

Zhiqiang Zang

Research Interests

Software engineering and programming languages, with a focus on just-in-time compiler testing and optimization.

Education

- 2018–Aug 2023 **Ph.D.**, *The University of Texas at Austin (UT Austin)*, Austin, TX, U.S..
(Expected) Software Engineering and Systems Advisor: Milos Gligoric
- 2014–2018 **B.E.**, *Beijing University of Posts and Telecommunications (BUPT)*, Beijing, China.
Telecommunication Engineering

Publications

- [3] **Zhiqiang Zang**, Nathaniel Wiatrek, Milos Gligoric, and August Shi. Compiler Testing using Template Java Programs. In *International Conference on Automated Software Engineering*, page to appear, 2022. **ACM SIGSOFT Distinguished Paper Award**.
- [2] Pengyu Nie, Marinela Parovic, **Zhiqiang Zang**, Sarfraz Khurshid, Aleksandar Milicevic, and Milos Gligoric. Unifying Execution of Imperative Generators and Declarative Specifications. In *Conference on Object-Oriented Programming, Systems, Languages, and Applications*, pages 217:1–217:26, 2020.
- [1] Ben Buhse, Thomas Wei, **Zhiqiang Zang**, Aleksandar Milicevic, and Milos Gligoric. VeDebug: Regression Debugging Tool for Java. In *International Conference on Software Engineering, Tool Demonstrations Track*, pages 15–18, 2019.

Patents

- [1] Hiroaki Yoshida, **Zhiqiang Zang**, Mukul R. Prasad. Generation of Software Program Repair Examples. US11099817B1. Aug 24, 2021.

Industry Experience

- Summer 2022 **Software Engineer Intern**, *Meta Platforms*, Bellevue, WA, U.S..
Migrated a large-scale data analytic engine onto a new environment to improve performance and maintainability.
- Summer 2020 **Research Intern**, *Fujitsu Laboratories of America*, Sunnyvale, CA, U.S. (Remote).
Improved software program repair examples by minimizing buggy/fixed pairs of programs.
- Summer 2019 **Software Engineer Intern**, *NIO*, Beijing, China.
Developed and improved a physical and visual simulator for autonomous vehicles (Unreal Engine).

Selected Projects

- JAttack <https://github.com/EngineeringSoftware/jattack>
A template-based compiler testing framework that 1) accepts as input a template Java program written in a domain-specific language in Java; 2) executes the template to generate concrete programs and 3) tests Java JIT compilers using the generated programs. JavaParser, OW2 ASM, DSL, JIT, Java. Team Project. Discovered CVEs: CVE-2020-14792, CVE-2022-21305.
- VeDebug <https://github.com/EngineeringSoftware/VeDebug>
A Java debugging tool that 1) automatically sets breakpoints where the current execution diverges from the previously captured one, and 2) provides video player features e.g., speed up/slow down the replay. Bytecode instrumentation(OW2 ASM, javaagent), Java. Team project.

Honors & Awards

- 2022 ACM SIGSOFT Distinguished Paper Award, for [3]
2017 Outstanding Undergraduate Award Finalist, *BUPT*
2015–2016 Qualcomm Innovation Scholarship, *BUPT*
2014–2015 Qualcomm Innovation Scholarship, *BUPT*

Teaching Experience

- Fall 2021 Teaching Assistant, *UT Austin*
EE 379K: Programming Paradigms (38 students)
- Spring 2021 Teaching Assistant, *UT Austin*
EE 360T: Software Testing (101 students)
- Spring 2020 Teaching Assistant, *UT Austin*
EE 312H: Software Design and Implementation I (35 students)
- Fall 2019 Teaching Assistant, *UT Austin*
EE 312: Software Design and Implementation I (60 students)
- Spring 2019 Teaching Assistant, *UT Austin*
EE 422C: Software Design and Implementation II (172 students)

Professional Service

- ICSE 2023 External reviewer, *International Conference on Software Engineering*
ICSE 2022 External reviewer, *International Conference on Software Engineering*
ASE 2021 External reviewer, *International Conference on Automated Software Engineering*
ISSTA-AE 2020 Artifact evaluation committee member, *International Symposium on Software Testing and Analysis*
ISSTA 2020 External reviewer, *International Symposium on Software Testing and Analysis*

Technical Skills

- Programming Languages: Java (everyday), Bash (everyday), C, Python
- Tools: Emacs (everyday), Git (everyday), OW2 ASM, JavaParser