Columbus, Ohio
614-407-8980
genc.5@osu.edu
kaangenc.me
SeriousBug
January 4, 1995

Kaan Genç

Education

Ph.D. **The Ohio State University**, *Computer Science & Engineering*, August 2017 – present.

B.Sc. **İzmir University of Economics**, *Software Engineering*, September 2013 – June 2017.

Work

Teaching **Advanced C Programming**, *Ohio State University*, Autumn 2017, Spring 2018.

The class had around 35 students enrolled for both semesters. I was given the full responsibilities for the class, including teaching the class, writing assignments and exams, grading, and holding office hours. I revised the course material I was given, making it more comprehensive and adding unique insights on building and optimizing advanced C programs.
The students were satisfied with my teaching, with my Student Evaluation of Instruction mean scores being 4.6 and 4.4 out of 5 for both semesters, a score above the university mean for classes of similar size.

Internship **ZetaOps Inc.**, June – September 2016.

ZetaOps focuses on scalable cloud applications. During my internship, I worked on the backend for their open source projects ZEngine, a BPMN workflow framework, and Ulakbüs, a complete information management system for universities. I implemented their internationalization system, revised their permisions system, added course timetabling support, and fixed many bugs.
ZEngine Ulakbüs

Activity

Review **Artifact Evaluation Committee member**, OOPSLA 2020, September 2020.

Research

My research focuses on efficient, scalable, persistent (durable) transactions combining commodity non-volatile memory with hardware transactional memory, and predictive data race detection methods. My goal is to allow scalable, efficient programs taking advantage of new hardware to be developed easily. Below is all of my publications, including links to our open source implementations.

Publications

PLDI 2020 **Crafty: Efficient, HTM-Compatible Persistent Transactions**, *Kaan Genç, Michael D. Bond, and Guoqing Harry Xu*, ACM SIGPLAN Conference on Programming Language Design and Implementation, Online, June 2020.

Extended Paper Paper Talk Implementation

Non-volatile memory combines byte-addressability of DRAM with durability of persistent storage, but it presents many challenges to consistency. Prior works providing consistency incur significant performance costs or require hardware modifications. We propose a new method that provides fully ACID transactions efficiently on existing hardware using our novel logging method, and our efficient adaptive method for providing consistency.

OOPSLA 2019 **Dependence-Aware, Unbounded Sound Predictive Race Detection**, *Kaan Genç, Jake Roemer, Yufan Xu, and Michael D. Bond*, ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications, Athens, Greece, October 2019.

Extended Paper Paper Talk Implementation

Data races can cause bugs that are hard to diagnose. Predictive data race analyses can find races from a multitude of program executions by just analysing a single execution, but prior works miss many races. We introduce two new analyses incorporating data and control dependence, improving data race detection capabilities compared to prior works.

PLDI 2020 **SmartTrack: Efficient Predictive Race Detection**, *Jake Roemer, Kaan Genç, and Michael D. Bond*, ACM SIGPLAN Conference on Programming Language Design and Implementation, Online, June 2020.

Extended Paper Paper Implementation

Predictive data race analyses detect hard-to-find races in programs, but cause large performance impacts compared to widely used happens-before analysis. Our work introduces a variety of optimizations, bridging the performance gap for two analyses from prior work and a new analysis we propose.

PLDI 2018 **High-Coverage, Unbounded Sound Predictive Race Detection**, *Jake Roemer, Kaan Genç, and Michael D. Bond*, ACM SIGPLAN Conference on Programming Language Design and Implementation, Philadelphia, PA, USA, June 2018.

Extended Paper Paper Implementation

Prior predictive data race analyses either miss some races or can not analyze full program executions. Our work presents a new analysis that finds all predictable data races from a single execution, but also finds some false races. We then present our novel algorithm which filters out false races, making the approach sound overall.