

Vignesh Kumar A



Database engineer · PostgreSQL internals and distributed databases

Chennai, India · vignesh@sudohulk.com · sudohulk.com · [GitHub](#) · [LinkedIn](#)

SUMMARY

Database engineer with four years at Zoho. Works on the internals of a horizontally scalable distributed PostgreSQL in C: the shard lifecycle operations that extend, retire and rebalance ranges while the cluster serves, each one rehearsable against live metadata before it runs, and a read-only verifier that checks shard placement, replication and two-phase commit state on every node. Before that, built the sync platform of a point-of-sale product in Java on Kafka and MySQL, where the hardest bugs were questions of durability, isolation and ordering. Writes plain-language explanations of PostgreSQL internals.

EXPERIENCE

Member of Technical Staff

Aug 2022 – present

Zoho Corporation · Chennai

- On the distributed database team, built a read-only consistency verifier for the cluster, modelled on PostgreSQL's amcheck (50 checks, 17K lines of C and SQL). It checks shard placement, colocation, replication residue and two-phase commit state on every node over asynchronous libpq with a deadline per node, fingerprints each finding so a fault keeps its identity across scheduled runs, and proves every check with a fault-injection test on a 10-node regression cluster.
- Rewrote the operation layer for range-partitioned shards in C, folding 2,800+ lines of user-defined function code into one operation model with cutoff expressions and operation-aware scheduling, and gave every range operation a dry run that shows what it would change in live metadata before a single shard is mutated.
- Extended the team's blue-green query validator (Java), used to test major-release upgrades, to export each coordinator's planning and execution time for every validated query, tagged by release for query-by-query comparison.
- Built the verifier's test and tooling stack: a tiered test framework (unit, 10-node integration, upgrade and coverage) behind one CI gate, a build-time lint that rejects any SQL with side effects, and a Python MCP server so engineers can run checks and triage findings from an AI assistant.
- On the point-of-sale platform, designed and built the event-driven sync layer in Java on Apache Kafka and MySQL: one event per change instead of $M \times N$ API calls, which cut upstream API load by 99% and brought new-service onboarding from weeks to under 5 minutes through a declarative framework adopted by multiple teams.
- Made sales durable across server crashes with a transactional outbox written in the same MySQL transaction as the sale, with background retry: 99.9% consistency at 1K+ transactions per minute between the registers and the finance platform.
- Split the distributed cache (JVM + Redis) into isolated segments with version-based invalidation, raising hit rate from 84.3% to 98.1% and removing 131M database queries per week.
- Delivered real-time push to 1,000+ concurrent WebSocket endpoints, cutting data conflicts by 98%; designed a three-level circuit breaker that opens within 15 seconds of downstream slowness; and built correlation-ID tracing across asynchronous threads that cut root-cause analysis from 3+ days to minutes.
- Wrote the team's code-review guidelines, published 12+ internal engineering blog posts, and mentored 3 engineers.

PROJECTS

- **Hybrid Storage Engine (C).** A small key-value engine that keeps values in an append-only log and keys in a B-tree (inspired by WiscKey and Bitcask), built to feel the trade-off between B-tree and LSM-tree designs rather than read about it.
- **Decentralized P2P Network (Rust).** Peer-to-peer messaging network on libp2p delivering messages in under 50 ms across 100+ nodes, with Ed25519 peer identity and Noise-protocol encryption.

SKILLS

Databases & storage	PostgreSQL internals (WAL and crash recovery, MVCC and tuple visibility, two-phase commit, logical decoding, background workers, shared memory, query planning), C extensions, index and consistency verification, B-tree and LSM-tree storage, MySQL, Redis
Distributed systems	sharding, distributed query execution, replication, two-phase commit, consensus (Raft), event-driven architecture, Apache Kafka, transactional outbox, circuit breakers, high availability
Languages	C, Java (SE/EE), Python, SQL, Rust, TypeScript, Shell
Tools & practice	Linux, LLDB/GDB, regression and TAP tests, fault-injection testing, CI, tracing, code review, technical writing, AI-assisted development (Claude Code, MCP)

EDUCATION

B.Sc., Computer Science, Sri Muthukumaran Arts & Science College, Chennai

Jun 2019 – May 2022

WRITING

Long-form explanations of database internals at sudohulk.com/writing