Cost per Thousand Transactions (CPT)

Can decentralized networks enable *data-rich* apps? Why *Cost Per Thousand Transactions* is the most important metric we should be measuring

Chris McCoy
Storecoin.com
San Francisco, CA
chris@storecoin.com

Rag Bhagavatha
Storecoin.com
San Francisco, CA
rag@storecoin.com

Jason Varner
Storecoin.com
San Francisco, CA
jason@storecoin.com

Abstract

In this paper, we introduce *Cost Per Thousand transactions* (CPT), a unit, which can be used to compare the cost of processing 1,000 transactions on various blockchain-based protocols. We compute CPT from the point of view of senders (developers/users) and miners because a) senders bear the cost of submitting transactions in the form of transaction fees and b) miners bear the cost in the form of infrastructure they need to run to process transactions. **We argue that CPT should be a primary variable in any network's economic model as it influences both the cost to end users and profitability of miners.** We conclude the following in this paper.

1. **Miner profitability is critical to the long term sustainability of any blockchain network.** The profitability cannot be described purely in the form of miner's stake and the potential block reward to calculate the return on investment (ROI). Cost to run the infrastructure matters and CPT is one such measure that can help with the accurate calculation of the ROI.
2. Network usage is affected by transaction fees. **Low or no transaction fees attract users and developers.** CPT for end users helps with estimating this cost, so developers, for example, **can determine if it is profitable to build their apps on a particular platform**.
3. All blockchain networks assess a "decentralization premium" — the cost overhead due to *decentralization*, which is a systematic approach to replace *trust*. The cost overhead must not be too high compared to centralized cloud services because **if the decentralization premium of a network is too high, it may not be economically viable to develop apps on that network**.

In this paper, we measure the CPT for some of the well known Proof-of-Work (PoW) and Proof-of-Stake (PoS)-based blockchain protocols, so their respective costs can be compared side by side.

We have specifically excluded analyzing layer-2 scaling solutions in this paper because of the permutations possible with layer-1. We examine CPT from layer-1's perspective.

Key takeaways

We show in this paper that networks like Ethereum, EOS, Tron, etc. are not designed to run real world, complex, data-rich applications. We make a case for why dApps can only run what they are intended for — smart contracts with little to no data (so, nothing much to persist permanently) and very little processing requirements (so, not much RAM or CPU cycles are required) — and why such architectures cannot run even a low end application because of the costs associated with renting RAM/CPU and bandwidth. The implications for the industry are:

1. the **impracticality of launching real world applications on decentralized platforms.** Not all application logic can be wrapped into a set of smart contracts, so platforms must be economically viable for running data-rich web applications
2. **exorbitant cost of running applications on decentralized platforms** — high *decentralization premium*, and
3. **having to secure resources such as RAM and bandwidth in advance** to be able to run the applications. Since the resources are rented or purchased in respective marketplaces, it is extremely hard to forecast the cost of acquiring the required resources in advance, which in turn makes it harder to predict the cost of running the apps on these platforms.

Terminology

CPT — Cost per Thousand Transactions. This is the cost to process one thousand transactions in a blockchain network. This cost is measured against end users or developers and the miners in the network.

Sender (Developer/User) CPT — This is the CPT for end users or developers, depending on who is sending transactions. This cost represents the transaction fee and any other fee associated with reserving network resources, such as bandwidth, CPU, and memory. These fees are paid by developers or users. For example, users pay transactions fees in Bitcoin and Ethereum networks, but reserve RAM and bandwidth on EOS. For dApps launched on EOS, developers may pay for

reserving network resources on behalf of their users. In Storecoin, developers and users pay no transaction fees for settlement transactions, but developers pay for p2p cloud compute resources for hosting their apps on the Storecoin Platform.

Miner CPT — This is the CPT for the miners of the network. Miners run the infrastructure to process transactions and secure the network. Miner CPT measures that cost of running the infrastructure per one thousand transactions.

Miner revenue per thousand transactions — This is the revenue the miners earn per one thousand transactions. When we don't have the data to measure this, we use Sender (Developer/User) CPT as the potential revenue for the miners.

Low end app — An app that stores 512 bytes of data and runs for 20 ms per app instance. This models an app that is equivalent to a smart contract dApp on decentralized networks.

High end app — An app that stores ~9.26 MB of data and runs for 1 second per app instance. This app models a complex, real world, data-rich application.

Summary of CPTs for PoW and PoS networks

Table 1 summarizes the Sender (Developer/User) CPT, miner CPT, and miner revenue for 1,000 transactions for different protocols analyzed in this paper. We'll discuss specific models used for each of the protocols, assumptions, how to interpret the results, and other details later in this paper.

Protocol	Sender (Developer/User) estimated CPT	Total estimated miner revenue per 1,000 transactions	Total estimated miner CPT	Notes
Bitcoin (July 2018 to July 2019)	$1,137	$38,360	$53,823	- Transactions: ~512 bytes. - No other resource renting/purchase necessary. - Predominently settlement transactions.
Cosmos — 100 Validators	$0.827 (EBS) $0.756 (S3)	$1.055	$0.621 (EBS) $0.567 (S3)	
Ripple (July 2018 to July 2019)	$0.651	No Data Available	$0.264	
STORE (settlement) — 92 miners	Zero-Fee	$0.0185	$0.0139	
STORE (platform) — 20 miners per cloud market	Users: Zero-Fee Developers: Low end app: $0.00416 (EBS) $0.00357 (S3) High end app: $18.098 (EBS) $4.5316 (S3)	Low end app: $0.00416 (EBS) $0.00357 (S3) High end app: $18.098 (EBS) $4.5316 (S3)	Low-end app: $0.00312 (EBS) $0.00268 (S3) High end app: $13.574 (EBS) $3.3987 (S3)	- Dynamic Transaction sizes. - Renting/purchasing resources like RAM, bandwidth, etc. may be necessary. - Developer/user CPT is a **range** — computed for both a low and a high-end application. - Support smart contracts, dApps, tokenized Apps, etc.
AWS	Users: Zero-Fee Developers: Low end app: $0.000688 (EBS) $0.000344 (S3) High end app: $10.854 (EBS) $2.8047 (S3)	Low end app: $0.000688 (EBS) $0.000344 (S3) High end app: $10.854 (EBS) $2.8047 (S3)	No Data Available	
EOS — 21 block producers	Low end app: $564 High end app: $10,705,440	$5.93	$10.74	
Tron	Low end app: $9,767 High end app: $58,650,713	No Data Available	No Data Available	
Ethereum (July 2018 to July 2019)	$192	$6,129	$5,823	

Table 1 — Comparison of Sender (Developer/User) CPT and miner CPT for different protocols

Table 1 compares CPTs for Bitcoin, Cosmos, Ripple, and Storecoin settlement layers as one group as they are used predominantly for settlement transactions — transactions that process payments or used to update global state of the blockchain. The Storecoin Platform, AWS, EOS, Tron, and Ethereum are compared as a second group. These platforms run applications of varying capacities — from smart contract based dApps to full fledged data and processing-heavy enterprise apps. For both groups, we calculate the sender CPT separately from the miner CPT, so the cost to senders as well as miners can be compared.

Fig. 1 compares the Sender (Developer/User) CPT differences between various protocols in group 1. Fig. 2 compares the same for protocols in group 2. As an example, consider the first row in fig. 1. The Sender CPT for Bitcoin is ~1,136 times of that for Cosmos. Figures 1 and 2 compare sender costs of one protocol relative to other protocols. Please visit http://storecoin.com/cptdata for more details.

Difference Amount (Sender Cost)	Bitcoin (July 2018 to July 2019)	Cosmos – 100 Validators	Ripple (July 2018 to July 2019)	STORE (settlement) – 92 miners
Bitcoin (July 2018 to July 2019)	$0.000	$1,136.499	$1,136.639	$1,137.290
Cosmos – 100 Validators	-$1,136.499	$0.000	$0.141	$0.792
Ripple (July 2018 to July 2019)	-$1,136.639	-$0.141	$0.000	$0.651
STORE (settlement) – 92 miners	-$1,137.290	-$0.792	-$0.651	$0.000
STORE (platform) low end apps – 20 miners per cloud market	-$1,137.286	-$0.788	-$0.647	$0.004
STORE (platform) high end apps – 20 miners per cloud market	-$1,125.975	$10.523	$10.664	$11.315
AWS low end app	-$1,137.290	-$0.791	-$0.650	$0.001
AWS high end app	-$1,130.461	$6.038	$6.178	$6.829
EOS – 21 block producers - low end app	-$573.290	$563.209	$563.349	$564.000
EOS – 21 block producers - high end app	$10,704,302.710	$10,705,439.209	$10,705,439.349	$10,705,440.000
Tron - low end app	$8,629.710	$9,766.209	$9,766.349	$9,767.000
Tron - high end app	$58,649,575.710	$58,650,712.209	$58,650,712.349	$58,650,713.000
Ethereum (July 2018 to July 2019)	-$945.136	$191.362	$191.503	$192.154

Fig. 1 — Sender cost difference between different protocols in group 1

Difference Amount (Sender Cost)	STORE (platform) low end apps – 20 miners per cloud market	STORE (platform) high end apps – 20 miners per cloud market	AWS low end app	AWS high end app	EOS – 21 block producers - low end app	EOS – 21 block producers - high end app	Tron - low end app	Tron - high end app	Ethereum (July 2018 to July 2019)
Bitcoin (July 2018 to July 2019)	$1,137.286	$1,125.975	$1,137.290	$1,130.461	$573.290	-$10,704,302.710	-$8,629.710	-$58,649,575.710	$945.136
Cosmos – 100 Validators	$0.788	-$10.523	$0.791	-$6.038	-$563.209	-$10,705,439.209	-$9,766.209	-$58,650,712.209	-$191.362
Ripple (July 2018 to July 2019)	$0.647	-$10.664	$0.650	-$6.178	-$563.349	-$10,705,439.349	-$9,766.349	-$58,650,712.349	-$191.503
STORE (settlement) – 92 miners	-$0.004	-$11.315	-$0.001	-$6.829	-$564.000	-$10,705,440.000	-$9,767.000	-$58,650,713.000	-$192.154
STORE (platform) low end apps – 20 miners per cloud market	$0.000	-$11.311	$0.003	-$6.825	-$563.996	-$10,705,439.996	-$9,766.996	-$58,650,712.996	-$192.150
STORE (platform) high end apps – 20 miners per cloud market	$11.311	$0.000	$11.314	$4.485	-$552.685	-$10,705,428.685	-$9,755.685	-$58,650,701.685	-$180.839
AWS low end app	-$0.003	-$11.314	$0.000	-$6.829	-$563.999	-$10,705,439.999	-$9,766.999	-$58,650,712.999	-$192.153
AWS high end app	$6.825	-$4.485	$6.829	$0.000	-$557.171	-$10,705,433.171	-$9,760.171	-$58,650,706.171	-$185.325
EOS – 21 block producers - low end app	$563.996	$552.685	$563.999	$557.171	$0.000	-$10,704,876.000	-$9,203.000	-$58,650,149.000	$371.846
EOS – 21 block producers - high end app	$10,705,439.996	$10,705,428.685	$10,705,439.999	$10,705,433.171	$10,704,876.000	$0.000	$10,695,673.000	-$47,945,273.000	$10,705,247.846
Tron - low end app	$9,766.996	$9,755.685	$9,766.999	$9,760.171	$9,203.000	-$10,695,673.000	$0.000	-$58,640,946.000	$9,574.846
Tron - high end app	$58,650,712.996	$58,650,701.685	$58,650,712.999	$58,650,706.171	$58,650,149.000	$47,945,273.000	$58,640,946.000	$0.000	$58,650,520.846
Ethereum (July 2018 to July 2019)	$192.150	$180.839	$192.153	$185.325	-$371.846	-$10,705,247.846	-$9,574.846	-$58,650,520.846	$0.000

Fig. 2 — Sender cost difference between different protocols in group 2

Setup and assumptions for computing CPT

In this analysis, we compute the CPT for the following protocols and compare them against the CPT for a centralized cloud service. We use the Amazon Web Service (AWS) for its widespread adoption and cheaper cost to the application developers. Similar models can be built to compare the CPT with other centralized cloud services also.

1. Bitcoin
2. Ethereum 1
3. Cosmos
4. EOS
5. Ripple
6. Tron
7. Storecoin (not yet launched).

Some of the networks listed above (Bitcoin, Ripple) are used predominantly to send payment transactions while others (Ethereum, EOS, Tron) are used to develop smart contract based dApps. The Cosmos Network is an *internet of blockchains* and in this analysis we compute the CPT for the Cosmos Hub. Finally, Storecoin is a zero-fee payments and p2p cloud computing public blockchain that allows deployment of complex, data rich applications called "tokenized apps" (tApps). Since Storecoin supports both settlement transactions and tApps, the CPT is computed separately for each layer.

Some of the data we use in this analysis represent estimates while others represent actual information based on available on-chain statistics. We will mark this clearly when we discuss the models below.

Transaction costs are measured across 3 vectors where data is available. They are:

1. Developer / user cost
2. Miner revenue
3. Miner cost

When data for the miner cost is not available, but data for the miner revenue is available, we use the latter with the assumption that marginal revenue is equal to marginal cost.

A major cost factor is the *human cost* — the cost to manage the respective networks. However, this cost is not easy to normalize across different networks. For example, a Validator node on the Cosmos Network can be managed by a competent devops person, but a large Bitcoin mining farm

may require a much bigger team to manage its operations. For this reason, we do not include this cost in the miner CPT calculations.

Methodology

The protocols analyzed in this paper belong to different categories (PoW vs PoS), provide different capabilities (payments, smart contracts, application development, etc.), work in different setups (private, centralized, permissionless, and authenticated), and use different economic incentive models to reward their miners. So, a single methodology cannot be used to compute the CPT across all protocols. At the same time, the methodology selected for a protocol must be *fair*, so the results are comparable. In this section, we describe the methodology used to compute the CPT for each protocol listed in table 1.

AWS (Amazon Web Services — Centralized cloud service)

We analyze developing and running apps on a centralized cloud service, such as the AWS, to model the cost where the "decentralization premium" is zero. Cost to developers should be lowest in this case. We make the following assumptions in this model.

1. *Transactions* in decentralized networks are equivalent to *instances* of apps on centralized cloud services.
2. *Throughput* in transactions/second (TPS) in decentralized networks is equivalent to *concurrency* on centralized cloud services.
3. Developers bear the cost of hosting their apps on centralized cloud services.
4. The apps query and read 5 x the data created and written by them. So, read operations and data transfers are 5 x the write operations and data transfers.
5. A typical cloud deployment requires using load balancers and firewalls. While AWS provides these services [6.b, 6.c, 6.d, 6.e], the cost of using them parallels or outstrips the cost of running compute instances and storage services on AWS. The cost is mainly due to the volume of data transfers arising from the high throughput we assume in this model. For this reason, we assume that app developers use alternative approaches listed below as cheaper options of these services.
    a. Nginx based application load balancer [3.c].
    b. Hardware based firewall [3.d].

    Using these options requires expertise on them and operational overheads exist, but cost savings justify using them.

In order to calculate the CPT for developers, we assume two scenarios.

1. An app that stores 512 bytes of data per app instance and runs for 20 ms. This models an app that is equivalent to a smart contract dApp on decentralized networks.
2. An app that stores ~9.26 MB of data per app instance and runs for 1 second. This app models a complex, real world, data-rich application.

Parameters	512 byte, 20 ms/instance app	9.26 MB, 1 second/instance app
Throughput (tps)	5,000	5,000
Transaction (record) size (tsz)	512 bytes	9.26 MB
Memory required in MB to run one instance of the app (m)	32	512
Execution duration in ms (t)	20	1,000
Number of app instances that can be run in 1GB memory in 1 second (gbs) = 1,024/m x 1000/t. Practically, the efficiency is ~half of this number.	1,600/2 = 800	2/1 = 1
RAM required in GB to support claimed throughput (r) = tps/gbs	6.25	5,000
Amazon EC2 instance [3.b] to match the memory capacity required above. We assume developers prepay annually.	2 x c5n.large @$946.08 each	7 x r5.24xlarge @ $52,980.48 each
Annual cost of Amazon EC2 instances (Cec2)	$1,892.16	$370,863.36
Nginx server cost for load balancing [3.c] amortized over 3 years (Cng)	2 x $1,100 (10 gbps throughput) = $2,200/3 = $733.33	2 x $1,100 (10 gbps throughput) $2,200/3 = $733.33
Firewall [3.d] cost amortized over 3 years (Cfw)	2 x $1,150.60/3 = $767.06	2 x $1,150.60/3 = $767.06
Estimated annual storage[1] in GB (S) = tps x tsz	75,187.6831	1,425,895,312.5
Annual cost of storage with EBS (SSD gp2 volume) @ $0.10 per GB-month (Cebs)	$90,225.22	$1,711,074,375
Annual cost of storage with S3 @ $0.023 per GB-month for up to 50TB and $0.021 per GB-month for higher tiers + data access + data transfer (Cs3)	@$2992.28 per month = $35,907.36	@ 36,821,747.63 per month = $441,860,971.56

[1] In the first year of app deployment, the storage accumulates on a daily basis, so the annual storage estimated is not rented from day 1. However, over time, estimated annual storage needs to be rented for 1 year's worth of data, so we assume that for calculating storage cost.

Enterprise broadband internet connection $7,800 per year (Cbb)	$7,800	$7,800
Electricity cost @ $4,200 per year (Cel)	$4,200	4,200
Rent cost [4.b] @ $3,000 per year (Cre)	$3,000	$,3000
Total annual cost to developer with EBS storage (Tebs) = Cec2 + Cng + Cfw + Cebs + Cbb + Cel + Cre	$108,617.77	$1,711,461,738.75
Total annual cost to developer with S3 storage (Ts3) = Cec2 + Cng + Cfw + Cs3 + Cbb + Cel + Cre	$54,299.91	$442,248,335.31
Developer CPT with EBS storage = Tebs / (Annual Tx) x 1,000	**$0.000688**	**$10.854**
Developer CPT with S3 storage = Ts3 / (Annual Tx) x 1,000	**$0.000344**	**$2.8046**
Miner CPT	No data available	No data available
Miner (Amazon) revenue per 1,000 transactions with EBS storage (same as the developer CPT)	**$0.000688**	**$10.854**
Miner (Amazon) revenue per 1,000 transactions with S3 storage (same as the developer CPT)	**$0.000344**	**$2.8046**

Table 2 — Developer CPT on AWS

Table 2 illustrates that developer cost and hence their CPT depends largely on the choices they make on storage services, amount of data being read and written, and optimizations to other services they need, such as the load balancer, firewall, etc. In this analysis, we assume that the data is read frequently, so we don't consider modeling with infrequent storage options provided by AWS.

Data to compute the miner (Amazon) CPT is not available for this model.

Bitcoin

Average transaction size [1.b]	~500 bytes
Period for transaction fee data [1.c]	July 8, 2018 — July 7, 2019
Transaction fees collected in the above period	$120,483,216
Number of transactions in the above period (N)	105,938,859

Average cost per transaction (Tc)	$1.137
Sender (Developer/User) CPT = Tc x 1,000	**$1,137**
Miner revenue in the above period [1.c] (R)	$4,063,786,796
Revenue per transaction (Tr) = R / N	$38.36
Miner CPT (method 1: assuming marginal revenue can be set equal to marginal cost) = Tr x 1,000)	**$38,360**
Average daily hashrate in the above period [1.c]	46,948 ph/second
Number of S9 Antminers [1.f] (@14 th/second) required to produce daily hashrate	3,353,428.5714
Daily electricity required (@1.475kW) by S9 Antminers	118,711,371.42
Electricity cost per day in China [1.f] @$0.08/kWh (Ec)	$9,496,909.71
Amortized cost of S9 Antminers @$2,000 per unit, over 3 years Ac = ((3,353,428.5714 x $2,000) / 3) / 365	$6,124,983.69
Total annual cost to miners Mc = (Ec + Ac) x 365	$5,701,991,091
Miner CPT (Method 2, based on the cost to miners) = (MC / N) x 1,000	**$53,823**

Table 3 — Sender (Developer/User) CPT and miner CPT for Bitcoin

Cosmos

The Validators in the Cosmos Hub don't interpret what the transactions are and don't impose any size restrictions. The transaction size depends on individual zones connected to the Cosmos Hub. For this analysis, we assume a transaction of ~500 bytes.

Average transaction size	~500 bytes
Period used to compute average ATOM price [2.c]	June 15, 2019 — July 14, 2019
Average price of ATOM in the above period (A)	**$5.73**
Number of Validators in the Cosmos Hub (N)	100
Average block reward given to Validators (r)	3.81
Average block time	6.892 seconds

Estimated blocks created annually (B)	4,575,740
Estimated annual block rewards (Br) = B x r	17,433,569
Estimated Validator commission (vc)	10%
Estimated annual miner revenue (R) = vc x Br x A	$9,981,881
Estimated average throughput (th)	300 TPS
Estimated revenue for 1,000 transactions	**$1.055**
Estimated block size (bs)	2MB
Estimated average hardware cost (2 x servers @$10,000 each + 2 HSM @ $150 each) [2.e] amortized over 3 years (hc)	$6,767
Estimated average annual operating cost: - Cost for firewall and server operation = $900/month - Backup location operation = $900/month - 5 Sentry nodes on AWS = $300/month (oc)	$25,200
Annual storage required = B x bs	9.15148 TB
Cost of storage @0.125 per GB per month and $0.065 per provisioned IOPS-month on Amazon EBS Provisioned IOPS[2.f] SSD volume[2] (cebs)	(9.15148 x 1000 x 0.0625 x 12) + (0.065 x 1,000 x 12) = **$7,643.61**
Cost of storage @0.023 per GB per month and $0.005 per 1,000 PUT and $0.0004 for 1,000 GET requests [2.f] on Amazon S3 (cs3). We assume 1 GET and 1 PUT requests per block.	(9.15148 x 1000 x 0.023 x 12) + (0.005 + 0.0004) x 4,575,740 / 1,000 = **$2,550.51**
Estimated average self staking per Validator [2.b] in ATOMs	10,000
Cost of self staking per Validator, spread over 3 years (sc)	$19,100
Total annual cost with EBS storage to Cosmos Validators Cebs = N x (hc + oc + cebs + sc)	$5,871,027
Total annual cost with S3 storage to Cosmos Validators Cs3 = N x (hc + oc + cs3 + sc)	$5,361,717
Miner CPT with EBS storage = (Cebs / total annual transactions) x 1,000	**$0.6205**
Miner CPT with S3 storage = (Cs3 / total annual transactions) x 1,000	**$0.5667**

[2] We assume storage on Amazon because this avoids having to buy, provision, maintain, and upgrade SSD storage servers as data usage increases. With cloud storage, the cost is accrued as the storage is used.

Sender (Developer/User) CPT with EBS storage = Miner CPT / 0.75 (assumes 25% profit margin on miner cost)	**$0.82733**
Sender (Developer/User) CPT with S3 Storage	**$0.7555**

Table 4 — Sender (Developer/User) CPT and miner CPT for Cosmos

At present, transaction fees are not uniformly assessed and enforced by Validators. Cosmos allows each Validator to determine the transaction fees based on their cost of running the nodes and desired profitability. We are unable to get reliable data on transaction fees, so we assume that Validators expect a 25% profit margin and use it to determine the Sender (Developer/User) CPT in table 3.

At present, there is no data available on how Validators store blockchain data. In this analysis, we assume storage on Amazon EBS and S3[2.f] since both can be provisioned and used on demand to control the cost of storage. For S3, we assume that Validators perform all persistence related calculations locally and minimize reading from and writing to S3.

Ripple

Period used to compute the annual number of transactions and transaction fees collected [3.a]	July 08, 2018 — July 07, 2019
Annual number of transactions [3.a] (Tx)	207,314,968
Annual transaction fees[3.a] (Tf)	$134,944
Sender (Developer/User) CPT = (Tf / Tx) x 1,000	**$0.65**
Number of Validators in Ripple private network (N)	5
Estimated annual storage (St) [3.e]	3.0 TB
Estimated average transaction size	500 bytes
Estimated hardware cost per Validator (I3 High I/O Quadruple Extra Large) [3.b, 3.e]. This instance comes with 3.8TB of SSD storage. (C)	$10,932
Estimated total cost to run 5 Validators (Ct) = N x C	**$54,660**
Estimated miner CPT = (Ct / Tx) x 1,000	**$0.2637**
Estimated throughput Ripple is capable of based on [3.f]	1,500

Estimated miner CPT based on the above throughput = (Ct / 1,500 x 60 x 60 x 24 x 365) x 1,000	**$0.00115**
Miner revenue	No data is available

Table 5 — Sender (Developer/User) CPT and miner CPT for Ripple

The Sender (Developer/User) CPT is computed based on the annual number of transactions and transaction fees collected [3.a]. Since this data is readily available, Sender (Developer/User) CPT calculation is straightforward. We need the cost of running Ripple nodes in order to compute the miner CPT. However, this information is not available. So, we use the recommendation [3.e] Ripple provides to determine the estimated cost of running a Ripple node and the resulting miner CPT. Ripple claims it can handle 1,500 transactions per second [3.f]. We also estimate the miner CPT based on this claim. As the throughput increases, the miner CPT decreases.

We are also unable to collect data regarding miner revenue and hence we are not able to calculate the miner revenue per 1,000 transactions.

Storecoin (settlement layer)

Storecoin settlement layer supports zero-fee transactions. **Developers/user neither pay a transaction fee nor reserve network resources to have their transactions processed.** The Storecoin network in the settlement layer consists of two types of nodes — a) compute nodes, called Validators and b) consensus and storage nodes, called Messagenodes. So, miner CPT and miner revenue per 1,000 transactions are computed for these two node types. Storecoin's Byzantine Fault Tolerant consensus algorithm [4.a] assembles and validates blocks in a pipelined, multi-stage process, resulting in multiple blocks finalized on a continuous basis. Table 6 analyzes the cost of the Storecoin settlement layer in one of its launch phases.

Parameters	Validators	Messagenodes
Number of nodes in the network (N)	70	22
Estimated number of blocks finalized per second (b)	5	
Estimated number of transactions in a block (tb)	550	
Estimated block size in KB (bsz) (transactions + header + signatures)	300	
Estimated throughput in TPS (Tx) = b x tb	2,750	

Estimated annual storage in GB (s) = b x (bsz in GB) x number of seconds in 1 year	44,055.28	
Nginx server (entry level) cost for load balancing [3.c] amortized over 3 years (Cng)	2 x $750 (1 gbps throughput) x 70 /3 = $35,000	2 x $750 (1 gbps throughput) x 22 /3 = $11,000
Firewall [3.d] (entry level) cost amortized over 3 years (Cfw)	2 x $807.40 x 70/3 = $37,678.67	2 x $807.40/3 = $11,841.87
1/ Hardware option:		
Supermicro SYS-E300-8D server @ $1,500 per server, amortized over 3 years (Cser)	x 2 x 70 / 3 = $70,000	x 2 x 22 / 3 = $22,000
Storage @ $47/GB (Cst) = s x $47 (Each Messagenode stores its own copy)		$2,070,598.16 x 22 = $45,553,159.51
Enterprise broadband internet connection $7,800 per year (Cbb)	7,800 x 70 = $546,000	7,800 x 22 = $171,600
Electricity cost @ $4,200 per year (Cel)	4,200 x 70 = $294,000	4,200 x 22 = $92,400
Rent cost [4.b] @ $3,000 per year (Cre)	$3,000 x 70 = $210,000	$,3000 x 22 = $66,000
Total cost with hardware option (Thw) = Cng + Cfw + CSer + Cst + Cbb + Cel + Cre	$1,192,678.67	$45,928,001.38
Total annual network cost for all Validators and Messagenodes (Chw)	**$47,120,680.04**	
Miner CPT = Chw / (annual Tx) x 1,000	**$0.54334**	
Estimated miner revenue per 1,000 transactions = 25% profit over miner CPT	**$0.7244**	
1.a/ Storage optimized with erasure coding and 3 copies of data		
Storage @ $47/GB (Cst) = s x $47 (Messagenode pool storage with 3 copies of erasure encoded data)		$2,070,598.16 x 1.2 x 3 = $7,454,153.38
Total annual network cost with optimized storage	**$9,021,673.91**	
Miner CPT with optimized storage	**$0.1040**	
Estimated miner revenue per 1,000 transactions = 25% profit over miner CPT	**$0.1387**	
2/ Cloud option (nodes are hosted on AWS):		

Amazon EC2 (Cser)	c5d.large @ $840.96 per instance x 2 x 70= $117,734.4	m5d.large @ $989.88 x 4 x 22 = $87,109.44
Storage: latest 1 month's data on EBS (SSD gp2 volume) and 11 month's data on S3 x 3 copies (Cst)		a) 3671.2733 GB @ $0.1 GB-month x 22 = **$8,076.80** b) 40,384.0066 x 3 GB on S3 + (data access + data transfer) x 5 x number of blocks in 11 months @ $4,273.52 per month = **$51,282.24** Total: **$59,359.04**
Enterprise broadband internet connection $7,800 per year (Cbb)	7,800 x 70 = $546,000	7,800 x 22 = $171,600
Electricity cost @ $1,440 per year (Cel) (Estimated only for the firewall)	1,440 x 70 = $100,800	4,200 x 22 = $31,680
Total cost with hardware option (Tcloud) = Cng + Cfw + CSer + Cst + Cbb + Cel	$837,213.07	$372,590.35
Total annual network cost for all Validators and Messagenodes (Cloud)	**$1,209,803.42**	
Miner CPT = Cloud / (annual Tx) x 1,000	**$0.01395**	
Estimated miner revenue per 1,000 transactions = 25% profit over miner CPT	**$0.0186**	
Developer CPT (Zero-fee transactions for developers)	**$0**	
User CPT (Zero-fee transactions for users)	**$0**	

Table 6 — Miner CPT and miner revenue per 1,000 transactions for Storecoin settlement layer

Miner Revenue per 1,000 Transactions — Since the Storecoin network is not launched yet, there is no information on the revenue for miners. For the sake of analysis, we will assume the value of revenue earned by Storecoin miners will be calculated based on a 25% margin requirement set on total miner costs. This means if the miner CPT is $1.00, then the revenue per 1,000 transactions would be $1.33. Note: ~$0.33 in gross profit divided by $1.33 in total revenue is ~25% gross profit margin.

Storecoin (platform layer)

Storecoin peer-to-peer decentralized cloud platform allows arbitrary web applications to run in a secure sandbox. It is not a smart contract platform to developer dApps. Developers pay for network resources, such as memory, storage, and bandwidth and the developer experience will be similar to centralized cloud services like AWS. For that reason, we'll use the same scenarios we used for analyzing the developer cost for AWS, so the costs can be compared for "decentralization premium".

In the Storecoin Platform, miners organize into "Cloud Markets" to provide cost effective cloud services to app developers. A Cloud Market consists of a subset of Validators and Messagenodes, who help secure the apps hosted on it. For this analysis, we assume a cloud market consisting of 10 Validators and 10 Messagenodes. Table 7 models the miner cost analysis for the Storecoin Platform. Since the goal is to compute the "decentralization premium", this model also assumes that Storecoin nodes will be hosted on AWS — this model doesn't explore the hardware-only option.

Parameters	512 byte, 20ms/instance app	9.26 MB, 1 second/instance app
Number of Validators in the Cloud Market (Nv)	10	
Number of Messagenodes in the Cloud Market (Nm)	10	
Estimated throughput (tps)	5,000	5,000
Transaction (record) size (tsz)	512	9.26 MB
Memory required in MB to run one instance of the app (ma)	32	512
Memory overhead of the sandbox in MB, per app instance (ms)	128	128
Total memory required in MB to run one instance of the app (m)	160	640
Execution duration of the app in ms (ta)	20	1,000
Startup cost of the sandbox in ms (ts)	2 x 5	2 x 5
Total duration to run one instance of the app (t)	30	1,010
Number of app instances that can be run in 1GB memory in 1 second (gbs) = 1,024/m x 1000/t. Practically, the efficiency is ~half of this number.	213/2 = 106	1.5841 / 2 = 0.7920

RAM required in GB to support claimed throughput (r) = tps/gbs	47.1698	6,313.1313
Apps are distributed to Validators for execution. Average RAM required per Validator = r / Nv	4.71698	631.31313
Amazon EC2 instance [3.b] to match the memory capacity required above. We assume Validators prepay annually.	(2 x c5n.large @$946.08 each) x 10	(61 x c5n.xlarge @ $1,892.16 each) x 10
Annual cost of Amazon EC2 instances for Validators (Vec2)	$18,921.60	$1,154,217.6
Annual cost of Amazon EC2 instances for Messagenodes (Mec2)	m5d.large @ $989.88 x 4 x 10 = $39,595.2	m5d.large @ $989.88 x 4 x 10 = $39,595.2
Nginx server cost for load balancing [3.c] amortized over 3 years (Cng) (For Validators + Messagenodes)	2 x (2 x $750 (1 gbps throughput) x 10) /3 = $10,000	(2 x $1,100 (10 gbps throughput) x 10) / 3 = $14,666.66
Firewall [3.d] cost amortized over 3 years (Cfw) (For Validators + Messagenodes)	2 x (2 x $807.40) x 10 /3 = $10,765.33	2 x (2 x $1,150.60) x 10 /3 = $15,333.33
Enterprise broadband internet connection $7,800 per year (Cbb) (For Validators + Messagenodes)	7,800 x 10 x 2 = $156,000	7,800 x 10 x 2 = $156,000
Electricity cost @ $4,200 per year (Cel) (For Validators + Messagenodes)	4,200 x 10 x 2 = $84,000	4,200 x 10 x 2 = $84,000
Rent cost [4.b] @ $3,000 per year (Cre) (For Validators + Messagenodes)	$3,000 x 10 x 2 = $60,000	$3,000 x 10 x 2 = $60,000
Estimated annual storage[3] in GB (S) = tps x tsz with 25% overhead for metadata	75,187.6831 x 1.25 = 93,984.60	1,425,895,312.5 x 1.25 = 1,782,369,140.625
Annual cost of storage with EBS (SSD gp2 volume) @ $0.10 per GB-month (Cebs) with erasure coded data (All Messagenodes share this cost)	$11,2781.52	$2,138,842,968.75
Annual cost of storage with S3 at $0.021 per GB-month for higher tiers + data access + data transfer (Cs3)	@$3,726.31 per month = $44,715.72	@ 44,532,276.53 per month = $534,387,318.36
Total annual cost to all Validators (Vc) = Vec2 + (Cng + Cfw + Cbb + Cel + Cre) / 2	$179,304.27	$1,319,217.60

[3] In the first year of app deployment, the storage accumulates on a daily basis, so the annual storage estimated is not rented from day 1. However, over time, estimated annual storage needs to be rented for 1 year's worth of data, so we assume that for calculating storage cost.

Total annual cost to Messagenodes with EBS storage (Mebs) = Mec2 + (Cng + Cfw + Cbb + Cel + Cre) / 2 + Cebs	$312,759.385	$2,139,047,563.945
Total annual cost to Messagenodes with S3 storage (Ms3) = Mec2 + (Cng + Cfw + Cbb + Cel + Cre) / 2 + Cs3	$244,693.585	$534,591,913.555
Validator CPT = Vc / (Annual Tx) x 1,000	**$0.00113**	**$0.00836**
Messagenode CPT with EBS storage = Mebs / (Annual Tx) x 1,000	**$0.00198**	**$13.565**
Messagenode CPT with S3 storage = Ms3 / (Annual Tx) x 1,000	**$0.00155**	**$3.390**
Miner CPT with EBS storage = (Vc + Mebs) / (Annual Tx) x 1,000	**$0.00312**	**$13.574**
Miner CPT with S3 storage = (Vc + Ms3) / (Annual Tx) x 1,000	**$0.00268**	**$3.3987**
Miner revenue per 1,000 transactions with EBS storage = 25% profit over miner CPT	**$0.00416**	**$18.098**
Miner revenue per 1,000 transactions with S3 storage = 25% profit over miner CPT	**$0.00357**	**$4.5316**
User CPT (Zero-fee tokenized app transactions for users)	**$0**	
Developer CPT with EBS storage (same as the Miner revenue)	**$0.00416**	**$18.098**
Developer CPT with S3 storage (same as the Miner revenue)	**$0.00357**	**$4.5316**

Table 7 — Sender (Developer/User) CPT, Miner CPT and miner revenue per 1,000 transactions for Storecoin platform layer

The developer CPT is the same as miner revenue above because the Storecoin Platform is not live yet. We expect that miners require this payment from developers.

An interesting aspect of a developer running on Storecoin platform is that if they are paying miners in the STORE token, it means they get to keep 100% of the datacoins minted from their data usage. This could help offset costs they pay to miners. The amount of revenue generation that represents for the developer would depend on the value of their data over time. If miners think the data is

going to be valuable for a given app, they are likely to give them free compute resource in return for a guaranteed percentage share of the datacoin revenue and/or issuance over time.

EOS

Period used to compute the annual transactions and block rewards for EOS	July 08, 2018 — July 07, 2019
Average block producer cost per year [7.e and 7f] (Bc) (low: $80,000 and high: $1,400,000)	$925,625
Number of block producers (N)	21
Total estimated cost to block producers (C) = Bc x N	$19,438,125
Total number of annual transactions [7.b and 7.d] (Tx)	1,809,078,749
Throughput (tps) based on Tx above	~**58**
Miner CPT = (C / Tx) x 1,000	**$10.745**
Starting EOS supply on July 08, 2018 [7.d] (Ss)	886,777,795
Estimated annual EOS issuance at 5% rate (Ai) = Ss x 5%	44,338,890
Annual issuance to block producers (bi) = 20% x Ai	8,867,778
Annual issuance to 21 top block producers (Bi) = 25% x bi	2,216,944
Average price of EOS in the above period	4.84
Average revenue in the same period (R)	$10,730,011
Miner revenue per 1,000 transactions = (R / Tx) x 1,000	**$5.93**

Table 8 — Miner CPT and miner revenue for 1,000 transactions for EOS

EOS has no transaction fees, but developers/users must stake in EOS to use compute resources from block producers (BPs). Developers/users have to lockup EOS tokens until they are done using network resources. This presents some challenges and major cost variances.

In order to calculate Sender (Developer/User) CPT, we assume two scenarios.

3. An app that stores 512 bytes of data per app instance. So, RAM usage is 512 bytes. This models a simple smart contract application.

4. An app that stores ~9.26 MB of data per app instance. This app models a complex, real world, data-rich application.

For each scenario, we assume that the annual transactions shown in the table 8 are of that type. Table 9 builds Sender (Developer/User) CPT based on the above assumptions.

Parameters	512 byte app	9.26 MB app
Cost per kilobyte as of early May 2019 [7.a] (Ck)	$1.129	$1.129
Storage required per transaction in kilobytes (Sk)	0.5	9,482.24
Cost of storage per transaction (Tc) = Ck x Sk	$0.5645	$10,705.44
Sender (Developer/User) CPT = Tc x 1,000	**$564.5**	**$10,705,440**

Table 9 — Miner CPT for EOS

On EOS network [7.a], developer/users may not even be able to reserve the required RAM because not enough RAM may be available in the marketplace. The purpose of this analysis is to demonstrate that even a simple app that stores half a kilobyte of data has a high cost to developers/users. Of course, the staked EOS tokens are returned when users release the reserved storage, but otherwise, tokens are locked making this a true cost to run apps on EOS.

For this analysis, we didn't consider reserving the bandwidth [7.a], which is required in addition to reserving storage. Since the cost per transaction is already unreasonable, we didn't consider that including the bandwidth in the calculation alters that conclusion.

TRON

TRON is a hybrid between requiring developers/users to pay and requiring them to stake for using network resources. TRON measures its resources in terms of energy points (to measure CPU use) and bandwidth (bytes). A certain amount of TRX is required per energy point and per byte used in order run dApps on TRON network. In this analysis we assume two app scenarios as we did in EOS model.

1. An app that stores 512 bytes of data and takes 20ms to run per app instance. This models a simple smart contract application.
2. An app that stores ~9.26 MB of data and takes 1 second to run per app instance. This app models a complex, real world, data-rich application.

.

Parameters	512 byte, 20ms app	9.26 MB, 1 second app
Runtime of the transaction in ms (rt)	20	1,000
Energy points required per transaction at 1 energy point per microsecond (e)	20,000	1,000,000
Price per energy point in[8.a and 8.c] TRX (pe)	0.009726	0.009726
TRX required per transaction (Te) = pe x e	194.52	9,726
Transaction size in bytes (Tx)	512	9,709,814
Price per byte[8.a and 8.b] in TRX (be)	0.17459	0.17459
TRX required per transaction (Tb) = be x Tx	89.39	1,695,236.42
Total cost per transaction (C) = Te + Tb	283.91	1,704,962.58
Price per TRX (p)	$0.0344	$0.0344
Sender (Developer/User) CPT = C x p x 1,000	$9,766.5	$58,650,713
Miner CPT	No data available	No data available
Miner revenue per 1,000 transactions	No data available	No data available

Table 10 — Sender (Developer/User) CPT for TRON

Ethereum

Ethereum is a PoW-based smart-contract platform to develop dApps. There is no limit on the transaction size, but there is a gas size limit per block, which determines how much computation can be included in a given block [9.a]. This means there is some upper limit for the computation of a given transaction.

Period used to compute the annual transactions and block rewards for Ethereum	July 08, 2018 — July 07, 2019
Annual number of transactions in the above period [9.b] (Tx)	226,499,001
Annual transaction fees collected in the above period (f)	$43,522,664
Sender (Developer/User) CPT = f/Tx x 1,000	**$192**
Miner revenue [9.b] from block rewards and transaction fees in the above period (R)	$1,388,219,425

Miner revenue per 1,000 transactions = R/Tx x 1,000 (Assuming marginal revenue can be set equal to marginal cost, this is one way to calculate miner CPT.)	**$6,129**
Average daily hashrate [9.b] in mH/second (h)	201,747,525.13
Daily hashrate of Sapphire [9.c] Radeon RX 570 GPU used by miners in mH/second (hs)	25
Daily power used by Sapphire Radeon RX 570 GPU in kWh at a rating of 130 watts (Ps) = 0.13 x 24	3.12
Average cost of Sapphire Radeon RX 570 GPU (cs)	$217
Theoretical number of GPUs required to produce average daily hashrate (Ns) = h / hs	8,069,901
Total daily power required for all GPUs (P) = Ns x Ps	25,178,091.12
Total daily cost of power required at $0.08 per kWh (Cp) = P x $0.08	$2,014,247.29
Daily cost of GPUs required to produce daily hashrate, amortized over 3 years for its useful life (Cs) = (cs x Ns) / 365 / 3	$1,599,240.65
Total cost to produce daily hashrate (Cd) = Cs + Cp	$3,613,487.94
Estimated annual cost of hashrate (C) = Cd x 365	$1,318,923,098.1
Miner CPT = C / Tx x 1,000 (Based on the cost to miners)	**$5,823.08**

Table 11 — Sender (Developer/User) CPT, miner CPT, and miner revenue per 1,000 transactions for Ethereum

References

1. Bitcoin —
    a. https://en.bitcoin.it/wiki/Maximum_transaction_rate
    b. https://tradeblock.com/blog/analysis-of-bitcoin-transaction-size-trends
    c. https://www.blockchain.com/charts/
    d. https://medium.com/@andrew_young/valuing-bitcoin-694d97e5bdf2
    e. https://www.ovoenergy.com/guides/energy-guides/average-electricity-prices-kwh.html
    f. https://en.bitcoinwiki.org/wiki/Antminer_S9
2. Cosmos Network —
    a. https://www.mintscan.io/

   b. https://cosmos.bigdipper.live/
   c. https://coinmarketcap.com/currencies/cosmos/historical-data/
   d. https://stakingrewards.com/asset/atom
   e. https://medium.com/@davekaj/how-to-become-a-cosmos-validator-276862d5bfc7
   f. https://aws.amazon.com/ebs/pricing/ and https://aws.amazon.com/s3/pricing/
3. Ripple —
   a. https://coinmetrics.io
   b. https://www.ec2instances.info/?cost_duration=annually
   c. https://cdn-1.wp.nginx.com/wp-content/files/nginx-pdfs/Sizing-Guide-for-Deploying-NGINX-on-Bare-Metal-Servers.pdf
   d. https://www.firewalls.com/barracuda-nextgen-firewall-f18-26818.html?gclid=EAIaIQobChMInv2zovjq3gIVWyCtBh1lyQLMEAYYASABEgIZm_D_BwE
   e. https://xrpl.org/capacity-planning.html
   f. https://www.ripple.com/xrp/market-performance/
4. Storecoin (settlement layer) —
   a. https://research.storecoin.com/BlockfinBFT
   b. https://www.atlantech.net/colocation-pricing
5. Storecoin (platform layer) —
   a. https://storecoin.com/cloud
6. AWS —
   a. https://aws.amazon.com/s3/pricing/
   b. https://aws.amazon.com/elasticloadbalancing/pricing/
   c. https://aws.amazon.com/config/pricing/
   d. https://aws.amazon.com/waf/
   e. https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html
7. EOS —
   a. https://www.eosrp.io/
   b. https://coinmetrics.io/
   c. https://stakingrewards.com/asset/eos
   d. https://coinmarketcap.com/currencies/eos/#charts
   e. https://docs.google.com/spreadsheets/d/1OFoOvzgqkkaGAgemi5kNBCbzyp3a8sCk6CCEbG7Y9zA/edit#gid=230278607
   f. https://medium.com/coinmonks/survey-of-eos-block-producers-cf9677561db7
8. TRON —
   a. https://tronstation.io/calculator
   b. https://developers.tron.network/docs/bandwith
   c. https://developers.tron.network/docs/energy
9. Ethereum —

a. https://medium.com/@piyopiyo/how-to-get-ethereum-block-gas-limit-eba2c8f32ce
b. https://etherscan.io/charts
c. https://miningchamp.com/gpus/150/Sapphire-Radeon-RX-570-8GB-hashrate