mev-boost improvement proposal #0

Exclude withdrawals to fee recipient when computing a payload's value.

Context

mev-boost today selects the highest bid by value from a set of bids provided by connected relays in a given slot. This value represents the amount of ETH a builder transfers to a proposer in exchange for the sale of that proposer's execution block space.

Relays compute this value in one of two ways, referred to as "payment schemes":

PS1. The value of the last transaction in the execution payload, whichmust exactly match the value claimed by the builder in their submission.

PS2. Comparing the balance of the fee_recipient in the post-state after the execution payload to the pre-state amount.

(2) becomes problematic when a given block contains consensus-layer withdrawals to the samefee_recipient the builder intends to pay as this inflates the payload's value. This inflation misreports the amount of MEV in a block which:

1. weakens builder and relay accountability by complicating public analysis
2. lowers revenues to proposers by masking higher MEV payloads
3. weakens the ability of censorship-resistance techniques like use of themin-bid flag in mev-boost

The inclusion of withdrawals with respect to execution is a bit of a quirk of block processing and given their deterministic nature in a given slot are easy to compute and subsequently remove from any bid scoring computation.

For a recent example, see this tweet reporting mev-boost bids where a bid's value was ~320.17 ETH and almost all of this was from withdrawals to the proposer in the same block

For a more through exploration of this phenomenon, seethis Dune dashboard by @Data_Always.

Proposal

| Constant | Value|

|----------|------|

|WITHDRAWALS_SCORING_ACTIVATION_EPOCH| **TBD** |

If the current epoch is greater or equal to theWITHDRAWALS_SCORING_ACTIVATION_EPOCH, relays **MUST** validate a block's value in the following way:

If an incoming builder's bid is using **PS1**, no change is required. The value of the bid is validated to*exactly* match the value of the final transaction in the payload. Note this indirectly excludes any withdrawal amounts by design.

If an incoming builder's bid is using **PS2**, sum all withdrawals in the block's slot to the payload's fee recipient. This total withdrawal amount is subtracted from the final balance difference to the fee recipient.

In pseudo-code following definitions in the consensus-specs:

```python
def verify_bid_value(execution_payload, fee_recipient, bid_value, balance_difference):
    target_amounts = [w.amount for w in execution_payload.withdrawals if w.address == fee_recipient]
    excluded_amount = sum(target_amounts)
    proposer_payment = balance_difference - excluded_amount
    assert proposer_payment == bid_value
```

verify_bid_value should execute completely without error. Theexecution_payload, fee_recipient, and bid_value are all provided by the builder in their payload submission. The balance_difference is computed by the relay during simulation of theexecution_payload where balance_difference = post_state_balance - pre_state_balance . pre_state_balance is the ether amount at thefee_recipient's address in the execution state before applying the execution_payload and the post_state_balance is the same data after applying the

execution_payload.

Considerations and comments

Performance

The withdrawals for a given block can be computed as soon as the parent is known, so the `excluded_amount` above can be computed by the relay as soon as the builder's intended proposer fee recipient is known. An additional optimization would compute all possible `excluded_amounts` (n.b. there can only be 16 distinct withdrawal recipients in a given block) even before the builder has provided any payload so that the implementation of `verify_bid_value` can consist of a simple lookup in a table if the selection and sum approach above is not sufficient.

Compatibility with optimistic relaying

There is no negative impact on optimistic relaying. Optimistic relays have even more time to compute the expected bid value and can demote builders from the optimistic setting if they misrepresent their bid value due to withdrawals.

Lack of bid distortion

As noted above, withdrawals are fixed in a given slot as determined by consensus so there is no possibility for this change to distort a bid's possible value as found in more general payment schemes across any combination of builders, relays, and proposers.