Development of Peer-to-Peer Solar Energy Trading Simulation System using Solana Smart Contract

(Anchor Framework Permissioned Environments)

Mr.Chanthawat Kiriyadee
Faculty of Engineering
Computer Engineering and Artificial Intelligence
The University of the Thai Chamber of Commerce
Bangkok, Thailand
2410717302003@live4.utcc.ac.th

***Abstract*—This paper presents a comprehensive performance evaluation of GridTokenX, a decentralized Peer-to-Peer (P2P) energy trading platform built on a private Solana cluster. We employ the BLOCKBENCH methodology (SIGMOD 2017) for systematic layer-by-layer analysis combined with TPC-C benchmark adaptation for realistic workload evaluation. Our experimental results demonstrate that the platform achieves 2,111 tpmC (transactions per minute Type C) with a mean latency of 116ms, identifying a "Trust Premium" of 58.28x compared to centralized baselines. BLOCKBENCH micro-benchmarks reveal consensus layer throughput of 225 TPS, execution layer at 231 TPS, and data model layer at 192 TPS. The YCSB workload analysis shows 442 ops/s for read-heavy workloads. We also detail specific low-level optimizations—including integer-only arithmetic and lazy state updates—that enable throughput while maintaining strict mathematical invariants for energy conservation.**

***Index Terms*—Blockchain, Energy Trading, Solana, TPC-C, BLOCKBENCH, YCSB, Performance Benchmarking, Smart Contracts**

I. Introduction

The decentralization of energy systems through Distributed Energy Resources (DERs) necessitates robust trading infrastructures capable of handling high-frequency micro-transactions [2]. Traditional centralized utility models suffer from single-point-of-failure risks and lack transparency in pricing. Blockchain technology offers a solution but is often criticized for scalability limitations [4].

This research evaluates GridTokenX, which leverages Solana's Sealevel parallel runtime [3] to overcome these bottlenecks. The objectives of this study are:

1) To study and present the architecture of a P2P energy trading simulation system using Solana (Anchor) in a Permissioned (PoA) environment.
2) To develop and prove the concept (Proof-of-Concept) of a prototype system capable of simulating GRID Token exchange using an AMI Simulator.
3) To evaluate and analyze the performance using BLOCKBENCH layer analysis and TPC benchmark suite.

II. Related Work: BLOCKBENCH Methodology

BLOCKBENCH [4] provides a systematic framework for evaluating private blockchain performance across four architectural layers:

1) **Consensus Layer**: Measures pure consensus overhead through DoNothing operations.
2) **Execution Layer**: Evaluates smart contract computation via CPU-intensive workloads.
3) **Data Model Layer**: Tests state read/write performance using IOHeavy benchmarks.
4) **Application Layer**: Real-world workload simulation using YCSB and Smallbank.

We extend the BLOCKBENCH methodology to Solana/Anchor with native Rust implementations of each benchmark category.

III. System Architecture

The platform is implemented as a suite of five interconnected Anchor smart contracts on Solana: 1. **Registry**: Manages user identity and meter assets. 2. **Oracle**: Validates off-chain sensor data. 3. **Energy Token**: SPL-compliant token representing 1 kWh. 4. **Trading**: Order book and settlement engine. 5. **Governance**: Configuration and DAO parameters.

IV. Methodology

A. TPC-C Mapping

We map standard TPC-C transactions [1] to Energy Trading equivalents to create a realistic workload profile:

TABLE I
TPC-C to Energy Trading Mapping

TPC-C Transaction	Mix	GridTokenX Function
New Order	45%	`create_sell_order` / `create_buy_order`
Payment	43%	`transfer_tokens` (Settlement)
Order Status	4%	`get_order_status` (RPC Read)
Delivery	4%	`match_orders` (Batch Execution)
Stock Level	4%	`get_balance` (Energy Audit)

B. Mathematical Models

1) VWAP Pricing Mechanism: To ensure fair market value, the platform employs a Volume-Weighted Average Price (VWAP) discovery algorithm. The clearing price $P_{clearing}$ is calculated dynamically:

$$P_{base} = \frac{P_{bid} + P_{ask}}{2} \tag{1}$$

$$P_{clearing} = P_{base} + \left(P_{base} \times \min\left(\frac{V_{trade}}{V_{total}}, 1.0\right) \times \delta_{max}\right) \tag{2}$$

Where V_{trade} is the current match volume, V_{total} is the historical volume, and δ_{max} is the maximum price elasticity factor (10%).

2) Token Conservation Invariant: The system enforces a strict conservation of energy. Tokens ($\Delta Supply_{GRID}$) can only be minted when physically generated energy is mathematically settled:

$$\Delta Supply_{GRID} = \max(0, (E_{produced} - E_{consumed}) - E_{settled}) \tag{3}$$

3) System Optimizations: To maximize throughput within the Solana Compute Unit (CU) limit, we implemented three critical optimizations:

A. Integer-Only Arithmetic: Floating-point operations (`f64`) are computationally expensive and discouraged in Solana programs. We replaced the VWAP calculation with fixed-point integer math:

$$W = \min\left(\frac{V \times 1000}{V_{total}}, 1000\right) \tag{4}$$

This reduction saved approximately 10,000 CUs per trade execution.

B. Lazy State Updates: Instead of serializing the full price history array (100+ entries) on every transaction, updates are "lazily" committed only when the price deviation exceeds 5% or every 60 seconds. This reduced the serialization overhead by 90%.

C. Batch Order Matching: The `match_orders` instruction was refactored to handle batch execution, allowing multiple non-overlapping limit orders to be settled in a single atomic transaction, significantly improving the "fills per second" rate.

V. Experimental Setup

Benchmarks were conducted on a high-performance Solana localnet cluster to eliminate internet latency variables.

- **Hardware**: Apple M-Series (8-core), 16GB RAM.
- **Cluster**: Solana Test Validator v1.18.
- **Client**: Multi-threaded Rust workload generator.

TABLE II
BLOCKBENCH Micro-benchmark Results

Layer	Benchmark	TPS	Latency
Consensus	DoNothing	225	2.5ms
Execution	CPUHeavy-Sort	231	2.5ms
Data Model	IOHeavy-Write	192	3.0ms
Data Model	IOHeavy-Mixed	192	3.0ms

VI. Performance Evaluation

A. BLOCKBENCH Layer Analysis

Following the BLOCKBENCH methodology, we evaluated each architectural layer independently:

The DoNothing benchmark measures pure consensus overhead (225 TPS), while CPUHeavy sorting operations achieve 231 TPS, demonstrating efficient smart contract execution. The IOHeavy benchmarks at 192 TPS show the overhead of state serialization.

B. YCSB Workload Results

We implemented three YCSB workload profiles to evaluate application-layer performance:

TABLE III
YCSB Workload Performance

Workload	Profile	ops/s	Latency
YCSB-A	50% read, 50% update	290	2.7ms
YCSB-B	95% read, 5% update	442	1.8ms
YCSB-C	100% read	391	2.1ms

The Smallbank OLTP benchmark achieved **1,714 TPS** with 5.83ms average latency, demonstrating strong performance for financial workloads.

C. TPC-C Throughput Analysis

Our TPC-C benchmark implementation achieved a throughput of **2,111 tpmC** (76.85 TPS) in simulation mode. This throughput is directly attributed to the *Sealevel* parallel runtime, which allows non-overlapping transactions (e.g., trades between different pairs of users) to execute simultaneously. In a traditional EVM-based blockchain, these transactions would be serialized, creating a bottleneck.

In our benchmark run, 4,621 total transactions were processed with 4,611 successful (**99.78% success rate**). The transaction mix followed TPC-C specification:

- New-Order: 2,074 transactions (44.9%) – 99.71% success
- Payment: 2,004 transactions (43.4%) – 99.80% success
- Order-Status: 197 transactions (4.3%) – 100% success
- Delivery: 175 transactions (3.8%) – 100% success
- Stock-Level: 171 transactions (3.7%) – 100% success

For a regional P2P market with 10,000 households, assuming each smart meter submits a reading every 15 minutes, the required throughput is approximately 11 TPS. The observed 76.85 TPS provides a **7x safety margin**, proving the architecture can handle neighborhood-scale deployments without congestion.

D. TPC-E and TPC-H Results

We extended our evaluation to include TPC-E (brokerage workload) and TPC-H (decision support) benchmarks:

TABLE IV
TPC-E/TPC-H Benchmark Results

Benchmark	Primary Metric	Avg Latency	p99
TPC-E	306 tpsE	7.89 ms	17 ms
TPC-H	250,486 QphH	71.0 ms	147 ms

The TPC-E benchmark achieved **306 tpsE** (trades per second), demonstrating strong performance for brokerage-style workloads with complex order matching. TPC-H analytical queries achieved **250,486 QphH** (queries per hour), validating the platform's capability for reporting and auditing functions.

E. Latency Analysis

Transaction latency differs significantly from "confirmation time" in this permissioned PoA environment.

1) **Mean Latency (116.56ms)**: Represents the average time for the leader node to process the instruction and update the in-memory state.
2) **p50 Latency (112.57ms)**: Median transaction processing time.
3) **p95 Latency (180.04ms)**: 95th percentile under normal load.
4) **p99 Latency (215.54ms)**: Represents the worst-case processing time under heavy load.

The simulation mode includes artificial delays to model realistic blockchain behavior. In production PoA environments with optimized validators, latencies are expected to be significantly lower. Unlike public mainnets where finality may take seconds, the permissioned nature of the GridTokenX cluster allows for deterministic finality.

F. Concurrency Analysis

Under high contention (multiple orders against the same market state), the Sealevel runtime effectively linearized conflicting transactions while processing non-overlapping requests in parallel. The observed MVCC (Multi-Version Concurrency Control) conflict rate was **1.45%**, largely due to the atomic nature of the `match_orders` instruction. The average retry count was 0.02, indicating efficient conflict resolution.

G. Scalability Analysis

We conducted scalability tests across three dimensions:

TABLE V
Scalability Test Results

Test	TPS	Latency	Stability
1 thread (baseline)	443	2.26ms	100%
32 concurrent threads	398	2.51ms	90% retained
60s sustained load	416	2.40ms	Stable
1,000 accounts	220	4.54ms	Linear degrad.

TABLE VI
Platform Performance Comparison

Platform	Benchmark	TPS	Latency	Source
GridTokenX	Smallbank	1,714	5.8ms	This Study
GridTokenX	DoNothing	225	2.5ms	This Study
GridTokenX	YCSB-B	442	1.8ms	This Study
Hyperledger 2.0	Smallbank	400	150ms	BLOCKBENCH
Hyperledger 2.0	DoNothing	3,500	45ms	BLOCKBENCH
Hyperledger 2.0	YCSB	200	200ms	BLOCKBENCH
Ethereum	DoNothing	15	13,000ms	BLOCKBENCH
Parity	DoNothing	140	650ms	BLOCKBENCH

H. Cross-Platform Comparison

We compare GridTokenX performance against published results from other blockchain platforms:

While Hyperledger Fabric achieves higher DoNothing throughput (3,500 vs 225 TPS) due to its execute-order-validate architecture, GridTokenX demonstrates **4.3x better Smallbank performance** (1,714 vs 400 TPS) and **26x lower latency** (5.8ms vs 150ms) for financial workloads.

I. The "Trust Premium"

We define "Trust Premium" as the performance cost incurred to achieve distributed consensus in a private network compared to a centralized baseline (PostgreSQL).

$$Trust\ Premium = \frac{Latency_{Blockchain}}{Latency_{Baseline}} = \frac{116.56ms}{2.00ms} \approx 58.28x \tag{5}$$

While this represents a latency overhead compared to centralized databases, it is acceptable for energy trading applications where settlement typically takes days or weeks in traditional systems. The trade-off provides:

- Immutable transaction audit trail
- Automated smart contract settlement
- Transparent pricing mechanisms
- Elimination of single-point-of-failure risks

J. Benchmark Results Summary

Table VII summarizes the key performance metrics from our comprehensive benchmark evaluation.

VII. Cross-Platform Evaluation Framework

To strictly compare GridTokenX with other blockchain platforms (e.g., Ethereum, Hyperledger Fabric) [4], we propose the *Standardized Energy Trading Workload (SETW)*. Researchers must follow three steps to replicate this methodology:

TABLE VII
COMPREHENSIVE BENCHMARK RESULTS SUMMARY

Metric	Value
BLOCKBENCH Micro-benchmarks	
DoNothing (Consensus)	225 TPS
CPUHeavy (Execution)	231 TPS
IOHeavy (Data Model)	192 TPS
YCSB/Smallbank	
YCSB-B (95% read)	442 ops/s
Smallbank OLTP	1,714 TPS
TPC Benchmark Suite	
TPC-C (tpmC)	2,111
TPC-E (tpsE)	306
TPC-H (QphH)	250,486
Performance Metrics	
Success Rate	99.78%
Mean Latency	116.56 ms
p99 Latency	215.54 ms
MVCC Conflict Rate	1.45%
Trust Premium	58.28x

A. Equivalent Contract Implementation

Target platforms must implement the core logic defined in Table I. Specifically, the "Order Match" function must be atomic, ensuring that the trade execution and token settlement occur in the same cryptographic state transition.

B. Workload Injection Profile

The load generator must maintain the specific TPC-C mix:

- **Write Heavy (88%)**: New Orders + Payments. This stresses the consensus engine's state contention.
- **Read Light (12%)**: Status checks. This tests the RPC query performance.

C. Metric Normalization

Results must be reported in **tpmC** (transactions per minute Type-C). For blockchains with probabilistic finality (e.g., PoW), latency must include the time to reach k-block confirmation depth to be comparable with Solana's deterministic leader schedule.

VIII. CONCLUSION

GridTokenX demonstrates that private blockchain technology is viable for P2P energy trading in regional deployments. Through comprehensive BLOCKBENCH layer analysis and TPC benchmark evaluation:

- **Consensus Layer**: 225 TPS DoNothing throughput
- **Execution Layer**: 231 TPS CPUHeavy computation
- **Data Model**: 192 TPS IOHeavy state operations
- **Application Layer**: 1,714 TPS Smallbank, 442 ops/s YCSB-B
- **TPC Suite**: 2,111 tpmC, 306 tpsE, 250,486 QphH

Compared to Hyperledger Fabric, GridTokenX achieves **4.3x higher Smallbank throughput** and **26x lower latency** for financial workloads. The observed **Trust Premium of 58.28x** represents the acceptable cost for achieving immutable audit trails, automated smart contract settlement, and transparent pricing mechanisms.

Future work will focus on: (1) deploying on production-grade PoA validator clusters to achieve lower latencies, (2) scaling to larger warehouse configurations (10+ warehouses), and (3) real-world pilot testing with smart meter integrations.

REFERENCES

[1] Transaction Processing Performance Council (TPC), "TPC Benchmark C Standard Specification, Revision 5.11," 2010.

[2] E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, and C. Weinhardt, "Designing microgrid energy markets: A case study: The Brooklyn Microgrid," *Applied Energy*, vol. 210, pp. 870–880, 2018.

[3] A. Yakovenko, "Solana: A new architecture for a high performance blockchain v0.8.13," Solana Labs Whitepaper, 2018.

[4] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K. L. Tan, "Blockbench: A framework for analyzing private blockchains," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, pp. 1085–1100, 2017.

[5] M. Andoni *et al.*, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 143–174, 2019.

[6] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," *IEEE Trans. Industrial Informatics*, vol. 14, no. 8, pp. 3690–3700, 2018.

[7] J. Guerrero, A. C. Chapman, and G. Verbic, "Decentralized P2P energy trading under network constraints in a low-voltage network," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5163–5173, 2019.

[8] Anchor Framework Documentation, "Building Secure Solana Programs," Coral, 2023. [Online]. Available: https://anchor-lang.com