ALEKSANDR MARAMZIN

maramzinalexandr@gmail.com

EDUCATION

Moscow Institute of Physics and Technology (MIPT) *2010-2014 (BS), 2014-2016 (MS)*
Bachelor and Master of Applied Mathematics and Physics
Department of Radio Engineering and Cybernetics
Overall GPA *is 4.48/5.00 (BS), 4.52/5.00 (MS)*
Bachelor's Thesis*: "Investigation of negative side effects on the memory subsystem of standard x86 HW data prefetchers in the environment of experimental highly parallel microprocessor architecture".*
Master's Thesis*: "Adaptation of standard x86 microprocessor performance estimation methodology (by the means of software simulation) to architecture with binary translation".*

The University of Edinburgh *2017-2018 (MSc by Research), 2018-Present*
MSc by Research *(With Merit)*
Center for Doctoral Training (CDT) in Pervasive Parallelism
School of Informatics
Edinburgh, United Kingdom

ABOUT ME

- SHORT A technical specialist with a background and a high-level understanding of CPU architecture, compilation technologies, platforms (peripheral devices, drivers, DMA controllers, PCI, etc.). I have applied and understand the basics of machine learning. I have a fundamental education in math and physics obtained in a top-tier Russian university with a solid computer science component. I can dive deeper into my areas of specialization with a minimum required learning. I am eager to extend my expertise to other technical areas as well. My main programming language is C/C++. I use Python and Bash for auxiliary tasks. I work in Unix-like environment (Bash, Make, CMake, Vim). A systematic and self-motivated person comfortable with working independently as well as a team player.

TECHNICAL SKILLS

Programming Languages	**C/C++** (main), Python (auxiliary), Java (used once for a class project)
Assembly Languages	x86, LLVM IR
Machine Learning	Scikit-Learn (used for one project)
Programming Environment	Vim, CMake, GNU Make, Unix-like, Bash Shell
Software & Tools	Git, SVN, Latex

Code Samples
${GITHUB}="https://github.com/av-maramzin"
${C_CXX}={ "${GITHUB}/PParMetrics/src", "${GITHUB}/PParMetrics/include"}
${C_CXX}+="${GITHUB}/DCP/abstract"
${PYTHON}={ "${GITHUB}/icc-opt-report-compiler", "${GITHUB}/PParMetrics/scripts/ml"}

EXPERIENCE

Intel R&D Moscow *June 26, 2012 - September 15, 2016*
Undergraduate/Graduate Software Engineering Intern

- JOINT UNIVERSITY PROGRAM 4 years on a joint university program, equivalent to 3 full-time years of on-site project internship work. Used the results of my work to defend my Bachelor's and Master's theses at my university.

- Projects and Work 2 research projects on HW/SW co-design with binary translation (BT) on board. Worked in 2 cycle-accurate simulators (C/C++) of CPU microarchitecture. One is that of an experimental highly parallel architecture and the other is the main simulator of conventional x86 core used at Intel internally. Worked in the memory subsystem model (Load/Store Buffers, x86 cache hierarchy, x86 hardware prefetchers). Conducted a research on negative side effects (cache pollution, DCU Fill Buffer overflow with useless prefetches, etc.) of x86 hardware prefetching in the experimental architecture. In the x86 core simulator Aleksandr implemented required HW support for aggressive compiler optimizations and some microarchitectural refinements (zero-idiom (*xor eax, eax*) HW elimination). Aleksandr enabled all external event (interrupts, DMA, I/O) traces to run in accordance with the standard performance modelling methodology accepted at Intel in the context of BT system (x86 precise state maintenance, rollbacks, etc.) by implementing required support on the microarchitectural warm up stage.

Auriga PLC, Moscow *December 5, 2016 - July 14, 2017*
Software Engineer

- Role Contractor software engineer on a project of a world's major microprocessor production company. Worked in the platform part of a full platform functional simulator.
- Technologies C/C++, Device Modelling Language (DML), Python.
- Tasks. **Thunderbolt DMA controller model** (1500 LOC) Developed a software model of Thunderbolt DMA controller (according to the technical specification device of memory mapped registers and their read/write functionality). Controller maintains a ring buffer with its sender/receiver ifaces for packets transmission. Developed a device driver (using Python) to unit-test the model. **BMC IERR pre-reset error collection** (1500 LOC) Developed information gathering functionality on the receipt of a reset signal event. Linked-list of enumerated PCI devices and dumped x86 MSRs state are saved inside Baseboard Management Controller for a post-reset crash investigation. **Bugfixing and support**

The University of Edinburgh *September 11, 2017 - Present*
Postgraduate Research Student

- Group Center for Doctoral Training (CDT) in Pervasive Parallelism. Institute for Computing Systems Architecture. School of Informatics.
- Compiling Techniques Coursework. (5000 Java LOC) Developed a basic (non-optimizing) compiler. The latter compiles a subset of the C language to MIPS architecture and runs it on SPIM MIPS simulator. Developed the whole front-end (lexical, syntax and semantic analysis) as well as the back-end (code generation for MIPS).
- ML-based Parallelization Assistant (5000 C/C++, 4000 Python LOC) Application of Machine Learning (ML) in compilers literature review. Collected static code features for loops of SNU NAS benchmark suite using LLVM Pass Framework. Parsed Intel C/C++ compiler parallelization reports in order to label loops. Trained ML-based model of loop parallelizability using scikit-learn. Published AI-SEPS 2019 paper.
- Data-Centric Parallelization Automatic data structure recognition literature review. Feasibility study for SPEC CPU2006 and Olden benchmarks. Olden benchmarks parallelization with algorithmic skeletons.
- Publication. AI-SEPS 2019: Proceedings of the 6th ACM SIGPLAN International Workshop on AI-Inspired and Empirical Methods for Software Engineering on Parallel Computing Systems October 2019 Pages 1–10 https://doi.org/10.1145/3358500.3361567