Empirically Analyizing Ethereum's Gas Mechanism

Renlord Yang[1,2], Toby Murray[1], Paul Rimba[2], Udaya Parampalli[1]

[1]University of Melbourne

[2]Data61, CSIRO

June 19, 2019

First DoS Exploitation

The Ethereum network is currently undergoing a DoS attack

Posted by Jeffrey Wilcke on September 22, 2016

URGENT ALL MINERS: The network is under attack. The attack is a computational DDoS, ie. miners and nodes need to spend a very long time processing some blocks. This is due to the EXTCODESIZE opcode, which has a fairly low gasprice but which requires nodes to read state information from disk; the attack transactions are calling this opcode roughly 50,000 times per block. The consequence of this is that the network is greatly slowing down, but there is NO consensus failure or memory overload. We have currently identified several routes for a more sustainable medium-term fix and have developers working on implementation.

DoS Response

Announcement of imminent hard fork for EIP150 gas cost changes

Posted by Martin Swende on October 13, 2016

During the last couple of weeks, the Ethereum network has been the target of a sustained attack. The attacker(s) have been very crafty in locating vulnerabilities in the client implementations as well as the protocol specification.

While the recent patches have led to an overall increased resiliency in the client implementations, the attacks have also demonstrated that a lower-level change to the EVM pricing model is needed.

For many users, the most visible consequence is probably that they are having difficulties getting transactions included in blocks, and full nodes are facing memory limitations in managing the bloated state.

The end of all attacks?

More responses over time...

Further Underpriced Opcodes:

- EIP150 Many Opcode Gas Price Changes
- EIP160 — EXP Cost Increase
- `BLOCKHASH` opcode gas price increase from 20 to 800
- and many more minute details...

More responses over time...

Further Underpriced Opcodes:

- EIP150 Many Opcode Gas Price Changes[1]
- EIP160 — EXP Cost Increase
- `BLOCKHASH` opcode gas price increase from 20 to 800[2]
- and many more minute details...

Summary: Finding the right gas price for the decentralized consensus computer is non-trivial.

[1] **aleth/aleth@e4c6f977e1a1e3c7efb77f7e287ec1b9909681fea**
[2] **aleth/aleth@35d92843b1babda6f0bd354c1bfbb4c0e2e025a1**

Trust Assumptions and Constraints

- Don't Trust, Verify — Independent Transaction Verification. Require keeping full state in node.
- Decentralization — As many nodes as possible should be able to participate and keep up with consensus.

Trust Assumptions and Constraints

- Don't Trust, Verify — Independent Transaction Verification. Require keeping full state in node.
- Decentralization — As many nodes as possible should be able to participate and keep up with consensus.

Challenge #1

Gas Pricing for Heterogeneous Hardware

Finding a gas price for each EVM opcode that works on different hardware machines is difficult.

Challenge #2

Performance Stability

Once a gas price is fixed for the opcode, making sure time spent variance is small is difficult.

Challenge #3

Unbounded State Growth

Increasing state also lead to degraded output done by backend database driver. More later.

Opcodes and Gas

Gas is fuel for computation in the Ethereum Virtual Machine.

Gas Cost is the amount of gas required to execute each EVM Opcode.

Gas Price is the current bidded price for each unit of Gas.

Fee is the sum of gas required to execute a transaction multiplied by the Gas Price.

Gas Limit is the maximum allowable gas in a block[3].

[3] Used to curb EVM computation

Gas Price Determination

Gas prices are determined by means of benchmarking using a "representative" machine with relatively common hardware specifications.

Our work shows that with different hardware specification, the Gas "economy" of running EVM operations can be vary significantly.

Participants with *good* hardware want to lift gas limit; while participants with *poorer* hardware want to lower gas limits.

Interplay of I/O with the EVM

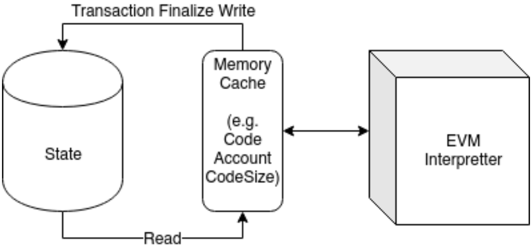


Figure: EVM Interpreter with State

1. Hardware components involved: Disk, Memory, CPU.
2. Latency plays a big role in EVM performance.

Hardware

Machine	A	B
Storage Type	PCIe NVMe SSD	SATA3 SSD
CPU	Intel® Xeon® Platinum 8180M@2.50GHz	Intel® Core™ i7-4770@3.40GHz
Threads (Core)	2 (112)	1 (4)
Cores (Socket)	28 (56)	4 (4)
Sockets	2	1
Memory	1.5TB DDR4 2300MHz	16GB DDR3 1600MHz
OS	Ubuntu 16.04 LTS	Ubuntu 16.04 LTS
Kernel	Linux 4.15.0-33	Linux 4.15.0-33

Metric

We collected our time measurements for the execution of each EVM opcode by instrumenting our `aleth` client with the `chrono` library.

The time measurements were then used to derive the time-to-gas ratio which measures execution time per unit gas.

The *baseline* for the gas prices are set at 1 gas per μs.

Selected Results (Top Means) on Different Machines

Opcode	μ	Median	IQR	σ	$\sigma \div \mu$
BLOCKHASH	34343	110	86971	34284.32	1.00
SLOAD	921	32	147	917.96	1.00
BALANCE	867	596	183	400.32	0.46

Figure: Machine A (Server)

Opcode	μ	Median	IQR	σ	$\sigma \div \mu$
BLOCKHASH	35156	117	81521	35097.84	1.00
SLOAD	16808	31	201	16805.73	1.00
BALANCE	12883	7070	3231	7808.86	0.61

Figure: Machine B (Desktop)

For full results, please check out our paper!

External Distribution Plots

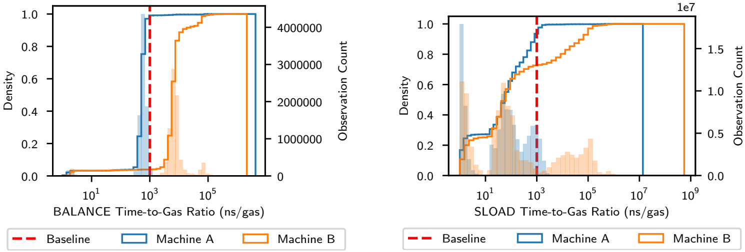


Figure: Distribution of `BALANCE` and `SLOAD` time-to-gas ratios

Computation Distribution Plots

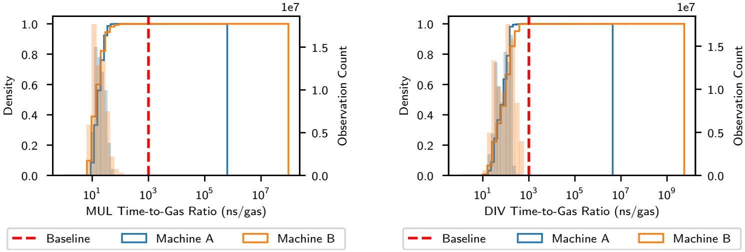


Figure: Distribution of `MUL` and `DIV` time-to-gas ratios

Computation vs. I/O

Takeaway Message:

- I/O operations yield high variance performance profile
- On top of variance, it is sensitive to differing hardware capabilities.
- EVM gas mechanism works relatively well with pure computation.

Computation vs. I/O

Takeaway Message:

- I/O operations yield high variance performance profile
- On top of variance, it is sensitive to differing hardware capabilities.
- EVM gas mechanism works relatively well with pure computation.

Computation vs. I/O

Takeaway Message:

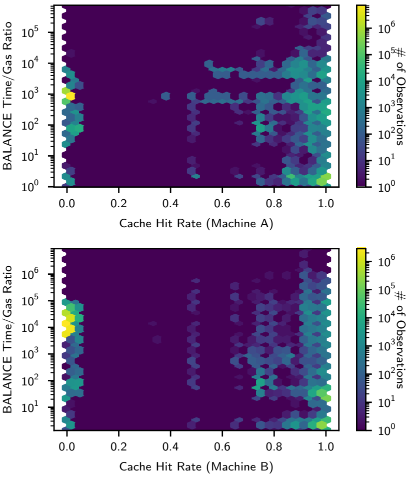
- I/O operations yield high variance performance profile
- On top of variance, it is sensitive to differing hardware capabilities.
- EVM gas mechanism works relatively well with pure computation.

Computation vs. I/O

Takeaway Message:

- I/O operations yield high variance performance profile
- On top of variance, it is sensitive to differing hardware capabilities.
- EVM gas mechanism works relatively well with pure computation.

Does existing caching help?



No, not really. At 100% hit rate, observed performance is very volatile.[4]

[4] This is just naive memory caching

Case Study: BALANCE underlying I/O operation time cost

Using low-level callgraph tracing:

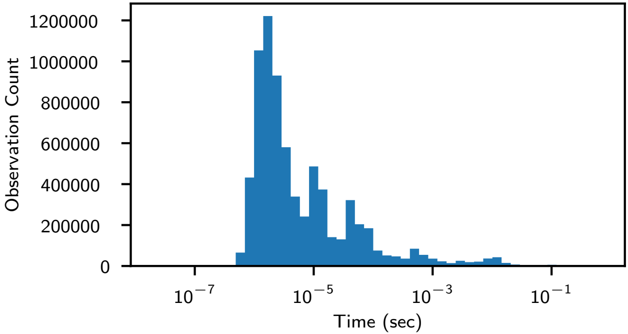


Figure: Time Distribution for Trie DB Lookup when executing `BALANCE` or `EXTCODESIZE`

Variance Implications

- Hardware choice is imperative to achieve lower transaction replay times. More so, if you're running a mining operation.
- Caching at its current state exaberates variance, makes it more difficult to target an optimal gas price for I/O opcodes.
- There's a good chance that competitive advantage in gas economy for well-resourced peer hurts node diversity and decentralization.

Renlord Yang, Toby Murray, Paul Rimba, Udaya Parampalli

Variance Implications

- Hardware choice is imperative to achieve lower transaction replay times. More so, if you're running a mining operation.
- Caching at its current state exaberates variance, makes it more difficult to target an optimal gas price for I/O opcodes.
- There's a good chance that competitive advantage in gas economy for well-resourced peer hurts node diversity and decentralization.

Renlord Yang, Toby Murray, Paul Rimba, Udaya Parampalli

Variance Implications

- Hardware choice is imperative to achieve lower transaction replay times. More so, if you're running a mining operation.
- Caching at its current state exaberates variance, makes it more difficult to target an optimal gas price for I/O opcodes.
- There's a good chance that competitive advantage in gas economy for well-resourced peer hurts node diversity and decentralization.

Variance Implications

- Hardware choice is imperative to achieve lower transaction replay times. More so, if you're running a mining operation.
- Caching at its current state exaberates variance, makes it more difficult to target an optimal gas price for I/O opcodes.
- There's a good chance that competitive advantage in gas economy for well-resourced peer hurts node diversity and decentralization.

Denial-of-Service

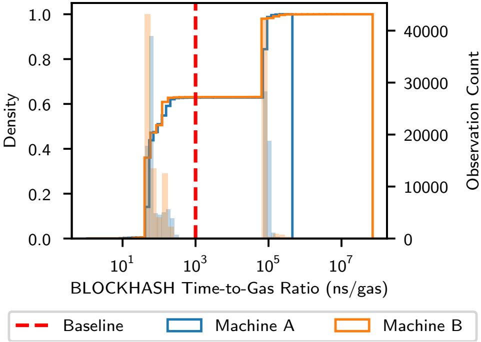


Figure: `BLOCKHASH` distribution plot

Consistent re-producible poor performance can lead to DoS exploitation by an adversary.

This was patched in the Constaniople Hard-Fork.

Possible Solutions

There exist a couple of approaches to solve this:

- **I/O Optimization** Database-backend design that better fits the I/O job profile [1].
- **State Channels** Reduce on-chain transactions [2, 3]
- **Sharding** Reduce amount of state stored on node by state-sharding [4, 5, 6, 7]
- **ZK-Proofs** Avoid transaction replay all together and do pure cryptographic verification [8, 9]

All come with their own set of tradeoffs.

TL;DR Summary

- We don't quite know how to price EVM opcodes for a heterogenuous ecosystem.
- High variance for opcode execution times poses risk.
- I/O-based opcodes have very high variance.

TL;DR Summary

- We don't quite know how to price EVM opcodes for a heterogenuous ecosystem.
- High variance for opcode execution times poses risk.
- I/O-based opcodes have very high variance.

TL;DR Summary

- We don't quite know how to price EVM opcodes for a heterogenuous ecosystem.
- High variance for opcode execution times poses risk.
- I/O-based opcodes have very high variance.

TL;DR Summary

- We don’t quite know how to price EVM opcodes for a heterogenuous ecosystem.
- High variance for opcode execution times poses risk.
- I/O-based opcodes have very high variance.

Post-Talk Information

https://github.com/renlord/bookish-octo-barnacle

Renlord Yang, Toby Murray, Paul Rimba, Udaya Parampalli

References I

P. Raju, R. Kadekodi, V. Chidambaram, and I. Abraham, “PebblesDB,” in *Proceedings of the 26th Symposium on Operating Systems Principles - SOSP '17*, USENIX Association. ACM Press, 2017. [Online]. Available: https://doi.org/10.1145/3132747.3132765

S. Dziembowski, S. Faust, and K. Hostáková, “General state channel networks,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 949–966.

A. Miller, I. Bentov, R. Kumaresan, and P. McCorry, “Sprites: Payment channels that go faster than lightning,” *CoRR*, vol. abs/1702.05812, 2017. [Online]. Available: http://arxiv.org/abs/1702.05812

References II

M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, and G. Danezis, "Chainspace: A sharded smart contracts platform," in *Proceedings 2018 Network and Distributed System Security Symposium*. Internet Society, 2018. [Online]. Available: https://doi.org/10.14722/ndss.2018.23241

E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *2018 IEEE Symposium on Security and Privacy (SP)*, IEEE. IEEE, May 2018, pp. 583–598. [Online]. Available: https://doi.org/10.1109/sp.2018.000-5

References III

L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*, ACM. ACM Press, 2016, pp. 17–30. [Online]. Available: https://doi.org/10.1145/2976749.2978389

D. Sel, K. Zhang, and H.-A. Jacobsen, "Towards solving the data availability problem for sharded ethereum," in *Proceedings of the 2nd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers - SERIAL'18*, ACM. ACM Press, 2018, pp. 25–30. [Online]. Available: https://doi.org/10.1145/3284764.3284769

References IV

E. Ben-Sasson, A. Chiesa, D. Genkin, E. Tromer, and M. Virza, "Snarks for C: verifying program executions succinctly and in zero knowledge," in *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, 2013, pp. 90–108. [Online]. Available: https://doi.org/10.1007/978-3-642-40084-1_6

S. Bowe, A. Chiesa, M. Green, I. Miers, P. Mishra, and H. Wu, "Zexe: Enabling decentralized private computation," *IACR Cryptology ePrint Archive*, vol. 2018, p. 962, 2018. [Online]. Available: https://eprint.iacr.org/2018/962