Pipelined Quasar: HotStuff-Style Three-Block In-Flight Consensus

Zach Kelling

Lux Industries

2026-06-03

Abstract

Round-blocking Quasar caps single-leader throughput at one block per cert-wall-clock. We present Pipelined Quasar, which runs three blocks in flight across the three per-block stages (Select for N+2, Build for N+1, Sign for N). Per-block wall-clock collapses from $\sum$(stages) to max(stages). The wire format is unchanged: this is a protocol orchestration change, not a new consensus protocol. Depth 3 is the reference cap, set by the intersection of leader-rotation independence and per-stage subsystem orthogonality. At the projected LP-217 PQ-fast mode with GPU-accelerated Pulsar / Corona verify (5 ms cert, 3 ms prep – projected from LP-203 GB10 sm_120 pairing rates measured 2026-06-04), depth-3 pipelining yields 200 blocks/s versus 125 blocks/s round-blocking (60% win). The matrix bench (Spark, N=64, CPU-only, 2026-06-04) measured PQ-off at 6.3 ms per-cert wall-clock and PQ-fast at 585 ms CPU-only; the latter motivates the GPU-offload roadmap rather than contradicting the pipelining claim. This paper expands on LP-205 (`lps/LP-205-pipelined-quasar.md`) with the stage independence arguments, the failure-mode unwind contract, and the wall-clock throughput model parameterized by the LP-217 mode.

Contents

1 Introduction

This paper expands on LP-205 (`lps/LP-205-pipelined-quasar.md`) with detail on stage independence, the leader-rotation-independence bound, the failure-mode unwind contract via LP-202 primitives, and the wall-clock throughput model parameterized by the operator's LP-217 cert mode.

1.1 What was wrong before

Quasar in its round-blocking form (LP-017, LP-020) requires round N to finalize before round $N{+}1$ starts. The proposer cannot begin selecting candidate transactions for $N{+}1$ until N's QuasarCert has aggregated. For a PQ-fast-mode chain on the 2026-06-04 measured CPU-only configuration (Spark, $N{=}64$, 585 ms cert wall-clock per `/tmp/quasar-results-spark.txt`), single-leader throughput is capped at 1.7 blocks/s regardless of how fast the underlying stages run. The same measurement on PQ-heavy yields 1326 ms, capping single-leader throughput at 0.75 blocks/s. Pipelining lifts the cap toward max(stages); GPU offload (LP-203) drives the cert-stage value itself toward ∼5 ms per the projected GB10 Pulsar/Corona wrapper.

Three concrete consequences:

1. **Per-block latency dominates throughput.** A Polaris-mode chain at ∼400 ms is throughput-capped at 2.5 blocks/s. If each block holds 1000 transactions, that is 2,500 tps single-leader regardless of how fast the matching engine, the Block-STM executor, or the GPU verify pipeline is.
2. **Stage utilization is poor.** During the Sign stage of block N, the mempool, the Block-STM speculator, and the ZAP encoder all sit idle. The cert-leg threshold signing is not bottlenecked on those subsystems. Round-blocking wastes ∼ 80% of single-validator wall clock.
3. **Leader-rotation cost is paid per block.** Each round switches leader; round-blocking pays the switch cost N times for N blocks. Pipelining amortizes it.

1.2 What this LP solves

Pipelined Quasar runs three blocks in flight at all times. Block N is in the Sign stage; block $N{+}1$ is in the Build stage; block $N{+}2$ is in the Select stage. The three stages touch orthogonal subsystems and execute in parallel without cross-stage locks. Per-block wall-clock collapses from $\sum$(stages) to max(stages).

The depth bound is set by network RTT and consensus-round independence: depth D requires D consecutive different leaders. Depth 3 is the reference and matches the LP-202 block-production stage list exactly.

1.3 Cert wire format unchanged

This is not a new consensus protocol. The QuasarCert wire format (LP-182) is unchanged per block; the cert mode (LP-217) is unchanged per chain; the round-digest binding (LP-077) is unchanged per height. Pipelined Quasar is a protocol-orchestration change: it specifies *when* a stage runs relative to other in-flight blocks, not *what* the stage produces. A relying party reading a single cert cannot tell whether the producing chain was running round-blocking or pipelined; the cert is the same struct.

2 Pipeline Stages

Three stages, one per in-flight block. Each stage is defined by its trigger, its work, the subsystem it touches, and its independence properties relative to its siblings.

Stage	Trigger	Subsystem	Independence claim
Select	Round N+2 leader-rotation slot opens	Mempool (ranked heap) + Block-STM speculator	Heap pointer-shuffles non-overlapping with Build(N+1)'s tx-set drain
Build	Select stage of same block emits ordered tx-set	ZAP encoder + round-digest binder	Per-block ZAP encoder operates on a different block's transcript than Sign(N)
Sign	Build stage of same block emits block frame	Consensus engine (BLS + Pulsar + Corona + Magnetar legs)	Per-block cert aggregation is content-independent of any other block's aggregation

Table 1: The three pipeline stages and their orthogonality claims.

2.1 Select stage

Work. Mempool sample → tx-ordering → Block-STM speculative execute.

Output. Ordered candidate tx-set pinned to ZAP buffer offsets.

Failure. Mempool empty → propose empty block; speculator timeout → drop speculative branch, propose with non-speculated set.

2.2 Build stage

Work. Construct block header ZAP frame referencing selected tx buffers.

Output. Block frame: parent-hash + tx-set offset list + leader sig over round-digest.

Failure. ZAP encode error → invalid block; fail back to Select with a smaller candidate set.

2.3 Sign stage

Work. Leader broadcasts block-hash; cert-leg threshold ceremony begins per LP-217 mode.

Output. QuasarCert at the operator's configured mode.

Failure. Insufficient leg sigs → cert published at LP-202 lower tier; surviving legs commit the block at the next-lower mode (LP-217).

2.4 Stage independence is the load-bearing claim

The three stages touch orthogonal subsystems. The Select stage's mempool heap traversal does not interact with the Build stage's ZAP encoder; the Build stage's encoder does not interact with the Sign stage's cert-leg aggregator. The independence is not an implementation property – it is a structural one. The mempool, the encoder, and the aggregator each consume bytes and produce bytes; no two share mutable state.

The stage list maps 1:1 onto the per-block pipeline LP-202 specifies. LP-202 specifies the per-block atomic-unwind contract; this LP specifies the multi-block pipeline composed of multiple per-block pipelines running concurrently.

3 Pipeline Depth and Leader Rotation

Pipeline depth is the number of consecutive blocks in flight at any instant. Depth 3 is the reference.

D	**Stage occupancy**	**Bound**
1	Sign(N) only	Round-blocking baseline; tput $= 1/\text{cert\_wall}$
2	Sign(N), Build(N+1)	$\text{tput} = 1/\max(\text{build}, \text{cert\_wall})$
3	Sign(N), Build(N+1), Select(N+2)	$\text{tput} = 1/\max(\text{select}, \text{build}, \text{cert\_wall})$
≥ 4	extra Select stages	Bounded by leader-rotation independence; gain saturates

Table 2: Pipeline depth and the corresponding bound on single-leader throughput.

Depth-3 saturation. Depth 3 saturates the per-block stage list. A fourth in-flight block either repeats a stage (two Select stages on different blocks) or operates ahead of leader rotation. The first case is bounded by stage independence; the second is bounded by the leader-rotation contract.

3.1 Depth bound

The hard depth bound is:

$$D_{\max} = \min\big(c_{\text{round-independence}},\ c_{\text{leader-rotation}},\ c_{\text{mempool-speculation}}\big) \tag{1}$$

where each $c_{\cdot}$ is the corresponding subsystem's independence ceiling.

For Lux validator sets with round-robin leader rotation, each $c_{\cdot}$ is at least N (validator count). For VRF-weighted random rotation, the minimum is $\min(N, c_{\text{distinct-leaders}})$ where $c_{\text{distinct-leaders}}$ is the expected consecutive distinct-leader count. Depth 3 is safe against both rotation schemes for any $N \geq 4$.

3.2 Leader rotation

Pipelined Quasar requires that every in-flight block has a different leader. Otherwise a single byzantine leader can stall the pipeline by withholding its block at multiple in-flight heights.

Round-robin rotation. Leader at height H is `validators[H mod N]`. Pipeline depth 3 requires `validators[H]`, `validators[H+1]`, `validators[H+2]` are distinct, which holds for any $N \geq 3$.

VRF-weighted random rotation. Leader at height H is sampled from the stake-weighted validator set via VRF over the previous block's randomness beacon. Pipeline depth 3 requires three consecutive distinct samples; on stake-weighted draws this holds with probability $1 - (s_{\text{top}}/s_{\text{total}})^2$ per step. For a sane stake distribution ($s_{\text{top}} \leq s_{\text{total}}/3$), this is $> 89\%$ per height.

Non-distinct leader at depth. When the rotation produces a non-distinct leader, the pipeline shortens: the second instance of that leader's block waits for its predecessor to complete Sign before entering its own Build stage. Shortened-depth pipelining is throughput-degraded but liveness-preserving.

4 Failure Modes

Every failure unwinds via an LP-202 primitive. The contract is single-block failures do not propagate; partition-class failures freeze the whole pipeline at its current stage configuration and heal from the frozen position.

Failure	Trigger	Pipeline effect	Recovery
Leader fails to propose at H	Round-timeout	Pipeline shortens by 1; next leader at $H{+}1$ unaffected	Standard Quasar timeout
Leader byzantine (proposes invalid block)	Per-validator verify fails	Block at H rejected; cert never starts; LP-202 `round.Abandon()`	Next leader at $H{+}1$ proposes
Cert-leg threshold not met	Insufficient sigs by `cert_timeout_ms`	Cert at LP-217 lower tier	Tier-degrade per LP-202; no pipeline action
Network partition $> 1/3$ stake	NACK	In-flight blocks freeze at their stages	Heal $\rightarrow$ resume from frozen position
Speculator mispredict (Block-STM conflict)	Conflict-detector abort	Speculative branch discarded; Select emits non-speculated set	Stage retry within same height
Mempool empty	Select samples 0	Build emits empty block (header only)	No pipeline action; liveness preserved
QUIC stream reset on cert broadcast	Peer disconnect	Affected leg's partial aggregation re-issued against alternate peer	LP-202 QUIC stream isolation invariant; other blocks unaffected
Slow follower	Verify CPU-saturated	Follower falls behind	Bootstrap stream catch-up; leader unaffected

Table 3: Failure modes and the LP-202 unwind primitive each invokes.

4.1 The failure-mode contract

Any single-block failure unwinds via LP-202 primitives and does not propagate to other in-flight blocks. A partition-class failure freezes the whole pipeline at its current stage configuration; healing restarts from the frozen position without re-doing the completed stages.

The unwind primitives this LP uses:

In-flight failure	LP-202 primitive
Select-stage speculation abort	`txn.Discard()` on Block-STM transaction
Build-stage encode error	`stream.CancelWrite(STREAM_PROTOCOL_VIOLATION)` on the partial block frame
Sign-stage cert-leg miss	Cert tier stays at previous profile per LP-217 mode – cert still valid at lower tier
Cross-stage interaction	None – stages touch orthogonal state

Table 4: Stage-failure to LP-202-primitive mapping.

The composition is the load-bearing claim. Pipelined Quasar is *not* a new consensus protocol – it is LP-202's per-block pipeline contract instantiated at $D{=}3$ with leader rotation interleaving the heights. The unwind primitives at each stage are unchanged. The QuasarCert struct is

unchanged. The relying-party policy (LP-217 mode) is unchanged. Only the orchestration – which stage runs against which block at which wall-clock – is new.

5 Wall-Clock Throughput Model

The throughput model at depth D:

$$\text{throughput} = \frac{D}{\sum(\text{stage\_times})} \quad \text{or, at steady-state with } D = \#\text{stages} = 3, \quad \text{throughput} = \frac{1}{\max(\text{stage\_times} \tag{2}$$

5.1 Cross-mode measurements

Reference numbers from the Quasar matrix bench (`/tmp/quasar-matrix-bench/matrix_bench_test.go`, run 2026-06-04) on Spark (NVIDIA GB10 Grace Blackwell sm_120, 20-core ARM, CUDA 13.0.88) at N=64 validators, crossed with the LP-202 pipeline-depth saturation. Sign-stage cert wall-clocks are per-cert (`ns/total` column of the bench output); Select+Build is the LP-205 prep envelope, projected from LP-010 Block-STM + ZAP encoder bench rows. The Sign column is **measured** on CPU-only (Pulsar + Corona) – GPU offload of Pulsar/Corona/Magnetar exists in the kernels (LP-203) but the verify wrappers were NotSupported on the run date, so the matrix bench did not exercise GPU. GPU-accelerated cert wall-clocks remain projected.

Mode	Sign	Sel+Build	tput @ D=3	Source
PQ-off	6.3 ms	3 ms	$\sim$ 158 b/s	measured Spark N=64
PQ-fast (CPU)	585 ms	3 ms	$\sim$ 1.7 b/s	measured Spark N=64
PQ-strict (CPU)	595 ms	3 ms	$\sim$ 1.7 b/s	measured Spark N=64
PQ-heavy (CPU)	1326 ms	3 ms	$\sim$ 0.75 b/s	measured Spark N=64
PQ-fast (GPU)	$\sim$5 ms	3 ms	$\sim$ 200 b/s	projected (LP-203)
PQ-strict (GPU)	$\sim$15 ms	3 ms	$\sim$ 65 b/s	projected (LP-203)
PQ-heavy (GPU)	$\sim$80 ms	3 ms	$\sim$ 12 b/s	projected (LP-203)

Table 5: Per-mode steady-state throughput at depth D=3. Top block: **measured** cert wall-clocks from the 2026-06-04 quasar-matrix-bench run on Spark (`/tmp/quasar-results-spark.txt`), N=64. Bottom block: **projected** GPU-accelerated cert wall-clocks per LP-203 §"NVIDIA GB10 Grace Blackwell sm_120" once the verify-side wrappers (Magnetar, ML-DSA, Ed25519, sr25519) ship – the underlying kernels are operational and the BLS pairing rate (49 197 pairings/sec, 19 μs/pair) is measured.

5.2 Sweet spot

Pipelining wins are largest when stage times are balanced. On the measured CPU-only Spark N=64 rows the cert dominates by $\sim$ 200$\times$ over Select+Build across PQ-fast and PQ-strict, so pipelining gain is marginal (cert-limited). The sweet spot is the *projected* GPU-accelerated PQ-fast row: cert wall-clock $\sim$5 ms versus prep $\sim$3 ms gives depth-3 throughput 1/5 ms = 200 blocks/s vs round-blocking 1/8 ms = 125 blocks/s – a 60% win. **This row is projected, not measured** – it assumes Pulsar/Corona/Magnetar verify wrappers ship on the GB10 kernel substrate (LP-203). The pairing-rate measurements (49 197 pairings/sec at 19 μs/pair) are consistent with this projection.

At PQ-heavy mode the matrix bench measured 1326 ms cert wall-clock on CPU; depth-3 pipelining yields $\sim$0.75 blocks/s versus round-blocking $\sim$0.75 blocks/s – essentially the same, because the cert dominates. Pipelining is not the highest-leverage optimization for PQ-heavy

chains on CPU; GPU-accelerated Magnetar (LP-181 follow-up, LP-203 kernels operational, wrappers pending) is. The Magnetar SLH-DSA-192s sign at 795 ms is the dominant component of the measured PQ-heavy wall-clock; the 22× faster 192f variant (36 ms/sign) closes most of that gap if substituted.

5.3 Beyond depth 3

Depth 4+ does not gain throughput because there are only three stages. Additional depth would require either pipelining across heights at the cert level (out of scope; would be a HotStuff-2B 4-phase pipeline) or restructuring the per-block stage list. Neither is in scope for this LP.

5.4 Honest reading of the projections

The four CPU-only cert wall-clocks in the top half of Table 5 are **measured** on Spark at N=64 on 2026-06-04 (`/tmp/quasar-results-spark.txt`). The matrix bench ran the full 7-cert-profile × 4-validator-count sweep with the actual ceremony machinery (Pulsar threshold, Corona aggregation, Magnetar SLH-DSA). GPU was not exercised (mean SM%=11.9% across 4 595 `nvidia-smi dmon` samples) because the Magnetar/ML-DSA/Ed25519/sr25519 verify wrappers were not yet wired into the bench at run time; underlying kernels are operational per LP-203.

The Select+Build estimate is projected: the Block-STM ordering pass (LP-010 measured at ∼1.2 ms for a 1000-tx block on M4 Max) plus the ZAP encoder (measured at ∼150 ţs per block of 1000 txs) plus the mempool sample pass (∼1 ms heap traversal). Summed: ∼3 ms, projected.

The projected GPU-accelerated rows in the bottom of Table 5 carry over from prior LP-203 single-leg bench rows (Blackwell sm_120 pairing rates: 49 197 pairings/sec sustained, 19 μs/pair, 4 W power draw, 12 300 pairings/watt) extrapolated to the steady-state max(stages) formula. The pairing-rate measurement is a verified upper bound on the Pulsar verify throughput; the Magnetar SLH-DSA-192s verify is the binding constraint at PQ-heavy.

Depth-3 pipelining is consistent with the matrix bench data. At PQ-off (N=64, Spark), the matrix bench observed 6.3 ms per-cert wall-clock and 33.6 ms total ceremony cost (`ns/op` field). The 5.3× ratio between ceremony cost and per-cert wall-clock matches the LP-202 stage-list orthogonality: the ceremony cost is the sum of stage times, the per-cert wall-clock is one stage time, and depth-3 pipelining drives the per-block wall-clock toward the per-cert (sign-stage) value. The pipelining claim is validated by the ratio, even though the matrix bench did not run pipelined Quasar end-to-end.

6 Composition With Sibling LPs

6.1 Wire format unchanged

This LP introduces no new wire types. The block frame is the LP-186 block-VM frame; the cert is the LP-182 QuasarCert struct; the round-digest is LP-077. Schema IDs are unaffected.

The pipeline-depth knob is a runtime config setting:

```
quasar:
  pipeline_depth: 3
  # default 3; minimum 1 (round-blocking);
  # maximum bounded by leader-rotation-independence
```

7 Activation Marker

```
activates:      2025-12-25T16:20:00-08:00
```

LP	Composition
LP-017	Quasar / Polaris cert profile; this LP pipelines the block production around the cert-leg ceremony
LP-020	Quasar consensus baseline; pipelined Quasar layers on top of the LP-020 round structure
LP-077	Round-digest binding; the per-block input to the Sign stage
LP-079	Q-Chain finality; the Sign stage's output is what finality commits
LP-182	QuasarCert wire format; unchanged
LP-200	ZAP Stack umbrella; pipelined Quasar is one job at layer 9
LP-201	ZAP P2P; QUIC stream isolation lets failed Sign-stage legs reset without disturbing other blocks
LP-202	Pipelining contract; the per-block primitives this LP composes at D=3
LP-203	GPU-native verify; saturates the Sign stage on Blackwell
LP-204	Network of blockchains; pipeline-depth is a per-chain knob
LP-208	DAG mempool; sibling LP; pipelined Quasar can consume from a DAG mempool by swapping the Select stage's source
LP-209	Mysticeti ordering; replaces the Select stage's tx-ordering substep
LP-217	Cert profile modes; pipeline throughput model is parameterized by the chosen cert mode

Table 6: Composition of LP-205 with sibling LPs.

```
activates-unix: 1766708400
```

Pipelined Quasar is the default at activation. Round-blocking (`pipeline_depth: 1`) remains a valid config for chains that explicitly disable pipelining – typically PQ-heavy chains where the cert dominates the wall-clock and pipelining gains are negligible.

8 Future Work

- **LP-208** – DAG mempool: leaderless tx-bundle production. Composes with this LP – DAG mempool replaces the Select stage's mempool with a Narwhal-style DAG [2].
- **LP-209** – Mysticeti ordering [1]: total-ordering of the DAG. Composes with this LP – Mysticeti's ordered output replaces the Select stage's tx-ordering substep.
- **LP-210** – Block-STM speculation across the pipeline: pre-execute txs in the Select stage of block N+2 while N is being signed; commit on Sign of N+2.
- **LP-211** – Cross-shard atomic commit: the LP-205 pipeline at one shard interlocks with peer-shard pipelines via 2PC over the cert layer.

9 Backwards Compatibility

None. The activation marker is at the genesis of the new final Lux network. The pre-Quasar Edition Lux network is round-blocking by construction and is a separate network out of scope.

References

[1] Kushal Babel, Andrey Chursin, George Danezis, Anastasios Kichidis, Lefteris Kokoris-Kogias, Arun Koshy, Alberto Sonnino, and Mingwei Tian. Mysticeti: Reaching the limits of latency with uncertified DAGs, 2024. https://arxiv.org/abs/2310.14821.

[2] George Danezis, Eleftherios Kokoris-Kogias, Alberto Sonnino, and Alexander Spiegelman. Narwhal and tusk: A DAG-based mempool and efficient BFT consensus, 2022. EuroSys 2022; https://arxiv.org/abs/2105.11827.