

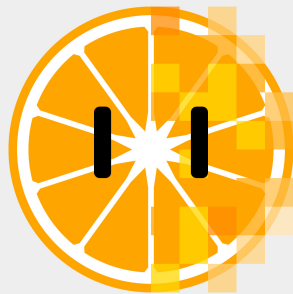
PHPerKaigi 2024 で発表した WebAssembly ランタイムのその後

nsfisis (いまむら)

第 166 回 PHP 勉強会@東京

自己紹介

いまむら
nsfisis



@ デジタルサーカス株式会社

前回までのあらすじ

前回までのあらすじ

WebAssembly とは

- Wasm とも
- ブラウザなどで実行できるポータブルな仮想命令セット
- 元々のモチベーション: ブラウザでの高速な処理
- ブラウザ以外の環境でも動く

Emscripten

- C/C++ のソースコードを wasm に変換
 - LLVM を使う言語
- C/C++ で書かれた膨大な資産をブラウザなどの上で動かせる

Wasm の活用例

- PHP の処理系は C で書かれている
- Emscripten を使って wasm に変換できる
- Wasm に変換すると PHP の処理系をブラウザなどの上で動かせる

PHP runtime on wasm runtime

PHPerKaigi 2024 で発表

- PHP 処理系
- の上で動く PHP 製の wasm 処理系 (自作ランタイム)
- の上で動く Emscripten で wasm に変換された PHP 処理系
- の上で動く `echo "Hello, World!\n";`

オーバヘッドが酷すぎて実行に 30 秒かかる

前回までのあらすじ終わり

前回までのあらすじ終わり

PHPerKaigi 2024 にて

- 「RubyVM を PHP で実装する～ Hello World を出力するまで」
- めもりーさんの発表
- PHP で実装された Ruby の VM

PHPPerKaigi 2024 にて

- 「RubyVM を PHP で実装する～ Hello World を出力するまで」
- めもりーさんの発表
- PHP で実装された Ruby の VM

お？

これを

- PHP 処理系
- の上で動く PHP 製の wasm 処理系 (自作ランタイム)
- の上で動く Emscripten で wasm に変換された PHP 処理系
- の上で動く `echo "Hello, World!\n";`

- PHP 処理系
- の上で動く RubyVM (めもりーさん作)
- の上で動く `puts "Hello, World!"`

こうしたら？

- PHP 処理系
- の上で動く PHP 製の wasm 処理系 (自作ランタイム)
- の上で動く Emscripten で wasm に変換された PHP 処理系
- の上で動く RubyVM (めもりーさん作)
- の上で動く `puts "Hello, World!"`

必要なこと

- パフォーマンス改善
 - Hello, World! に 30 秒かかるのに RubyVM は動かせない
- Wasm 仕様への準拠
 - PHPerKaigi でのデモを動かすためだけのやっつけ実装
- Emscripten との戦い
 - 後述

パフォーマンス改善

- Ya8 で発表
- PHPerKaigi 時点から 10 倍ほど高速化
- 33 秒 => 3 秒を切る
- 2.2 GiB => 308 MiB

公式のテストを通す

- WebAssembly には公式のテストスイートがある
- 元々はほぼ通らなかった
 - 15 % => 85 %

公式のテストを通す

- WebAssembly には公式のテストスイートがある
- 元々はほぼ通らなかった
 - 15 % => 85 %
- とんでもないバグが多数

loop 命令

```
$m = count($blockType->params);  
$n = count($blockType->results);  
$label = new Label($n);
```

```
$m = count($blockType->params);  
$n = count($blockType->results);  
$label = new Label($m);
```

f64.div

```
$return $x / $y;
```

```
return fdiv($x, $y);
```

除算演算子は IEEE 754 に準拠していない

Emscripten

Emscripten と格闘する

Emscripten

- C/C++ のソースコードを wasm に変換
 - LLVM を使う言語

Emscripten

- C/C++ のソースコードを wasm に変換
 - LLVM を使う言語
- Wasm 単体では外の世界に干渉できない
 - 標準入出力、ファイルシステム、etc
 - Wasm ランタイム側で実装して wasm へ渡す

Emscripten

- C/C++ のソースコードを wasm に変換
 - LLVM を使う言語
- Wasm 単体では外の世界に干渉できない
 - 標準入出力、ファイルシステム、etc
 - Wasm ランタイム側で実装して wasm へ渡す
 - JavaScript 実装の関数を出力する
 - PHP からは扱えない

Emscripten

- C/C++ のソースコードを wasm に変換
 - LLVM を使う言語
- Wasm 単体では外の世界に干渉できない
 - 標準入出力、ファイルシステム、etc
 - Wasm ランタイム側で実装して wasm へ渡す
 - JavaScript 実装の関数を出力する
 - PHP からは扱えない
 - Emscripten が生成した JS の関数群を PHP に移植する

RubyVM を動かす

デモ