

## Education

- 09/2018 – 12/2021      Stony Brook University  
**M.S. in Computer Science**  
**Overall GPA 3.9 / 4.0**  
**Relevant Courses:** Operating Systems (A), System Security (A),  
Data Science Fundamentals (A), Analysis of Algorithms (A),  
Theory of Database Systems (A)
- 09/2014 – 07/2018      University of Electronic Science and Technology of China (UESTC)  
**B.E. in Computer Science and Technology**  
**Overall GPA 3.78 / 4.0**

## Professional Experience

- Research Engineering Intern, Capacity Engineering & Analysis, Facebook**      Menlo Park, CA  
→ *Warm Storage Service Time Modeling and Attribution*      05/2021 – 08/2021
- ◆ Evaluated the existing in-use I/O service time models, and discovered that they have consistently overestimated the Warm Storage service time by 5~30%;
  - ◆ Conducted in-depth analysis of LBA offset distribution of the Warm Storage workloads and performed fine-grained benchmarks on 10TB and 14TB hard drives, thereby finding the weighted overall I/O throughputs of the Warm Storage workloads;
  - ◆ Proposed adjustments to service time models that reduced up to 70% of the prediction errors;
  - ◆ Trained ML-based models to predict I/O service time in Warm Storage clusters, achieving error reduction of 40~77% compared to the original model.
- Research Assistant, File Systems and Storage Lab (FSL)**      Stony Brook, NY  
→ *MFCFS: Logic Model Checking of File Systems*      03/2020 – 05/2021
- ◆ Developed MCFS: a file system checking framework that executes permutations of system call sequences on multiple file systems and performs N-version programming on their behaviour. (Contributed 6,000 lines of C/C++, leader of a team of 6 students)
  - ◆ Developed VeriFS: a full-featured user-space in-RAM file system with the ability of saving and restoring its entire states. (2,500 lines of C/C++)
  - ◆ Annotated NFSv4 RFC documents to help the NLP team parse the NFS protocol text.
- *TC-NFS: NFSv4 with Transactional Compound*      06/2019 – 03/2020
- ◆ Implemented the undo executor and the lock manager to provide ACID features for NFSv4 compound execution. My contributions included 6,000 lines of C and C++ code.
  - ◆ Revised and optimized the transaction layer the project added to the NFS-Ganesha server.
  - ◆ Performed a series of benchmarks to evaluate the performance, which revealed that the transaction layer only introduced modest overhead to the NFS server.
- Remote Part-time Software Developer, Woohelps**
- *Overseas New Life Website Backend Development*      10/2017 – 04/2018
- ◆ Developed a multi-client WeChat bot platform to provide auxiliary services for online merchants. (2,000 lines of Python, in Django framework)
  - ◆ Implemented the homepage recommendation API and the location based merchant selection API in Overseas New Life's backend service. (1,000 lines of Python, in Django framework)

## Teaching Experience

- *Graduate Teaching Assistant, Stony Brook University, 09/2020 – 12/2020*
- ◆ CSE 320 – System Fundamentals II, Fall 2020, Professor: Eugene Stark

→ **Graduate Teaching Assistant, Stony Brook University, 09/2018 – 05/2019**

- ◆ CSE 306 – Operating System, Fall 2018, Professor: Eugene Stark
- ◆ CSE 306 – Operating System, Spring 2019, Professor: Michael Kifer

## **Publications**

→ **Model-Checking Support for File System Development**

Wei Su, Yifei Liu, Gomathi Ganesan, Gerard Holzmann, Scott Smolka, Erez Zadok, et al.  
The 13th ACM Workshop on Hot Topics in Storage and File Systems (HotStorage '21)

→ **Supporting Transactions for Bulk NFSv4 Compounds**

Wei Su, Akshay Aurora, Ming Chen, and Erez Zadok  
13<sup>th</sup> ACM International Systems and Storage Conference (SYSTOR 2020)

## **Academic Projects**

→ **CPENC System Call (Individual Project, CSE-506 Operating Systems)**

A Linux kernel module that provides a new system call “cpenc”, which encrypts and decrypts files with user-specified secret keys. It supports various symmetric cipher algorithms in the kernel and allows the user to specify the length of the key to achieve customized levels of security. The project has approximately 1,300 lines of kernel code and 550 lines of user code in C. It also contains over 800 lines of Bash shell script for testing.

→ **Backup File System (BKPFs) (Individual Project, CSE-506 Operating Systems):**

A stackable file system that automatically backs up files. It manages the backups in a centralized manner, keeps them invisible to users and makes use of extended attributes (*xattrs*) to maintain file version metadata. It also supports a retention policy that limits the maximum number of versions of a file. Users can list versions, read earlier versions, remove them and restore the file to an earlier version via IOCTL. BKPFs is written in C and contains 1,800 lines of kernel code and 500 lines of user code. It also has 600 lines of Shell test scripts.

→ **Kernel Hacking Demo (Team Project of 2 People, CSE-506 Operating Systems):**

Demonstrated a variety of Kernel Hacking options provided by Linux kernel that assist kernel developers to debug and discover potential issues. This project is written in C and has 1,200 lines of kernel code and 500 lines of user code. My contributions included demos of 5 kernel hacking features and implementation of the user driver program.

→ **Security-Enhanced Browser Extension (Team of 4 People, CSE-509 System Security):**

A Chrome extension that enforces security on the user’s side, mainly by preventing password reuse, detecting user credentials on the wrong site and warning users of unpopular sites. The project also features a simple test server written in Python to demonstrate the features. My contribution mainly involved the implementation of the test website which had 700 lines of Python, JS and HTML code.

## **Honors**

★ **Excellence as a TA Award, Computer Science Department of Stony Brook University, 2019**

## **Skill Sets**

→ **Programming Languages:**

- ◆ **Fluent** ( $\geq 3$  years of experience or  $\geq 5,000$  LoC): C, Python
- ◆ **Intermediate** (1~2 years or  $\geq 2,000$  LoC): C++, JavaScript, Bash
- ◆ **Languages that I have used** (Less than 1 year): Java, C#, PHP, LaTeX, HTML, Promela

→ **Software and Frameworks:**

- ◆ **Development Frameworks:** Python Flask (3 years), Linux Kernel (1 year), Spin (6 months)
- ◆ **Distributed Systems:** NFS-Ganesha (1 year), Hadoop/HDFS (6 months), Ceph (6 months)
- ◆ **Development Tools:** Vim (4 years), Git (4 years), VSCode (2 years)
- ◆ **Productivity Tools:** Microsoft Office (4 years), Google Docs (1 year)