10 Minutes in WASM

WebAssembly in 2018

with DRAX!

About This Talk

This talk was developed and presented in 2018 based on lessons learned while implementing a spellcheck feature for a web-based writing tool.

The talk was originally presented as a live lightning talk for the NDev software development community as part of NDev meetup #27.

What is WebAssembly

Are we talking about C++ programmers writing websites?

NO. WELL, NOT EXACTLY. C++ is one of a bunch of languages supported, and Rust seems to be leading the pack right now.

Wait, is this a binary format for the web?

SORT OF. It's a **portable** binary format, so it targets a virtual machine with access to a limited subset of hardware capabilities

Is it at least a separate-but-equal VM in the browser?

ISH. "Equal but not separate" might be the simplest way to put it. JS is the lingua franca of the web, and WASM interoperates where it can.

Where is WebAssembly?

Can I use this in Lynx 2.8.8? I only browse in text!

BEWARE. Most browsers got their WASM implementation within the last 12 months.

What tools are available?

IT'S COMPLICATED. ***Emscripten*** (LLVM->asm.JS) and ***binaryen*** (asm.js->WASM) are the most widely used tools.

Is this on the INTERWEBS?

/TUMBLEWEED. Support is recent and advantages are limited (qv), so not much (yet).

Wait, is this...IN THE FUTURE?

THIS MARGIN IS TOO NARROW. It's not clear yet, but expect some deep internals to get rewritten at minimum

Why is WebAssembly?

So I write WASM instead of JS now?

UNDER NO CIRCUMSTANCES DO THIS. WebAssembly is best for “tight loops”. JS is already highly tuned for app code

Wait, I have to learn TWO languages?

WELCOME TO PROGRAMMING. Think of WASM as the big gun, with all that implies.

When do I write WebAssembly?

YOU DON’T. If you have a good compiled-language implementation of a CPU-intensive module, however, you can now bring it into the browser.

Am I going to hate it?

DO YOU HATE YOUR OWN MOTHER? Choose your compiled language(s) wisely.

How is WebAssembly?

Will I be set on fire the first time I fire this baby up?

> **YOU WILL NOT.** WebAssembly in the general case tends to be about 20-30% faster than JS. In specific, non-boundary-crossing scenarios, however, it can be much faster.

Is this how the hackers break my computer?

> **WHAT IS THE SOUND OF ONE TAB CRASHING?** Too early to say, but it's still a VM.

Can I get it into my app this weekend?

> **PATIENCE, YOUNG ONE.** Integrating pre-rolled modules from, say, NPM, is surprisingly easy. HOWEVER getting the full toolchain up and running is a lot harder.

What am I watching out for?

> **HERE BE DRAGONS.** At minimum, you'll need to load the module out-of-band.

When is WebAssembly?

Seriously, though: can I plug this in today?

NOT RECOMMENDED. But you should be doing experiments in the next 12 months!

Why does Safari a crash?

TRIPPING HAZARD. The support is extremely uneven right now, particularly in Safari.

Hasn't this been around for years?

Chrome 57, Firefox 52, Safari 11.1/11.2(Full), EdgeHTML 16(MVP of MVP)/17

What if I'm planning ahead?

CONFORM! The MVP is complete and W3C standardization underway.

Who is WebAssembly?

Who's onboard?

ALL YOUR BROWSERS ARE BELONG TO US. Mozilla, Google, Microsoft, WebKit

Who is already working with WASM?

AutoCAD - http://through-the-interface.typepad.com/through_the_interface/2018/03/the-future-of-autocad.html

Unity - https://forum.unity.com/threads/webgl-roadmap.334408/

Unreal - https://s3.amazonaws.com/mozilla-games/ZenGarden/EpicZenGarden.html

Is WebAssembly?

Is this really going to work?

RESISTANCE IS FUTILE. The major browsers are in the working group and games and industry are showing up in force.

How annoying is the build system?

IT GOES TO ELEVEN. Everything you hate about GCC, but with more compile steps!

Do I WebAssembly?

Why did I want to use WASM?

Hunspell. Ohhhh Hunspell.

What happened?

DEATH DARKNESS LET US NEVER SPEAK OF IT AGAIN. Actually, if we didn't have to support Safari, we'd have it in production right now

Will I use it again?

TEAM WEBASSEMBLY. It's fast, it's widely supported, it interoperates with JS, and it gives me an excuse to dick around in C++ and Rust. What's not to love?

Should You WebAssembly?

How do I sell this to Lumbergh?

> **I'M GONNA NEED YOU TO GO AHEAD AND OPTIMIZE THAT, MMKAY?** Anywhere your application pegs CPU, you can probably improve it with WASM.

What if I'm network-bound?

> **IT'S NOT ALL ABOUT YOU, WINSTON.** JS is still fast! WASM is a niche tool, and you can safely ignore it unless and until you need it.

TEACH ME CEE PLUS PLUS!

> **You do not want this.**

WebAssembly of Christmas Future

What’s next?

Threading, reference types, ES Modules, oh my! Proposals available at https://github.com/WebAssembly

Coming to a browser near you!

MVP is basically implemented in all browsers, but probably safest if you can wait for Safari 11.2 and EdgeHTML 17 to reach full penetration.

WASM without all the WASM

NPM and other package managers already have WASM modules (like hunspell-asm!) available. Keep an eye out for these!