Digital Currency Initiative
Massachusetts Institute of Technology

Parallelized Architecture for Scalably Executing smart Contracts (PArSEC)

Executive Summary

August 1st, 2023

MIT Digital Currency Initiative

The views expressed in this paper are those of the authors and do not necessarily reflect the views of the Federal Reserve Bank of Boston, the Board of Governors or the Federal Reserve System.

Introduction

Many central banks are exploring how public money might be transformed into a digital asset. Today nearly all central banks are conducting research into or are actively experimenting with the issuance of central bank-issued digital currency, or CBDC.

The benefits of digitizing money have so far included increasing the speed, efficiency, and convenience of payments, as seen in the digitization of commercial bank and private money. Yet digitalization may offer additional functionality for money that is not currently available with our existing payment systems.

With this effort, we sought to explore an architecture that examined the tradeoffs of including more complex functionality, specifically supporting smart contracts. Smart contracts are electronic agreements that self-execute according to predefined rules.[1] They enable users of the system to engage voluntarily in pre-specified contractual financial transactions.

Researchers and policymakers have identified understanding the potential tradeoffs of including more complex functionality, like programmability and smart contracts, as a priority for CBDC design.[2] Automating some money-based functions may unlock future innovations in the financial services industry, but many questions remain regarding which use cases might be most impactful, and how to make the appropriate trade-offs around security and resilience. In our research we exclusively address technology questions. The accompanying paper presents our latest research on this flexible technical architecture that can execute a wide variety of smart contracts at scale.

Learn more at https://dci.mit.edu/opencbdc or download & contribute at https://github.com/mit-dci/opencbdc-tx

Summary of Results

We introduce a design and implementation for a new centralized platform called PArSEC (Parallelized Architecture for Scalably Executing smart Contracts). PArSEC is a distributed platform for running a variety of smart contract virtual machines (VMs). It supports flexible research and testing: by executing smart contracts in these VMs, the system can support clients easily updating their contracts' logic without having to understand and alter the core architecture of the underlying transaction processor. We focused on smart contracts because they provide the highest degree of expressivity and functionality to users. Smart contracts are also composable: users can create transactions that combine the results of multiple contracts, even if they were not originally written to work together.

This platform enables the following:

1. **High throughput and linear scalability.** We run smart contracts in a centralized setting with linear scalability, achieving throughput of over 100K transactions per second.[3] PArSEC achieves linear scalability by enabling parallel execution of smart contracts, distributing processing requirements across many machines. Specifically, we observed linear scaling of our workload on PArSEC as more computers are added to the system and a peak throughput for non-conflicting ERC-20 transactions of 118K transactions per second with an average transaction time under 1.6 seconds.

2. **Flexibility.** PArSEC has a modular architecture which supports running many different types of smart contract environments simultaneously. This means that researchers and developers can experiment with different smart contract languages and functionality without having to change the underlying transaction processor. This might make PArSEC especially useful as a platform for experimentation.

3. **Reusing existing tools.** We designed PArSEC so that once an existing virtual machine is set-up, developers can use existing, unmodified smart contracts directly from that ecosystem and open source tools to deploy and call smart contracts directly. We demonstrate this by porting an implementation of the Ethereum Virtual Machine (EVM) and running unmodified Ethereum smart contracts with existing tooling.

We invite researchers and policymakers to work with us to develop the system further. With this intent, we are releasing all software from our research publicly on GitHub under the OpenCBDC project.[4]

Potential Use Cases

PArSEC enables anyone to test applications for smart contracts at scale, exploring opportunities to unlock new functionality. Potential use cases for PArSEC include the following:

1. **Private-sector experimentation with applications like decentralized exchanges**

   A number of applications for programmability exist in today's financial system from deposit tokens[5] to automated recurring payments. One of the most important innovations to arise from the decentralized finance ("DeFi") ecosystem in recent years is the decentralized exchange of tokenized assets via automated market-makers (AMMs). AMMs are therefore on the research agendas of many central banks and are the central focus of the BIS Innovation Hub Singapore Center's Project Mariana.[6]

   Because PArSEC supports ERC-20 tokens, our implementation can run AMM smart contracts with minimal modifications. Unlike public permissionless blockchains, however, PArSEC can support far higher transaction throughput with considerably lower time to finality.

   An AMM deployed on PArSEC could also be made to trade other assets such as bonds, tokenized securities and repurchase agreements. Smart contract composability allows AMMs to swap anything expressed as an ERC-20 standard token.

2. **Cross-border interoperability platforms**

   Cross-border interoperability between currency systems is a critical research question that has been studied extensively in recent years.[7] Central banks and ecosystem participants are seeking to understand how to improve current settlement methods.

   PArSEC provides the linear scalability needed to support a large cross-border settlement platform. Such a platform could also enable cross-border contracting, providing new ways to automate cross-border supply chains and compliance checks.[8]

   PArSEC can be architected on an experimental basis to support multiple VMs running simultaneously on the system, and the system could be modified to allow different actors to run different VMs. This would enable central banks as well as private companies to collaborate even if they are using different underlying systems. However, there are many remaining security and user privacy questions which would need to be addressed before implementing such a system.

Conclusion

As exploratory research on the implications of different design choices, this work is not intended for a pilot or public launch. Significant research would be necessary to consider the security of the system, key management, and data migration tooling, among other critical topics that have not yet been fully explored. We hope that researchers and organizations engage with this software to further explore use cases and tradeoffs for smart contract implementations.

MIT DCI makes its work freely available as published papers and open source code to advance scientific knowledge, share potential engineering solutions, and encourage others to build on our work or to correct it where it may be wrong. Our goal is to share insight, encourage dialogue, and work with others to build a future for money and the financial system that better serves the needs of all.

Endnotes

[1] Szabo N. (1997), Formalizing and Securing Relationships on Public Networks, *First Monday*, 2, No. 9

[2] U.S. Department of the Treasury (2023), Future of Money and Payments, https://home.treasury.gov/system/files/136/Future-of-Money-and-Payments.pdf accessed 06/23/2023

[3] Linear scalability, or the ability to increase capacity by proportionally adding resources, holds where there is no data contention. Linear scaling will not hold in environments where a small number of keys are used by many transactions.

[4] https://github.com/mit-dci/opencbdc-tx accessed 06/23/2023

[5] See JPMorgan, Oliver Wyman (2023), Deposit Tokens: A foundation for stable digital money, https://www.jpmorgan.com/onyx/content-hub/deposit-tokens.htm accessed 06/23/2023

[6] https://www.bis.org/about/bisih/topics/cbdc/mariana.htm accessed 06/09/2023

[7] CPMI, BISIH, IMF, and WB (2022), Options for access to and interoperability of CBDCs for cross-border payments, https://www.bis.org/publ/othp52.htm accessed 06/23/2023

[8] Adrian T. et al (2022), "A Multi-Currency Exchange and Contracting Platform, *IMF Working Papers*, No. 2022/217