Duke CS+ Summer 2021: Decentralized Finance: Cryptocurrency and Blockchain on the Internet Computer: https://ic.pubpub.org/pub/7ms2ilwh/

Scalability Analysis of Dfinity's Implementation of Orthogonal Persistence as a Decentralized Data Storage Solution

By Oum Lahade
Project Leads: Profs. Luyao Zhang, Yulin Liu, Kartik Nayak, Fan Zhang
Graduate mentor: Derrick Adam

Summary:

Ethereum is currently the leading architecture upon which decentralized finance applications are built. The Internet Computer is a new public blockchain network developed by the DFINITY foundation. Among other novelties, the Internet Computer has implemented orthogonal persistence into the network to persist the memory pages of all Update Call code that is run, allowing developers to not think about storing data, as data persists perpetually. This is fundamentally different from developing on Ethereum. Due to large data storage costs on Ethereum, developers must use one of 2 types of data storage mechanisms: Centralized storage with blockchain hashes or Permissioned (private) blockchains. Ultimately, this proposal aims to measure the efficiency and scalability of Dfinity's orthogonal persistence against these options utilizing data storage costs as a proxy for efficiency. This will be done through the lens of the Liquity / WaterPark solutions. Decentralized borrowing/stablecoin protocols will be created on the Internet Computer and Ethereum networks encompassing all the 4 types of data storage methods currently in development on blockchain networks: directly on-chain, centralized storage + hashing, permissioned blockchains, and Dfinity's implementation of Orthogonal Persistence, as well as on a centralized network as well.

Intellectual Merit:

There is significant intellectual merit to this line of research. If copies of WaterPark are successfully deployed utilizing these 4 different types of storage solutions, as well as on a centralized network, analyzing the cost implications associated with these solutions will be extremely influential in understanding the true efficiency and scalability of Dfinity's implementation of Orthogonal Persistence or current Ethereum solutions such as IPFS. To ensure that performance gains are coming solely from the data storage solution, we will utilize fees associated solely with tasks related to writing / reading to storage. This way, the fault of the network that these protocols are built on does not impact our understanding of the scalability of the system. Furthermore, this allows us to make valid comparisons between protocols when looking at Gas Fees, because gas is paid in ETH for all protocols and the gas fee model is static across all implementations. This is incredibly important as it can contribute to future literature and development in the field by guiding developers towards implementing their data storage in a manner that is most efficient on a blockchain level, both in terms of cost and scalability.

Table of Contents

Abstract

Ethereum is currently the leading architecture upon which decentralized finance applications are built. The blockchain network currently utilized a proof-of-work consensus

mechanism - although it is moving towards proof-of-stake - and a gas fee model where users pay GAS fees determined in Ether. In theory, Ethereum enables infinite storage space. But, in return, you have to provide gas for every read/write operation. This cost changes all the time, depending on the network, the market and the way Ethereum develops. However, because Ethereum wasn't designed to store any other records but simple transactions, storing any serious amount of data on a public blockchain like Ethereum is very expensive. The Internet Computer is a new public blockchain network developed by the DFINITY foundation. Among other novelties, the Internet Computer has implemented orthogonal persistence into the network to persist the memory pages of all Update Call code that is run, allowing developers to not think about storing data, as data persists perpetually. This is fundamentally different from developing on Ethereum. Due to large data storage costs on Ethereum, developers must use one of 2 types of data storage mechanisms: Centralized storage with blockchain hashes or Permissioned (private) blockchains. Ultimately, this proposal aims to measure the efficiency of Dfinity's orthogonal persistence against these options utilizing data storage costs as a proxy for efficiency.

Introduction and Background

Internet Computer

Internet Computer canisters use orthogonal persistence to persist data beyond the length of the program running. This means that data can be stored in data structures embedded in the code rather than in files that need to be written to and read from. Additionally, the architecture of the system utilizes sharding, meaning that data is split among the data centers running the canisters. Over time, however, as update calls are made, commonly called data begins to be shared over multiple data centers.

There are three main implementations of orthogonal persistence: System images, Journals, and Dirty writes. The Internet Computer implements orthogonal persistence in a unique way by persisting the memory pages within a single software canister in a manner similar to the system images approach.

However, software canisters are limited by WebAssembly limitations that only allow canisters to hold up to 4GB of data. This is where the network's architecture steps in and allows for scalability through the front and back end canisters.

Finally, the internet computer is controlled by the nervous network system. This is a decentralized tokenized governance system for the internet computer. "The NNS can perform tasks such as upgrading node machines to update the protocol or apply security fixes, tweaking economic parameters, or forming new subnet blockchains to increase network capacity, at any time. It runs within the Internet Computer's protocols, and is able to make these changes without interrupting the network's operation or breaking security." Through the NNS, users can stake ICP to create voting neurons, forming a sort of liquid democracy in the system. The system can "upgrade the protocol and software used by the node machines that host the network; it can create new subnet blockchains to increase network capacity; it can split subnets to divide their load; it can configure economic parameters that control how much must be paid by users for compute capacity; and, in extreme situations, it can freeze malicious canister software in order to protect the network; and many other things. The NNS works by accepting proposals, and deciding to adopt or reject them based on voting activity by "neurons" that network participants have created." Ultimately, the NNS controls the forking of new canisters, and through the use of BigMap enables the scalability of data bucket canisters.

WaterPark Overview

There is significant motivation for the development of WaterPark. Firstly the system generates a stablecoin against ICP as collateral. Unlike other systems, WaterPark offers borrowing of its SDR stablecoin against Ether at a 0% interest rate. This allows users to retain the upside potential of a volatile asset like Ethereum while still extracting value that can be used in smart contracts (or by turning the SDR to cash). Additionally, because SDR is a stablecoin, smart contracts built around the token are "stable," which is a feature that will become more valuable as DeFi continues to mature.

WaterPark also is more capital efficient than other systems. This is due to the relatively low liquidity ratio of 110%. This lower ratio means that risk-averse borrowers can still feel comfortable at a lower collateral-to-debt ratio than on other platforms, enabling better capital efficiency and ultimately a more WaterPark Ether market.

Finally, WaterPark's low liquidation ratio is enabled by it's liquidation process which differs from many other prominent platforms that essentially auction off the collateral. Rather than auctioning Ether, which could cause the protocol to sell the collateral at a significant discount, the current liquidation protocol means that the majority of the liquidated trovers are liquidated at a profit (that is then distributed to stability providers) as the value of the collateral is general liquidated at between 100-110% of the value of the outstanding debt.

WaterPark Functionality

WaterPark's functionality ultimately depends upon three classes of actors: users, stability providers, and redemption users. Users support the primary functionality of the WaterPark protocol. Users deposit stablecoins into individualized troves created for each user. These troves are overcollateralized at a rate of 110%, meaning that if they deposit $110 worth of Ethereum, they are able to withdraw $100 of the SDR stablecoin. They are able to close their trove by fully repaying their outstanding debt. However, users face issues if their troves have a liquidation triggered. Liquidations are triggered on troves when the users have a collateral ratio lower than 110%, or if their ratio is below 150% and the system collateral ratio falls drastically.

In the event of a liquidation, the trove is closed and the collateral is sent to the stability pool as well as instructions to burn all SDR outstanding by the trove from the stability pool. In effect the stability pool backs debt in the troves, and when users default the collateral is transferred to the stability pool while the debt is burned (to ensure that the value of the entire system remains constant). In the case that the stability pool does not have enough SDR to cover the outstanding debt, both the debt and collateral are sent to other troves on a pro rata basis. To ensure that the stability pool has enough SDR, stability providers are incentivized to deposit SDR into the trove in return for the liquidated collateral

Finally, the system needs to ensure the stability of it's SDR stablecoin. It does so through the assumptions that small price dislocations will vanish due to arbitrage. In the event that SDR is worth more than $1, users will borrow more SDR, resulting in an increase of supply of the coin. In the case that SDR is worth less than $1, WaterPark has a redemption feature where individuals can deposit SDR tokens and receive the equivalent amount in ethereum (as if the SDR was worth $1), resulting in a reduction in supply as the redeemed SDR is burned.

Liquity Business Model

Individuals can stake LQTY or deposit SDR into the stability pool. The LQTY coin is not a governance coin, as Liquity is completely immutable and therefore does not require governance. When users deposit SDR into the stability pool they are rewarded with LQTY. When troves are overcollateralized, which is generally the case as troves are liquidated at a ratio above 100%, the system turns a profit, as the collateral seized is greater than the SDR burned. This Ether is then distributed pro rata to LQTY holders. Therefore, LQTY can be viewed as a way for individuals to gain access to the cash flows from Liquity.

However, individuals who utilize the redemption feature of the protocol can also generate returns as they are only incentivized to perform arbitrage when the price of SDR is above $1, in this case they keep the difference between the price of the redeemed collateral and the SDR deposited (minus a redemption fee).

Additionally, there are borrowing fees enacted when users add debt to a Trove, while the protocol has no interest rates, there is a fee applied when the user is given the SDR (meaning that the SDR received is a bit less than what is to be expected).

Finally, those that provide a web interface for the protocol (Frontend Operators) are entitled to a share of the LQTY that the users they bring generate. Of course, in our new system on the Internet Computer, this will not exist since the Internet Computer allows for the development of the front end directly onto IC Canisters.

Aims

This project aims to explore three main questions:

1. How does the use of orthogonal persistence when developing on Dfinity's Internet Computer affect the scalability of Waterpark as compared to Liquity?
2. How does orthogonal persistence affect the storage costs associated with the data collected by WaterPark and Liquity
3. How does the canister limit of 4gb affect the latency of the system as more users are added (additionally, does use of BigMap and NNS effectively overcome this obstacle)?

Ultimately, this proposal explores the efficiency and scalability of Dfinity's implementation of orthogonal persistence through the lens of the WaterPark protocol - measured via data storage costs.

Methodology

The methodology for this is fairly straightforward. The WaterPark protocol has been built utilizing the Internet Computer's default orthogonal persistence implementation. Here we propose building differing versions of WaterPark utilizing different techniques on both the Ethereum and Internet Computer networks. These would include implementations on the Internet Computer with and without BigMap, and on Ethereum via storage directly on the network, storage on a centralized storage solution with blockchain hashes, and on permissioned blockchains.

These implementations would be compared against each other by simulating the same growth of the system (users, redemptions, stability users, stablecoin etc.) and measuring the efficiency of data storage on these systems via the gas fees paid.

Of course, the logical question arises of accidentally misrepresenting the true efficiencies by including the costs of purely computational tasks (sorting, transferring of tokens, etc.). However, this worry is easily quelled due to the fact that, because the networks provide a continuous stream of gas fees / cycle burns, we are able to isolate fees associated solely with tasks related to writing / reading to storage. This allows us to measure the efficiency of these solutions with respect to only the data storage implementations.

Finally, there is the concern of unfeasibly large data storage costs associated with storing data on Ethereum. For this particular solution (and potentially for others), when a particular relationship between the amount of data stored / read and the gas fees has been determined (say for example a linear or quadratic memory to cost relationship), further results can be extrapolated from initial results.

Intellectual Merit

There is significant intellectual merit to this line of research. If copies of WaterPark is successfully deployed utilizing these 4 different types of storage solutions as well as on a centralized network, analyzing the cost implications associated with these solutions will be extremely influential in understanding the true efficiency of Dfinity's implementation of Orthogonal Persistence or current Ethereum solutions such as IPFS. To ensure that performance gains are coming solely from the data storage solution, versions of WaterPark will only be created on the Ethereum network, and information collected from Dfinity will be used to replicate the same implementation of orthogonal persistence. This way, the fault of the network that these protocols are built on does not impact the scalability of the system. Furthermore, this allows us to make valid comparisons between protocols when looking at Gas Fees, because gas is paid in ETH for all protocols and the gas fee model is static across all implementations. This is incredibly important as it can contribute to future literature and development in the field by guiding developers towards implementing their data storage in a manner that is most efficient on a blockchain level, both in terms of cost and scalability.