Executing WebAssembly in Teaclve

June 24, 2021

Hongbo Chen
ya0guang@protonmail.com

Why WebAssembly?

- Compatible
- Secure
- Fast Enough

WebAssembly Micro Runtime (WAMR)

- https://github.com/bytecodealliance/wasm-micro-runtime
- Very small runtime for WASM
- Almost self-contained implementation
- Embeddable!
- Customizable!

How to Support WASM?

- Modify WAMR
  - Add a new platform (`teaclave-sgx`) in WAMR
  - Add platform-specific implementations (e.g. `malloc`)
  - Compile and archive to a static library
- Embed WAMR into Teaclave
  - New document: Adding Executors
  - Initialize WAMR
  - Register Teaclave native functions (e.g. protected files)
  - Instantiate VM instance
  - Prepare arguments and memory
  - Execute the WASM function payload
  - Clean up the environment

How to Use?

- Compile the source code to WASM payload
  - My choice: `clang` shipped with wasi-sdk
  - Compilation options:
  - `-nostdlib \`
  - `-Wl,--export-all \`
  - `-Wl,--no-entry \`
  - `-Wl,--allow-undefined \`
- Using a Python script to upload the payload and execute the function

Source Code

- Modified WAMR part
  - Platform-specific implementations
- Teaclave part
  - WAMR executor in Teaclave (My fork)
  - WASM sample C code
  - Teaclave PF C header
  - Python script

Limitation & TODOs

- No common library support (e.g. `stdlib`)
- AoT compilation
- More examples & documents about compiling source code of various language to WASM

Thanks!