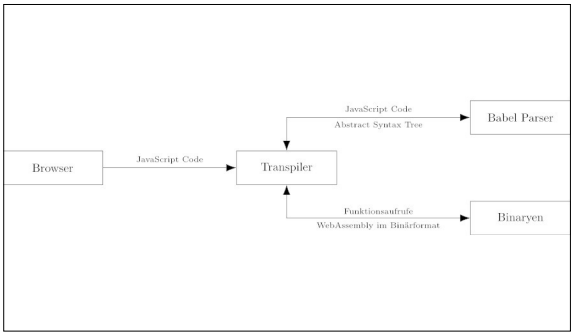
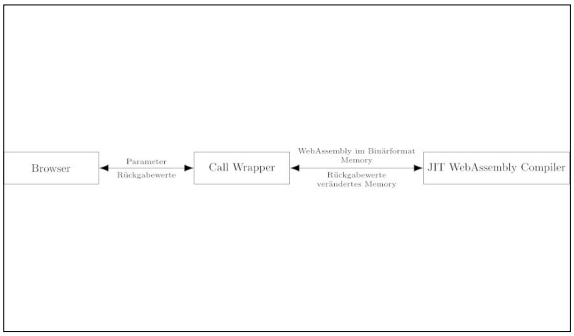
Mike
Marti

Studenten	Mike Marti, Matteo Kamm
Examinator	Prof. Dr. Luc Bläser
Themengebiet	Software

JavaScript to WebAssembly Cross Compiler



Übersicht Kompilationszeit



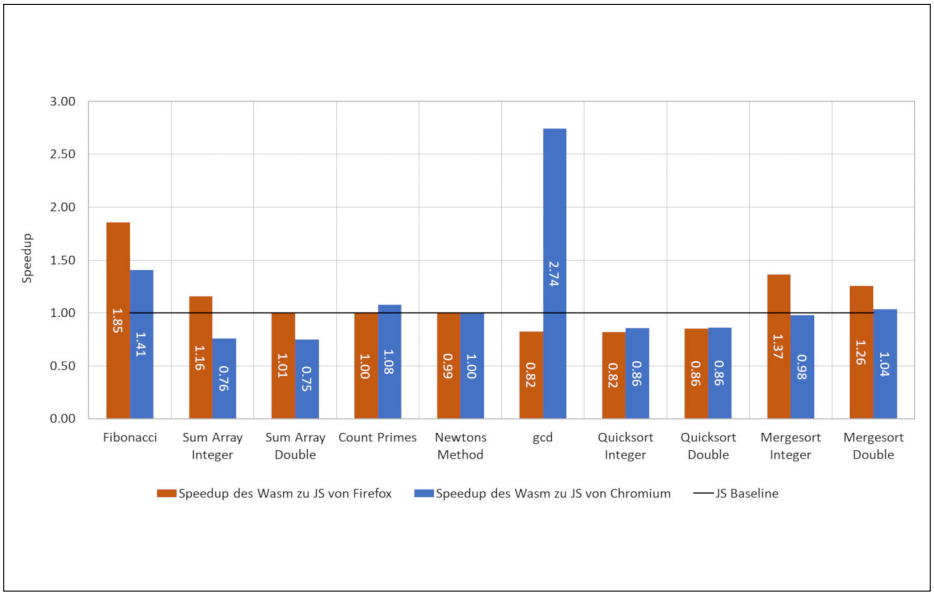
Übersicht Laufzeit

Ausgangslage: Moderne Webbrowser unterstützen die Ausführung des binären WebAssembly Formats, das eine hohe Performance sowie geringere Fluktuationen der Programmlaufzeiten zwischen den Browsern ermöglicht.

Verschiedene Compiler erlauben bereits heute das Übersetzen der Programmiersprachen C, C++ sowie Rust zu WebAssembly. Da diese Sprachen nicht im Browser ausgeführt werden können, sind solche Compiler nur ahead-of-time nutzbar. Für eine just-in-time Kompilation im Browser müsste der Compiler in JavaScript implementiert werden. JavaScript Code könnte dann durch eine Übersetzung direkt beim Client im Browser beschleunigt werden.

Vorgehen / Technologien: Damit die Ausführung ohne weitere Tools im Browser lauffähig ist, wurde der Transpiler in TypeScript geschrieben. Der Transpiler kann über eine dokumentierte Schnittstelle angesprochen werden. Übersetzt wird ein stark typisiertes und für rechenintensive Aufgaben ausgelegtes Subset von JavaScript. Anschliessend wird das WebAssembly über ein Call-Wrapper aufgerufen. Für das Erstellen des abstrakten Syntaxbaums wurde BabelJS verwendet. Das Generieren des binären WebAssemblies wird von Binaryen übernommen.

Ergebnis: Das Resultat ist ein schlanker und performanter Transpiler, der JavaScript zu WebAssembly übersetzt. Der Endnutzer benötigt für die Ausführung des Transpilers lediglich einen Browser und keine weiteren Programme. Die Transpilierung zu WebAssembly konnte trotz zusätzlicher Transpilationszeit in einigen Fällen eine schnellere Ausführung erzielen. Die Arbeit hat gezeigt, dass die WebAssembly Infrastruktur noch Verbesserungspotential aufweist. Es könnte beispielsweise noch mehr optimiert und dadurch die Laufzeit verbessert werden.



Speedup von WebAssembly zu JavaScript