Wasm-R3

Jakob Getz
University of Stuttgart / KAIST

Vision

Automated creation of WebAssembly benchmarks via **Record and Replay**

Why

- Useful for performance mesurement, evaluation of static analyses, ...
- So far not many executable benchmarks out there

How

Wasm-R3 Workflow:

```
Url    ---->    Wasm-R3    ---->    Browser    ---->    Packaged Benchmark
```

WebAssembly

`index.wasm`:

```
(module
    (import "env" "foo" (func $foo))
    (func (export "main")
        i32.const 69
        i32.const 420
        i32.store
        call $foo
        i32.const 69
        i32.load    ;; ??
    )
    (memory (export "mem") 1)
)
```

`host.js`:

```
const binary = readFile('index.wasm')
const imports = {
    env: {
        foo: () => { /*...*/ }
    }
}

const wasm =
    await WebAssembly.instantiate(
        binary, imports
    )
wasm.instance.exports.foo()
```

Record and Replay

- The execution of software is determined by Environment `E` and Application `A`
- `A` is deterministic but it uses interaces of `E`
  - `A` may **call** a function provided by `E` (eg. syscall)
  - `A` may get **called** by `E` (eg. interupt)
- `E` is a blackbox, its actions appear unpredictable to `A`
- In **Record and Replay** we
  - Record the behavior of `E` during program execution
  - Replay the behavior in a replay phase
- Our case: WebAssembly code is `A` and Javascript code is `E`

Collecting Traces

```
(module
    (import "env" "foo" (func $foo))
    (func (export "main")
        i32.const 69
        i32.const 420
        i32.store
        call $foo
        i32.const 69
        i32.load    ;; 12
    )
    (memory (export "mem") 1)
)
```

Trace

```
ExportCall;1;main;
ImportCall;0;foo;
ImportReturn;0;
Load;12
```

Replay Binary

```
function foo() {
    wasm.instance.exports.mem[69] = 12
}
```

Current State

- Instrument WebAssembly with Wasabi
- Collect traces and generate replay for 50+ (micro)tests
- Record and Replay of real website with Wasm-R3

CLI

```
$ r3 record https://playgameoflife.com/ trace.r3
```

```
$ r3 generate trace.r3 replay.js
```

That’s All