Continuation/Research Progression Projects Form (7)

Required for projects that are a continuation/progression in the same field of study as a previous project.

This form must be accompanied by the previous year's abstract and Research Plan/Project Summary.

Student's Name(s) Michael Lin

To be completed by Student Researcher: List all components of the current project that make it new and different from previous research. The information must be on the form; use an additional form for previous year and earlier projects.

Components	Current Research Project	Previous Research Project: Year: 2018
1. Title	Design of a Blockchain Protocol Scalability Comparison Apparatus Using Virtual Machines	Design of a Blockchain Protocol Scalability Comparison Apparatus Using Virtual Machines
2. Change in goal/ purpose/objective	Refine the implementation of the apparatus to incorporate parallelized RPC requests and selective throttling, and gather the first "profile" of dependent variables observed.	Create the first serial implementation of the apparatus on Bitcoin Core. Deploy the apparatus to gather initial results.
3. Changes in methodology	Modifications on existing code. Virtual machine specification is weakened to reduce operating costs without degradation of experiment data.	Code base from scratch, model first created during 2018. Uses GCE virtual machines. All scripts are licensed under GPLv3 for use by anyone.
4. Variable studied	Same, subject to additions.	PoolSize - Multi-parameter regression MinFee - Minimum estimate of atomic fee MedFee - Median estimate of atomic fee All variables are time-derivatives.
5. Additional changes	N/A	N/A

Attached are:

☑ Abstract and Research Plan/Project Summary, Year 2018

I hereby certify that the above information is correct and that the current year Abstract & Certification and project display board properly reflect work done only in the current year.

Michael Lin	Michael Lin Digitally signed by Michael Lin Date: 2019.12.15 17:27:03 -05'00'	12/15/19
Student's Printed Name(s)	Signature	Date of Signature (mm/dd/yy)

Experimental Comparison of Blockchain Scaling Protocols Using Virtual Machines

Research Plan

Michael Lin, Yorktown High School

Category: Systems Software (SOFT)

a. This research will entail the process of deriving performance metrics (a.k.a. "scalability") from blockchains, a combination of peer-to-peer networks and append-only cryptographic data structures for financial or general-purpose state transitions (changes in the state or current progress of a system, such as currency transactions). Blockchains (linearly hashed blocks containing state transitions) are consistently updated and replicated across the network in such a way that liveness is maintained. In this way, blockchains are a special type of state machine. Using blockchains, any computational task can be accomplished without a central managing authority to regulate, such as a bank or state institution. Due to strict consensus rules, any computation can be done with determinism, lack of trust, and a negligible probability of failure. A blockchain is a promising environment to host a multitude of applications in various private sectors, government agencies, and other areas where permissioned and reliable computation is required for servicing customers and denizens.
b. Although blockchains are a desirable place to keep Internet applications, there lies the problem that [in general] blockchains cannot handle high loads to the extent of current servers such as those used in banks or online payment services. Generalized as a trilemma (a choice between three desirable properties, in which only two may be chosen and the third sacrificed), blockchains should have security, scalability, and a decentralized nature. One goal of research in this field is to solve this trilemma in a pragmatic way. The goal of this research project will be to evaluate existing solutions from the pool of ideas presented by other researchers experimentally and to establish a comparison between various solutions in terms of security and scalability. Many of these solutions fall within one of two categories: On-chain scaling protocols which are directly written into client code and affect the structure of the blockchain, and off-chain scaling protocols which relieve pressure off the blockchain through an external protocol. The alternate hypothesis formed for this project is that off-chain scaling protocols will be more scalable but less secure to attack and fault than on-chain scaling protocols.
c. Experiment Outline (Subject to change through *Project Summary Addendum*):
    A. Procedure
        1. Technical specifications will be determined to sustain the most resource intensive private blockchain network of at least 1000 full nodes.
        2. Cloud virtual machines (VMs) will be rented out for each protocol configuration from Google Cloud Platform. These virtual machines will be identical in specifications.

    3. Each VM will be set up to run its own private testing network (“testnet”) of at least 1000 full nodes (exact number TBD) using premade VM images to deploy an instance group on Google Cloud Platform
4. For each VM, peer gossip will be started by introducing peer IPs to other peer tables to create a decentralized network graph.
5. For each VM, the genesis block (first block of the blockchain) will be deployed and auxiliary miners (who will generate new valid blocks via PoW) will be deployed if necessary to keep the network running.
6. Transactions shall be sent out at a discrete interval randomly from various nodes in each testnet, with each timestamp recorded by a centralized oracle. Each peer prioritizes this oracle without perceptible latency, but regular peer gossip becomes latent on the order of milliseconds to simulate a real network. This latency is deterministic, derived using recent block data and transaction data to pseudorandomly change the latency or produce a failure using a hash function.
7. After running all VMs for 5 days, derive the following variables (equations are in Word):

Table 1. Elementary Symbols and Functions

CT_s	Median block time in seconds
CT_b	Minimum number of block confirmations before safety in a selfish mining attack
SafeTime	Minimum time before safety, in seconds — $SafeTime \equiv CT_s \times CT_b$
Tx	A single transaction (determined by context)
Blk	A single block (determined by context)
Blk_α	Genesis block of the applicable blockchain
B	The blockchain, a vector of blocks, which are vectors of transactions, all sorted by time
r	The current instantaneous count of transactions per second
B_r	The blockchain segment of TPS *r*, such that $B_r \in B$
Blk_r	First block of the TPS segment B_r.
R()	TPS interval of *Blk* or *Tx*.
min()	Returns smallest valued argument of input arguments
max()	Returns largest valued argument of input arguments
mod(*x*,*y*)	Calculate remainder of the Euclidean division of *x* by *y*
$\Omega()$	Best-case scenario (asymptotic)
O()	Worst-case scenario (asymptotic)
BT()	Number of blocks before inclusion of *Tx*, returns a signed integer (int)
BH()	Height of input *Tx* or *Blk*, returns an unsigned integer (uint) or float.
Fee()	Returns the fee of *Tx* in atomic cryptocurrency units, such as satoshis or wei.
T()	Returns the timestamp of *Tx* or *Blk* as seconds past January 1st, 1970, 00:00:00 UTC as a uint.
|*x*|	Returns the cardinality of set/tuple/vector *x*. Not to be confused with absolute value.
⌈*x*⌉	Ceiling Function. If *x* is an integer, return *x*. If *x* is a float, truncate the decimal portion and add 1.
hash(*x*,*y*)	Calculates hash of x using a hash function, identified by *y*.
$\Sigma(x)$	Returns $T(Tx \text{ or } Blk)\ T(Blk_\alpha)$, or the time from genesis.
Π(x)	Returns $T(Tx \text{ or } Blk)\ -\ T(Blk_{R(Tx\ or\ Blk)})$, or the time from the first block of the TPS interval.
Rwd()	If block, return total block reward without transaction fees. If blockchain, return original block reward.

Table 2. Intermediate and Dependent Variable Definitions

Lowest safe minimum fee during TPS interval in atomic units	$AtomMinFee_r \coloneqq min\big(\Omega\big(\mathrm{Fee}(Tx)\big)\ \forall\, Tx \in B_r\big)$
The maximum block wait time for a $MinFee$ transaction	$MaxBlock \coloneqq max\left(\mathcal{O}\big(BT(Tx \mid MinFee)\big)\right)\ \forall\, Tx \in B$
Median transaction fee	$AtomMedFee_r \coloneqq \mathrm{med}(Fee(Tx))\ \forall\, Tx \in B_r$
Maximum time in seconds for a *MinFee*	$MaxTime \equiv MaxBlock \times CT_s$
Average of transaction timestamp vs block timestamp (Unused)	$ConfTime \coloneqq \frac{T(\lceil BH(Tx) \rceil) - T(Tx)}{\lvert B \rvert}\ \forall\, Tx \in B$
Mempool size modified sawtooth wave regression	$PoolSize \equiv poolSize(r) \coloneqq mod(r \times t, \mathrm{CT_s}) + mt - b$
Block timestamp (s)	$sum(T(Blk, node1), T(Blk, node2), \ldots, T(Blk, node1000))/1000$
Atomic units per Reward Unit ()	$RU \coloneqq (Rwd(\mathrm{B}) \times CT_b)$
Cross-chain comparable *MinFee*	$RUMinFee_r \equiv AtomMinFee_r \div RU$
Cross-chain comparable *MedFee*	$RUMedFee_r \equiv AtomMedFee_r \div RU$

10. Make charts for each dependent variable, including data from every VM.

B. Risk and Safety:
   - There will be no risks in this experiment. No personal data will be collected or stored, and all computations will be performed in a virtualized and sandboxed environment.

C. Data Analysis (Subject to change through the *Project Summary Addendum*):
   - A copy of the blockchain will be stored locally in each node, acquiring it will be trivial.
   - Transaction timestamps will be collected by a centralized oracle, which will have no latency and will accurately represent the time a transaction is initially broadcast.
   - From this data, the dependent variables can be derived using their respective equations.

D. References

Back, A. (2002, August 1). Hashcash — A Denial of Service Counter-Measure. Retrieved from http://www.hashcash.org/papers/hashcash.pdf

Bitcoin Avg. Transaction Fee chart. (n.d.). Retrieved August 15, 2018, from https://bitinfocharts.com/comparison/bitcoin-transactionfees.html#1y

Buterin, V. (2015, September 14). On Slow and Fast Block Times. Retrieved August 11, 2018, from https://blog.ethereum.org/2015/09/14/on-slow-and-fast-block-times/

Buterin, V., & Griffith, V. (2017). Casper the Friendly Finality Gadget. *Computing Research Repository (CoRR)*, arXiv:1710.09437v2.

Buterin & Griffith detail a new type of PoS consensus mechanism, one that discourages forks and makes finality possible. The new system, called Casper, can only be implemented on blockchains that are Turing-complete.

Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., . . . Wattenhofer, R. (n.d.). On Scaling Decentralized Blockchains. Retrieved August 15, 2018, from https://fc16.ifca.ai/bitcoin/papers/CDE 16.pdf

Hall, T. A., & Keller, S. S. (2014, March 18). *The FIPS 186-4 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)* (United States, NIST, Information Technology Laboratory). Retrieved from https://csrc.nist.gov/CSRC/media/Projects/Cryptographic-Algorithm-Validation-Program/documents/dss2/ecdsa2vs.pdf

Kemp, S. (2018, January 30). Digital in 2018: World's internet users pass the 4 billion mark. Retrieved September 23, 2018, from https://wearesocial.com/blog/2018/01/global-digital-report-2018

King, Sunny, and Nadal, Scott. “PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake.” Peercoin, 19 Aug. 2012, https://peercoin.net/assets/paper/peercoin-paper.pdf.

In this article, King & Nadal formally describe the first implementation of a Proof-of-Stake consensus mechanism into a public deployment blockchain.

Koblitz, N. (2001). *Introduction to elliptic curves and modular form*. New York: Springer.

Kwon, J. (2018, June 22). Tendermint Documentation. Retrieved August 11, 2018, from https://media.readthedocs.org/pdf/tendermint/master/tendermint.pdf

Merkle, R. C. (1979). *U.S. Patent No. 4309569(A)*. Washington, DC: U.S. Patent and Trademark Office.

Nakamoto, S. (2008, October 31). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved March, from https://bitcoin.org/bitcoin.pdf/

This article, written by Nakamoto, describes the novel idea of using blockchains for the first time in history. It has not been modified since its original release and was published before Bitcoin’s genesis block. While not a formal specification, it describes the core components of Bitcoin’s structure.

Percival, C., & Josefsson, S. (2016, August). The scrypt Password-Based Key Derivation Function. Retrieved from https://tools.ietf.org/html/rfc7914

This is IETF RFC 7914, a formal specification of the scrypt PBKDF.

Ray, J. (2018, August 22). On sharding blockchains. Retrieved September 23, 2018, from https://github.com/ethereum/wiki/wiki/Sharding-FAQs#this-sounds-like-theres-some-kind-of-scalability-trilemma-at-play-what-is-this-trilemma-and-can-we-break-through-it

Rescorla, E. (2018, August). RFC 8446 - The Transport Layer Security (TLS) Protocol Version 1.3. Retrieved September 23, 2018, from https://tools.ietf.org/html/rfc8446

Rivest, R. L., et al. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.” Jan. 1978, doi:10.21236/ada606588.

Srinivasan, R., (2011, October 01). Mathematics Towards Elliptic Curve Cryptography-by Dr. R.Srinivasan. Retrieved from https://www.slideshare.net/municsaa/mathematics-towards-elliptic-curve-cryptographyby-dr-rsrinivasan

Stark, J. (2018, February 12). Making Sense of Ethereum's Layer 2 Scaling Solutions: State Channels, Plasma, and Truebit. Retrieved August 15, 2018, from https://medium.com/l4-media/making-sense-of-ethereums-layer-2-scaling-solutions-state-channels-plasma-and-truebit-22cb40dcc2f4

Szabo, N. (1997). Formalizing and Securing Relationships on Public Networks. *First Monday*,*2*(9). doi:10.5210/fm.v2i9.548

Szilágyi, Péter. *Clique PoA Protocol & Rinkeby PoA Testnet*. 4 Apr. 2017, https://github.com/ethereum/EIPs/issues/225. Accessed 11 Aug. 2018.

United States, NIST, Information Technology Lab. (2015, August). *Secure Hash Standard (FIPS 180-4)*. March 23, 2018, http://dx.doi.org/10.6028/NIST.FIPS.180-4

Vicco, A. (2016, September 8). Code is Law and the Quest for Justice. Retrieved from https://ethereumclassic.github.io/blog/2016-09-09-code-is-law/

Wang, L., & Pustogarov, I. (2017). Towards Better Understanding of Bitcoin Unreachable Peers. *CoRR, Abs/1709.06837*.

Wood, G., & Buterin, V. (2018, June 5). Ethereum: A Secure Decentralized Transaction Ledger. Retrieved from https://ethereum.github.io/yellowpaper/paper.pdf

Ethereum, a decentralized application platform built from a blockchain and virtual machine substrate, is technically defined in this paper. This paper is purely informational and is a

formal specification of the Ethereum protocol, as of the Metropolis vByzantium hard fork consensus rules at mainnet block 4370000. This paper is identifiable by its revision ID, `0xe94ebda`

e. Project Summary Addendum

- Many blockchains use the PoW consensus mechanism, which requires the use of computational power to produce proofs that a certain amount of effort was put into making an immutable collection of transactions (blocks). In turn, these "miners" get rewarded with the native currency of the blockchain. PoW, while preventing a blockchain from being spammed by malicious blocks from an attacker, is not efficient and costs energy. Some blockchains which will be tested use Proof of Stake (PoS), which functions by integrating the stake (or money set aside) for the purposes of creating blocks, and rewards "forgers" differently. Energy use is minimal, and blocks are minted instead.
- Blockchains that have a regression test ("regtest") mode allow instantaneous minting of blocks without any proof (mainly in Bitcoin Core and forked projects such as Litecoin). This is possible in the experiment, as there is no risk of an attack on the private network. In production, this should never be used as it allows for the network to be spammed. Regtest is preferred over PoW or PoS, due to efficiency gains.
- In public production chains, the difficulty is determined by previous block times, to tend toward a certain block time. This is almost never perfect, and block times in a regtest network will reflect this in a method similar to the one that will be used for adding transaction latency.
- CT_s will default to difficulty retarget rule's time in a client's official release, if present. Many clients have such a rule.
- If the blockchain supports sharding into smaller universes ("subchains"), an equal quantity of nodes will be dedicated to each universe, and less will be dedicated toward the unifying blockchain. Miners will perform PoW for both collations (blocks in a subchain) and blocks.
- If the blockchain supports Distributed Proof of Stake (DPoS), a single node will be chosen as the forger, and all stake will go towards that node. DPoS, unlike PoS, collects the stake of endorsers to elect a single forger to produce blocks, wherein PoS, the forger uses their own stake.
- Some blockchains are hybrids between these schemes. Modifications for these chains will be determined on a case-by-case basis.
- CT_b will be 10 blocks for Ethereum, according to a realistic adversary model (Buterin, 2015).

- Ethereum does not possess a regtest mode. However, it does possess both a "devtest" mode, which has very low difficulty, and an alternate Proof of Authority consensus mechanism used purely for testing in a non-deployment environment (Szilágyi, 2017). PoA is very similar to regtest but allows for multiple "sealers" to agree on a block. In this experiment, only one sealer will be used as no attacker is present. This saves on computational costs.
- The variables *ConfTime*, *MaxBlock*, and similar have been dropped. They can be predicted mathematically, and will not be tested.
- The oracle and miner nodes are replaced with a random selection of nodes. This eliminates bias by evenly spreading the functions of transaction timestamp collection and block production.
- Instead of adding latency, a bandwidth limiter is added instead to simulate its effects. This is achieved with the wondershaper utility.
- All scripts are in Python 3.X.X syntax. This code will be published separately.
- All software used are licensed under the GNU Public License, copyright holders should find attribution in the code itself.
- Raw data is stored in generic Comma Separated Values.
- No Turing-complete contracts are used in this experiment. All transactions will be controlled by private keys.

2018-2019

OFFICIAL ABSTRACT and CERTIFICATION

Experimental Comparison of Blockchain Scaling Protocols Using Virtual Machines

Michael Lin

The Internet has enabled a vast amount of applications that require fast and near-instant communication across increasingly powerful computers. With new developments in both computation and cryptography, an experimental yet proven technology has emerged: the blockchain. The blockchain enables timestamped, replicated histories of computation stored across the Internet with succinct proofs of truth. Trust is no longer required in a single party as computation is verified many times over by independent, uncoordinated parties. Trustless finances and verifiable computation have resulted from this new substrate but has not proven to be scalable enough to replace existing centralized finances and cloud services. The fault tolerance that is provided by a decentralized perpetual truth machine is an incentive to move computing, storage, and market applications to it, and provide an ecosystem for various applications. These are also known as smart contracts, enabling any automated action to be completed deterministically without dispute. In order to achieve this ideal, the scalability problem must be met without sacrificing the current benefits of using centralized systems. Because of the problems faced with this scalability trilemma, researchers have taken to creating protocols that attempt to maintain security while increasing scalability. Finding the best protocols and their optimal implementations, as well as their future-proofness is necessary for strengthening application infrastructure. While plenty of theoretical research exists, experimental verification of these theories is sparse, failing to compare existing protocols in a heuristic fashion. Due to their decentralized nature, protocols can be compared using virtual machines in an experimental setting.

Category
Pick one only—
mark an "X" in
box at right

- Animal Sciences ☐
- Behavioral and Social Science ☐
- Biochemistry ☐
- Cellular & Molecular Biology ☐
- Chemistry ☐
- Computational Bio/ Bioinformatics ☐
- Computer Science ✔
- Earth Science ☐
- Engineering ☐
- Environmental Science ☐
- Mathematical Sciences ☐
- Medicine and Health ☐
- Microbiology ☐
- Neuroscience ☐
- Physics and Astronomy ☐
- Plant Sciences ☐

1. As a part of this research project, the student directly handled, manipulated, or interacted with (check ALL that apply):
   ☐ human subjects ☐ potentially hazardous biological agents
   ☐ vertebrate animals ☐ microorganisms ☐ rDNA ☐ tissue
2. This abstract describes only procedures performed by me/us, reflects my/our own independent research, and represents one year's work only — ☒ Yes ☐ No
3. I/we worked or used equipment in a regulated research institution or industrial setting: — ☐ Yes ☒ No
4. This project is a continuation of previous research. — ☐ Yes ☒ No
5. My display board includes non-published photographs/visual depictions of humans (other than myself): — ☐ Yes ☒ No
6. I/we hereby certify that the abstract and responses to the above statements are correct and properly reflect my/our own work. — ☒ Yes ☐ No

This stamp or embossed seal attests that this project is in compliance with all federal and state laws and regulations and that all appropriate reviews and approvals have been obtained including the final clearance by the Scientific Review Committee.