HUA ZHAO, M.S CS

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

Hua Zhao is an engineer and programmer with experience and interest in computer science and machine learning.
Programming languages: **Python, C/C++, PHP, MySQL**
Experience with systems & tools: **LAMP stack, Git, DVC, Linux Shell, LaTeX, Hadoop (MapReduce, pig, hive)**

Education Background

Georgia Institute of Technology, M.S, Computer Science, Computing Systems, 2020.01 – 2022.05

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

Brandeis University, M.S, Financial Mathematics, 2016.08 – 2017.12

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

Southwest Uni. of Fin & Eco, B.S, Financial Engineering, 2012.09 – 2016.06

- 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 / Risk Analyst

- Building an economic model and developing a risk management software based on it, in Python, to do scenario analysis for the company to better manage future cash flow.
  - Set up **DVC stages** to store, version data and reproduce pipelines.
  - Separated source code and model data/parameter code in **Git** for better model version control.
  - Wrote **user CLI** such as automatic model run, results save, results show, etc. for better model utilization.
  - Added **HTMLs** reporting in production using **jupyter nbconvert**.
  - Built a light-weighted **rDBMS** using **sqlite2** embedded in model to communicate data with company's trading system.
  - Participated in modeling: such as PCA in stochastic gas prices, calibrating generation and contract model parameters.
- Automated daily Trade Risk Reporting: Built several Python programs **scrapping web price data**, running stochastic simulations, and writing reports. The automated reporting has been utilized by company since Nov 2019 on each business day.

Project Experience

A simple Web Server: Client + Proxy + Cache + Http

Built a basic web server in **C** that implements file transfers between client and proxy/cache/http server:

- [Phase 1] Wrote **POSIX socket codes** between Client and Server. Both server and client used **POSIX multithreads** to implement boss-worker mode for client requests.
- [Phase 2] Upgraded Server to **Proxy server** and **Cache server**. When Client issue a request to Proxy, Proxy will first pass requests to cache through **message queues**. If Cache server finds the target file cached, it sends the file to Proxy **through shared memories**, then to Client. If the target file is not cached, Proxy continues to visit Http server.

A simple DFS with C++ gRPC

Built a basic **distributed file system** between clients and a central server using **C++ gRPC**. The DFS implemented client whole-file caching, client-server file storage synchronization, server file writer lock through following details:

- Client uses **synchronous gRPC calls** for client to issue file requests to server: fetch, store, delete, status, etc.
- Client uses a thread called Async Thread to handle **asynchronous gRPC calls**, waiting for server's file change notifications, and issuing correspondent file requests to synchronize.
- Client uses another thread called Watcher Thread to consistently check local file updates, and to notify them to server.
- A **lock struct** is added to ensure One Writer per File in server file directory.

A web application for a dog shelter with LAMP stack

Wrote a simple web application using LAMP stack for a dog shelter: Mo's Mutt House:

- [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 PHP and built a locally functional AMP application. Presented individual demo and earned 95.38/100 grade.
- [Phase 4 (extra)] Deployed the demo to my personal website for better presentation, hosted by AWS EC2.
- [Demo]   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*