Investigation of the potential for using the Bitcoin Blockchain as the world's primary infrastructure for Internet commerce

Craig Wright (craig@ncrypt.com), Antoaneta Serguieva (antoaneta@ncrypt.com)

Abstract. Bitcoin has the potential to replace all existing payment systems to become the world's primary infrastructure for commerce over the Internet. However, expanding in scale to meet the demands of such a system would require the constraints that are currently imposed to be lifted. This prompted us to build a model based on live data to simulate the existing network (by using high-performance computing) and extrapolate it to alternative future networks to test the hypothesis that the infrastructure would be capable of accommodating increased block sizes in its blockchains. In addition, we also simulated the effect of introducing a fast payment network that enables merchants to pay a premium for faster propagation of their transactions. We show that the network can scale up indefinitely with no limit on block sizes and that a fast payment network mitigates double-spend attacks.

KEYWORDS

1. Bitcoin, 2. Blockchain, 3. Payment systems, 4. Fast payment network, 5. Block size, 6. Double-spend attack

1. Introduction

Our research involved conducting an extensive research program utilizing virtualized high-performance computing (HPC) to investigate the possibility of scaling up the Bitcoin Blockchain protocol to enable it to sustain the global economy; that is, we determined the feasibility of replacing the global financial infrastructure with one based on the Blockchain. This could be expected to contribute a host of benefits including (but not limited to) increased cyber-security, permanency of records, and transparency (e.g., for tax purposes). Based on our research, our calculations showed that such a vision is feasible provided the current restrictions on transaction size and block size are removed. In this paper, we explain the research and present the first set of results. Further results will

be published separately to limit the size of papers. Here, we present the results of research on the processing of cryptographic functions, the effect of block size variations, and alternative configurations of the proposed fast payment network (FPN). We conclude with a set of recommendations for the future of the Blockchain, to be supported by forthcoming papers.

The research program involves modeling the existing Bitcoin network and creating models for a potential alternative future Bitcoin infrastructure. The existing Bitcoin network was modeled by simulating every node in the real Blockchain network and configuring the operational parameters of the model using live data captured from the real network in near real time. Parameter data included variables such as transaction sizes, block sizes, latency, and transaction queues. This is believed to be the first model of the Bitcoin system that has been created to run large-scale tests. These data were used to study the operation of the current version of Blockchain in the real environment and to derive relationships between the different variables. The model of the existing Blockchain network was compared to the actual Blockchain network to verify its operational accuracy and hence the validity of the parametric settings and their interrelationships. This information was then extrapolated and used in the HPC models of alternative potential future Blockchain network configurations. These futuristic models were used to investigate how the Blockchain might operate under different conditions and with different configurations (for example, if the current restrictions on block size were relaxed in future, or if tiers of paid and unpaid services were introduced). Our overall conclusion is that the Bitcoin Blockchain can be scaled up to replace all existing payment system networks to become the world's single global economic infrastructure.

The primary concern here relates to the latency effects and transaction processing times that would be associated with increasing the block size that would result from a combination of more complex transactions and larger transaction pools. In this research, we modeled both the effect of increasing the number of transactions as well as the complexity involved in each individual transaction. We extrapolated diverse types of transactions that add complexity by simulating the validation of the cryptographic hashes and capturing the time used in various process types.

In addition to investigating the scalability of the Blockchain, it was important to examine several related issues, such as resistance to cyber-attacks (such as double spending), the ability to track possible criminal activities such as money laundering, and robustness against financial shocks such as currency fluctuations and credit crises. These latter two issues lie outside the scope of this paper and the results will be published later. The results of investigating the vulnerability to double-

spending attacks are central to the management of the Blockchain itself and are reported in this paper. The work presented in this paper is concerned with the ability to control and manage the introduction of transactions to the mining network. In addition, the model probes the implications of propagating a transaction to most nodes faster than an attacker as well as the failure to do so. The primary reason for modeling such an interaction is economic. A merchant would need information as to the risk associated with a transaction. The node network would then be able to add a cost function to this calculation to enable us to add a level of certainty even to insecure networks.

2. Background

2.1. Current Bitcoin network size and constraints

As of December 2015, the Bitcoin network was composed of approximately 5200 active nodes. The current maximum block size is 1 MB. To be valid, a transaction output is required to exceed 546 satoshi (known as 'dust')[1]. These factors place a limit on the capacity of the network. Furthermore, most of the nodes operate without reward, contributing their CPU power for free. These factors confer a self-imposed limit upon the network. These limits were imposed on the early version of the network because of the small number of nodes and the fact that most nodes were run as a hobby by most people conducting trials on the system. This has changed significantly. The mining of Bitcoin and hence the security of the network has moved from a "home as a hobby" function into a significantly funded corporate system. The results of this change have led to considerable economic inefficiency associated with the limitations on the initial design. The network did not account for the economic effects of storage or processing. As the network matures and grows along with increasing public acceptance, the future profile of the Bitcoin network will be significantly different. To meet the future demands on the network and to ensure its capability to sustain the multitude of innovative uses planned for it, the network needs to be enhanced.

The primary limit on the unspent transaction outputs (UTXO) has its origins in memory constraints and significantly affects small nodes. Sufficient memory is necessary to quickly look up each of the unspent transactions and process it in a timely manner. In this paper, we propose a scenario using large specialized nodes capable of transmitting and receiving gigabytes of data quickly and processing elliptic curve signature data in close to real time. We aim to show here that,

[1] See https://github.com/bitcoin-dot-org/bitcoin.org/issues/592

in such a system, the size of the unspent transaction pool matters less than the amount being processed.

Fig. 1 shows the change in various Bitcoin metrics as a function of time. The figure clearly shows that the size of the transaction pool has been growing in a linear fashion. In our research we show that, even if this pool was to grow with the extent of transaction usage, our system of large specialized nodes would be able to accommodate the load. Cached access using fast SSDs and a memory cache is sufficient to scale to well over 200,000 transactions per second (TPS) using present consumer hardware. Such a system would require at least 32 Gb and potentially up to 64 Gb RAM which is well within the range of consumer equipment. The UTXO is not stored in RAM; in the default software, this information is saved to disk using the LevelDB command. The mempool (pool of pending transactions that have not been included in a block) is stored in RAM. Both values are configurable. In the scenario, we have been testing a distributed database utilizing PostgreSQL enables far higher throughput and performance.

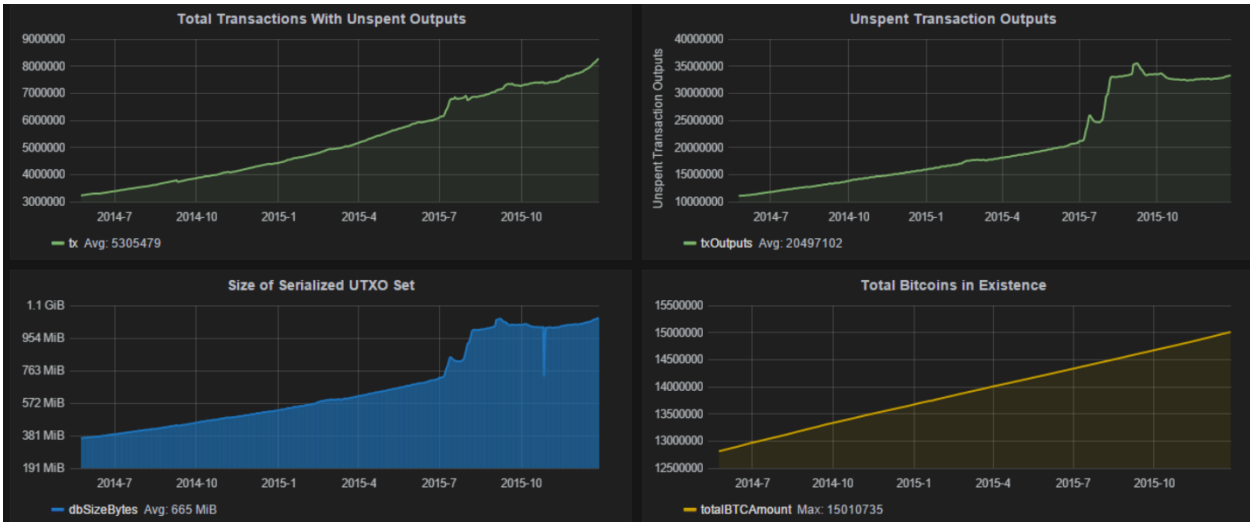

Fig. 1. Bitcoin growth metrics showing (a) total transactions with unspent outputs, (b) unspent transaction outputs, (c) size of serialized UTXO set, (d) total Bitcoins in existence

As the network grows, it is necessary to introduce a market-based methodology that will allow for the growth of systems and services in a competitive framework that takes user needs into account. This would enable individuals with an economic incentive to have their transaction guaranteed more rapidly and to ensure that those who would not be able to wait for an individual transaction to be written to the block are assured of the security of a transaction by introducing risk functions and payment guarantees.

2.2. The double-spending issue

Bitcoin transactions are broadcast using an epidemic model that is commonly referred to as a gossip protocol within computer science (for example, see Leitão et al. 2007). Each node that receives a transaction verifies that:

(i) the total output value of the transaction does not exceed the total input value and hence that the amount spent does not create a negative amount on the ledger;

(ii) the electronic signature used to sign the transaction matches the public key of the sending account; and

(iii) outputs can only be spent once.

If these three aspects of the transaction are shown to be valid the node forwards the transaction to connected neighbors. The first two of these steps can be accomplished based solely on the data in the transaction itself but the third – the prevention of double spending – requires crosschecking against other transactions. If an attempt is made to spend the same Bitcoin in multiple transactions, only the first such spend transaction accepted into the Blockchain is considered valid. One approach attackers can follow in an attempt to double spend is to manipulate broadcasts in such a way that the fraudulent transaction is accepted by the network ahead of the legitimate one. Therefore, this possibility had to be monitored in our simulations of alternative Blockchain configurations (i.e., in those simulations involving fast payment subnetworks for premium services).

In the classic double-spend attack an attacker provides two transactions, Transaction A and Transaction B. Each transaction spends the same output from the Bitcoin address given to the merchant and this means that these transactions cannot both be valid when processed. Transaction B denotes the transaction that transfers the amount to the attacker whereas Transaction A is a transfer to the merchant. The attacker needs to convince the merchant (or rather the merchant's service provider) that Transaction A is valid while simultaneously broadcasting Transaction B into the network in such a manner that it is more likely that B will be the successful transaction. That is, a merchant who has received Transaction A should only know about Transaction A and should not see Transaction B until the goods or services have been provided to the customer irreversibly. To prevent the attack from being completed, for example, a customer at a store or vending machine would need to be held there sufficiently long to ensure that Transaction B is less likely to occur than Transaction A. Further, for a double-spending attempt to be successful, Transaction B needs to be confirmed by the Bitcoin network; that is, the miners would need to select Transaction B over Transaction A. In

any instance where over 50% of the miners supported Transaction B over Transaction A, regardless of whether this is through collusion or through random propagation to those miners, Transaction A will not become valid over time.

The ability to economically manage the double-spending issue within the Blockchain is a prime concern. One of the primary reasons for introducing an FPN is to ensure the merchant can trust the payment based on a probabilistic risk function for attack. This would enable a user to economically reduce the risk by using a game-based function. In the case in which a user has a known identity, and has been interacting with a merchant continuously over time, as opposed to engaging in a single interaction, the identity of the parties can be known between the two parties while remaining pseudonymous. In this way, we can have a layered effect where the risk of loss is known and limited. This would enable the FPN to ensure that payments are processed without the risk of double spending and where a double-spending attack does manage to occur that this can be economically accounted for. This approach would also enable competing networks to optimize their losses. Thus, a merchant would not necessarily seek to achieve zero losses if the cost of stopping or mitigating that loss exceeds the cost of the loss itself. Different payment processing networks will compete with the aim of minimizing this loss function. In a competitive environment, merchants who can reduce their costs through a combination of loss minimization against processing cost will become more competitive over time.

3. Method

3.1. Hardware and software configuration

The Bitcoin network models were built on successive iterations of virtualized high-performance computers that have become increasingly powerful since 2009. The initial system comprised 680 Windows XP machines that were subsequently updated to run CentOS Linux. These early machines, which were not clustered, acted as both mining and capture hosts. The first cluster was created in 2012 as an SGI-based supercomputer. Prior to this, a combination of machines individually ran *tcpdump* node software and analysis software. In 2011, it became apparent that a parallelized cluster would be needed. With the introduction of GPU, then FPGA, and subsequently ASIC-based systems, the demand to process transactions increased significantly. Although it was understood that the number of transactions would increase over time, the rate of this increase was significantly higher than was expected. Between January 2011 and mid-2012 many systems clustered in parallel were trialed. The devices were implemented across the various networks and allowed the high-speed data

that was written across a variety of online drives to be captured and analyzed. The amount of captured information grew significantly in this period, thereby leading to the need to run multiple machines. Even the use of GPU-based structures did not succeed in capturing all the data on individual machines and hence started running systems in parallel. The current HPC system in use, known as *Tulip Trading*, runs on over 265,000 Supermicro cores and is currently ranked at position 17 in the Top 500 Supercomputer list[2] (see *Appendix 1: Tulip Trading* HPC for full specification).

We combined several utilities and software packages, many of which are open source, to create the model of the Bitcoin system. The simulation software we used was the Common Open Research Emulator[3] (CORE), a real-time network emulator that allows the rapid instantiation of multiple technologies in a hybrid fashion. CORE was selected as it allows for the integration of both real hardware and virtualized nodes including router and switch configurations. The integrated multi-protocol network Emulator/Simulator (IMUNES)[4] from the University of Zagreb was patched into CORE to port the virtual stacks from FreeBSD[5] into scientific Linux. Each virtualized network stack is assigned its own process space. We have ported the FreeBSD Jar mechanism into Linux to form a lightweight virtual machine. These virtual machines are configured with SSH and the Bitcoin protocol and no other services. Our model includes support for wireless networks, mobility scripting, IPSec, distributed emulation over multiple machines, which is used across different machine nodes, and the control of external Linux routers that have been integrated within our system.

Using CORE, we emulated both wired and wireless settings and replicated the node distributions of the Bitcoin network including IP address and network positioning. The derived hardwired/wireless scenario composes both physical and emulated nodes including the full archive nodes of the Bitcoin Blockchain as well as limited nodes that are used for propagation only. The system is tuned to allow up to 50,000 full nodes at any time. At present, no scenario has occurred where 50,000 nodes have ever come online in the real Bitcoin network. The system has captured and can emulate the traffic flows between each of the nodes over time.

Our implementation of CORE uses operating system virtualization tools (Zen, UML, KDM and Open VZ, VMware, and Virtual PC) primarily to allow for the isolation of hosts or multi-server

[2] http://www.top500.org/
[3] http://www.nrl.navy.mil/itd/ncs/products/core
[4] http://imunes.net/
[5] https://www.freebsd.org/

environments. We used the Quagga[6] open source package as well as virtualized CISCO images to instantiate emulated routers in conjunction with physical routers across our network. The Quagga routing package was configured using *Open Shortest Path First* protocols OSPFv2 and OSPFv3. The CISCO IOS virtualized platform simulated Border Gateway Protocol (BGP) and Virtual Router Redundancy Protocol (VRRP) routing. Bandwidth performance and throughput measurements were recorded using the *Iperf*[7] network testing tool. This system can scale up to 1.2 million simultaneous virtualized implementations of Bitcoin nodes. At the same time, the system integrates the supporting infrastructure and simulates this over time. Thus, the system can simulate the existing network and capture live information in real time and can also be extrapolated to model alternative future configurations.

3.2. Node simulation

Our network simulates the mining and propagation processes natively (the mining process has been simplified to reduce complexity) enabling a limited number of machines to simulate the existing mining infrastructure (the node infrastructure has not been simplified). The types of nodes simulated are:

(i) Nodes that create/broadcast transactions;

(ii) Nodes that validate/propagate transactions; and

(iii) Miners.

In addition, there are a total of 10 nodes in each of five continents that act as collection and monitoring nodes to capture information around the Bitcoin network. In two centralized locations in the UK and the Americas an additional collection node infrastructure has been configured each with 50 nodes that capture data in those locations. Each of these nodes is tuned to have a mean collection of 50 nodes. It is configured to attempt to remember more systems if the number of connected nodes drops below 40 and to stop advertising for additional nodes when the number of connections exceeds 75. The systems update to share the IP address and node information such that there is limited overlap between the nodes collected. The Bitcoin clients on each of these nodes are configured with maximum connections set to 200 and simultaneously capture approximately 100 node links on average. We averaged this number of links with a distribution that varies between 75 and 200 links. The system would attempt to learn more links if the number dropped toward 75 and would stop

[6] http://www.nongnu.org/quagga/

[7] http://web.archive.org/web/20080907164521/http://sourceforge.net/projects/iperf/?

adding links if the number of connections exceeded 200 links. These are seeded using the -addNode command line config option for a few key collectors and then once each of them has collected several nodes, the nodes collected are blocked from the other collectors to allow a maximum distribution of known nodes in the collectors.

At present the node infrastructure has recorded up to 15,000 nodes running simultaneously. In the few instances where more than 15,000 nodes existed, a process was used to cycle through nodes dropping and rejecting connectivity using IP tables to block any connectivity between nodes for a period and then adding additional nodes. Given this capacity, the existing real Bitcoin network number count of between 5,000 and 7,000 nodes at any point in time is easily monitored. Each monitored node has its IP address, the version of Bitcoin protocol being run and packet information captured using *tcpdump*. This is recorded in a database and the header information is analyzed using the traffic fingerprinting tool P0F[8] to analyze the version of the operating system and supplementary data of the machine. The *traceroute* command is run daily between the collector nodes and each of the collected nodes; this enables the formation of a detailed router map providing connectivity details, redundancy details and resilience details for each of the nodes, and it also provides estimation of the bandwidth and the type of connectivity. For each of the hops recorded using *traceroute*, we analyzed the router type and modeled these in CORE. The links were emulated with known router link statistics and where available, real-time physical networks were integrated with the CORE solution.

The integration of the system allows for the application of link effects and bandwidth shaping delay, loss and duplicates. The recorded information allows us to create a link between routers, for example with a bandwidth up to 50 GB per second and 7 ns propagation delay with a bit error ratio of 1 in 100 billion or lower. The configuration can be changed to enable parameters to be adjusted instantly and each of the emulations can be changed to model the differences over time. The addition of a random jitter delay option has been incorporated to model future transactions and the growth of network bandwidth over diverse types of networks. We have modeled this over centralized, decentralized, and distributed international networks and have extrapolated network growth over the period 2015 to 2025 assuming no block size limit.

[8] http://lcamtuf.coredump.cx/p0f3/

Each simulation was measured against a combination of Bitcoin data captured from nodes running between 2009 and June 30, 2015. The growth models for this were analyzed against bootstrapped data as well as against financial system data from a variety of public sources. In testing the growth pattern, we first used the classical estimates for orders of growth over a complete function, which provides the measure of growth as:

$$M_f(R) = \ln \| f \|_R \tag{1}$$

Equation (1) follows a log growth pattern from a "Bayesian Averaging of Classical Estimates" based system (Doppelhofer et al., 2000). The data and growth patterns are expected to follow a continual nonlinear pattern and various analyses of the data demonstrated that the anticipated growth for Blockchain-based financial systems would be expected to comprise a meromorphic function rather than a classical growth model.

3.3. Data capture

The initial set-up was aimed at modeling the real Bitcoin network by capturing network behavior over time, including both mining systems and broadcast nodes over each version of the Bitcoin software and its interactions. This enabled us to model interactions against the number of nodes, the estimated bandwidth and packet capability of the monitored nodes, and the distribution of nodes. Our system has been scaled to allow a total receiving traffic more than 600,000 packets per second. This capability is provided using a combined infrastructure of 2,000 nodes each running Xeon Phi or NVidia GPU cards. The information is recorded and stored in 10-second intervals such that a complete model of the Bitcoin network over time was captured. This information enables a complete network replay of the processes occurring over the Bitcoin network including a high-level capture of Bitcoin client IP addresses that talk to the network. This enables us to map a sizable percentage of client wallets and their IP addresses as they communicate between nodes. The capture provided a database of over 30 million source IP addresses belonging to nodes and client wallets. The database also includes the routers and switches connecting these devices. The infrastructure also modeled many of the miner networks over time.

The system was analyzed in a manner that allowed us to establish the amount of network traffic we expected the Bitcoin network to be able to process. Various CORE scenarios including increasing the number of routers linked together, the number of hops, and the alteration of throughput were investigated. Using the recoded information, many scenarios were produced by considering

simulated mining environments. In addition, simulations were recorded by using a simulated simplified algorithm. This was achieved by altering the mining difficulty in each of the simulated networks such that the true difficulty in mining was reduced for the simulated network. In this way a network was created that could simulate the existing infrastructure and create solutions that would make it possible to sign packets and blocks without the current mining overhead which would not be viable to simulate.

4. Simulation runs

The models enabled many experiments using distinct categories of simulations. These include measurements of network capacity and behavior by varying factors such as the block size, number of nodes, node capacity, use of different compression algorithms, alternative data structures, exotic trade types (such as tokenized transactions), alternative crypto libraries, and specialized hardware. In this paper, we report on only three categories of experiments (with further results to be published in due course): Cryptographic Functions Benchmarking, Block Size Variations, and Fast Payment Networks. This section describes the simulations and the parameters that were used; the results are presented in section 5.

4.1. Cryptographic functions benchmarking

We executed benchmarking measurements on the processing and propagation speeds of transactions using alternative configurations of hardware clusters using E5 Intel cores, NVidia GPU-based CUDA systems, and Xeon Phi coprocessors. We chose these products to benchmark as they are the main types of architecture that exist across most systems today. Slight variances in different CPU and other hardware types were beyond the scope of our tests. The processors were tested for their performance on cryptographic functions such as ECDSA signature verification and hashing operations. We aim to present these data in a supplementary paper. It is important to note that the processing time used to validate the ECDSA signatures forms the primary limit on processing rather than the network propagation times or other processing aspects formed in validating a transaction for propagation.

4.2 Block size variations

We enabled the creation of large datasets that can be run against potential future scenarios with the Blockchain. We tested block sizes of up to 340 GB using both packet sizes and transaction sizes of 1 KB up to 20 MB.

Optimization techniques were applied to explore the optimal settings for block sizes and transaction sizes based on their effects on transmission latency and queue sizes. The simulation network configuration enabled us to run forecasting on 1,000 simultaneous simulations of the current Bitcoin network globally. These simulation runs tracked the distribution of nodes and the transport of gossip-network-based communications excluding the Bitcoin mining protocol.

4.3. Fast payment network

FPNs are subnetworks designed to preferentially propagate some transactions over others. For example, some merchants may be prepared to pay a premium on their transactions for such a service. Previous studies on FPNs (Karame et al., 2012; Bamert et al., 2013) showed that it is possible to secure transactions for small-scale rapid payment systems (such as a vending machine). This can be extended to high-value payments and even high-frequency trading systems. We hypothesized that the securing of an FPN could be conducted on a risk basis: smaller payments should be more likely to be sent quickly without the risk of double spend owing to the limited loss associated with each of these transactions. We used simulations of the network incorporating a variety of different nodes: a verification node, a transmission node, a node that stores the full Blockchain and a propagation node. The aim was to test simulations of a network that allows merchants to assure that their fast payments can be quickly integrated into a network that does not permit the double spending of their payment due to the faster propagation of a fraudulent transaction. This can be probabilistically calculated to assess whether it is successful based on the propagation to miners, which can be implemented selectively via nodes given a suitable distribution. Our model used the Floyd-Warshall algorithm for shortest-path calculations. In our simulations, we utilized data generated by a Visa-type network and modeled merchants' reaction to fees of between 0.1% to 3% of transaction costs versus no fees, and then considered the load a free network would be expected to carry versus the cost of that node. We compared the latency and propagation delays between the existing network and our proposed FPN of specialized nodes, as shown in section 5.3. The expectation is that a market-based rate would eventuate for transactions and the proposed system would allow for a fee market to develop.

5. Results and discussion

5.1. Cryptographic functions benchmarking

The four panels in Fig. 2 show sample simulation data for arbitrarily chosen scales. The data represent one configuration of the processors and are intended to demonstrate the relationship between the block size (in Mb), the CPU processing power (in TFlops), and the amount of ECDSA

processes to be processed (in number of processes per transaction). These graphs show that the relationships closely approximate linearity across the scales likely to be experienced in practice.

Different simulation settings were trialed for a single node with a combination of NVidia, Xeon Phi such that the smaller sections of code (smaller registers) are routed to NVidia, the large code registers are route to Xeon Phi, and the remaining codes balanced between the two. Of note is that the validation of the cryptographic functions including hashes and ECDSA signature verifications consume more time than the network transaction propagation latency. The simulation of tokenized[9] processes requires more processing than the validation of signatures for standard transactions: the introduction of non-standard transactions requires additional signature checks. A single node dedicated to ECDSA running with 20 E5 Intel cores[10] can scale to be able to handle a maximum of 36-Mb blocks at a total average processing speed of 345 TPS (Fig. 2(b)). Using either Xeon Phi or NVidia clusters we have been able to significantly increase this rate. We have noted that the Xeon Phi coprocessor cards handle the validation of ECDSA signature checking in a manner that is far superior to the NVidia GPU-based CUDA system. Moreover, NVidia or even the use of systems in conjunction with specialized ASIC systems is able to validate and process hash-based calculations much more efficiently than the Xeon Phi. Therefore, it is unlikely that we will see a node based on a single architecture being developed in future. Rather, we expect to see a clustered implementation of both Xeon Phi and NVidia machines in a clustered infrastructure. This is to achieve the best return of power consumed to the number of Flops processed, thereby suggesting that

[9] Tokenization here refers broadly to the usage of the Blockchain to carry non-standard transactions such as smart contracts and fiat currency representations.
[10] The rating in GFlops for CPUs and GPU based systems can be downloaded from vendor sites; e.g., http://download.intel.com/support/processors/xeon/sb/xeon_E5-2600.pdf

we need to move towards co-processor-based systems.

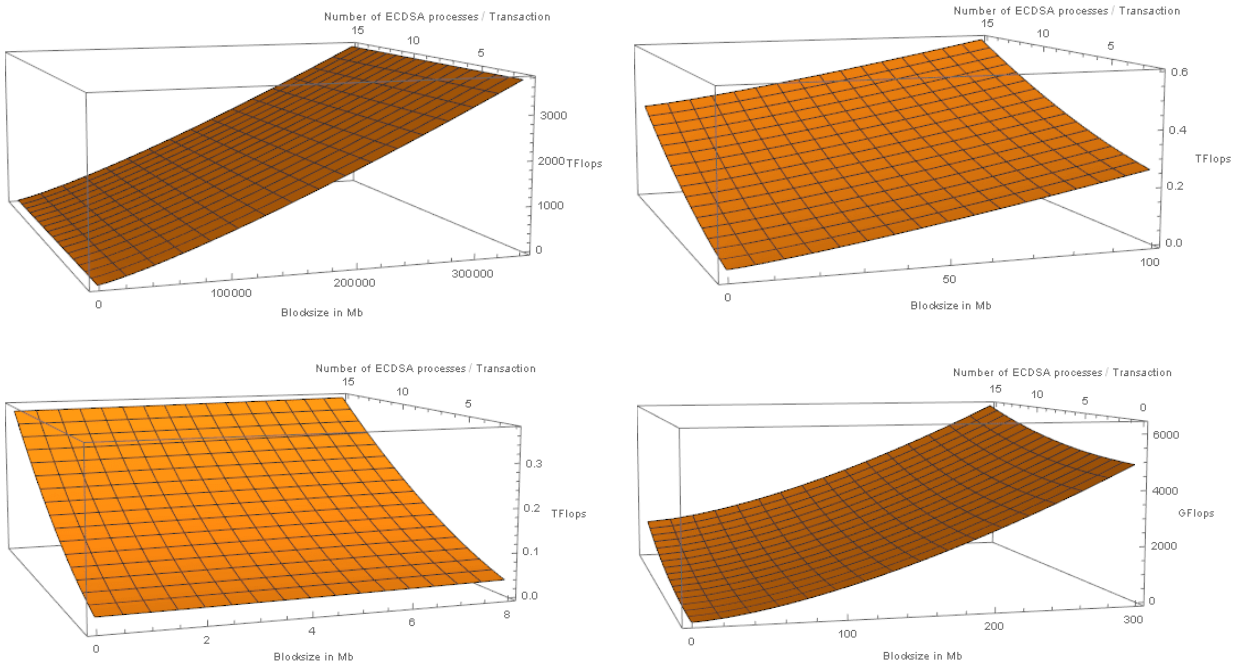

Fig. 2. Four different scales of sample data showing relationship between block size, CPU processing power, and number of ECDSA signatures to be processed

We see from the data presented in Fig. 2(a) that nodes scale with processing power and that as we increase the number of transactions being processed in the block, that the number of ECDSA operations that are processed within a transaction matters less. The simulations proved that it is possible to scale the Bitcoin protocol to support a block size exceeding 300 GB. This can be achieved by combining parallel processing and the validation of transactions in the software and the use of a high-powered system. A 3-petaFlop HPC system can process a transaction stream 300,000 times greater than the network currently supports. This is more than 1,000 times the processing capacity of the existing VISA network.

In the existing Bitcoin-QT system (Fig. 2(c)), the number of ECDSA processes within each transaction makes a significant difference to the amount of processing power utilized by the system. By parallelizing these processes, we significantly increase the ability of the system to scale (Fig. 2(b)).

In an optimized form (Fig. 2(d)) that has been developed to process and validate all transactions in under 500 ms, the use of CUDA-optimized code allows for a system running dual NVidia K80 GPU cards to process and validate an increased block size of over 300 Mb on a single hardware node.

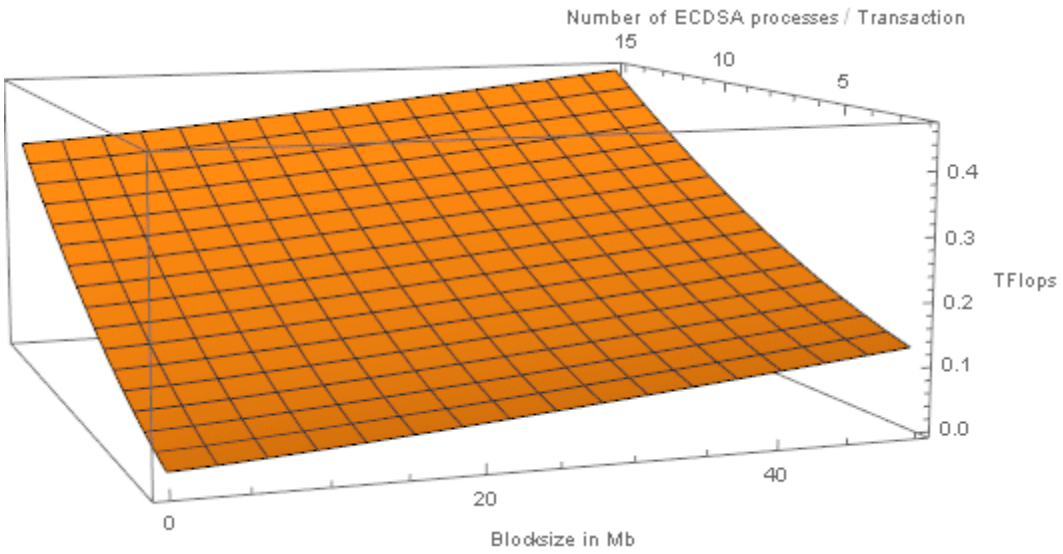

Fig. 3. Relationship between block size, CPU processing power, and number of ECDSA signatures to be processed on a single-node-based system

A single-node-based system using two current generation Intel Xeon CPUs can expect to return between 160 and 200 GFlops (LINPAC) processing power. Consequently, it is not difficult to determine that software optimized to run in parallel using a GPU (CUDA) and co-processor cards that can individually deliver up to 8 TFlops can scale. A cluster of the systems used for Fig. 2(d) can scale to allow unlimited block size.

5.2. Block size increases and storage

The resulting storage problem will be addressed in a complimentary paper detailing a form of validated storage node. By assigning special tasks to individual functions of the Bitcoin nodes, we can allow for the creation of a market-based solution to the issue of transaction costs and spam. The maximum storage required for the Blockchain can be calculated and is a linear relationship over time. Fig. 4 displays the required storage for a 10-year period. Here we see that even using a 1,000 MB block size maximum limit, the maximum limit for the storage required is calculated to be 120 TB in 2025.

The rate at which the size of hard drives has been increasing has remained steady and is predicted to grow at a similar rate for at least the next 10 years[11]. The primary misunderstanding of the growth capabilities in the Bitcoin Blockchain does not originate from scientific analysis, but from the failure to understand exponential growth. The Blockchain is a system that, when limited, can

[11] https://technology.ihs.com/406733/storage-space-market-brief-issue-12-2012

grow at a maximum linear rate. Hard drives and systems processing grow exponentially. The consequence is that systems that are considered to be large now quickly become the norm.

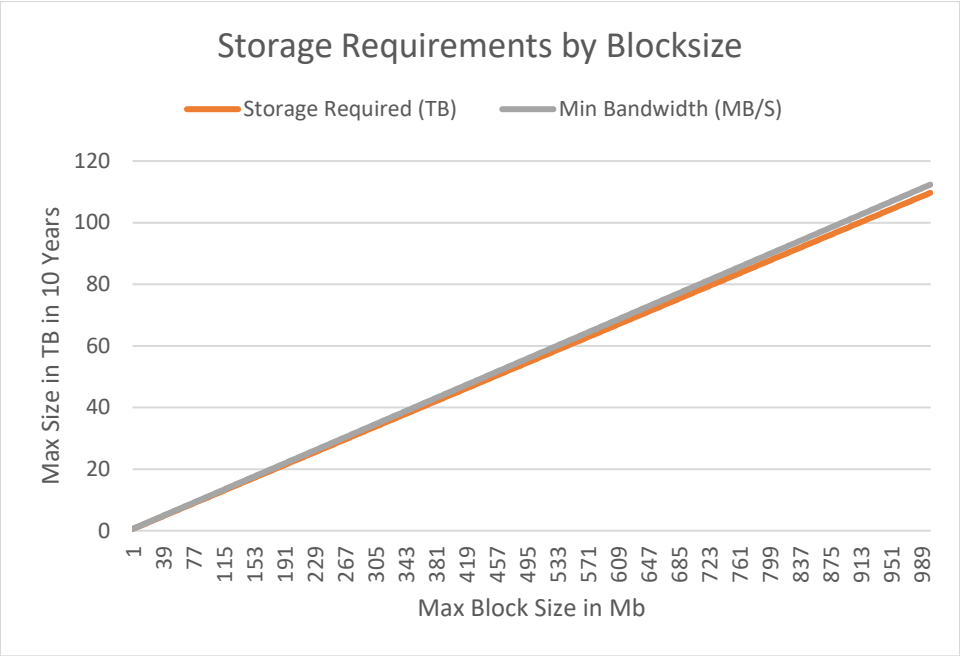

Fig. 4. Data storage size as a function of block size

The annual storage requirements for a block size of up to 100 Mb (Fig. 5) are capped at 1.4 TB. This is well within the limits a modern computer system is able to contain.

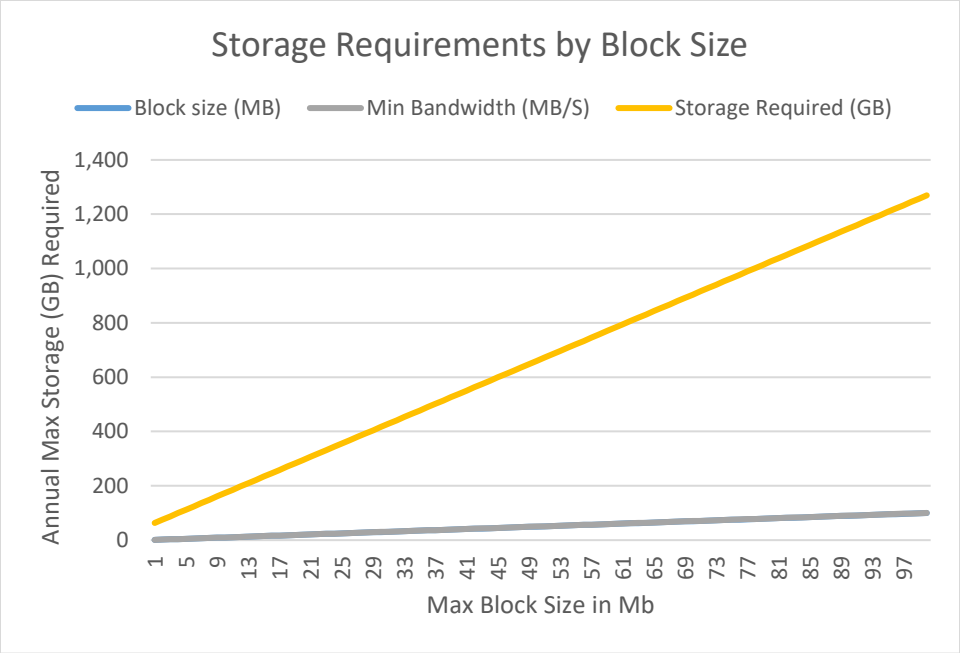

Fig. 5. Data storage size as a function of block size

Given that systems growth is exponential (Fig. 6) and the Bitcoin system is limited to a linear growth rate, it is simple to see that the system can grow at a far higher rate than has been proposed.

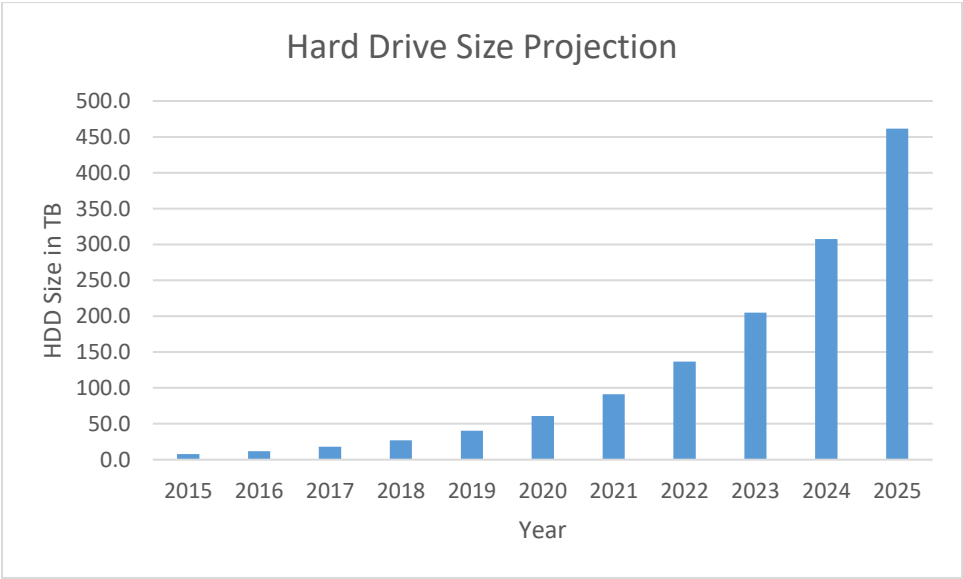

Fig. 6. Data storage size as a function of block size

The result is that drive space is not a limiting factor for increasing the size of the Blockchain. Even with a sustained block size of 5,000 Mb, the total storage capacity of the Blockchain would not exceed the predicted storage capacity of a single server. With the introduction of a dedicated storage

node, the growth of the system can be extended to a 300-GB block size and beyond. This eventuality was foreseen by Nakamoto (2008), as mentioned in section 7 of his paper. Here, most nodes would operate with a limited subset of the Blockchain. In this scenario, specialized nodes that act as a store of the complete Blockchain can be referenced with only the Merkle hash necessary for the payment nodes.

5.3 Fast payment network

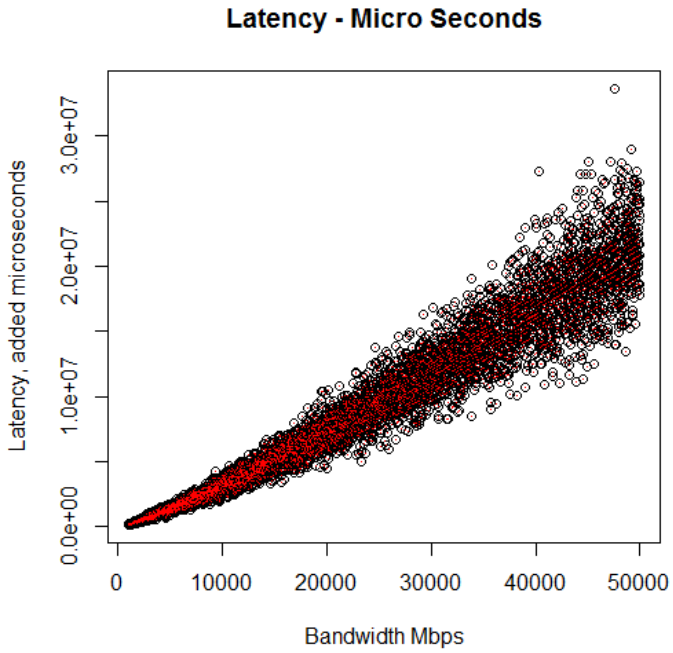

Fig. 7. Latency against bandwidth, existing network

Fig. 7 shows the latency measured against the bandwidth for the existing network. Fig. 8 shows a plot of the latency against the continuous bandwidth needed for increased transaction sizes. In this plot, we have also incorporated the error range for each of the measurements. Fig. 9 plots the expected propagation delay in TPS against the existing network. This system has been modeled utilizing standard commercial hardware. The system used is Intel i7 with 32 GB of RAM and an SSD consisting of 100 IOPs. Fig. 10 shows the propagation delays on the simulations using specialized nodes. We can clearly see that dividing the node functionality increases the effectiveness of the network significantly.

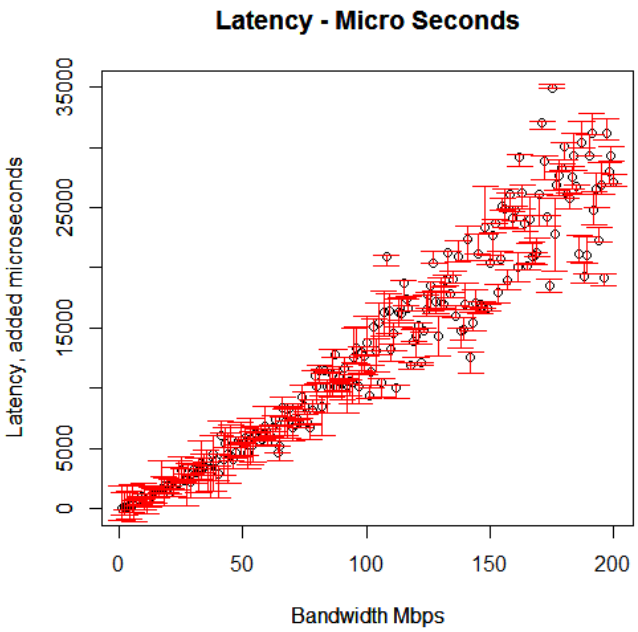

Fig. 8. Latency against bandwidth for the proposed network of specialized nodes

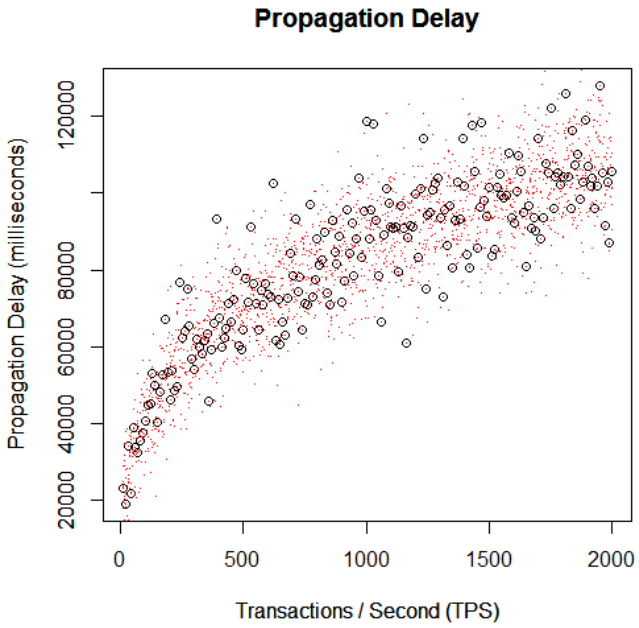

Fig. 9. Propagation delay as a function of transactions per second (TPS)for the existing network

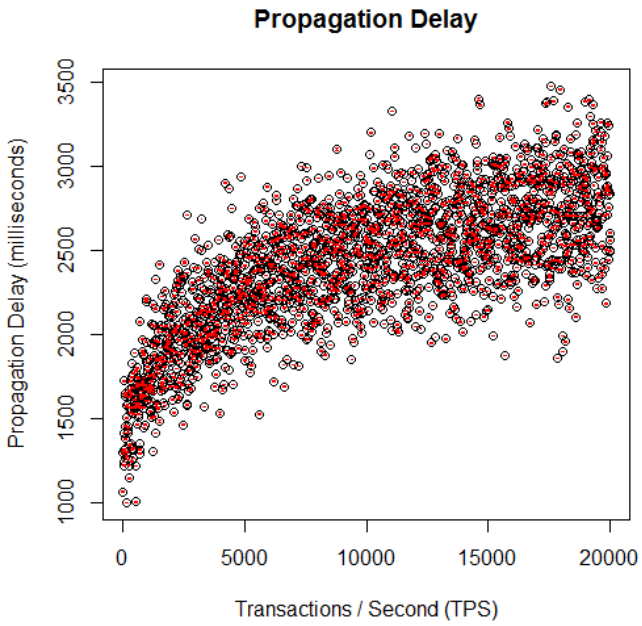

Fig. 10. Propagation delay as a function of transactions per second (TPS) for the proposed network of specialized nodes

For an attack to be successful an attacker would have to connect to several nodes in a network and broadcast the double-spend transaction claiming the outputs quickly such that the miner is more likely to receive this transaction ahead of the legitimate one and process it. The attacker cannot inhibit the communication between nodes and will not be able to identify the neighbors of adjoining nodes. The attacker would not be able to utilize the paid merchant transmission nodes and if an attacker was to succeed their information could be recorded, in which case such a network would be able to instantiate an assurance protocol. This means that anyone propagating over the network would have to present an assurance contract of Bitcoin such that any double spend or attack would violate the conditions allowing for the payment of an assurance contract exceeding the covering of a double spend or another such attack.

In our simulations of FPNs, we included nodes that incorporate a monitoring function for double-spend attacks that would alert individual merchants to attacks of this nature. We showed that such a network could interact within milliseconds allowing a propagation that has occurred across the network to be broadcast faster than the gossip network broadcast between miners and other instantiated instances of the Bitcoin network. This distributed autonomous corporation (DAC) would then be able to alert merchants of a double spend that had occurred anywhere on the Bitcoin network and allow the merchant to reject the transaction. This rejection could happen within seconds, and we

show that the creation of a suitable merchant network for nodes would enable a fast spend network that could detect double spending and reject this in a period of less than 5 s. In such a system, we believe that it would be possible to reject small transaction double spends quickly enough that even where a vending machine system has been deployed, goods could be re-routed rather than going to a customer system if a double spend had occurred. This leads to a truly extensive payment system that could be used globally.

Even with pseudonymous accounts it is possible to utilize the identification of master accounts, thereby allowing merchants to incorporate knowledge of clients that would allow them to verify information over time and quickly transfer value to trusted individuals. Even without this trust, the integration of propagating nodes would allow clients and merchants to quickly propagate information to miners in a manner that would allow payment with a low probability of double spend. This probability could be insured and assured with successful node integration by providing a network of payment authorities that could be integrated into a DAC.

In this system, a user selecting an item would, after payment, wait for a small amount of time such as 4-5 s, and, if no double spend is detected in that period, the goods would be delivered. In this manner, a merchant accepting fast payment can act without incurring the risk that a transaction will not be confirmed by the network and can reduce the risk of an economic loss. In case an economic loss does occur this can be incorporated into the price of a sale over time and the loss will in any event be minimized with information about the user being recorded.

An FPN should be secured on a risk basis: the likelihood of smaller payments being sent quickly without the risk of double spend should be higher due to the limited loss that can be associated with each of these transactions. We showed that a variety of different nodes (verification node, a transmission node, a node that stores the full Blockchain, and a propagation node) can be integrated to extend the current node system without modification of the Bitcoin protocol. Merchants selectively using nodes could use the core Bitcoin protocol with the addNode function. A future implementation of this system would enable merchants to pay for preferred nodes that could propagate information around the network with a guaranteed propagation time.

In this system, information eclipsing is minimized where the merchant forwards selectively to neighboring nodes that will accept their transaction rather than any attack transaction, i.e., when a node has received a double spend from a trusted merchant it will drop their information and alert the merchant to the scenario. In general information eclipsing, when the merchant forwards Transaction

A to its validated nodes these will propagate between the network quickly over a fast back channel which will then propagate through the remainder of the Bitcoin network (Decker and Wattenhofer, 2013). This implementation of a modified SIS epidemic network infrastructure (Pastor-Satorras and Vespignani, 2001) allows a merchant to quickly forward their transaction as it is received at many nodes. In this instantiation nodes can be autonomously verified and checked to ensure that any information sent to them will be expected to respond quickly and securely around the network. Random checks by third parties should be conducted on any such network to ensure the honesty of any DAC in this system. This would be quickly and easily verified and it would be expected that any DAC participating in such a system would lose merchants and clients extremely quickly if it did not propagate in the manner advertised. In this scenario, when the merchant transmits Transaction A to the connected nodes these connect across to the neighboring nodes in such a manner that they can quickly propagate a transaction around the entire Bitcoin network in a maximum of 3 and generally 2 hops. This would be one hop into the merchant connected node that could be conducted in less than 3 ms in many instances and the merchant could pay for a service that provides a connectivity of less than one millisecond, the connection between verified nodes would enable the transmission of even large packets to occur within less than 5 ms to the primary wired nodes. These wired nodes could propagate up to 10,000 nodes within 7-8 ms and in an ideal fast network scenario this could be conducted globally in fewer than 4 ms in urban centers. In this scenario, the attacker sending Transaction B would be expected to require their transaction to be sent to many miners faster than the merchant network. Here, the verified node would monitor for the merchant any conflicting transactions such as Transaction B and where any transaction such as Transaction B occurred they would know of the double-spending attempt. The modification does not require any change to the existing Bitcoin protocol and can be implemented by nodes with slight protocol extensions without any other node being aware that these are merchant nodes.

Using this system, the merchant nodes could implement a distributed update system in which they could capture at least 90-95% of nodes connected to the network at any time and could potentially monitor 100% of nodes connected to the Bitcoin network. In doing so, any attacker sending an invalid transaction would reach the merchant within 2 hops, any transaction received by a node would propagate into the merchant network in 1 hop, and the merchant network would instantaneously respond by transmitting a message to the merchant. In this scenario, the merchant would not accept any transactions and would invalidate Transaction A, thereby stopping the transmission of the goods and record information about the transaction. Transaction B would be

broadcast to the merchant network, the merchant network would then record information about the double spend ensuring that the information on each of the double-spend attacks was recorded and monitored over time and that information about the double spend could be used to build up a profile of any such attackers. Further, if no nodes have been selectively forwarding double-spend attacks or known IPs or other such source details have been recorded, it would be possible for information to be collected and stored about known attackers that could be used to profile them over time by rejecting any further information. Merchant nodes could instantly block any known attacker by updating information such as IP table filters to reject IP addresses or other identifying information in such a manner that known attackers would be unlikely to propagate information across the network.

The ability to randomize merchant connectivity in a manner similar to that over our CORE simulations means that it is possible to quickly simulate alternative network configurations using a virtualized node system that would thwart any attempt by an attacker to block such information. The periodic randomization of the merchant network in terms of reconstructing information, IP addresses, and node connectivity while updating this automatically between the nodes would make it computationally difficult for the attacker to propagate any information and would minimize the risk of such an attack.

The ability to randomize connectivity and map over 90% and sometimes 100% of the node network allows the merchant network to reduce timing attacks and monitor for attacks efficiently. The merchant is thus able to examine a propagation depth that ensures up to a desired level of risk is mitigated. For instance, in our propagation testing we set a Six Sigma level of risk mitigation meaning that in either 2 or 3 node hops we are able to distribute sufficient data to the Bitcoin network to ensure that any transactions received by the merchant have either not been alerted to the merchant within seconds or that the instance of propagation would require more network nodes than currently exists for the attacker to be successful. In a scenario such as this the Six Sigma level of risk could be taken by the merchant meaning that fewer than one in a million attacks would succeed and the merchant could minimize their risk by maintaining the level of loss against that propagation rate. It would be infeasible for the attacker to maintain knowledge of a suitable level of merchant connections to block these and yet propagate across a suitable number of other nodes in the Bitcoin network. Here only a small minority of nodes would receive and broadcast Transaction B into the mining network. Given this scenario, Transaction B would be received only by a small number of miners who would be less likely to put this into a block. This scenario could be further extended if

mining entities were paid to reject known double spends and an assurance level was paid to protect the client from any adverse inference.

The implementation of an accounting system that monitors for any double spends would allow the merchant to ensure that where double spends have occurred the customer is not able to validly also seek a refund or return payment. The primary aspect of this system would be chaining the payment and associated goods together. In this way, the exchange of a tokenized item or quantity on the Blockchain network would be chained to a single transaction involving the payment from the client. In cases where sizable items or individually recorded items or even digital goods are sold, the transaction involving the transfer of the rights to the goods would be tied to the payment to the merchant, in which case a double-spend attack would lead to the rejection of the transfer of the item.

A simple system would involve separating payment and provision of the goods or reversing payment of the goods. Physical delivery of the goods could be quickly separated or the rights to access digital goods would be unavailable. Where physical goods are tied to a real-world implementation such as a digital signature that unlocks a car, access to the car would not be available even on a hire contract when the transaction paying for the contract had not validly been submitted and processed over the Blockchain network.

In creating such a system, we have taken the information collected over our CORE[12] network on Tulip and extended this to simulate the injection of double spends by attackers and have simulated the creation of a merchant node network. Each merchant network would compete to gain favor with merchants selectively sending to known trusted merchant networks, and this would be conducted using information such as the IP address, Alternatively, it could be conducted using a fast flux network implemented from tax scenarios that have been noticed over the malware industry in a manner that would enable the IP address and other information to be hidden but still propagated to the merchant.

The node network with the addition of a fast flux network could cycle through IP addresses in a manner not allowing merchants to determine the known addresses of merchants and simultaneously propagating across the Bitcoin network in a manner that does not cause instability to the gossip protocol. Merchants would be able to verify the availability and rate of distribution from the

[12] The Common Open Research Emulator (CORE) was used to model and simulate the existing Bitcoin network and to extrapolate growth over distributed nodes (http://www.nrl.navy.mil/itd/ncs/products/core).

merchant node network and could selectively verify the existence of the complete Blockchain through archive nodes. Merchants could pay for an enhanced service allowing both the use of merchant nodes for free commerce at a limited rate or for complete merchant propagation at a high rate for a pre-determined fee. Contracts could be built into the Bitcoin protocol allowing the merchant to both pay and check using a smart contract that pays for the utilization of the network ensuring the security of the node network and the propagation for merchants. Such a network would ensure that nodes were maintained and available and the localization of selective node networks could be integrated into collections of global node networks allowing for both the propagation of distributed node organizations and for the reporting and availability of all this information.

Each node in the merchant node system would need to have sufficient bandwidth to receive connections from random inputs as well as connections from the paid merchant system. Although merchant payment transactions could be promoted to the network faster, the node would be required to also receive transactions randomly through the rest of the network. The node could propagate its transaction policy such that any other wallet on the network would be able to selectively add or reject merchants, but the merchant system would be able to supply bandwidth at a rate to the merchant ensuring a set level and standard of service. In this manner, as the double-spend attack occurs, the two conflicting transactions transmitted into the network would be quickly detected. Large packets could be propagated through the network within seconds and detection of any double-spend transactions would happen at the same time.

The forward belief network was calculated using a Bayesian probability matrix for which we used the previous values and, given the movement over time with various scenarios, extrapolated using a back-propagation network and deep learning algorithms to reproduce the likely scenario given each of the proposed sizes of blocks etc. This required us to model the network sizes for each of the nodes and extrapolate the loss of nodes and the change in node types that would be necessary for a sustainable network. Over time this sustainable network would change with the commercialization and professionalization of node management.

The current scenario involves many amateur networks, which incorporate transient nodes that join and leave the network. These networks are less secure and flapping or transient responses can occur. Each of the simulations was conducted by utilizing a reconstructed network based on the responses of the remaining nodes and an extrapolated response by an increased number of commercial nodes.

6. Removing the cap

Finney and Dillinger lobbied Satoshi Nakamoto heavily in 2009 to implement a block cap[13]. The reason for this was that Satoshi's vision of an economic market was heavily reliant on the value of transactions being able to restrict an attack. In 2010 when the cap was implemented, it would have been possible to flood the network on a sustained basis for under USD 1 a day. However, as the value of Bitcoin increased by over 15,000 times[14] this attack was no longer viable. The addition of tiered fees, as was originally envisioned by Satoshi, was incorporated into his code to mitigate flooding attacks. These fees would quickly render any flooding attack prohibitive and at the same time fund the miner into being able to economically profit from any attack.

In 2010, Satoshi determined that the block size would not be a limiting factor and that the cap was temporary. Shortly before he stopped actively developing code and direct interaction with the Bitcoin community, he implemented the change shown in Fig. 11[15].

```
if (blocknumber > 115000)
   maxblocksize = largerlimit
```

Fig. 11. Satoshi Bitcoin Talk post, October 4, 2010

Satoshi's original plan[16] was to implement variable rate limiting to make it more expensive to attack Bitcoin whilst not impeding economic activity and allowing the market to determine the level of transactions that will be accepted. This change rewards all miners for accepting larger blocks and compensates them for the additional resources they require to mine larger blocks.

In alignment with Satoshi's vision, the proposal is to implement a scaling process of:

- 8 MB
- 32 MB
- No fixed limit with market-based fee and economic fee limits

In the seven years since the introduction of the cap, Moore's law has held and the processing power of computers has increased more than 25 times. By the end of the decade, the computing power will have increased by over 100 times relative to that of the systems that existed when the cap was introduced. During this time, the cap has not changed and has become a severe limitation that is

---

[13] https://bitcointalk.org/index.php?topic=946236.msg10388435#msg10388435
[14] https://en.wikipedia.org/wiki/History_of_bitcoin
In July 2010, a Bitcoin sold for USD 0.08. The recent price is USD 1,200.
[15] http://satoshi.nakamotoinstitute.org/posts/bitcointalk/485/
[16] https://github.com/trottier/original-bitcoin/blob/master/src/main.cpp

harshly restricting the growth of Bitcoin as a practical payment system. Clearly, the move from small home user systems to professional mining farms[17] demonstrates the ability to safely move to larger block sizes as the fee system is developed and rigorously tested to enable the complete removal of any limits on the block size.

This process will involve implementing the code in two stages. The first stage will see an increase in the block size that reflects the increase in computer power, after which a fee- and market-based system is planned, as seen in Fig. 12:

```
if (blocknumber > 486000)              // Around July 2017
   maxblocksize = 8000000              // 8MB

if (blocknumber > 538000)              // Around Jan 2018
   maxblocksize = 32000000             // 32MB

if (blocknumber > 642500)              // Around Jan 2019
   maxblocksize = ((blocknumber – 642500) / 125) + 48000000

if (blocknumber > 950000)              // Around Jan 2025
   maxblocksize = 1000000000000        // effectively uncapped and fee based

// Block size is configured to grow faster than Moore’s law and

// is controlled at this point forward with a market system of fees.

// The initial size is set and grows with time and is rate limited

// allowing Bitcoin to grow to a point where it can compete with Visa and PayPal
```

Fig. 12. Planned staged increase of Bitcoin block size

A dynamic system that is unrestrained and which implements a change responsibly and predictably followed by a fee-based system and the removal of all limits would enable the market to adjust to the changes and allow Bitcoin to become the system it should be. As the size of a block is determined by fees, it would become uneconomical for an attacker to mount a sustained attack on Bitcoin, because any attempt at a sustained flooding attack would fund the systems needed to stop the attack.

[17] http://satoshi.nakamotoinstitute.org/emails/cryptography/2/

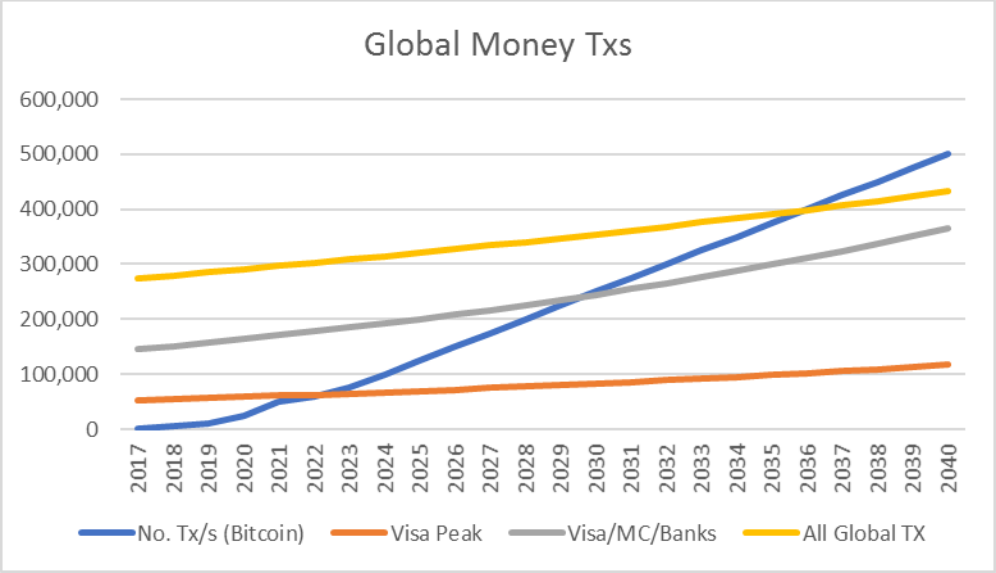

Fig. 13. Transaction propagation in transactions/second with a proposed scaling network of specialized nodes (Tx denotes transactions)

These changes would be implemented using the initial code shown in Fig. 14, and which is taken directly from Satoshi's original code implementation. The implementation of a fee-based market structure will be created by allowing both community and miner interaction to ensure that the implementation is safe, tested, and is aligned to the market.

```
// Transaction fee requirements, mainly only needed for flood control
// Under 10K (about 80 inputs) is free for first 1,000 transactions
// Base rate is 0.00001 per KB


int64 nMinFee = tx.GetMinFee(pblock->vtx.size() < 1000);
```

Fig. 14. Code for implementing the change in the block size based upon Satoshi's original code implementation

The value in Fig. 14 is based on the original reference implementation by Satoshi. The change is effectively a transition from 100 to 1,000 free transactions in a block to encourage growth. The minimum fee would be capped at USD 0.001 (one tenth of a cent) and this level can be referenced to the monthly average value of Bitcoin based on a three-month rolling average.

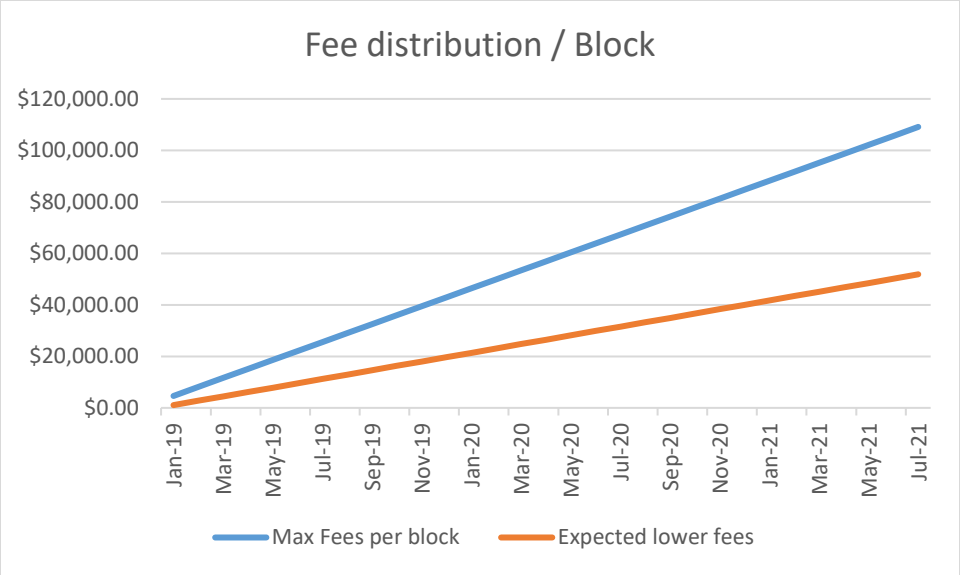

Fig. 15. Fee structure based on uncapped volume.

The fee-based rate controls will be implemented by adding two new variables derived from the original code:

```
int64 nStdFee      for the next 10,000 Transactions
int64 nMaxFee      for all other Transactions in a block
```

nStdFee will encompass the majority of transactions at a rate of USD 0.005 (one half of a cent).

nMaxFee is the flood limiter. Users who are willing to pay in times of congestion would be able to send transactions quickly at a rate of USD 0.05 (five cents).

The fees are summarized in Table 1.

Table 1

Fee structure and example breakdown

Fee per transaction (USD)	Transaction #	Probability to confirm in 10 min	Average transaction size (bytes)	Additional fee (satoshis / byte)
none	0 - 1,000	10%	2,000	n/a
0.001	1,001 – 10,000	40%	225	10.00
0.005	10,001 – 50,000	70%	225	50.00
0.050	50,000+	99%	225	500.00

Currently, the fees are flexible and decided by the market. The change will be implemented by January 2018 to allow the miners time to change to the new code and to prepare for the introduction of a variable block size. Miners can stop accepting transactions at any time knowing that other miners will accept these and incorporate a growing fee base that will be lost to a miner not accepting larger block sizes when the transactions are at a level the miner chooses not to process. Over time, miners will become increasingly incentivized to incorporate as many transactions as they can.

The arguments against nodes have misconstrued the nature of a node. Rather than being a full implementation of the Bitcoin software, a node consists of software with mining enabled.

Nodes have been defined in the original readme.txt[18] file distributed with the reference client. This document states:

> To support the network by running a node, select:
>
> Options->Generate Coins
>
> and keep the program open or minimized. It runs at idle priority when no other programs are using the CPU. Your computer will be solving a very difficult computational problem that is used to lock in blocks of transactions. The time to generate a block varies each time, but may take days or months, depending on the speed of your computer and the competition on the network. It's not a computation that has to start over from the beginning if you stop and restart it. A solution might be found at any given moment it's running. As a reward for supporting the network, you receive coins when you successfully generate a block.

The implication is that the reference client is not a node unless it is also mining blocks.

[18] https://github.com/trottier/original-bitcoin/blob/master/readme.txt

7. Concluding remarks and recommendations

Our analysis of the Fast Payment Network entailed simulating a variety of scenarios based on existing node distribution models. The growth of the network takes away certain possible scenarios such as the widespread distribution of home nodes. The primary reason for such a variation has its origins in the need to process many transactions. External transactions alone account for over 5,000 TPS across the major card processing companies. When we incorporate the full range of electronic transactions we obtain a figure of the order of 100,000 TPS for the current state of the world financial system. The rate of 5,000 TPS at the level of the existing card processing companies provides for a level of transactions that equates to a block size of 240 MB. This is the minimum block size Bitcoin would need to achieve to be competitive. This assumes the use of the Blockchain for only electronic commercial usage.

A complete change of the financial system would require the ability to scale to over 100,000 TPS and a block size of the order of 10 to 12 GB for each 10-minute period. We can increase the block size at Moore's Law for a single host node and this could be achieved slightly faster by creating specialized systems, which would enable us to revert to following exponential Moore's Law. Traffic that incorporates the Internet of Things and the proposed growth rate of what could become trillions of devices needs to scale further to be effective. This would require the incorporation of two aspects. First, we would need to investigate the time taken to distribute block sizes as large as those proposed above. Second, we would need to consider the time taken to process the transactions within this block size. A miner can reduce the necessity of validating all information they receive prior to starting to analyze and solve the block puzzle if they are able to rely on the information they have received. A market for information can be provided where a miner can reduce their infrastructure cost by specializing in the solution of the hash puzzle and allowing for a separate infrastructure dedicated to the validation of information contained within the block. This distribution function leads to a scenario where the miner specializes in the solution of a hash problem associated with previously validated information. The merchant node not only assumes the role of validating the system to ensure that the signatures are correct and that the hash function is effective but also provides it to the miner in a timelier manner. In time, different node functions are expected to become more specialized. Rather than maintaining all aspects of the Bitcoin Blockchain, a processing node would be able to utilize the benefits of the Merkle tree structure to hold only a prior selection of data. Specialized nodes that hold complete copies of the Bitcoin Blockchain would be able to fulfill the supporting roles that could be offset from both mining and processing nodes. This distribution

function within the Bitcoin Blockchain would enable the infrastructure to scale far more than has been envisioned. In doing this, we would also provide a means of growing the distribution of node entities.

The current system has changed such that a small number of mining conglomerates control the majority of hash power within the network. This has led to a distributed non-aligned cartel-like system with much of the power held within a limited group and outside the control of most user nodes. The belief that 6,000 nodes presently control the Bitcoin network is a fallacy based on a lack of understanding of how the voting system operates. The system we are proposing allows far more competition and overcomes many of the flaws introduced by centralized mining organizations that rely on a collection of free nodes.

Our research demonstrates the feasibility of a propagation system that can take generalized connectivity between any system incorporating both free nodes and paid nodes allowing for the propagation of anything from single Satoshi transactions up to including large transactions that are guaranteed across the network. What we have shown is the need not so much of increasing the number of nodes but the amount of processing power, where we take the data going up to 10,000 individual nodes to where we expect 1000-1200 enhanced nodes in the future. The implementation of a DAC in the system would allow users to invest in the network and would allow merchants who pay this to fund the creation of autonomous systems that propagate information and act in a provably secure manner. Each of these nodes would be verifiable independently outside of the network.

This study, in which we modeled double spend attacks and network scalability, was primarily concerned with the implementation of paid specialized nodes can enable the fast dissemination of data to render double spending unviable. This would mean that, even in large networks, merchants would be able to detect from their connected nodes that an attempted double spend had occurred within seconds enabling a merchant to cancel a transaction and stop the transfer of goods to a client. The ability to link transactions consisting of multiple transactions for the transfer of goods and payment simultaneously means that there are more effective solutions for the integrated provision of digital rights and other exchanges that cannot be conducted using a double spend. For instance, if a transaction for the sale of property was linked to the payment of goods in a single transaction, a double spend would violate this principal and violate the transfer of the goods itself. Therefore, we did not concentrate on this and the solution to these issues will be provided in a subsequent paper.

As competing networks seek to optimize their losses, a merchant will not necessarily seek to achieve zero losses if the cost of stopping or mitigating that loss was to exceed the cost of the loss itself. This ability of the proposed network to encourage competitive growth is expected to allow it to become a global payment system.

The basis of the Bitcoin network is derived through epidemic propagation. Rather than proposing to change this, we seek to introduce an additional layer into the network that improves the efficiency of the propagation of transactions in a manner that allows neighboring systems to interact more rapidly. Competing node networks would still have to broadcast transactions that are sent to them across the remainder of the network. Where there is no trust between these entities a gossip network remains the preferred solution. However, the efficiency of the network will also be maintained and increased not only through competing nodes but through an amalgamation of both competing and cooperating nodes.

References

Bamert, T., Decker, C., Elsen, L., Welten, S., Wattenhofer, R., 2013. Have a snack, pay with bitcoin. IEEE International Conference on Peer-to-Peer Computing (P2P), Trento, Italy.

Cormen, T., Leiserson, C., Rivest, R., Stein, C., 2001. Introduction to Algorithms, second ed. MIT Press, Cambridge, Massachusetts and London, UK.

Decker, C., Wattenhofer, R., 2013. Information Propagation in the Bitcoin Network. 13th IEEE International Conference on Peer-to-Peer Computing.

Doppelhofer, G., Miller, R.I., Sala-i-Martin, X., 2000. Determinants of long-term growth: a Bayesian averaging of classical estimates (BACE) approach. Economics department working paper no. 266, OECD, ECO/WKP(2000)39.

http://www.tik.ee.ethz.ch/file/49318d3f56c1d525aabf7fda78b23fc0/P2P2013_041.pdf (accessed)

Leitão, J., Pereira, J., Rodrigues, L., 2007. HyParView: A Membership Protocol for Reliable Gossip-Based Broadcast. 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks.

Karame, G., Androulakis, E., Calkin, S., 2012. Two bitcoins at the price of one? Double-spending attacks on fast payments in bitcoin, Cryptology ePrint Archive, Report 2012/248. http://eprint.iacr.org/ (accessed …)

Nakamoto, S., 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf (accessed….)

Pastor-Satorras, R., Vespignani, A., 2001. Epidemic Spreading in Scale-Free Networks. Phys. Rev. Lett. 86, 3200.

Appendix 1: Tulip Trading HPC

The following specifications are current as of November 2015:

C01N - SGI ICE X/SUPERBLADE SBI-7127RG-E, INTEL XEON E5-2695V2 12C 2.4GHZ, INFINIBAND FDR, NDIVA M2090/INTEL XEON PHI 7120P

Site:	Tulip Trading
Manufacturer:	Supermicro/SGI
Cores:	265,440
Linpack Performance (Rmax)	3,521 TFlop/s
Theoretical Peak (Rpeak)	4,470.41 TFlop/s
Nmax	1,847,808
Power:	4,499.87 kW
Memory:	251,904 GB
Processor:	Intel Xeon E5-2695v2 12C 2.4GHz
Interconnect:	Infiniband FDR
Operating System:	CentOS