LEON DO

Garden Grove, CA · leontdo2004@gmail.com · Portfolio: koishite.ru

EDUCATION

California State University of Long Beach Expected Grad: May 2027
Bachelors of Science in Computer Science, Minor in Mathematics
Relevant Courses: Data Structures & Algorithms, Optimization, Systems Programming, Computer Architecture, Deep Learning

CSULB Programming Team
Authored detailed editorial write-ups and solution breakdowns for advanced algorithms, used by peers in competitive programming training.

EXPERIENCE

Beach Investment Group Long Beach, CA
Quantitative Developer Intern June 2024 - May 2025

- **Developed predictive equity models** (regression, neural networks, factor-based) for analysts to use.
- **Accelerated model training 60%** by offloading to GPU with CuPy; re-implemented preprocessing in Polars to scale efficiently on multi-GB time-series datasets.
- **Improved signal orthogonality** through custom factor transformations, increasing independence and boosting clarity in predictive regressions.
- Delivered reproducible SPY model outputs that secured **$115K in funding** (CFAOC RFP 2024).

PROJECTS

Chewse | Dual-User Swiping for Date Spot Matcher (Full-Stack)

- Built Redis-backed session system with TTL expiry, enabling ephemeral shared swipe rooms with low-latency sync.
- Cut Google Places API usage by ~40% with caching, request coalescing, and adaptive rate limiting.
- Built a partner-adaptive re-ranking engine, improving match conversion ~25%.
- Prototyped Go microservice adapter with goroutine pooling for throughput gains.

Tori | High-Performance Event Dispatcher (Golang)

- Designed burst-tolerant dispatcher with sliding-window rate limiting and timestamp pruning.
- Built lock-free ring buffers using bitmask indexing to cut CPU and avoid branching.
- Minimized GC with preallocated structs and slice reuse in hot paths.
- Added cancellation-safe goroutines for graceful shutdown and eliminated dangling processes.
- Preserved real-time responsiveness by evicting stale jobs and prioritizing fresh events (LRU backpressure).

High-Performance Monte Carlo Benchmarking Engine (C++)

- Achieved **10× speedup** through **AVX2 vectorization**, branch elimination, and cache-aligned memory.
- Cut cache miss rate 25%→5% with thread-local bump allocators & 64B alignment.
- Eliminated heap contention using thread-local RNG and memory pools for lock-free parallelism.
- Profiled with perf stat to analyze cache hierarchies (L1–L3) and guide further tuning.
- Shipped telemetry pipeline (Parquet → ClickHouse → Grafana) for interactive perf analysis (CI/CD).

SKILLS

Programming Languages: *C++, Python Typescript, Rust, Golang, Javascript, SQL*
Systems & Performance: *SIMD, Linux, Multithreading, Memory & Cache Optimization, Docker, Redis, Websocket*
Toolings & Infra: *REST APIs, Ninja, GitHub, CMake, Bash, Polars, Pandas, Clickhouse, Grafana, Parquet, CI/CD*
HPC & ML / AI: *Cuda, PyTorch, TensorFlow, SciPy, Numpy*
Software Engineering: *System Design, Object-Oriented Design, Data Structures & Algorithms, Profiling, Benchmarking*