

# GE JIANG (DJANGO)

(470) 230-3964 | [chellyreol@gmail.com](mailto:chellyreol@gmail.com) | [in ge-django-jiang-41371224a](https://www.linkedin.com/in/ge-django-jiang-41371224a) | [egotist0](https://github.com/egotist0)

## EDUCATION

### Georgia Institute of Technology

*Master of Science in Computer Science*

Aug. 2023 - Dec. 2024 (Expected)

Atlanta, GA, USA

### Shanghai Jiao Tong University (SJTU)

*B.S. in Bioinformatics*

Sep. 2019 - Jun. 2023

Shanghai, CN

## SKILLS

- **Languages:** C++, Golang, Python, Java, JavaScript, SQL, HTML/CSS, Shell
- **Frameworks:** Vue, LLVM, Clang, Django, Flask, gRPC, Hibernate, PyTorch, Cutlass
- **Databases:** RocksDB, MySQL, PostgreSQL, MongoDB, Elasticsearch, Redis, HBase, Memcached, Spark
- **Tools:** Git, Docker, Docker Compose, Kubernetes, Helm, AWS, CMake, Bazel, Linux, Nginx, Gerrit, Redmine

## WORK

### Naive Systems

Sep. 2022 - May. 2023

*Software Engineer Intern, Static Code Analysis Group*

- Develop **Static Analyzer** for AUTOSAR and MISRA with **Clang LibTooling**, **Clang AST**, **LLVM** and **Bazel**. Implement rule-specific checkers for **Abstract Syntax Trees** and **runtime behaviors**.
- Improve **Cppcheck** by Python to support **Google C++ Style** on Token Level, support Cross Translation Unit Analysis.
- Integrate these **300+** checkers into a single analyzer by **Golang** and support for **Docker** runs.
- Create test cases with **gtest** to reduce false positives. Engage in full development process with **Gerrit** and **Redmine**.

### NetEase Game, Inc.

May. 2022 - Sep. 2022

*Backend Engineer Intern, Infrastructure Group*

- Improve search speed of the Localization Platform by **90%** (20s -> 2s) by replacing **MongoDB**'s Full-Text Search with **Elasticsearch** and migrating game text to Elasticsearch. Implement asynchronous jobs using **Celery** and **Redis**.
- Develop **80+** REST API in the HR Platform using **Flask**, **MySQL**, **Redis**, **Celery** and **Vue**. Features include: Interviewee Score Statistics; Periodically Synchronize Data; User Permission Management and History Records.
- Develop and maintain a Teams Bot utilizing **Microsoft Bot Framework** and **.NET core**, enabling developers to easily query databases and request internal APIs through natural language, saving **10** working hours per month.

### Baidu, Inc.

Mar. 2022 - Apr. 2022

*Software Engineer Intern, Autonomous Driving Group (Baidu Apollo)*

- Port autopilot middlewares and integrate them into **RTOS** (Ubuntu) and solve the problems of middleware crashes.
- Build System on Chip (SoC) and deploy **QT** platform. Add tool functions for the Scheduler of Baidu Apollo.
- Write **Makefile** for three cross-compilation and test environments. Increase **20%** efficiency.

## PROJECTS

### SLM DB: a Single-Level Merge DB based on LSM and B+ tree

- Design a **LSM**(Log-structured merge-tree) based key-value database, merging the read advantage of **B+ Tree**:
- Use **Bbolt** to maintain an in-memory B+ tree, enabling fast and range queries; Support append writes similar to **LevelDB** to optimize bulk data writes; Set a **Write-ahead log** to ensure data durability; Implement **SkipList** based **Multi-level Memtables** (managed by **Arena Memory Pool**) to guarantee  $O(\log N)$  efficiency; Design a single-level **SSTable** and uses **Selective Compaction** to manage garbage collection and mitigate Write/Space Amplification.

### Bloom Filter plugin for Apache Kvrocks

- Build the Probabilistic Datatypes Module (similar to **RedisBloom**) on Kvrocks, a distributed **key-value database** that uses **RocksDB** as the storage engine and is compatible with **Redis** protocol. Features include:
- Develop **Bloom/Cuckoo Filter** in C++, enable judge the existence and the number of occurrences for given value in  $O(1)$ , setting a controllable trade-off between accuracy and memory consumption.
- Implement **Block Split Bloom Filter** and use **Spatial Locality** techniques, achieve **37%** reduction in memory usage.

### In-silico platform Development

- Build a platform by **Django** to support **SVM** and **GNN** in cancer diagnosis, enable it to support a high QPS over 2000 leveraging **Redis** for caching, **Celery** for asynchronous message processing and **Nginx** for load balancing.
- Set up a **CI/CD** procedure to automate the deployment of the application on 4 **MongoDB Clusters** (gridFS) on **Alibaba Cloud** and implemented a server monitoring system using **Prometheus** and **Grafana**.