Name: Md Shafiuzzaman

Perm number: 6846802

Github repo: https://github.com/Shafiuzzaman-Hira/WebAssembly.git

WebAssembly is a low-level, statically typed language that can be run in all major browsers. It is introduced in the developers' community as an assembly-like language for the web that does not require garbage collection and supports interoperability with JavaScript [1]. However, the specification of WebAssembly claims that it does not make any Web-specific assumptions or provide Web-specific features. Rather, it is designed as a compilation target for other languages [2]. Additionally, the low-level control over the memory layout and close mapping to hardware instructions provide a near-native performance [3]. Though, it was initially targeted for client-side computations in browsers, WebAssembly's simplicity and generality has sparked interest to use it as a platform for many other domains, e.g., on the server side in conjunction with Node.js, for "serverless" cloud computing, Internet of Things and embedded devices, or even as a standalone runtime [4]. However, the key points behind this popularity are its feature as a multi-language platform and aim to execute at native speed. The vision of this project is to investigate how WebAssembly works as a multi-language platform and how close it belongs to native codes in terms of performance. Particularly, I am interested in exploring the language features such as typing of instructions, function calls, module-based encapsulation and abstractions that supports a particular language (e.g. C, C++, java or C#) as well as comparison of its performance while running through WebAssembly vs running as a native code.

Reference:

[1] Jangda, Abhinav, Bobby Powers, Emery D Berger, and Arjun Guha. "Not So Fast: Analyzing the Performance of WebAssembly vs. Native Code," n.d., 15.

[2] Group, WebAssembly Community, and Andreas Rossberg. "WebAssembly Specification," n.d., 185.

[3] Lehmann, Daniel, and Michael Pradel. "Wasabi: A Framework for Dynamically Analyzing WebAssembly." In Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, 1045–58. Providence RI USA: ACM, 2019. https://doi.org/10.1145/3297858.3304068.

[4] Lehmann, Daniel, Johannes Kinder, and Michael Pradel. "Everything Old Is New Again: Binary Security of WebAssembly," n.d., 19.