–68. ACM, 2010.
- [12] A. Rigo and S. Pedroni. PyPy’s approach to virtual machine construction. In *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*, pages 944–953. ACM, 2006.
- [13] C.F. Bolz, A. Cuni, M. Fijalkowski, and A. Rigo. Tracing the meta-level : PyPy’s tracing JIT compiler. In *Proceedings of the 4th workshop on the Implementation, Compilation, Optimization of Object-Oriented Languages and Programming Systems*, pages 18–25. ACM, 2009.
- [14] P. Ratanaworabhan, B. Livshits, and B.G. Zorn. JSMeter : Comparing the behavior of JavaScript benchmarks with real Web applications. In *Proceedings of the 2010 USENIX conference on Web application development*, page 3. USENIX Association, 2010.
- [15] A. Gal, B. Eich, M. Shaver, D. Anderson, D. Mandelin, M.R. Haghighat, B. Kaplan, G. Hoare, B. Zbarsky, J. Orendorff, et al. Trace-based just-in-time type specialization for dynamic languages. In *Proceedings of the 2009 ACM SIGPLAN conference on Programming language design and implementation*, pages 465–478. ACM, 2009.
- [16] J. Ha, M.R. Haghighat, S. Cong, and K.S. McKinley. A concurrent trace-based just-in-time compiler for single-threaded JavaScript. *PESPMA 2009*, page 47, 2009.
- [17] M. Chang, E. Smith, R. Reitmaier, M. Bebenita, A. Gal, C. Wimmer, B. Eich, and M. Franz. Tracing for web 3.0 : trace compilation for the next generation web applications. In *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, pages 71–80. ACM, 2009.