Universidade de Itaúna

Aimée S. S. Ferreira

Isis Estevan Mendonça

Italo Médici S. Souza

WebAssembly

Itaúna

Outubro/2022

Sumário

O que é ?

O WebAssembly é um novo tipo de código que pode ser executado em browsers modernos — se trata de uma linguagem de baixo nível como assembly, com um formato binário compacto que executa com performance quase nativa e que fornece um novo alvo de compilação para linguagens como C/C++, para que possam ser executadas na web. Também foi projetado para funcionar em conjunto com o JavaScript, permitindo que ambos trabalhem juntos.

Principal Objetivo:

O grande propósito da WebAssembly é a compatibilidade e seu principal objetivo do WebAssembly é facilitar aplicativos de alto desempenho em navegadores web, apesar do formato ter sido construído para ser executado e integrado também a outros ambientes.

Inovação:

A grande inovação deste código binário é que ele traz o desempenho de aplicativos nativos para a web de uma forma completamente segura e ainda permite uma funcionalidade completa de jogos, aplicativos principais e todo o espectro de software que pode ser executado em um computador.

Por isso, desenvolvedores de código aberto estão continuamente contribuindo para o conjunto de recursos do WebAssembly até que possam capturar com eficiência as principais linguagens. Eles estão entusiasmados com a ideia de ter um alvo de compilação universal que poderia eventualmente desbloquear a habilidade de qualquer linguagem de almejá-lo.

Benefícios:

Um dos grandes benefícios do WebAssembly é que ele oferece a possibilidade de compilar código em outras linguagens.O Wasm possui um compilador official source to source chamado Emscripten. O Emscripten permite várias otimizações de build e possui um backend LLVM, mas é suportado apenas para as linguagens C/C++ e Rust. Com a expansão da tecnologia, hoje existem algumas opções para outras linguagens de alto nível como Go, C#, Java, D, Perl, Python, Scala, Kotlin, Swift e Ruby.

O WebAssembly foi projetado para funcionar bem na Web, mas não se limita a ela. A WASI é uma família de APIs para WebAssembly projetada e padronizada por meio do WASI Subgroup do W3C WebAssembly Community Group. Inicialmente, o foco está em APIs orientadas ao sistema, abrangendo arquivos, redes e vários recursos nativos por meio de Capability-based security, só que isso já é outra história.

Problemas que o WebAssembly soluciona

O principal problema que o WebAssembly pretende resolver é o desempenho, Embora tenhamos JavaScript na web, ele não foi realmente projetado para ser rápido com um aplicativo muito grande. Já o Wasm, por ser construído apenas como um formato binário e muito compacto para fazer o download, torna-se muito eficiente para compilar e executar.

Quem está usando?

Google, Microsoft, Mozilla e Apple são alguns nomes que têm trabalhado em conjunto para tornar os navegadores mais rápidos. Para se ter uma ideia, no protótipo atual criado, o navegador processa o WebAssembly 23 vezes mais rápido que um código JavaScript tradicional.

Funcionamento:

Para criar um módulo WebAssembly é necessário escolher uma linguagem que suporte a compilação para a arquitetura Wasm. Utilizando o Emscripten, é possível compilar um código C++ para WebAssembly e executá-lo em um navegador. Na Tabela 1 é possível ver o código fonte em C++, resultado equivalente em Wat (WebAssembly Text Format) e no formato binário. O formato de texto representa um módulo WebAssembly e permite a leitura e inspeção do binário na web.

C++	Binário (hexadecimal)	WAT (formato textual)

`int factorial(int n) {`
`  if (n == 0)`
`  return 1;`
`  else`
`  return n * factorial(n-1);`
`}`
	`20 00 42 00 51 04 7e`
	`42 01 05 20 00 20 00`
	`42 01 7d 10 00 7e 0b`
		`(func (param i64) (result i64)`
		`  local.get 0`
		`  i64.eqz`
		`  if (result i64)`
		`  i64.const 1`
		`  else`
		`  local.get 0`
		`  local.get 0`
		`  i64.const 1`
		`  i64.sub`
		`  call 0`
		`  i64.mul`
		`  end)`

Conclusão

Imagine um mundo onde as aplicações serão compatíveis entre vários dispositivos, independente do sistema operacional e do navegador. O WebAssembly oferece a mesma segurança e políticas de permissões dos navegadores, e ainda trabalha em harmonia com outras tecnologias web, mantendo a compatibilidade retroativa. O que se acredita é que ainda há otimizações que vão levar o WebAssembly ainda mais longe, permitindo que aplicativos enormes carreguem rapidamente, mesmo em dispositivos móveis.

Bibliografia

https://webassembly.org/https://webassembly.org/

https://www.cin.ufpe.br/~tg/2020-3/TG_SI/tg_vgcp.pdf

https://www.supero.com.br/blog/webassembly-o-que-voce-precisa-saber-antes-de-usar/