

การพัฒนาระบบจำลองการซื้อขายพลังงานแสงอาทิตย์แบบ Peer-to-Peer โดยใช้ Solana Smart Contract

(Anchor Framework ในสภาพแวดล้อมแบบ Permissioned)

นายจันทร์ พิริยาดี
คณบดีวิศวกรรมศาสตร์
สาขาวิศวกรรมคอมพิวเตอร์และปัญญาประดิษฐ์
มหาวิทยาลัยหอการค้าไทย
กรุงเทพมหานคร ประเทศไทย
2410717302003@live4.utcc.ac.th

บทคัดย่อ—บทความนี้ นำเสนอ การประเมิน ประสิทธิภาพ อายุการ ครอบคลุม ของ GridTokenX ซึ่งเป็นแพลตฟอร์มการซื้อขายพลังงานแบบ Peer-to-Peer (P2P) บน กระจายศูนย์ที่สร้างบน Solana cluster แบบส่วนตัว เราใช้วิธีการ BLOCKBENCH (SIGMOD 2017) สำหรับการวิเคราะห์แบบชั้นต่อชั้นอย่างเป็นระบบ ร่วมกับการตัดแปลง TPC-C benchmark สำหรับการประเมินภาระงานจริง ผลการทดลองแสดงให้เห็นว่าแพลตฟอร์มสามารถทำได้ 2,111 tpmC (transactions per minute Type C) โดยมีค่าเฉลี่ยความหน่วง 116ms และระบุ "Trust Premium" ที่ 58.28 เท่า เมื่อเทียบกับระบบรวมศูนย์ BLOCKBENCH micro-benchmarks แสดงปริมาณงานของ consensus layer ที่ 225 TPS, execution layer ที่ 231 TPS และ data model layer ที่ 192 TPS การวิเคราะห์ YCSB workload แสดง 442 ops/s สำหรับภาระงานที่อ่อนน้ำก นอกเหนือไปยังอิทธิพลการปรับปรุงระดับต่ำโดยเฉพาะ รวมถึงการคำนวณแบบจำนวนเพิ่มเท่านั้นและการอัปเดตสถานะแบบ lazy ที่ช่วยเพิ่มปริมาณงานขณะรักษาค่าคงที่ทางคณิตศาสตร์อย่างแม่นใจสำหรับการอนุรักษ์พลังงาน

Index Terms—บล็อกเชน, การซื้อขายพลังงาน, Solana, TPC-C, BLOCKBENCH, YCSB, การทดสอบประสิทธิภาพ, Smart Contracts

I. บทนำ

การจะนำ ยศูนย์ ของระบบ พลังงาน ผ่าน ทรัพยากร พลังงาน แบบ กระจาย (Distributed Energy Resources หรือ DERs) ต้องการโครงสร้างพื้นฐานการซื้อขายที่แข็งแกร่ง สามารถจัดการธุรกรรมขนาดเล็กความถี่สูงได้ [2] รูปแบบสารสนเทศแบบรวมศูนย์แบบตั้งเติมประสบปัญหาความเสี่ยงจุดล้มเหลวเดียว และขาดความโปร่งใสในการกำหนดราคา เทคโนโลยีลือกโซนเสนอทางออกแต่ มักถูกวิพากษ์วิจารณ์เรื่องข้อจำกัดด้านความสามารถในการขยายตัว [4]

งานวิจัยนี้ประเมิน GridTokenX ซึ่งใช้ประโยชน์จาก Sealevel parallel runtime ของ Solana [3] เพื่ออาจนน化ลดเวลา วัตถุประสงค์ของการศึกษา นี้คือ:

- เพื่อศึกษาและนำเสนอสถาปัตยกรรมของระบบจำลองการซื้อขายพลังงาน P2P โดยใช้ Solana (Anchor) ในสภาพแวดล้อม Permissioned (PoA)
- เพื่อพัฒนาและพิสูจน์แนวคิด (Proof-of-Concept) ของระบบต้นแบบที่สามารถจำลองการแลกเปลี่ยน GRID Token โดยใช้ AMI Simulator
- เพื่อประเมิน และ วิเคราะห์ ประสิทธิภาพ โดยใช้ การ วิเคราะห์ ชั้น BLOCKBENCH และชุด TPC benchmark

II. งานวิจัยที่เกี่ยวข้อง: วิธีการ BLOCKBENCH

BLOCKBENCH [4] ให้กรอบการทำงานอย่างเป็นระบบสำหรับการประเมินประสิทธิภาพลือกโซนส่วนตัวผ่านสถาปัตยกรรม:

- Consensus Layer:** วัดค่าใช้จ่าย consensus บริสุทธิ์ผ่านการทำงาน DoNothing
- Execution Layer:** ประเมินการคำนวณ smart contract ผ่านภาระงานที่ใช้ CPU มาก
- Data Model Layer:** ทดสอบประสิทธิภาพการอ่าน/เขียนสถานะโดยใช้ IOHeavy benchmarks
- Application Layer:** การจำลองภาระงานจริง โดยใช้ YCSB และ Smallbank

เราขยายวิธีการ BLOCKBENCH ไปยัง Solana/Anchor ด้วยการใช้งาน Rust ดังเดิมของแต่ละหมวดหมู่ benchmark

III. สถาปัตยกรรมระบบ

แพลตฟอร์มถูกใช้งานเป็นชุดของห้า Anchor smart contracts ที่เชื่อมต่อกันบน Solana: 1. **Registry:** จัดการตัวตนผู้ใช้และทรัพย์สินมิเตอร์ 2. **Oracle:** ตรวจสอบข้อมูลเชื่อมต่อเครือข่าย 3. **Energy Token:** โทเคนที่เป็นไปตาม SPL แทน 1 kWh 4. **Trading:** order book และเครื่องมือการคำนวณราคา 5. **Governance:** การกำหนดค่าและพารามิเตอร์ DAO

IV. วิธีการ

A. การแปลง TPC-C

เราแปลงโปรแกรม TPC-C มาตรฐาน [1] กับการซื้อขายพลังงานที่เทียบเท่าเพื่อสร้างประโยชน์ของภาระงานที่สมจริง:

ตารางที่ 1:
การแปลง TPC-C กับการซื้อขายพลังงาน

ธุรกรรม TPC-C	สัดส่วน	ฟังก์ชัน GridTokenX
New Order	45%	create_sell_order / create_buy_order
Payment	43%	transfer_tokens (การชำระเงิน)
Order Status	4%	get_order_status (RPC Read)
Delivery	4%	match_orders (การทำงานแบบ Batch)
Stock Level	4%	get_balance (การตรวจสอบคงเหลือ)

B. แบบจำลองทางคณิตศาสตร์

1) กลไกการกำหนดราคา VWAP: เพื่อให้มั่นใจในมูลค่าตลาดที่เป็นธรรม แพลตฟอร์มใช้อัลกอริทึมการค้นหาราคาถาวรเฉลี่ยต่อวันหักปริมาณ (VWAP) ราคา clearing $P_{clearing}$ คำนวณแบบด้านนี้:

$$P_{base} = \frac{P_{bid} + P_{ask}}{2} \quad (1)$$

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

โดยที่ V_{trade} คือปริมาณการจับคู่ปัจจุบัน, V_{total} คือปริมาณในอดีต และ δ_{max} คือปัจจัยความยืดหยุ่นราคาน้ำหนัก (10%)

2) ค่าคงที่การอนุรักษ์ໂທเค็น: ระบบบังคับให้การอนุรักษ์พลังงานอย่างเข้มงวด ໂທเค็น ($\Delta SupplyGRID$) สามารถสร้างได้เฉพาะเมื่อพลังงานที่ผลิตทางภายนอกจำกัดทางคณิตศาสตร์:

$$\Delta SupplyGRID = \max(0, (E_{produced} - E_{consumed}) - E_{settled}) \quad (3)$$

3) การปรับปรุงระบบ: เพื่อเพิ่มปริมาณงานสูงสุดภายในขีดจำกัด Compute Unit (CU) ของ Solana เราใช้การปรับปรุงสำคัญสามประการ:

A. การคำนวณแบบจำนวนเต็มเท่านั้น: การดำเนินการจุดศูนย์ยืน (f64) มีค่าใช้จ่ายคำนวณสูงและไม่แนะนำในโปรแกรม Solana เราแทนที่การคำนวณ VWAP ด้วยคณิตศาสตร์ง่ายๆ ที่คำนวณเต็ม:

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

การลดน้ำหนักต้องการ CU 10,000 CU ต่อการทำธุรกรรม

B. การอัปเดตสถานะแบบ Lazy: แทนที่จะ serialize อาร์เรย์ประวัติราคาทั้งหมด (100+ รายการ) ทุกธุรกรรม การอัปเดตจะถูก "commit แบบ lazy" เฉพาะ เมื่อการเปลี่ยนราคาเกิน 5% หรือทุก 60 วินาที ซึ่งลดค่าใช้จ่าย serialization ลง 90%

C. การจับคู่คำสั่งแบบ Batch: คำสั่ง match_orders ถูก refactor เพื่อจัดการการทำงานแบบ batch ทำให้สามารถคำนวณ limit orders หลายรายการที่ไม่ต้องหักกันในธุรกรรมของตัวมีกิจเดียว ปรับปรุงอัตรา "fills per second" อย่างมีนัยสำคัญ

V. การตั้งค่าการทดลอง

Benchmarks ดำเนินการบน Solana localnet cluster ประสิทธิภาพสูงเพื่อจัดตัวแปรความหน่วงในเครือเน็ต

- ฮาร์ดแวร์:** Apple M-Series (8-core), 16GB RAM
- คลัสเตอร์:** Solana Test Validator v1.18
- ลูกค้า:** ตัวสร้างภาระงาน Rust แบบ multi-threaded

VI. การประเมินประสิทธิภาพ

A. การวิเคราะห์ชิ้น BLOCKBENCH

ตามวิธีการ BLOCKBENCH เราประเมินแต่ละชิ้นสถาปัตยกรรมอย่างอิสระ:

DoNothing benchmark วัดค่าใช้จ่าย consensus บริสุทธิ์ (225 TPS) ในขณะที่การดำเนินการ CPUHeavy sorting ทำได้ 231 TPS และการทำงาน

ตารางที่ II
ผลลัพธ์ BLOCKBENCH Micro-benchmark

ชิ้น	Benchmark	TPS	ความหน่วง
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

smart contract ที่มีประสิทธิภาพ IOHeavy benchmarks ที่ 192 TPS แสดงค่าใช้จ่ายของ state serialization

B. ผลลัพธ์ YCSB Workload

เราใช้สามโปรแกรม YCSB workload เพื่อประเมินประสิทธิภาพ application-layer:

ตารางที่ III
ประสิทธิภาพ YCSB Workload

Workload	โปรไฟล์	ops/s	ความหน่วง
YCSB-A	50% อ่าน, 50% อัปเดต	290	2.7ms
YCSB-B	95% อ่าน, 5% อัปเดต	442	1.8ms
YCSB-C	100% อ่าน	391	2.1ms

Smallbank OLTP benchmark ทำได้ 1,714 TPS โดยมีความหน่วงเฉลี่ย 5.83ms แสดงประสิทธิภาพที่แข็งแกร่งสำหรับภาระงานทางการเงิน

C. การวิเคราะห์บริการงาน TPC-C

การใช้งาน TPC-C benchmark ของเรามีความต้องการบริการงานนี้เกิดจาก Sealevel parallel runtime โดยตรง ซึ่งอนุญาตให้ธุรกรรมที่ไม่ต้องหักกัน (เช่น การซื้อขายระหว่างคู่ผู้ใช้ที่ต่างกัน) ทำงานพร้อมกัน ในแบบล็อกเชนแบบ EVM ดังเดิม ธุรกรรมเหล่านี้จะถูก serialize ทันทีให้เกิดความรวดเร็ว

ในการรัน benchmark ของเรา มีธุรกรรมทั้งหมด 4,621 รายการถูกประมวลผลโดย 4,611 รายการสำเร็จ (อัตราสำเร็จ 99.78%) ส่วนผิดพลาดเป็นไปตามข้อกำหนด TPC-C:

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

สำหรับตลาด P2P ระดับภูมิภาคที่มี 10,000 ครัวเรือน สมมติว่าสามารถทำได้ต่อไป แต่ละตัวส่งค่าทุก 15 นาที บริการงานที่ต้องการประมวลผล 11 TPS 76.85 TPS ที่สังเกตได้ให้ ส่วนเพิ่มความปลอดภัย 7 เท่า พิสูจน์ว่าสถาปัตยกรรมสามารถจัดการภาระติดตั้งระดับบ้านได้โดยไม่มีความแออัด

D. ผลลัพธ์ TPC-E และ TPC-H

รายละเอียดการประเมินเพื่อรวม TPC-E (ภาระงานนายหน้า) และ TPC-H (การสนับสนุนการตัดสินใจ) benchmarks:

TPC-E benchmark ทำได้ 306 tpsE (trades per second) และประสิทธิภาพที่แข็งแกร่งสำหรับภาระงานแบบนายหน้าที่มีการจับคู่คำสั่งซึ่งกัน

ตารางที่ IV
ผลลัพธ์ TPC-E/TPC-H Benchmark

Benchmark	เมตริกหลัก	ความหน่วงเฉลี่ย	p99
TPC-E	306 tpsE	7.89 ms	17 ms
TPC-H	250,486 QphH	71.0 ms	147 ms

TPC-H analytical queries ทำได้ 250,486 QphH (queries per hour) ยืนยันความสามารถของแพลตฟอร์มสำหรับฟังก์ชันการรายงานและการตรวจสอบ

E. การวิเคราะห์ความหน่วง

ความหน่วงธุรกรรมแตกต่างอย่างมากจาก "เวลาอีนยัน" ในสภาพแวดล้อม PoA แบบ permissioned นี้

- 1) ความหน่วงเฉลี่ย (116.56ms): แทนเวลาเฉลี่ยสำหรับ leader node ในการประมวลผลคำสั่งและอัปเดตสถานะใหม่นั้นๆ ความจำ
- 2) ความหน่วง p50 (112.57ms): เวลาประมวลผลธุรกรรมมัธยฐาน
- 3) ความหน่วง p95 (180.04ms): เปอร์เซ็นไทล์ที่ 95 ภายใต้โหลดปกติ
- 4) ความหน่วง p99 (215.54ms): แทนเวลาประมวลผลกรณีล่วรร้ายที่สุด ภายใต้โหลดหนัก

โหมดจำลองรวมความถ้วนที่เพื่อจำลองพฤติกรรมบล็อกเชนที่สมจริง ในสภาพแวดล้อม PoA การผลิตที่มี validators ที่ปรับปรุงแล้ว ความหน่วงคาดว่า จะต่ำกว่าอย่างมีนัยสำคัญ ไม่เหมือนกับ mainnets สาธารณะที่ finality อาจใช้เวลาหลายวินาที ธรรมชาติ permissioned ของ GridTokenX cluster อนุญาตให้มี deterministic finality

F. การวิเคราะห์ Concurrency

ภายใต้การ แบ่งขั้น สูง (หมาย คำ สั่ง ต่อ สถานะ ตลาด เดียวกัน) Sealevel runtime linearized ธุรกรรมที่ขัดแย้งกันอย่างมีประสิทธิภาพในขณะที่ประมวลผลคำขอที่ไม่ทับซ้อนกันแบบขนาน อัตราความขัดแย้ง MVCC (Multi-Version Concurrency Control) ที่สังเกตได้คือ 1.45% ส่วนใหญ่เนื่องจากธรรมชาติของตัวมิกของคำสั่ง `match_orders` จำนวนครั้งการลองใหม่เฉลี่ยต่อ 0.02 แสดงการแก้ไขความขัดแย้งที่มีประสิทธิภาพ

G. การวิเคราะห์ความสามารถในการขยายตัว

เราทำการทดสอบความสามารถในการขยายตัวในสามมิติ:

ตารางที่ V
ผลลัพธ์การทดสอบความสามารถในการขยายตัว

การทดสอบ	TPS	ความหน่วง	ความเสถียร
1 thread (baseline)	443	2.26ms	100%
32 concurrent threads	398	2.51ms	คงไว้ 90%
60s sustained load	416	2.40ms	เสถียร
1,000 accounts	220	4.54ms	ลดลงแบบชิงเส้น

H. การเปรียบเทียบข้ามแพลตฟอร์ม

เรา เปรียบเทียบ แพลตฟอร์ม GridTokenX กับ ผลลัพธ์ ที่ เมย์ พร์ จาก แพลตฟอร์มบล็อกเชนอื่น:

แม้ว่า Hyperledger Fabric จะทำได้ปริมาณงาน DoNothing สูงกว่า (3,500 vs 225 TPS) น่องจากสถาปัตยกรรม execute-order-validate GridTokenX

ตารางที่ VI
การเปรียบเทียบประสิทธิภาพแพลตฟอร์ม

แพลตฟอร์ม	Benchmark	TPS	ความหน่วง	แหล่งที่มา
GridTokenX	Smallbank	1,714	5.8ms	การศึกษา
	DoNothing	225	2.5ms	การศึกษา
	YCSB-B	442	1.8ms	การศึกษา
Hyperledger 2.0	Smallbank	400	150ms	BLOCKBENCH
	DoNothing	3,500	45ms	BLOCKBENCH
	YCSB	200	200ms	BLOCKBENCH
Ethereum	DoNothing	15	13,000ms	BLOCKBENCH
	DoNothing	140	650ms	BLOCKBENCH

แสดง ประสิทธิภาพ Smallbank ต่ำกว่า 4.3 เท่า (1,714 vs 400 TPS) และ ความหน่วงต่ำกว่า 26 เท่า (5.8ms vs 150ms) สำหรับภาระงานทางการเงิน

I. "Trust Premium"

เรานิยาม "Trust Premium" เป็นต้นทุนประสิทธิภาพที่เกิดขึ้นเพื่อให้บรรลุ distributed consensus ในเครือข่ายส่วนตัวเมื่อเทียบกับ baseline แบบรวมศูนย์ (PostgreSQL)

$$\text{Trust Premium} = \frac{\text{LatencyBlockchain}}{\text{LatencyBaseline}} = \frac{116.56ms}{2.00ms} \approx 58.28x \quad (5)$$

แม้ว่าจะแทนค่าใช้จ่ายความหน่วงเมื่อเทียบกับฐานข้อมูลแบบรวมศูนย์ แต่ เป็นต้องมีรับตัวสำหรับแอปพลิเคชันการซื้อขายพลังงานที่การชำระราคายอดทั่วไปใช้เวลาหลายวันหรือหลายสัปดาห์ในระบบดังเดิม การแลกเปลี่ยนให้:

- เส้นทางการตรวจสอบธุรกรรมที่ไม่เปลี่ยนแปลง
- การชำระราคา smart contract อัตโนมัติ
- กลไกการกำหนดราคาที่โปร่งใส
- การจัดความเสี่ยงจุดล้มเหลวเดียว

J. สรุปผลลัพธ์ Benchmark

ตาราง VII สรุปเมทริกประสิทธิภาพหลักจากการประเมิน benchmark อย่างครอบคลุมของเรา

VII. กระบวนการประเมินข้ามแพลตฟอร์ม

เพื่อเปรียบเทียบ GridTokenX กับ แพลตฟอร์ม บล็อกเชน อื่น อย่าง เช่น งวด (เช่น Ethereum, Hyperledger Fabric) [4] เราเสนอ Standardized Energy Trading Workload (SETW) นักวิจัยต้องทำงานตามสามขั้นตอนเพื่อทำข้อวิธีการนี้:

A. การใช้งาน Contract ที่เขียนเบื้องต้น

แพลตฟอร์มเป้าหมายต้องใช้งาน logic หลักที่กำหนดในตาราง I โดยเฉพาะฟังก์ชัน "Order Match" ต้องเป็น atomic ให้แน่ใจว่าการทำธุรกรรมและการชำระราคากोเคลื่อนเกิดขึ้นในการเปลี่ยนสถานะ cryptographic เดียวกัน

B. โปรดีฟลัฟการฉีดภาระงาน

ตัวรับ荷载ต้องรักษาส่วนผสม TPC-C ที่เฉพาะเจาะจง:

- เขียนหนัก (88%): New Orders + Payments สิ่งนี้เน้น state contention ของ consensus engine
- อ่านเบา (12%): Status checks สิ่งนี้ทดสอบประสิทธิภาพ RPC query

ตารางที่ VII
สรุปผลลัพธ์ Benchmark อ่ายครอบคุณ

เมตريค	ค่า
BLOCKBENCH Micro-benchmarks	
DoNothing (Consensus)	225 TPS
CPUHeavy (Execution)	231 TPS
IOHeavy (Data Model)	192 TPS
YCSB/Smallbank	
YCSB-B (95% อ่าน)	442 ops/s
Smallbank OLTP	1,714 TPS
ชุด TPC Benchmark	
TPC-C (tpmC)	2,111
TPC-E (tpsE)	306
TPC-H (QphH)	250,486
เมตريคประสิทธิภาพ	
อัตราสำเร็จ	99.78%
ความหน่วงเฉลี่ย	116.56 ms
ความหน่วง p99	215.54 ms
อัตราความขัดแย้ง MVCC	1.45%
Trust Premium	58.28x

- [2] E. Mengelkamp, J. Gärtner, 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>

C. การ Normalize เมตريค

ผลลัพธ์ต้องรายงานใน **tpmC** (transactions per minute Type-C) สำหรับบล็อกเชนที่มี probabilistic finality (เช่น PoW) ความหน่วงต้องรวมเวลาถึง k -block confirmation depth เพื่อเปรียบเทียบกับ deterministic leader schedule ของ Solana

VIII. สรุป

GridTokenX แสดงให้เห็นว่าเทคโนโลยีบล็อกเชนส่วนตัวเป็นไปได้สำหรับการซื้อขายพลังงาน P2P ใน การ ติดตั้ง ระดับ ภูมิภาค ผ่าน การ วิเคราะห์ ขั้น BLOCKBENCH อ่ายครอบคุณและการประเมิน TPC benchmark:

- **Consensus Layer:** บริษัท DoNothing 225 TPS
- **Execution Layer:** การคำนวณ CPUHeavy 231 TPS
- **Data Model:** การดำเนินการสถานะ IOHeavy 192 TPS
- **Application Layer:** Smallbank 1,714 TPS, YCSB-B 442 ops/s
- **ชุด TPC:** 2,111 tpmC, 306 tpsE, 250,486 QphH

เมื่อเทียบ กับ Hyperledger Fabric GridTokenX ทำได้ ประมาณ งาน Smallbank สูงกว่า 4.3 เท่า และ ความหน่วงต่ำกว่า 26 เท่า สำหรับภาระงานทางการเงิน Trust Premium ที่ 58.28 เท่า ที่สังเกตได้แทนต้นทุนที่ยอมรับได้สำหรับการบรรลุสิ่งที่ต้องการตรวจสอบที่ไม่เปลี่ยนแปลง การซื้อขาย smart contract อัตโนมัติ และกลไกการกำหนดราคาที่ปรับเปลี่ยน

งานในอนาคตจะมุ่งเน้นที่: (1) การติดตั้งบันทึก PoA validator clusters ระดับภาระเพื่อให้ได้ความหน่วงต่ำลง (2) การขยายไปยังการกำหนดค่า warehouse ขนาดใหญ่ (10+ warehouses) และ (3) การทดสอบนำร่องในโลกจริงด้วยการผ่านรวมสมาร์ทมิเตอร์

หนังสืออ้างอิง

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