Blockchain Models and Benchmarks

—

Blockchain from a Performance Engineering Perspective

Aad van Moorsel, Amjad Aldweesh
Maher Alharby, Paul Ezhilchelvan

Newcastle University, UK

EPSRC National Cyber Security Centre a part of GCHQ
Academic Centre of Excellence in Cyber Security Research

Newcastle University

Outline

- Blockchain explained
- Bitcoin
- Ethereum
- Three Performance Layers in Blockchain
  - Processing layer
  - Connector layer
  - Incentives layer

  Benchmarks and models in these three layers
- Conclusion and Outlook

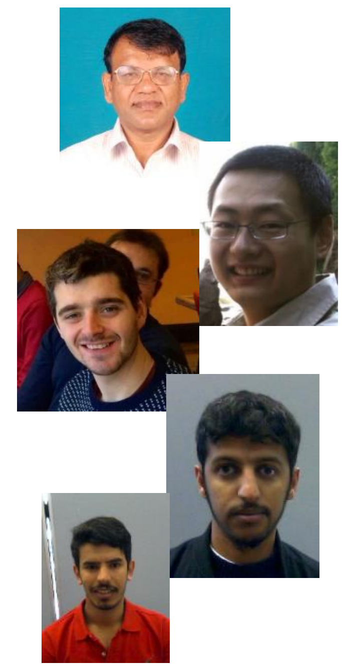
FinTech Research @ Newcastle

http://www.ncl.ac.uk/computing/research/groups/srs/#staff

Interfacing effectively with mobile users

Securing tomorrow's payment systems

Blockchain as disruptive technology



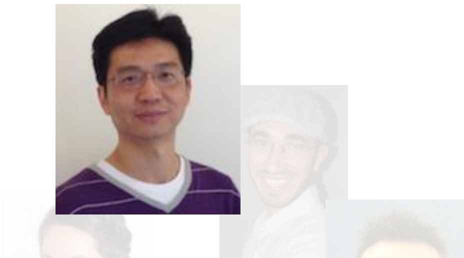
EPSRC National Cyber Security Centre a part of GCHQ

Academic Centre of Excellence in Cyber Security Research

Newcastle University

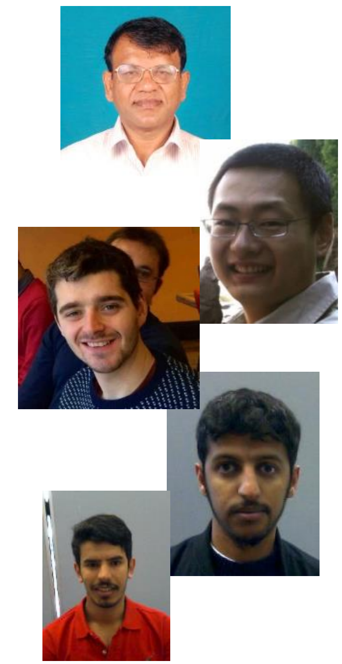
Blockchain Research @ Newcastle

Interfacing effectively with mobile users



Securing tomorrow's payment systems

Blockchain as disruptive technology



Number of uses of blockchain smart contracts:

- Changyu Dong, et al., "Betrayal, trust and rationality: Smart counter-collusion contracts for **verifiable cloud computing**", CCS 2017
- Paul Ezhilchelvan, et al., "Non-blocking **two phase commit** using blockchain", MobiSys CryBlock, 2018 (submitted)
- Patrick McCorry et al., "A smart contract for boardroom **voting** with maximum voter privacy", Financial Crypto, 2017

EPSRC | National Cyber Security Centre a part of GCHQ

Academic Centre of Excellence in Cyber Security Research

Newcastle University

Blockchain Research @ Newcastle

Interfacing effectively with mobile users

Securing tomorrow's payment systems

Blockchain as disruptive technology

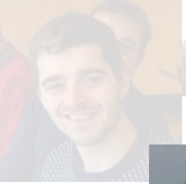
Amjad Aldweesh:

- "A survey about blockchain software architectures", *UKPEW 2017*

Maher Alharby:

- "Blockchain-based Smart Contracts: A Systematic Mapping Study", *Computer Science and Information Technology, UAE,* 2017
- "The impact of profit uncertainty on miner decisions in blockchain systems", *UKPEW*, extended in *Electronic Notes in Theoretical Computer Science*, 2017

£360K Project with Atom Bank: automated services for secured lending with blockchain as integration platform



Security Centre

Academic Centre of Excellence
in Cyber Security Research

Newcastle University

Blockchain Models & Benchmarks

—

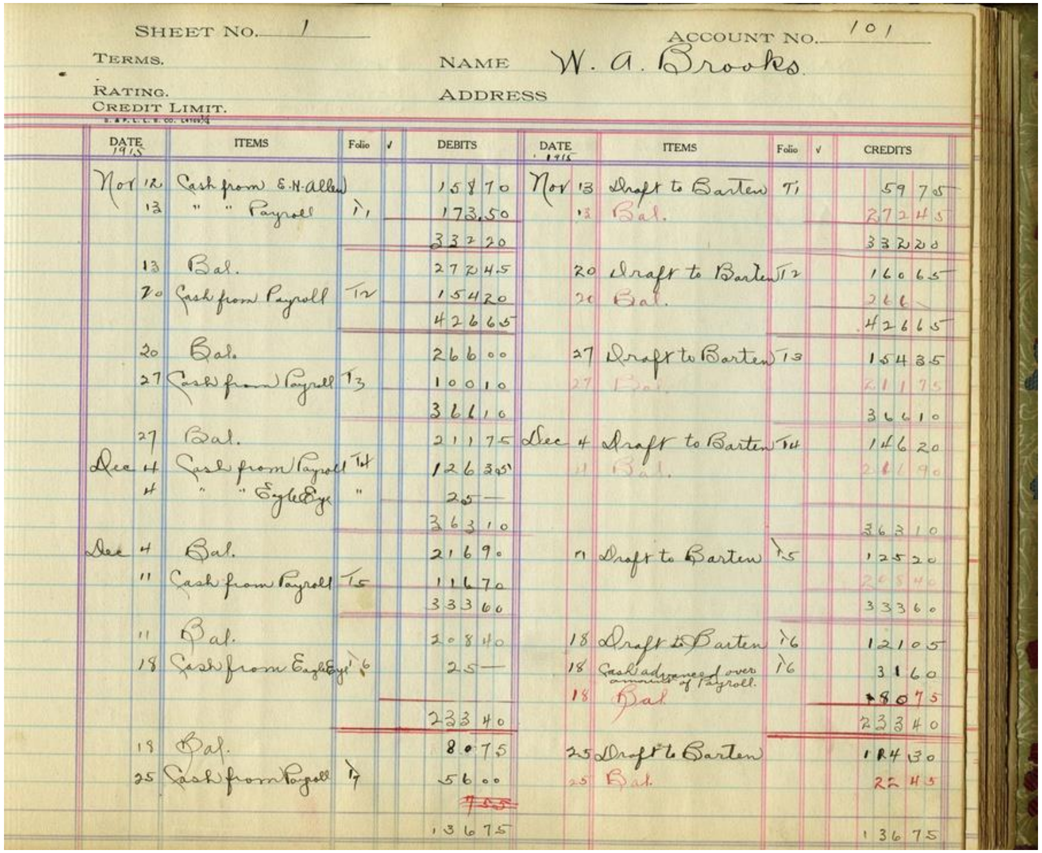
Blockchain explained

EPSRC National Cyber Security Centre
a part of GCHQ
Academic Centre of Excellence
in Cyber Security Research

Newcastle University

Blockchain explained (1)

blockchain is a **ledger**



Picture from PBS, copyright unkown

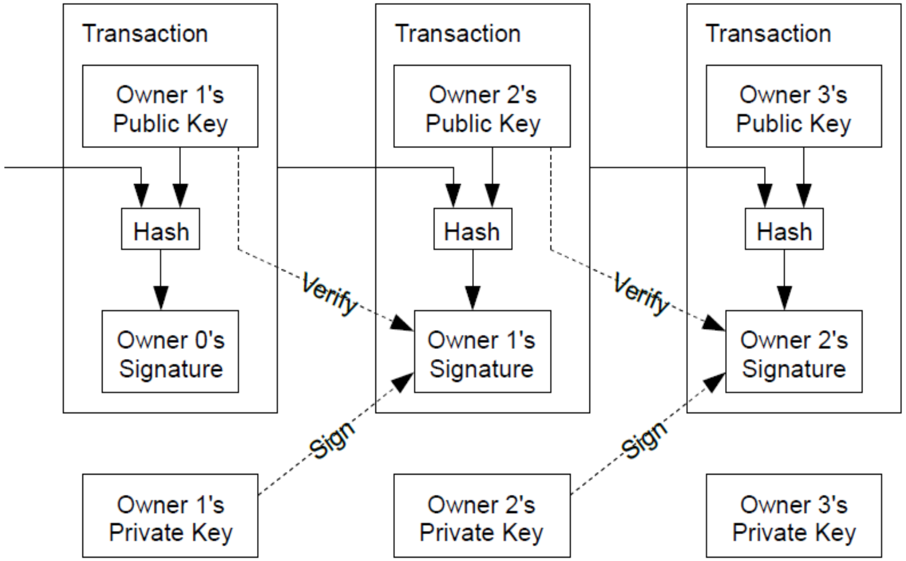
EPSRC National Cyber Security Centre
a part of GCHQ
Academic Centre of Excellence in Cyber Security Research

Newcastle University

Blockchain explained (2)

blockchain stores **only digital elements**

- Sign with private key: clear ownership
- Verify with public key: find out overspend
- Unmutable, uncopyable
- Coin as unit/currency



From Nakamoto 2008

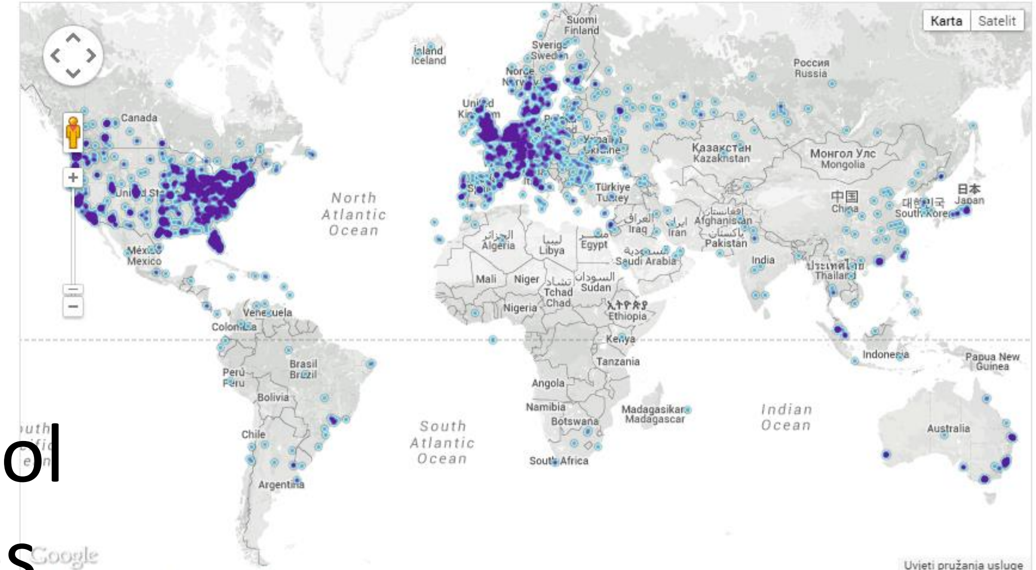
EPSRC National Cyber Security Centre a part of GCHQ
Academic Centre of Excellence in Cyber Security Research

Newcastle University

Blockchain explained (3)

blockchain ledger is **distributed**

- Each miner keeps a copy of the ledger
- Peer-to-peer protocol to distribute updates



Map shows concentration of reachable Bitcoin nodes found in countries around the world.

From Coindesk.com

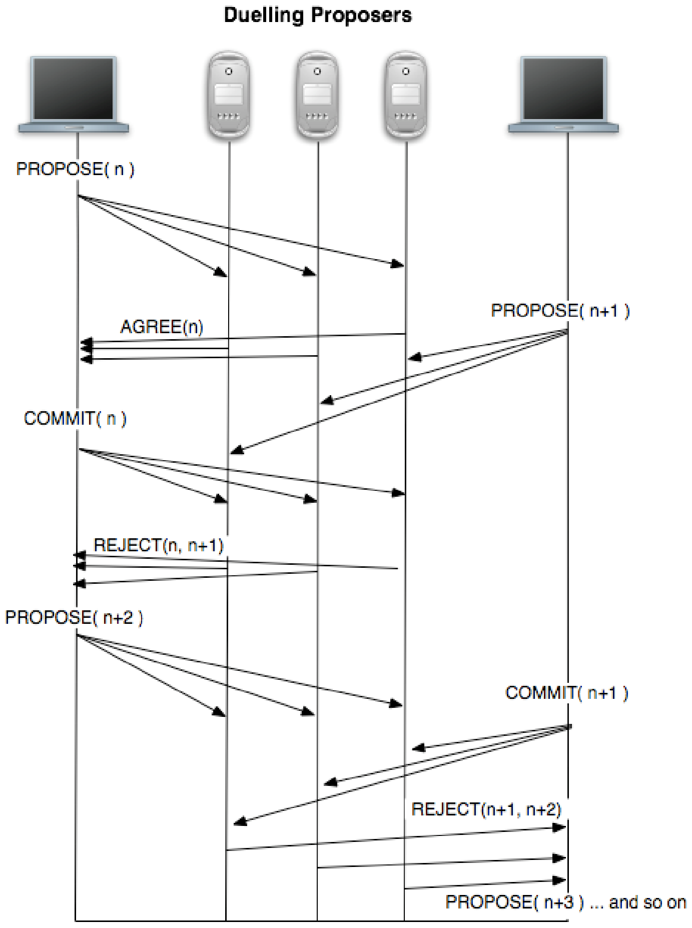
EPSRC
National Cyber Security Centre
a part of GCHQ
Academic Centre of Excellence in Cyber Security Research

Newcastle University

Blockchain explained (4)

Because blockchain is **distributed,** miners need to reach **consensus** about all updates and verify them:

- a consensus algorithm needed



From inerciatech.com

EPSRC National Cyber Security Centre a part of GCHQ

Academic Centre of Excellence in Cyber Security Research

Newcastle University

Blockchain explained (5)

Everyone must be able join the blockchain, how can we trust them?

- **Proof of Work**
  - Need to invest (CPU cycles), for a reward, so you gain a stake in the blockchain
  - Introduces a competition, that no single party can always win

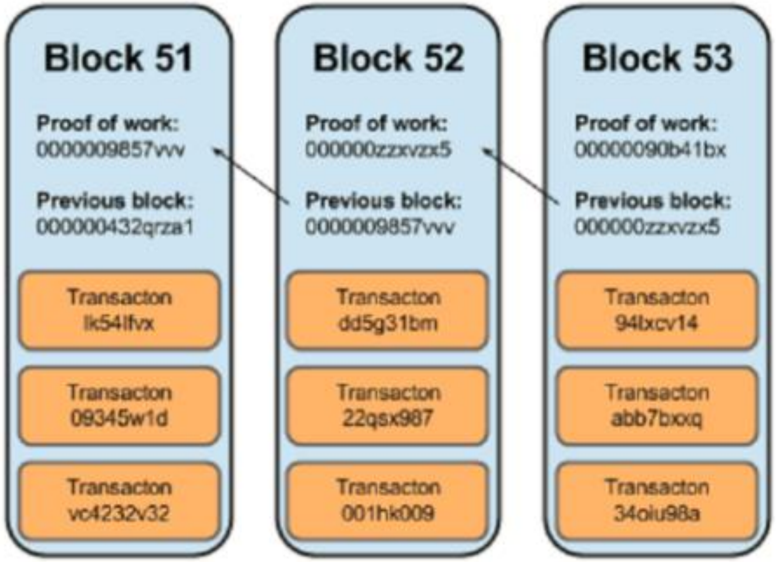
5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

From wikimedia.org

Blockchain explained (6)

Since there will be many transactions, we cannot carry out Proof of Work for every transaction:

- Group transactions in a **block**
- Miner that wins PoW send block around to update the ledger



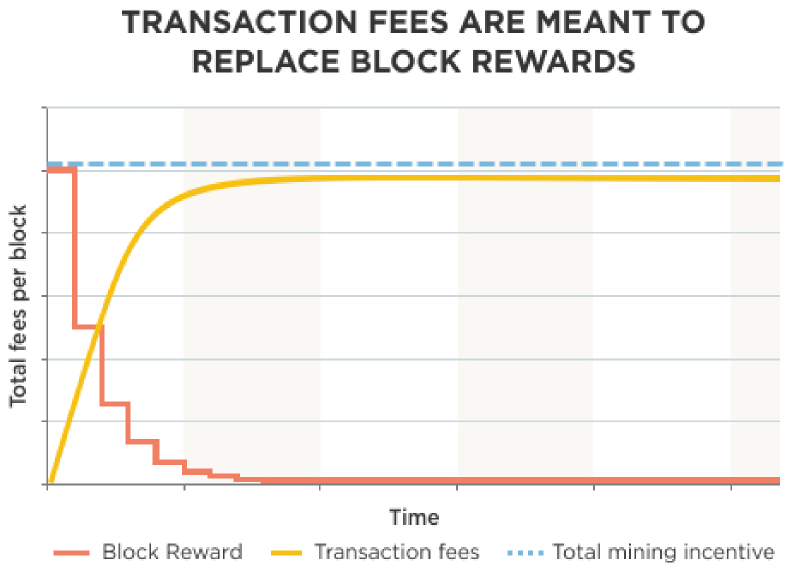
From www.processexcellencenetwork.com

Blockchain explained (7)

Miner gets reward:

- For PoW: **Block reward**
- **Transaction rewards**

Reward is proportional to CPU power invested



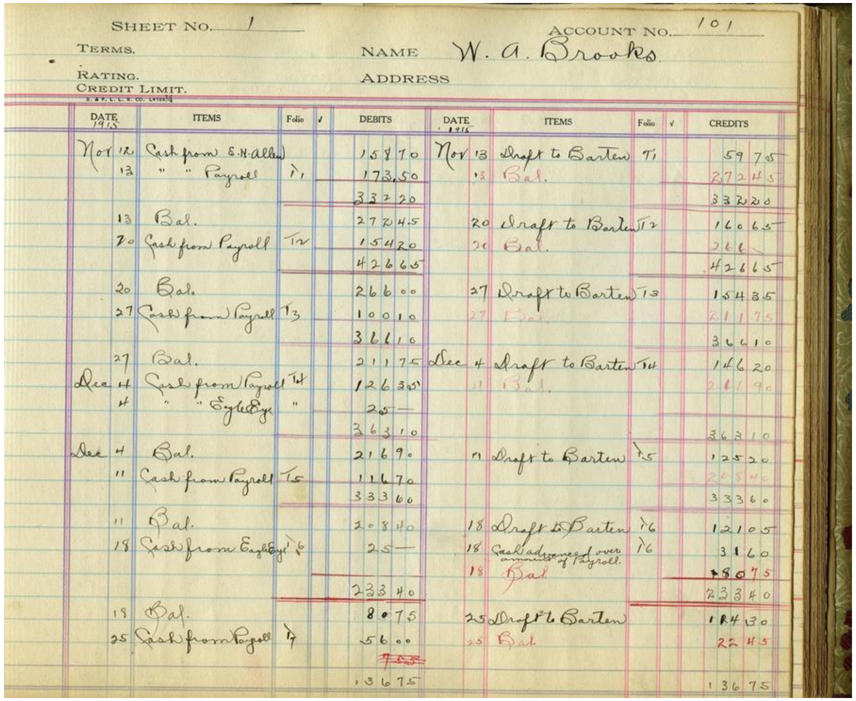
From bitsonblocks.files.wordpress.com

Blockchain explained (8)

Can you get bogus transactions accepted?

- Work with a few friends
- Try to win PoW for a block with bogus Tx
- But you need to win *N* times to get it accepted throughout the network
- And no good guy must have verified and found the bogus Tx

Not practically feasible, as long as 51% CPU power is honest



Picture from PBS, copyright unkown

Blockchain Models & Benchmarks

—

Bitcoin

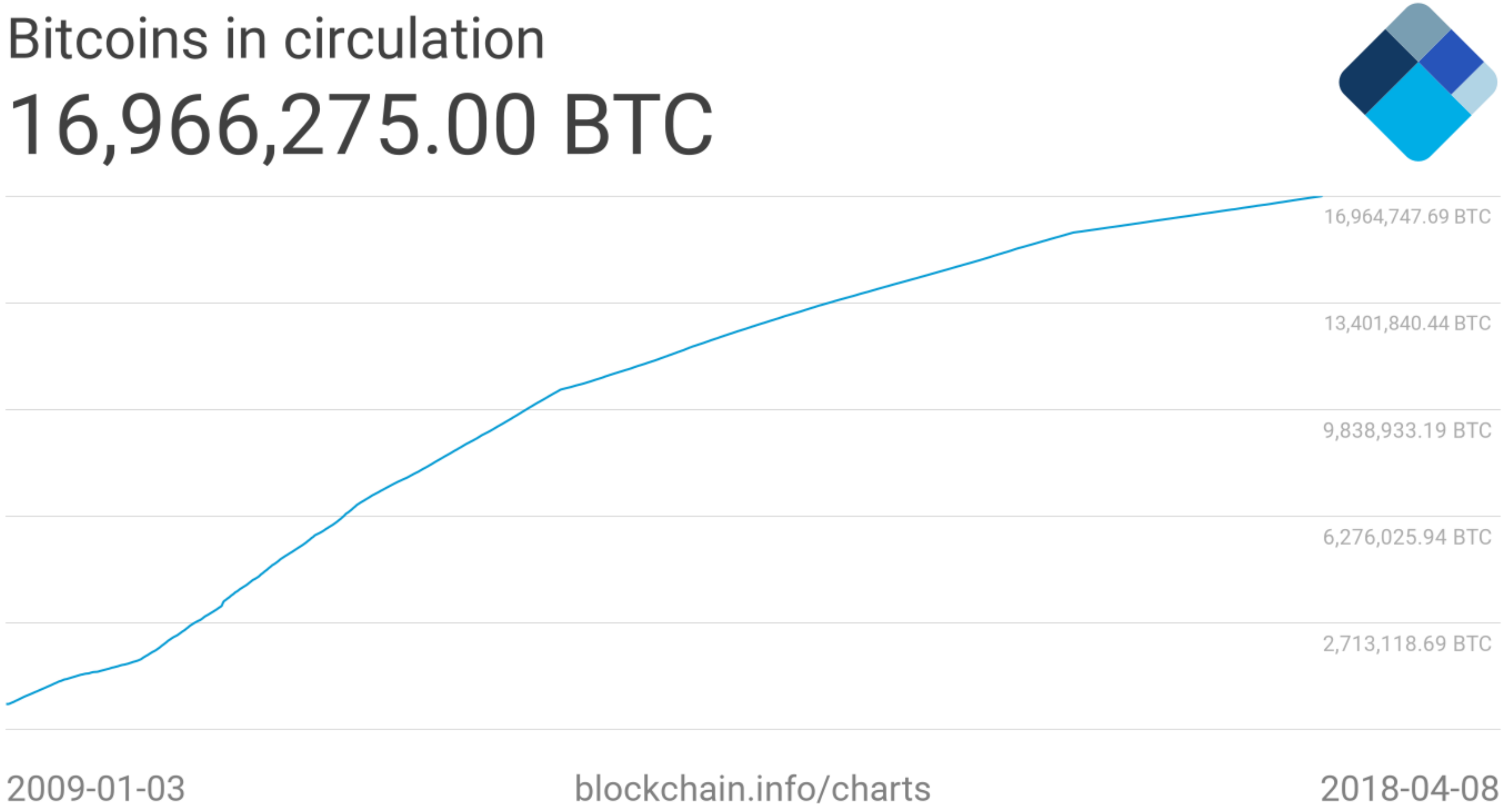
Bitcoin

Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, Oct 2008.

Software released Jan 2009.

- New coins with every generated block
- Block reward halved every 4 years
- Block every 10 minutes
- In 100 years: 21 million coins

Bitcoins in circulation

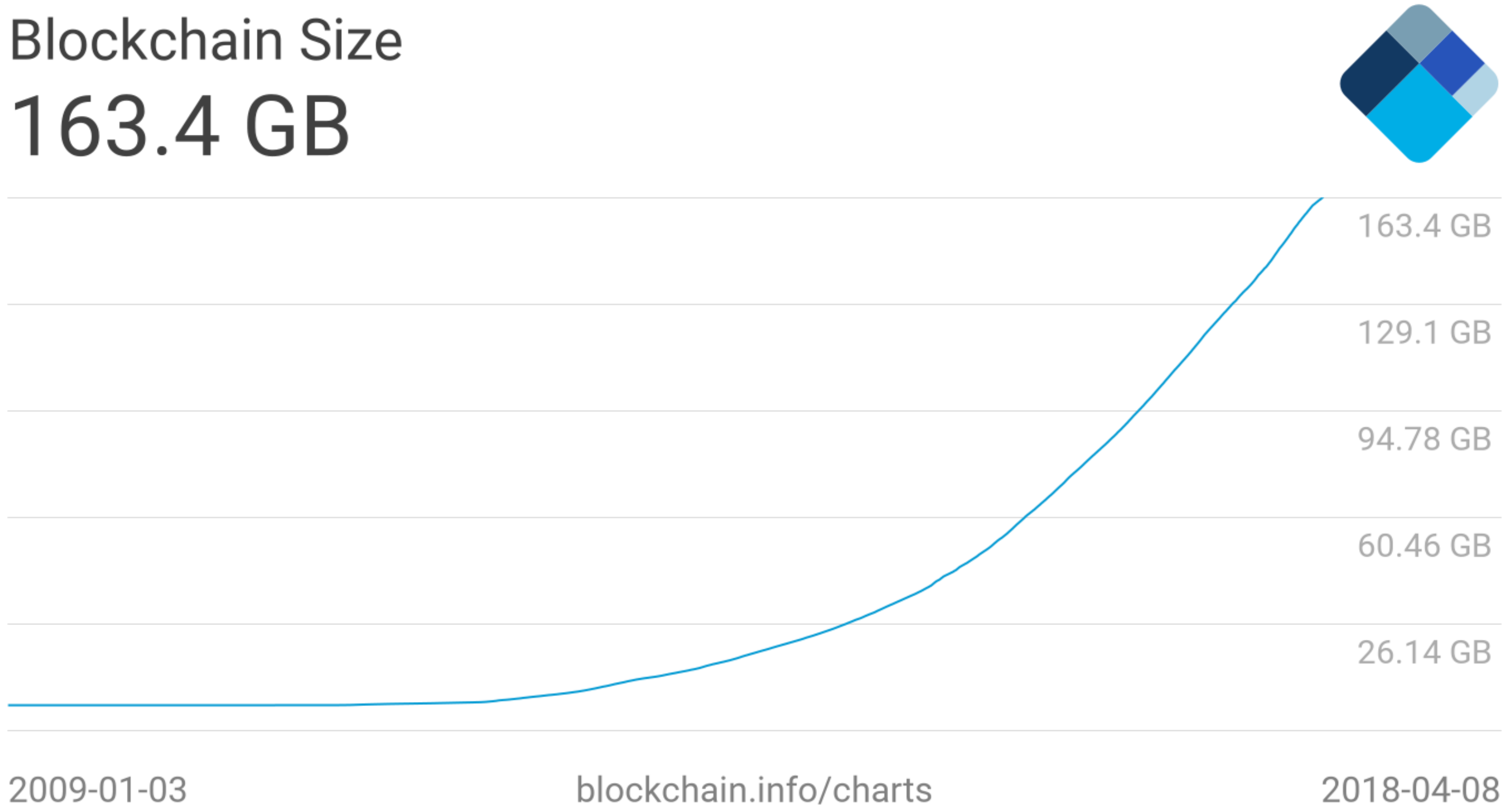


EPSRC National Cyber Security Centre a part of GCHQ

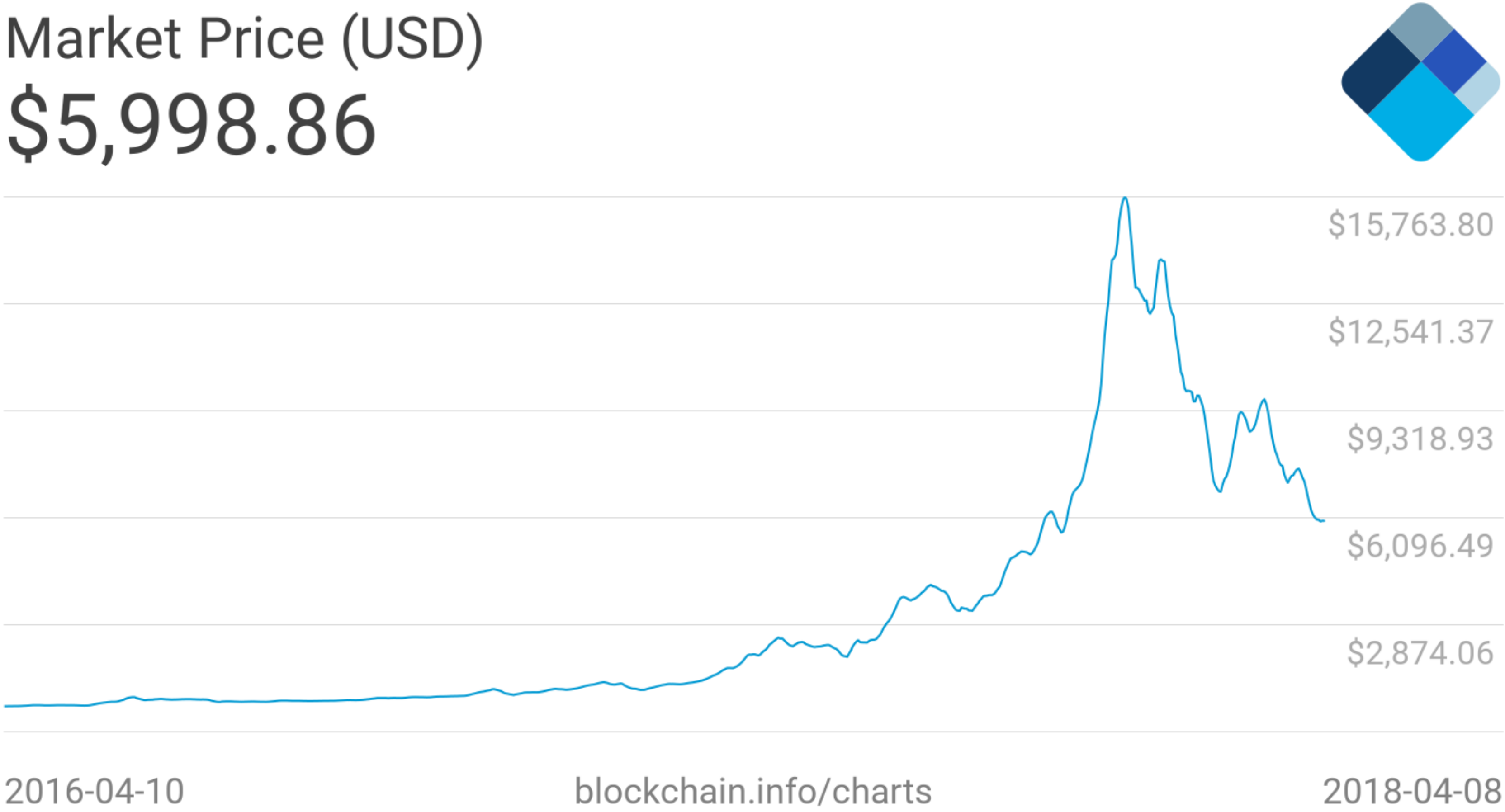
Academic Centre of Excellence in Cyber Security Research

Newcastle University

Bitcoin blockchain size



Bitcoin value



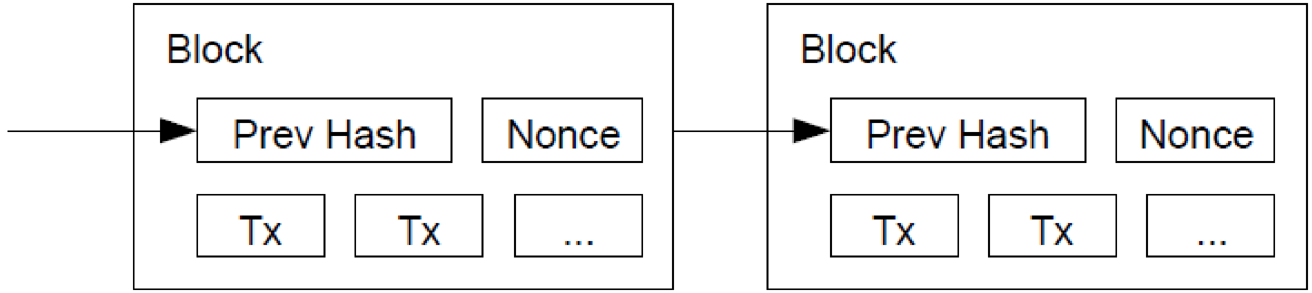
Bitcoin electricity usage

Description	Value
Bitcoin's annual electricity consumption* (TWh)	59.87
Annualized global mining revenues	$5,510,539,006
Annualized estimated global mining costs	$2,993,259,080
Current cost percentage	54.32%
Country closest in terms of electricity consumption	Colombia
Estimated electricity used over previous day (KWh)	164,014,196
Electricity consumed per transaction (KWh)	952
U.S. households that could be powered by Bitcoin	5,543,072
U.S. households powered for 1 day for a single transaction	32.18
Bitcoin as a percentage of the world's electricity consumption	0.27%
Annual carbon footprint (kt of CO2)	29,334
Carbon footprint per transaction (kg of CO2)	466.53

From https://digiconomist.net/bitcoin-energy-consumption, April 2018

Performance currently dominated by Proof of Work

- PoW: 'endlessly' try nonces until the hash of block satisfies a certain condition (eg, starting with at least 32 zeros) → first to do that gets block award
- Performance measured in hash/sec and hash/sec/$
- Energy use measured in hash/Joules



From Nakamoto 2008

Benchmark: Bitcoin hardware

2008: mine with CPU

2010: mine with GPU

- GPUs do less than 1GHash per second, ASICs > 1000 times more
- GPU data still available at bitcoin wiki, best performance: 3MHash/J, 2500 MHash/s, 4 MHash/s/$

2011: mine with ASICs

  - Ebit E10: 18000 GHash/s, 11 GHash/J (China only)
  - Ebit E9++: 6 GHash/s/$ (China only)

Bitcoin PoW hash ASIC

Bitcoin double SHA256 ASIC mining hardware

Product	Advertised Mhash/s	Mhash/J	Mhash/s/$	Watts	Price (USD)	Currently shipping
Ebit E9+ [23]	9,000,000	6900	6428	1300	1400	Yes
AntMiner S9 [9]	14,000,000	10182	5833	1,375	2,400	Yes
Avalon741	7,300,000	6350	5035	1150	1450	Yes
Avalon761	8,800,000	6670	4730	1320	1860	Yes
Ebit E9 [22]	6,300,000	7140	4468	882	1410	No
Avalon821	11,000,000	9170	3800	1200	2900	Bulk only
Ebit E9++ [24]	14,000,000	10500	3600	1330	3880	Yes
Ebit E10 [25]	18,000,000	11100	3440	1620	5230	Yes
AntMiner S5+ [7]	7,722,000	2247	3347	3,436	2,307	No
AntMiner S5 [6]	1,155,000	1957	3121	590	370	Discontinued
AntMiner S7 [8]	4,860,000	4000	2666	1,210	1,823	No

From: https://en.bitcoin.it/wiki/Mining_hardware_comparison

Blockchain Models & Benchmarks

—

Ethereum

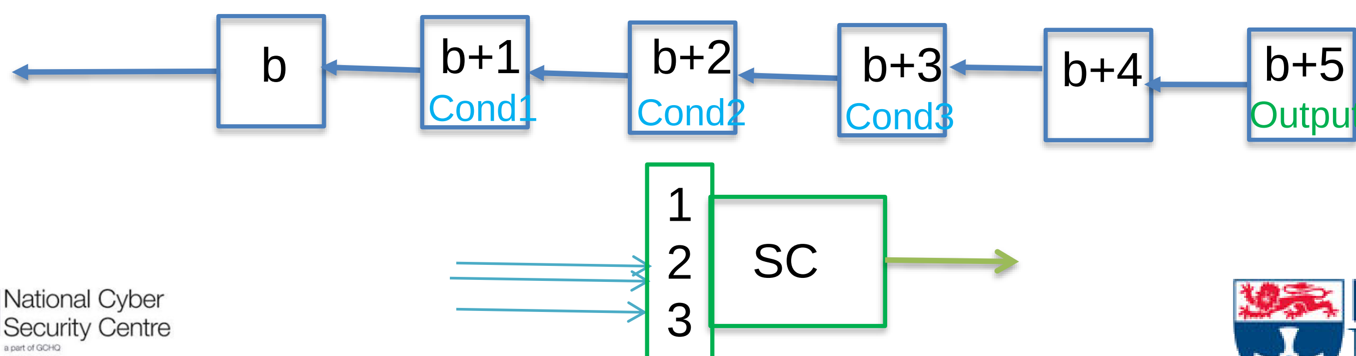
Ethereum philosophy

General cryptocurrency platform for a large set of distributed applications, with as design goals:

- As simple as possible for programmer
- Universal through Turing complete smart contracts
- Modular
- Agile (nothing cast in stone)
- Non-discrimination / non-censorship

Ethereum Smart Contracts

- Turing complete programs
- Smart contracts can call smart contracts
- Executed when specified conditions are satisfied in blockchain, e.g., monthly payment
- Execution output made available in blockchain, e.g. mortgage agreement
- Transaction fee for miner based on ‘gas’ used



EPSRC National Cyber Security Centre a part of GCHQ

Academic Centre of Excellence in Cyber Security Research

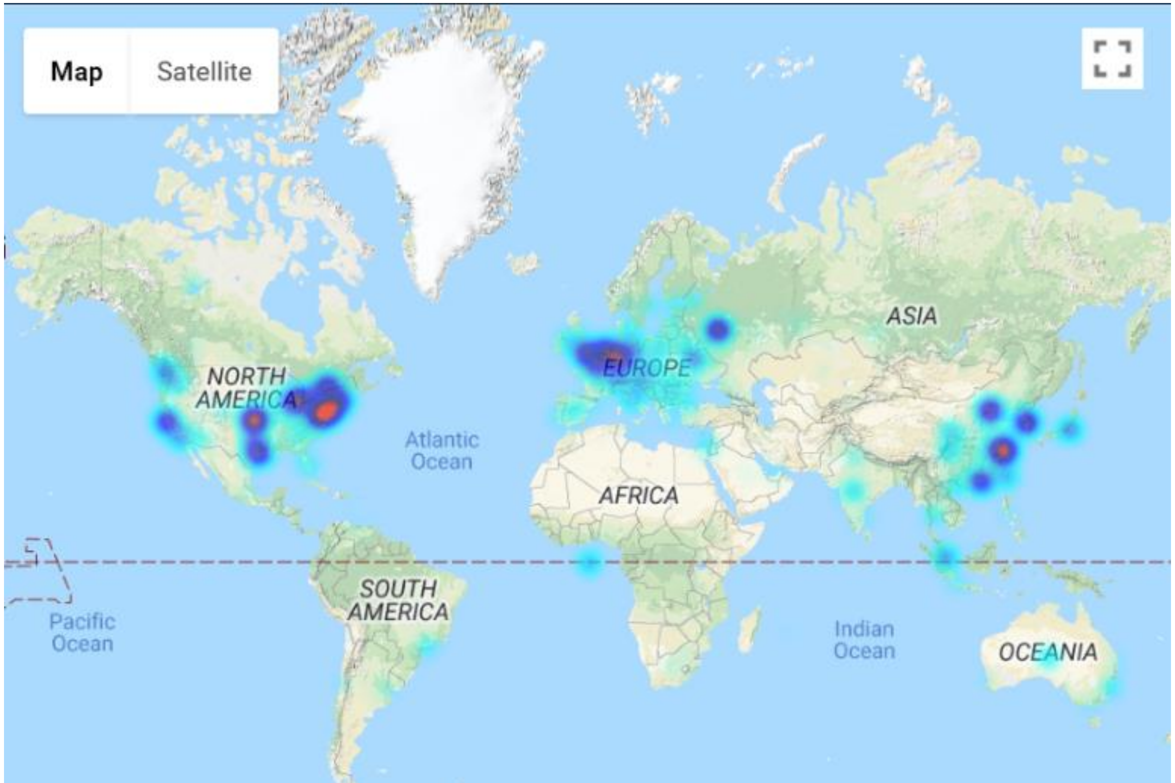
Newcastle University

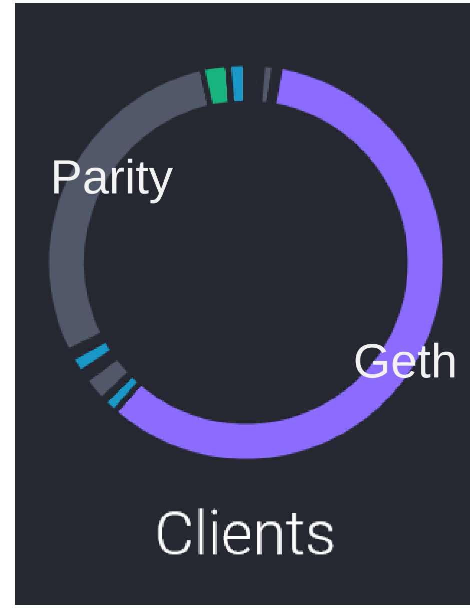
Ethereum Smart Contracts

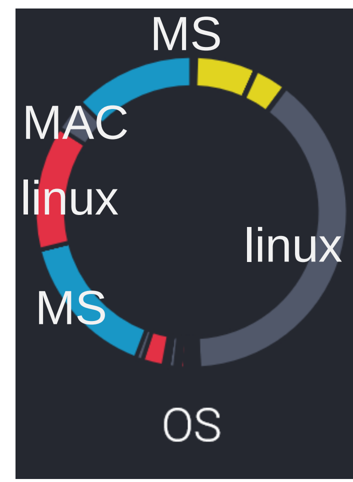
Many exciting applications thought off:

- Monthly recurring payments (‘direct debits’)
- Payment for parcel at delivery triggered by IoT sensors
- The process of mortgaging governed through smart contracts
- Etc…
- We did: 2 Phase Commit, game-theoretic contracts for verified cloud computing, e-voting

Ethereum







From: https://www.ethernodes.org/network/1

Ethereum performance

- Miners are interested in surplus:

  block/transaction award – energy cost
- Computational effort currently dominated by PoW, but:
  - memory-bound, so ASIC for hashing not effective
- Ethereum will move from PoW to Proof of Stake:
  - Transaction execution performance more important
  - Smart contract execution needs to be optimized and benchmarked

Ethereum benchmarks

Monitoring information on web sites

Fairly little benchmarking activity:

- Ethereum clients (with Ethereum Virtual Machine)
- Smart contract execution time

Benchmark: Ethereum clients

Time it takes to process blocks includes:

- PoW (which is memory-bound: no ASIC, but GPU beats CPU)
- transaction signature checking
- Merkle tree operations, with varying storage options
- EVM code execution
- receipt verification
- uncle validation
- database population

Main parameter:

- Set a database of blocks (eg 1000000 from mainnet chain)

Ethereum Client Benchmarks

	Eth	EthereumJ	Geth	Parity
Time	4h 33m	7h 7m	8h 43m	2h 31m
CPU (avg)	123%	90%	70%	107%
Memory (avg)	921MB	3.168GB	1.5GB	365MB

- From https://github.com/ethereum/wiki/wiki/Benchmarks

Spec:

- Digital Ocean 4GB droplet running Ubuntu 14.04.3 x64
- Start-up time for 1 million blocks, verifying them, etc
- Eth→C++, EthereumJ→Java, Geth→Go, Parity→Rust

There exist several more clients, no benchmarks known

EPSRC National Cyber Security Centre a part of GCHQ
Academic Centre of Excellence in Cyber Security Research

Newcastle University

Smart contracts rewards

- Transaction submitter sets a gas price and a max gas
- System (EVM) counts how much gas is used at smart contract execution
  - for most opcodes, uses a table, per opcode
  - for some opcodes, it uses a formula depending on inputs
- Transaction reward = gas used x gas price
- Miner's transaction cost = energy used

Smart contract rewards

Reasons for rewards (paid by the transaction issuer) are twofold:

- Avoid malicious smart contract to use excessive resources—denial of service attack
- Reward the miners for executing the transactions with smart contracts

For both situations: potential problems if fee paid is not proportional to energy used

Denial of Service Attack on Poorly Benchmarked Smart Contracts

https://blog.ethereum.org/2

Ethereum Blog

Uncategorized

The Ethereum network is currently undergoing a DoS attack

Posted by Jeffrey Wilcke on September 22nd, 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.

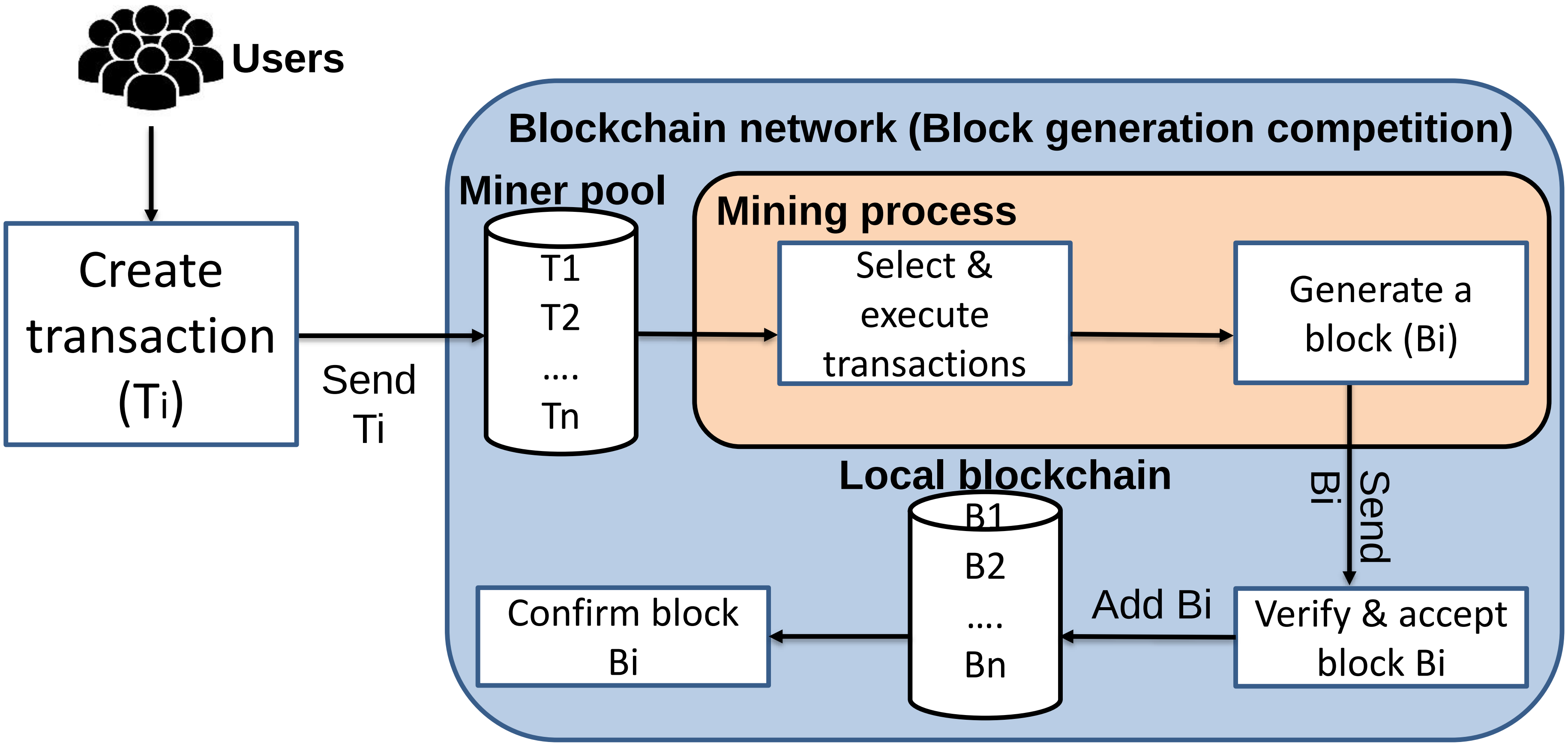
From blog.ethereum.org

What if energy use is not proportional to calculated gas used?

A modelling study

"The impact of profit uncertainty on miner decisions in blockchain systems",
Maher Alharby and Aad van Moorsel
UKPEW 2017, extended in *Electronic Notes in Theoretical Computer Science*, 2018

Blockchain Workflow



Research Challenges

- Miners know only the maximum income of executing a transaction.

  *Maximum income = gas limit * gas price*

- Miners do not know the exact income they can get from executing a transaction.
- Miners do not know the cost of executing a transaction.
- Miners are uncertain about the profit they can get from executing a transaction.

Experimental Design

Groups	Scenarios	Available Information	Sorting/Execution Criteria	Income Certainty	Cost Certainty
Baseline	Gas price	Gas limit	Highest gas price	NO	NO
	Maximum income	Gas price	Highest gas limit * gas price	NO	NO
Solution	Exact income	Used gas Gas price	Highest Used gas * gas price	YES	NO
	Maximum profit	Gas limit Gas price CPU time	Highest (gas limit * gas price) / CPU time	NO	YES
	Exact profit	Used gas Gas price CPU time	Highest (used gas * gas price)/ CPU time	YES	YES

EPSRC National Cyber Security Centre a part of GCHQ
Academic Centre of Excellence in Cyber Security Research

Newcastle University

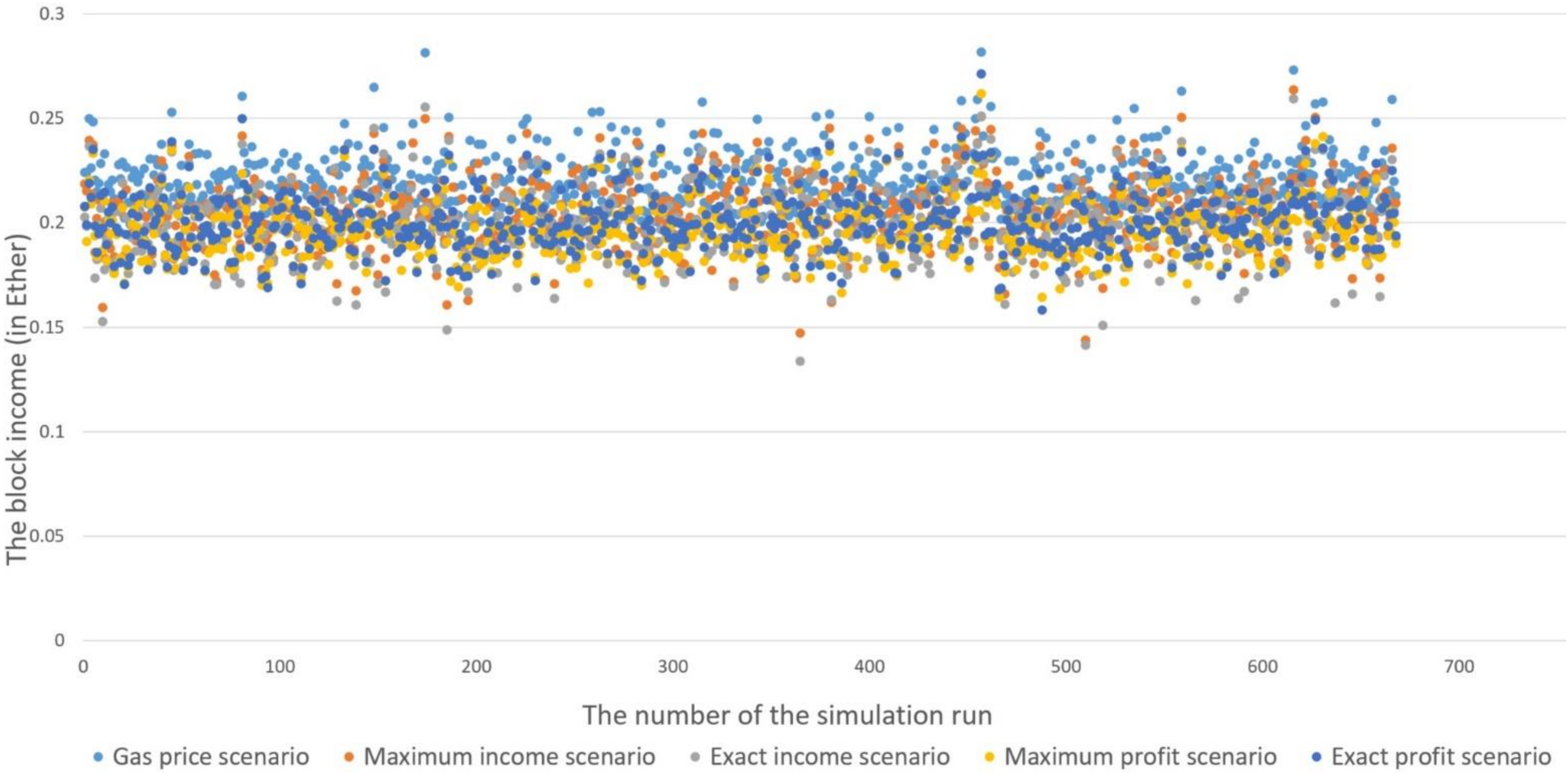
Results and Discussion

Cost Uncertainty: The uncertainty miners perceive about the cost of executing transactions has a significant impact on the block profit.

- Certainty about the cost of executing transactions can help miners quadruple their block profit.

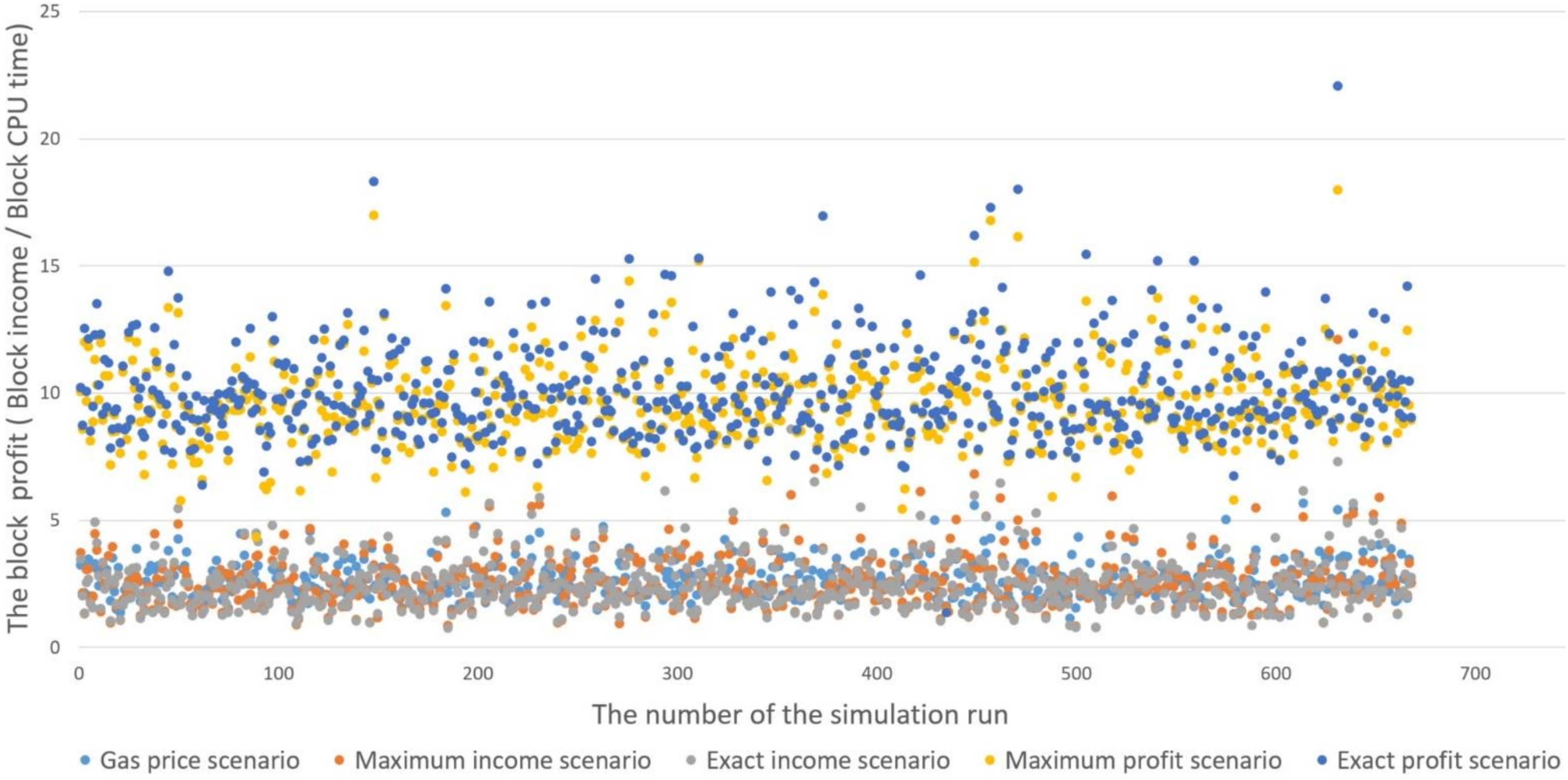
Income Uncertainty: The uncertainty miners perceive about the income of executing transactions does not have an impact on the block profit.

Results: Block Income



Results: Block CPU Time

Results: Block Profit



EPSRC National Cyber Security Centre a part of GCHQ

Academic Centre of Excellence in Cyber Security Research

Newcastle University

Results

Scenarios	Average Block Income (in Ether)	Average Block CPU time (in Second)	Average Block Profit (Income / CPU time)
Gas price scenario	0.221	0.091	2.604
Maximum income scenario	0.205	0.091	2.538
Exact income scenario	0.199	0.091	2.488
Maximum profit scenario	0.196	0.021	9.538
Exact profit scenario	0.201	0.021	10.070

Table 2
A summary of the experiment's outputs for all the five scenarios. The experiment's outputs are the average block income, the average block CPU time and the average block profit. The average value for each output is taken from 668 simulation runs. The confidence intervals (95%) for the experiment's outputs are not given here, but all intervals are within 3% of the average value.

We used etherscan.io data about gas price, max gas and gas used to parameterize the simulation

Conclusion

- Best strategy: execute the smart contracts that have the best award/CPU ratio
- Uncertainty about the energy use: you cannot choose the best contracts

Open questions:

- Can we benchmark cost of smart contract and opcode execution?
- Can we build it in the decision maker when choosing transactions?

Blockchain Models & Benchmarks
—
Ethereum ‘used gas’ benchmarks

Gas per opcode

Yellow paper defines gas for the 70 (or 117) opcodes (Appendix G), EMV tracks it

- Categories of upcode:
  - base (2 gas), eg, POP, ADDRESS, GASPRICE
  - verylow, (3 gas), eg., AND, OR, ADD
  - low (5 gas), eg., MUL, DIV
  - mid (8 gas), eg., JUMP, ADDMOD
  - high (10 gas), eg JUMPI
- One-offs, eg. BALANCE (400), EXTCODESIZE (700)
- A formula for some, eg. EXP, SHA

Current informal benchmarks for gas per opcode

- No official benchmarks reported
- Interesting, somewhat convoluted approach reported at Github:
  - Cycles/OP as comparable metric
  - No clear isolation of individual opcodes (some stack ops are mixed in)
  - Limited set of opcodes considered
  - For considered opcodes it runs large tests: 320 million test of each operation
  - Clever tests that check that the final result is correct

Ethereum opcode benchmark

Ethereum 'Performance suite':
https://github.com/ethereum/cpp-ethereum/tree/develop/test/unittests/performance

a. *.asm tests for individual opcodes
   - nanoseconds/test; nanoseconds/gas; nanoseconds/opcode

b. *.sol tests for larger units (PRNG, Encryption)

Results with only 3 out of 8 existing clients, rudimentary tests for limited amount of opcodes

EPSRC National Cyber Security Centre a part of GCHQ
Academic Centre of Excellence in Cyber Security Research

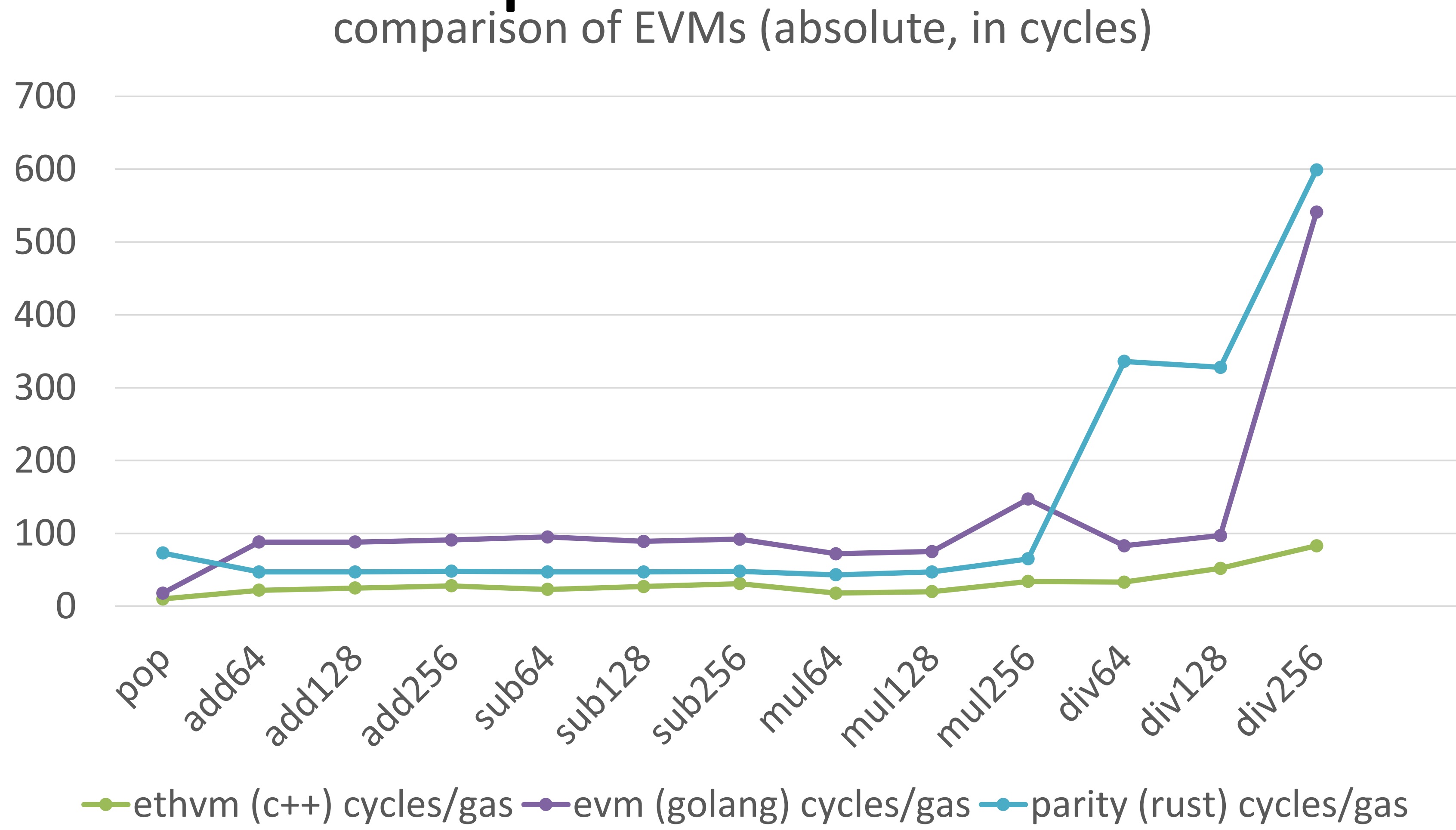
Newcastle University

Benchmarks for opcode gas

	ethvm (c++) cycles/gas	evm (go) cycles/gas	parity (rust) cycles/gas
pop	10	18	73
add64	22	88	47
add128	25	88	47
add256	28	91	48
sub64	23	95	47
sub128	27	89	47
sub256	31	92	48
mul64	18	72	43
mul128	20	75	47
mul256	34	147	65
div64	33	83	336
div128	52	97	328
div256	83	541	599

From https://github.com/ethereum/cpp-ethereum/issues/4073

Opcode benchmark: comparison of EVMs



Data from https://github.com/ethereum/cpp-ethereum/issues/4073

Benchmark for opcode gas

Approach:

- Measure CPU use for the specific opcode only (discount the stack ops)
- Discount startup/shutdown of EVM, discount for-loop, avoid any optimization
- Write the tests that run with a small stack

To do:

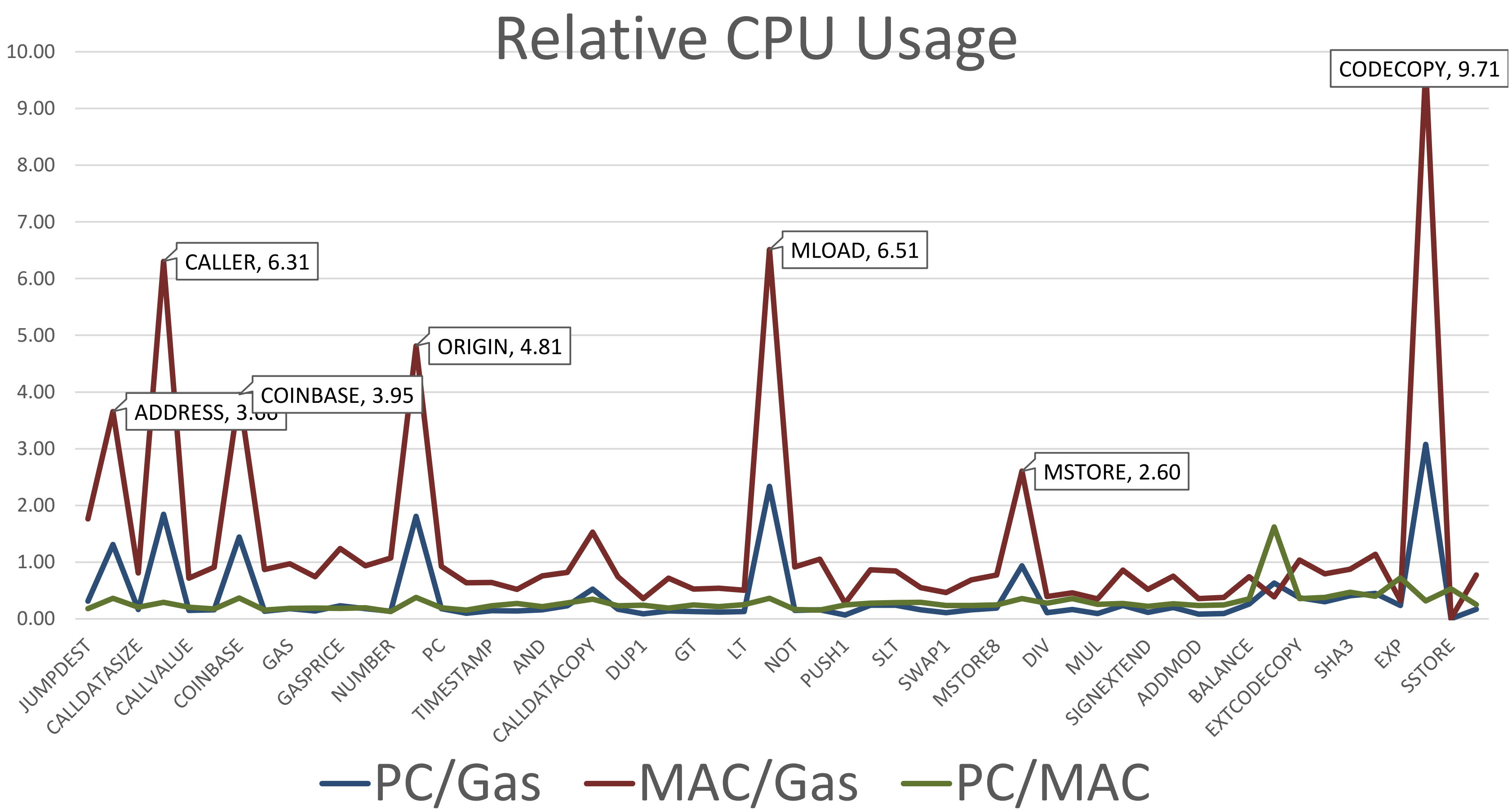
- Input-dependent benchmark for the opcodes that depend strongly on inputs (eg, EXP)
- Account for different EVM implementations, account for different platforms: cross-platform metric
- Benchmark for contracts…

Benchmark for opcode gas

Experiment:

- Implementation in PyEthApp EVM client, using PyEthereum libraries
- All opcodes
- Two OS's:
  - MAC: a MacBook Pro with a 2.8 GHz Intel i5 CPU and 8 GB RAM. OS: MACOS High Sierra
  - Desktop:a desktop with a 3.20GHz Intel i7 CPU and 8 GB RAM . OS: Ubuntu Mate 16.04.09
- Results
  - Absolute, in msec
  - Relative: straight lines mean platforms behave similar for various opcodes

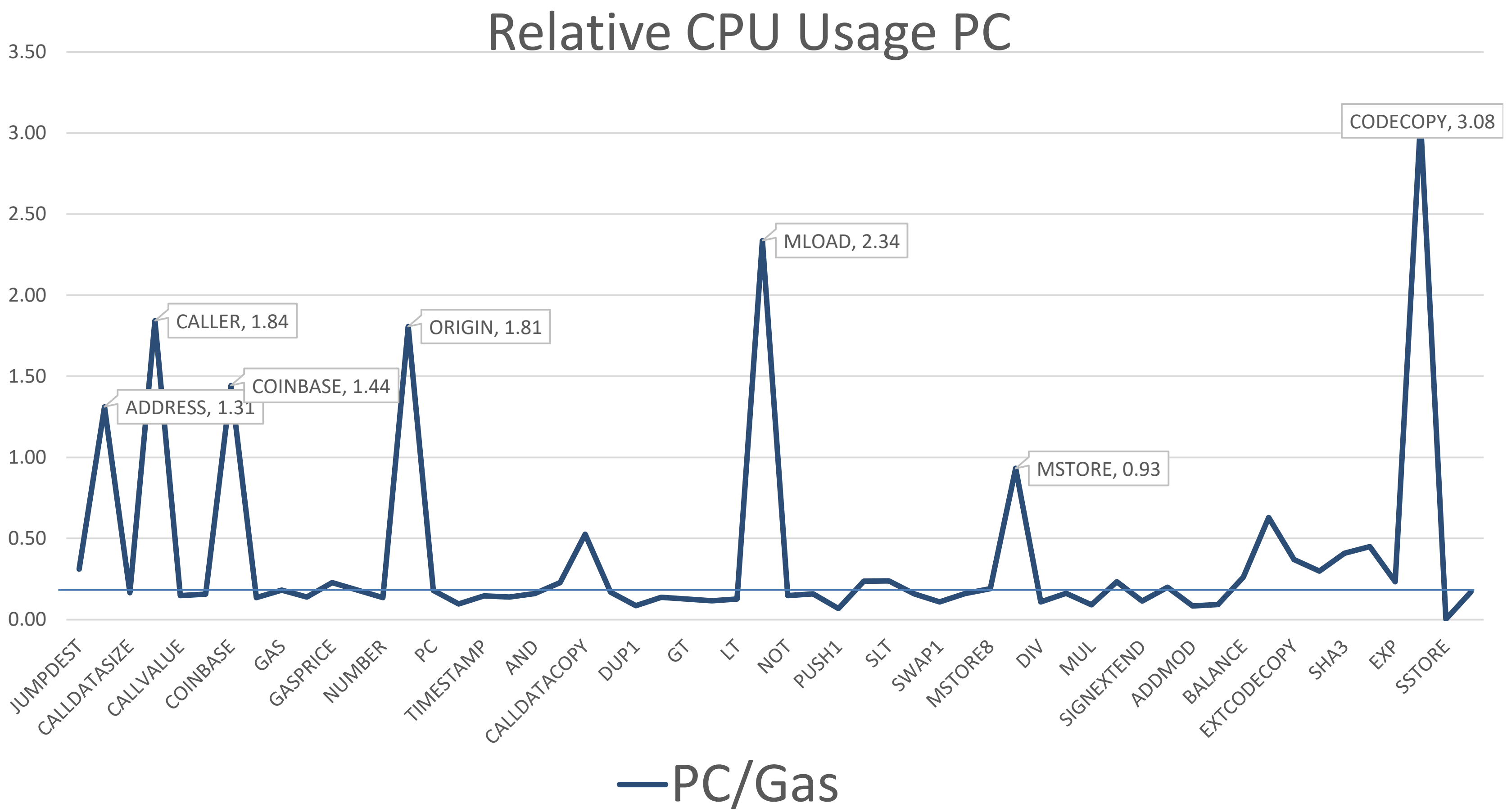
Benchmark Results PyEthereum



EPSRC National Cyber Security Centre a part of GCHQ
Academic Centre of Excellence in Cyber Security Research

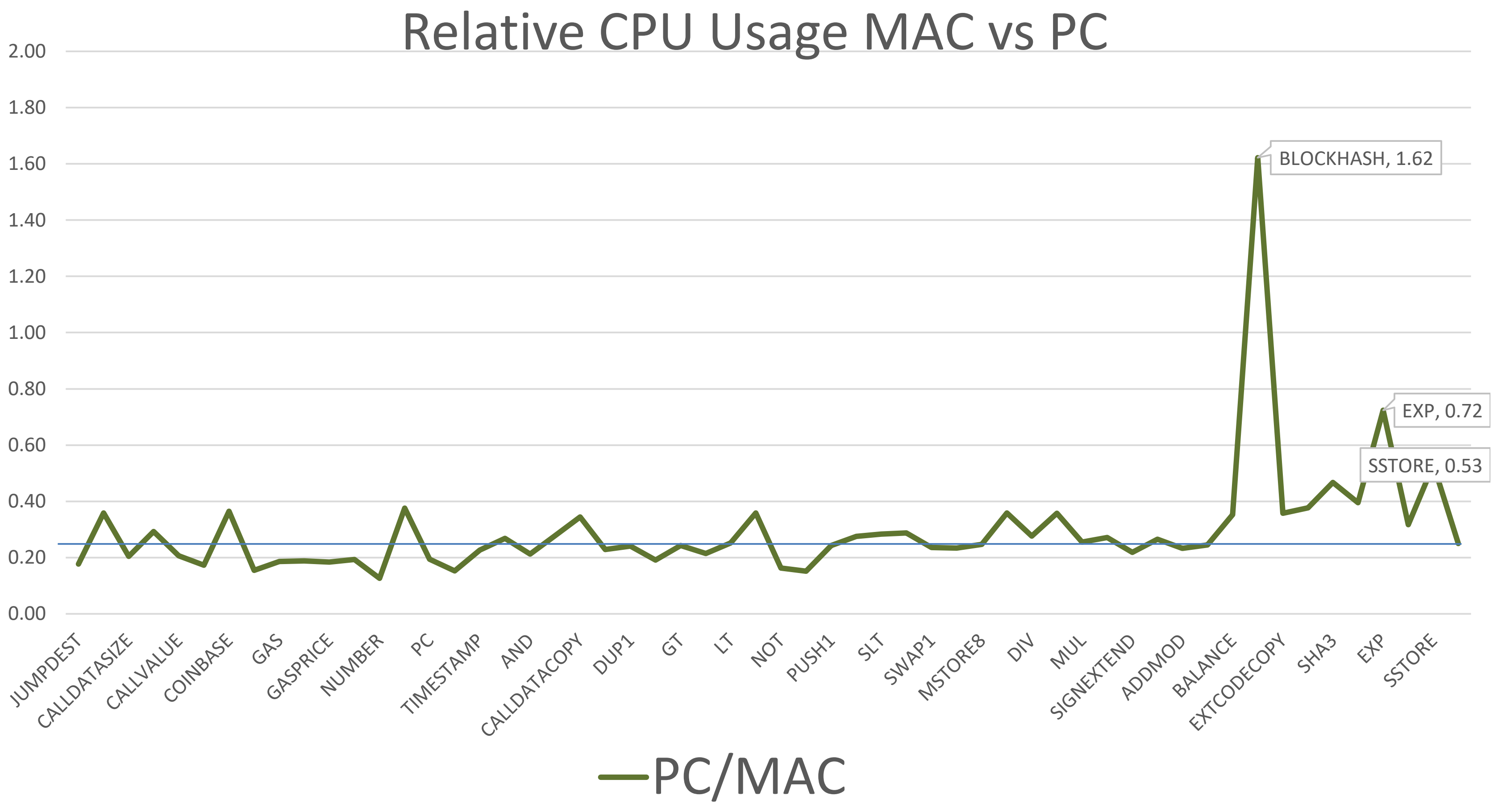
Newcastle University

Benchmark Results PC



Academic Centre of Excellence in Cyber Security Research

Benchmark MAC vs PC

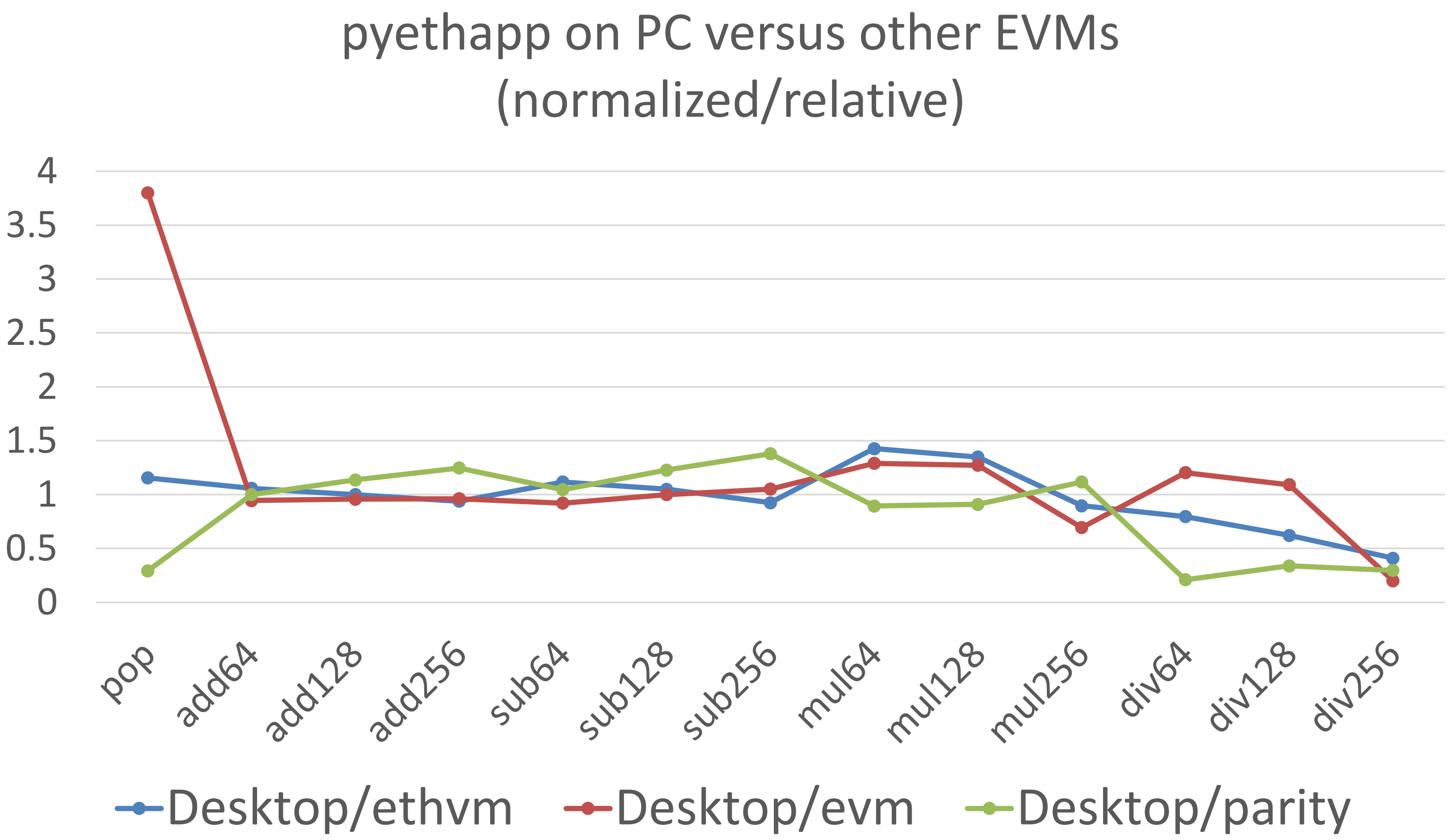


EPSRC National Cyber Security Centre a part of GCHQ

Academic Centre of Excellence in Cyber Security Research

Newcastle University

Comparison different clients



EPSRC National Cyber Security Centre a part of GCHQ

Academic Centre of Excellence in Cyber Security Research

Newcastle University

Conclusion Smart Contract Benchmarks

- Gas set as payment for opcodes not overly accurate
- No good benchmarks yet for opcodes
- Number of challenges are being resolved
- Transaction (i.e. smart contract) execution will determine the miner income → desire to find best platform for typical smart contract executions
- Number of contract benchmark questions out there to be addressed

Blockchain Models & Benchmarks

—

Other Blockchains

Variety of blockchains

- Examples of blockchains:
  - Cryptocurrency variants:
    - all coins pre-mined
    - difference in total amount of coins
    - differences in block and transaction fees over time
  - Proof of Stake instead of Proof of Work
  - Smart contracts: general purpose transactions
  - On-chain and off-chain variants, eg. Hyperledger: flexible configurable

Different blockchains, for different applications

Network	Coin	Issuance	Block-making incentive
Bitcoin	BTC	Created according to schedule. Total 21 million BTC in 2140.	Block reward + transaction fees
Ripple	XRP	100% pre-mined. 100 billion XRP created.	None
NXT	NXT	100% pre-mined. 1 billion NXT created.	Transaction fees
Ethereum	ETH	72 million pre-mined plus ongoing issuance of 18 million ETH per year.	Block reward + computation fees

From bitsonblocks.net/2015/09/28/a-gentle-introduction-to-digital-tokens/

Blockchain performance in layers

Incentives layer
(stakeholder concerns, profit-making, …)

Connector layer
(consensus algorithm, smart contracts, …)

Processing layer
(bare metal, OS, …)

Connector Layer

Performance of Consensus

- Consensus in blockchain based on PoW, which purposely makes it slow to reach consensus
  - Allows arbitrary nodes to participate
  - Creates effort invested that nodes don't want to loose
- Even without PoW, consensus does not scale well → too many messages for desired speed of updates
- Performance of Bitcoin 1/10,000th of VISA's transaction volume…
- Various improvement proposed, but PoW cannot be remedied…

Performance of Consensus Layer

Results from ‘Consensus in the age of Bitcoin’, 2017, with a lot of subtleties/caveats. Note, VISA is designed for 10,000 tx/s.

	Measured throughput	Measured latency
Bitcoin	7 tx/s	600s
Hyperledger	110 tx/s	<1s
Byzcoin	1000 tx/s	10-20s

EPSRC National Cyber Security Centre a part of GCHQ
Academic Centre of Excellence in Cyber Security Research

Newcastle University

Blockchain as a Software Connector

Different applications need different blockchains:

- is PoW needed (open to any participant)?
- what in the system is subject to consensus?
- what physical artifacts are represented?
- does it need a coin?
- how and what to search?
- …

Find the best design for your app and evaluate the resulting properties

Need for Model-Based Evaluation of the Connector Layer

- Consensus properties usually proven under assumptions
- But assumptions behind these properties hold probabilistically
- Follow 'Probabilistic Verification' approach by Sanders et al, Probabilistic Verification of a Synchronous Round-Based Consensus Protocol, SRDS 1997
- Configure the Connector Layer based on the analysis

Recap

Incentives layer

Connector layer

Processing layer

Stakeholder perspectives

Users: can this system be trusted in practice?

Miner gains block and transaction fees

Designers: no need for trusted third party

Developer: is consensus guaranteed 100%?

Politician: is energy use damaging societal values/goals?

Miner: is used CPU/storage proportional to received fee?

EPSRC National Cyber Security Centre a part of GCHQ
Academic Centre of Excellence in Cyber Security Research

Newcastle University

Incentives layer

- Monitoring: learn from what is happening:
  - Longitudinal studies (change over time)
- Model-based analysis of:
  - Long term behaviours and incentive shifts
  - Miner and miner pool strategies
- Need for tools that support
  - Game theory, incentives theory
  - Markov decision processes
- Start considering societal concerns in incentives, eg environmental

Connector layer

- Monitoring: learn from what is happening:
  - Availability studies such as Weber et al, SRDS'17
- Need for model-based tools that support
  - Configure the connector for the application at hand
  - Optimization models for transaction selection, in particular under smart contracts
  - Probabilistic Validation approach to augment 'proofs' under non-real assumptions, in particular for consensus
- Benchmarks:
  - Comparison of throughput and latency for consensus variants
  - Comparison of other basic modules: crypto, smart contracts, PRNG, hashing, …

Processing layer

- Benchmarks of bitcoin ASIC PoW hashing will continue by industry

Progress needed in:

- Benchmarks of ‘blockchain virtual machine’ software (i.e., clients)
- Benchmarks of smart contract opcodes

My wish list

1. Simulation framework:
   a. Game-theoretic, Markov decision, for incentives layer
   b. Probabilistic Verification approach for connector layer
   c. Integrated from processing to incentives, for many stakeholders
2. Benchmark framework:
   a. Smart contract benchmarks
   b. Connector benchmarks

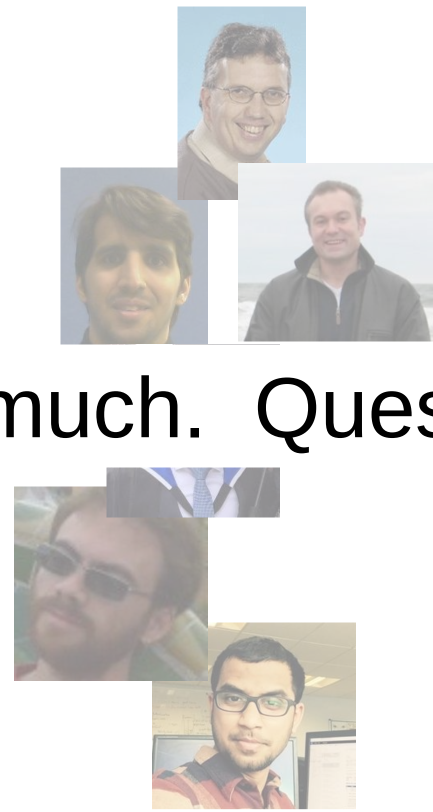
Conclusion and Outlook

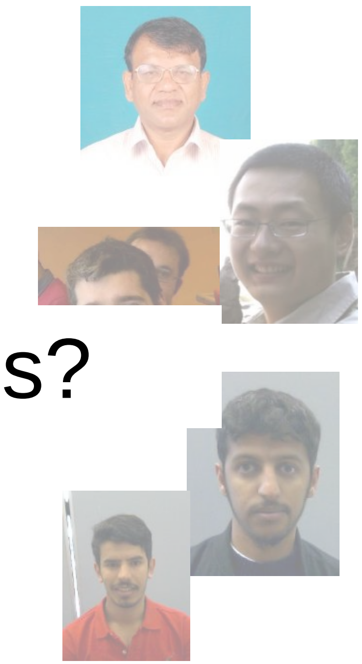
Many computer scientists:
"Blockchain is here to stay, but Bitcoin is not"

HMG new research group:
"Bitcoin is here to stay, but Blockchain is not"

Assume they're both here to stay: arguably, blockchains developments can use some sound performance engineering as underpinning

Blockchain Models and Benchmarks





Thanks much. Questions?