

Analisi e Valutazione delle Performance di WebAssembly System Interface

Candidato: Conti Francesca

Relatore: Chiar.mo Prof. Ing. Paolo Bellavista

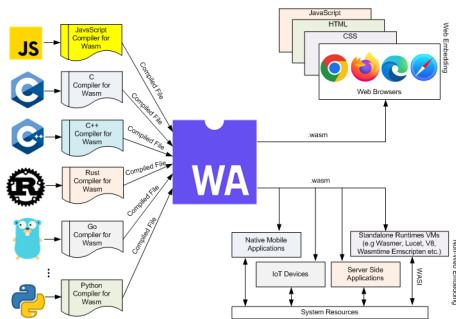
Correlatore: Ing. Andrea Garbugli

19th March, 2024

WebAssembly

Al giorno d'oggi le applicazioni Web stanno diventando sempre più complesse e pesanti e l'inefficienza di JavaScript inizia ad essere un problema.

- WebAssembly nasce per eseguire codice in modo efficiente all'interno del **browser**.
- Linguaggio di basso livello simile ad Assembly.
- Il codice WebAssembly viene eseguito all'interno dei **runtime**.



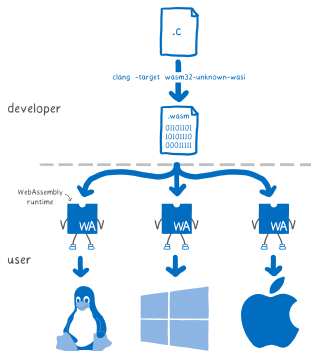
Crediti:¹

¹ Partha Ray. *An Overview of WebAssembly for IoT: Background, Tools, State-of-the-Art, Challenges, and Future Directions*

WASI

WASI permette al codice WebAssembly di accedere alle risorse di sistema in modo **standardizzato**.

- Apre nuove possibilità per applicazioni Web.
- Portabilità su diverse architetture e sistemi operativi.
- Sicurezza migliorata: **sandboxing** e **capabilities**.



Crediti:²

² Lin Clark. *Standardizing WASI: A system interface to run WebAssembly outside the web*.

Runtime

I vari runtime utilizzati sono i seguenti:

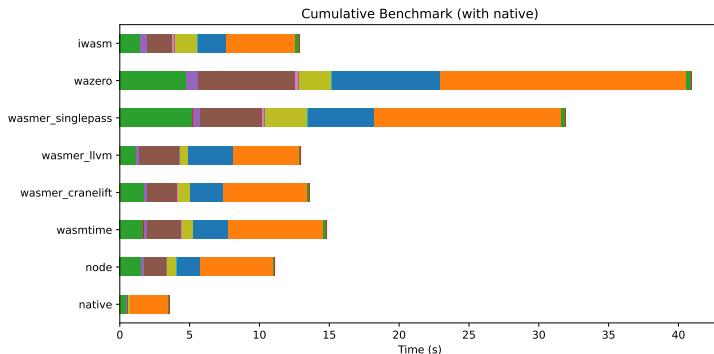
- Wasmer (LLVM, Single Pass e Cranelift)
- Wazero
- iwasn
- Node.js
- Wasmtime



Test Libsodium

- I test sono basati su **Libsodium**, libreria crittografica open-source con funzioni per crittografia, firme digitali e hashing.
- È stata scelta la libreria Libsodium in modo da poter stressare specifiche parti del sistema.
- In tutto sono state testate 6 diversi tipi di funzionalità della libreria.
- Ripetizione 100 volte dei test su ogni runtimes.

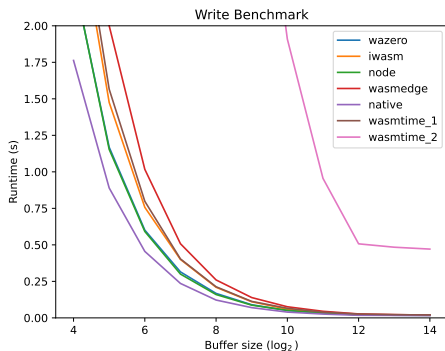
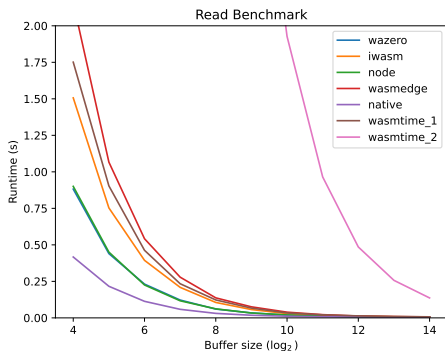
Risultati Libsodium



- Wazero e Wasmer singlepass sono i peggiori.
- Node, iwasm e Wasmer LLVM sono i migliori.

Test e risultati lettura/scrittura file

Sono inoltre state testate le **velocità** di lettura e scrittura di un file di circa 50 MB con i diversi runtime e con il codice nativo.



Conclusioni - Sviluppi futuri

- WebAssembly è una tecnologia **estremamente promettente** anche grazie a WASI.
- È però ancora **in fase di sviluppo** e l'integrazione con WASI non sempre è funzionante.



- Test per la comunicazione di rete usando le **socket**, al momento non ancora funzionante.