WebAssembly Compilation Pipeline

PLCT Intern Tech Report

Jiang Yuchen

2021.1.27

Contents

A. General Introduction
   1. WASM Compilation Pipeline in V8
   2. Liftoff and TurboFan Introduction
   3. Comments

B. Detailed Liftoff Implementation
   1. Liftoff by Part
   2. The Output Form of Liftoff
   3. The Platform-determined Implementations
   4. Work to be done

Background

- V8 is Google’s open source high-performance JavaScript and WebAssembly engine, written in C++.
- WebAssembly is a binary instruction format for a portable stack-based virtual machine, speeding up the web service with AoT compilation.
- There are three ways to run wasm bytecode:
  - Interpreter
  - Liftoff way
  - TurboFan way

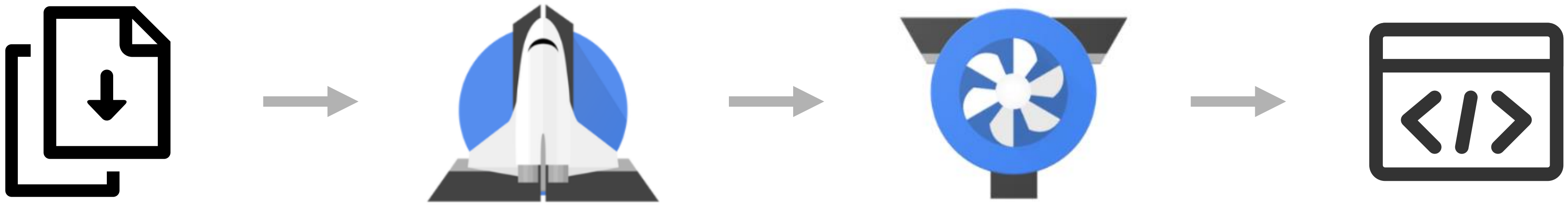
Reference: https://v8.dev/docs/wasm-compilation-pipeline

Background

- V8 is Google’s open source high-performance JavaScript and WebAssembly engine, written in C++.
- WebAssembly is a binary instruction format for a portable stack-based virtual machine, speeding up the web service with AoT compilation.
- There are three ways to run wasm bytecode:
  - Interpreter
  - Liftoff way
  - TurboFan way
- V8 uses the latter two.

Reference: https://v8.dev/docs/wasm-compilation-pipeline

Wasm Compilation Pipeline in V8



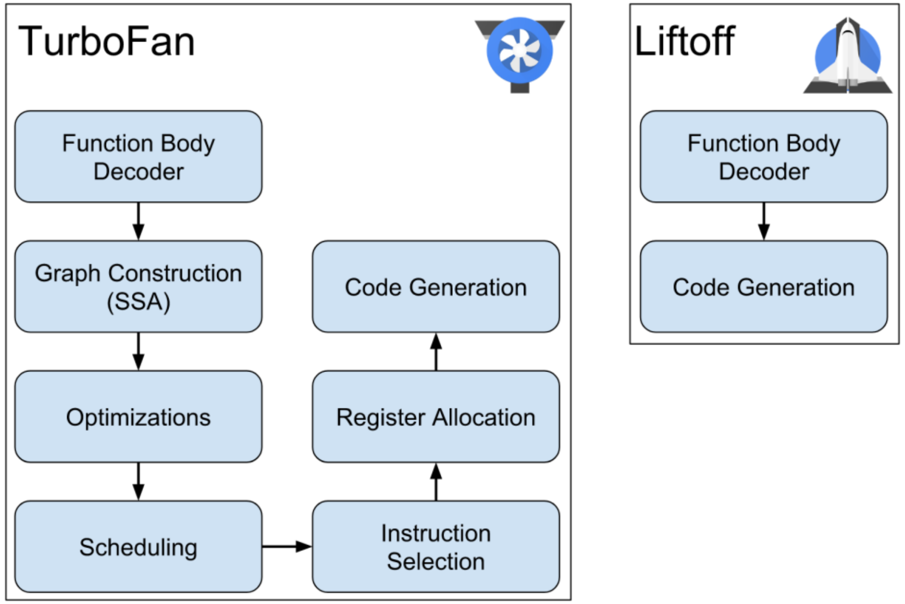
Wasm Bytecode

Liftoff Baseline

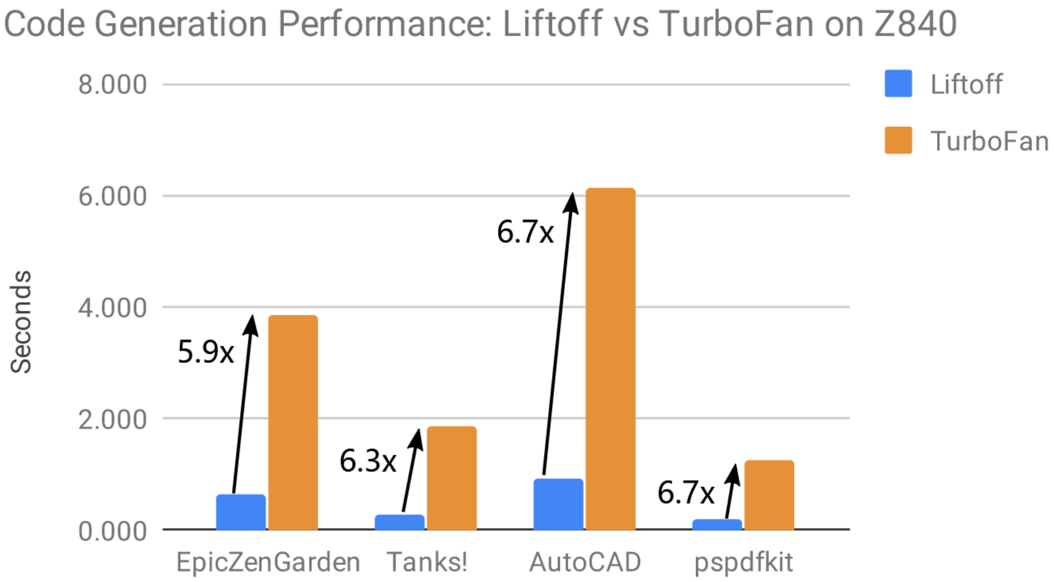
TurboFan Optimization

Local Binary

Why Liftoff?



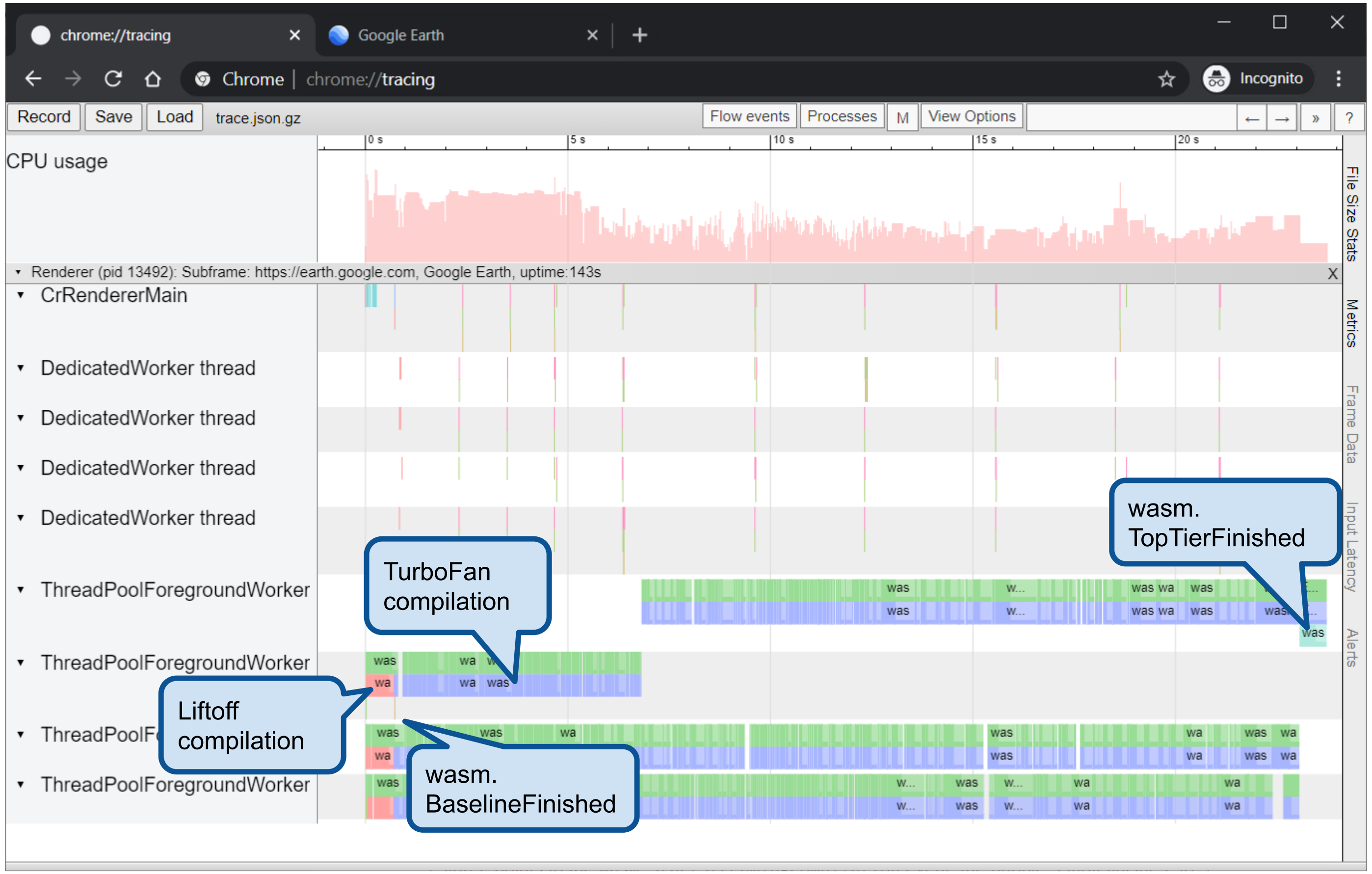
The Liftoff compilation pipeline is much simpler compared to the TurboFan compilation pipeline.



Code generation performance of Liftoff vs. TurboFan on a Z840
Conclusion: baseline & fast codegen.

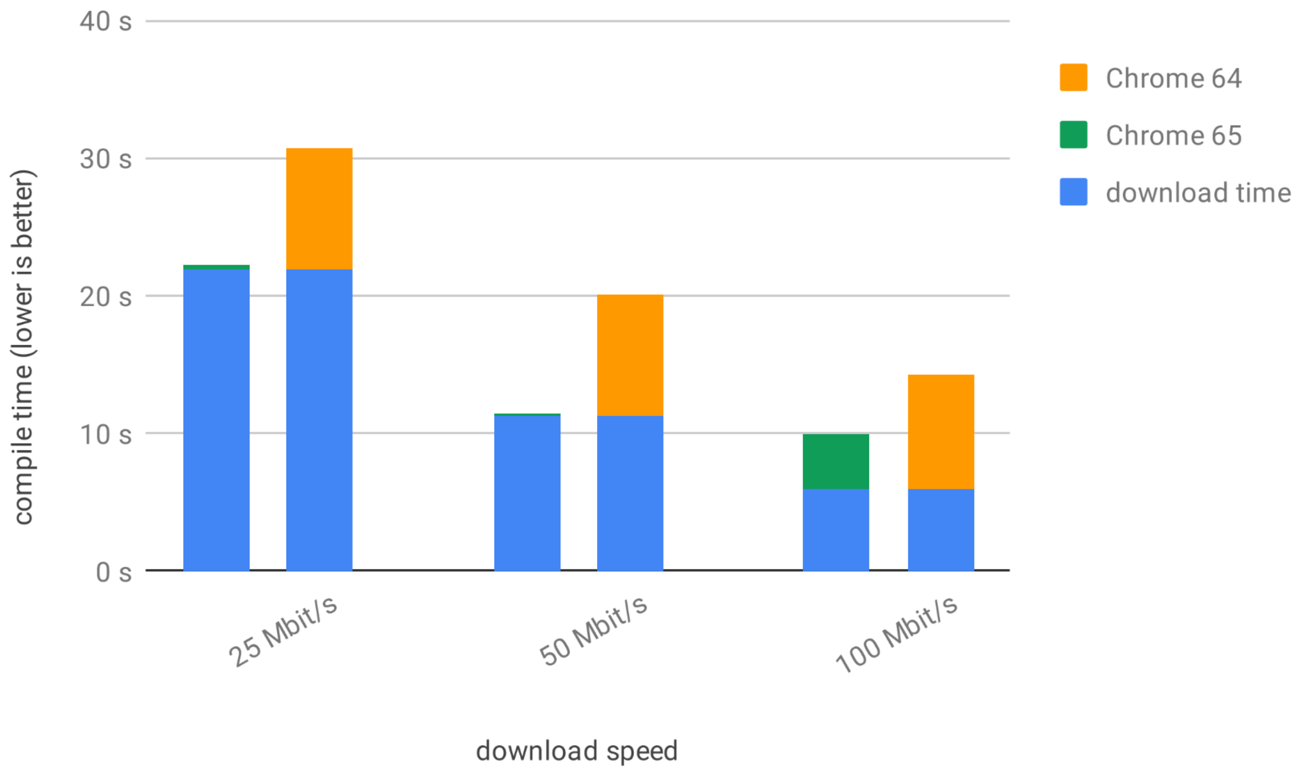
Reference: https://v8.dev/blog/liftoff

Single Pass V.S. Multi Pass



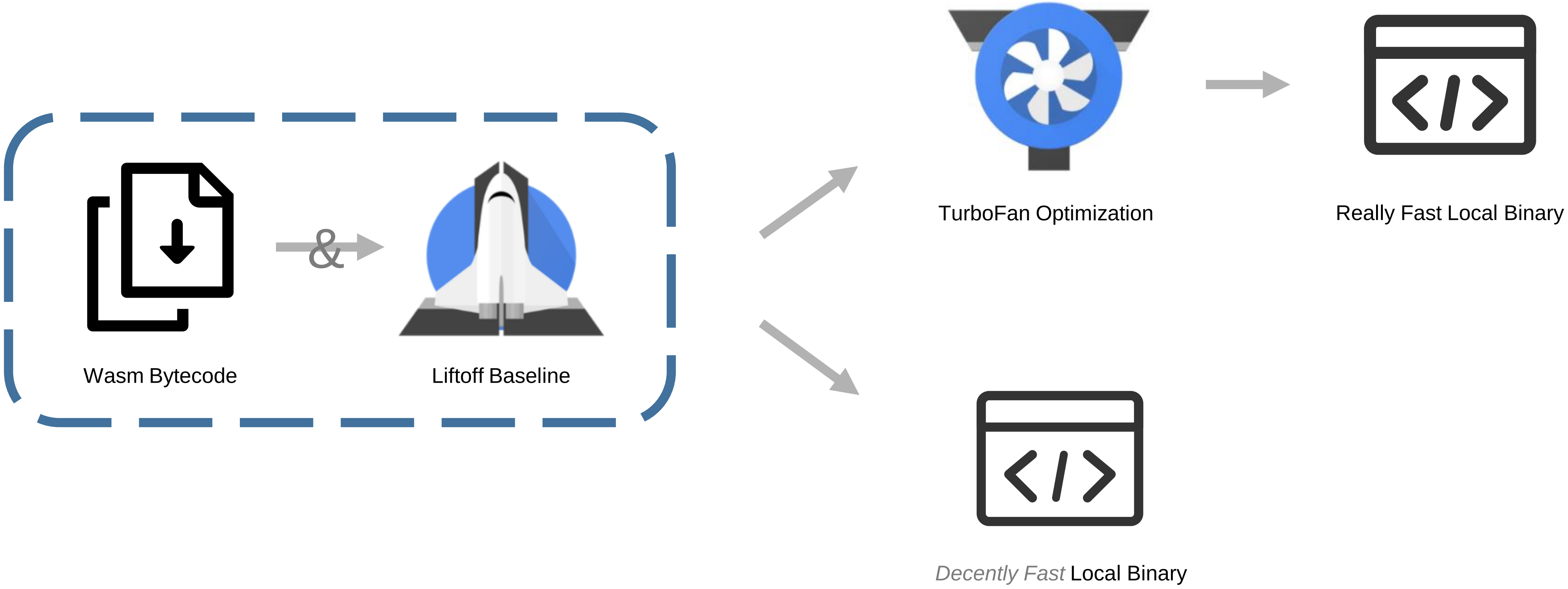
Reference: https://v8.dev/docs/wasm-compilation-pipeline

WebAssembly Streaming APIs



Reference: https://v8.dev/blog/v8-release-65

Wasm Compilation Pipeline in V8 - True



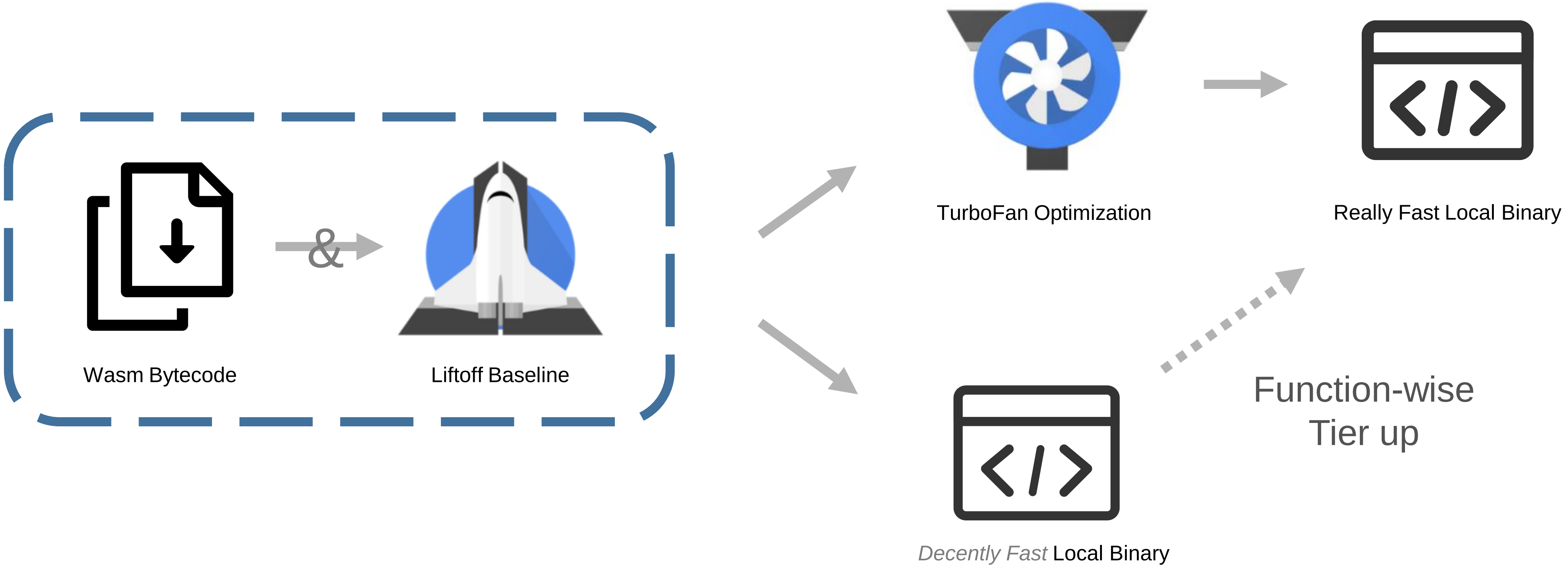
“Tier up”

- Switch to TurboFan with optimizations:
    - Register allocation improvement
    - Redundant load elimination
    - Strength reduction
    - Function inlining
- “As soon as Liftoff compilation is finished, V8 immediately starts to "tier up" the module by recompiling all functions with TurboFan.”

“Tier up”

- Switch to TurboFan with optimizations:
  - Register allocation improvement
  - Redundant load elimination
  - Strength reduction
  - Function inlining
- “As soon as Liftoff compilation is finished, V8 immediately starts to "tier up" the module by recompiling all functions with TurboFan.”
- Nevertheless, tier up does not do on-stack-replacement;
  As a result the codes are substituted function by function.

Wasm Compilation Pipeline in V8 - True x2



Observing Source Code

- The following contents are contained in the markdown file, since the slides are not appropriate for showing too much code.
- The contents include:
  1. Details of WebAssembly Format Representations
  2. Liftoff Compilation Implementation

Questions?

Thank you.