BNB Chain Foundation - OpBNB

Executive Summary

This audit report was prepared by Quantstamp, the leader in blockchain security.

Type	L2, Optimistic Rollup
Timeline	2023-08-07 through 2023-10-02
Language	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	opBNB audit gist opBNB doc website
Source Code	- bnb-chain/opbnb #1d70e51 - bnb-chain/op-geth #174353d
Auditors	- Andy Lin Senior Auditing Engineer - Guillermo Escobero Auditing Engineer - Adrian Koegl Auditing Engineer

Documentation quality	Medium
Test quality	Low
Total Findings	21 Fixed: 10 Acknowledged: 10 Mitigated: 1
High severity findings	0
Medium severity findings	3 Fixed: 2 Mitigated: 1
Low severity findings	13 Fixed: 5 Acknowledged: 8
Undetermined severity findings	0
Informational findings	5 Fixed: 3 Acknowledged: 2

Summary of Findings

Quantstamp conducted a detailed audit of opBNB's code changes, focusing primarily on the v0.1.2 releases. We also looked into potential challenges that may arise from deploying Optimism contracts on the BNB chain instead of Ethereum. Our examination of the contracts involved a review of the differences between the two chains, including consensus methods, block times, and variations in EIP implementations. This scrutiny aimed to uncover any potential vulnerabilities that might have been overlooked.

In our analysis of the v0.1.2 release code changes in the `opbnb` and `op-geth` repositories, we examined modified code from pull requests and commits and explored potential attack strategies. Notably, most changes were performance enhancements already implemented in the BNB chain. However, we observed some oversights during the audit. These included cases where features were ported with unused caches and where changes to hardcoded values were not consistently updated elsewhere. This emphasizes the importance of meticulous attention to detail during code modifications. While the code changes were generally understandable, bundling multiple changes within a single commit or pull request made categorizing them a challenge. We recommend adopting an approach of atomic commits for future code alterations.

Last, despite the fact that many of the identified issues have a low severity rating, we advise addressing all identified concerns to ensure robust code integrity.

Fix Review Update: The BNB team has addressed or acknowledged all reported issues. They have provided us with the PRs, which may still be open, for direct communication within the PRs. They will be responsible for merging them. The PRs for each fix are included in the issue update.

ID	DESCRIPTION	SEVERITY	STATUS
BNB-1	**Manipulate the Cost for Calldata**	● Medium	Mitigated
BNB-2	**Missing Usage of `accountTrieCache` and `storageTrieCache`**	● Medium	Fixed
BNB-3	**Race Condition Can Lead to Storing Invalid Blocks in DB**	● Medium	Fixed
BNB-4	**Feasibility of DoS Attack in Deposit Transactions**	● Low	Acknowledged

ID	DESCRIPTION	SEVERITY	STATUS
BNB-5	Delayed Transaction Re-Announcement Due to Nonce-Timestamp Ordering Discrepancy	Low	Acknowledged
BNB-6	Inconsistent L1.5 Cache Usage in `makeEnv()` Function	Low	Acknowledged
BNB-7	Inconsistency on Number of Snapshot Layers	Low	Fixed
BNB-8	Wrong Event Values	Low	Fixed
BNB-9	L1 Validators Exploit L2 by Reclaiming Burned Gas	Low	Acknowledged
BNB-10	Concurrent Prefetching Can Cause Block Data to Be Warmed up Incorrectly	Low	Acknowledged
BNB-11	Cache Inconsistency Between L1 and L1.5 Cache	Low	Acknowledged
BNB-12	Risk of Data Inconsistency Between Cache and DB Causing Inability to Retry	Low	Fixed
BNB-13	Risk of Caching Receipt That Failed to Derive Feilds	Low	Fixed
BNB-14	Network DoS Risk with the Re-announce Transaction Feature	Low	Acknowledged
BNB-15	Incorrect configuration would lead to pruning failure	Low	Acknowledged
BNB-16	Unhandled Error	Low	Fixed
BNB-17	Hardcoded Value For BSC Base Fee	Informational	Acknowledged
BNB-18	Duplicated Patch for Race Condition in Batcher	Informational	Acknowledged
BNB-19	Long-Running Tasks Occupying Go Pool Workers	Informational	Fixed
BNB-20	Redundant Lock	Informational	Fixed
BNB-21	Unnecessary `newRPCTransactionFromBlockHash()` Function	Informational	Fixed

Assessment Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

> **Disclaimer**
>
> This is a differential audit, so issues not related to the specific code changes are out of scope for this audit. Additionally, this audit assumes the correctness of the L1/L2 contracts for Optimism bedrock. Only issues related to the differences in L1 being different are within the scope of this audit.
>
> Only features that are contained within the repositories at the commit hashes specified on the front page of the report are within the scope of the audit and fix review. All features added in future revisions of the code are excluded from consideration in this report.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors

- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

1. Code review that includes the following
   1. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   2. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   3. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
   1. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   2. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarity, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Scope

The scope of this audit is limited to the code changes in the commits of the v0.1.2 release(s) and potential issues arising from deploying the forked Optimism L1/L2 contracts to the BNB chain.

The L1/L2 contracts can be found in the following paths:

1. L1 contract: https://github.com/bnb-chain/opbnb/tree/v0.1.2/packages/contracts-bedrock/contracts/L1
2. L2 contract: https://github.com/bnb-chain/opbnb/tree/v0.1.2/packages/contracts-bedrock/contracts/L2

The v0.1.2 release(s) changes are as follows:

op-bnb: https://github.com/bnb-chain/opbnb/releases/tag/v0.1.2

1. feat: support non-eip-1559 L1 & compatibility for BSC
2. feat: op-proposer propose currentL1Hash behavior config by AllowNonFinalizedFlag
3. feat: ResourceMetering.sol compatible with BSC
4. fix(op-batcher): solve race condition of BatchSubmitter

op-geth: https://github.com/bnb-chain/op-geth/releases/tag/v0.1.2

1. perf: concurrency and memory improvements for execution layer
2. perf: op-node related API improvement
3. feat: reannounce local pending transactions

Findings

BNB-1 Manipulate the Cost for Calldata

• Medium ⓘ Mitigated

Update

The team planned to address the issue in two phases. In the first phase, they will use a hardcoded gas cost instead of one derived from the transactions. The testnet will be set to 5 gwei, and the mainnet will be set to 3 gwei. The team will later modify the algorithm in the second phase, the specifics of which are currently undetermined.

The code change for the first phase has been completed in PR opbnb#65. According to the implementation, this change will take effect during the 'Fermat' upgrade. However, the specific block for the testnet and mainnet forking has not been set in the PR. The team will need to update the configuration in `op-node/chaincfg/chains.go`.

At the same time, we would like to highlight that hardcoding the gas could put the sequencer/batcher at risk of being unable to cover operational costs through the L2 gas fee. In situations where the L1 gas price increases, the hardcoded fee will be insufficient to pay for the L1 data availability submission. We have discussed this with the team, and they are aware of this risk.

The following is the original statement from the team:

> The gas price market on BSC differs from Ethereum. Unlike Ethereum, BSC does not have a base fee and the minimum gas price is determined by the validator who creates the block. Currently, the community has reached a consensus on a fixed

> value of 3 gwei. However, this price will increase as the TPS on BSC increases. In certain scenarios, some validators may accept a lower price, such as 1 gwei.
>
> We will take two steps to address this issue. First, we will use a fixed L1 gas price of 5 gwei for Testnet and 3 gwei for Mainnet in the short term. Second, we will modify the algorithm to establish the L1 gas price and set a minimum value through the L1 smart contract. This adjustment will ensure that the DA transaction can be successfully sent to L1.
>
> The PR for the first step has been merged bnb-chain/opbnb#65
>
> The second step will be scheduled during our next hardfork.

File(s) affected: `op-service/bsc/compact.go (opbnb)`, `L2/L1Block.sol`

Description: The `compact.go:BaseFeeByTransactions()` function calculates the base fee with the average transaction price of the L1 block. This base fee will eventually be set as the `basefee` of the `L1Block` contract. Subsequently, op-geth will use this value to compute the `l1Cost`, representing the calldata cost for L2 transactions.

However, there is a risk that attackers can inflate the result of `BaseFeeByTransactions()` by simply sending several dummy transactions with very high gas prices but using super low gas, thereby inflating the `BaseFeeByTransactions()` with low cost, as all transactions in the block have the same weight. Additionally, L1 validators can trigger the attack even more affordably, as (most of) the gas fee will be returned to them as part of the block reward.

Exploit Scenario: Here are some potential scenarios:

1. An attacker creates a dummy contract that does nothing, and the attacker sends lots of transactions calling the dummy contract with a high transaction gas price. The `l1Cost` eventually becomes too high for L2 users.
2. A malicious validator creates transactions with a high gas price and manipulates the L1 average gas price, rendering the L2 unusable as it might charge higher than L1 itself. The malicious validator can then collect most of the gas fee back from the block reward.

Recommendation: To mitigate this, consider incorporating the following into the `BaseFeeByTransactions()` function:

1. The average price should be weighted by the actual gas used. Instead of `sum(tx.gasprice) / txNumbers`, use `sum(tx.gasPrice * tx.receipt.gasUsed) / sum(tx.receipt.gasUsed)`.
2. Consider using the medium price or the TWAP price of the past N blocks to increase the cost of the attack.

BNB-2 Missing Usage of `accountTrieCache` and `storageTrieCache`

● Medium ⓘ Fixed

> **Update**
>
> The team fixed the issue as recommended in the PR op-geth#10.
>
> Please note that during our initial review of the fix, the team overlooked applying `Copy()` to the usage of `accountTrieCache` (see: this comment). The team has confirmed in the shared Telegram channel that they will make this change, so we are marking the issue as fixed with this assumption. We did not directly verify the final change on the PR.

File(s) affected: `core/state/database.go`

Description: The commit `f80e72bcd` (perf: concurrency and memory improvements for execution layer) introduced `accountTrieCache` and `storageTrieCache`. However, the only place these caches are utilized is when pruning the cache. In other words, account and storage tries are only cached and pruned, which seems to provide no value for performance. Upon cross-checking with the reference PR on the BNB chain (link), it appears that the team missed porting the change to `OpenTrie()` and `OpenStorageTrie()`. In the `bsc` repo, those functions use the caches.

We discussed this with the team during the audit, and they confirmed that they missed updating the `OpenTrie()` and `OpenStorageTrie()` functions.

The impact of the issue itself should be considered low according to our normal standards. However, it is evident that this behavior deviates from the intended functionality of the feature. As a result, we have decided to escalate its severity to medium.

Recommendation:

1. Conduct a comprehensive review of the codebase to pinpoint specific areas or operations where `accountTrieCache` and `storageTrieCache` need to be accessed.
2. Test the updated code to verify that the caching mechanisms function as expected and indeed contribute to performance improvements.
3. Refer to the implementation in the BSC client repository (PR#257) and incorporate the usage of the caches.

BNB-3 Race Condition Can Lead to Storing Invalid Blocks in DB

● Medium ⓘ Fixed

> **Update**
>
> The team fixed the issue in PR op-geth#14 as recommended.

File(s) affected: `core/block_validator.go`, `core/blockchain.go`

Description: The commit `f80e72bcd` (perf: concurrency and memory improvements for execution layer) parallelizes the block verification process. The function `block_validator.go:ValidateBody()` executes `validateFuns` concurrently using `gopool`. Under normal circumstances, `ValidateBody()` waits for all `validateFuns` to complete. However, if any of the `validateFuns` returns an error, `ValidateBody()` promptly returns with that error. Consequently, if more than one validation function returns an error, it can lead to non-deterministic selection of which error is returned.

After examining the flow and call sources of `ValidateBody()`, we discovered that certain errors, specifically `consensus.ErrUnknownAncestor` and `consensus.ErrPrunedAncestor`, are returned before any errors related to transactions or withdrawals. This sequence could cause the client database to store invalid blocks. To clarify with an example: If a block contains invalid transactions or withdrawals (meaning the hashes in the header don't match the expected calculations) and it also references an unrecognized parent block, there's a risk. If the third goroutine is the first to respond, the main calling function might wrongly assume the block's transactions and withdrawals are correct because it didn't detect the anticipated errors.

These errors receive special handling in `blockchain.go:insertChain()#L1721-1736`. For instance, the code attempts to insert the side-chain, recover the ancestor chain, or add it to a future block candidate. While most processes involve re-insertion of the block using the same function (`insertChain()`), a particularly concerning scenario occurs in the `blockchain.go:insertSideChain()` function, as it may call `bc.writeBlockWithoutState()` to write the invalid block data into the database.

This behavior contradicts the expected input of `insertSideChain()`:

```
The method writes all (header-and-body-valid) blocks to disk.
```

It is important to note that the state data is not saved as part of the `bc.writeBlockWithoutState()` function. Therefore, the impact appears to be limited to the invalid block (meta)data. Consequently, the severity of this issue is lower, as these blocks cannot become part of the canonical chain (they will be revalidated in `ValidateBody()` and no errors must be raised to be added to the chain). However, other Geth APIs may access this database and return an invalid block. Depending on external usage, this could potentially lead to a higher impact scenario.

Recommendation: In the `block_validator.go:ValidateBody()` function, instead of returning on the first seen error, collect all errors, and only return `ErrUnknownAncestor` or `ErrPrunedAncestor` error if other errors are not seen.

BNB-4 Feasibility of DoS Attack in Deposit Transactions

● Low ⓘ Acknowledged

> **Update**
>
> The team acknowledged the issue with the following statement:
>
> > The worst-case scenario is when an attacker front-runs a user's deposit transaction, exhausting the gas quota to revert the user's deposit. The attacker's cost exceeds the user's loss by over 100 times, and it will continue to increase if the attacker persists due to the presence of an EIP-1559-like price adjustment mechanism.
> >
> > Considering the aforementioned reasons, we will acknowledge and accept the associated risk.

File(s) affected: `L1/ResourceMetering.sol`

Description: A security issue has been identified within the L1 `ResourceMetering` contract, allowing attackers to exploit the resource allocation mechanism. This vulnerability enables malevolent actors to strategically deplete the entire pool of available resources allocated for deposit transactions within a specific block.

As a result, this resource depletion can cause all subsequent deposit transactions in that block to revert due to the unavailability of gas resources.

Upon analysis of on-chain values, it has been determined that on the Binance Smart Chain (BSC) mainnet, the maximum resource limit allocated for deposit transactions is capped at 80 million units. With a gas price of 3 Gwei on the BSC network, an attacker can execute an operation to fully consume all 80 million units, incurring a mere cost of $17.26 in BNB (assuming a BNB price of $216).

This issue was also found in the Optimism system. The mitigation was to include a configurable resource limit, and the team raised it from 8 million to 20 million units. Consequently, the cost of executing such an attack on the Ethereum network increased from $12.80 to $32 (with the Ethereum price at $1600).

All these calculations assume a `prevBaseFee` of `1e9` (`minimumBaseFee` set in both mainnet systems).

Here are some of the references where the issue was originally found in Optimism audit:

- https://github.com/sherlock-audit/2023-01-optimism-judging/issues/277
- https://github.com/sherlock-audit/2023-01-optimism-judging/issues/209
- https://github.com/sherlock-audit/2023-01-optimism-judging/issues/252

Recommendation: We are currently not aware of any easy fix for this issue. The current setup of 80M seems reasonably high, and increasing it could potentially help, but it also has its limits. The team should monitor the situation after launch and be able to react with the situation if anything unexpected occurs.

Meanwhile, one possible mitigation to consider is setting a practical limit, such as 50M, and beyond that, increasing the "virtual gas price" with a steep curve. In other words, the `ResourceMetering` contract should consider implementing a mechanism where each gas spent after reaching the threshold would result in a higher burn rate. This would serve as an economic deterrent against such an attack.

BNB-5 Delayed Transaction Re-Announcement Due to Nonce-Timestamp Ordering Discrepancy

● Low ⓘ Acknowledged

Update

The team clarified that this is acceptable.

File(s) affected: `core/txpool/txpool.go`, `eth/handler.go`

Description: The re-announcement mechanism ensures that transactions stuck in the pending state for an extended duration are re-propagated to peers, ultimately increasing the chances of inclusion in a mined block. The re-announcement mechanism will periodically review pending transactions and re-announce the transaction hash if the timestamp of the transaction exceeds a configurable threshold.

An interesting scenario arises when two transactions are sent by the same address, each with a different nonce and timestamp. However, the transaction with a higher nonce number somehow has an earlier timestamp.

For example, if `tx A` has a nonce of 3 and a time of 100, and `tx B` has a nonce of 2 and a time of 200, and the current time is 1000:

- `tx B` (nonce 2) has a `timeElapsed` of 800 (1000 - 200).
- `tx A` (nonce 3) has a `timeElapsed` of 900 (1000 - 100).

If a time threshold (e.g., 850) is set for re-announcement, `tx A` meets the criteria (`timeElapsed` > 850), but `tx B` does not. The loop will break when evaluating `tx B`, and `tx A` will not be re-announced until `tx B` meets the threshold, as the list is ordered by the nonce.

The timestamp and nonce discrepancy can happen when a user tries to prioritize the transaction and bump the gas price for `tx B`, causing the transaction to have a later timestamp but a lower nonce. The impact is that the re-announcement of transaction A will be delayed by transaction B.

Recommendation: We do not recommend naively re-announcing tx A in such a case as it could bring DOS concerns as tx A would not be processable without tx B.

Please clarify if the delay caused by the issue is acceptable. If acceptable, consider adding monitoring to the occurrence of this issue to enable data collection on the potential upgrade of the future system.

BNB-6 Inconsistent L1.5 Cache Usage in `makeEnv()` Function

● Low ⓘ Acknowledged

Alert

The team has communicated that they agree with the findings. However, since the code has been functioning well in the production environment and meets the team's performance objectives, they will maintain the current state for now. They will conduct further investigation and testing to address the issue or enable the feature in the future.

We have marked the issue's status as acknowledged, but we want to alert readers that the matter is still pending further action.

Description: An inconsistency has been identified in the `makeEnv()` function within the `worker.go` module. Specifically, the function initializes the state object with a call to `w.chain.StateAt(parent.Root)`, which does not initialize the L1.5 cache pool in the returned object. The subsequent invocation of `state.EnableWriteOnSharedStorage()` raises questions about the intended use of the L1.5 cache in this context. Consequently, functions using state objects returned by `makeEnv()` will not benefit from L1.5 cache improvements, rendering the call `newStatedb.EnableWriteOnSharedStorage()` in `PrefetchMining()` ineffective. The intention behind employing the L1.5 cache in these cases remains unclear based on the available information.

We have discussed the issue with the team during the audit, and they stated that worker commits consist mostly of new transactions, and caches are already present in the diff layer. Thus, enabling the 1.5-layer cache here does not add significant value.

Recommendation: Please clarify whether the L1.5 cache should be enabled or not. If there is no intention to utilize the cache, consider removing the potentially confusing line `newStatedb.EnableWriteOnSharedStorage()` in `PrefetchMining()`.

If the L1.5 cache is indeed intended for use, the code should ensure that it is appropriately set up and utilized. In this case, consider using the `StateAtWithSharedPool()` function to ensure that the L1.5 cache is actively engaged. Please refer to the approach followed in the BSC client repository.

BNB-7 Inconsistency on Number of Snapshot Layers

● Low ⓘ Fixed

Update

The team resolved the issue in PR op-geth#11. The `NewPruner()` function now accepts `CombinedOptions`, which includes both pruner options and snapshot options. The snapshot is set to the same value as `TrieInMemoryFlag`.

File(s) affected: `cmd/geth/snapshot.go` , `core/state/pruner/pruner.go`

Description: The opBNB team added a flag to configure the number of layers of the geth snapshots. By default, it is set to 128. The value can be changed when a new snapshot is created calling `snapshot.New()` . The caller needs to pass a setter function as a parameter to set this layer limit (`snapshot.SetCapLimit(int(triesInMemory))`).

All the snapshots created are set through the configured `triesInMemory` value, except for the one created when pruning the geth client in `pruner.go` :

```
snaptree, err := snapshot.New(snapconfig, db, trie.NewDatabase(db), headBlock.Root())
```

Instead of:

```
snaptree, err := snapshot.New(snapconfig, db, trie.NewDatabase(db), headBlock.Root(),
snapshot.SetCapLimit(int(triesInMemory)))
```

The `snaptree` variable will always have a layer limit of 128, instead of the configured value in `triesInMemory` flag.

Recommendation: Set the layers limit of the snapshot in `prunerNew()` to `triesInMemory` . Another approach is to always create all snapshots objects with the cap limit set to `triesInMemory` (modify `snapshot.New()`).

BNB-8 Wrong Event Values

• Low ⓘ Fixed

Update

The team fixed the issue as recommended in PR op-geth#16.

File(s) affected: `core/state/pruner/pruner.go`

Description: In `pruner.go` , at line `318` , a message with `Info` severity is logged:

```
log.Info("Selecting the bottom-most difflayer as the pruning target", "root", root, "height",
p.chainHeader.Number.Uint64()-127)
```

This message is not updated to reflect the optimization of `TriesInMemory` diff layers. Currently, the number is variable and not fixed at 128.

Recommendation: Consider modifying the message in function of `triesInMemory` value:

```
log.Info("Selecting bottom-most difflayer as the pruning target", "root", root, "height",
p.chainHeader.Number.Uint64()-int(p.triesInMemory))
```

BNB-9 L1 Validators Exploit L2 by Reclaiming Burned Gas

• Low ⓘ Acknowledged

Update

The team acknowledged the issue with the following statement:

> The OP Stack documentation states that there will be an option to send ETH to the Optimism Portal for purchasing L2 gas, but this feature is not currently supported. It is currently acceptable, and we can consider adopting this strategy if the OP Stack makes any changes.

File(s) affected: `L1/ResourceMetering.sol` , `libraries/Burn.sol`

Description: When a transaction moves from L1 to L2, the `ResourceMetering._metered()` function incurs a cost on L1 known as "burned" gas. The actual "burning" occurs in the `Burn.gas()` function, where the gas is effectively consumed. This security measure prevents potential DoS attacks from L1 users attempting to execute high gas-cost transactions on L2. However, L1 validators have the ability to reclaim this gas.

In Ethereum, post EIP-1559, the base fee is permanently burned, making it inaccessible even to validators. Furthermore, the original contract code determines the `gasCost` based on the `baseFee` , preventing validators from gaining economic advantage through such an attack.

In contrast, the BNB chain operates on a distinct mechanism. According to BEP-95 (documentation), the initial burn ratio stands at 10%. Consequently, OpBNB (L2) may have reduced protection against DoS attacks from BNB (L1) validators, as they only need to cover 10% of the cost.

Recommendation: Evaluate whether the risk is acceptable. If so, implement monitoring to identify validators engaged in malicious behavior, enabling the community to take appropriate action. Alternatively, consider modifying the design, either by "burning" through transferring tokens to `address(0)` or to the L2 sequencer fee collection address.

BNB-10 Concurrent Prefetching Can Cause Block Data to Be Warmed up Incorrectly

● Low ⓘ Acknowledged

> **Update**
> The team acknowledged the issue stating that the network remains functional in their pressure test scenario. Thus, the potential issue is deemed acceptable.

File(s) affected: `core/state_prefetcher.go`

Description: In `state_prefetcher.go`, both `Prefetch()` and `PrefetchMining()` simulate the transactions of a block concurrently. Instead of simulating the transactions one-by-one, there are `prefetchThread` amount of goroutines running the simulation job together, and they do not follow the order of the transactions. Since the prefetch does not rely on the simulated result, but simply aims to warm up the caches so the data can be moved from the DB to the caches.

When dependent transactions are processed out of order, the cache may end up containing incorrect state trie nodes or fail to warm up caches due to transaction reverts. This should be manageable as long as not too many dependent transactions are within the same block. Additionally, most applications typically wait for confirmations. However, it's important to note that an attacker could potentially exploit this by triggering a flood of dependent transactions, leading to a failure in the cache warm-up process.

Recommendation: Here are two improvement suggestions:

1. Verify that the network can still function in a worst-case scenario. The team should check that even in cases where prefetching is not working at all, the network will not break.
2. Consider mitigation plans, including reverting back to running the prefetch with a single goroutine, or alongside the concurrent prefetching, have one more goroutine to run all transactions one-by-one to ensure proper cache warm-up.

BNB-11 Cache Inconsistency Between L1 and L1.5 Cache

● Low ⓘ Acknowledged

> **Alert**
> The team has communicated that they agree with the findings. However, since the code has been functioning well in the production environment and meets the team's performance objectives, they will maintain the current state for now. They will conduct further investigation and testing to address the issue or enable the feature in the future.
>
> We have marked the issue's status as acknowledged, but we want to alert readers that the matter is still pending further action.

File(s) affected: `core/state/state_object.go`

Description: In the `state_object.go:updateTrie()` function, it sets the `originStorage` value (`s.originStorage[key] = value`) but not the `sharedOriginStorage`. Consequently, the shared pool (L1.5 cache) might contain inconsistent data compared to the L1 cache (`originStorage`). During the audit review, we did not identify an exploit resulting from this inconsistency, but it is also challenging to have full confidence due to the complexity of the code and the concurrent nature of Geth execution.

Recommendation: Please clarify if this is intended or not. If not, consider also updating the `sharedOriginStorage` as part of the `updateTrie()` function.

BNB-12 Risk of Data Inconsistency Between Cache and DB Causing Inability to Retry

● Low ⓘ Fixed

> **Update**
> The team fixed the issue as recommended in PR op-geth#17.

File(s) affected: `core/blockchain.go`, `eth/catalyst/api.go`

Description: In `blockchain.go`, both `CacheBlock()` and `CacheReceipts()` are called before the block is committed into the database in the `insertChain()` function. In the event that `bc.writeBlockWithState()` or `bc.writeBlockAndSetHead()` return an error, the block and receipt data might exist in the cache but not in the database.

Based on this rollup engine graph, it is evident that `newPayloadV1` is crucial for the rollup driver to trigger the op-geth block insertion. If the data is successfully inserted into the cache but fails to be written to the database, resulting in an error, the `newPayloadV1` API call will initially return an error. However, in a subsequent call (e.g. in a retry), the `newPayloadV1` will erroneously return a successful response since the `api.go:newPayload()` function will return an `engine.VALID` status when `api.eth.BlockChain().GetBlockByHash(params.BlockHash)` is not `nil`. This is because `GetBlockByHash()` first attempts to fetch the data from the cache, potentially leading to the retrieval of the block data.

This renders the system fragile, as the data solely resides in the cache and has not yet been written to the database. Additionally, new blocks might be built upon this cache-only block, and in the event of a node restart, it may necessitate re-syncing the data from where inconsistencies arise.

Recommendation: In `blockchain.go`, consider invoking `CacheBlock()` and `CacheReceipts()` only after successful execution of `bc.writeBlockWithState()` or `bc.writeBlockAndSetHead()`.

BNB-13 Risk of Caching Receipt That Failed to Derive Feilds

• Low ⓘ Fixed

Update

The team fixed the issue as recommended in PR op-geth#18.

File(s) affected: `core/blockchain.go`

Description: In `blockchain.go:insertChain()#L1873-1876`, the `types.Receipts(receipts).DeriveField()` call can potentially return an error. In the case that the `DeiveField()` fails, the `receipts` will not have the expected data attached to it. However, the code caches the block data (`CacheBlock()`) and the receipt data (`CacheReceipts()`) no matter whether the `DeriveField()` errors out or not in the `insertChain()` function. Incomplete or invalid data can be cached and risk causing unexpected scenario.

Recommendation: Consider bubbling up the error when `DeriveField()` fails. If the error should not be bubbled up, please re-check whether the block and receipts data should be cached in such a case.

BNB-14 Network DoS Risk with the Re-announce Transaction Feature

• Low ⓘ Acknowledged

Update

The team decided to disable the feature to prevent the DoS risk. Here is the original statement:

> The potential DoS Risk with re-announce feature does exist. This feature was introduced to address the problem of dropped transactions by the sequencer. In such cases, there is a possibility that the transaction will be re-announced and included in the next block. This situation is common across all blockchain networks, and users expect their pending transactions to be dropped occasionally. To resolve this, users can send a new transaction with a higher gas price and the same nonce to replace the dropped one. So we decide to disable the feature to prevent the DoS risk.

File(s) affected: `core/txpool/txpool.go`, `eth/handler.go`, `core/types/transaction.go`, `cmd/utils/flags.go`

Description: The commit `174353d` introduces the feature of periodically re-announcing local pending transactions. This means that the transaction hashes of local pending transactions will be broadcasted multiple times across the network, with the frequency currently set at one minute, if the transaction remains pending for longer than the configured `ReannounceTime`.

Given `N` nodes, for any pending transaction, the transaction hash will be transmitted `N * sqrt(N)` times throughout the network (assuming each node re-announces to `sqrt(N)` nodes) periodically. This process continues as long as the pending transaction remains unmined. Even if all nodes are already aware of the transaction, it will still generate this network load.

It's important to note that the implementation exclusively re-announces transactions in the `pool.pending` list. These are transactions that have been promoted from the `pool.queue`. In the promotion process, transactions that are non-executable are filtered out, preventing unprocessable transactions from triggering endless re-announcements.

However, in a congested network, this could exacerbate the situation as network nodes will also re-announce transaction hashes that cannot be mined due to the congestion, further slowing down the network.

More details on these considerations can be found here.

Recommendation: It's advisable to reassess the re-announcement mechanism to determine whether its benefits in terms of reliability and performance outweigh the newly introduced security vulnerabilities.

If the feature should be kept, please consider the following potential mitigations:

1. Introduce jitter to the re-announcement period. This will reduce the risk of multiple nodes re-announcing at the same time and causing a temporary network load spike.
2. Implement a mechanism similar to exponential backoff for re-announcing the same pending transaction, with progressively increasing wait times.

3. Introduce a throttle mechanism for the re-announcement process. If the same pending transaction has been re-announced too many times, it's likely that peers are already aware of it, and further re-announcements can be stopped.

BNB-15 Incorrect configuration would lead to pruning failure

● Low ⓘ Acknowledged

Update

Since the cap limit is currently the same as the tries in memory, the team has deemed the validations unnecessary for now, though this may change in the future. Here is the original statement from the team:

> Currently, the capLimit is equal to triesInMemory, so we don't need to do this check in our code. However, in the future, these two configurations may be separated, and if capLimit is allowed to be configured separately, we'll add the check then.

File(s) affected: `core/state/pruner/pruner.go` , `core/state/snapshot/snapshot.go`

Description: In `Prune()` of `pruner.go` , no pruning is performed if `len(layers) != int(p.triesInMemory)` . The intention is that pruning should only happen if the maximum depth is reached. However, if `capLimit` is set below `triesInMemory` through `SetCapLimit()` in `snapshot.go` , then pruning will always fail.

We have reviewed the code and currently, we do not find any instances where the `capLimit` is set to be below the `triesInMemory` . However, we would recommend embedding some validations to prevent future operational errors.

Recommendation: While there is currently no path that sets `capLimit` below `triesInMemory` , we recommend adding a safeguard to prevent pruning from failing.

This can be achieved by adding a requirement to `SetCapLimit()` to prevent setting `capLimit` below `triesInMemory` .

BNB-16 Unhandled Error

● Low ⓘ Fixed

Update

The team fixed the issue as recommended in PR op-geth#19.

File(s) affected: `core/state_prefetcher.go` , `gopool/pool.go` , `core/blockchain.go`

Description: There are several places not handling returning error. The following is a list of places we found during the audit and also the gosec analysis that are related to our audit scope:

1. In `core/state_prefetcher.go:Prefetch()` function, the error of the `precacheTransaction()` call is not handled.
2. The `gopool/pool.go:Submit()` function returns an error, but none of the callers handle it. Based on our analysis, since the `defaultPool` is not created with the `Nonblocking` flag, the underlying call `defaultPool.Submit()` does not expect an error. In this case, it is recommended to consider changing the function signature of `pool.go:Submit()` to not return an error and instead log an error message within the function.
3. In `core/blockchain.go:writeBlockWithState()#L1440` , the error is not handled in the call of `bc.triedb.Commit()` .
4. In `core/blockchain.go:writeBlockWithState()#L1421` function, the error is not handled in the call of `bc.triedb.Cap()` .

Recommendation: Please consider handling the errors listed in the description section. At a minimum, logging should be implemented to capture these errors.

BNB-17 Hardcoded Value For BSC Base Fee

● Informational ⓘ Acknowledged

Update

The team acknowledge the issue stating that it is acceptable for now and they will use a smart contract to control the value in the future.

File(s) affected: `L1/ResourceMetering.sol`

Description: The opBNB team modified the L1 contract `ResourceMetering.sol` to make it compatible with the Binance Smart Chain (BSC).

This is not compatible with the original Optimism code of `ResourceMetering` contract. This contract implements a "gas fee toll" when transactions are bridged from L1 to L2 (deposit transactions) through Optimism Portal. It implements an approach similar to EIP-1559: if the requested gas passes a threshold, the price will go up, and if the gas is below the threshold, the price will go down.

This amount is converted to L1 gas, that will be burned. This is the "payment" done by the user to get the transaction executed. In practice, it is like the user is buying L2 gas, by burning L1 gas.

In the original code this amount is calculated:

```
uint256 gasCost = resourceCost / Math.max(block.basefee, 1 gwei);
```

The BSC team changed it to:

```
uint256 gasCost = resourceCost / 3 gwei;
```

BSC does not implement the EIP-1559 proposal, so `BASEFEE` opcode is set to always return 0. Gas price in BSC is set to 3 gwei, following the legacy fee system (pre-EIP-1559).

Historically, the gas price on the BNB chain has undergone changes. For instance, it was initially set at 5 Gwei and later reduced to 3 Gwei. If the value undergoes further changes, the contract will require an upgrade.

Recommendation: Although the output of this calculation is just an approximation, if the gas price changes in L1 (i.e. BSC), the hardcoded value should be modified in `ResourceMetering.sol` to measure the resource cost as precisely as possible. Please consider adding a setter function to allow for value changes without the necessity of a complete contract upgrade.

BNB-18 Duplicated Patch for Race Condition in Batcher

● Informational ⓘ Acknowledged

> **Update**
> The team will revert the previous commit once they merged with the latest Optimism in the future.

File(s) affected: `op-batcher/batcher/driver.go (opbnb)`

Description: The opBNB team suggested a fix in the Optimism main repository (PR#6093) for a race condition found in `BatchSubmitter` module, that caused a crash. While it fixes the bug, Optimism team merged a patch (PR#6878) making the "parent" module, ChannelManager, safe for concurrent use.

From our review, we believe Optimism's fix is slightly more elegant. Although it's not included in the v0.1.2 release yet, it seems that Optimism's patch is also integrated into the later version of opbnb. This results in redundant patches and could potentially add complexity to code maintenance.

Recommendation: Consider implementing the fix proposed by Optimism (PR#6878). The scope of this fix affects to all operations from `ChannelManager`, and can prevent future bugs in other parts of the system.

BNB-19 Long-Running Tasks Occupying Go Pool Workers

● Informational ⓘ Fixed

> **Update**
> The team fixed the issue as recommended in PR op-geth#20.

File(s) affected: `p2p/dial.go`, `eth/bloombits.go`,

Description: One of the changes in commit `f80e72bcd` (perf: concurrency and memory improvements for execution layer) is to use `gopool` instead of spinning off a go-routine. The `gopool` limits the go-routines to run at a time, keeping the total go-routine memory in a manageable state.

However, for a long-running task, it will occupy a worker from the pool and never return, causing the effective pool size to decrease. The following are the places with long-running tasks:

1. `bloombits.go: startBloomHandlers()`: This function triggers a long-running goroutine that waits for the `eth.bloomRequests` and will only be closed when the Geth node is stopped (see: `backend.go:Stop()` function).
2. `dial.go: newDialScheduler()`: This function triggers two goroutines. Both `d.readNodes(it)` and `d.loop(it)` seem to be long-running tasks that will only be closed on the `Server` ends. The `server.go:run()` has the `defer srv.discmix.Close()` and `defer srv.dialsched.stop()` calls to trigger the context closing there.

Recommendation: We suggest refraining from using `gopool` for long-running tasks.

BNB-20 Redundant Lock

● Informational ⓘ Fixed

> **Update**
> The team removed the lock in PR op-geth#21

File(s) affected: `core/blockchain.go`

Description: In the `blockchain.go:writeBlockWithState()` function, the first function of the `postCommitFuncs` uses the `bc.commitLock`. However, this lock appears unnecessary, as it is only used for this function and the function concludes after `state.Commit()`. We have not identified any specific data that require this lock.

We discussed this with the team, and they also confirmed that this is problematic code, and the lock can be safely removed.

Recommendation: Clarify the necessity of the lock; consider removing it if it proves to be redundant.

BNB-21 Unnecessary `newRPCTransactionFromBlockHash()` Function

● Informational ⓘ Fixed

Update

The team fixed the issue as recommended in PR op-geth#22.

File(s) affected: `internal/ethapi/api.go`

Description: The commit `0d9dc40a` (`perf: op-node related api improvement`) introduced two new parameters, `idx int` and `tx *types.Transaction`, to the `api.go:newRPCTransactionFromBlockHash()` function. In the updated implementation, the `newRPCTransactionFromBlockHash()` function no longer uses the `hash` parameter, instead, it simply calls the `newRPCTransactionFromBlockIndex()` function. This renders the `newRPCTransactionFromBlockHash()` function redundant and confusing, as it is no longer related to "block hash".

Recommendation: We suggest removing the `newRPCTransactionFromBlockHash()` function and directly using `newRPCTransactionFromBlockIndex()` inside the `RPCMarshalBlock()` function. Additionally, consider reverting `newRPCTransactionFromBlockIndex()` to its previous code, eliminating the unnecessary complexity associated with the `if tx == nil {}` condition and the extra `tx` input.

Definitions

- **High severity** – High-severity issues usually put a large number of users' sensitive information at risk, or are reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
- **Medium severity** – Medium-severity issues tend to put a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or are reasonably likely to lead to moderate financial impact.
- **Low severity** – The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low impact in view of the client's business circumstances.
- **Informational** – The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
- **Undetermined** – The impact of the issue is uncertain.
- **Fixed** – Adjusted program implementation, requirements or constraints to eliminate the risk.
- **Mitigated** – Implemented actions to minimize the impact or likelihood of the risk.
- **Acknowledged** – The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).

Code Documentation

op-geth

1. In `blockchain.go:insertChain()#L1871`, there is a typo in the comment `// pre-cache the block and receipts, so that it can be retrieved quickly by rcp`. It might want to say "rpc" instead of "rcp".

Adherence to Best Practices

op-geth

1. In `core/state_prefetcher.go`, the go-routine in the `Prefetch()` function bails out whenever there is an error on `TransactionToMessage()`. However, the other concurrent go-routines will be still running despite stoppable. Consider add a mechanism to close all go-routines together whenever an error occurs and decides to bail out. Similar applies to the `PrefetchMining()` function on the error of `ToMessageNoNonceCheck()`.
2. In `core/state_prefetcher.go`, both the `Prefetch()` and `PrefetchMining()` functions could consider bailing out when `precacheTransaction()` returns error.

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:
- GoSec dev

Steps taken to run the tools:
- Install gosec: go get github.com/securego/gosec/cmd/gosec
- run gosec againt all modules: gosec ./...

Automated Analysis

GoSec

Most of the results are out of scope for this audit. We have included in the report only the findings that are related to our audit scope.

Test Suite Results

Here are the commands that we used to run the tests, and it appears that there are some test failures. We recommend the team to double-check those failed tests and address them accordingly.

op-geth:

```
make test
```

opbnb:

```
make test-unit
```

contract-bedrock: (inside the `contract-bedrock/` folder)

```
foundryup -C da2392e58bb8a7fefeba46b40c4df1afad8ccd22
yarn install
yarn test
```

And the test results are as follow:

Below is for **op-geth**

```
env GO111MODULE=on go run build/ci.go install
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go build -ldflags "-X github.com/ethereum/go-
ethereum/internal/version.gitCommit=174353d341e08fbec262cbb94dc72bad8f22a086 -X github.com/ethereum/go-
ethereum/internal/version.gitDate= -s" -tags urfave_cli_no_docs -trimpath -v -o
/Users/poanlin/Downloads/opbnb-audit/op-geth/build/bin/abidump ./cmd/abidump
github.com/ethereum/go-ethereum/cmd/abidump
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go build -ldflags "-X github.com/ethereum/go-
ethereum/internal/version.gitCommit=174353d341e08fbec262cbb94dc72bad8f22a086 -X github.com/ethereum/go-
ethereum/internal/version.gitDate= -s" -tags urfave_cli_no_docs -trimpath -v -o
/Users/poanlin/Downloads/opbnb-audit/op-geth/build/bin/abigen ./cmd/abigen
github.com/ethereum/go-ethereum/cmd/abigen
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go build -ldflags "-X github.com/ethereum/go-
ethereum/internal/version.gitCommit=174353d341e08fbec262cbb94dc72bad8f22a086 -X github.com/ethereum/go-
ethereum/internal/version.gitDate= -s" -tags urfave_cli_no_docs -trimpath -v -o
/Users/poanlin/Downloads/opbnb-audit/op-geth/build/bin/bootnode ./cmd/bootnode
github.com/ethereum/go-ethereum/cmd/bootnode
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go build -ldflags "-X github.com/ethereum/go-
ethereum/internal/version.gitCommit=174353d341e08fbec262cbb94dc72bad8f22a086 -X github.com/ethereum/go-
ethereum/internal/version.gitDate= -s" -tags urfave_cli_no_docs -trimpath -v -o
/Users/poanlin/Downloads/opbnb-audit/op-geth/build/bin/checkpoint-admin ./cmd/checkpoint-admin
github.com/ethereum/go-ethereum/cmd/checkpoint-admin
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go build -ldflags "-X github.com/ethereum/go-
ethereum/internal/version.gitCommit=174353d341e08fbec262cbb94dc72bad8f22a086 -X github.com/ethereum/go-
```

```
ethereum/internal/version.gitDate= -s" -tags urfave_cli_no_docs -trimpath -v -o
/Users/poanlin/Downloads/opbnb-audit/op-geth/build/bin/clef ./cmd/clef
# github.com/karalabe/usb
In file included from ../../../go/pkg/mod/github.com/karalabe/usb@v0.0.2/libs.go:50:
../../../go/pkg/mod/github.com/karalabe/usb@v0.0.2/libusb/libusb/os/darwin_usb.c:53:29: warning: macro
'ATOMIC_VAR_INIT' has been marked as deprecated [-Wdeprecated-pragma]
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/lib/clang/14.0.3/inclu
de/stdatomic.h:51:41: note: macro marked 'deprecated' here
github.com/ethereum/go-ethereum/cmd/clef
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go build -ldflags "-X github.com/ethereum/go-
ethereum/internal/version.gitCommit=174353d341e08fbec262cbb94dc72bad8f22a086 -X github.com/ethereum/go-
ethereum/internal/version.gitDate= -s" -tags urfave_cli_no_docs -trimpath -v -o
/Users/poanlin/Downloads/opbnb-audit/op-geth/build/bin/devp2p ./cmd/devp2p
github.com/ethereum/go-ethereum/cmd/devp2p
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go build -ldflags "-X github.com/ethereum/go-
ethereum/internal/version.gitCommit=174353d341e08fbec262cbb94dc72bad8f22a086 -X github.com/ethereum/go-
ethereum/internal/version.gitDate= -s" -tags urfave_cli_no_docs -trimpath -v -o
/Users/poanlin/Downloads/opbnb-audit/op-geth/build/bin/ethkey ./cmd/ethkey
github.com/ethereum/go-ethereum/cmd/ethkey
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go build -ldflags "-X github.com/ethereum/go-
ethereum/internal/version.gitCommit=174353d341e08fbec262cbb94dc72bad8f22a086 -X github.com/ethereum/go-
ethereum/internal/version.gitDate= -s" -tags urfave_cli_no_docs -trimpath -v -o
/Users/poanlin/Downloads/opbnb-audit/op-geth/build/bin/evm ./cmd/evm
github.com/ethereum/go-ethereum/cmd/evm
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go build -ldflags "-X github.com/ethereum/go-
ethereum/internal/version.gitCommit=174353d341e08fbec262cbb94dc72bad8f22a086 -X github.com/ethereum/go-
ethereum/internal/version.gitDate= -s" -tags urfave_cli_no_docs -trimpath -v -o
/Users/poanlin/Downloads/opbnb-audit/op-geth/build/bin/faucet ./cmd/faucet
github.com/ethereum/go-ethereum/cmd/faucet
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go build -ldflags "-X github.com/ethereum/go-
ethereum/internal/version.gitCommit=174353d341e08fbec262cbb94dc72bad8f22a086 -X github.com/ethereum/go-
ethereum/internal/version.gitDate= -s" -tags urfave_cli_no_docs -trimpath -v -o
/Users/poanlin/Downloads/opbnb-audit/op-geth/build/bin/geth ./cmd/geth
# github.com/karalabe/usb
In file included from ../../../go/pkg/mod/github.com/karalabe/usb@v0.0.2/libs.go:50:
../../../go/pkg/mod/github.com/karalabe/usb@v0.0.2/libusb/libusb/os/darwin_usb.c:53:29: warning: macro
'ATOMIC_VAR_INIT' has been marked as deprecated [-Wdeprecated-pragma]
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/lib/clang/14.0.3/inclu
de/stdatomic.h:51:41: note: macro marked 'deprecated' here
github.com/ethereum/go-ethereum/cmd/geth
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go build -ldflags "-X github.com/ethereum/go-
ethereum/internal/version.gitCommit=174353d341e08fbec262cbb94dc72bad8f22a086 -X github.com/ethereum/go-
ethereum/internal/version.gitDate= -s" -tags urfave_cli_no_docs -trimpath -v -o
/Users/poanlin/Downloads/opbnb-audit/op-geth/build/bin/p2psim ./cmd/p2psim
github.com/ethereum/go-ethereum/cmd/p2psim
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go build -ldflags "-X github.com/ethereum/go-
ethereum/internal/version.gitCommit=174353d341e08fbec262cbb94dc72bad8f22a086 -X github.com/ethereum/go-
ethereum/internal/version.gitDate= -s" -tags urfave_cli_no_docs -trimpath -v -o
/Users/poanlin/Downloads/opbnb-audit/op-geth/build/bin/rlpdump ./cmd/rlpdump
github.com/ethereum/go-ethereum/cmd/rlpdump
env GO111MODULE=on go run build/ci.go test
>>> /usr/local/Cellar/go/1.20.4/libexec/bin/go test -p 1 ./...
?       github.com/ethereum/go-ethereum [no test files]
ok      github.com/ethereum/go-ethereum/accounts    (cached)
ok      github.com/ethereum/go-ethereum/accounts/abi    (cached)
ok      github.com/ethereum/go-ethereum/accounts/abi/bind   11.924s
ok      github.com/ethereum/go-ethereum/accounts/abi/bind/backends  0.748s
?       github.com/ethereum/go-ethereum/accounts/external   [no test files]
ok      github.com/ethereum/go-ethereum/accounts/keystore   8.547s
?       github.com/ethereum/go-ethereum/accounts/scwallet   [no test files]
# github.com/karalabe/usb
In file included from ../../../go/pkg/mod/github.com/karalabe/usb@v0.0.2/libs.go:50:
../../../go/pkg/mod/github.com/karalabe/usb@v0.0.2/libusb/libusb/os/darwin_usb.c:53:29: warning: macro
'ATOMIC_VAR_INIT' has been marked as deprecated [-Wdeprecated-pragma]
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/lib/clang/14.0.3/inclu
de/stdatomic.h:51:41: note: macro marked 'deprecated' here
?       github.com/ethereum/go-ethereum/accounts/usbwallet  [no test files]
?       github.com/ethereum/go-ethereum/accounts/usbwallet/trezor   [no test files]
?       github.com/ethereum/go-ethereum/beacon/engine   [no test files]
?       github.com/ethereum/go-ethereum/cmd/abidump [no test files]
ok      github.com/ethereum/go-ethereum/cmd/abigen  0.473s
?       github.com/ethereum/go-ethereum/cmd/bootnode    [no test files]
?       github.com/ethereum/go-ethereum/cmd/checkpoint-admin    [no test files]
ok      github.com/ethereum/go-ethereum/cmd/clef    0.930s
ok      github.com/ethereum/go-ethereum/cmd/devp2p  0.774s
ok      github.com/ethereum/go-ethereum/cmd/devp2p/internal/ethtest 20.350s
?       github.com/ethereum/go-ethereum/cmd/devp2p/internal/v4test  [no test files]
?       github.com/ethereum/go-ethereum/cmd/devp2p/internal/v5test  [no test files]
ok      github.com/ethereum/go-ethereum/cmd/ethkey  0.647s
ok      github.com/ethereum/go-ethereum/cmd/evm 1.428s
?       github.com/ethereum/go-ethereum/cmd/evm/internal/compiler   [no test files]
?       github.com/ethereum/go-ethereum/cmd/evm/internal/t8ntool    [no test files]
ok      github.com/ethereum/go-ethereum/cmd/faucet  0.396s
ok      github.com/ethereum/go-ethereum/cmd/geth    37.040s
?       github.com/ethereum/go-ethereum/cmd/p2psim  [no test files]
ok      github.com/ethereum/go-ethereum/cmd/rlpdump 0.255s
ok      github.com/ethereum/go-ethereum/cmd/utils   0.583s
ok      github.com/ethereum/go-ethereum/common  0.263s
ok      github.com/ethereum/go-ethereum/common/bitutil  0.282s
?       github.com/ethereum/go-ethereum/common/compiler [no test files]
ok      github.com/ethereum/go-ethereum/common/fdlimit  0.301s
?       github.com/ethereum/go-ethereum/common/gopool   [no test files]
ok      github.com/ethereum/go-ethereum/common/hexutil  0.654s
ok      github.com/ethereum/go-ethereum/common/lru  0.311s
ok      github.com/ethereum/go-ethereum/common/math 0.342s
ok      github.com/ethereum/go-ethereum/common/mclock   0.574s
ok      github.com/ethereum/go-ethereum/common/prque    2.679s
?       github.com/ethereum/go-ethereum/consensus   [no test files]
?       github.com/ethereum/go-ethereum/consensus/beacon    [no test files]
ok      github.com/ethereum/go-ethereum/consensus/clique    1.387s
ok      github.com/ethereum/go-ethereum/consensus/ethash    15.603s
ok      github.com/ethereum/go-ethereum/consensus/misc  0.445s
ok      github.com/ethereum/go-ethereum/console 4.717s
?       github.com/ethereum/go-ethereum/console/prompt  [no test files]
ok      github.com/ethereum/go-ethereum/contracts/checkpointoracle  1.876s
?       github.com/ethereum/go-ethereum/contracts/checkpointoracle/contract [no test files]
ok      github.com/ethereum/go-ethereum/core    143.498s
ok      github.com/ethereum/go-ethereum/core/asm    0.326s
ok      github.com/ethereum/go-ethereum/core/bloombits  0.837s
ok      github.com/ethereum/go-ethereum/core/forkid 0.249s
ok      github.com/ethereum/go-ethereum/core/rawdb  70.324s
ok      github.com/ethereum/go-ethereum/core/state  10.336s
?       github.com/ethereum/go-ethereum/core/state/pruner   [no test files]
ok      github.com/ethereum/go-ethereum/core/state/snapshot 4.470s
ok      github.com/ethereum/go-ethereum/core/txpool 15.905s
ok      github.com/ethereum/go-ethereum/core/types  2.337s
ok      github.com/ethereum/go-ethereum/core/vm 16.168s
ok      github.com/ethereum/go-ethereum/core/vm/runtime 2.674s
ok      github.com/ethereum/go-ethereum/crypto  0.480s
ok      github.com/ethereum/go-ethereum/crypto/blake2b  0.448s
ok      github.com/ethereum/go-ethereum/crypto/bls12381 8.024s
?       github.com/ethereum/go-ethereum/crypto/bn256    [no test files]
ok      github.com/ethereum/go-ethereum/crypto/bn256/cloudflare 0.951s
ok      github.com/ethereum/go-ethereum/crypto/bn256/google 3.515s
ok      github.com/ethereum/go-ethereum/crypto/ecies    0.608s
ok      github.com/ethereum/go-ethereum/crypto/secp256k1    5.241s
ok      github.com/ethereum/go-ethereum/crypto/signify  0.481s
ok      github.com/ethereum/go-ethereum/eth 12.971s
ok      github.com/ethereum/go-ethereum/eth/catalyst    4.387s
ok      github.com/ethereum/go-ethereum/eth/downloader  197.571s
?       github.com/ethereum/go-ethereum/eth/ethconfig   [no test files]
ok      github.com/ethereum/go-ethereum/eth/fetcher 7.709s
ok      github.com/ethereum/go-ethereum/eth/filters 1.743s
ok      github.com/ethereum/go-ethereum/eth/gasprice    0.650s
ok      github.com/ethereum/go-ethereum/eth/protocols/eth   0.902s
ok      github.com/ethereum/go-ethereum/eth/protocols/snap  11.749s
ok      github.com/ethereum/go-ethereum/eth/tracers 0.933s
ok      github.com/ethereum/go-ethereum/eth/tracers/internal/tracetest  0.539s
ok      github.com/ethereum/go-ethereum/eth/tracers/js  1.531s
?       github.com/ethereum/go-ethereum/eth/tracers/js/internal/tracers [no test files]
ok      github.com/ethereum/go-ethereum/eth/tracers/logger  0.359s
?       github.com/ethereum/go-ethereum/eth/tracers/native  [no test files]
ok      github.com/ethereum/go-ethereum/ethclient   0.607s
ok      github.com/ethereum/go-ethereum/ethclient/gethclient    0.584s
```

```
?       github.com/ethereum/go-ethereum/ethdb   [no test files]
?       github.com/ethereum/go-ethereum/ethdb/dbtest    [no test files]
ok      github.com/ethereum/go-ethereum/ethdb/leveldb   0.286s
ok      github.com/ethereum/go-ethereum/ethdb/memorydb  0.253s
ok      github.com/ethereum/go-ethereum/ethdb/pebble    0.357s
?       github.com/ethereum/go-ethereum/ethdb/remotedb  [no test files]
ok      github.com/ethereum/go-ethereum/ethstats    0.503s
ok      github.com/ethereum/go-ethereum/event   0.979s
ok      github.com/ethereum/go-ethereum/graphql 0.833s
?       github.com/ethereum/go-ethereum/internal/build  [no test files]
?       github.com/ethereum/go-ethereum/internal/cmdtest    [no test files]
?       github.com/ethereum/go-ethereum/internal/debug  [no test files]
ok      github.com/ethereum/go-ethereum/internal/ethapi 0.782s
ok      github.com/ethereum/go-ethereum/internal/flags  0.582s
ok      github.com/ethereum/go-ethereum/internal/guide  2.790s
ok      github.com/ethereum/go-ethereum/internal/jsre   0.462s
?       github.com/ethereum/go-ethereum/internal/jsre/deps  [no test files]
?       github.com/ethereum/go-ethereum/internal/shutdowncheck  [no test files]
?       github.com/ethereum/go-ethereum/internal/syncx  [no test files]
?       github.com/ethereum/go-ethereum/internal/testlog    [no test files]
ok      github.com/ethereum/go-ethereum/internal/utesting   0.257s
?       github.com/ethereum/go-ethereum/internal/version    [no test files]
?       github.com/ethereum/go-ethereum/internal/web3ext    [no test files]
ok      github.com/ethereum/go-ethereum/les 63.088s
ok      github.com/ethereum/go-ethereum/les/catalyst    0.463s
?       github.com/ethereum/go-ethereum/les/checkpointoracle    [no test files]
ok      github.com/ethereum/go-ethereum/les/downloader  13.913s
ok      github.com/ethereum/go-ethereum/les/fetcher 8.056s
ok      github.com/ethereum/go-ethereum/les/flowcontrol 9.643s
ok      github.com/ethereum/go-ethereum/les/utils   3.799s
?       github.com/ethereum/go-ethereum/les/vflux   [no test files]
ok      github.com/ethereum/go-ethereum/les/vflux/client    3.738s
ok      github.com/ethereum/go-ethereum/les/vflux/server    20.718s
ok      github.com/ethereum/go-ethereum/light   1.568s
ok      github.com/ethereum/go-ethereum/log 0.383s
ok      github.com/ethereum/go-ethereum/metrics 6.176s
?       github.com/ethereum/go-ethereum/metrics/exp [no test files]
?       github.com/ethereum/go-ethereum/metrics/influxdb    [no test files]
?       github.com/ethereum/go-ethereum/metrics/librato [no test files]
ok      github.com/ethereum/go-ethereum/metrics/prometheus  0.344s
ok      github.com/ethereum/go-ethereum/miner   14.504s
?       github.com/ethereum/go-ethereum/miner/stress/1559   [no test files]
?       github.com/ethereum/go-ethereum/miner/stress/beacon [no test files]
?       github.com/ethereum/go-ethereum/miner/stress/clique [no test files]
?       github.com/ethereum/go-ethereum/miner/stress/ethash [no test files]
ok      github.com/ethereum/go-ethereum/node    3.177s
ok      github.com/ethereum/go-ethereum/p2p 1.405s
ok      github.com/ethereum/go-ethereum/p2p/discover    64.642s
ok      github.com/ethereum/go-ethereum/p2p/discover/v4wire 0.310s
ok      github.com/ethereum/go-ethereum/p2p/discover/v5wire 0.325s
ok      github.com/ethereum/go-ethereum/p2p/dnsdisc 1.092s
ok      github.com/ethereum/go-ethereum/p2p/enode   1.805s
ok      github.com/ethereum/go-ethereum/p2p/enr 0.265s
ok      github.com/ethereum/go-ethereum/p2p/msgrate 0.258s
ok      github.com/ethereum/go-ethereum/p2p/nat 2.904s
ok      github.com/ethereum/go-ethereum/p2p/netutil 0.268s
ok      github.com/ethereum/go-ethereum/p2p/nodestate   0.833s
ok      github.com/ethereum/go-ethereum/p2p/rlpx    0.309s
ok      github.com/ethereum/go-ethereum/p2p/simulations 2.564s
ok      github.com/ethereum/go-ethereum/p2p/simulations/adapters    0.423s
?       github.com/ethereum/go-ethereum/p2p/simulations/examples    [no test files]
?       github.com/ethereum/go-ethereum/p2p/simulations/pipes   [no test files]
?       github.com/ethereum/go-ethereum/p2p/tracker [no test files]
ok      github.com/ethereum/go-ethereum/params  0.251s
ok      github.com/ethereum/go-ethereum/rlp 0.274s
?       github.com/ethereum/go-ethereum/rlp/internal/rlpstruct  [no test files]
ok      github.com/ethereum/go-ethereum/rlp/rlpgen  1.407s
ok      github.com/ethereum/go-ethereum/rpc 11.147s
ok      github.com/ethereum/go-ethereum/signer/core 5.828s
ok      github.com/ethereum/go-ethereum/signer/core/apitypes    0.266s
ok      github.com/ethereum/go-ethereum/signer/fourbyte 7.654s
ok      github.com/ethereum/go-ethereum/signer/rules    0.740s
ok      github.com/ethereum/go-ethereum/signer/storage  0.265s
ok      github.com/ethereum/go-ethereum/tests   1.286s
ok      github.com/ethereum/go-ethereum/tests/fuzzers/abi   0.342s
?       github.com/ethereum/go-ethereum/tests/fuzzers/bitutil   [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/bls12381  [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/difficulty    [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/difficulty/debug  [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/keystore  [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/les   [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/les/debug [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/rangeproof    [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/rangeproof/debug  [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/rlp   [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/runtime   [no test files]
ok      github.com/ethereum/go-ethereum/tests/fuzzers/secp256k1 0.335s
?       github.com/ethereum/go-ethereum/tests/fuzzers/snap  [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/snap/debug    [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/stacktrie [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/stacktrie/debug   [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/trie  [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/txfetcher [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/vflux [no test files]
?       github.com/ethereum/go-ethereum/tests/fuzzers/vflux/debug   [no test files]
ok      github.com/ethereum/go-ethereum/trie    32.426s
```

Below is for **opbnb**

```
make -C ./op-node test
go test -v ./...
?       github.com/ethereum-optimism/optimism/op-node   [no test files]
?       github.com/ethereum-optimism/optimism/op-node/chaincfg  [no test files]
=== RUN   TestPollingClientSubscribeUnsubscribe
--- PASS: TestPollingClientSubscribeUnsubscribe (0.03s)
=== RUN   TestPollingClientErrorRecovery
--- PASS: TestPollingClientErrorRecovery (0.00s)
=== RUN   TestPollingClientClose
--- PASS: TestPollingClientClose (0.00s)
PASS
ok      github.com/ethereum-optimism/optimism/op-node/client    0.488s
?       github.com/ethereum-optimism/optimism/op-node/cmd   [no test files]
?       github.com/ethereum-optimism/optimism/op-node/cmd/batch_decoder [no test files]
?       github.com/ethereum-optimism/optimism/op-node/cmd/batch_decoder/fetch   [no test files]
?       github.com/ethereum-optimism/optimism/op-node/cmd/batch_decoder/reassemble  [no test files]
?       github.com/ethereum-optimism/optimism/op-node/cmd/doc   [no test files]
?       github.com/ethereum-optimism/optimism/op-node/cmd/genesis   [no test files]
?       github.com/ethereum-optimism/optimism/op-node/cmd/stateviz  [no test files]
=== RUN   TestPrivPub2PeerID
=== RUN   TestPrivPub2PeerID/with_a_private_key
=== RUN   TestPrivPub2PeerID/with_a_public_key
=== RUN   TestPrivPub2PeerID/with_bad_hex
--- PASS: TestPrivPub2PeerID (0.01s)
    --- PASS: TestPrivPub2PeerID/with_a_private_key (0.00s)
    --- PASS: TestPrivPub2PeerID/with_a_public_key (0.00s)
    --- PASS: TestPrivPub2PeerID/with_bad_hex (0.00s)
PASS
ok      github.com/ethereum-optimism/optimism/op-node/cmd/p2p   0.310s
?       github.com/ethereum-optimism/optimism/op-node/http  [no test files]
?       github.com/ethereum-optimism/optimism/op-node/metrics   [no test files]
?       github.com/ethereum-optimism/optimism/op-node/p2p/cli   [no test files]
?       github.com/ethereum-optimism/optimism/op-node/p2p/mocks [no test files]
?       github.com/ethereum-optimism/optimism/op-node/rollup/derive/test    [no test files]
?       github.com/ethereum-optimism/optimism/op-node/sources/caching   [no test files]
?       github.com/ethereum-optimism/optimism/op-node/testlog   [no test files]
?       github.com/ethereum-optimism/optimism/op-node/testutils [no test files]
?       github.com/ethereum-optimism/optimism/op-node/testutils/fuzzerutils [no test files]
?       github.com/ethereum-optimism/optimism/op-node/version   [no test files]
=== RUN   TestAccountResult_Verify
--- PASS: TestAccountResult_Verify (0.00s)
=== RUN   TestOPB01
--- PASS: TestOPB01 (0.00s)
```

```
=== RUN   TestOPB04
--- PASS: TestOPB04 (48.63s)
=== RUN   TestInputError
--- PASS: TestInputError (0.00s)
=== RUN   FuzzAccountResult_StorageProof
--- PASS: FuzzAccountResult_StorageProof (0.00s)
=== RUN   FuzzAccountResult_AccountProof
--- PASS: FuzzAccountResult_AccountProof (0.00s)
=== RUN   FuzzExecutionPayloadUnmarshal
--- PASS: FuzzExecutionPayloadUnmarshal (0.00s)
=== RUN   FuzzExecutionPayloadMarshalUnmarshal
--- PASS: FuzzExecutionPayloadMarshalUnmarshal (0.00s)
=== RUN   FuzzOBP01
--- PASS: FuzzOBP01 (0.00s)
PASS
ok      github.com/ethereum-optimism/optimism/op-node/eth   49.831s
=== RUN   TestOptionalFlagsDontSetRequired
--- PASS: TestOptionalFlagsDontSetRequired (0.00s)
=== RUN   TestUniqueFlags
--- PASS: TestUniqueFlags (0.00s)
PASS
ok      github.com/ethereum-optimism/optimism/op-node/flags 0.907s
=== RUN   TestBeat
--- PASS: TestBeat (0.00s)
PASS
ok      github.com/ethereum-optimism/optimism/op-node/heartbeat 1.279s
=== RUN   TestUnixTimeStale
--- PASS: TestUnixTimeStale (0.00s)
=== RUN   TestOutputAtBlock
    server_test.go:130: PASS:    InfoByHash(common.Hash)
    server_test.go:130: PASS:    GetProof(common.Address,[]common.Hash,string)
    server_test.go:131: PASS:    BlockRefWithStatus(uint64)
--- PASS: TestOutputAtBlock (0.03s)
=== RUN   TestVersion
--- PASS: TestVersion (0.00s)
=== RUN   TestSyncStatus
--- PASS: TestSyncStatus (0.00s)
PASS
ok      github.com/ethereum-optimism/optimism/op-node/node  1.247s
=== RUN   TestBandScorer_ParseDefault
--- PASS: TestBandScorer_ParseDefault (0.00s)
=== RUN   TestBandScorer_BucketCorrectly
--- PASS: TestBandScorer_BucketCorrectly (0.00s)
=== RUN   TestBandScorer_BucketInverted
--- PASS: TestBandScorer_BucketInverted (0.00s)
=== RUN   TestBandScorer_ParseEmpty
--- PASS: TestBandScorer_ParseEmpty (0.00s)
=== RUN   TestBandScorer_ParseWhitespace
--- PASS: TestBandScorer_ParseWhitespace (0.00s)
=== RUN   TestGuardGossipValidator
--- PASS: TestGuardGossipValidator (0.00s)
=== RUN   TestVerifyBlockSignature
=== RUN   TestVerifyBlockSignature/Valid
=== RUN   TestVerifyBlockSignature/WrongSigner
=== RUN   TestVerifyBlockSignature/InvalidSignature
=== RUN   TestVerifyBlockSignature/NoSequencer
--- PASS: TestVerifyBlockSignature (0.04s)
    --- PASS: TestVerifyBlockSignature/Valid (0.00s)
    --- PASS: TestVerifyBlockSignature/WrongSigner (0.00s)
    --- PASS: TestVerifyBlockSignature/InvalidSignature (0.00s)
    --- PASS: TestVerifyBlockSignature/NoSequencer (0.00s)
=== RUN   TestP2PSimple
--- PASS: TestP2PSimple (0.05s)
=== RUN   TestP2PFull
--- PASS: TestP2PFull (0.09s)
=== RUN   TestDiscovery
--- PASS: TestDiscovery (5.11s)
=== RUN   TestP2PMocknet
--- PASS: TestP2PMocknet (0.03s)
=== RUN   TestPeerParams
=== RUN   TestPeerParams/TestAvailablePeerScoreParams
=== RUN   TestPeerParams/TestDisabledPeerScoreParams
=== RUN   TestPeerParams/TestGetPeerScoreParams
=== RUN   TestPeerParams/TestLightPeerScoreParams
=== RUN   TestPeerParams/TestNewPeerScoreThresholds
=== RUN   TestPeerParams/TestParamsZeroBlockTime
=== RUN   TestPeerParams/TestPeerScoreConstants
--- PASS: TestPeerParams (0.00s)
    --- PASS: TestPeerParams/TestAvailablePeerScoreParams (0.00s)
    --- PASS: TestPeerParams/TestDisabledPeerScoreParams (0.00s)
    --- PASS: TestPeerParams/TestGetPeerScoreParams (0.00s)
    --- PASS: TestPeerParams/TestLightPeerScoreParams (0.00s)
    --- PASS: TestPeerParams/TestNewPeerScoreThresholds (0.00s)
    --- PASS: TestPeerParams/TestParamsZeroBlockTime (0.00s)
    --- PASS: TestPeerParams/TestPeerScoreConstants (0.00s)
=== RUN   TestSigningHash_DifferentDomain
--- PASS: TestSigningHash_DifferentDomain (0.00s)
=== RUN   TestSigningHash_DifferentChainID
--- PASS: TestSigningHash_DifferentChainID (0.00s)
=== RUN   TestSigningHash_DifferentMessage
--- PASS: TestSigningHash_DifferentMessage (0.00s)
=== RUN   TestSigningHash_LimitChainID
--- PASS: TestSigningHash_LimitChainID (0.00s)
=== RUN   TestSinglePeerSync
=== PAUSE TestSinglePeerSync
=== RUN   TestMultiPeerSync
=== PAUSE TestMultiPeerSync
=== RUN   TestPeerGater
=== RUN   TestPeerGater/TestPeerGater_UpdateBansPeers
=== RUN   TestPeerGater/TestPeerGater_UpdateNoBanning
=== RUN   TestPeerGater/TestPeerScoreConstants
--- PASS: TestPeerGater (0.00s)
    --- PASS: TestPeerGater/TestPeerGater_UpdateBansPeers (0.00s)
    --- PASS: TestPeerGater/TestPeerGater_UpdateNoBanning (0.00s)
    --- PASS: TestPeerGater/TestPeerScoreConstants (0.00s)
=== RUN   TestPeerScorer
=== RUN   TestPeerScorer/TestScorer_OnConnect
=== RUN   TestPeerScorer/TestScorer_OnDisconnect
=== RUN   TestPeerScorer/TestScorer_SnapshotHook
=== RUN   TestPeerScorer/TestScorer_SnapshotHookBlocksPeer
--- PASS: TestPeerScorer (0.01s)
    --- PASS: TestPeerScorer/TestScorer_OnConnect (0.00s)
    --- PASS: TestPeerScorer/TestScorer_OnDisconnect (0.00s)
    --- PASS: TestPeerScorer/TestScorer_SnapshotHook (0.00s)
    --- PASS: TestPeerScorer/TestScorer_SnapshotHookBlocksPeer (0.00s)
=== RUN   TestPeerScores
=== RUN   TestPeerScores/TestNegativeScores
--- PASS: TestPeerScores (7.90s)
    --- PASS: TestPeerScores/TestNegativeScores (7.90s)
=== CONT  TestSinglePeerSync
=== CONT  TestMultiPeerSync
--- PASS: TestSinglePeerSync (0.03s)
--- PASS: TestMultiPeerSync (0.04s)
PASS
ok      github.com/ethereum-optimism/optimism/op-node/p2p   15.146s
=== RUN   TestGetEmptyScoreComponents
--- PASS: TestGetEmptyScoreComponents (0.00s)
=== RUN   TestRoundTripGossipScore
--- PASS: TestRoundTripGossipScore (0.00s)
=== RUN   TestUpdateGossipScore
--- PASS: TestUpdateGossipScore (0.00s)
=== RUN   TestStoreScoresForMultiplePeers
--- PASS: TestStoreScoresForMultiplePeers (0.00s)
=== RUN   TestPersistData
--- PASS: TestPersistData (0.00s)
=== RUN   TestUnknownScoreType
--- PASS: TestUnknownScoreType (0.00s)
=== RUN   TestRoundtripScoresV0
--- PASS: TestRoundtripScoresV0 (0.00s)
=== RUN   TestParseHistoricSerializationsV0
=== RUN   TestParseHistoricSerializationsV0/GossipOnly
--- PASS: TestParseHistoricSerializationsV0 (0.00s)
    --- PASS: TestParseHistoricSerializationsV0/GossipOnly (0.00s)
PASS
```

```
ok      github.com/ethereum-optimism/optimism/op-node/p2p/store 2.303s
=== RUN   TestConfigJSON
--- PASS: TestConfigJSON (0.00s)
=== RUN   TestValidateL1Config
--- PASS: TestValidateL1Config (0.00s)
=== RUN   TestValidateL1ConfigInvalidChainIdFails
--- PASS: TestValidateL1ConfigInvalidChainIdFails (0.00s)
=== RUN   TestValidateL1ConfigInvalidGenesisHashFails
--- PASS: TestValidateL1ConfigInvalidGenesisHashFails (0.00s)
=== RUN   TestCheckL1ChainID
--- PASS: TestCheckL1ChainID (0.00s)
=== RUN   TestCheckL1BlockRefByNumber
--- PASS: TestCheckL1BlockRefByNumber (0.00s)
=== RUN   TestRandomConfigDescription
=== RUN   TestRandomConfigDescription/named_L2
=== RUN   TestRandomConfigDescription/named_L1
=== RUN   TestRandomConfigDescription/unnamed
=== RUN   TestRandomConfigDescription/regolith_unset
=== RUN   TestRandomConfigDescription/regolith_genesis
=== RUN   TestRandomConfigDescription/regolith_date
--- PASS: TestRandomConfigDescription (0.00s)
    --- PASS: TestRandomConfigDescription/named_L2 (0.00s)
    --- PASS: TestRandomConfigDescription/named_L1 (0.00s)
    --- PASS: TestRandomConfigDescription/unnamed (0.00s)
    --- PASS: TestRandomConfigDescription/regolith_unset (0.00s)
    --- PASS: TestRandomConfigDescription/regolith_genesis (0.00s)
    --- PASS: TestRandomConfigDescription/regolith_date (0.00s)
=== RUN   TestRegolithActivation
--- PASS: TestRegolithActivation (0.00s)
=== RUN   TestValidateL2Config
--- PASS: TestValidateL2Config (0.00s)
=== RUN   TestValidateL2ConfigInvalidChainIdFails
--- PASS: TestValidateL2ConfigInvalidChainIdFails (0.00s)
=== RUN   TestValidateL2ConfigInvalidGenesisHashFails
--- PASS: TestValidateL2ConfigInvalidGenesisHashFails (0.00s)
=== RUN   TestCheckL2ChainID
--- PASS: TestCheckL2ChainID (0.00s)
=== RUN   TestCheckL2BlockRefByNumber
--- PASS: TestCheckL2BlockRefByNumber (0.00s)
=== RUN   TestConfig_Check
=== RUN   TestConfig_Check/BlockTimeZero
=== RUN   TestConfig_Check/ChannelTimeoutZero
=== RUN   TestConfig_Check/SeqWindowSizeZero
=== RUN   TestConfig_Check/SeqWindowSizeOne
=== RUN   TestConfig_Check/NoL1Genesis
=== RUN   TestConfig_Check/NoL2Genesis
=== RUN   TestConfig_Check/GenesisHashesEqual
=== RUN   TestConfig_Check/GenesisL2TimeZero
=== RUN   TestConfig_Check/NoBatcherAddr
=== RUN   TestConfig_Check/NoOverhead
=== RUN   TestConfig_Check/NoScalar
=== RUN   TestConfig_Check/NoGasLimit
=== RUN   TestConfig_Check/NoBatchInboxAddress
=== RUN   TestConfig_Check/NoDepositContractAddress
=== RUN   TestConfig_Check/NoL1ChainId
=== RUN   TestConfig_Check/NoL2ChainId
=== RUN   TestConfig_Check/ChainIDsEqual
=== RUN   TestConfig_Check/L1ChainIdNegative
=== RUN   TestConfig_Check/L1ChainIdZero
=== RUN   TestConfig_Check/L2ChainIdNegative
=== RUN   TestConfig_Check/L2ChainIdZero
--- PASS: TestConfig_Check (0.00s)
    --- PASS: TestConfig_Check/BlockTimeZero (0.00s)
    --- PASS: TestConfig_Check/ChannelTimeoutZero (0.00s)
    --- PASS: TestConfig_Check/SeqWindowSizeZero (0.00s)
    --- PASS: TestConfig_Check/SeqWindowSizeOne (0.00s)
    --- PASS: TestConfig_Check/NoL1Genesis (0.00s)
    --- PASS: TestConfig_Check/NoL2Genesis (0.00s)
    --- PASS: TestConfig_Check/GenesisHashesEqual (0.00s)
    --- PASS: TestConfig_Check/GenesisL2TimeZero (0.00s)
    --- PASS: TestConfig_Check/NoBatcherAddr (0.00s)
    --- PASS: TestConfig_Check/NoOverhead (0.00s)
    --- PASS: TestConfig_Check/NoScalar (0.00s)
    --- PASS: TestConfig_Check/NoGasLimit (0.00s)
    --- PASS: TestConfig_Check/NoBatchInboxAddress (0.00s)
    --- PASS: TestConfig_Check/NoDepositContractAddress (0.00s)
    --- PASS: TestConfig_Check/NoL1ChainId (0.00s)
    --- PASS: TestConfig_Check/NoL2ChainId (0.00s)
    --- PASS: TestConfig_Check/ChainIDsEqual (0.00s)
    --- PASS: TestConfig_Check/L1ChainIdNegative (0.00s)
    --- PASS: TestConfig_Check/L1ChainIdZero (0.00s)
    --- PASS: TestConfig_Check/L2ChainIdNegative (0.00s)
    --- PASS: TestConfig_Check/L2ChainIdZero (0.00s)
PASS
ok      github.com/ethereum-optimism/optimism/op-node/rollup    0.782s
=== RUN   TestAttributesQueue
    mock.go:322: FAIL:    InfoByHash(common.Hash)
                at: [/Users/poanlin/Downloads/opbnb-audit/opbnb/op-
node/rollup/derive/mock_eth_client.go:24 /Users/poanlin/Downloads/opbnb-audit/opbnb/op-
node/rollup/derive/attributes_queue_test.go:39]
    mock.go:322: FAIL: 0 out of 1 expectation(s) were met.
            The code you are testing needs to make 1 more call(s).
            at: [/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/panic.go:884
/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/mock_eth_client.go:46
/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/attributes.go:59
/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/attributes_queue.go:85
/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/attributes_queue_test.go:83]
--- FAIL: TestAttributesQueue (0.00s)
panic:
assert: mock: I don't know what to return because the method call was unexpected.
    Either do Mock.On("InfoAndTxsByHash").Return(...) first, or remove the InfoAndTxsByHash() call.
    This method was unexpected:
        InfoAndTxsByHash(common.Hash)
        0: 0x547dea8ff339566349ed0ef6384876655d1b9b955e36ac165c6b8ab69b9af5cd
    at: [/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/mock_eth_client.go:46
/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/attributes.go:59
/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/attributes_queue.go:85
/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/attributes_queue_test.go:83] [recovered]
    panic:
assert: mock: I don't know what to return because the method call was unexpected.
    Either do Mock.On("InfoAndTxsByHash").Return(...) first, or remove the InfoAndTxsByHash() call.
    This method was unexpected:
        InfoAndTxsByHash(common.Hash)
        0: 0x547dea8ff339566349ed0ef6384876655d1b9b955e36ac165c6b8ab69b9af5cd
    at: [/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/mock_eth_client.go:46
/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/attributes.go:59
/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/attributes_queue.go:85
/Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/attributes_queue_test.go:83]

goroutine 28 [running]:
testing.tRunner.func1.2({0x100a21240, 0xc00045e230})
    /usr/local/Cellar/go/1.20.4/libexec/src/testing/testing.go:1526 +0x24e
testing.tRunner.func1()
    /usr/local/Cellar/go/1.20.4/libexec/src/testing/testing.go:1529 +0x39f
panic({0x100a21240, 0xc00045e230})
    /usr/local/Cellar/go/1.20.4/libexec/src/runtime/panic.go:884 +0x213
github.com/stretchr/testify/mock.(*Mock).fail(0xc00044a230, {0x100bb3295?, 0x4?}, {0xc000434d00?, 0x1?,
0x1?})
    /Users/poanlin/go/pkg/mod/github.com/stretchr/testify@v1.8.1/mock/mock.go:322 +0x145
github.com/stretchr/testify/mock.(*Mock).MethodCalled(0xc00044a230, {0x100b7cb2c, 0x10}, {0xc00045e100,
0x1, 0x1})
    /Users/poanlin/go/pkg/mod/github.com/stretchr/testify@v1.8.1/mock/mock.go:485 +0x729
github.com/ethereum-optimism/optimism/op-node/testutils.(*MockEthClient).InfoAndTxsByHash(0xc00044a230,
{0x100aea9c0?, 0x1?}, {0x54, 0x7d, 0xea, 0x8f, 0xf3, 0x39, 0x56, ...})
    /Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/testutils/mock_eth_client.go:46 +0x98
github.com/ethereum-optimism/optimism/op-node/rollup/derive.
(*FetchingAttributesBuilder).PreparePayloadAttributes(0xc000446d20, {0x100e22680, 0xc000446db0}, {{0xd9,
0xf0, 0x87, 0x75, 0x7a, 0xcc, 0xe3, ...}, ...}, ...)
    /Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/attributes.go:59 +0x1ea
github.com/ethereum-optimism/optimism/op-node/rollup/derive.
(*AttributesQueue).createNextAttributes(0xc0000d7e50, {0x100e22648, 0xc00003c120}, 0xc0000d7ef0, {{0xd9,
0xf0, 0x87, 0x75, 0x7a, 0xcc, ...}, ...})
    /Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/attributes_queue.go:85 +0x1ba
github.com/ethereum-optimism/optimism/op-node/rollup/derive.TestAttributesQueue(0xc000377d40)
```

```
    /Users/poanlin/Downloads/opbnb-audit/opbnb/op-node/rollup/derive/attributes_queue_test.go:83 +0xa0f
testing.tRunner(0xc000377d40, 0x100d2d520)
    /usr/local/Cellar/go/1.20.4/libexec/src/testing/testing.go:1576 +0x10b
created by testing.(*T).Run
    /usr/local/Cellar/go/1.20.4/libexec/src/testing/testing.go:1629 +0x3ea
FAIL    github.com/ethereum-optimism/optimism/op-node/rollup/derive 1.108s
=== RUN   TestConfDepth
=== RUN   TestConfDepth/zero_conf_future
    conf_depth_test.go:37: PASS:    L1BlockRefByNumber(uint64)
=== RUN   TestConfDepth/zero_conf_present
    conf_depth_test.go:37: PASS:    L1BlockRefByNumber(uint64)
=== RUN   TestConfDepth/zero_conf_past
    conf_depth_test.go:37: PASS:    L1BlockRefByNumber(uint64)
=== RUN   TestConfDepth/one_conf_future
=== RUN   TestConfDepth/one_conf_present
=== RUN   TestConfDepth/one_conf_past
    conf_depth_test.go:37: PASS:    L1BlockRefByNumber(uint64)
=== RUN   TestConfDepth/two_conf_future
=== RUN   TestConfDepth/two_conf_present
=== RUN   TestConfDepth/two_conf_not_like_1
=== RUN   TestConfDepth/two_conf_pass
    conf_depth_test.go:37: PASS:    L1BlockRefByNumber(uint64)
=== RUN   TestConfDepth/easy_pass
    conf_depth_test.go:37: PASS:    L1BlockRefByNumber(uint64)
=== RUN   TestConfDepth/genesis_case
    conf_depth_test.go:37: PASS:    L1BlockRefByNumber(uint64)
=== RUN   TestConfDepth/no_L1_state
    conf_depth_test.go:37: PASS:    L1BlockRefByNumber(uint64)
--- PASS: TestConfDepth (0.01s)
    --- PASS: TestConfDepth/zero_conf_future (0.00s)
    --- PASS: TestConfDepth/zero_conf_present (0.00s)
    --- PASS: TestConfDepth/zero_conf_past (0.00s)
    --- PASS: TestConfDepth/one_conf_future (0.00s)
    --- PASS: TestConfDepth/one_conf_present (0.00s)
    --- PASS: TestConfDepth/one_conf_past (0.00s)
    --- PASS: TestConfDepth/two_conf_future (0.00s)
    --- PASS: TestConfDepth/two_conf_present (0.00s)
    --- PASS: TestConfDepth/two_conf_not_like_1 (0.00s)
    --- PASS: TestConfDepth/two_conf_pass (0.00s)
    --- PASS: TestConfDepth/easy_pass (0.00s)
    --- PASS: TestConfDepth/genesis_case (0.00s)
    --- PASS: TestConfDepth/no_L1_state (0.00s)
=== RUN   TestDurationRecorded
=== RUN   TestDurationRecorded/L1BlockRefByLabel
    metered_l1fetcher_test.go:109: PASS:    RecordL1RequestTime(string,time.Duration)
    metered_l1fetcher_test.go:109: PASS:    L1BlockRefByLabel(eth.BlockLabel)
=== RUN   TestDurationRecorded/L1BlockRefByNumber
    metered_l1fetcher_test.go:109: PASS:    RecordL1RequestTime(string,time.Duration)
    metered_l1fetcher_test.go:109: PASS:    L1BlockRefByNumber(uint64)
=== RUN   TestDurationRecorded/L1BlockRefByHash
    metered_l1fetcher_test.go:109: PASS:    RecordL1RequestTime(string,time.Duration)
    metered_l1fetcher_test.go:109: PASS:    L1BlockRefByHash(common.Hash)
=== RUN   TestDurationRecorded/InfoByHash
    metered_l1fetcher_test.go:109: PASS:    RecordL1RequestTime(string,time.Duration)
    metered_l1fetcher_test.go:109: PASS:    InfoByHash(common.Hash)
=== RUN   TestDurationRecorded/InfoAndTxsByHash
    metered_l1fetcher_test.go:109: PASS:    RecordL1RequestTime(string,time.Duration)
    metered_l1fetcher_test.go:109: PASS:    InfoAndTxsByHash(common.Hash)
=== RUN   TestDurationRecorded/FetchReceipts
    metered_l1fetcher_test.go:109: PASS:    RecordL1RequestTime(string,time.Duration)
    metered_l1fetcher_test.go:109: PASS:    FetchReceipts(common.Hash)
--- PASS: TestDurationRecorded (0.01s)
    --- PASS: TestDurationRecorded/L1BlockRefByLabel (0.00s)
    --- PASS: TestDurationRecorded/L1BlockRefByNumber (0.00s)
    --- PASS: TestDurationRecorded/L1BlockRefByHash (0.00s)
    --- PASS: TestDurationRecorded/InfoByHash (0.00s)
    --- PASS: TestDurationRecorded/InfoAndTxsByHash (0.00s)
    --- PASS: TestDurationRecorded/FetchReceipts (0.00s)
=== RUN   TestOriginSelectorAdvances
    origin_selector_test.go:54: PASS:   L1BlockRefByHash(common.Hash)
    origin_selector_test.go:54: PASS:   L1BlockRefByNumber(uint64)
--- PASS: TestOriginSelectorAdvances (0.00s)
```

```
=== RUN   TestOriginSelectorRespectsOriginTiming
    origin_selector_test.go:95: PASS:    L1BlockRefByHash(common.Hash)
    origin_selector_test.go:95: PASS:    L1BlockRefByNumber(uint64)
--- PASS: TestOriginSelectorRespectsOriginTiming (0.00s)
=== RUN   TestOriginSelectorRespectsConfDepth
    origin_selector_test.go:135: PASS:    L1BlockRefByHash(common.Hash)
--- PASS: TestOriginSelectorRespectsConfDepth (0.00s)
=== RUN   TestOriginSelectorStrictConfDepth
    origin_selector_test.go:177: PASS:    L1BlockRefByHash(common.Hash)
--- PASS: TestOriginSelectorStrictConfDepth (0.00s)
=== RUN   TestOriginSelectorSeqDriftRespectsNextOriginTime
    origin_selector_test.go:216: PASS:    L1BlockRefByHash(common.Hash)
    origin_selector_test.go:216: PASS:    L1BlockRefByNumber(uint64)
--- PASS: TestOriginSelectorSeqDriftRespectsNextOriginTime (0.00s)
=== RUN   TestOriginSelectorHandlesLateL1Blocks
    origin_selector_test.go:284: PASS:    L1BlockRefByHash(common.Hash)
    origin_selector_test.go:284: PASS:    L1BlockRefByHash(common.Hash)
    origin_selector_test.go:284: PASS:    L1BlockRefByHash(common.Hash)
    origin_selector_test.go:284: PASS:    L1BlockRefByNumber(uint64)
--- PASS: TestOriginSelectorHandlesLateL1Blocks (0.00s)
=== RUN   TestSequencerChaosMonkey
    sequencer_test.go:370: avg build time: 1.068738411s, clock timestamp: 102298, L2 head time: 102300,
L1 origin time: 102226, avg txs per block: 5.148000
--- PASS: TestSequencerChaosMonkey (0.27s)
PASS
ok      github.com/ethereum-optimism/optimism/op-node/rollup/driver 1.674s
=== RUN   TestFindSyncStart
=== RUN   TestFindSyncStart/already_synced
=== RUN   TestFindSyncStart/small_reorg_long_chain
=== RUN   TestFindSyncStart/L1_Chain_ahead
=== RUN   TestFindSyncStart/L2_Chain_ahead_after_reorg
=== RUN   TestFindSyncStart/genesis
=== RUN   TestFindSyncStart/reorg_one_step_back
=== RUN   TestFindSyncStart/reorg_two_steps_back,_clip_genesis_and_finalized
=== RUN   TestFindSyncStart/reorg_three_steps_back
=== RUN   TestFindSyncStart/unexpected_L1_chain
=== RUN   TestFindSyncStart/unexpected_L2_chain
=== RUN   TestFindSyncStart/offset_L2_genesis
=== RUN   TestFindSyncStart/offset_L2_genesis_reorg
=== RUN   TestFindSyncStart/reorg_past_offset_genesis
--- PASS: TestFindSyncStart (0.01s)
    --- PASS: TestFindSyncStart/already_synced (0.00s)
    --- PASS: TestFindSyncStart/small_reorg_long_chain (0.00s)
    --- PASS: TestFindSyncStart/L1_Chain_ahead (0.00s)
    --- PASS: TestFindSyncStart/L2_Chain_ahead_after_reorg (0.00s)
    --- PASS: TestFindSyncStart/genesis (0.00s)
    --- PASS: TestFindSyncStart/reorg_one_step_back (0.00s)
    --- PASS: TestFindSyncStart/reorg_two_steps_back,_clip_genesis_and_finalized (0.00s)
    --- PASS: TestFindSyncStart/reorg_three_steps_back (0.00s)
    --- PASS: TestFindSyncStart/unexpected_L1_chain (0.00s)
    --- PASS: TestFindSyncStart/unexpected_L2_chain (0.00s)
    --- PASS: TestFindSyncStart/offset_L2_genesis (0.00s)
    --- PASS: TestFindSyncStart/offset_L2_genesis_reorg (0.00s)
    --- PASS: TestFindSyncStart/reorg_past_offset_genesis (0.00s)
PASS
ok      github.com/ethereum-optimism/optimism/op-node/rollup/sync   1.475s
=== RUN   TestFetchBatched
=== RUN   TestFetchBatched/empty
=== RUN   TestFetchBatched/simple
    batching_test.go:173: PASS:    getBatch([]rpc.BatchElem)
=== RUN   TestFetchBatched/single_element
    batching_test.go:173: PASS:    getSingle(*string,string,int)
=== RUN   TestFetchBatched/unbatched
    batching_test.go:173: PASS:    getSingle(*string,string,int)
    batching_test.go:173: PASS:    getSingle(*string,string,int)
    batching_test.go:173: PASS:    getSingle(*string,string,int)
    batching_test.go:173: PASS:    getSingle(*string,string,int)
=== RUN   TestFetchBatched/unbatched_with_retry
    batching_test.go:173: PASS:    getSingle(*string,string,int)
    batching_test.go:173: PASS:    getSingle(*string,string,int)
    batching_test.go:173: PASS:    getSingle(*string,string,int)
    batching_test.go:173: PASS:    getSingle(*string,string,int)
    batching_test.go:173: PASS:    getSingle(*string,string,int)
=== RUN   TestFetchBatched/split
    batching_test.go:173: PASS:    getBatch([]rpc.BatchElem)
    batching_test.go:173: PASS:    getBatch([]rpc.BatchElem)
=== RUN   TestFetchBatched/efficient_retry
    batching_test.go:173: PASS:    getBatch([]rpc.BatchElem)
    batching_test.go:173: PASS:    getBatch([]rpc.BatchElem)
    batching_test.go:173: PASS:    getBatch([]rpc.BatchElem)
    batching_test.go:173: PASS:    getBatch([]rpc.BatchElem)
=== RUN   TestFetchBatched/repeated_sequential_retries
    batching_test.go:173: PASS:    getBatch([]rpc.BatchElem)
    batching_test.go:173: PASS:    getBatch([]rpc.BatchElem)
    batching_test.go:173: PASS:    getBatch([]rpc.BatchElem)
=== RUN   TestFetchBatched/context_timeout
    batching_test.go:173: PASS:    getBatch([]rpc.BatchElem)
--- PASS: TestFetchBatched (0.02s)
    --- PASS: TestFetchBatched/empty (0.00s)
    --- PASS: TestFetchBatched/simple (0.00s)
    --- PASS: TestFetchBatched/single_element (0.00s)
    --- PASS: TestFetchBatched/unbatched (0.00s)
    --- PASS: TestFetchBatched/unbatched_with_retry (0.01s)
    --- PASS: TestFetchBatched/split (0.00s)
    --- PASS: TestFetchBatched/efficient_retry (0.00s)
    --- PASS: TestFetchBatched/repeated_sequential_retries (0.00s)
    --- PASS: TestFetchBatched/context_timeout (0.00s)
=== RUN   TestEthClient_InfoByHash
    eth_client_test.go:118: PASS:    CallContext(*context.emptyCtx,**sources.rpcHeader,string,[]interface
{})
    eth_client_test.go:123: PASS:    CallContext(*context.emptyCtx,**sources.rpcHeader,string,[]interface
{})
--- PASS: TestEthClient_InfoByHash (0.00s)
=== RUN   TestEthClient_InfoByNumber
    eth_client_test.go:141: PASS:    CallContext(*context.emptyCtx,**sources.rpcHeader,string,[]interface
{})
--- PASS: TestEthClient_InfoByNumber (0.00s)
=== RUN   TestEthClient_WrongInfoByNumber
    eth_client_test.go:159: PASS:    CallContext(*context.emptyCtx,**sources.rpcHeader,string,[]interface
{})
--- PASS: TestEthClient_WrongInfoByNumber (0.00s)
=== RUN   TestEthClient_WrongInfoByHash
    eth_client_test.go:178: PASS:    CallContext(*context.emptyCtx,**sources.rpcHeader,string,[]interface
{})
--- PASS: TestEthClient_WrongInfoByHash (0.00s)
=== RUN   TestEthClient_FetchReceipts
=== RUN   TestEthClient_FetchReceipts/alchemy
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    alchemy_getTransactionReceipts(string)
=== RUN   TestEthClient_FetchReceipts/alchemy_sticky
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    alchemy_getTransactionReceipts(string)
    receipts_test.go:173: PASS:    alchemy_getTransactionReceipts(string)
=== RUN   TestEthClient_FetchReceipts/alchemy_fallback_1
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    alchemy_getTransactionReceipts(string)
    receipts_test.go:173: PASS:    eth_getBlockReceipts(string)
=== RUN   TestEthClient_FetchReceipts/alchemy_low_tx_count_cost_saving
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
=== RUN   TestEthClient_FetchReceipts/quicknode
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    debug_getRawReceipts(string)
=== RUN   TestEthClient_FetchReceipts/quicknode_fallback_1
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    debug_getRawReceipts(string)
    receipts_test.go:173: PASS:    eth_getBlockReceipts(string)
=== RUN   TestEthClient_FetchReceipts/quicknode_low_tx_count_cost_saving
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    debug_getRawReceipts(string)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
=== RUN   TestEthClient_FetchReceipts/infura
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
=== RUN   TestEthClient_FetchReceipts/nethermind
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    parity_getBlockReceipts(string)
=== RUN   TestEthClient_FetchReceipts/geth_with_debug_rpc
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    debug_getRawReceipts(string)
=== RUN   TestEthClient_FetchReceipts/erigon
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    eth_getBlockReceipts(string)
=== RUN   TestEthClient_FetchReceipts/basic
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
    receipts_test.go:173: PASS:    eth_getTransactionReceipt(common.Hash)
=== RUN   TestEthClient_FetchReceipts/any_discovers_alchemy
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    alchemy_getTransactionReceipts(string)
=== RUN   TestEthClient_FetchReceipts/any_discovers_parity
    receipts_test.go:173: PASS:    eth_getBlockByHash(common.Hash,bool)
    receipts_test.go:173: PASS:    alchemy_getTransactionReceipts(string)
    receipts_test.go:173: PASS:    debug_getRawReceipts(string)
    receipts_test.go:173: PASS:    eth_getBlockReceipts(string)
    receipts_test.go:173: PASS:    parity_getBlockReceipts(string)
--- PASS: TestEthClient_FetchReceipts (0.76s)
    --- PASS: TestEthClient_FetchReceipts/alchemy (0.12s)
    --- PASS: TestEthClient_FetchReceipts/alchemy_sticky (0.15s)
    --- PASS: TestEthClient_FetchReceipts/alchemy_fallback_1 (0.13s)
    --- PASS: TestEthClient_FetchReceipts/alchemy_low_tx_count_cost_saving (0.04s)
    --- PASS: TestEthClient_FetchReceipts/quicknode (0.08s)
    --- PASS: TestEthClient_FetchReceipts/quicknode_fallback_1 (0.10s)
    --- PASS: TestEthClient_FetchReceipts/quicknode_low_tx_count_cost_saving (0.02s)
    --- PASS: TestEthClient_FetchReceipts/infura (0.02s)
    --- PASS: TestEthClient_FetchReceipts/nethermind (0.02s)
    --- PASS: TestEthClient_FetchReceipts/geth_with_debug_rpc (0.01s)
    --- PASS: TestEthClient_FetchReceipts/erigon (0.02s)
    --- PASS: TestEthClient_FetchReceipts/basic (0.03s)
    --- PASS: TestEthClient_FetchReceipts/any_discovers_alchemy (0.01s)
    --- PASS: TestEthClient_FetchReceipts/any_discovers_parity (0.02s)
=== RUN   TestBlockJSON
--- PASS: TestBlockJSON (0.00s)
PASS
ok      github.com/ethereum-optimism/optimism/op-node/sources   1.935s
=== RUN   TestParseMessagePassed
=== RUN   TestParseMessagePassed/withdrawal_through_bridge
--- PASS: TestParseMessagePassed (0.00s)
    --- PASS: TestParseMessagePassed/withdrawal_through_bridge (0.00s)
PASS
ok      github.com/ethereum-optimism/optimism/op-node/withdrawals   0.639s
FAIL
make[1]: *** [test] Error 1
make: *** [test-unit] Error 2
```

Below is for `contract-bedrock`

```
yarn run v1.22.19
$ yarn build:differential && yarn build:fuzz && forge test
$ go build -o ./scripts/differential-testing/differential-testing ./scripts/differential-testing
$ (cd test-case-generator && go build ./cmd/fuzz.go)
[⠒] Compiling...
```

```
[⠢] Compiling 1 files with 0.4.26
[⠆] Compiling 184 files with 0.8.15
[⠃] Solc 0.4.26 finished in 41.79ms
[⠆] Solc 0.8.15 finished in 55.68s
Compiler run successful

Running 1 test for contracts/test/BenchmarkTest.t.sol:SetPrevBaseFee_Test
[PASS] test_setPrevBaseFee_succeeds() (gas: 11515)
Test result: ok. 1 passed; 0 failed; finished in 2.87ms

Running 2 tests for contracts/test/DeployerWhitelist.t.sol:DeployerWhitelist_Test
[PASS] test_owner_succeeds() (gas: 7582)
[PASS] test_storageSlots_succeeds() (gas: 33395)
Test result: ok. 2 passed; 0 failed; finished in 335.08µs

Running 1 test for contracts/test/BenchmarkTest.t.sol:GasBenchMark_L1StandardBridge_Finalize
[PASS] test_finalizeETHWithdrawal_benchmark() (gas: 40409)
Test result: ok. 1 passed; 0 failed; finished in 5.99ms

Running 8 tests for contracts/test/GasPriceOracle.t.sol:GasPriceOracle_Test
[PASS] test_baseFee_succeeds() (gas: 8325)
[PASS] test_decimals_succeeds() (gas: 6167)
[PASS] test_gasPrice_succeeds() (gas: 8294)
[PASS] test_l1BaseFee_succeeds() (gas: 10656)
[PASS] test_overhead_succeeds() (gas: 10636)
[PASS] test_scalar_succeeds() (gas: 10677)
[PASS] test_setGasPrice_doesNotExist_reverts() (gas: 5910)
[PASS] test_setL1BaseFee_doesNotExist_reverts() (gas: 5911)
Test result: ok. 8 passed; 0 failed; finished in 1.16ms

Running 2 tests for contracts/test/BenchmarkTest.t.sol:GasBenchMark_L1CrossDomainMessenger
[PASS] test_sendMessage_benchmark_0() (gas: 159768)
[PASS] test_sendMessage_benchmark_1() (gas: 1025763)
Test result: ok. 2 passed; 0 failed; finished in 6.51ms

Running 4 tests for contracts/test/OptimismPortal.t.sol:OptimismPortalUpgradeable_Test
[PASS] test_initialize_cannotInitImpl_reverts() (gas: 10994)
[PASS] test_initialize_cannotInitProxy_reverts() (gas: 15918)
[PASS] test_params_initValuesOnProxy_succeeds() (gas: 21774)
[PASS] test_upgradeToAndCall_upgrading_succeeds() (gas: 180547)
Test result: ok. 4 passed; 0 failed; finished in 4.94ms

Running 1 test for contracts/test/AddressAliasHelper.t.sol:AddressAliasHelper_applyAndUndo_Test
[PASS] testFuzz_applyAndUndo_succeeds(address) (runs: 256, μ: 364, ~: 364)
Test result: ok. 1 passed; 0 failed; finished in 6.71ms

Running 4 tests for contracts/test/L2StandardBridge.t.sol:L2StandardBridge_Test
[PASS] test_initialize_succeeds() (gas: 24292)
[PASS] test_receive_succeeds() (gas: 174641)
[PASS] test_withdraw_ether_succeeds() (gas: 140793)
[PASS] test_withdraw_insufficientValue_reverts() (gas: 16485)
Test result: ok. 4 passed; 0 failed; finished in 9.50ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_FinalizeETHWithdrawal_Test
[PASS] test_finalizeETHWithdrawal_succeeds() (gas: 61722)
Test result: ok. 1 passed; 0 failed; finished in 7.99ms

Running 1 test for contracts/test/BenchmarkTest.t.sol:GasBenchMark_L2OutputOracle
[PASS] test_proposeL2Output_benchmark() (gas: 88513)
Test result: ok. 1 passed; 0 failed; finished in 1.57ms

Running 10 tests for contracts/test/GovernanceToken.t.sol:GovernanceToken_Test
[PASS] test_approve_succeeds() (gas: 133293)
[PASS] test_burnFrom_succeeds() (gas: 122733)
[PASS] test_burn_succeeds() (gas: 114588)
[PASS] test_constructor_succeeds() (gas: 21298)
[PASS] test_decreaseAllowance_succeeds() (gas: 137008)
[PASS] test_increaseAllowance_succeeds() (gas: 137118)
[PASS] test_mint_fromNotOwner_reverts() (gas: 17030)
[PASS] test_mint_fromOwner_succeeds() (gas: 108592)
[PASS] test_transferFrom_succeeds() (gas: 146273)
[PASS] test_transfer_succeeds() (gas: 138108)
```

```
Test result: ok. 10 passed; 0 failed; finished in 8.69ms

Running 4 tests for contracts/test/BenchmarkTest.t.sol:GasBenchMark_L1StandardBridge_Deposit
[PASS] test_depositERC20_benchmark_0() (gas: 277779)
[PASS] test_depositERC20_benchmark_1() (gas: 1448536)
[PASS] test_depositETH_benchmark_0() (gas: 213608)
[PASS] test_depositETH_benchmark_1() (gas: 1228875)
Test result: ok. 4 passed; 0 failed; finished in 27.83ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_Getter_Test
[PASS] test_getters_succeeds() (gas: 32173)
Test result: ok. 1 passed; 0 failed; finished in 14.84ms

Running 4 tests for contracts/test/L2ToL1MessagePasser.t.sol:L2ToL1MessagePasserTest
[PASS] testFuzz_initiateWithdrawal_succeeds(address,address,uint256,uint256,bytes) (runs: 256, μ: 73879,
~: 73718)
[PASS] test_burn_succeeds() (gas: 112572)
[PASS] test_initiateWithdrawal_fromContract_succeeds() (gas: 70445)
[PASS] test_initiateWithdrawal_fromEOA_succeeds() (gas: 75896)
Test result: ok. 4 passed; 0 failed; finished in 28.66ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_Initialize_Test
[PASS] test_initialize_succeeds() (gas: 22050)
Test result: ok. 1 passed; 0 failed; finished in 6.20ms

Running 2 tests for contracts/test/FeeVault.t.sol:FeeVault_Test
[PASS] test_constructor_succeeds() (gas: 10736)
[PASS] test_minWithdrawalAmount_succeeds() (gas: 10713)
Test result: ok. 2 passed; 0 failed; finished in 5.90ms

Running 9 tests for contracts/test/LegacyERC20ETH.t.sol:LegacyERC20ETH_Test
[PASS] test_approve_doesNotExist_reverts() (gas: 10702)
[PASS] test_burn_doesNotExist_reverts() (gas: 10637)
[PASS] test_crossDomain_succeeds() (gas: 6341)
[PASS] test_decreaseAllowance_doesNotExist_reverts() (gas: 10724)
[PASS] test_increaseAllowance_doesNotExist_reverts() (gas: 10690)
[PASS] test_metadata_succeeds() (gas: 15470)
[PASS] test_mint_doesNotExist_reverts() (gas: 10627)
[PASS] test_transferFrom_doesNotExist_reverts() (gas: 12957)
[PASS] test_transfer_doesNotExist_reverts() (gas: 10755)
Test result: ok. 9 passed; 0 failed; finished in 1.27ms

Running 9 tests for contracts/test/DisputeGameFactory.t.sol:DisputeGameFactory_Test
[PASS] testDiff_getGameUUID_succeeds(uint8,bytes32,bytes) (runs: 256, μ: 11509, ~: 11693)
[PASS] testFuzz_create_noImpl_reverts(uint8,bytes32,bytes) (runs: 256, μ: 16621, ~: 16769)
[PASS] testFuzz_create_sameUUID_reverts(uint8,bytes32,bytes) (runs: 256, μ: 242809, ~: 239368)
[PASS] testFuzz_create_succeeds(uint8,bytes32,bytes) (runs: 256, μ: 179488, ~: 177663)
[PASS] test_owner_succeeds() (gas: 7582)
[PASS] test_setImplementation_notOwner_reverts() (gas: 11191)
[PASS] test_setImplementation_succeeds() (gas: 32635)
[PASS] test_transferOwnership_notOwner_reverts() (gas: 10979)
[PASS] test_transferOwnership_succeeds() (gas: 13180)
Test result: ok. 9 passed; 0 failed; finished in 149.47ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_Receive_Test
[PASS] test_receive_succeeds() (gas: 283504)
Test result: ok. 1 passed; 0 failed; finished in 6.19ms

Running 1 test for contracts/test/LegacyMessagePasser.t.sol:LegacyMessagePasser_Test
[PASS] test_passMessageToL1_succeeds() (gas: 34524)
Test result: ok. 1 passed; 0 failed; finished in 739.21μs

Running 3 tests for contracts/test/CrossDomainMessenger.t.sol:CrossDomainMessenger_BaseGas_Test
[PASS] testFuzz_baseGas_portalMinGasLimit_succeeds(bytes,uint32) (runs: 256, μ: 34923, ~: 34877)
[PASS] testFuzz_baseGas_succeeds(uint32) (runs: 256, μ: 20546, ~: 20546)
[PASS] test_baseGas_succeeds() (gas: 20412)
Test result: ok. 3 passed; 0 failed; finished in 44.14ms

Running 9 tests for contracts/test/L2CrossDomainMessenger.t.sol:L2CrossDomainMessenger_Test
[PASS] test_messageVersion_succeeds() (gas: 8434)
[PASS] test_relayMessage_retry_succeeds() (gas: 163771)
[PASS] test_relayMessage_succeeds() (gas: 48946)
```

```
[PASS] test_relayMessage_toSystemContract_reverts() (gas: 29021)
[PASS] test_relayMessage_v2_reverts() (gas: 11711)
[PASS] test_sendMessage_succeeds() (gas: 123768)
[PASS] test_sendMessage_twice_succeeds() (gas: 135456)
[PASS] test_xDomainMessageSender_reset_succeeds() (gas: 48422)
[PASS] test_xDomainSender_senderNotSet_reverts() (gas: 10612)
Test result: ok. 9 passed; 0 failed; finished in 13.09ms

Running 10 tests for contracts/test/BondManager.t.sol:BondManager_Test
[PASS] testFuzz_post_duplicates_reverts(bytes32,address,uint256,uint256) (runs: 256, μ: 115903, ~:
115903)
[PASS] testFuzz_post_succeeds(bytes32,address,uint256,uint256) (runs: 256, μ: 112569, ~: 112569)
[PASS] testFuzz_post_zeroAddress_reverts(bytes32,address,uint256) (runs: 256, μ: 11704, ~: 11704)
[PASS] testFuzz_post_zeroAddress_reverts(bytes32,uint256,uint256) (runs: 256, μ: 17621, ~: 17936)
[PASS] testFuzz_reclaim_succeeds(bytes32,address,uint256,uint256) (runs: 256, μ: 116255, ~: 116296)
[PASS] testFuzz_seizeAndSplit_succeeds(bytes32,uint256,bytes) (runs: 256, μ: 1034702, ~: 1022427)
[PASS] testFuzz_seize_expired_reverts(bytes32,address,uint256,uint256) (runs: 256, μ: 110397, ~: 110397)
[PASS] testFuzz_seize_missingBond_reverts(bytes32) (runs: 256, μ: 17144, ~: 17144)
[PASS] testFuzz_seize_succeeds(bytes32,uint256,bytes) (runs: 256, μ: 901973, ~: 891360)
[PASS] testFuzz_seize_unauthorized_reverts(bytes32,address,uint256,uint256) (runs: 256, μ: 750540, ~:
750540)
Test result: ok. 10 passed; 0 failed; finished in 292.83ms

Running 16 tests for contracts/test/L2ERC721Bridge.t.sol:L2ERC721Bridge_Test
[PASS] test_bridgeERC721To_localTokenZeroAddress_reverts() (gas: 26431)
[PASS] test_bridgeERC721To_remoteTokenZeroAddress_reverts() (gas: 21814)
[PASS] test_bridgeERC721To_succeeds() (gas: 147356)
[PASS] test_bridgeERC721To_wrongOwner_reverts() (gas: 29449)
[PASS] test_bridgeERC721_fromContract_reverts() (gas: 22148)
[PASS] test_bridgeERC721_localTokenZeroAddress_reverts() (gas: 24310)
[PASS] test_bridgeERC721_remoteTokenZeroAddress_reverts() (gas: 19628)
[PASS] test_bridgeERC721_succeeds() (gas: 144958)
[PASS] test_bridgeERC721_wrongOwner_reverts() (gas: 29258)
[PASS] test_constructor_succeeds() (gas: 10110)
[PASS] test_finalizeBridgeERC721_alreadyExists_reverts() (gas: 29218)
[PASS] test_finalizeBridgeERC721_interfaceNotCompliant_reverts() (gas: 236327)
[PASS] test_finalizeBridgeERC721_notFromRemoteMessenger_reverts() (gas: 19874)
[PASS] test_finalizeBridgeERC721_notViaLocalMessenger_reverts() (gas: 16104)
[PASS] test_finalizeBridgeERC721_selfToken_reverts() (gas: 17659)
[PASS] test_finalizeBridgeERC721_succeeds() (gas: 169375)
Test result: ok. 16 passed; 0 failed; finished in 5.30ms

Running 26 tests for contracts/test/L2OutputOracle.t.sol:L2OutputOracleTest
[PASS] test_computeL2Timestamp_succeeds() (gas: 37298)
[PASS] test_constructor_badTimestamp_reverts() (gas: 70991)
[PASS] test_constructor_l2BlockTimeZero_reverts() (gas: 45954)
[PASS] test_constructor_submissionInterval_reverts() (gas: 45942)
[PASS] test_constructor_succeeds() (gas: 33805)
[PASS] test_deleteL2Outputs_afterLatest_reverts() (gas: 212306)
[PASS] test_deleteL2Outputs_finalized_reverts() (gas: 108990)
[PASS] test_deleteL2Outputs_ifNotChallenger_reverts() (gas: 18918)
[PASS] test_deleteL2Outputs_nonExistent_reverts() (gas: 107339)
[PASS] test_deleteOutputs_multipleOutputs_succeeds() (gas: 302462)
[PASS] test_deleteOutputs_singleOutput_succeeds() (gas: 181016)
[PASS] test_getL2OutputIndexAfter_multipleOutputsExist_succeeds() (gas: 267098)
[PASS] test_getL2OutputIndexAfter_noOutputsExis_reverts() (gas: 17937)
[PASS] test_getL2OutputIndexAfter_previousBlock_succeeds() (gas: 96044)
[PASS] test_getL2OutputIndexAfter_sameBlock_succeeds() (gas: 95973)
[PASS] test_getL2Output_succeeds() (gas: 101612)
[PASS] test_latestBlockNumber_succeeds() (gas: 96940)
[PASS] test_nextBlockNumber_succeeds() (gas: 17468)
[PASS] test_proposeL2Output_emptyOutput_reverts() (gas: 26668)
[PASS] test_proposeL2Output_futureTimetamp_reverts() (gas: 28647)
[PASS] test_proposeL2Output_notProposer_reverts() (gas: 25806)
[PASS] test_proposeL2Output_proposeAnotherOutput_succeeds() (gas: 101006)
[PASS] test_proposeL2Output_unexpectedBlockNumber_reverts() (gas: 28381)
[PASS] test_proposeL2Output_unmatchedBlockhash_reverts() (gas: 29404)
[PASS] test_proposeL2Output_wrongFork_reverts() (gas: 28984)
[PASS] test_proposeWithBlockhashAndHeight_succeeds() (gas: 95253)
Test result: ok. 26 passed; 0 failed; finished in 3.71ms

Running 2 tests for contracts/test/Bytes.t.sol:Bytes_equal_Test
```

```
[PASS] testDiff_equal_works(bytes,bytes) (runs: 256, μ: 1344, ~: 1326)
[PASS] testFuzz_equal_notEqual_works(bytes,bytes) (runs: 256, μ: 4281, ~: 4275)
Test result: ok. 2 passed; 0 failed; finished in 14.58ms

Running 4 tests for contracts/test/L2OutputOracle.t.sol:L2OutputOracleUpgradeable_Test
[PASS] test_initValuesOnProxy_succeeds() (gas: 26208)
[PASS] test_initializeImpl_alreadyInitialized_reverts() (gas: 15149)
[PASS] test_initializeProxy_alreadyInitialized_reverts() (gas: 20175)
[PASS] test_upgrading_succeeds() (gas: 180481)
Test result: ok. 4 passed; 0 failed; finished in 1.92ms

Running 8 tests for contracts/test/Bytes.t.sol:Bytes_slice_Test
[PASS] testFuzz_slice_lengthOverflows_reverts(bytes,uint256,uint256) (runs: 256, μ: 4235, ~: 4235)
[PASS] testFuzz_slice_memorySafety_succeeds(bytes,uint256,uint256) (runs: 256, μ: 5357, ~: 5470)
[PASS] testFuzz_slice_outOfBounds_reverts(bytes,uint256,uint256) (runs: 256, μ: 4750, ~: 4752)
[PASS] testFuzz_slice_rangeOverflows_reverts(bytes,uint256,uint256) (runs: 256, μ: 4544, ~: 4544)
[PASS] test_slice_acrossMultipleWords_works() (gas: 9413)
[PASS] test_slice_acrossWords_works() (gas: 1430)
[PASS] test_slice_fromNonZeroIdx_works() (gas: 17240)
[PASS] test_slice_fromZeroIdx_works() (gas: 20826)
Test result: ok. 8 passed; 0 failed; finished in 363.47ms

Running 4 tests for contracts/test/Bytes.t.sol:Bytes_toNibbles_Test
[PASS] testDiff_toNibbles_succeeds(bytes) (runs: 256, μ: 48933, ~: 37253)
[PASS] test_toNibbles_expectedResult128Bytes_works() (gas: 129874)
[PASS] test_toNibbles_expectedResult5Bytes_works() (gas: 6132)
[PASS] test_toNibbles_zeroLengthInput_works() (gas: 944)
Test result: ok. 4 passed; 0 failed; finished in 82.63ms

Running 3 tests for contracts/test/BenchmarkTest.t.sol:GasBenchMark_OptimismPortal
[PASS] test_depositTransaction_benchmark() (gas: 42616)
[PASS] test_depositTransaction_benchmark_1() (gas: 42395)
[PASS] test_proveWithdrawalTransaction_benchmark() (gas: 169237)
Test result: ok. 3 passed; 0 failed; finished in 444.93ms

Running 5 tests for contracts/test/MintManager.t.sol:MintManager_mint_Test
[PASS] test_mint_afterPeriodElapsed_succeeds() (gas: 148117)
[PASS] test_mint_beforePeriodElapsed_reverts() (gas: 140433)
[PASS] test_mint_fromNotOwner_reverts() (gas: 10987)
[PASS] test_mint_fromOwner_succeeds() (gas: 137219)
[PASS] test_mint_moreThanCap_reverts() (gas: 142523)
Test result: ok. 5 passed; 0 failed; finished in 2.26ms

Running 15 tests for contracts/test/L1CrossDomainMessenger.t.sol:L1CrossDomainMessenger_Test
[PASS] test_messageVersion_succeeds() (gas: 24738)
[PASS] test_relayMessage_legacyOldReplay_reverts() (gas: 49395)
[PASS] test_relayMessage_legacyRetryAfterFailureThenSuccess_reverts() (gas: 209744)
[PASS] test_relayMessage_legacyRetryAfterFailure_succeeds() (gas: 203642)
[PASS] test_relayMessage_legacyRetryAfterSuccess_reverts() (gas: 124002)
[PASS] test_relayMessage_legacy_succeeds() (gas: 77316)
[PASS] test_relayMessage_retryAfterFailure_succeeds() (gas: 197549)
[PASS] test_relayMessage_succeeds() (gas: 74252)
[PASS] test_relayMessage_toSystemContract_reverts() (gas: 56518)
[PASS] test_relayMessage_v2_reverts() (gas: 12365)
[PASS] test_replayMessage_withValue_reverts() (gas: 31063)
[PASS] test_sendMessage_succeeds() (gas: 200692)
[PASS] test_sendMessage_twice_succeeds() (gas: 609432)
[PASS] test_xDomainMessageSender_reset_succeeds() (gas: 84694)
[PASS] test_xDomainSender_notSet_reverts() (gas: 24253)
Test result: ok. 15 passed; 0 failed; finished in 15.99ms

Running 3 tests for contracts/test/MintManager.t.sol:MintManager_upgrade_Test
[PASS] test_upgrade_fromNotOwner_reverts() (gas: 10974)
[PASS] test_upgrade_fromOwner_succeeds() (gas: 23434)
[PASS] test_upgrade_toZeroAddress_reverts() (gas: 11003)
Test result: ok. 3 passed; 0 failed; finished in 1.36ms

Running 15 tests for contracts/test/L1ERC721Bridge.t.sol:L1ERC721Bridge_Test
[PASS] test_bridgeERC721To_localTokenZeroAddress_reverts() (gas: 52707)
[PASS] test_bridgeERC721To_remoteTokenZeroAddress_reverts() (gas: 27310)
[PASS] test_bridgeERC721To_succeeds() (gas: 251701)
[PASS] test_bridgeERC721To_wrongOwner_reverts() (gas: 60934)
```

```
[PASS] test_bridgeERC721_fromContract_reverts() (gas: 25666)
[PASS] test_bridgeERC721_localTokenZeroAddress_reverts() (gas: 50564)
[PASS] test_bridgeERC721_remoteTokenZeroAddress_reverts() (gas: 25124)
[PASS] test_bridgeERC721_succeeds() (gas: 249281)
[PASS] test_bridgeERC721_wrongOwner_reverts() (gas: 60830)
[PASS] test_constructor_succeeds() (gas: 10200)
[PASS] test_finalizeBridgeERC721_notEscrowed_reverts() (gas: 22119)
[PASS] test_finalizeBridgeERC721_notFromRemoteMessenger_reverts() (gas: 19797)
[PASS] test_finalizeBridgeERC721_notViaLocalMessenger_reverts() (gas: 16049)
[PASS] test_finalizeBridgeERC721_selfToken_reverts() (gas: 17615)
[PASS] test_finalizeBridgeERC721_succeeds() (gas: 220793)
Test result: ok. 15 passed; 0 failed; finished in 5.86ms

Running 11 tests for contracts/test/OptimismMintableERC20.t.sol:OptimismMintableERC20_Test
[PASS] test_bridge_succeeds() (gas: 7643)
[PASS] test_burn_notBridge_reverts() (gas: 11164)
[PASS] test_burn_succeeds() (gas: 50996)
[PASS] test_erc165_supportsInterface_succeeds() (gas: 7809)
[PASS] test_l1Token_succeeds() (gas: 7621)
[PASS] test_l2Bridge_succeeds() (gas: 7621)
[PASS] test_legacy_succeeds() (gas: 14344)
[PASS] test_mint_notBridge_reverts() (gas: 11121)
[PASS] test_mint_succeeds() (gas: 63566)
[PASS] test_remoteToken_succeeds() (gas: 7689)
[PASS] test_semver_succeeds() (gas: 8812)
Test result: ok. 11 passed; 0 failed; finished in 19.49ms

Running 1 test for
contracts/test/invariants/OptimismPortal.t.sol:OptimismPortal_CanAlwaysFinalizeAfterWindow
[PASS] invariant_canAlwaysFinalize() (runs: 256, calls: 3840, reverts: 3833)
Test result: ok. 1 passed; 0 failed; finished in 752.27ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_BridgeETHTo_Test
[PASS] test_bridgeETHTo_succeeds() (gas: 277629)
Test result: ok. 1 passed; 0 failed; finished in 42.70ms

Running 4 tests for contracts/test/OptimismMintableERC20Factory.t.sol:OptimismMintableTokenFactory_Test
[PASS] test_bridge_succeeds() (gas: 7580)
[PASS] test_createStandardL2Token_remoteIsZero_succeeds() (gas: 9390)
[PASS] test_createStandardL2Token_sameTwice_succeeds() (gas: 2523203)
[PASS] test_createStandardL2Token_succeeds() (gas: 1268564)
Test result: ok. 4 passed; 0 failed; finished in 10.51ms

Running 1 test for contracts/test/CrossDomainMessenger.t.sol:CrossDomainMessenger_RelayMessage_Test
[PASS] testFuzz_relayMessageReenter_succeeds(address,uint256) (runs: 256, μ: 622746, ~: 622461)
Test result: ok. 1 passed; 0 failed; finished in 549.20ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_BridgeETH_Test
[PASS] test_bridgeETH_succeeds() (gas: 271558)
Test result: ok. 1 passed; 0 failed; finished in 35.01ms

Running 7 tests for contracts/test/OptimismMintableERC721.t.sol:OptimismMintableERC721_Test
[PASS] test_burn_notBridge_reverts() (gas: 136966)
[PASS] test_burn_succeeds() (gas: 118832)
[PASS] test_constructor_succeeds() (gas: 29003)
[PASS] test_safeMint_notBridge_reverts() (gas: 11143)
[PASS] test_safeMint_succeeds() (gas: 140524)
[PASS] test_supportsInterfaces_succeeds() (gas: 9027)
[PASS] test_tokenURI_succeeds() (gas: 163441)
Test result: ok. 7 passed; 0 failed; finished in 9.88ms

Running 1 test for
contracts/test/invariants/L2OutputOracle.t.sol:L2OutputOracle_MonotonicBlockNumIncrease_Invariant
[PASS] invariant_monotonicBlockNumIncrease() (runs: 256, calls: 3840, reverts: 3840)
Test result: ok. 1 passed; 0 failed; finished in 502.45ms

Running 2 tests for contracts/test/CrossDomainOwnable.t.sol:CrossDomainOwnable_Test
[PASS] test_onlyOwner_notOwner_reverts() (gas: 10597)
[PASS] test_onlyOwner_succeeds() (gas: 34883)
Test result: ok. 2 passed; 0 failed; finished in 947.54μs

Running 2 tests for contracts/test/L2StandardBridge.t.sol:L2StandardBridge_BridgeERC20To_Test
```

```
[PASS] test_bridgeERC20To_succeeds() (gas: 390277)
[PASS] test_withdrawTo_withdrawingERC20_succeeds() (gas: 390510)
Test result: ok. 2 passed; 0 failed; finished in 7.65ms

Running 3 tests for contracts/test/OptimismMintableERC721Factory.t.sol:OptimismMintableERC721Factory_Test
[PASS] test_constructor_succeeds() (gas: 8285)
[PASS] test_createOptimismMintableERC721_succeeds() (gas: 2321842)
[PASS] test_createOptimismMintableERC721_zeroRemoteToken_reverts() (gas: 9418)
Test result: ok. 3 passed; 0 failed; finished in 4.17ms

Running 4 tests for contracts/test/CrossDomainOwnable2.t.sol:CrossDomainOwnable2_Test
[PASS] test_onlyOwner_notMessenger_reverts() (gas: 8416)
[PASS] test_onlyOwner_notOwner2_reverts() (gas: 57515)
[PASS] test_onlyOwner_notOwner_reverts() (gas: 16588)
[PASS] test_onlyOwner_succeeds() (gas: 73543)
Test result: ok. 4 passed; 0 failed; finished in 9.14ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_DepositERC20To_Test
[PASS] test_depositERC20To_succeeds() (gas: 516473)
Test result: ok. 1 passed; 0 failed; finished in 13.39ms

Running 5 tests for contracts/test/L2StandardBridge.t.sol:L2StandardBridge_BridgeERC20_Test
[PASS] test_bridgeERC20_succeeds() (gas: 385784)
[PASS] test_bridgeLegacyERC20_succeeds() (gas: 394056)
[PASS] test_withdrawLegacyERC20_succeeds() (gas: 394382)
[PASS] test_withdraw_notEOA_reverts() (gas: 251758)
[PASS] test_withdraw_withdrawingERC20_succeeds() (gas: 386012)
Test result: ok. 5 passed; 0 failed; finished in 44.87ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_DepositERC20_Test
[PASS] test_depositERC20_succeeds() (gas: 514174)
Test result: ok. 1 passed; 0 failed; finished in 8.77ms

Running 12 tests for contracts/test/CrossDomainOwnable3.t.sol:CrossDomainOwnable3_Test
[PASS] test_constructor_succeeds() (gas: 10554)
[PASS] test_crossDomainOnlyOwner_notMessenger_reverts() (gas: 28334)
[PASS] test_crossDomainOnlyOwner_notOwner2_reverts() (gas: 73991)
[PASS] test_crossDomainOnlyOwner_notOwner_reverts() (gas: 32000)
[PASS] test_crossDomainTransferOwnership_succeeds() (gas: 91548)
[PASS] test_localOnlyOwner_notOwner_reverts() (gas: 13193)
[PASS] test_localOnlyOwner_succeeds() (gas: 35220)
[PASS] test_localTransferOwnership_succeeds() (gas: 52128)
[PASS] test_transferOwnershipNoLocal_succeeds() (gas: 48610)
[PASS] test_transferOwnership_noLocalZeroAddress_reverts() (gas: 12015)
[PASS] test_transferOwnership_notOwner_reverts() (gas: 13437)
[PASS] test_transferOwnership_zeroAddress_reverts() (gas: 12081)
Test result: ok. 12 passed; 0 failed; finished in 3.85ms

Running 5 tests for contracts/test/L2StandardBridge.t.sol:L2StandardBridge_Bridge_Test
[PASS] test_finalizeBridgeETH_incorrectValue_reverts() (gas: 23843)
[PASS] test_finalizeBridgeETH_sendToMessenger_reverts() (gas: 23982)
[PASS] test_finalizeBridgeETH_sendToSelf_reverts() (gas: 23870)
[PASS] test_finalizeDeposit_depositingERC20_succeeds() (gas: 93824)
[PASS] test_finalizeDeposit_depositingETH_succeeds() (gas: 92700)
Test result: ok. 5 passed; 0 failed; finished in 26.66ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_DepositERC20_TestFail
[PASS] test_depositERC20_notEoa_reverts() (gas: 22320)
Test result: ok. 1 passed; 0 failed; finished in 34.59ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_DepositETHTo_Test
[PASS] test_depositETHTo_succeeds() (gas: 277706)
Test result: ok. 1 passed; 0 failed; finished in 7.70ms

Running 1 test for contracts/test/L2StandardBridge.t.sol:L2StandardBridge_FinalizeBridgeETH_Test
[PASS] test_finalizeBridgeETH_succeeds() (gas: 43155)
Test result: ok. 1 passed; 0 failed; finished in 7.24ms

Running 1 test for contracts/test/invariants/OptimismPortal.t.sol:OptimismPortal_CannotFinalizeTwice
[PASS] invariant_cannotFinalizeTwice() (runs: 256, calls: 3840, reverts: 3828)
Test result: ok. 1 passed; 0 failed; finished in 499.84ms
```

```
Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_DepositETH_Test
[PASS] test_depositETH_succeeds() (gas: 271652)
Test result: ok. 1 passed; 0 failed; finished in 17.16ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_FinalizeBridgeETH_Test
[PASS] test_finalizeBridgeETH_succeeds() (gas: 51674)
Test result: ok. 1 passed; 0 failed; finished in 24.55ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_DepositETH_TestFail
[PASS] test_depositETH_notEoa_reverts() (gas: 40780)
Test result: ok. 1 passed; 0 failed; finished in 6.17ms

Running 3 tests for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_FinalizeBridgeETH_TestFail
[PASS] test_finalizeBridgeETH_incorrectValue_reverts() (gas: 34204)
[PASS] test_finalizeBridgeETH_sendToMessenger_reverts() (gas: 34310)
[PASS] test_finalizeBridgeETH_sendToSelf_reverts() (gas: 34279)
Test result: ok. 3 passed; 0 failed; finished in 8.13ms

Running 1 test for contracts/test/L1StandardBridge.t.sol:L1StandardBridge_FinalizeERC20Withdrawal_Test
[PASS] test_finalizeERC20Withdrawal_succeeds() (gas: 496128)
Test result: ok. 1 passed; 0 failed; finished in 9.62ms

Running 2 tests for
contracts/test/L1StandardBridge.t.sol:L1StandardBridge_FinalizeERC20Withdrawal_TestFail
[PASS] test_finalizeERC20Withdrawal_notMessenger_reverts() (gas: 31206)
[PASS] test_finalizeERC20Withdrawal_notOtherBridge_reverts() (gas: 31562)
Test result: ok. 2 passed; 0 failed; finished in 17.74ms

Running 1 test for contracts/test/invariants/OptimismPortal.t.sol:OptimismPortal_CannotTimeTravel
[PASS] invariant_cannotFinalizeBeforePeriodHasPassed() (runs: 256, calls: 3840, reverts: 3836)
Test result: ok. 1 passed; 0 failed; finished in 413.82ms

Running 1 test for contracts/test/OptimismPortal.t.sol:OptimismPortalResourceFuzz_Test
[PASS]
testFuzz_systemConfigDeposit_succeeds(uint32,uint8,uint8,uint32,uint32,uint128,uint64,uint64,uint128,uint
8) (runs: 256, μ: 229951, ~: 50726)
Test result: ok. 1 passed; 0 failed; finished in 1.35s

Running 1 test for contracts/test/Hashing.t.sol:Hashing_hashDepositSource_Test
[PASS] test_hashDepositSource_succeeds() (gas: 633)
Test result: ok. 1 passed; 0 failed; finished in 1.28ms

Running 1 test for contracts/test/invariants/CrossDomainMessenger.t.sol:XDM_MinGasLimits_Reverts
[PASS] invariant_minGasLimits() (runs: 256, calls: 3840, reverts: 0)
Test result: ok. 1 passed; 0 failed; finished in 4.75s

Running 7 tests for contracts/test/L1Block.t.sol:L1BlockTest
[PASS] testFuzz_updatesValues_succeeds(uint64,uint64,uint256,bytes32,uint64,bytes32,uint256,uint256)
(runs: 256, μ: 69650, ~: 70615)
[PASS] test_basefee_succeeds() (gas: 7554)
[PASS] test_hash_succeeds() (gas: 7576)
[PASS] test_number_succeeds() (gas: 7629)
[PASS] test_sequenceNumber_succeeds() (gas: 7630)
[PASS] test_timestamp_succeeds() (gas: 7640)
[PASS] test_updateValues_succeeds() (gas: 60482)
Test result: ok. 7 passed; 0 failed; finished in 21.90ms

Running 3 tests for contracts/test/L1BlockNumber.t.sol:L1BlockNumberTest
[PASS] test_fallback_succeeds() (gas: 18655)
[PASS] test_getL1BlockNumber_succeeds() (gas: 10625)
[PASS] test_receive_succeeds() (gas: 25384)
Test result: ok. 3 passed; 0 failed; finished in 6.74ms

Running 1 test for contracts/test/CrossDomainOwnable.t.sol:CrossDomainOwnableThroughPortal_Test
[PASS] test_depositTransaction_crossDomainOwner_succeeds() (gas: 68931)
Test result: ok. 1 passed; 0 failed; finished in 6.74ms

Running 1 test for contracts/test/invariants/CrossDomainMessenger.t.sol:XDM_MinGasLimits_Succeeds
[PASS] invariant_minGasLimits() (runs: 256, calls: 3840, reverts: 0)
Test result: ok. 1 passed; 0 failed; finished in 5.65s

Running 1 test for contracts/test/MintManager.t.sol:MintManager_constructor_Test
```

```
[PASS] test_constructor_succeeds() (gas: 10579)
Test result: ok. 1 passed; 0 failed; finished in 1.37ms

Running 2 tests for contracts/test/Hashing.t.sol:Hashing_hashCrossDomainMessage_Test
[PASS] testDiff_hashCrossDomainMessage_succeeds(uint240,uint16,address,address,uint256,uint256,bytes)
(runs: 256, μ: 27746, ~: 27558)
[PASS] testFuzz_hashCrossDomainMessageV0_matchesLegacy_succeeds(address,address,bytes,uint256) (runs:
256, μ: 2413, ~: 2309)
Test result: ok. 2 passed; 0 failed; finished in 32.45s

Running 3 tests for contracts/test/ResolvedDelegateProxy.t.sol:ResolvedDelegateProxy_Test
[PASS] testFuzz_fallback_delegateCallFoo_succeeds(uint256) (runs: 256, μ: 24648, ~: 24648)
[PASS] test_fallback_addressManagerNotSet_reverts() (gas: 605906)
[PASS] test_fallback_delegateCallBar_reverts() (gas: 24783)
Test result: ok. 3 passed; 0 failed; finished in 17.33ms

Running 1 test for contracts/test/Hashing.t.sol:Hashing_hashOutputRootProof_Test
[PASS] testDiff_hashOutputRootProof_succeeds(bytes32,bytes32,bytes32,bytes32) (runs: 256, μ: 66273, ~:
92771)
Test result: ok. 1 passed; 0 failed; finished in 31.80s

Running 1 test for contracts/test/Hashing.t.sol:Hashing_hashDepositTransaction_Test
[PASS] testDiff_hashDepositTransaction_succeeds(address,address,uint256,uint256,uint64,bytes,uint64)
(runs: 256, μ: 65950, ~: 65264)
Test result: ok. 1 passed; 0 failed; finished in 32.18s

Running 11 tests for contracts/test/Proxy.t.sol:Proxy_Test
[PASS] test_delegatesToImpl_succeeds() (gas: 45207)
[PASS] test_implementationKey_succeeds() (gas: 20909)
[PASS] test_implementation_isZeroAddress_reverts() (gas: 47626)
[PASS] test_implementation_zeroAddressCaller_succeeds() (gas: 14752)
[PASS] test_ownerKey_succeeds() (gas: 19059)
[PASS] test_ownerProxyCall_notAdmin_succeeds() (gas: 34615)
[PASS] test_proxyCallToImp_notAdmin_succeeds() (gas: 30008)
[PASS] test_upgradeToAndCall_functionDoesNotExist_reverts() (gas: 104565)
[PASS] test_upgradeToAndCall_isPayable_succeeds() (gas: 53742)
[PASS] test_upgradeToAndCall_succeeds() (gas: 125190)
[PASS] test_upgradeTo_clashingFunctionSignatures_succeeds() (gas: 101359)
Test result: ok. 11 passed; 0 failed; finished in 6.62ms

Running 23 tests for contracts/test/ProxyAdmin.t.sol:ProxyAdmin_Test
[PASS] test_chugsplashChangeProxyAdmin_succeeds() (gas: 35586)
[PASS] test_chugsplashGetProxyAdmin_succeeds() (gas: 15675)
[PASS] test_chugsplashGetProxyImplementation_succeeds() (gas: 51084)
[PASS] test_chugsplashUpgradeAndCall_succeeds() (gas: 82311)
[PASS] test_chugsplashUpgrade_succeeds() (gas: 48988)
[PASS] test_delegateResolvedChangeProxyAdmin_succeeds() (gas: 33936)
[PASS] test_delegateResolvedGetProxyAdmin_succeeds() (gas: 17691)
[PASS] test_delegateResolvedGetProxyImplementation_succeeds() (gas: 62028)
[PASS] test_delegateResolvedUpgradeAndCall_succeeds() (gas: 98039)
[PASS] test_delegateResolvedUpgrade_succeeds() (gas: 58482)
[PASS] test_erc1967ChangeProxyAdmin_succeeds() (gas: 33812)
[PASS] test_erc1967GetProxyAdmin_succeeds() (gas: 15616)
[PASS] test_erc1967GetProxyImplementation_succeeds() (gas: 52071)
[PASS] test_erc1967UpgradeAndCall_succeeds() (gas: 78969)
[PASS] test_erc1967Upgrade_succeeds() (gas: 50078)
[PASS] test_isUpgrading_succeeds() (gas: 19442)
[PASS] test_onlyOwner_notOwner_reverts() (gas: 22767)
[PASS] test_owner_succeeds() (gas: 9738)
[PASS] test_proxyType_succeeds() (gas: 20533)
[PASS] test_setAddressManager_notOwner_reverts() (gas: 10578)
[PASS] test_setImplementationName_notOwner_reverts() (gas: 11111)
[PASS] test_setImplementationName_succeeds() (gas: 38945)
[PASS] test_setProxyType_notOwner_reverts() (gas: 10814)
Test result: ok. 23 passed; 0 failed; finished in 2.10ms

Running 8 tests for contracts/test/RLPReader.t.sol:RLPReader_readBytes_Test
[PASS] test_readBytes_bytestring00_succeeds() (gas: 1878)
[PASS] test_readBytes_bytestring01_succeeds() (gas: 1855)
[PASS] test_readBytes_bytestring7f_succeeds() (gas: 1876)
[PASS] test_readBytes_invalidListLength_reverts() (gas: 3903)
[PASS] test_readBytes_invalidPrefix_reverts() (gas: 3961)
```

```
[PASS] test_readBytes_invalidRemainder_reverts() (gas: 4155)
[PASS] test_readBytes_invalidStringLength_reverts() (gas: 3857)
[PASS] test_readBytes_revertListItem_reverts() (gas: 3998)
Test result: ok. 8 passed; 0 failed; finished in 1.95ms

Running 28 tests for contracts/test/RLPReader.t.sol:RLPReader_readList_Test
[PASS] test_readList_dictTest1_succeeds() (gas: 23202)
[PASS] test_readList_empty_succeeds() (gas: 4612)
[PASS] test_readList_incorrectLengthInArray_reverts() (gas: 3976)
[PASS] test_readList_int32Overflow2_reverts() (gas: 4139)
[PASS] test_readList_int32Overflow_reverts() (gas: 4138)
[PASS] test_readList_invalidRemainder_reverts() (gas: 4114)
[PASS] test_readList_invalidShortList_reverts() (gas: 3967)
[PASS] test_readList_invalidValue_reverts() (gas: 3878)
[PASS] test_readList_leadingZerosInLongLengthArray1_reverts() (gas: 3982)
[PASS] test_readList_leadingZerosInLongLengthArray2_reverts() (gas: 3945)
[PASS] test_readList_leadingZerosInLongLengthList1_reverts() (gas: 3984)
[PASS] test_readList_listLongerThan32Elements_reverts() (gas: 38615)
[PASS] test_readList_listOfLists2_succeeds() (gas: 12169)
[PASS] test_readList_listOfLists_succeeds() (gas: 9504)
[PASS] test_readList_longList1_succeeds() (gas: 28416)
[PASS] test_readList_longList2_succeeds() (gas: 196834)
[PASS] test_readList_longListLessThan56Bytes_reverts() (gas: 4023)
[PASS] test_readList_longStringLength_reverts() (gas: 3946)
[PASS] test_readList_longStringLessThan56Bytes_reverts() (gas: 4009)
[PASS] test_readList_multiList_succeeds() (gas: 11742)
[PASS] test_readList_nonOptimalLongLengthArray1_reverts() (gas: 3999)
[PASS] test_readList_nonOptimalLongLengthArray2_reverts() (gas: 4044)
[PASS] test_readList_notEnoughContentForList1_reverts() (gas: 4115)
[PASS] test_readList_notEnoughContentForList2_reverts() (gas: 4117)
[PASS] test_readList_notEnoughContentForString1_reverts() (gas: 4072)
[PASS] test_readList_notEnoughContentForString2_reverts() (gas: 4094)
[PASS] test_readList_notLongEnough_reverts() (gas: 3955)
[PASS] test_readList_shortListMax1_succeeds() (gas: 39769)
Test result: ok. 28 passed; 0 failed; finished in 3.44ms

Running 9 tests for contracts/test/RLPWriter.t.sol:RLPWriter_writeList_Test
[PASS] test_writeList_dictTest1_succeeds() (gas: 37134)
[PASS] test_writeList_empty_succeeds() (gas: 1721)
[PASS] test_writeList_listoflists2_succeeds() (gas: 16656)
[PASS] test_writeList_listoflists_succeeds() (gas: 10901)
[PASS] test_writeList_longlist1_succeeds() (gas: 40489)
[PASS] test_writeList_longlist2_succeeds() (gas: 281280)
[PASS] test_writeList_multiList_succeeds() (gas: 22546)
[PASS] test_writeList_shortListMax1_succeeds() (gas: 36918)
[PASS] test_writeList_stringList_succeeds() (gas: 10742)
Test result: ok. 9 passed; 0 failed; finished in 3.19ms

Running 8 tests for contracts/test/RLPWriter.t.sol:RLPWriter_writeString_Test
[PASS] test_writeString_bytestring00_succeeds() (gas: 976)
[PASS] test_writeString_bytestring01_succeeds() (gas: 976)
[PASS] test_writeString_bytestring7f_succeeds() (gas: 997)
[PASS] test_writeString_empty_succeeds() (gas: 1643)
[PASS] test_writeString_longstring2_succeeds() (gas: 258779)
[PASS] test_writeString_longstring_succeeds() (gas: 16972)
[PASS] test_writeString_shortstring2_succeeds() (gas: 15386)
[PASS] test_writeString_shortstring_succeeds() (gas: 2480)
Test result: ok. 8 passed; 0 failed; finished in 2.32ms

Running 8 tests for contracts/test/RLPWriter.t.sol:RLPWriter_writeUint_Test
[PASS] test_writeUint_mediumint2_succeeds() (gas: 8736)
[PASS] test_writeUint_mediumint3_succeeds() (gas: 9113)
[PASS] test_writeUint_mediumint_succeeds() (gas: 8372)
[PASS] test_writeUint_smallint2_succeeds() (gas: 7279)
[PASS] test_writeUint_smallint3_succeeds() (gas: 7301)
[PASS] test_writeUint_smallint4_succeeds() (gas: 7302)
[PASS] test_writeUint_smallint_succeeds() (gas: 7280)
[PASS] test_writeUint_zero_succeeds() (gas: 7749)
Test result: ok. 8 passed; 0 failed; finished in 3.21ms

Running 5 tests for contracts/test/SequencerFeeVault.t.sol:SequencerFeeVault_Test
[PASS] test_constructor_succeeds() (gas: 5526)
```

```
[PASS] test_minWithdrawalAmount_succeeds() (gas: 5442)
[PASS] test_receive_succeeds() (gas: 17373)
[PASS] test_withdraw_notEnough_reverts() (gas: 9331)
[PASS] test_withdraw_succeeds() (gas: 163543)
Test result: ok. 5 passed; 0 failed; finished in 8.74ms

Running 2 tests for contracts/test/StandardBridge.t.sol:StandardBridge_Stateless_Test
[PASS] test_isCorrectTokenPair_succeeds() (gas: 49936)
[PASS] test_isOptimismMintableERC20_succeeds() (gas: 33072)
Test result: ok. 2 passed; 0 failed; finished in 5.31ms

Running 21 tests for contracts/test/OptimismPortal.t.sol:OptimismPortal_FinalizeWithdrawal_Test
[PASS] testDiff_finalizeWithdrawalTransaction_succeeds(address,address,uint256,uint256,bytes) (runs: 256,
µ: 246321, ~: 247908)
[PASS] test_finalizeWithdrawalTransaction_ifOutputRootChanges_reverts() (gas: 204022)
[PASS] test_finalizeWithdrawalTransaction_ifOutputTimestampIsNotFinalized_reverts() (gas: 207475)
[PASS] test_finalizeWithdrawalTransaction_ifWithdrawalNotProven_reverts() (gas: 41731)
[PASS] test_finalizeWithdrawalTransaction_ifWithdrawalProofNotOldEnough_reverts() (gas: 199419)
[PASS] test_finalizeWithdrawalTransaction_onInsufficientGas_reverts() (gas: 205848)
[PASS] test_finalizeWithdrawalTransaction_onRecentWithdrawal_reverts() (gas: 180184)
[PASS] test_finalizeWithdrawalTransaction_onReentrancy_reverts() (gas: 243823)
[PASS] test_finalizeWithdrawalTransaction_onReplay_reverts() (gas: 245561)
[PASS] test_finalizeWithdrawalTransaction_paused_reverts() (gas: 53510)
[PASS] test_finalizeWithdrawalTransaction_provenWithdrawalHash_succeeds() (gas: 234996)
[PASS] test_finalizeWithdrawalTransaction_targetFails_fails() (gas: 8797746687696163867)
[PASS] test_finalizeWithdrawalTransaction_timestampLessThanL2OracleStart_reverts() (gas: 196997)
[PASS] test_proveWithdrawalTransaction_onInvalidOutputRootProof_reverts() (gas: 85690)
[PASS] test_proveWithdrawalTransaction_onInvalidWithdrawalProof_reverts() (gas: 137350)
[PASS] test_proveWithdrawalTransaction_onSelfCall_reverts() (gas: 52947)
[PASS] test_proveWithdrawalTransaction_paused_reverts() (gas: 73673)
[PASS] test_proveWithdrawalTransaction_replayProveChangedOutputRootAndOutputIndex_succeeds() (gas:
346739)
[PASS] test_proveWithdrawalTransaction_replayProveChangedOutputRoot_succeeds() (gas: 279571)
[PASS] test_proveWithdrawalTransaction_replayProve_reverts() (gas: 192548)
[PASS] test_proveWithdrawalTransaction_validWithdrawalProof_succeeds() (gas: 180486)
Test result: ok. 21 passed; 0 failed; finished in 33.37s

Running 21 tests for contracts/test/OptimismPortal.t.sol:OptimismPortal_Test
[PASS] testFuzz_depositTransaction_smallGasLimit_succeeds(bytes,bool) (runs: 256, µ: 30889, ~: 34460)
[PASS] test_constructor_succeeds() (gas: 19402)
[PASS] test_depositTransaction_contractCreation_reverts() (gas: 14342)
[PASS] test_depositTransaction_createWithZeroValueForContract_succeeds() (gas: 44289)
[PASS] test_depositTransaction_createWithZeroValueForEOA_succeeds() (gas: 44655)
[PASS] test_depositTransaction_largeData_reverts() (gas: 512221)
[PASS] test_depositTransaction_noValueContract_succeeds() (gas: 44307)
[PASS] test_depositTransaction_noValueEOA_succeeds() (gas: 44652)
[PASS] test_depositTransaction_smallGasLimit_reverts() (gas: 14556)
[PASS] test_depositTransaction_withEthValueAndContractContractCreation_succeeds() (gas: 51313)
[PASS] test_depositTransaction_withEthValueAndEOAContractCreation_succeeds() (gas: 42953)
[PASS] test_depositTransaction_withEthValueFromContract_succeeds() (gas: 50993)
[PASS] test_depositTransaction_withEthValueFromEOA_succeeds() (gas: 51609)
[PASS] test_isOutputFinalized_succeeds() (gas: 121831)
[PASS] test_minimumGasLimit_succeeds() (gas: 17650)
[PASS] test_pause_onlyGuardian_reverts() (gas: 22196)
[PASS] test_pause_succeeds() (gas: 42175)
[PASS] test_receive_succeeds() (gas: 60878)
[PASS] test_simple_isOutputFinalized_succeeds() (gas: 32971)
[PASS] test_unpause_onlyGuardian_reverts() (gas: 46098)
[PASS] test_unpause_succeeds() (gas: 31756)
Test result: ok. 21 passed; 0 failed; finished in 29.30ms

Running 1 test for contracts/test/invariants/SystemConfig.t.sol:SystemConfig_GasLimitLowerBound_Invariant
[PASS] invariant_gasLimitLowerBound() (runs: 256, calls: 3840, reverts: 1823)
Test result: ok. 1 passed; 0 failed; finished in 234.66ms

Running 1 test for contracts/test/SystemConfig.t.sol:SystemConfig_Initialize_TestFail
[PASS] test_initialize_lowGasLimit_reverts() (gas: 148848)
Test result: ok. 1 passed; 0 failed; finished in 1.25ms

Running 4 tests for contracts/test/SystemConfig.t.sol:SystemConfig_Setters_Test
[PASS] testFuzz_setBatcherHash_succeeds(bytes32) (runs: 256, µ: 23189, ~: 23189)
[PASS] testFuzz_setGasConfig_succeeds(uint256,uint256) (runs: 256, µ: 29509, ~: 29772)
```

```
[PASS] testFuzz_setGasLimit_succeeds(uint64) (runs: 256, μ: 31076, ~: 30989)
[PASS] testFuzz_setUnsafeBlockSigner_succeeds(address) (runs: 256, μ: 23381, ~: 23381)
Test result: ok. 4 passed; 0 failed; finished in 78.97ms

Running 9 tests for contracts/test/SystemConfig.t.sol:SystemConfig_Setters_TestFail
[PASS] test_setBatcherHash_notOwner_reverts() (gas: 10546)
[PASS] test_setGasConfig_notOwner_reverts() (gas: 10622)
[PASS] test_setGasLimit_notOwner_reverts() (gas: 10615)
[PASS] test_setResourceConfig_badMinMax_reverts() (gas: 13002)
[PASS] test_setResourceConfig_badPrecision_reverts() (gas: 15603)
[PASS] test_setResourceConfig_lowGasLimit_reverts() (gas: 16082)
[PASS] test_setResourceConfig_notOwner_reverts() (gas: 11790)
[PASS] test_setResourceConfig_zeroDenominator_reverts() (gas: 13039)
[PASS] test_setUnsafeBlockSigner_notOwner_reverts() (gas: 10616)
Test result: ok. 9 passed; 0 failed; finished in 15.39ms

Running 2 tests for contracts/test/TransferOnion.t.sol:TransferOnionTest
[PASS] test_constructor_succeeds() (gas: 564855)
[PASS] test_unwrap_succeeds() (gas: 724958)
Test result: ok. 2 passed; 0 failed; finished in 1.14ms

Running 1 test for contracts/test/invariants/SafeCall.t.sol:SafeCall_Fails_Invariants
[PASS] invariant_callWithMinGas_neverForwardsMinGas_reverts() (runs: 256, calls: 3840, reverts: 3840)
Test result: ok. 1 passed; 0 failed; finished in 564.32ms

Running 1 test for contracts/test/Hashing.t.sol:Hashing_hashWithdrawal_Test
[PASS] testDiff_hashWithdrawal_succeeds(uint256,address,address,uint256,uint256,bytes) (runs: 256, μ:
22990, ~: 22793)
Test result: ok. 1 passed; 0 failed; finished in 32.01s

Running 1 test for contracts/test/invariants/SafeCall.t.sol:SafeCall_Succeeds_Invariants
[PASS] invariant_callWithMinGas_alwaysForwardsMinGas_succeeds() (runs: 256, calls: 3840, reverts: 1)
Test result: ok. 1 passed; 0 failed; finished in 714.65ms

Running 10 tests for contracts/test/ResourceMetering.t.sol:ResourceMetering_Test
[PASS] testFuzz_meter_largeBlockDiff_succeeds(uint64,uint256) (runs: 256, μ: 100921, ~: 24988)
[PASS] test_meter_denominatorEq1_reverts() (gas: 6690712)
[PASS] test_meter_initialResourceParams_succeeds() (gas: 12423)
[PASS] test_meter_updateNoGasDelta_succeeds() (gas: 678335)
[PASS] test_meter_updateOneEmptyBlock_succeeds() (gas: 20810)
[PASS] test_meter_updateParamsNoChange_succeeds() (gas: 17049)
[PASS] test_meter_updateTenEmptyBlocks_succeeds() (gas: 23663)
[PASS] test_meter_updateTwoEmptyBlocks_succeeds() (gas: 23619)
[PASS] test_meter_useMax_succeeds() (gas: 6687380)
[PASS] test_meter_useMoreThanMax_reverts() (gas: 19549)
Test result: ok. 10 passed; 0 failed; finished in 192.19ms

Running 2 tests for contracts/test/Semver.t.sol:Semver_Test
[PASS] test_behindProxy_succeeds() (gas: 506748)
[PASS] test_version_succeeds() (gas: 9418)
Test result: ok. 2 passed; 0 failed; finished in 474.29μs

Running 5 tests for contracts/test/SafeCall.t.sol:SafeCall_Test
[PASS] testFuzz_callWithMinGas_hasEnough_succeeds(address,address,uint64,uint64,bytes) (runs: 256, μ:
49203, ~: 50807)
[PASS] testFuzz_call_succeeds(address,address,uint256,uint64,bytes) (runs: 256, μ: 45032, ~: 46491)
[PASS] testFuzz_send_succeeds(address,address,uint256,uint64) (runs: 256, μ: 44446, ~: 46066)
[PASS] test_callWithMinGas_noLeakageHigh_succeeds() (gas: 1021670598)
[PASS] test_callWithMinGas_noLeakageLow_succeeds() (gas: 1095190710)
Test result: ok. 5 passed; 0 failed; finished in 2.93s

Running 5 tests for contracts/test/Encoding.t.sol:Encoding_Test
[PASS] testDiff_decodeVersionedNonce_succeeds(uint240,uint16) (runs: 256, μ: 12717, ~: 12734)
[PASS] testDiff_encodeCrossDomainMessage_succeeds(uint240,uint8,address,address,uint256,uint256,bytes)
(runs: 256, μ: 88468, ~: 91153)
[PASS]
testDiff_encodeDepositTransaction_succeeds(address,address,uint256,uint256,uint64,bool,bytes,uint64)
(runs: 256, μ: 105596, ~: 101844)
[PASS] testFuzz_encodeCrossDomainMessageV0_matchesLegacy_succeeds(uint240,address,address,bytes) (runs:
256, μ: 55696, ~: 50691)
[PASS] testFuzz_nonceVersioning_succeeds(uint240,uint16) (runs: 256, μ: 702, ~: 702)
Test result: ok. 5 passed; 0 failed; finished in 52.08s
```

```
Running 1 test for contracts/test/ResourceMetering.t.sol:ArtifactResourceMetering_Test
[PASS] test_meter_generateArtifact_succeeds() (gas: 4806896538)
Test result: ok. 1 passed; 0 failed; finished in 20.10s

Running 31 tests for contracts/test/MerkleTrie.t.sol:MerkleTrie_get_Test
[PASS] testFuzz_get_corruptedProof_reverts(bytes4) (runs: 256, μ: 80777, ~: 85712)
[PASS] testFuzz_get_emptyKey_reverts(bytes4) (runs: 256, μ: 33149, ~: 32987)
[PASS] testFuzz_get_extraProofElements_reverts(bytes4) (runs: 256, μ: 193061, ~: 192874)
[PASS] testFuzz_get_invalidDataRemainder_reverts(bytes4) (runs: 256, μ: 81907, ~: 79240)
[PASS] testFuzz_get_invalidInternalNodeHash_reverts(bytes4) (runs: 256, μ: 138305, ~: 142059)
[PASS] testFuzz_get_invalidLargeInternalHash_reverts(bytes4) (runs: 256, μ: 141493, ~: 145111)
[PASS] testFuzz_get_invalidRoot_reverts(bytes4) (runs: 256, μ: 140200, ~: 143234)
[PASS] testFuzz_get_partialProof_reverts(bytes4) (runs: 256, μ: 98600, ~: 103808)
[PASS] testFuzz_get_prefixedValidKey_reverts(bytes4) (runs: 256, μ: 146116, ~: 147936)
[PASS] testFuzz_get_validProofs_succeeds(bytes4) (runs: 256, μ: 296901, ~: 293918)
[PASS] test_get_corruptedProof_reverts() (gas: 5736)
[PASS] test_get_extraProofElements_reverts() (gas: 60631)
[PASS] test_get_invalidDataRemainder_reverts() (gas: 35852)
[PASS] test_get_invalidInternalNodeHash_reverts() (gas: 50810)
[PASS] test_get_nonexistentKey1_reverts() (gas: 59671)
[PASS] test_get_nonexistentKey2_reverts() (gas: 23385)
[PASS] test_get_smallerPathThanKey1_reverts() (gas: 53525)
[PASS] test_get_smallerPathThanKey2_reverts() (gas: 55006)
[PASS] test_get_validProof10_succeeds() (gas: 50593)
[PASS] test_get_validProof1_succeeds() (gas: 61688)
[PASS] test_get_validProof2_succeeds() (gas: 71579)
[PASS] test_get_validProof3_succeeds() (gas: 32827)
[PASS] test_get_validProof4_succeeds() (gas: 23623)
[PASS] test_get_validProof5_succeeds() (gas: 84262)
[PASS] test_get_validProof6_succeeds() (gas: 73021)
[PASS] test_get_validProof7_succeeds() (gas: 79719)
[PASS] test_get_validProof8_succeeds() (gas: 50550)
[PASS] test_get_validProof9_succeeds() (gas: 50550)
[PASS] test_get_wrongKeyProof_reverts() (gas: 53871)
[PASS] test_get_zeroBranchValueLength_reverts() (gas: 43248)
[PASS] test_get_zeroLengthKey_reverts() (gas: 3632)
Test result: ok. 31 passed; 0 failed; finished in 58.68s
✨  Done in 150.44s.
```

Changelog

- 2023-10-02 - Initial report
- 2023-10-27 - Fix review update

About Quantstamp

Quantstamp is a global leader in blockchain security. Founded in 2017, Quantstamp's mission is to securely onboard the next billion users to Web3 through its best-in-class Web3 security products and services.

Quantstamp's team consists of cybersecurity experts hailing from globally recognized organizations including Microsoft, AWS, BMW, Meta, and the Ethereum Foundation. Quantstamp engineers hold PhDs or advanced computer science degrees, with decades of combined experience in formal verification, static analysis, blockchain audits, penetration testing, and original leading-edge research.

To date, Quantstamp has performed more than 500 audits and secured over $200 billion in digital asset risk from hackers. Quantstamp has worked with a diverse range of customers, including startups, category leaders and financial institutions. Brands that Quantstamp has worked with include Ethereum 2.0, Binance, Visa, PayPal, Polygon, Avalanche, Curve, Solana, Compound, Lido, MakerDAO, Arbitrum, OpenSea and the World Economic Forum.

Quantstamp's collaborations and partnerships showcase our commitment to world-class research, development and security. We're honored to work with some of the top names in the industry and proud to secure the future of web3.

Notable Collaborations & Customers:

- Blockchains: Ethereum 2.0, Near, Flow, Avalanche, Solana, Cardano, Binance Smart Chain, Hedera Hashgraph, Tezos
- DeFi: Curve, Compound, Maker, Lido, Polygon, Arbitrum, SushiSwap
- NFT: OpenSea, Parallel, Dapper Labs, Decentraland, Sandbox, Axie Infinity, Illuvium, NBA Top Shot, Zora
- Academic institutions: National University of Singapore, MIT

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by Quantstamp; however, Quantstamp does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication or other making available of the report to you by Quantstamp.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites&aspo; owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on any website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any output generated by such software.

Disclaimer

The review and this report are provided on an as-is, where-is, and as-available basis. To the fullest extent permitted by law, Quantstamp disclaims all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. You agree that your access and/or use of the report and other results of the review, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE. This report is based on the scope of materials and documentation provided for a limited review at the time provided. You acknowledge that Blockchain technology remains under development and is subject to unknown risks and flaws and, as such, the report may not be complete or inclusive of all vulnerabilities. The review is limited to the materials identified in the report and does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. The report does not indicate the endorsement by Quantstamp of any particular project or team, nor guarantee its security, and may not be represented as such. No third party is entitled to rely on the report in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. Quantstamp does not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party, or any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, or any related services and products, any hyperlinked websites, or any other websites or mobile applications, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third party. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate.