Sicheng Pan

sicheng0129@gmail.com | +1 (510) 301-0622 | LinkedIn

Education

University of California, Berkeley 2023 – 2024
M.S. in Electrical Engineering and Computer Sciences, GPA: 3.94/4.0
Thesis: *Extensible Rule Language for Query Optimizer*, advised by Alvin Cheung

University of California, Berkeley 2019 – 2023
B.A. in Computer Science and Statistics, GPA: 4.0/4.0
Highest Distinction in General Scholarship · Honors in Computer Science

Experience

Chroma Aug 2024 – Present
Member of Technical Staff, Data Plane

- Integrated 4-bit RaBitQ quantization into the SPANN index, reducing compaction time from 20 min to 2 min per 1M vectors (1536-dim) with 5× lower memory usage, enabling collections to scale from 5M to 50M with 40ms query latency and >90% recall@10.
- Designed and implemented a hybrid search API supporting composable KNN expressions (e.g., reciprocal rank fusion), sparse vector indexing (BM25, SPLADE) via Block-Max WAND, achieving sub-100ms latency at 1M scale. Shipped end-to-end across engine, API, Python/JS/Rust clients, and docs.
- Co-led rewrite of the distributed frontend from Python to Rust (tokio/axum), owning the read path (query plan serialization, executors). Throughput increased from 800 to 6,000+ RPS on 16 cores with latency spikes eliminated. Also shipped as the new Rust-based local client.
- Designed a serializable query plan and pushed query orchestration from the frontend to the query server, reducing network round trips from 3 to 1 per query and eliminating large intermediate data transfers.
- Implemented instant collection forking with copy-on-write semantics, enabling users to checkpoint datasets and share sample collections without incurring storage copy.
- Built regex query support via two-stage approach: extracting required literals from regex patterns to narrow candidates via the trigram index, then brute-force matching survivors, achieving sub-100ms latency at 1M scale.
- Implemented efficient limit/offset pagination, negation filters using roaring bitmaps, CMEK encryption (GCP), and group-by deduplication for chunked-document search results.

Duolingo Summer 2022
Software Engineer Intern, Data Infrastructure and Experimentation Team

- Implemented approximate query pipeline on BigQuery for the analytics dashboard, saving >50% query time at <1% uncertainty.
- Implemented caching mechanism for common queries based on historical frequencies (AWS, Jenkins), saving >80% time for analysts.

R-Polars Project Summer 2023
Contributor, Google Summer of Code

- Exported Polars features to R including streaming I/O in Apache Parquet and Arrow formats.
- Refactored error handling with recoverable errors from Rust and implemented background query pipeline via multi-threading, saving >50% user wait time.

Research

QED: A Powerful Query Equivalence Decider for SQL 2021 – 2024
UC Berkeley EECS, advised by Alvin Cheung · Published at VLDB 2024

- Co-developed QED, a SQL query equivalence prover in Rust using a novel formalism (Q-expressions) under bag semantics with a complete checking algorithm for a general query fragment parameterized by first-order theories.
- Verified 299/444 query rewrite pairs from Apache Calcite and 979/1287 from CockroachDB, more than 2× the coverage of prior state-of-the-art.

Languages: Rust, Python, SQL, R, Nix
Tools & Infrastructure: Linux, Git, gRPC/Protobuf, tokio, axum, AWS, GCP, Kubernetes, Docker