ParaState: Evolving Ethereum on Polkadot

The Ethereum protocol is the dominant protocol for DeFi (Decentralized Finance) and Dapp (Decentralized Apps) today. Almost all blockchain-based Dapp software and a large portion of programmable digital assets run on the Ethereum Virtual Machine (EVM). However, originally designed and developed by a loose group of inexperienced software developers, including a young Dr. Gavin Wood, the EVM is also a simplistic, rigid, and slow code execution sandbox. That has resulted in many problems we see on Ethereum today, including unsafe smart contracts, very limited programming language support, slow performance, and high gas fees.

> “The [Ethereum] platform itself is full of risk, as you would expect with a nascent technology. … It’s the price you pay for the ability to access this world-changing software in its days of inception.” – Dr. Gavin Wood, commenting on Ethereum in an interview with Unitimes in 2019

ParaState aims to provide an industry standard, highly optimized, and extensible smart contract execution environment, while maintaining Ethereum compatibility. ParaState provides a suite of open source infrastructure software that includes a Substrate pallet that can be incorporated into any Substrate-based blockchains, including all Polkadot ParaChains. The development effort of ParaState is supported by a developer treasury that collects fees from all transactions processed by the ParaState runtime software.

WebAssembly to the rescue

The solution to the above mentioned EVM problems is to upgrade the blockchain virtual machine. The WebAssembly virtual machine has emerged as a leading candidate for such an upgrade. In fact, next-gen blockchain systems such as EOS, Polkadot (Substrate and Ink!), Cosmos (CosmWasm), Oasis, NEAR, Solana, and others, have all chosen WebAssembly as the runtime engine for blockchain applications and on-chain smart contracts. WebAssembly is a well established industry standard. It is adopted by communities ranging from web browser application developers to cloud computing providers to blockchains. As a result, WebAssembly is well supported on a wide range of computer hardware, operating systems, and programming languages. It is heavily optimized for performance.

However, most of those WebAssembly-based blockchain virtual machines are NOT compatible with existing EVM applications. That creates hurdles for existing Dapp developers. To address those issues and create a path forward for existing Ethereum ecosystem developers, the Ethereum Foundation proposed an on-chain WebAssembly VM for EVM applications, called the WebAssembly flavored Ethereum Virtual Machine, or Ewasm. Ewasm would not be compatible with EVM at the bytecode level. However, Solidity programs written for the EVM can be simply re-compiled into WebAssembly and run on Ewasm. Ewasm shares the same accounts and blocks data structure with the EVM, and provides a compatible web3 RPC interface for external Dapps.

The Second State WebAssembly VM (SSVM) is an industry leading WebAssembly implementation. According to a research paper published on IEEE Software, it is the highest performing WebAssembly VM on the market. The SSVM also supports standard and draft WebAssembly extensions, such as the WebAssembly Systems Interface (WASI), interface types, and bulk memory operations. Through its extension framework, the SSVM fully supports the Ewasm specification. It is one of the only two WebAssembly VMs that pass the entire Ewasm testsuite created by the Ethereum Foundation. In 2020, the web3 foundation funded Second State to port the SSVM Ewasm to the Substrate ecosystem. As part of the grant project, Second State created the SSVM pallet for Substrate blockchains.

The SSVM pallet is a key component in the ParaState software suite. It can be installed side-by-side with the EVM pallet on the same Substrate blockchain. ParaState is a one-stop shop for present and future Ethereum developers.

Ethereum on Polkadot

A blockchain network that supports the EVM as a smart contract runtime is an Ethereum compatible blockchain. Examples of Ethereum compatible blockchains include Ethereum, Ethereum Classic, Oasis Ethereum ParaTime, CyberMiles, RSK, Athereum, Binance Smart Chain, and many others. Those blockchains are interoperable with Ethereum and with each other at the software and API level.

- EVM smart contracts, written in Solidity or other front end languages, can be deployed to any of the Ethereum compatible blockchains without change.
- Dapp, including DeFi app, front end UIs can also be ported across Ethereum compatible blockchains with reasonable amount of efforts.
- Crypto assets, such as native tokens and ERC-20 tokens, can be exchanged across Ethereum compatible blockchains through decentralized Atomic Swap smart contracts and centralized gateways.

The ParaState software suite is compatible with Ethereum applications. We are committed to port popular Ethereum DeFi applications, such as Uniswap, Balancer, Compound, and many others to ParaState's runtime environment. We will also work with Ethereum bridge providers in the Polkadot ecosystem to bring Ethereum-based token assets to ParaState-enabled Polkadot parachains.

Extend the developer ecosystem

With WebAssembly, developers will be able to write smart contracts in over 20+ programming languages LLVM supports in addition to Solidity. As Solidity’s shortcomings in developer productivity and application security become apparent, developers are looking for alternatives in more mature programming languages such as C/C++, Go, Rust, and even Java. Programming language support is also crucial for attracting new developers into the smart contract ecosystem.

Furthermore, as blockchain applications become specialized, smart contract developers are moving toward Domain specific languages (DSL) for use cases such as finance applications to archive better security and performance. Examples of financial smart contract DSLs include Facebook Libra’s MOVE for stable coin applications, Certik’s DeepSEA for formally verified contracts, Digital Asset Modeling Language (DAML) for asset tokenization, and many more.

With an open source and widely used compiler toolchain, Ewasm also supports customization of programming language features. For example, Second State and Oasis Labs proposed an extension to the Solidity language to support confidential data fields inside smart contracts. Such language extensions can be proposed and implemented similar to DSLs on the WebAssembly-based toolchain. They are impossible with the traditional and rigid solc compiler and EVM.

The Second State SOLL compiler is the only LLVM-based Solidity and YUL language compiler to compile existing Ethereum smart contracts to run on Ewasm. Furthermore, ParaState provides SDKs and libraries for popular programmings such as Rust, C, and C++ so that smart contracts can be written in those languages. The SOLL compiler and language SDKs are all part of the ParaState software suite.

Performance beyond the TPS

The current Ethereum is notorious for its low transaction throughout and high gas fees associated with the congestion. Measured in transactions per second (TPS), Ethereum mainnet can only process around 25 TPS, which is far below the requirement for consumer Internet applications. In comparison, the VISA network reaches 2000 TPS regularly for credit card transactions. The Polkadot ecosystem can support at least 10,000 TPS through its parachain architecture. Each parachain can support 1000+ TPS.

However, for smart contract platforms, TPS is not a good measure for real world performance. We need to evaluate how fast the on-chain virtual machine can execute smart contracts. The EVM, due to it’s simple interpreter design and a general lack of optimization, is a very slow virtual machine by modern standards.

The mature ecosystem around WebAssembly allows the Ewasm smart contracts to archive much higher performance than EVM smart contracts even if they are compiled from the same Solidity source code. For instance, the Ewasm can utilize JIT (Just in Time) and AOT (Ahead of Time) compiler optimizations at runtime to improve performance by 100x compared with EVM's interpreter execution mode.

The Second State VM (SSVM) is a one of the fastest WebAssembly VMs. It features AOT optimization across multiple CPU and operation system architectures. It is one of the only Ewasm compatible VM implementations on the market today, with special performance optimization for Ethereum's native 160-bit integers.

The ParaState software suite features the SSVM as its Ewasm execution engine. It provides a large performance boost for Ewasm applications.

Developer treasury and the STATE token

The ParaState software suite consists of the SSVM pallet, the EVM pallet, the SOLL compiler toolhain, high level language SDKs for Ethereum compatibility, and ported Ethereum DeFi applications. All of these components are open source software. The non-profit ParaState Foundation funds and manages the development of the software suite. It has a novel approach to fund sustainable open source software development.

All Substrate-based blockchains are free to incorporate all or part of the ParaState software suite in their node software stack. By doing so, those blockchains gain the developer tools, developer communities / ecosystem, and high-performance runtimes from ParaState. In exchange, the ParaState-enabled blockchain will collect a percentage of (e.g., 20%) the gas fee for each on-chain smart contract transaction executed by the SSVM, and send the collected fee (i.e., a "gas tax") to a developer treasury account controlled by the ParaState Foundation. The SSVM pallet performs this collection automatically without any source code change. The developer treasury collects a basket of native cryptocurrencies from participating blockchains.

The STATE token is minted by the ParaState Foundation and is backed by the basket of tokens in the developer treasury. There is a fixed total supply of STATE. The ParaState Foundation is the custodian of the developer treasury. The Foundation will periodically sell tokens from the developer treasury for STATE. It would support the price of STATE and allow the Foundation to accumulate STATE. The Foundation then sells STATE in the open market to raise funds for developer salaries and community development.

Conclusions

The ParaState software suite brings Ethereum protocol support to the Substrate / Polkadot ecosystem. While it is backward compatible with today’s EVM applications, it future-proofs the Ethereum protocol by bringing the LLVM and WebAssembly developer communities into the Polkadot ecosystem. It is Ethereum on Steroids.

Join ParaState. Experience tomorrow’s Ethereum developer experience today!