Proceedings of the Second workshop on Virtual Machines and Intermediate Languages for emerging modularization mechanisms (VMIL 2008), held at the Twenty-third ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, October 21-23, Nashville, Tennessee, United States

VMIL '08

Workshop Chair:
Hridesh Rajan

The Association for Computing Machinery
1515 Broadway
New York New York 10036

ACM ISBN: 978-1-60558-384-6

Workshop Organization

Program Committee:

- Christoph Bockisch, Darmstadt University of Technology
- Eric Bodden (McGill University, Canada)
- Juan Chen (Microsoft Research, USA)
- Shigeru Chiba (Tokyo Institute of Technology, Japan)
- Sophia Drossopoulou (Imperial College, UK)
- Robert Dyer (Iowa State University, USA)
- Eric Eide (University of Utah, USA)
- Matthew Flatt (University of Utah, USA)
- Michael Haupt (Hasso Plattner Institute, University of Potsdam, Germany)
- Gregor Kiczales (University of British Columbia, Canada)
- Hidehiko Masuhara (University of Tokyo, Japan)
- Greg Morrisett (Harvard University, USA)
- Angela Nicoara (ETH Zurich, Switzerland)
- Harold Ossher (IBM Research, USA)
- Hridesh Rajan, Iowa State University
- Don Syme (Microsoft Research, UK)

Organization Committee:

- Hridesh Rajan, Iowa State University
- Christoph Bockisch, Darmstadt University of Technology
- Michael Haupt (Hasso Plattner Institute, University of Potsdam, Germany)
- Robert Dyer (Iowa State University, USA)

Invited Talks

Memory Management for Hard Real-time Systems

Jan Vitek
Purdue University

Abstract:
The Java programming language has become a viable platform for real-time systems with applications in avionics, shipboard computing, audio processing, industrial control and the financial sector. High performance real-time Java virtual machines (RT JVMs) are now available from multiple vendors.

One of the main challenges in using a high level programming language, such as Java or C#, to program hard-real time system is to deal with heap-allocated data structures. Traditional techniques such as pre-allocation and object pooling are ill-suited to modern software engineering practices. In this talk I describe two approaches that we have experimented with in the context of the Ovm real-time Java virtual machine: region-based allocation and real-time garbage collection. I will demonstrate that for tasks which can tolerate latencies on the order of 1 millisecond real-time collectors are perfectly adequate, but, in order to obtain sub-milliseconds latencies other approaches are required. The talk will also give an overview of new results in non-blocking concurrent garbage collection.

About the Speaker:
Jan Vitek is an Associate Professor in Computer Science at Purdue University. He leads the Secure Software Systems lab. He obtained his PhD from the University of Geneva in 1999, and a MSc from the University of Victoria in 1995. His research interests include programming language, virtual machines, mobile code, software engineering and information security.

Compiling the Web--Building a Just-in-Time Compiler for JavaScript

Andreas Gal
Mozilla Corporation

Abstract:

Over the last decade we have made great strides towards improving the execution performance of virtual-machine (VM) based high level programming languages. Today, dynamic compilation is standard in most Java and C# VMs, enabling programs written in these languages to execute with similar efficiency as legacy type-unsafe C or assembly code.

However, for the past decade much of the research and development effort in the Just-in-Time compilation domain was focused on the runtime compilation of statically typed languages, leaving an important and steadily growing field of programming languages behind: dynamically typed high level languages such as JavaScript, Python, PHP and Ruby. Combined with an explosive growth of web applications and the wide-spread use of dynamically typed programming languages for the client and server side of such web applications, this has created a situation where bytecode interpretation is suddenly again the predominant mode of execution for much of the code used on a daily basis on desktop computers, including popular web programs and services like Google Mail or Google Docs.

In this talk I will report on the design and development of TraceMonkey, the JavaScript Just-in-Time compiler in Mozilla's Firefox web browser. I will discuss the fundamental differences between statically typed and dynamically typed languages from the perspective of a compiler constructor, and I will highlight some of the unique challenges for compiling dynamically typed languages such as JavaScript.

About the Speaker:

Andreas Gal is a Project Scientist at the Computer Science Department of the University of California, Irvine. He is currently on leave, working with Mozilla on TraceMonkey, a Just-in-Time compiler for JavaScript. Andreas received his PhD from the University of California, Irvine, in 2006. His research interests include virtual machines, dynamic compilation, programming languages and mobile code.

Liquid Metal: Blurring the Hardware/Software Boundary

David F. Bacon
IBM Research

Abstract:
The goal of the Liquid Metal project is to allow a heterogeneous system of conventional processors and reconfigurable hardware (FPGAs) to be programmed in a single language with transparent, dynamic execution across the aggregate computing resources -- to "JIT the hardware". Achieving this goal requires significant innovation across the entire system: language design, compiler technology, hardware synthesis, the run-time system, and hardware protocols. I will give an overview of the Liquid Metal language and the tool chain we have built, present some initial results, and describe challenges for the future.

About the Speaker:
David F. Bacon is a Research Staff Member at IBM's T.J. Watson Research Center. He leads the Metronome project which pioneered hard real-time garbage collection, opening the use of high-level languages like Java for time-critical systems in financial trading, aerospace, defense, video gaming, and telecommunications.

Dr. Bacon's algorithms are included in most compilers and run-time systems for modern object-oriented languages, and his work on Thin Locks was selected as one of the most influential contributions in the 20 years of the Programming Language Design and Implementation (PLDI) conference. His recent work focuses on high-level real-time programming, embedded systems, programming language design, and reconfigurable hardware. He received his Ph.D. in computer science from the University of California, Berkeley and his A.B. from Columbia University. He holds 7 patents and has served on numerous program committees including POPL, OOPSLA, ECOOP, LCTES, and EMSOFT. He is a member of the IBM Academy of Technology, Distinguished Scientist of the ACM, and is on the governing boards of ACM SIGPLAN and SIGBED.

Table of Contents