HUA ZHAO, M.S CS

hzhaoc.io | LA, USA (Relocatable) | ☎ 781-996-9376 | linkedin.com/in/hzhaoc | ✉ hzhaoconnor@gmail.com

I'm a **data engineer** / **software engineer** with knowledge and experience in computer science and statistics
I'm experienced with LAMP stack: Apache2 + PHP + MySQL, Python, C/C++, Git, DVC

Education Background

Georgia Institute of Technology, M.S, Computer Science, Computing Systems, Jan 2020 – May 2022

- Coursework**:** *Database System Design, Grad Intro into OS, Big Data for Health Informatics, High Performance Computing*

Brandeis University, M.S, Financial Mathematics, Aug 2016 - Dec 2017

- Coursework**:** Computer *Simulation & Risk Assessment (Python), Forecasting in Economics (Python)*

Southwest Uni. of Fin & Eco, B.S, Financial Engineering, Sep 2012 – Jun 2016

- Coursework**:** *Linear Algebra, C++/C, Statistics, Probabilities, Arithmetical Analysis, JAVE EE, JAVE SE*

Work Experience

Cleco Corporate Holdings LLC| Louisiana, USA | 2019.05 – Present

Data Engineer / Quantitative Analyst

- Worked on developing an economic model into a **risk management software**, written in Python. The software will be used by middle and front office users. Did following work up to today:
  - Set up **DVC stages** to store, version model data and reproduce pipelines.
  - Separated source code and production code in **Git**.
  - Wrote **user command interface** of the software instructions, including automation commands for production server.
  - Improved **PCA** in stochastic gas futures modeling, calibrated generations' and customer contracts' Python classes.
  - Added **HTMLs** reports using **jupyter nbconvert** for diagnostics.
  - Built a light-weighted **rDBMS** using **sqlite2** in Python to improve data management and interface with ETRM system.
- Automated daily Credit Risk Reporting: Built several Python programs **scrapping web trade data**, running stochastic simulations, and writing reports. The automated reporting has been utilized by company since Nov 2019 every business day.

Project Experience

Proxy-Cache-Http Web Server

I built a simplified **web server** in **C** that implements data transfers between client and server in 2 phases:

- [Phase 1] Wrote **socket codes** between **Client** and **Server**. Both server and client used **POSIX multithreading** to implement boss-worker mode for client requests.
- [Phase 2] Implemented **Proxy Server** and **Cache Server** where Proxy and Client used the same methods in Phase 1, but Cache server is added to implement local file caching before Proxy querying Http Server. Proxy and Cache use **message queues** to pass commands based on client request, and **POSIX shared memory segments** to transfer cached files. I also did correct global clean-up to ensure no client hung and independent launch of Proxy and Cache.

gRPC DFS

I implemented a basic **distributed file system** between clients and a single server using **C++ gRPC**. The DFS system achieved client whole-file caching and client-server file storage synchronization through following implementations I did:

- Wrote **synchronous gRPC** methods that enable client to fetch, store, or delete files from or into remote server.
- Wrote **asynchronous gRPC** methods called by Client **Async Thread** to wait for server's file change notifications.
- Built another Client **Watcher Thread** to consistently notify server its local file changes.
- Wrote a **lock struct** to ensure One Writer per File in Server directory.

LAMP application: Mo's Mutt House

I wrote a web application using LAMP stack for a dog shelter called Mo's Mutt House in 3 phases:

- [Phase 1] Drafted **Information Flow Diagram** to scope tasks, then the **EER** diagram and **task decomposition**.
- [Phase 2] Mapped EER to **relational schema**; Wrote schema creation, demo data loading in Python, and task SQL queries.
- [Phase 3] Wrote all PHP scripts and configured AMP stack to have a fully functional AMP application. Presented demo and earned 95.38/100 grade.
- [Extra] Deployed the LAMP stack and the demo to a cloud server AWS EC2 on Amazon Linux 2. Check here for project description, and here for demo.

Certificates

- Coursera Machine Learning, *Stanford University, Andrew Ng*.
- Coursera Applied Machine Learning in Python, *University of Michigan*
- Coursera Greedy Algorithms, Minimum Spanning Trees, and DP, *Stanford University*
- Coursera Graph Search ,Shortest Paths, and Data Structures, *Stanford University*
- Coursera Divide and Conquer, Sorting and Searching, and Randomized Algorithms, *Stanford University*