# Yuqi Xiang

https://yuqixiang.info/

Email : yuqi.xiang@outlook.com
Mobile : +1-412-627-3352


## Education

- **Carnegie Mellon University** — Pittsburgh, PA
  *Master of Science in Information Networking; GPA: 4.00* — *Aug. 2024 – Dec. 2025*
  *Related Courseworks: Distributed Systems, Database Systems, Intro to Computer Systems*
- **Nanjing University** — Nanjing, China
  *Bachelor of Science in Computer Science; GPA: 4.70/5.00, Rank: 1/115* — *Sept. 2020 – June 2024*
  *Honors: Outstanding graduates, China National Scholarship (Top 0.2% nationwide)*

## Experience

- **TikTok** — Shanghai, China
  *Software Development Engineer Intern (Golang; Thrift, Protobuf, GORM, ServiceMesh)* — *April 2024 – Aug. 2024*
  - Developed a data access control backend for data usage limitation using Golang, **Apache Thrift** for RPC framework, **Protocol Buffers** for data serialization, and **Hertz** and **Kitex** for building **microservices**.
  - Built a **MySQL**-based distributed database with multi-region support using **GORM**, optimized database performance with **Redis** caching. Improved performance and stability by transforming from cronjobs to asynchronous order processing with **RabbitMQ** transactional messages, reducing query delay by **90%**.
  - Integrated a platform for user registration for data permissions, utilizing **Elasticsearch** for search functionality and **Service Mesh** for network proxies, and **Kubernetes** for container orchestration, supporting intercontinental RPC microservice with more than **4** virtual regions including SG, US, EU, ROW etc.
- **UC Berkeley** — Berkeley, CA
  *Research Assistant (Python; PyTorch, Data Parallelism, Llama, Azure)* — *Aug. 2023 - Jan. 2024*
  - Introduced PhyGrasp, which integrates large language models (Llama) and vision-language models (PointNeXt), combining language instructions and 3D shapes to generate contact-aware manipulation strategies for robots. It achieves about a **10%** higher success rate compared to known strategies such as GraspNet.
  - Constructed PhyPartNet, a comprehensive **dataset** containing **195K** object instances with their detailed physical properties and language descriptions, providing a rich variety of object types and physical scenarios.
  - Trained a bridge model incorporating physical commonsense reasoning with strong generalization capabilities. Utilized **data parallelism** with **Azure** Clouding Computing, speeding up by **80%**.

## Projects

- **Online Psychological Consulting Web Service** *(Java; Spring Boot, Hibernate, Kafka)* — *Aug. 2022 – July. 2023*
  - Developed a web service for online consulting using **Java** with **Spring Boot**, **Hibernate**, and **ReactJS**.
  - Created an **Ajax-based** frontend using **Yarn**, **Webpack**, **ReactJS**, and **Redux** for middleware and reducers, incorporating WebSocket for real-time chatting and notifications.
  - Automated system deployment using **Docker** and **Nginx** on AWS EC2/ECS, with **AWS ELB** as load balancer.
  - Designed strategies enhancing read-update performance by **50%** with **Apache Kafka** for log/event analysis.
- **BusTub** *(C++; Relational Database, Indexes, Hash, Lock Manager, MVCC)* — *Aug. 2024 – Present*
  - Implemented a **buffer pool manager** moving data back and forth between disk and memory.
  - Developed a disk-backed hash index based on the **extensible hashing** scheme, reducing query delay by **70%**.
  - Built **volcano style query executors** (sequential scan, hash-join, aggregation, etc.).
  - Implemented a **lock manager** supporting different isolation levels on transactions.
- **Custom Linux Kernel Module Development** *(C; Concurrency, Thread Schedule, Perf)* — *Jan. 2021 – July. 2021*
  - Used the Buddy system (page-sized allocations) and Slab strategy (small allocations) to implement **multi processor safe** memory allocation and recovery, utilizing **producer-consumer model** for efficiency testing.
  - Managed kernel multi-threading, by implementing synchronization mechanisms like **Spinlocks** and **Semaphores**, and a **Multi-level Feedback Queue thread scheduling algorithm** to enhance thread coordination.
  - Designed performance benchmarks and reduced space amplification by **2x** using tools like **perf** and **stress-ng**.

## Programming Skills

- **Languages**: Golang, C/C++, Python, Java, JavaScript, Swift, SQL, HTML5, CSS
- **Cloud Technologies**: AWS, GCP, Azure, Kubernetes, Nacos, Docker, Microservices, Service Mesh