

Vladimir Kostyukov

Compiler Engineer

+7(923)1319766

✉ vladimir.v.kostyukov@gmail.com

<http://vkostyukov.ru>

Summary

Software Engineer with 3 years of experience developing compilers and VMs. Creative and passionate programmer with in-depth knowledges of managed runtimes and compilers. Excellent Java and C/C++ programming skills allied with experience in performance analysis and benchmarking.

Skills

Programming Languages	Fluent in <i>Java</i> and <i>Scala</i>
	Efficient in <i>C/C++</i>
	Knowledges in <i>JavaScript</i> , <i>Python</i> , <i>Perl</i>
Scopes	Programmig Languages Design and Implementation, Performance Analysis, Object Oriented Programming, Funtional Programming, Purely Functional Data Structures, Design and Analysis of Algorithms

Experience

2011/04 – ...	<p>Software Engineer, Intel Corporation, Novosibirsk, Russia.</p> <p>I am a member of the Managed Runtimes team. My first project there was development (from scratch) a proof of concept PKCS#11 Java Crypto Provider (5k LOC), which based on Intel IPP libraries. The developed prototype showed 6X speedup relative to the default Java implementation.</p> <p>I am currently involved into development of the x86 Trace-JIT compiler (targeted to Intel® Atom™ Architecture) for Dalvik VM. More precisely, I am responsible for development of both back-end (including code geneartion and instruction scheduling) and middle-end (including data-flow analysis and optimizations on CFG) components of the compiler.</p> <p>Tools: <i>Linux Shell</i>, <i>Git</i>, <i>Gerrit</i>, <i>Bugzilla</i>, <i>GCC</i>, <i>Intel VTune</i>, <i>Intel TBB</i>, <i>Intel IPP</i></p> <p>Keywords: <i>Performance Analysis</i>, <i>Compilers and Interpreters</i>, <i>JIT Compilation</i>, <i>Low-Level and Hi-Level Optimizations</i>, <i>Data-Flow Analysis</i>, <i>CFG Construction and Analysis</i>, <i>Instruction Scheduling</i>, <i>Registarization</i>, <i>Benchmarking</i>, <i>Debugging</i></p>
2010/10 – 2011/04	<p>Software Intern, Intel Corporation, Novosibirsk, Russia.</p> <p>As a member of Compilers and Languages group I was responsible for performance tracking and analysis of Intel Compiler for MIC platform (a GPUGP chip with up to 128 cores). I gained an Intel SSG Award for being a pioneer of Intel MIC compiler performance tracking (developed a Perl-based harness and ported initial four workloads from NVidia CUDA SDK).</p> <p>Tools: <i>Intel Compiler Collections</i>, <i>Perl</i>, <i>Intel VTune</i>, <i>Linux Shell</i></p> <p>Keywords: <i>Pefrormance Analysis</i>, <i>GPU Offload</i>, <i>Benchmarking</i>, <i>Multithreading and Conrurency</i>, <i>Synchronization</i>, <i>Parallel Algorithms</i>, <i>Lock-Free Alogrithms</i></p>

- 2010/07 – 2010/08 **Summer School Intern**, [Intel Corporation](#), Novosibirsk, Russia.
I was working in **Java Xeon** team on the analysis of bottlenecks in the SPECjvm2008.serial test at Intel's modern architectures. The suggested solution (based on reducing number of stack frames) showed up to 50% speedup on WSM-EX platform (in a multithreaded mode).
Tools: *Linux Shell, JDK, GCC, Intel VTune, Vim, SPECjvm2008, Eclipse*
Keywords: *Java Performance Analysis, Benchmarking, Concurrency, JMM, HotSpot Internals, Serialization*
- 2007/10 – 2010/07 **Technician**, [Altai State Technical University](#), Barnaul, Russia.
While working in IT department, I was responsible for maintaining network environment of university campus. I also was leading a technical support team of ACM ICPC NEERC.
Tools: *Linux Shell, Clonezilla*
Keywords: *Scripting, OS Cloning, Network Administration*

Projects

- la4j **Linear Algebra for Java**, <http://la4j.org>.
The la4j is a lightweight and 100% Java library that provides Linear Algebra primitives and algorithms. It is highly popular sparse/dense matrix library, which combines both fluent API and good performance.
Tools: *Java SE, Eclipse, Maven, JUnit, Git, Travis-CI*
Keywords: *Linear Algebra, Math, API Design, TDD, Design Patterns, Open Source*
- Quipu **Quipu Programming Language**, <http://esolangs.org/wiki/Quipu>.
The Quipu is an Esoteric programming language inspired by «talking knots» – recording devices historically used by Incas. It is a «believed Turing-complete» language, which means author believes that the language is Turing-complete, but no formal proof was provided.
Tools: *Scala*
Keywords: *Source Code Parsing, Interpretation*

Education

- 2006 – 2011 **Master of Science** in CS, [Altai State Technical University](#), Barnaul.
Master's thesis: *"Distributed monitoring and dispatching system of the processes in heterogenous environment"*.
Grade: 95/100

Certificates

- 2010 **IPPP-2-12**, Intel Parallel Programming Professional.
2010 **HPC School 2010**, Participant Certificate.
2010 **Intel Summer School 2010**, Participant Certificate.
2010 **Intel Winter School**, Participant Certificate.