ACE Chain First-Principles Throughput Analysis

Architecture-derived estimates — not implementation benchmarks

ACE Labs

March 2026

ACE Chain First-Principles Throughput Analysis

Version: 2.0 | March 2026 **Methodology:** First-principles derivation from MVP architecture, industry hardware baselines, and explicit optimisation assumptions

---

1. Methodology

This report derives ACE Chain's theoretical throughput from first principles:

1. Survey the hardware configurations used by peer L1 chains for their published peak TPS benchmarks
2. Model ACE Chain on equivalent hardware, stage by stage through the transaction pipeline
3. Identify bottlenecks and compute per-scenario throughput ceilings
4. Derive configuration parameters that avoid artificial caps

No target TPS number is assumed. All figures are execution-only upper-bound estimates derived from the architecture plus the stated optimisation assumptions.

Scope note: Unless explicitly stated otherwise, this report models the execution path only. It excludes consensus, network propagation, signature / attestation verification, and durable persistence, and should not be read as a benchmark of the current implementation on current hardware.

---

2. Industry Hardware Baselines

2.1 Peer Chain Benchmark Hardware

Chain	Benchmark TPS	CPU	RAM	Network	GPU	Sustained?
Solana	65K (claimed) / 2–4K (mainnet)	24+ core AMD EPYC, 4 GHz+	512 GB	10 Gbps	Optional (CUDA SigVerify)	Mainnet: ~4K sustained

Chain	Benchmark TPS	CPU	RAM	Network	GPU	Sustained?
Aptos	170K (execution-only) / 30K (target)	32-core AMD EPYC (AWS c5a.16xlarge)	128 GB	N/A (single node)	No	Execution-only burst
Sui	297K (PTB=100) / ~11K (PTB=1)	24-core AMD	256 GB	25 Gbps	No	Peak on controlled workload
Monad	10K	16-core AMD Ryzen 7950X	32 GB	300 Mbps	No	Claimed sustained
Sei v2	12.5K	4 cores	8 GB	100 Mbps	No	Design target

2.2 Key Observations

- **Headline peak TPS numbers are execution-only, single-machine tests** — no consensus, no networking, no signature verification, no storage persistence.
- Mainnet sustained throughput is typically **3–10%** of the headline peak.
- Signature verification is the **#1 bottleneck** across all chains (Firedancer data: each verify tile handles only 20–40K TPS). Solana offloads this to GPU via CUDA.
- State I/O (storage reads/writes) is the **#2 bottleneck**, driving NVMe requirements (Aptos: 60K IOPS minimum).

2.3 Selected Baseline for ACE Chain Modelling

32-core AMD EPYC, 128 GB RAM, NVMe SSD (60K IOPS), 10 Gbps NIC

This matches the Aptos Block-STM benchmark hardware (AWS c5a.16xlarge) and is representative of a production validator node.

---

3. ACE Chain Pipeline Cost Model

3.1 Per-Stage Timing (derived from architecture + optimisation assumptions)

Stage	Per-tx Cost	Parallelisable?	Source
Attestation check (rayon batched)	~2–5 µs	Yes	`ace-runtime/src/pipeline/atte`

Stage	Per-tx Cost	Parallelisable?	Source
Write-set extraction + scheduling	~1–3 µs	No (sequential scan)	`ace-n-vm/src/scheduler.rs`
Execution — Native transfer	**~60–90 µs**	**Yes (per batch)**	`ace-n-vm/src/dispatcher.rs` → `AceNativeEngine`
Execution — EVM simple call	~200–300 µs	**No** (`WriteSet::Global`)	`ace-n-vm/src/evm/engine.rs` (revm)
Execution — SVM transfer	~50–100 µs	Yes	`ace-n-vm/src/svm/engine.rs`
Execution — BVM transfer	~50 µs	Yes	`ace-n-vm/src/bvm/engine.rs`
State write (BTreeMap)	~5–10 µs	Included in exec	`ace-model/src/state_tree.rs`
State write (RocksDB)	~10–50 µs	Included in exec	`ace-model/src/rocks_state_db.`
Merkle root computation	~10–20 ms/block	No	`ace-model/src/state_tree.rs`
ZK proof generation (GPU)	~30 ms/tx, 1,024 GPU threads	Async/pipelined	`ace-runtime/src/crypto/proof.`
Block serialisation	~300 bytes/tx	N/A	`ace-runtime/src/types/block.r`

3.2 Architectural Advantages vs Peers

1. **O(1) auth verification per block** — A single Groth16 pairing (~0.5 ms) replaces per-tx signature verification. This eliminates the #1 industry bottleneck entirely. On Solana/Firedancer, SigVerify tiles each handle only 20–40K TPS; ACE needs zero SigVerify capacity.
2. **No Proof-of-History serial overhead** — Solana requires a serial SHA-256 chain that consumes one full core. ACE has no such constraint.
3. **ZK proof generation is pipelined** — Proving is performed asynchronously (GPU) while the next block is being built. It does not sit on the critical path of block production.

---

4. Throughput Derivation

4.1 Slot Budget

```
Slot duration:                    400 ms
Fixed overhead:
  Merkle root computation:        ~15 ms
  Scheduling + dispatch overhead:  ~5 ms
```

```
  Block assembly + serialisation:  ~5 ms
                                   ──────
Available for execution:           ~375 ms
```

4.2 Scenario A — Pure Native Transfers (In-Memory BTreeMap)

```
Per-tx cost:       ~75 µs average (execution + state write)
Single-core:       375,000 / 75 = 5,000 tx/slot

32 cores, rayon parallel execution:
  Native transfers write {sender, recipient} — random addresses have
  near-zero conflict rate.
  Effective parallelism: ~85%
  Effective cores:       32 × 0.85 = 27.2
  Tx/slot:               27.2 × 5,000 = 136,000
  TPS:                   136,000 / 0.4 = 340,000
```

Adjustment for state clone overhead:

The current implementation clones the full `StateTree` per tx in multi-tx batches (`dispatcher.rs:335`). Clone cost is ~50–200 µs depending on state size, roughly doubling effective per-tx cost:

```
Effective per-tx:  ~150 µs (with clone)
Adjusted tx/slot:  27.2 × (375,000 / 150) = ~68,000
Adjusted TPS:      68,000 / 0.4 = ~170,000
```

With **copy-on-write snapshots** (optimisation opportunity):

```
Clone cost → ~1–5 µs
Effective per-tx:  ~80 µs
Optimised TPS:     27.2 × (375,000 / 80) / 0.4 ≈ 320,000
```

4.3 Scenario B — Mixed Workload (60% Native + 20% SVM + 20% EVM)

EVM transactions carry `WriteSet::Global`, forcing serialisation. Each EVM tx occupies its own batch.

```
Budget: 375 ms

Assume a practical block with E EVM transactions:
  Each EVM tx: ~250 µs (serial)
  E = 200 → 200 × 250 µs = 50 ms consumed

Remaining for parallel native + SVM:  325 ms
  Per-tx: ~75 µs, 27.2 effective cores
  Parallel tx: 27.2 × (325,000 / 75) ≈ 117,800

Total tx/slot: 200 + 117,800 ≈ 118,000
TPS: 118,000 / 0.4 ≈ 295,000
```

In practice, EVM ratios will vary. A summary:

EVM share	EVM tx count	Parallel tx	Total tx/slot	TPS
0%	0	136,000	136,000	340,000
10%	100	131,500	131,600	329,000
20%	200	117,800	118,000	295,000
50%	500	90,700	91,200	228,000
100%	1,500	0	1,500	3,750

4.4 Scenario C — Persistent Storage (RocksDB)

```
RocksDB random read:  ~10–50 µs (vs BTreeMap ~0.1 µs)
Per-tx total:          ~150–300 µs

32 cores, ~70% effective parallelism (I/O contention):
  Effective cores: 22.4
  Tx/slot = 22.4 × (375,000 / 225) ≈ 37,300
  TPS = 37,300 / 0.4 ≈ 93,000

NVMe SSD at 60K IOPS (Aptos spec):
  I/O ceiling: ~30K random reads/slot → ~75,000 TPS upper bound
```

4.5 Summary

Scenario	Tx/Slot	TPS (MVP architecture ceiling estimate)	TPS (with roadmap optimisations)	Primary Bottleneck
Pure native (in-memory)	68,000–136,000	170,000–340,000	~320,000	State clone / CPU
Mixed 60/20/20 (in-memory)	50,000–118,000	125,000–295,000	~300,000	EVM serialisation
Persistent (RocksDB)	30,000–37,000	75,000–93,000	~100,000	Storage I/O
EVM-heavy (100%)	1,000–1,500	2,500–3,750	~5,000	`WriteSet::Global`

---

5. Network Propagation Constraint

```
Block wire size:  header (256 B) + N × ~300 B/tx
At 80,000 tx:     256 + 80,000 × 300 = ~24 MB
Block rate:        2.5 blocks/s → ~60 MB/s sustained
```

```
10 Gbps NIC:  ~1.25 GB/s → comfortably handles 60 MB/s
25 Gbps NIC:  ~3.1 GB/s → headroom for P2P gossip fanout

P2P gossip overhead (×4–8 fanout): 240–480 MB/s — fits within 10 Gbps.
```

Network propagation is NOT the bottleneck at 80,000 tx/block on 10 Gbps.

---

6. ZK Proof Pipeline Constraint

```
Per-tx proving:            ~30 ms (Groth16 per circuit)
GPU parallelism (A100):    1,024 threads
Tx provable per slot:      1,024 × (400 / 30) ≈ 13,600

For 80,000 tx/block:
  Requires ~6 slots of pipelined GPU proving
  (builder produces block in slot S, proof ready by slot S+6)

With recursive aggregation:
  Proof can aggregate sub-proofs → eliminates per-tx proving ceiling
```

The ZK proof pipeline is the reason `MAX_PROOF_BUNDLE_ENTRIES` (1,536) is **decoupled** from `MAX_TXS_PER_BLOCK` (80,000). Blocks with more transactions than the bundle limit will span multiple proof bundles or use recursive aggregation when available.

---

7. Derived Configuration Parameters

Parameter	Previous	New	Rationale
`MAX_TXS_PER_BLOCK`	1,536	**80,000**	Avoids being an artificial bottleneck; derived from 32-core in-memory ceiling (~136K tx/slot) with margin
`MAX_PROOF_BUNDLE_ENTRIES`	1,536	**1,536** (unchanged)	ZK verifier constraint, independent of tx capacity; recursive aggregation will lift this
`MAX_BLOCK_BYTES`	512 KiB	**32 MiB**	80,000 tx × ~300 B = ~24 MB; 32 MiB provides headroom
`MAX_P2P_MESSAGE_BYTES`	2 MiB	**64 MiB**	Must accommodate a full block + finality certificate in sync responses
`GPU_THREAD_PARALLELISM`	128	**1,024**	Reflects data-centre GPUs (A100 / H100); previous value was consumer RTX 4090

Parameter	Previous	New	Rationale
`SLOT_DURATION_MS`	400	**400** (unchanged)	Already aggressive; shortening would reduce consensus stability

---

8. Comparison with Peer Claims

Chain	Claimed Peak	Our Modelled Peak (same hardware class)	ACE Advantage
Solana	65K	170K–340K	O(1) auth eliminates SigVerify bottleneck
Aptos	170K (exec-only)	170K–340K	Comparable execution; no Block-STM overhead for non-conflicting tx
Sui	297K (PTB=100)	170K–340K	Apples-to-apples PTB=1: Sui ~11K vs ACE 170K+
Monad	10K	170K–340K	Consumer vs server hardware; architecture advantage on auth

Critical caveat: ACE Chain's numbers above are execution-only upper-bound estimates on equivalent hardware, extrapolated from an MVP architecture baseline rather than measured from the current implementation. Real mainnet sustained TPS will be materially lower and depends on consensus, networking, signature / attestation validation, and persistence. A directional sustained range of **10,000–30,000 TPS** remains consistent with industry patterns, but should be treated as a separate full-system estimate.

---

9. Optimisation Roadmap (Not Yet Implemented)

Optimisation	Expected Impact	Difficulty
Copy-on-write state snapshots	50–80% reduction in parallel batch overhead	Medium
Incremental Merkle trees	50–70% faster state root computation	Medium

Optimisation	Expected Impact	Difficulty
EVM write-set static analysis	Remove `WriteSet::Global` for simple EVM transfers	High
Recursive proof aggregation	Remove `MAX_PROOF_BUNDLE_ENTRIES` ceiling	High
Pipelined block execution	Overlap execution with previous block's proving	Medium
Custom state DB (replacing RocksDB)	2–5× I/O throughput (cf. MonadDB, SeiDB)	Very High

---

10. ACE Chain vs Solana: TPS and Algorithm Comparison

10.1 Consensus Mechanism

Dimension	ACE Chain	Solana
Consensus model	BFT + PoH + ZK proof	Tower BFT + PoH
Slot duration	400 ms	400 ms
Soft finality	~400 ms (⅔ stake-weighted votes)	~400 ms (optimistic confirmation)
Hard finality	~600 ms (Groth16 ZK proof, target)	~12 s (31 confirmations)
Block verification complexity	**O(1)** (single ZK proof per block)	**O(n)** (per-tx signature verification)

Hard finality is ~20× faster than Solana by design. This stems from the O(1) verification property of recursive Groth16 proofs: regardless of how many transactions the block contains, verification cost is fixed (~0.5 ms, 3 pairing checks).

10.2 Transaction Processing Pipeline

Solana pipeline: Gulf Stream SVM execution Turbine propagation Tower BFT voting - Each transaction requires Ed25519 signature verification (~76 µs/tx) - Signatures are stored on-chain (64 B signature + 32 B pubkey = 96 B/tx)

ACE pipeline: Attest Execute Prove (3-phase) - Phase 1a (Attest): HMAC-SHA256 credential creation (~1 µs/tx on client); Phase 1a skips HMAC credential verification (deferred to ZK circuit) - Phase 1b (Execute): n-VM parallel execution (~10–50 µs/tx) - Phase 2 (Prove): Groth16 ZK proof (GPU async, **off critical path**)

Core difference: ACE's identity-authorization separation model decouples on-chain identity (`idcom` commitment) from signing credentials. The HMAC credential is verified inside the ZK circuit (Phase 2), not by validators during block processing.

10.3 Parallel Execution Model

Solana (Sealevel): - Transactions pre-declare read/write account sets - Non-conflicting transactions execute in parallel - Developers must manually declare accounts; CPI chains can be error-prone

ACE (n-VM Scheduler): - Automatic write-set extraction from opcode prefix - Greedy batching: single-pass scan, group non-conflicting transactions - Same-sender transactions preserve nonce ordering - Batches execute sequentially; transactions within each batch run in parallel (rayon) - EVM/SVM/TVM generic contract calls marked as `WriteSet::Global` (conservative serial fallback)

Both approaches are conceptually similar. ACE's automatic write-set extraction is more developer-friendly, but generic contract calls degrade to serial execution in both systems.

10.4 TPS Comparison

Scenario	ACE Chain	Solana
Theoretical peak (in-memory)	~170K–340K tx/slot	~65K tx/slot (claimed)
Single-shard sustained (estimate)	~5,000–10,000	~4,000 (mainnet measured)
Multi-shard (4 shards, estimate)	~17,000	N/A (no native sharding)
Multi-shard (8 shards, estimate)	~31,000	N/A
Block limit	80,000 tx / 32 MiB	~48M CU (~tens of thousands of tx)

Note: ACE Chain's multi-shard figures are blueprint projections (see §12 of ACE_CHAIN_BLUEPRINT.md), not current implementation benchmarks. Solana's 4K TPS is mainnet-measured sustained throughput (excluding vote transactions, which account for ~65% of total on-chain TPS).

10.5 Structural Advantages

1. **Sub-second hard finality**: ~600 ms via ZK proof (target) vs Solana's ~12 s (31 confirmations). This is a finality-quality difference — ACE produces a cryptographic proof, Solana relies on a probabilistic time window.
2. **O(1) block verification**: A single recursive proof covers the entire block, regardless of transaction count. Verification nodes do not need to re-verify individual signatures.
3. **State sharding readiness**: HKDF context isolation enables zero-coordination parallel shards with linear TPS scaling. Solana's single-chain architecture scales via hardware vertical scaling + Firedancer client optimisation.
4. **Multi-protocol native execution**: Unified dispatch for Ethereum, Solana, Bitcoin, and Tron payloads via n-VM opcode routing. Solana only supports SVM natively; EVM compatibility requires third-party bridges (e.g., Neon).
5. **PQC migration path**: Tagged signatures allow algorithm swap (Ed25519 ML-DSA-44 HMAC-SHA256) without account model changes, because on-chain identity is an `idcom` commitment, not a public key.

11. Cost per Million Transactions: ACE Chain vs Solana

11.1 Solana Operating Cost Model

Data sources: mainnet validator economics reports (2025–2026).

Item	Value	Source
Real user TPS	~1,000 (excl. vote tx)	Chainspect, Solscan
Total TPS (incl. votes)	~3,500	Solana Compass
Active validators	~1,500	Solana Labs
Bare-metal monthly cost	$400–$1,200	Cherry Servers, Hivelocity
Hardware requirements	16+ core CPU, 256–512 GB RAM, NVMe	Agave docs
Power consumption	300–500 W continuous	Industry estimate
Electricity	~$50–80/month	US datacenter rates
Bandwidth	1–2 TB/month, ~$100–300/month	Provider estimates
Vote cost	~300 SOL/year/node (~$45,000 at $150/SOL)	Helius
Annual operating cost/node	~$60,000 (incl. votes)	Hivelocity
GPU requirement	None (optional CUDA SigVerify)	—

Per-million-transaction cost derivation:

```
Real user TPS:          1,000
Annual transactions:    1,000 × 86,400 × 365 = 31.5 B
Network annual cost:    1,500 × $60,000 = $90 M
Cost per million tx:    $90M / 31,500 ≈ $2.86

Excluding vote overhead (votes are ~65% of on-chain activity but a fixed cost):
  Adjusted estimate:    ~$3–5 per million tx
```

11.2 ACE Chain Operating Cost Model

Item	Value	Notes
Target TPS	~5,000 (single shard)	Blueprint estimate
Validator count	~200 (initial cap)	KYC-based admission
CPU node monthly cost	$400–$800	Lower RAM than Solana (128 vs 512 GB)
GPU prover nodes	2–3 (H100 class)	Pipelined, not per-validator
GPU cloud cost	~$3/GPU-hr	H100 on-demand (2025 pricing)
GPU power	~700 W/card	H100 TDP

Item	Value	Notes
GPU electricity	~$100–150/month/card	Datacenter rates
Vote cost	**$0**	BFT votes are off-chain messages
Bandwidth	~$100–200/month	O(1) proof reduces sync overhead

GPU proving cost per million transactions:

```
400 ms/slot, ~5,000 tx/slot
1M tx requires: 200 slots = 80 seconds of GPU time

H100 cloud: ~$3/GPU-hr = $0.00083/GPU-second
80 seconds × $0.00083 = ~$0.067

→ GPU proving: ~$0.07 per million tx
```

GPU proving cost is negligible because proofs are amortised across entire blocks (O(1) recursive proof), not per-transaction.

Full-network cost derivation (200 nodes + 2 GPU provers):

```
CPU validator nodes:  200 × ($600/month × 12) = $1.44 M/year
GPU prover nodes:     2 × ($3/hr × 24 × 365)  = $52,560/year
Annual total:         ~$1.5 M/year

Annual transactions:  5,000 × 86,400 × 365 = 157.7 B
Cost per million tx:  $1.5M / 157,700 ≈ $0.0095

→ Less than $0.01 per million tx
```

11.3 Cost Comparison Summary

Cost Dimension	Solana	ACE Chain	Advantage
Compute / electricity			
CPU validation	~300–500 W/node	~300–500 W/node	Comparable
GPU proving	None	H100 ~700 W (2–3 units)	Solana saves GPU
Vote tx compute	65% of total throughput	None	**ACE saves**
Hardware			
Per-node rental	$400–$1,200/month	$400–$800/month	**ACE lower** (less RAM)
GPU hardware	None	~$4,380/month (2 H100s)	Solana saves
Validator count	~1,500	~200 (initial)	**ACE 7.5× fewer**

Cost Dimension	Solana	ACE Chain	Advantage
Protocol overhead			
Vote cost	~$45K/year/node	$0	**ACE saves**
Network-wide votes	1,500 × $45K = **$67.5 M/year**	$0	**ACE major advantage**
Bandwidth			
Block propagation	~300–400 B/tx + signatures	~250 B/tx + fixed-size proof	**ACE slightly better**
Light-client sync	Per-tx signature verification	Verify 1 proof	**ACE much better**
Operations			
Tooling maturity	Mature ecosystem	New system, tooling TBD	**Solana better**
Estimated DevOps	$8K–15K/month	$8K–15K/month	Comparable

Metric	Solana	ACE (single shard)	ACE (4 shards)
Network annual cost	~$90 M	~$1.5 M	~$3 M
Annual transactions	~31.5 B	~157.7 B	~536 B
Cost per million tx	**~$2.86**	**~$0.0095**	**~$0.0056**

11.4 Caveats

1. **Solana costs are real-world data; ACE costs are model projections.** Actual costs post-mainnet may differ due to unforeseen operational overheads.
2. **ACE's lower validator count is due to pre-mainnet status**, not an architectural limitation. If ACE scales to 1,500 validators, CPU node costs rise proportionally — but GPU proving cost remains unchanged (O(1) verification means validators don't need GPUs, only provers do).
3. **Solana's vote cost is the largest hidden expense.** $67.5 M/year in vote transaction fees is a structural cost borne by the entire network, and it fluctuates with SOL price. ACE's BFT votes produce no on-chain transactions.
4. **GPU proving cost is extremely low** because it is amortised across the entire block. A single H100 GPU computation (~240 ms) covers all transactions in one slot. This is the economic advantage of O(1) recursive proofs.
5. **Operational maturity is the biggest unknown.** A new chain's monitoring, debugging, and incident-response tooling is far less mature than Solana's. Early-stage operations may require significantly more human investment.