Umang Goel

623-286-8901 | goelumangjob@gmail.com | linkedIn | goelumang.com

SUMMARY

Software Engineer with strong foundations in data structures, algorithms, concurrency, and systems programming. Proficient in C++, Java experienced optimizing performance-critical components through profiling, memory analysis, and concurrency control. Demonstrated ability to improve latency and memory efficiency in distributed systems. Passionate about building high-performance, scalable storage and data systems.

EDUCATION

Arizona State University **Aug 2022 - May 2026**

Bachelor's, Computer Science *Tempe, AZ*

- **GPA:** 3.8/4.0
- **Coursework:** Data Structures & Algorithms, Software Engineering, Distributed Systems, Software Analysis & Design, Cloud Engineering, Object-Oriented Programming

TECHNICAL SKILLS

- **Languages**: C, C++, Java, Python
- **Backend & System**: gRPC, Flask, FastAPI, REST APIs, Distributed Systems, Concurrency, Performance Optimization, Asynchronous Processing
- **Data and Storage**: SQL, PostgreSQL, Redis, RabbitMQ, ETL Pipelines, Data Modeling, Query Optimization, Caching Strategies
- **DevOps and Cloud**: Docker, Kubernetes, GKE, AWS, CI/CD, GitHub Actions, Linux, Scalability, Fault Tolerance, Load Balancing, System Design, TDD, Agile/Scrum

PROFESSIONAL EXPERIENCE

Honeywell and ASU Innovation Hub **Aug 2025 - Dec 2025**

Software Developer Extern *Tempe*

- Developed performance-sensitive backend components in C++ and C#, reducing end-to-end latency by 25% through architectural improvements and efficient data handling.
- Designed concurrent ETL pipelines with Airflow to ingest and transform 50K+ records, managing PostgreSQL schemas and Redis NoSQL databases for effective database management and enabling reliable, repeatable processing for analytics and model training environments.
- Profiled compute-intensive modules using memory and CPU profiling tools, reducing memory usage by 40% and improving runtime efficiency under high-load conditions.
- Applied concurrency control and thread-safe design patterns, ensuring system stability and eliminating race conditions in multi-threaded environments

DMML Lab, Arizona State University **Dec 2024 - May 2025**

Software Developer Intern *Tempe*

- Designed and implemented distributed backend services in C++ and Java, applying microservice and gRPC-based communication patterns to reduce p95 latency by 30% across real-time workloads.
- Built high-throughput data pipelines processing 300K+ records, applying concurrency control, retry mechanisms, and memory-efficient data handling to improve analytical accuracy by 15%.
- Developed event-driven systems using RabbitMQ with asynchronous consumers, sustaining 500+ events/min with fault tolerance and resilient load distribution.
- Improved system reliability and maintainability by introducing unit testing (TDD), automated CI/CD pipelines, and performance monitoring, reducing production defects and accelerating release cycles.

PUBLICATION

- Saketh Vishnubhatla, Alimohammad Beigi, Rui Heng Foo, Umang Goel, Ujun Jeong, Bohan Jiang, Adrienne Raglin, Huan Liu.An Interventional Approach to Real-Time Disaster Assessment via Causal Attribution. ACM International Conference on Information and Knowledge Management (CIKM 2025), Demo Track..*https://dl.acm.org/doi/10.1145/3746252.3761487*

PROJECTS

Real-Time Event Processing Engine **2025**

- Designed and implemented a high-performance event processing system capable of sustaining 10K+ events/sec with sub-50ms latency.
- Implemented worker coordination, retry logic, and memory-efficient data structures to maintain correctness under load.

Causal MMD **2025**

- Designed scalable Flask application integrating 1.03M+ records with ensemble classifiers, achieving 92% accuracy for real-time disaster response coordination
- Applied performance optimization through async I/O and batch processing with connection pooling, scaling throughput to 10K+ points/min while reducing latency by 30%.
- Built thread-safe REST API with versioned endpoints and OpenAPI/Claude docs, containerized with Docker and deployed via CI/CD with 95%+ test coverage.