A few words...
The contract call From 0x68b3465833fb72a70e... To 0x68b3465833fb72a70e... produced 11 Internal Transactions

<table>
<thead>
<tr>
<th>Type Trace Address</th>
<th>From</th>
<th>To</th>
<th>Value</th>
<th>Gas Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>delegatecall_0_1</td>
<td>0x68b3465833fb72a70e...</td>
<td>0x68b3465833fb72a70e...</td>
<td>0 Ether</td>
<td>280,744</td>
</tr>
<tr>
<td>call_0_1_1</td>
<td>0x68b3465833fb72a70e...</td>
<td>0xc02aa9b223fe6d0a...</td>
<td>0.152282790805995 Ether</td>
<td>265,058</td>
</tr>
<tr>
<td>call_0_1_1</td>
<td>0x68b3465833fb72a70e...</td>
<td>0xc02aa9b223fe6d0a...</td>
<td>0 Ether</td>
<td>241,034</td>
</tr>
<tr>
<td>staticcall_0_1_1</td>
<td>0x68b3465833fb72a70e...</td>
<td>0x91c7ee0b8130cc11d4f...</td>
<td>0 Ether</td>
<td>229,914</td>
</tr>
<tr>
<td>staticcall_0_1_1</td>
<td>0x68b3465833fb72a70e...</td>
<td>0xa619e92c4c34c29a22...</td>
<td>0 Ether</td>
<td>222,959</td>
</tr>
<tr>
<td>staticcall_0_1_1</td>
<td>0x68b3465833fb72a70e...</td>
<td>0xc02aa9b223fe6d0a...</td>
<td>0 Ether</td>
<td>219,642</td>
</tr>
<tr>
<td>call_0_1_1</td>
<td>0x68b3465833fb72a70e...</td>
<td>0xa619e92c4c34c29a22...</td>
<td>0 Ether</td>
<td>217,441</td>
</tr>
<tr>
<td>call_0_1_1_1</td>
<td>0xa619e92c4c34c29a22...</td>
<td>0x91c7ee0b8130cc11d4f...</td>
<td>0 Ether</td>
<td>203,339</td>
</tr>
<tr>
<td>staticcall_0_1_1_1</td>
<td>0xa619e92c4c34c29a22...</td>
<td>0x91c7ee0b8130cc11d4f...</td>
<td>0 Ether</td>
<td>72,408</td>
</tr>
<tr>
<td>staticcall_0_1_1_1</td>
<td>0xa619e92c4c34c29a22...</td>
<td>0xc02aa9b223fe6d0a...</td>
<td>0 Ether</td>
<td>71,108</td>
</tr>
<tr>
<td>staticcall_0_1_1</td>
<td>0x68b3465833fb72a70e...</td>
<td>0x91c7ee0b8130cc11d4f...</td>
<td>0 Ether</td>
<td>54,187</td>
</tr>
</tbody>
</table>
A set of information that represents the current state is updated when a transaction takes place on the network. The below is a summary of those changes:

<table>
<thead>
<tr>
<th>Address</th>
<th>Before</th>
<th>After</th>
<th>State Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x4675c7e5baafbfbc7... Miner</td>
<td>(Fee Recipient: 0x467...263) 0.08247733010973233 Eth</td>
<td>0.08270763210973233 Eth</td>
<td>0.000230302</td>
</tr>
<tr>
<td>0x678a0cae712c6d86ea...</td>
<td>0.1852282790805995 Eth</td>
<td>0.031153006713285318 Eth</td>
<td>-0.154075272367314182</td>
</tr>
<tr>
<td>0x91c7ee0b8130cc11d4f...</td>
<td>BGLDN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xa619e92c4c34c29a22...</td>
<td>UNI-V2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0xc02aaa39b223fe8d0a...</td>
<td>WETH</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Storage (1)

<table>
<thead>
<tr>
<th>Storage Address: 0x105be7362eb0a99d5db59066b0e7cf4ad5b5087033e8f7419ae8d0aee56a2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before: Hex → 8x0000000000000000000000000000000a724e517520b2156</td>
</tr>
<tr>
<td>After: Hex → 8x0000000000000000000000000000000a941e97b85709f4e</td>
</tr>
</tbody>
</table>
Section 1

Basics of tracing
eth_call

eth.call({
    from: '0x00',
    to: 'WETH_TOKEN_ADDRESS',
    data: 'encode("balanceOf(address)", "0x00")'
})
debug_traceCall

dbg.traceCall({
  from: '0x00',
  to: 'WETH_TOKEN_ADDRESS',
  data: 'encode("balanceOf(address)", "0x00")'
}, 'latest')
Tracing historical transactions

- debug_traceTransaction
- debug_traceBlockByNumber/Hash
- debug_traceChain
  - Subscription API via Websocket
Section 2

Built-in tracers
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Desc</th>
</tr>
</thead>
<tbody>
<tr>
<td>pc</td>
<td>uint64</td>
<td>Program counter</td>
</tr>
<tr>
<td>op</td>
<td>byte</td>
<td>Opcode name</td>
</tr>
<tr>
<td>gas</td>
<td>uint64</td>
<td>Remaining gas</td>
</tr>
<tr>
<td>gasCost</td>
<td>uint64</td>
<td>Cost of opcode</td>
</tr>
<tr>
<td>memory</td>
<td>[]byte</td>
<td>Memory</td>
</tr>
<tr>
<td>memSize</td>
<td>int</td>
<td>Memory size</td>
</tr>
<tr>
<td>stack</td>
<td>[]uint256</td>
<td>Stack</td>
</tr>
<tr>
<td>returnData</td>
<td>[]byte</td>
<td>Last call's return data</td>
</tr>
<tr>
<td>storage</td>
<td>map[hash]hash</td>
<td>Accessed storage</td>
</tr>
<tr>
<td>depth</td>
<td>int</td>
<td>Call depth</td>
</tr>
<tr>
<td>refund</td>
<td>uint64</td>
<td>Refund counter</td>
</tr>
<tr>
<td>error</td>
<td>string</td>
<td>Error message</td>
</tr>
</tbody>
</table>
 Opcode tracer notes

- Memory, stack, return data, and storage can be disabled/enabled
- Watch out for memory
  - Tracing #14742200 fails with 64Gb memory
- If you control node, consider `debug_standardTraceBlockToFile`
Call tracer

debug.traceCall(
    {...},
    'latest',
    { tracer: 'callTracer' }
)

debug.traceTransaction(
    TX_HASH,
    { tracer: 'callTracer' }
)
Prestate tracer

Two modes:

- Prestate: outputs accounts which are needed to execute a tx
- Diff: outputs the modifications to state during a tx (as in etherscan)
State pruning
Ever seen this error?

Tracing needs the state prior to the given historical tx

```
Error: required historical state unavailable (reexec=128)
at web3.js:6365:9(45)
at send (web3.js:5099:62(34))
at <eval>:1:23(7)
```
How is the state prepared?

- Fetch state of parent block from database
- Execute txes in block until just before target tx

But what if state of previous block is NOT in db?
State persistence

Archive node

Full sync

Snap sync

128 blocks

128 blocks
Reexec

How state is prepared, take 2:

- Fetch state of parent block from database
- If not available:
  - Go back up to `reexec` blocks for first available state
  - Execute blocks sequentially to build up state
- Execute txes in block until just before target
Tricks

- See available states via `debug_getAccessibleStates`
- Set `reexec` param higher accordingly
- If interested in a range of blocks consider turning on `--gcmode=archive`
Custom tracer
Detecting a poisonous token

Poisonous Token In A Nutshell

```solidity
function transfer(address _to, uint256 _amount) public returns (bool) {
    if (block.coinbase == HARDHAT_TESTNET_COINBASE) {
        super.transfer(_to, _amount);
    }
    return true;
}
```
Solidity method invocations

- Collect list of method signatures (the first 4 bytes of the signature hash)
- Client-side map them to known method names
Roadmap

- debug_traceMulticall
- trace_* namespace
Thank you!

Sina Mahmoodi

go-ethereum
sina.mahmoodi@ethereum.org

@sina_mahmoodi