Decentralized Cloud Storage

Putting Data in the Cloud without Losing Control

Introduction

- David Vorick, CEO + Blockchain Expert
- Bitcoin enthusiast since 2011
- Full time Blockchain Engineer since 2014

- Co-founded Sia in 2014
- Sia is a decentralized cloud storage platform, and the subject of today's talk

Goal: Eliminate Failure Modes

+ The Amazon S3 outage highlights something important: lots of our infrastructure is on systems with single points of failure
+ Even if Amazon solves the technical challenges, they are a political point of control. Amazon is a US company, under US control, beholden to US state interests.
+ Amazon controls their prices, controls their terms of service, has the ability to pull the plug on you at any time, can refuse to support you / serve you, can prevent you from migrating
+ With modern cloud systems, **you give control** to the cloud provider.

+ We can do better.

Claim: Better Across the Board

+ Lower latency
+ Higher throughput
+ Less downtime
+ Better resistance to major black swan events - natural disaster, war, government intervention, etc.
+ Lower cost
+ Better overall security
+ All while allowing the user/uploader to maintain full control

Core Strategy

+ JBOD, but with the cloud - use a bunch of cheap, untrusted hosts in a heavily redundant scheme to achieve a competitive service
+ Use encryption to protect sensitive data against random hosts
+ Use Reed-Solomon coding to stretch redundancy as far as possible
+ Use Blockchain smart contracts to incentivize reliability and penalize unreliability
+ Continuously monitor hosts and pick only the most stable and competitive. Replace any hosts that go offline quickly

File Contracts

+ The core use of blockchain
+ A renter will create a contract with a host. The renter puts money in the contract to pay the host, the host puts money into the contract as a promise to be reliable.
+ The contract specifies the Merkle root of some data which the host must store, and a time period that the host must store the data for
+ The blockchain will hold the money in escrow until the time period has passed
+ The host will provide a proof-of-storage to the blockchain. If the proof is valid + successful, the host gets paid. If the proof is invalid, the host loses both the renter's money and their own collateral

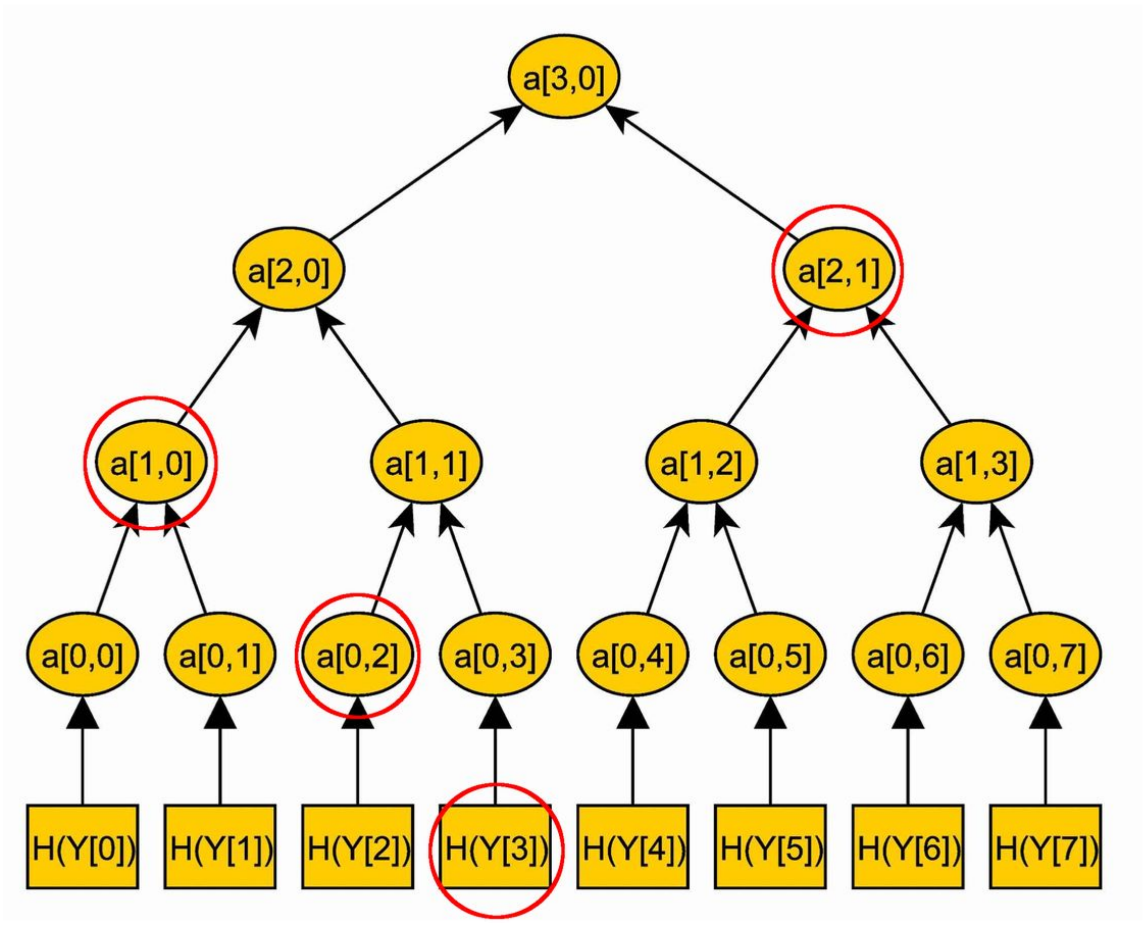
File Contracts are a Cryptographic SLA

+ If the host loses the data, they lose money
+ If the host keeps the data, they are guaranteed to receive the money
+ Once the contract has been formed, the host is not dependent on the renter being online to get paid. The host gets paid even if the renter disappears
+ This all happens without the use of a third party for escrow. The blockchain enforces the contract automatically, meaning there is no need for any trust

+ This also means no legal contracts, no lawsuits, no courts, no bureaucratic overhead. It’s a much cleaner, more efficient (albeit more limit) SLA

Storage Proof - Merkle Trees

+ The Merkle root of a file is gathered by splitting the file into 64-byte pieces, hashing each piece, and then repeatedly combining adjacent hashes (like a tree) until only one hash remains. That final hash is the Merkle root.
+ The file contract contains just the Merkle root of the data.
+ The host is asked to prove over a single 64 byte segment of the data. The host must provide that segment, and then must prove that the segment resides in the Merkle tree


a[3,0]
a[2,0]
a[2,1]
a[1,0]
a[1,1]
a[1,2]
a[1,3]
a[0,0]
a[0,1]
a[0,2]
a[0,3]
a[0,4]
a[0,5]
a[0,6]
a[0,7]
H(Y[0])
H(Y[1])
H(Y[2])
H(Y[3])
H(Y[4])
H(Y[5])
H(Y[6])
H(Y[7])

File Contract Game Theory

+ The blockchain randomly selects a single 64 byte piece for the host to prove. The host does not know which piece will be selected until the contract expires, preventing precomputation
+ The host proves possession of this single piece, and this is used as a proxy to check if the host is storing all the data
+ Failure has a strongly negative outcome - revenue and collateral are both forfeit
+ The expected value of cheating is negative. Cheating a tiny bit decreases costs a tiny bit, but risks huge penalties. Expected risks exceed expected cost reductions as long as the host is risking enough collateral

Reed-Solomon Redundancy

- 10-of-30 is probabilistically far, far, far superior to 1-of-3.
- Assuming each host is 90% reliable (and independent failures), a 1-of-3 scheme has a 1-in-1000 failure rate. A 10-of-30 scheme has a 1-in-1,000,000,000,000,000 failure rate.
- Hosts are owned by different people, running on different operating systems, and running on different continents. Independence of failure rate is much improved vs. traditional redundancy schemes.
- Protocol supports arbitrary redundancy schemes, meaning the right redundancy can be chosen to support each need.

Encryption

+ Applied before data ever leaves the client machine
+ Applied post-redundancy to prevent hosts from collaborating to deduplicate and reduce redundancy
+ Each piece has a separate password, derived from a master password

Monitoring

+ Each host is continually monitored, measuring things like uptime, latency, throughput, price.
+ If a host’s quality degrades relative to other potential hosts, that host can be replaced as though it has gone offline altogether
+ Each renter does monitoring separately. Malicious hosts have motivation to make fake renters and share results that favor them. Sharing is disabled until a secure solution is found.
+ Separate monitoring seems to be sufficient in practice, we are not actively looking for a solution

Competition

+ Hosts are in open competition. Each host has unique traits - geography, speed, latency, price, uptime, etc.
+ Renters have full freedom to select the hosts best suited to their problems
+ Due to the availability of high parallelism, default is to prefer price. This causes heavy downward price pressure on the network.
+ Using a 10-of-30 scheme, prices are currently $1.75 / TB / Mo. Raw storage is less than $0.25 / TB / Mo. on some hosts in our network.
+ Bandwidth price is similarly attractive - $1 / TB / Mo.

Architecture Overview

+ We use a blockchain with a cryptocurrency
+ Host announce themselves on the blockchain for easy and permanent discoverability
+ Renters engage hosts individually using file contracts. All payments are made with the cryptocurrency
+ File contracts + encryption give strong incentives for hosts to be reliable, and enables renters to interact with otherwise untrusted hosts
+ Broad redundancy is used is protect against unreliable hosts. If redundancy on a file falls too far, new hosts are contracted and redundancy is restored
+ Hosts are prioritized by feature, and are in constant competition with each other. Custom settings allow tuning for a broad range of use cases.

Blockchains Don’t Scale?

+ Sia blockchain limited to about 5 on-chain transactions per second
+ This is not an issue. The vast, vast majority of payments for storage and bandwidth occur in payment channels
+ Payment channels are a fully secure alternative to on-chain transactions that require some setup, and require locking up some coins for a few weeks
+ Can easily handle huge amounts of data. The biggest bottleneck is the storage proof, which is sized log(n) in the size of the data. 10^80 bytes is easily supported.
+ Maxes out around 100,000 users today. An enterprise user and a consumer are about equally expensive. Data volume is not relevant.
+ Solutions to the 100,000 user problem are available, but the state-of-the-art is improving. We are waiting to hit that scale before worrying.

Security - Data Withholding + Price Gouging

+ If a host attempts to hold data hostage, fall back to other hosts
+ Hosts are paid for bandwidth, so holding data hostage only makes sense if there is not enough redundancy to ignore the host
+ Interpreted by the software as a host with degraded quality. Host will be replaced if the host is frequently too expensive, offline, or otherwise un-ideal

+ Redundancy is high enough that data withholding and price gouging really only hurts the host attempting the attack

Security - Deduplication + Relocation

+ Each redundant piece is encrypted with a separate encryption key. Deduplication is impossible, meaning physical redundancy is at least guaranteed
+ Hosts are often valuable due to latency or geographic location. Data can be verified to be in the expected location by regularly downloading a small amount of data from the host and verifying that the latency matches the expected latency. Ping from multiple locations to triangulate the geography (if you want to be fancy)
+ Failure to meet expectation is considered a service degradation, host is at risk of being pruned just like in other attacks

Security - Refund Attacks

+ When hosts fail to provide a storage proof, the renter is not refunded.
+ If the renter was refunded, the renter would have explicit motivation to prevent their hosts from submitting storage proofs - ‘refund attacks’
+ Refund attacks are avoided by making sure the renter has no incentives to see the hosts fail

Security - Sybil Attacks

+ One host may try to get an advantage by pretending to be 1,000,000 hosts. Then, when selecting hosts, you see 1,000,090 hosts on the network, and 30 out of 30 of the hosts you choose belong to the attacker
+ The attacker really doesn't have this capacity, and is just attempting to be malicious
+ This can be stopped via proof-of-burn. Legitimate hosts prove legitimacy by burning coins. Burned coins are eventually recouped through revenue. You burn coins proportional to your expected revenue.
+ Renters give hosts a linear multiplier based on how many coins they burn. A host that burns 2x as many coins is 2x as likely to be selected.
+ A sybil attacker pretending to be half the network needs to burn as many coins as the rest of the network combined

Security - Hydra Attacks

+ Like a Sybil attack, except the attacker actually has enough hardware to dwarf the rest of the network combined.
+ Automated solutions may work if the renter is controlling for geography
+ Otherwise, manual intervention is needed
+ Getting a giant new host on the network is generally a good thing, but to preserve decentralization, that host can be manually blacklisted (just partially) until the network grows enough that the host is not such a mammoth by comparison
+ Decreasingly a concern as the network grows

Good for Hosts

+ Hosts are measured purely by infrastructural merit. No need to spend money on marketing, advertising, no need to build up a brand.
+ Hosts can join Sia at small scales and still win contracts
+ Sia automatically does not trust hosts beyond 98% reliability. There is little advantage to being more than 95% reliable, and no advantage to being more than 98% reliable. This means **hours of downtime per week is acceptable**!
+ To be an effective host, you must understand the burn mechanics and the collateral mechanics. But ultimately, it is far easier to be a competitive host on Sia than in the broader market.
+ Decreased barrier to entry means stronger competition and a better overall network

Continuously Competitive

+ New hosts can join the network at any time
+ Renters can easily prune old hosts and add new hosts as the competitive landscape changes
+ Price of storage as a result is fully independent of coin price
+ Price of storage as a result smoothly tracks the real cost of storage - unlike the step functions that traditional cloud providers use to set the price
+ Hosts can update prices in real-time, meaning that prices can be used to signal load and get the network to perform natural load-balancing (both for storage and for bandwidth)
+ Pay for only what you use

The Grand Plan

+ A decentralized replacement for Amazon S3
+ High parallelism means high throughput, even if the average host is only moderately fast
+ Redundancy settings + host choice can be tuned to favor extremely low latency
+ Open access to a global network of hosts is a great foundation for a CDN
+ Decentralization provides strong peace-of-mind

- Cloud compute not supported (encrypting data is easy, encrypting computation is hard, but an active area of study)

Sia Today

- Launched in June, 2015
- 90 hosts in 30 countries and 4 continents
- 300 mbps upload speeds, 50 mbps download speeds
- Scales to ~50 GB files, and multiple TB per node
- Access times around 30 seconds

- Limitations are largely due to unoptimized software. Significant improvements were made over the past 6 months, with significant improvements in the pipeline

Sia Today

+ $1.75 / TB / Mo for data storage (at 3x redundancy)
+ $1 / TB for downloads (un-optimized)

+ 200 TB of available storage on the network
+ 15 TB of storage being used on the network
+ Community of hundreds

+ > 1000 TB offline, waiting for demand to increase

Sia Today

+ Sia has a graphical client, a command line client, and a full-featured API.
+ We expect most users in the long run to be using the API, or to be using a third-party client that uses the API

+ Sia is fully decentralized. If my company shuts down tomorrow and all the devs quit, users will experience no disruption in service

+ Sia is fully open source. https://github.com/NebulousLabs/Sia

Questions