Contact

public@benj.me

www.linkedin.com/in/bnjbvr (LinkedIn)

Top Skills

Rust

C++

JavaScript

Languages

Français (Native or Bilingual)

English (Full Professional)

Publications

The state of SIMD.js performance in Firefox

Numerical methods for differential equations (French)

Efficient float32 arithmetic in JavaScript

Regular technical blog posts on benj.me

Benjamin Bouvier

Compiler Engineer at Embark Studios

Lyon, Auvergne-Rhône-Alpes, France

Summary

I am interested in designing and implementing efficient and fast software with large user scales and real-world usage. I have a lot of experience with WebAssembly and compiler development, having extensively participated to the implementation of WebAssembly (since it was designed) in Mozilla Firefox. While having a lot of experience with C++/Python/JavaScript, I do prefer to use the Rust programming language, the best combination of productive, safe and fast, which I've largely used for the last few years.

Bonus points if your company promotes decentralization, protecting the privacy of users in general, democracy, and/or fighting global climate change.

Experience

Embark Studios

Compiler Engineer

March 2021 - Present (2 years)

Worked on a new internal game engine that uses WebAssembly mods.

- Designed and implemented new core capabilities in the game engine, enabling the rest of our internal game dev team.
- Contributed upstream to many open-source projects that we use. Notably, extended the Wasmtime virtual machine to support Apple Silicon. I've also worked on compile-time improvements, including an incremental compilation cache that lowered cold compile times up to 20% for a large single module, and for which cache hits would reduce compile times for a module by up to a factor of 10.
- Implemented in-game admin features and majorly contributed to a full redesign of our multiplayer architecture.
- Maintained and implemented various tools to increase team productivity: Github bot, Rust analysis tool, proc-macros for WebAssembly bindings generation, misc command-line and script helpers.

Mozilla

7 years 1 month

Senior Compiler Engineer

September 2017 - January 2021 (3 years 5 months)

Cranelift, a new low-level machine code generator (Rust)

- Implemented and maintained the Cranelift wasm backend in Spidermonkey (Firefox's JavaScript engine) for years (glue code, compile-time improvements, features impl, etc.).
- Ported Cranelift's meta DSL from Python to Rust, simplifying it, eliminating classes of potential footguns.
- Implemented a fast linear scan register allocator in Rust, for usage in Cranelift.
- Participated to the implementation of the new backend targetting Aarch64. Wrote the new backend variant for x64, including enough features to run the whole WebAssembly MVP test suite.
- Set up fuzzing for our new register allocators, using a toy language (parser, execution semantics) and libfuzzer.

Firefox (C++)

- Implemented support for WebAssembly in Firefox: decoding, translating to internal intermediate representations, adding support for new features, generating machine code (x64/x86/arm32/arm64).
- Made WebAssembly function compilation much faster in Firefox by using parallelization; then even reduced further compilation overhead by batching work according to heuristically determined bundle sizes.
- Made WebAssembly to regular JavaScript calls up to 50% faster by removing implementation details in the VM. Then made calls from JavaScript to WebAssembly blazingly fast (as fast as calls from JS to JS).
- Implemented a domain specific WebAssembly fuzzer, trying to cause runtime assertions or crashes in debug builds, as well as comparing results of a same program across different platforms or architectures.
- Added Rust build support to the Spidermonkey build system, with all the possible configurations on all the different OSes supported by Firefox.

Misc:

- Was the principal maintainer of AreWeFastYet, a cluster of machines running performance benchmarks of web browsers, as well as a website showing results in a meaningful way.
- Gave several talks about WebAssembly and its future, presenting the project to Web developers.

Compiler Engineer
January 2014 - September 2017 (3 years 9 months)

- Added Float32 support to asm.js
- Added SIMD.js support to asm.js
- Participated to the implementation of WebAssembly in Firefox.

Mozilla
Platform Engineer Intern
April 2013 - September 2013 (6 months)
Mountain View, CA

- Platform Engineer Intern, in the JavaScript Engine team.
- Added a new type to both the Just In Time (general JavaScript, "IonMonkey") and the Ahead Of Time (specialized for asm.js, "OdinMonkey") JavaScript compilers. Changed all architectural levels, from intermediate representation to code generation. Designed an algorithm in SSA form that ensures whether we can run optimizations regarding this new type.
- Benchmarked these changes and obtained up to 50% speedup on some real world applications (C++, ARM, x86 and x64 assembly).
- Added support for the Linux tool "perf" to OdinMonkey.
- Various bugs fixing (crashes, misbehaviours).

Thales Research & Technology
Research Engineer
May 2012 - August 2012 (4 months)
Palaiseau, France

- Designed and implemented generic, flexible parallelization tools for an opensource framework of metaheuristics called Evolving Objects, using C++ and Message Passing Interface (OpenMPI).
- Implemented generic, configurable, ready to use, convenient tools for parallelizing the evaluation step of genetic algorithms and for starting multiple times the same algorithm with different initial parameters.
- Used these tools on an opensource temporal constraint planner called Descarwin DAE (Divide And Evolve), ran the software on a heterogeneous cluster of 250 cores and performed statistical analysis of performances so as to prove the efficiency of parallelization.

Atos Worldline

Software Engineer

May 2011 - August 2011 (4 months)

Lyon Area, France

- Designed and implemented the frontend part of a secured webmail for a CRM, using Java / J2EE, Google Web Toolkit (GWT) and Spring Security for the server side, HTML / CSS / GWT for the client side. This webmail, optimized so as to be fast, was generic enough to be ready to use for any new customer.
- Designed and implemented tests for the server side thanks to JUnit.
- Added basic roles management system to another secured webmail, using the same technologies.

Education

INSA Lyon

Master of Engineering, Computer Science · (2010 - 2013)

New Jersey Institute of Technology

Master's degree, Computer Science · (2012 - 2013)