

Education

University of Southern California — M.S. in Computer Science
Beijing University of Technology — B.E. in Computer Science

Jan 2024 – Dec 2025
Jul 2019 – Jun 2023

Work Experience

- **Alluxio Inc — Full-time Software Engineering Intern** Jul 2022 – Nov 2023
 - **Automated Testing Platform:** Co-developed a Go-based cluster-level testing platform; managed AWS infrastructure (EC2, S3, IAM) via **Terraform**, orchestrated deployment and test tasks through **dependency graph scheduling**, enabling one-click provisioning with arbitrary node counts and concurrent test execution; integrated **async-profiler** for automated performance data collection.
 - **Stress Testing & Performance Analysis:** Designed throughput and latency stress tests for large sequential writes and small random writes under varying Worker configurations; identified lock contention hotspots via **flame graph analysis**, uncovering **write amplification** flaws; drove lock granularity optimization and I/O refactoring; parallelized data preparation to multi-threaded generation, resolving metadata bottlenecks, **improving efficiency by 10x**.
 - **Journal Recovery Fault Tolerance:** Developed a **Protobuf binary log parsing tool** for the Alluxio Journal system, deserializing unreadable logs into structured format for fault diagnosis; designed a **segmented recovery strategy** prioritizing intact segments before isolating corrupted sections (latency jitter, concurrency conflicts), enabling recovery from previously unrecoverable crash scenarios.
 - **JMH Microbenchmarks:** Authored 50 JMH microbenchmarks for core read/write and metadata paths, establishing a continuous performance regression baseline.
 - **Rust Prototype:** Initiated a Rust rewrite of the Worker local cache module, integrating via JNI to eliminate JVM GC overhead and reduce tail latency, validating feasibility.

Personal Projects

- **AI-Driven Idle RPG**
 - Independently developed an AI Agent-driven idle RPG with a **dual-layer decoupled architecture**: engine exposes scene snapshots and candidate actions via interfaces, Agent returns decisions for engine to execute; core engine logic is **pure-functional**, enabling independent testing and **state tracking** without LLM dependency.
 - Built an Agent **Perceive–Decide–Remember–Reflect–Goal Update** cognitive loop: **vector retrieval** constructs context from relevant memories; decisions scored by **character-subjective attractiveness** with weighted random sampling to counter LLM optimization bias, producing **differentiated behavior** per personality; reflection triggered at emotional peaks, goal stagnation, and fixed intervals, driving a **three-tier goal state machine**.
 - Addressed **hallucination accumulation along memory→retrieval→goal chain** with multi-layer defenses: **prompt boundary constraints**, **Jaccard echo detection** filtering fabricated memories, **goal stagnation detection** forcing abandonment of stalled goals, and **fallback filtering** preventing degraded outputs from polluting long-term memory.
- **News Aggregator**
 - Built a multi-source scraping pipeline using Playwright with **RSS subscription + local hash table deduplication** for incremental crawling, per-site rate limiting, and full-text storage in PostgreSQL; constructed an **LLM processing pipeline** for summarization, translation, classification, and keyword extraction.
 - Implemented semantic vector search via **pgvector** combined with full-text search for multi-dimensional querying; built a Go API layer for search and filtering endpoints.
- **Key-Value Database**
 - Developed a multi-threaded asynchronous key-value database in Rust based on the Bitcask model, supporting efficient append-only writes and index-based fast reads.

Publications & Research Experience

- **Elastic VM Consolidation in Cloud Data Center Networks** Jul 2022 – Jul 2023
 - **[IEEE TPDS, 2025]** Online Elastic Resource Provisioning With QoS Guarantee in Container-Based Cloud Computing
 - Proposed a **two-tier distributed VM consolidation architecture** replacing centralized scheduling: partitioned the data center into regions, each governed by a **leader node** making autonomous consolidation decisions based on CPU, memory, and bandwidth utilization; **leaders negotiated cross-region transfers** based on residual capacity and migration cost when intra-region resources were insufficient. This design evolved into the two-tier prediction architecture (PSF_{ω} / PSL_{ω}) in the paper.
 - Co-designed an ω -**step sliding window load prediction mechanism** forecasting resource demands at host and leader levels, combining predictions with current utilization to trigger consolidation or scaling decisions, reducing performance jitter from reactive migrations.
 - Built a data center simulation prototype in Go, modeling resource contention, migration bandwidth and time costs for a 128-host, 100-tenant fat-tree topology; designed a host elasticity index quantifying elastic growth potential across computation and communication dimensions.
- **Network-Level Traffic Signal Coordination Optimization** Jan 2022 – Jul 2022
 - **[IEEE TITS, 2023]** Network-Level Traffic Signal Cooperation: A Higher-Order Conflict Graph Approach
 - Implemented intersection signal timing optimization via conflict graph modeling, representing each traffic flow as a graph node and solving the **Maximum Weight Independent Set (MWIS)** for optimal green-light sequencing.
 - Built a 4×4 (16-intersection) grid simulation in **SUMO**; implemented k -order neighborhood expansion to incorporate upstream/downstream congestion into signal decisions; demonstrated superiority over fixed-timing and actuated control baselines.

Skills

- **Languages & Tools:** Java, Rust, Go, Python, TypeScript | AWS, Terraform, Docker, Linux, PostgreSQL, Hadoop
- **AI / Data:** LLM, Prompt Engineering, pgvector, Mem0, async-profiler