Zhongjing Wei

github.com/walotta zhongjingwei.me
+1 341-766-8923 zhongjingwei.01@gmail.com

Research Interest

My research primarily focuses on **security and privacy**. Specifically, I am mainly working on building **zero-knowledge proof systems** and **zkSNARKs for ML**.

Education

University of Illinois Urbana-Champaign **Sep. 2024 – Present**
Doctor of Philosophy(PhD) Student in Computer Science *Champaign, IL, USA*

- Works on Zero-knowledge Proof system, advised by Prof. Yupeng Zhang.

Shanghai Jiao Tong University **Sep. 2020 – Jun. 2024**
Honors Bachelor of Engineering (B.Eng. Hons) in Computer Science *Shanghai, China*

- Member of **ACM Honors Class**, which is an elite CS program for top 5% talented students, advised by Prof. Yong Yu.

Research Experience

University of Illinois Urbana-Champaign (SPR&I) **Sep. 2024 – Present**
PhD Student in CS, advised by Prof. Yupeng Zhang *Champaign, IL, USA*

University of California, Berkeley **Jan. 2023 – May. 2024**
Research Assistant, advised by Prof. Dawn Song *Berkeley, CA, USA*

- **Efficient privacy-preserving serving of large language models(LLMs)**
  - This project aims to build a **privacy-preserving distributed system** for efficient **serving of LLMs** based on heterogeneous computing resources (various GPUs, very powerful CPUs, Apple M series) in a WAN setting.
  - Designed a scheduling technique to dynamically route the computing tasks based on the current status of nodes. With this technique, this system can automatically optimize the performance (latency/throughput) depending on the current worker nodes workload and guarantee reliability over unreliable nodes.
  - Designed a **multi-party computation (MPC)** technique that achieves strong privacy guarantees with practical performance. As a result, the performance of the system to serve the original 16-bit LLaMA-7b model with privacy protection improved **from 5 minutes per token to 3 seconds** compared with previous work (PUMA).
- **Comprehensive LLM serving speedtest toolkit**
  - This project is a specialized tool tailored for evaluating the performance of endpoints to serve the LLM, especially for their inference speed and throughput. This is the first tool that offers efficient parallel benchmarking for various LLM services, providing a ready-to-use workload generator, a dynamic dashboard, and comprehensive reports.
  - Designed the concept ”visit” to evaluate the ability of LLM endpoints to serve **multi-round conversations** and defined the serving object used in **visit-level SLO attainment** to evaluate the serving abilities.
  - Designed a **resource-efficient event-loop-based system** to track multiple **streaming** responses. This tool can simulate the largest workload while existing tools like Locust can not even hold 20 requests in the same machine.
  - Designed and deployed a front end with *React* as an off-the-shelf interface to easily start, real-time monitor a benchmark, and view its report. Users can also easily collaborate through the share feature in the WebUI.

- **Out-of-box multi-participant distributed application SDK**
  - The project aims to offer an out-of-the-box abstraction for developers to **build secure distributed applications**.
  - Designed a new **domain-specific language (DSL)** upon TOML, especially for defining decentralized applications.
  - Designed a novel **distributed execution engine** to run defined applications with access control, such as pub-key authentication, and offered necessary features like sharing data, communicating with multiple participants.
  - Key contributor to this project and deployed example application of federated learning with a single TOML file.
- **LLM-based generated Knowledge Graph for DeFi**
  - This project aims to generate a **knowledge graph** from open-source white papers and blogs about **decentralized finance** by extracting entities and relations from these documents with NLP technologies.
  - Designed a system offering an **LLM-based solution** for generating knowledge graphs from documents.
  - Prompt engineered to use LLM as an engine for **information extraction** and **Named Entity Recognition (NER)**. Specifically, use NER to structure the input documents to entities and relations and build a knowledge graph from extracted relations. The system can extract information from raw documents with increased accuracy **from 4% to 63%** when compared to traditional NLP works (Stanford OpenIE and AllenNLP).
  - Benefit from a great number of open-source documents about DeFi (this project uses about 400 documents to build the knowledge graph), this project can create an integrated and accurate knowledge graph automatically and can dynamically modify the graph when there is new progress in this area.

Shanghai Jiao Tong University (SAIL lab) **Jul. 2022 – Jul. 2023**
Research Assistant, advised by Prof. Chao Li *Shanghai, China*

- **Edge streaming graph computing**
  - The goal of this project is to build a graph computing framework in an edge environment. In this project, I implemented various graph computing algorithms on Nvidia Orin by defining CUDA kernels and optimizing the algorithms based on the characteristics of edge computing, such as shared memory.

Selected Projects

Mx Compiler (Coursework of *Compilers*) | *Java, LLVM IR, RISC-V, Antlr* **2022**

- Developed a compiler that compiles C-and-Java-like language (Mx*) to Assembly.
- Implemented the module that converts the AST to **LLVM IR** and verified the correctness of IR outputs with clang.
- Implemented **optimizations** such as constant propagation and loop unrolling, that outperformed GCC O1.
- As a single person project, this compiler includes **15k** lines of code.

RISC-V CPU (Coursework of *Computer Architecture*) | *Assembly, FPGA, Verilog, RISC-V* **2021**

- Designed a RISC-V CPU that supports RV32I Base Integer Instruction Set V2.2 (2.1-2.6).
- Designed a modified **Tomasulo** structure that can dynamically schedule instructions and allow **out-of-order execution**.

Teaching Eeperience

Teaching Assistant (Practice and Principles of Computer Algorithms) **Summer 2022**
Teaching Assistant (Data Structure) **Spring 2022**
Teaching Assistant (C++ Programming) **Fall 2021**