TOIChain™: A Proposal for High Performance Tamper Resistant Transactions without Scaling Limits

Justin Y. Shi

SMC Labs, 630 Freedom Business Drive, King of Prussia, PA 19406, USA
SERC 315, College of Science and Technology, Temple University, PA 19122, USA
justin@smclabs.tech or shi@temple.edu

Abstract. This paper proposes a blockchain-based high performance transaction processing system called TOIChain. A new programming paradigm and architecture using Active Content Addressable Networking protocol and Statistic Multiplexed Computing runtime is introduced to overcome the typical service infrastructure scalability challenges.

Keywords: high performance parallel blockchain, statistic multiplexed computing, unlimited scalability, tamper resistance, fault tolerance, active contend addressable networking, foundations of distributed and parallel computing.

1 Introduction

This paper presents a scalable blockchain system called TOIChain™ which is designed to facilitate the growth, speed, and efficiency of secure and tamper resistant transaction-based networks without scaling limits.

Web based services and mobile applications are evolving from Web 2.0 to Web 3.0 thanks to continuing innovations such as Blockchain. However, infrastructure flaws and limitations continue to restrict performance, reliability, and security of web services. Most traditional financial services still rely on inefficient and often flawed trust management systems.

Transaction processing is a timeless cornerstone of all infrastructures. Since 2008, the Blockchain protocol has demonstrated breakthroughs in service reliability and security of token transactions on a global scale. In 2014, the creation of a blockchain-like DLT (distributed ledger technology) infrastructure enabled the development of decentralized finance (DeFi) applications. Today, there are more than 1,000 blockchains running worldwide. All of these are designed to provide a reliable, trustless, tamper resistant and scalable transaction processing system.

The first principle of any scalable transaction processing service is to have zero single-point-failure. Almost all blockchain projects meet this requirement without relying on any traditional backup/restore methods. In contrast, legacy database systems must rely on backup/restore methods to mitigate single-point-failures while risking arbitrary data losses due to unplanned downtimes. Consequently, upscaling a legacy database introduces more single-point-failures. Legacy databases are the sitting ducks for Web 2.0 distributed denial of service (DDoS) attacks [1].

The second principle for transaction service is scalable performance which means regardless of transaction volumes, the response time for each transaction should be near a constant. Legacy databases use data partitioning methods to meet performance demands by sacrificing service availability (more single-point-failures). Similar efforts, such as chain partition (sharding), in blockchain protocols have also been developed. However, these have introduced new challenges in security and protocol complexity. To date, their performance benefits are still inconclusive [2].

For immutable ledgers, the storage layer must also be infinitely scalable in performance and capacity. Existing technologies do not meet these requirements.

In this paper we disclose a parallel blockchain platform called TOIChain™ that can deliver incrementally improved performance (and capacity), reliability, and security as the network expands indefinitely. TOIChain™ is not a new blockchain protocol per se, but rather, a blockchain protocol running on a new distributed and parallel architecture.

In French, the word 'toi' is a singular informal pronoun meaning 'you'. It is pronounced “twa”. TOI-715b is a remote planetary system that is one and half times as wide as Earth and orbit within the conservative habitable zone around its parent star [26]. TOIChain™ is an innovative network platform powered by a new computing technology called SMC (statistical multiplexed computing). SMC relies on a new Active Content Addressable Networking (ACAN) paradigm built on top of packet-oriented TCP/IP protocols. TOIChain™ is designed to empower users by providing incrementally better performance, reliability, and security, which is critically needed as the networking community and supporting infrastructures continued expansive evolution.

This paper is organized as follows:

- Section 2 Motivations,
- Section 3 Scalability Proof,
- Section 4 ACAN for Blockchain,
- Section 5 SMC for Blockchain,
- Section 6 TOIChain,
- Section 7 Smart Contracts, dApps, and Tamper Resistance,
- Section 8 Energy Efficiency and Self-Optimization,
- Section 9 Broad Impacts,
- Section 10 Summary.

2 Motivations

2.1 Single-point Failure: 1 + 1 < 1

Synchronous or asynchronous data (transaction) replication methods are the industry standards for mitigating single-point-failures in existing infrastructures.

Fig. 1 illustrates the problems in these standard failure mitigation methods using backup servers.

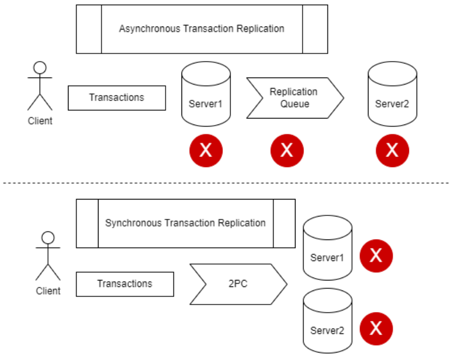


Fig. 1. Industry Standard Data Server Failure Mitigation Methods.

For asynchronous data (or transaction) replication, the primary server is responsible for all workloads and the replication of the transactions. It can deliver less than 100% performance due to replication overheads. The service uptime is also proportionally less than a single server configuration due to the increased probability of server failures. The only benefit is when "server1" fails, "server2" can resume operation. Since there are no reliable failure detection mechanisms possible, arbitrary data losses cannot be eliminated.

For synchronous data (or transaction) replication, a two-phase-commit (2PC) protocol is required. Either "server1" or "server2" failure will roll back the entire transaction. Data loss is prevented with a severe performance penalty. Two servers can deliver less than a single server performance. Service availability is also less than a single server configuration.

Both standards deliver the "1 + 1 < 1" analogy in service performance and availability. The backup resources are largely wasted. In business, the expenses are tax deductable as "cost of doing business".

The root cause of this phenomenon is the transaction programming paradigm's assumption in "reliable network". Without this assumption, the transaction programming paradigm falls apart. Assuming the network is reliable is the top fallacy (or the "cardinal sin" for mission critical service) in distributed computing [7]. It is responsible for all web2.0 infrastructure challenges.

In comparison, typical blockchain protocols have no device reliability assumptions. This results in higher service availability. But scalable performance remains a challenge.

2.2 ACAN and SMC

ACAN (Active Content Addressable Network) is a <key, value> overlay network over the physical interconnected networks. Unlike <key, value> data stores, the ACAN's Tuple Space abstraction enables arbitrarily tagged data communication and synchronization at the same time. This allows automated data-parallel computing without explicit parallel instructions [3]. Since a service infrastructure would contain many interconnected programs and data points located and/or replicated on different physical or virtual machines, coarse-grain parallelisms naturally exist at different network levels.

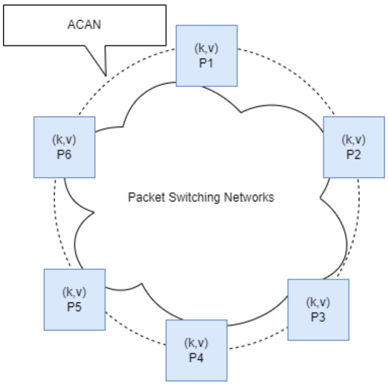


Fig. 2. Active Content Addressable Network.

In Fig. 2, each square represents a computer node holding some data in <key, value> format. ACAN is the overlay network in UVR (Unidirectional Virtual Ring) form. Programs running on a node can request and synchronize with any tuple with a matching <key> over the ACAN network.

UVR is an abstraction of the ACAN overlay network that can be implemented using Gossip protocols (as multiple parallel UVRs) for aggressive network propagation [4]. With ACAN's runtime, it forms a network-wide Tuple Space abstraction [5].

The <key, value> Tuple Space abstraction allows programs and data decoupled from physical computers and storage devices completely. Program and data may be distributed and replicated in any number of nodes retrievable via key pattern matching.

ACAN further allows the protocol runtime to enable statistical multiplexing of all available resources in real time (SMC). The SMC runtime can form dynamic SIMD (single instruction multiple data), MIMD (multiple instructions multiple data) and pipeline [6] processing clusters on the global overlay network. Fig.3 illustrates the three parallel topologies and potential speedups.

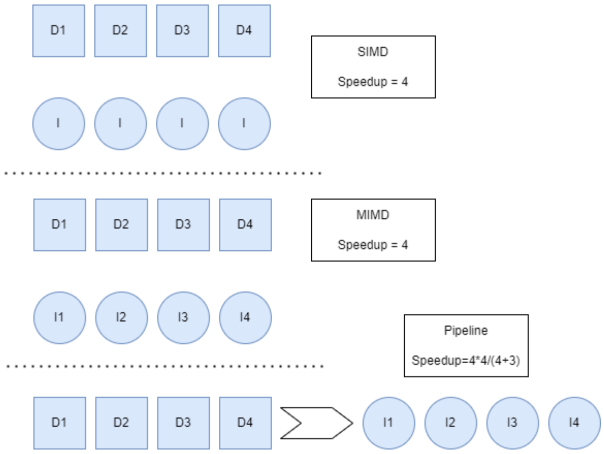


Fig. 3. Parallel Processing Topologies.

Automated parallel processing affords fault tolerance and load balancing automatically if the applications include a simple timeout-retransmission discipline. Fig. 4 illustrates the differences between web2.0 cloud computing and SMC enabled web3 computing where Cloud Services 1-3 are identical stateless services. The absence of the "retransmission discipline" and runtime program/data bindings to physical and virtual components are the direct result of assuming the network is reliable – the top fallacy of distributed computing [7].

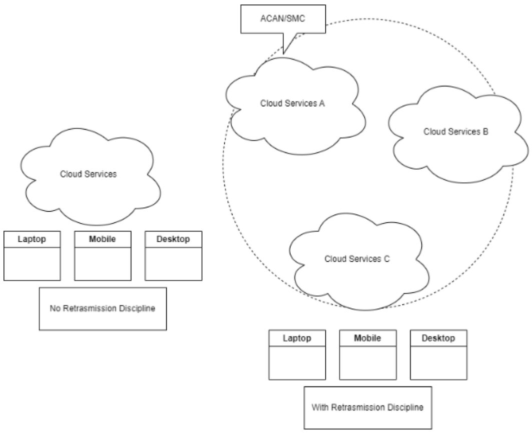


Fig. 4. Cloud and SMC Enabled Cloud Services.

The Blockchain and ACAN protocols do not have any device reliability assumptions. ACAN with SMC runtime can ultimately resolve the performance and reliability limitations for all service infrastructures.

3 Scalability Proof

The scalability of a distributed and parallel system is difficult to articulate and prove quantitatively. Fortunately, we have Amdahl's Law – an intuitive thought induced mathematical model to quantify the ultimate scalable performance for any parallel system.

Amdahl's Law was proposed in 1967 to assess the performance scalability of arbitrary programs on any parallel computer [8]. It defines $0 < x < 1$ as the percentage of serial instructions in the program. Given an infinite number of processors, its performance is above bounded to $\frac{1}{x}$. It is, however, not clear how x could be quantified for practical applications.

In 1988, Gustafson's Law [9] was proposed to invalidate Amdahl's since practical parallel experiments without sequential timing reference could not be quantified using Amdahl's Law. Gustafson's Law relies on actual time measures of parallel runs to derive the potential speedups. It seems to indicate infinite speedup if the program keeps on solving bigger problems. For the past three decades, these two laws have been referred to as "scaled speedup" and "non-scaled speedup." Although Amdahl's Law has a mathematical speedup bound, Gustafson's Law is an open-ended recurrence relation that does not have any mathematical bound.

A careful study has found that Amdahl's Law is the only stand-alone law while Gustafson's formula helps to quantify Amdahl's speedup bound inductively by solving bigger problems [10]. Fig. 5 illustrates the full power of Amdahl's Law.

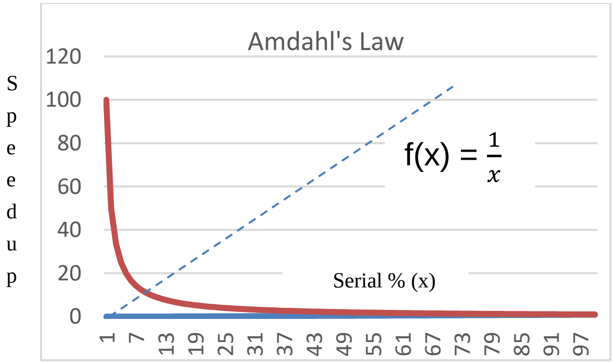


Fig. 5. Parallel Performance Scalability.

For programs solving problems of fixed sizes, given an infinite number of processors, the lower-right 45-degree zone covers all possible speedup limits including zero. This region quantifies the economic law of diminishing return. For programs solving problems of increasing sizes using more processors, the serial percentage will asymptotically approach zero. Thus, the upper-top 45-degree zone quantifies the asymptotic infinite speedup, when the problem size and the number of processors approaches infinity. This proof should not be a surprise. Practitioners have already put it in use, such as the TOP500 Supercomputer benchmark Linpack [24]. This is the international contest of supercomputers competing to solve the biggest Linpack problem sizes (https://top500.org).

Although neither Amdahl’s Law nor Gustafson’s Law account for communication and synchronization overheads, if the parallel acceleration exceeds the worst-case overheads, infinite speedup is still achievable as demonstrated in the TOP500 benchmark competitions.

Since infinity is a concept that is larger than any actual numbers, any parallel program has the potential to harness multiple quantum computers, if the reliability and security of the service can also improve when the system upscales.

4 ACAN for Blockchain

All blockchain protocols support “double-spent free” token transactions. For efficiency, token transactions are grouped into blocks to be verified by decentralized anonymous validators. In the blockchain overlay network, the basic unit of transmission is a **transaction**. Currently, the transactions are broadcast into the network as proposed in Satoshi’s original white paper [11] and implemented in production chains. Block verification methods vary depending on the consensus design, such as POW (proof of work), POS (proof of stake), POH (proof of history) and POP (proof of possession), etc [12].

Existing consensus protocols ignored runtime parallel processing. Blockchain sharding was added as an “after thought”. Existing sharding implementations delivered only marginal performance improvements with added protocol complexity and security challenges [13].

Introducing ACAN as the middle layer between TCP/IP delivered blocks and the consensus layer allows greater freedom in runtime parallelism exploitation solving the service scalability and energy efficiency challenges with incrementally better security and without increasing protocol complexity.

5 SMC Runtime for Blockchain

The presence of ACAN protocol allow the runtime implementations to take full advantage of available resources at hand. For blockchain applications, the runtime SMC implementation can form SIMD, MIMD and pipeline clusters automatically without explicit parallel coding. This is the advantage of dataflow parallel computing architecture [14].

Fig. 6 shows the Tendermint (deterministic consensus protocol) runtime state [15].

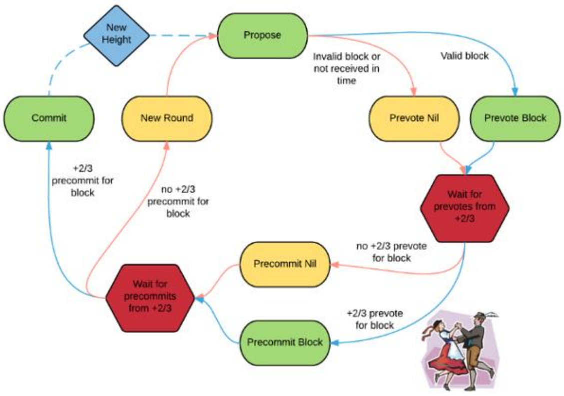


Fig. 6. Deterministic Consensus Protocol Runtime States.

Fig 7. illustrates the SMC blockchain POS consensus protocol runtime states. The data parallel runtime states are more complex than that in Fig. 6. The runtime states are created by the data parallel semantics automatically. Protocol coding is far simpler than the deterministic protocol illustrated in Fig. 6.

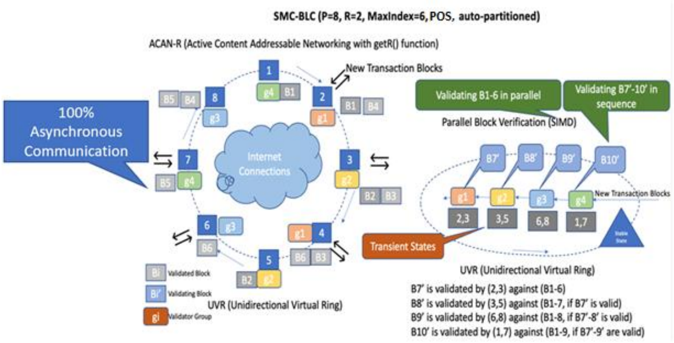


Fig. 7. SMC Blockchain consensus runtime states.

Theoretically, assuming there are an infinite number of transactions to be processed, and there are an infinite number of validating nodes in the network, the overall system's performance will asymptotically approach infinity. Since ACAN with SMC runtime are layered between the TCP/IP packets and the consensus protocol, netework security and transaction tamper resistance features continue to enhance as the network expands.

This proposal differs materially from the "Indefinite Scalability for Living Computation" [25] where reliability and security are completely absent in the design.

6 TOI CHAIN

TOIChain is an experimental network using POS (proof of stake) consensus. All nodes can become "stakers" by depositing tokens. The deposited tokens are locked in for a minimum time specified by the protocol. TOIChain maintains a group of "seed nodes" serving as the custom DNS for the chain in addition to the public DNS services. The consensus protocol requires greater than 66% of the randomly selected stakers approval of a transaction block before it can be included on the chain. ACAN and SMC are responsible for delivering the promised scalability benefits. In addition, TOIChain's script engine ensures transaction ACID properties for smart contracts. The TOIChain was designed to connect to any number of nodes globally. Mobile devices may also be stakers.

The TOIChain token is called **"TOIN"** which is represented as **UTXO** (Unspent Transaction Output) in the TOIChain network. Bitcoin 24.0.1 was customized to include ACAN and SMC concepts under the POS consensus protocol for scalability. Transactions are replicated on randomly selected R>7 nodes to deliver greater storage scalability and energy efficiency than any other chains.

The TOChain POS protocol is driven by a network Epoch initialized by seed nodes. Each new Epoch validates one block by randomly selected validators. An Epoch may be abandoned if the corresponding block verification fails. Otherwise, the verified block becomes the tip of the global ledger. The validated block is broadcast to the network to create replicas on other R random selected nodes. A new Epoch will be generated after this. If for some reason the network stalls, that is when the entire randomly selected validators crash at the same time or the network partitions, the seed nodes (TOIChain DNS) will generate a new Epoch based on the last witnessed Epoch. For Byzantine fault tolerance, each validation committee will be rewarded only in the next successful Epoch.

Unlike the Bitcoin network, TOIChain transactions are not replicated first. They are sent to the staking nodes for processing. All transactions in the selected staking nodes (validators) are included in the current block.

The selected validators form their own overlay network at runtime. The 66% or more members approval decides the eligibility of all transactions in the block. The approved block is propagated across the network for final validation, if approved it will become the current chain tip. A new Epoch will be generated.

TOI daemon code will be Open Source under the MIT License.

To deliver a Web 2.0-like user experience, TOIChain also supports **Smart Account** for users. A smart account can hold any number of wallets and all transaction histories (including the failed transactions). TOIChain accounts are not maintained on chain but managed by decentralized TOIChain Gateways.

The TOIChain Gateway supports non-custodial accounts (for public chains) and custodial accounts (for private chains). The TOI Gateway forms its own overlay network implemented in ACAN and SMC protocols. Therefore, it can also scale indefinitely (U.S. and international patent pending). The TOI Gateway code base is not Open Source.

Staking Policy:

Every node can request to be a validator by committing a minimum **TOINs** specified by the protocol. Higher values can increase the chance of being selected. Once approved, the node becomes a "staker". Stakers are organized in tiers based on their staking values. As mentioned earlier, each Epoch randomly selects R validators for the current block. The selection criteria include stake tiers, uptime, and locality. A region with many stakers enjoys higher verification speed as opposed to a region that may rely on remote stakers. A staker with a high downtime record is less likely to be selected.

Each verified block **earns TOINs** which are distributed to all selected and available stakers. Stakers receive both transaction verification and data hosting credits.

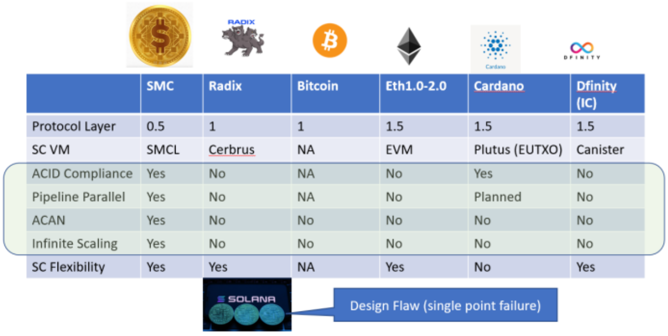
7 Smart Contracts, dApps and Tamper Resistance

SMCL is the smart contract language for TOIChain. Turing-completeness is a language design objective to support all possible calculations expressed in the language including loops. However, for smart contracts on any blockchain network, each transaction process requires network resources for validation. Contracts with infinite loops are not acceptable because they will incur infinite costs thus exhausting network resources. SMCL is intentionally not Turing-complete. Each contract is self-contained. However, it was designed to be extensible for future new use cases. We chose Python as the host language for ease of use in creating contracts. The TOIChain Python library has an option to translate smart contracts into Haskell language. The Haskell programs can be verified by Coq (Software Foundations) or similar tools for correctness before generating bytecode to execute on the TOIChain. The Turing incompleteness can prevent unwanted hacks.

SMCL scripting engine is ACID (atomicity, consistency, isolation, and durability) compliant. Therefore, the verified contracts should never yield unintended results regardless of concurrent executions on randomly selected validators. In other words, they are hack proof with exact accountability ensured.

According to De.fi database tracking [16], cryptocurrency hacks have reached $78 Billion with only $6 Billion recovered. Except for "rug pulls" (Ponzi schemes), the most hacks are through smart contracts. Ethereum smart contract using Turing-complete Solidity language has the dominating 30% share of successful hacks.

dApps are applications using the decentralized blockchain for transactions. The blockchain serves the finality of all transactions including NFTs (non-fungible tokens) or NFT with expirations. The infinite scalable feature of TOIChain makes it ideal to store any digital assets in perpetuity. There can be at least three types of dApps: Cloud and Quantum computer applications, AI applications with LLM training needs, and traditional dApps running their own tokens for different applications. Fig. 8 shows the technology comparisons of popular blockchain protocols.



	SMC	Radix	Bitcoin	Eth1.0-2.0	Cardano	Dfinity (IC)
Protocol Layer	0.5	1	1	1.5	1.5	1.5
SC VM	SMCL	Cerbrus	NA	EVM	Plutus (EUTXO)	Canister
ACID Compliance	Yes	No	NA	No	Yes	No
Pipeline Parallel	Yes	No	NA	No	Planned	No
ACAN	Yes	No	No	No	No	No
Infinite Scaling	Yes	No	No	No	No	No
SC Flexibility	Yes	Yes	NA	Yes	No	Yes

Fig. 8. Technology Comparisons.

8 Energy Efficiency and Self-optimization

POS chain experiments have demonstrated more than 99% of energy savings against POW consensus chains. However, asynchronous communications between hardware components can fail unexpectedly. The overall network security, degree of decentralization, fault tolerance and performance will deteriorate over time, especially after or during natural or human disasters or wars.

Due to the complete program/data decoupling, the TOIChain network contains self-optimizing functions that will automatically fine tune the R-value, resynchronizing lost data objects and adjust random validator selection criterions without service interruptions.

TOIChain's performance is also tunable by changing block size, replication factor R and validator committee size to find the sweet spot that balances performance, security and fault tolerance. For example, private chains do not need large validation committees (minimal 7 for Byzantine fault tolerance) and R can also be altered to meet community budget requirements and to ensure continued acceleration of transaction processing speeds. The value R decides the degree of parallel transaction pre-validation and the degree of replication after validation. It solves performance and capacity scalability problems at the same time. Therefore, TOIChain POS consensus can be dramatically more energy efficient than other POS protocols. The network adjustments can be completely automated without stopping services.

9 Broader Impacts

ACAN and SMC concepts can enable service infrastructures to harness all resources with self-optimizing features to deliver scalable performance, reliability, and security. They are applicable to all mission critical services. These include global DNS, databases, networked storage, self-driving cars, nuclear power plant monitoring, deep space probe controllers and real time machine learning services. Fig. 9 shows the industry sectors of the ACAN/SMC potential technology impacts.

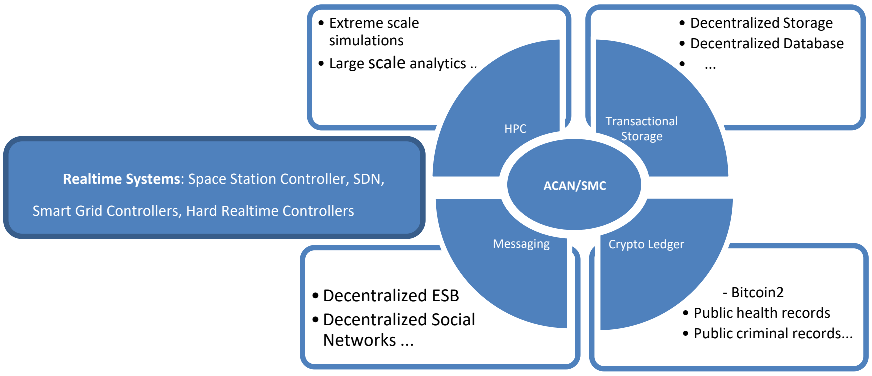


Fig. 9. Broader impact map.

10 Summary

The ACAN and SMC technologies are rooted in early 1990's Stateless Parallel Computing research [17] based on then mission critical service infrastructure requirements. These requirements remain critical after decades of technological evolutions. Decentralized mission critical database research started in 2001 [18] after the 9/11 terrorist attacks. The relationship between Amdahl's Law and Gustafson's Law was discovered in 1996 [10]. The infinite scalability proof materialized in 2012. In 2022, an EU patent was issued to ACAN/SMC design [19]. A US Patent was granted in February 2023

[20]. SMC blockchain provisional patent was filed in 2023. SMC Gateway patent was filed in early 2024.

The ACAN/SMC technology development history demonstrates that the early discoveries provided foresight followed by the computing, networking, and storage technology development trajectory. The first principle based research method worked well in developing solutions solving real world problems.

Today, there are insatiable computing needs for better solutions and innovations in all fields, especially the thriving AI and quantum computing sectors. The use of cybertechnologies to support mission critical requirements has become increasingly more important in all industries. The proposed ACAN/SMC programming paradigm offers a solid foundation for continued cyberinfrastructure technology development.

Acknowledgments. This paper benefited from anonymous reviewers' comments and suggestions on the earlier drafts.

References.

1. Hayden Mitman and Marc Levy, "PA Courts' Website Hit by Disabling Cyberattack, Official Says," NBC Philadelphia, Retrieved 2/10/2024: https://www.nbcphiladelphia.com/news/tech/pa-statewide-court-agencys-website-hit-by-disabling-cyberattack-officials-say/3766722/
2. Joseph Bonneau, "Why Blockchain Performance is Hard to Measure," Retrieved 2/10/2024: https://a16zcrypto.com/posts/article/why-blockchain-performance-is-hard-to-measure/
3. David Loshin, "Data Parallelism," ScienceDirect, Business Intelligence (2nd Edition), 2013: https://www.sciencedirect.com/topics/computer-science/data-parallelism
4. Prateek Gupta, "Gossip Protocol in Distributed Systems," Medium, 4/23/2023: https://medium.com/nerd-for-tech/gossip-protocol-in-distributed-systems-e2b0665c7135
5. J. Krishnamurthy, M. Maheswaran, in Internet of Things, 2016, ScienceDirect: https://www.sciencedirect.com/topics/computer-science/tuple-space
6. Michael Flynn, "Flynn's Taxonomy," Encyclopedia of Parallel Computing, pp 689-697, SpringerLink: https://link.springer.com/referenceworkentry/10.1007/978-0-387-09766-4_2
7. "Fallacies of Distributed Computing," Wikipedia.com, Retrieved 2/10/2024: https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing
8. Gene M. Amdahl, "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities," AFIPS Conference LNCS Homepage, http://www.springer.com/lncs, last accessed 2023/10/25 Proceedings (30): 483-485, *doi:10.1145/1465482.1465560. S2CID 195607370.*
9. *John L. Gustafson, "Reevaluating Amdahl's Law," Communications of ACM, 32(5): 532-533, doi:10.1145/42411.42415. S2CID 33937392.*
10. Yuan Shi, "Reevaluating Amdahl's Law and Gustafson's Law," 1996: https://cis.temple.edu/~shi/wwwroot/shi/public_html/docs/amdahl/amdahl.html
11. Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Retrieved 2/10/2024: https://bitcoin.org/bitcoin.pdf

12. Jie Xu, Cong Wang, Xiaohua Jia, “A Survey of Blockchain Consensus Protocols,” ACM Computing Surveys, 55, 13s, Article No.: 278, pp 1-35, https://doi.org/10.1145/3579845
13. V. K, J. R, G. Kommineni, M. Tanna and G. Prerna, "Sharding in Blockchain Systems: Concepts and Challenges," 2022 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Bangalore, India, 2022, pp. 1-7, doi: 10.1109/SMARTGENCON56628.2022.10083582. keywords: {Sharding;Fault tolerance;Databases;Scalability;Fault tolerant systems;Blockchains;Security;distributed systems;blockchain technology;sharding;scalability;challenges in sharding}
14. “Dataflow Architecture,” Wikipedia.com, Retrieved 2/10/2024: https://en.wikipedia.org/wiki/Dataflow_architecture
15. “Tendermint – Building the Most Powerful Tools for Distributed Networks,” Retrieved 2/10/2024: https://tendermint.com/
16. “Top Crypto Hacks,” REKT Database, Retrieved 2/10/2024: https://de.fi/rekt-database
17. Justin Shi, “Multi-Computer System and Method,” U.S. Patent #5,517,656, 5/1996.
18. Justin Shi, “High Performance Lossless ESB Architecture with Data Protection for Mission-Critical Applications,” 2009 World Congress on Computer Science and Information Engineering, 2009.
19. Justin Shi, “Statistic Multiplexed Computing,” European Patent Office, 17869136.6 – 1213/3539261, 7/14/2022.
20. Justin Shi, “Statistic Multiplexed Computing System for Network-Scale Reliable High-Performance Services,” U.S. Patent Office, #US 11,589,926 B1, 2/21/2023.
21. Prashant Jha, “Solana Network Down: Validators Act to Restore Service,” Retrieved 2/12/2024: https://cointelegraph.com/news/solana-network-outage-block-production-halted
22. Jim Keller, “Cuda’s a swamp, not a moat. x86 is also a swamp too,” https://www.tomshardware.com/tech-industry/artificial-intelligence/jim-keller-criticizes-nvidias-cuda-and-x86-cudas-a-swamp-not-a-moat-x86-was-a-swamp-too
23. Open AI, Research, “Introducing Triton: Open Source GPU Programming for Neural Networks,” https://openai.com/research/triton
24. Jack Dongarra, Cleve Moler, James Bunch, W Gilbert, “LINPACK User’s Guide,” SIAM, 1979, ISBN: 9780898711721 https://en.wikipedia.org/wiki/Special:BookSources/9780898711721
25. David H. Ackley, “Indefinite Scalability for Living Computation,” Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), AAAI-16 Senior Member Presentation Track “Blue Sky Ideas” talk, 2016.
26. NASA/JPL-Caltech, “Discovery Alert: A Super-Earth in the Habitable Zone,” Retrieved 7/7/2024: https://science.nasa.gov/universe/exoplanets/discovery-alert-a-super-earth-in-the-habitable-zone/