Yuhan Tan

(607)–233–3661 - [yuhant0712@gmail.com](mailto:yuhant0712@gmail.com) - linkedin.com/in/yuhan-aaron-tan

**EDUCATION**

**Cornell University Ithaca, US**

*Master of Information Science | GPA: 3.81/4.0 08/2023 - 12/2024*

**University of Liverpool Liverpool, UK**

*BEng in Computer Science & Electronic Engineering (First Class Hons) | GPA: 3.8/4.0 09/2019 - 07/2023*

**SKILLS**

**Languages:** Java, Python, JavaScript, SQL, HTML, CSS, TypeScript, R, Shell

**Framework & Tools:** Spring Boot, Spring Cloud, MyBatis Plus, RabbitMQ, Redis, Kafka, MySQL, Docker, Git, Linux, Vue3, PyTorch, Hadoop, HDFS, JVM, AWS, Spring MVC

**Expertise:** Backend Development, Microservices Architecture, Distributed Systems, RESTful APIs, Web Development, Asynchronous Task Processing, Agile Development, Machine Learning

**EXPERIENCE**

**Cornell University 08/2024 - Present**

*Teaching Assistant Ithaca, NY*

* Guided **20+** students in mastering applied machine learning, improving assignment scores **by 15%**.
* Resolved **over 90%** of student queries through weekly office hours and personalized consultations.

**American Express 02/2024 - 05/2024**

*Machine Learning Engineer Remote*

* Worked with **cross-functional** teams to enhance BERT model by detecting misclassification patterns by clustering.
* Analyzed BERT **internal activations** with Captum to visualize attention scores and uncover error-prone areas.
* Improved model accuracy **by 13%** through using a Specialized Attention Layer and Weighted Loss for key classes.

**NextTier 05/2024 - 08/2024**

*Software Engineer Intern Sacramento, CA*

* Collaborated with front-end teams to reduce API response time **by 95%** with Redis caching and custom serializers.
* Implemented **scheduled cache warming** with Spring Scheduler, improving initial access speed **by 97.7%**.
* Optimized database import for **1M** rows by using **custom thread pools**, reducing time **from 300s to 54s**.
* Led a **team of 3** to implement matching function with edit distance algorithm, reducing matching time **by 81%**.
* Implemented **distributed sessions** using Redis and enhanced **concurrency control** by distributed locks.

**Eth Technology 08/2023 - 12/2023**

*Software Engineer Intern Newark, CA*

* Partnered with senior engineers to develop **scalable streaming microservices** capable of handling **1000+ events** concurrently using Kafka and store events metadata with relational database.
* Designed and built **RESTful APIs** for event consumption, enabling efficient integration with backend systems.
* Conducted **unit** and **integration testing** with **90%** code coverage using JUnit, JMeter, and Embedded Kafka.

**PROJECTS**

**Online Coding Judgement System 09/2024 - Present**

* Designed a scalable modular backend architecture with **Spring Cloud**, separating user, problem, and judgement services. Used **Spring Cloud Gateway** for service aggregation and API routing.
* Built a responsive frontend with **Vue3** and **Arco Design**, supporting search, edit, and submission features.
* Developed a secure code sandbox using **Docker** and **Java Runtime** method for isolated code execution.
* Enhanced sandbox security using **JVM** options(*-Xmx*) to limit memory usage and **Java Security Manager** to restrict system access and execution permission.
* Decoupled workflows using **RabbitMQ** to queue problem evaluation tasks, improving system QPS by **32%**.

**Asynchronous Processing Framework: AaronFlow 03/2024 - 09/2024**

* Developed an asynchronous framework with a two-layer architecture separating scheduling and business logic.
* Designed a **loosely coupled** database schema, enabling efficient task management and fast retrieval via **indexing**.
* Implemented task monitoring and recovery using a **polling** mechanism to track task status and **automated table partitioning** when size thresholds were reached.
* Conducted **stress testing** with wrk and Lua scripts, improving throughput **from 200 to 1000 QPS** by utilizing custom MySQL connection pool and optimizing multi-work coordination.

**Building Data Lakes on AWS 08/2024 - 09/2024**

* Built a scalable data lake using **AWS Lake Formation**, automating data ingestion, cataloging, and **ETL** processes.
* Configured real-time data pipelines with **Amazon Kinesis** and **AWS Glue**, for near real-time analytics.
* Enhanced data lake security with **IAM policies** and **Lake Formation Permissions** for fine-grained access control.

**PUBLICATIONS**

* **Domain Specific AI Segmentation of IMPDH2 Rod/Ring Structures in Mouse Embryonic Stem Cells**. Accepted by *BMC Bioinformatics*; preprint available on *bioRxiv*.