Ramkumar Ramachandra

> *Budding Mathematician, Expert Engineer \ +1 (347) 586 4137 \ artagnon@gmail.com*

Work Experience

Compiler Engineer — Falcon Computing, Santa Clara [February — May 2018]
Designed a compiler for C++ programs to get them to run efficiently on FPGAs, centred around optimizing loops.

- Studied the architecture of FPGAs, and the requirements for lowering programs into RTL logic, bitstreams, and finally circuitry.
- Authored vális, an LLVM backend to generate C99 code from scratch without using the common backend infrastructure, relying heavily on C++17 compile-time expressions.

R&D Engineer — Onai, San José [August 2017 — January 2018]
Did research on Blockchain technology for sharing compute power.

- Set up Buildbot for experiments performed in Docker containers, hosted on AWS, for reliably reproducible builds — cobbled together a consistent interface using a Perl script.
- Solved a discrete optimization problem using a MiniZinc solver; a 50-node instance took 0.08s to solve.
- Designed a consensus protocol around the π-calculus described in the `Tendermint` paper; achieved fast and painless consensus across multiple nodes, using a Cloud Haskell program.
- Formalized various transactional invariants in the Blockchain, and proved their correctness using Coq.

Compiler Engineer — The MathWorks, Natick [February 2015 — July 2017]
Part of the core compiler team, managed by Dale Martin.

- Fixed the longest outstanding bugs in the organization by combing through the x86 assembly and LLVM IR with a fine-tooth comb.
- Independently authored and shipped a compiler transform from scratch, Loop Invariant Code Motion, relying heavily on Alias Analysis, Pst, and Dominator; guarded it with over 300 tests.
- Studied the Program Structure Tree and O(1)-query Dominator data structures present in the codebase, and invented incremental-update algorithms.
- Profiled and identified bottlenecks using Instruments.app; LICM hardly showed up in the charts due to its heavy reliance on the incremental update algorithms.
- Studied Sven Verdooleage's Integer Set Library, and used it to lay the groundwork for loop transform unification: Polyhedral Loop Optimizations.
- Worked on unifying the information from Alias Analysis and analysis of globals with UseDef, the uses and definitions of variables.

Education

Columbia University in the City of New York [January — December 2014]
Earned a Masters degree in Computer Engineering. Collaborated on projects for Compilers and Operating Systems — in the former, wrote my first compiler from scratch in OCaml for a toy language with simple syntax, named `rhine-ml`; in the latter, wrote a scheduler for the Linux kernel, which was among the projects that didn't crash for the entire duration of a stress-test on an Android tablet.

Indian Institute of Technology, Kharagpur [2007 — 2012]
Earned a Masters degree in Physics mentored by Sayan Kar, and sharpened skill in experimentation and programming. One of the significant projects was `phoenixfs`, a filesystem that would seamlessly version files, inspired by the way Git represents its data. Churned out code all four Summers — in the first year, it was a Django-based startup; in the second, it was an Emacs Lisp program; in the third, it was an internship with the IT-Universitetet i København while working on a GSoC in parallel; in the fourth, it was an internship with Amazon while working on the second GSoC in parallel.

Open Source Work

Contributor, Git and Subversion — Kharagpur, København, Chennai [2010 ~ 2015]
Completed two Summer of Code projects with Git over the Summers of 2010 and 2011, and continued on as an active contributor to the mailing list. Developed high coding discipline, principally working with Junio C Hamano, Jeff King, and Johnathan Nieder; over five years, touched nearly the entire codebase, totalling 196 commits. In the former SoC, worked with Bert Huijben of the Subversion community to author `svnrdump`, the first tool to import/export history from a remote svn server, and with David Michel Barr to develop the `vcs-svn` interop in git.

In the latter SoC, authored `sequencer.c` to persist data and continue operations after user-interruptions resulting from merge conflicts.

```
$ git cherry-pick --continue
$ git cherry-pick --abort
```

Contributor, LLVM Community — Brooklyn [January ~ February 2015]
Studied the infrastructure; contributed over 20 patches around the experimental garbage collection infrastructure, with Philip Reames reviewing the patches.

Contributor, Linux Kernel — Kharagpur, California, New York, Atlanta [2013 ~ 2014]
Studied filesystems, schedulers, and perf tools; contributed over 50 patches, principally improving the perf tools.

```
$ perf stat -e cycles dd if=/dev/zero of=/dev/null count=1000000
```