Theeraj Chandra

+1 (226) 899-7039 | tchandra@uwaterloo.ca | linkedin.com/in/theeraj | github.com/theerajchandra

Education

University of Waterloo Sep 2022 – Apr 2027

Bachelor of Applied Science in Computer Engineering (Honours)

- **Relevant Coursework:** Distributed Computing, Database Systems, Operating Systems, Computer Networks

Technical Skills

Languages: Java, TypeScript, Python, Ruby, JavaScript, Go, Rust, C++, SQL, Bash
Frameworks & APIs: Ruby on Rails, Node.js, React, Next.js, GraphQL, REST, gRPC, MCP
Infrastructure & Tools: AWS (Lambda, S3, Bedrock), Docker, Kubernetes, Buildkite, Prometheus, Grafana
Databases: PostgreSQL, MySQL, Redis, DynamoDB

Experience

Amazon Sep 2026 – Dec 2026

Software Engineer Intern (Alexa+) *Vancouver, BC*

- Built PromptAtelier, a **serverless** prompt optimization platform in Java and TypeScript cutting failure-to-fix time from **2 days to 2 hours** and saving ∼4000 engineer hours (**$400k**) yearly across **3** Alexa teams.
- Engineered an autonomous **MCP agent** that resolved **80% of critical harness failures** within 3 iterations, **cutting inference spend** per round by **85%** through selective reruns of failing turns and 60+ regression guards.
- Re-architected a synchronous Java inference tool into an asynchronous **AWS Lambda/SQS pipeline**, reducing runtime for 500 inferences from **40 minutes to 4 minutes**.
- Designed versioned DynamoDB/S3 persistence with **single-writer leases and snapshots**, enabling engineers and AI agents to safely collaborate across **50+ investigations**.

StackAdapt Jan 2026 – Apr 2026

Software Engineer Intern (QE Infra) *Toronto, ON*

- Built an agentic **TypeScript/Nx**-based AI PR-review CLI that calls **Claude on Amazon Bedrock** to review failing E2E PRs and emit structured analysis comments, with **MCP**-based tool calling, **Zod** validation, SHA-256 caching, and OIDC-based AWS authentication.
- Diagnosed a **PostgreSQL** bottleneck on a **45M-row, 19GB** table, identifying excess heap fetches and driving a partial-index optimization that reduced lookup latency from **25s to 0.919ms** (**25,000x speedup**).
- Designed **closed-loop learning** for an AI autofix pipeline, using PR outcomes and reviewer feedback to guide future fixes and avoid repeated errors.
- Unified **GraphQL logic across 8 integration specs**, eliminating **1,240 duplicated lines** and standardizing metric definitions.

SuperWorld Jan 2025 – Apr 2025

Software Engineer Intern *Miami, FL*

- Shipped geolocation-based **virtual check-in** for SuperWorld Map, using the **Haversine formula** to verify proximity within **300m** and prototype location-aware engagement.
- Built core UI flows for a **LLaMA 3**-powered conversational map, enabling location discovery, contextual questions, and product navigation through natural language.
- Developed a founder-facing **React + TypeScript** analytics dashboard with parallel API calls, live KPI cards, and **Recharts** visualizations to guide product decisions.

Projects

Distributed ML Training Runtime | *Rust, Python*

- Built a **fault-tolerant distributed runtime** in **Rust/Python**, coordinating async workers via **gRPC** for sharded data loading, checkpointing, and automatic rebalancing, with **PyO3** bindings exposing a PyTorch-compatible API.
- Implemented a two-phase **commit checkpoint protocol** with atomic writes to **S3-compatible** object storage and end-to-end **Prometheus** observability, ensuring zero data corruption across fault injection.

Zero-Allocation Limit Order Book | *Rust*

- Engineered a matching engine from scratch in **Rust** achieving **160ns** average latency and **6.2M ops/sec**.
- Architected a **slab/arena** allocator and intrusive data structures, eliminating heap churn and lock contention to ensure deterministic $O(1)$ allocation and cache-friendly $O(\log P)$ matching.