

# Andrew M. Zhang

(312) 900-2570 | [work@andrewmzhang.com](mailto:work@andrewmzhang.com) | [andrewmzhang.com](http://andrewmzhang.com) | [github.com/andrewmzhang](https://github.com/andrewmzhang)

## EDUCATION

---

### University of California, Berkeley

Aug 2016 – Jun 2021

B. A. in Computer Science GPA: 3.91/4.00

Research In: Machine Learning, Operating Systems, Computer Vision

Programming Skills: C/C++ ◦ Python ◦ Java (Intermediate)

## PUBLICATION CONTRIBUTIONS

---

- Symposium on Cloud Computing 2019: *Centaur - A Practical Serverless Framework for End-to-End ML Workflows*
- NeurIPS Workshop 2018: *A Case For Serverless ML*

## EXPERIENCE

---

### Jump Trading

Aug 2021 - Nov 2024

Quantitative Developer

Jump Execution - JPEX

- Designed a suite of programs that processes and analyzes arbitrary, real-time algo performance data over Clickhouse DB
- Researched and deployed models to predict FX volume time series using market data
- Created tools to semi-systematically update trading/alpha/exchange configs and to monitor these jobs for failure
- Designed a configurable system for receiving and publishing alpha within JPEX hosts and between other business units

Jump Client Trading - JCT

- Designed a fault-tolerant log ingestion tool that separates, verifies, and stores merged-CSV data into Clickhouse DB
- Built a heavily parallelized calculator that pulls data from Clickhouse and prints TWAP, VWAP, and PoV benchmarks
- Researched optimal liquidation of pre-market portfolio based on market and vendor data

### Citadel Securities - Market Making: Client Equities

June 2019 - Aug 2019 | June 2020 - Aug 2020

Quantitative Research Intern 2019, 2020

- Created a trading policy on exhaust orders in a reinforcement learning environment
- Drastically improved trading volume while staying within gross market value and symbol position limits
- Created a fast running trading simulation in Cython
- Researched various position and GMV tilt functions to encode opportunity cost as part of the trading utility function
- Improved predictions for which trades fall into the bottom ventile in terms of profit
- Ported and serialized my experimental Python model and preprocessor to C++
- Researched various methods of explaining the model's decisions and performance with respect to the data
- Significantly improved basis points (profit per execution dollar) of existing strategy

### RISELab, UC Berkeley – Serverless Machine Learning Research. Scalable Computing Research (MPI) Aug 2017 - Dec 2021

*Modin MPI Engine Implementation - Research Thesis. Advisors: Prof. Randy H. Katz, Prof. Anthony D. Joseph*

- A highly scalable drop-in replacement for Pandas
- Implemented an engine for Modin backed by Message Passing Interface (MPI)
- Scaled Pandas operations by the number of cores available in either a single machine or MPI cluster (ie 8 cores = 8x speedup relative to native Pandas)

*Twill Framework Research Project*

- Low-overhead synchronization primitives framework in OpenWhisk for serverless compute of complex tasks (SQL queries, ML tasks, Map-Reduce), eliminating the need for a high-performance coordinator server
- Implemented the Berkeley Big Data Benchmark SQL queries with various serverless and synchronization backends and identified performance issues with each. Backends: AWS Lambdas, OpenWhisk, Twill, Pywren, AWS S3, Redis, Pocket

*Cirrus/Centaur Framework Research Project - Link: [github.com/jcarreira/cirrus](https://github.com/jcarreira/cirrus). Paper links above*

- Low-cost machine learning and hyperparameter optimization framework in C++ that runs serverlessly on AWS Lambdas
- Improved scalability without performance loss using multiple parameter servers (model sharding) for Logistic Regression
- Implemented collaborative filtering model using SGD, helped fix mathematical errors with original implementation
- Made Cirrus converge 70% faster than Spark on datasets, including Criteo Ad-Click Logs and Netflix Recommendations
- Created a Plotly Dash UI to visualize hyperparameter search and kill diverging experiments in Jupyter Notebook
- Implemented logistic regression with map reduce using Pywren (AWS S3, Lambdas) for a comparison baseline

## PERSONAL PROJECTS

---

### Blinn: A C++ Raytracer for Metaballs - [github.com/andrewmzhang/blinn](https://github.com/andrewmzhang/blinn) - <https://youtu.be/dMIecrYXcoE>

- Created a raymarching metaball render from native C++
- Used Nvidia CUDA to aid in parallel renders, 1000x speedup over multithreaded CPU solution
- Renders video of several oscillating metaballs interconnected by springs with diffuse shading and Phong speculars