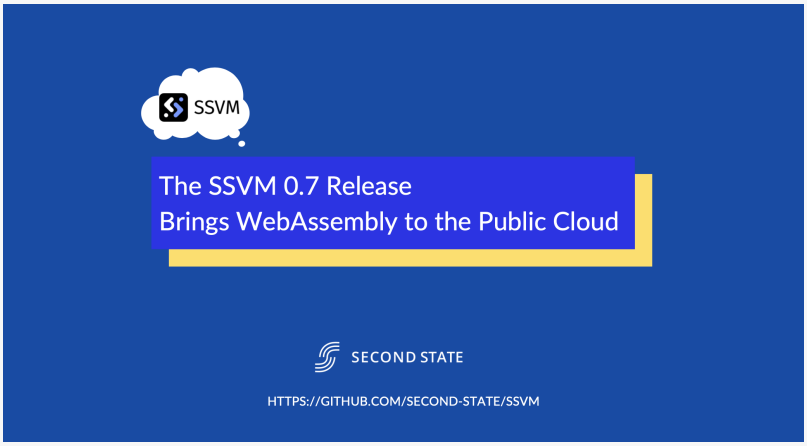
The SSVM 0.7 release brings WebAssembly to the public cloud

SSVM 0.7 is a significant release for an open source WebAssembly VM optimized for cloud, edge, blockchain and serverless computing.



AUSTIN, TEXAS, UNITED STATES, February 2, 2021 /EINPresswire.com/ -- The SSVM (Second State VM) is a popular WebAssembly VM optimized for high-performance applications on the server side. The recent SSVM release 0.7 brings a wide array of unique features that further differentiate and solidify the SSVM as a leading cloud native application runtime.

WebAssembly proposals

The SSVM supports optional WebAssembly features and proposals. Those proposals are likely to become official WebAssembly specifications in the future. For example, the SSVM has supported the WASI (WebAssembly Systems Interface) spec for WebAssembly programs to securely interact with the host Linux operating system. With release 0.7, the SSVM supports the following proposals.

- Reference Types allows WebAssembly programs to exchange data with host applications and operating systems.

- Bulk memory operations. The WebAssembly program sees faster memory access and performs better with bulk memory operations.

- SIMD (Single instruction, multiple data). For modern devices with multiple CPU cores, the SIMD allows data processing programs to fully take advantage of the CPUs. SIMD could greatly enhance performance of data applications.

Furthermore, the SSVM team is exploring the wasi-socket proposal to support network access in

WebAssembly programs.

Powerful WASI-like extensions

A key differentiator of the SSVM from other WebAssembly VMs is its support for non-standard extensions. The WASI spec provides a mechanism for developers to extend WebAssembly VMs efficiently and securely. The SSVM team created the following WASI-like extensions based on real-world customer demands for release 0.7.

- Developers can write Tensorflow inference functions using a simple Rust API, and then run the function securely and at native speed inside SSVM. This feature is supported in Second State Functions.

- Besides Tensorflow, the Second State team is building WASI-like extensions for AI frameworks such as ONNX and Tengine for SSVM.

- The SSVM storage interface allows WebAssembly programs to read and write a key value store. This feature is supported in Second State functions.

- SSVM allows WebAssembly functions to execute native commands in the host operating system. It supports passing arguments, environment variables, STDIN / STDOUT pipes, and security policies for host access. The http_proxy in Second State Functions is an example of using operating system commands in a serverless function.

- The SSVM Ewasm extension supports Ethereum smart contracts compiled to WebAssembly. It is a leading implementation for Ethereum flavored WebAssembly (Ewasm).

- The SSVM Pallet allows the SSVM to act as an Ethereum smart contract execution engine on any Substrate based blockchains.

Capability-based security

WASI and WASI-like extensions control access to the host system through a declarative and capability-based security model. This approach allows the WebAssembly VM instance to be secured by explicit rules that are independent from the user running the VM instance. For example, with WASI, the SSVM host application can specify which file folders in the host operating system the VM has access to.

SSVM 0.7 added support for operating system native commands. It is important that the sandboxed WebAssembly bytecode application only has access to the operating system commands it requires and declares. The SSVM host application can now whitelist commands the VM is allowed to access.

Cross-platform

WebAssembly VM on the server side allows applications to be compiled on one machine (eg the developer's laptop or a CI machine) and deployed on a variety of different platforms. For public and edge clouds, the standard runtimes are often older versions of Linux that are no longer widely used in development. For example, the collaboration between Second State and Tencent Cloud on Serverless Tensorflow functions require the SSVM and its WASI-like Tensorflow extension to run on a hardened CentOS 7 image.

The SSVM is developed on Ubuntu 20.04 to take advantage of advanced LLVM features for the AOT compiler. With SSVM 0.7, we are now building and releasing statically linked SSVM binaries for older Linux distributions. The manylinux1 build of the SSVM binaries can run on CentOS 5, which was released in 2007. Going forward, SSVM is now being ported to a wide array of server and embedded operating systems and hardware platforms.

Cloud-native support

The SSVM is a "cloud native" WebAssembly VM . It aims to provide cloud services to application developers. The SSVM 0.7 is integrated into cloud native application frameworks as a high-performance and secure runtime engine.

- The Second State Functions is a public serverless function-as-a-service (FaaS). It runs computationally intensive functions safely in SSVM. The Second State Functions service is built on the Wasm Joey open source project, which relies on Node.js to instantiate and manage SSVM instances.

- The Tencent Serverless Cloud Function collaborates with Second State to support Tensorflow inference functions through the SSVM. Developers can get started within minutes with a Serverless Framework template.

- YoMo is an open-source Streaming Serverless Framework for building Low-latency Edge Computing applications, built atop QUIC Transport Protocol and Functional Reactive Programming interface. It utilizes the SSVM to execute serverless functions in edge settings.

Going forward, SSVM will support the OCI (Open Container Initiative) specification, which will allow SSVM instances to be managed by cloud native orchestration tools such as Kubernetes.

Language support

Besides general-purpose programming languages, like C, C++, Rust, and AssemblyScript, the SSVM supports niche programming languages for specific industry verticals. For example, a common use case for SSVM is to run smart contracts on Ethereum compatible blockchains. The SSVM 0.7 is adopted by leading blockchains such the Oasis Network and Substrate / Polkadot

community as the Ewasm (Ethereum flavored WebAssembly) VM.

- Solidity is a leading programming language for Ethereum smart contracts. The SOLL compiler, developed by Second State, compiles Solidity programs to WebAssembly bytecode that can run inside the SSVM.
- Fe is a new and simplified smart contract language from the Ethereum foundation. The SOLL compiler, together with Ethereum Foundation's solc tool, can compile Fe smart contracts to WebAssembly for SSVM.

SSVM 0.7 is a significant release for an open-source WebAssembly VM optimized for cloud, edge, blockchain, and serverless computing.

Vivian Hu
Second State
+1 415-894-2578
email us here
