Properties of HotShot

HotShot's design intentionally aims to give chains fast confirmations to transactions, while being able to scale to a large number of participating nodes. However, the participating nodes do not execute transactions; hence, individual nodes only need assurance of data availability to vote in consensus, not to have full access to the data. This alleviates high hardware requirements for participation, without sacrificing throughput.

HotShot is based on the consensus techniques used within HotStuff and HotStuff-2.

For more details on HotShot and EspressoDA, see our post, *Designing the Espresso Network: Combining HotShot Consensus with Espresso's Data Availability Layer*

Key Properties of HotShot

Separating data availability (DA) and execution from consensus

The HotShot implementation is purpose-built for providing fast confirmations to a large number of generic chains. In particular, it does not perform execution, and the data availability requirement (i.e., ensuring the system has access to data) is handled by a separate DA solution (integrating chains can use the DA solution of their choice, but have default access to use our custom, low-cost DA layer, EspressoDA). This enables HotShot to process more data than typical state machine replication protocols. Such modularity also allows the use of various appropriate sub-protocols as needed.

Scalability

HotShot relies on all-to-leader and leader-to-all communication, thus reducing the consensus communication complexity to linear in the number of nodes. Since HotShot does not require every node to get a full copy of transaction data, low consensus communication is especially important. HotShot combines this optimistically with a content delivery network (CDN) to efficiently route data and perform computation. This reduces the leader bottleneck and supports a system with a heterogeneous set of nodes, without sacrificing safety and liveness guarantees. These improvements will help HotShot to scale to thousands of nodes, such that it can be run by a large number of Ethereum validators through restaking.

Responsiveness

HotShot is optimistically responsive and thus, under favorable conditions, commits new blocks as fast as the network allows. This ensures that the protocol's performance is directly related to the state of the network—under optimistic conditions, the protocol can have low latency and consequently high throughput, too. In HotShot, using a CDN at the network layer synergizes with the optimistic responsiveness property to provide even better performance.

EspressoDA

For a detailed technical overview of EspressoDA, see *EspressoDA: Our Three-Layered DA Solution*.

Data availability is the requirement that all transaction data included in blocks is available to every node participating in consensus before a decision can be reached. Because the amount of data can be quite large, requiring each node to download and verify the data before reaching

consensus presents a fundamental bottleneck in the throughput of consensus protocols (i.e., the data availability problem).

EspressoDA resolves this bottleneck while ensuring data availability via a three-layer system we've designed to balance performance and security:

- **VID Layer:** Stores erasure-coded data chunks across all nodes.
- **DA Committee Layer:** A small committee stores the full data and guarantees efficient recovery of data.
- **CDN Layer:** Uploads full data for retrieval efficiency.

VID Layer

EspressoDA eliminates the need for each storage node to download all block data by using Verifiable Information Dispersal (VID), a technique that encodes block data into erasure-coded chunks, which are disseminated among nodes in a way that recoverability is ensured. Nodes only need to store their chunk rather than the entire block. This method is more efficient than data availability sampling (DAS) as it limits unnecessary redundancies.

By using VID, EspressoDA guarantees a block will only be finalized if data is verified to be available.

DA Committee Layer

A small DA committee, selected from the network's nodes, receives the entire data blob and allows for very fast data retrievability, with the VID protocol acting as a fallback in case the DA committee fails to make data available.

EspressoDA ensures data is made available for rollups (optimistically by the DA committee, and guaranteed through VID) without incurring the high costs of posting transactions to the Ethereum L1 (though rollups may still choose to do so). It also avoids centralized DA solutions, which allow the DA operators to freeze the rollup and censor its users.

CDN Layer

We provide EspressoDA with web2-level performance by using a content delivery network (CDN) to quickly share a block's data to many different nodes. It can massively accelerate data dissemination. Benchmarks from our Cappuccino testnet show a data dissemination of around 5.7 MB/s with 100 nodes. The CDN can also help with efficient recovery of subsets of the data, such as single transactions.

Importantly, the CDN is not trusted for security and thus doesn't present a single point of failure. EspressoDA works perfectly fine without the CDN, which is only helpful for accelerating the DA and can easily be replaced or removed.

How It Works

A step-by-step guide on the data availability process

Overview

The data availability process initiates with sequencers submitting blocks to HotShot. Each view, a leader is selected within HotShot who bundles these blocks into a single block within HotShot. Rather than sending the full block data to other HotShot nodes, the leader only sends a commitment to the block for other nodes to vote on. All HotShot nodes also participate as storage nodes in the VID protocol and receive a small chunk representing their VID share of the respective block. The DA leader sends a DA proposal to the randomly sampled DA committee and all HotShot nodes. After receiving enough votes from the DA committee and the nodes in the network, the DA leader constructs a data availability certificate (DAC).

The DAC is composed of an optimistic DAC obtained from the DA committee and a retrievability certificate from the VID protocol. The optimistic DAC certifies that the proposed data is available to a quorum of the DA committee. The retrievability certificate in turn certifies that VID chunks are available to a quorum of nodes. The DAC design thus enables the best of both worlds, fast DA through DA committees, and robustness through VID. By combining the block commitment with the DAC, Tiramisu ensures that HotShot blocks will only finalized if data is guaranteed to be available.

Exhibit A: DA and Rollup Architecture

Process Steps

Step 1: The DA leader begins a broadcast to ensure data is available for all nodes in the network that consists of:

- Sending the DA proposal to the DA committee.
- Sending the VID chunks to all replicas/nodes.
- Gradually sending the DA proposal to all replicas/nodes. Broadcasting proceeds concurrently, prioritized by order of initialization.

A DA proposal will be rejected in either of the following cases:

- The view number is earlier than the view corresponding to the latest valid quorum certificate (QC).
- The proposal is not from the correct leader.

Step 2: Nodes receive the DA proposal (and/or VID share) and submit the data availability vote.

Anyone who receives and approves the DA proposal sends a strong DA vote to the DA leader, while anyone who only receives the VID share sends a normal DA vote.

Step 3: The DAC is formed.

The DAC is formed with the creation of the retrievability certificate and optimistic DAC certificate.

The retrievability certificate can be formed by:

- The DA leader receives f + 1 strong votes, which may come from nodes on the committee or just regular nodes who happened to receive the DA proposal quickly

enough. f = number of faulty nodes (nodes not performing correct function) in the entire network.

- f + m normal votes, where m is the number of VID shares the block was split into.

The optimistic DAC can be formed in the following way:

1. The DA leader receives 2f + 1 strong votes from the DA committee (where, unlike above, f is the number faulty nodes in the committee, not in the entire network)

The DA leader stops broadcasting to the nodes after the DAC is formed, or when the quorum certificate from the next leader is received.

Step 4: The block commitment proposal is sent to replicas/nodes. The block commitment is a a cryptographic proof that a block is valid and has guaranteed DA.

The leader, once getting sufficient quorum votes for the previous view and the DAC is obtained, sends the commitment proposal. A block can only be applied to a chain if consensus on the commitment proposal is reached by HotShot consensus nodes. The leader also sends the QC to the next leader if one can be constructed.

Step 5: The HotShot nodes validate the commitment proposal.

The replicas/nodes validate the commitment proposal if either of the following set is received:

- Commitment proposal and the DAC.
- Commitment proposal and DA proposal, which is simply a block and the view number proposed

The node will send a quorum vote to the next consensus leader once the commitment proposal is validated. As long as over 2/3 of HotShot stake is honest, it is impossible for an adversary, even if bribing the DA committee, to forge a DAC. Utilizing the randomly elected DA committee alongside VID thus enables fast and secure DA, and disincentivizes bribery attacks.

Exhibit B: DA Process Overview

Implementation Options

By integrating with the Espresso Network, chains can utilize EspressoDA without any additional integration work. However, chains retain the flexibility to use the DA solution of their choice (i.e., chains can use the Espresso Network for fast confirmations, but use Ethereum or another third-party provider for DA).