

BARIS KASIKCI

Research Assistant

Ecole Polytechnique Fédérale de Lausanne (EPFL)

EPFL - IC - DSLAB
Station 14, Office INN-321
1015 Lausanne, Switzerland

+41 (78) 707 19 13
baris.kasikci@epfl.ch
<http://www.bariskasikci.org>

RESEARCH INTERESTS

My research is centered around building techniques, tools and environments that will ultimately help us build more reliable and secure software. I am interested in finding solutions that will allow programmers to better reason about their code. In this regard, I strive to find efficient ways to detect bugs, classify them, and diagnose their root cause. I especially focus on bugs that manifest in production, because they are truly hard to deal with. I am also interested in hardware and runtime support for enhancing system security, efficient runtime instrumentation, and program analysis under various memory models.

EDUCATION

Ecole Polytechnique Fédérale de Lausanne (EPFL)

Lausanne, Switzerland

Ph.D. in Computer Science

Sep. 2010–Nov. 2015 (expected)

Thesis: Techniques for Advisor: Prof. George Candea

Middle East Technical University (METU)

Ankara, Turkey

M.Sc. in Electrical and Electronics Engineering

Sep. 2006–Jun. 2009

Thesis: Variability Modeling in Software Product Lines

Graduated with the top grade

Advisor: Prof. Semih Bilgen

B.Sc. in Electrical and Electronics Engineering

Sep. 2002–Jun. 2006

Project: Embedded target estimation, detection, and tracking software

Graduated with High Honors

Advisor: Prof. Arzu Koc

AWARDS AND HONORS

Intel Software and Services Group, Grant

2014–2016

VMware, Doctoral Fellowship

2014–2015

EPFL, Doctoral Fellowship

2010–2011

Scientific and Technological Research Council of Turkey, Master Scholarship

2006–2008

Middle East Technical University, Dean's High Honor List

2006

Middle East Technical University,

Undergraduate Final Project Best Team Performance Award

2006

Turkish Customs Association, Scholarship

2002–2006

RESEARCH AND WORK EXPERIENCE

Ecole Polytechnique Fédérale de Lausanne

Lausanne, Switzerland

Research Assistant

Sep. 2010–Nov. 2015 (expected)

Research on software reliability with an emphasis on concurrent software

- **Gist** is the first technique for accurately and automatically diagnosing the root causes of failures without custom hardware or runtime checkpointing support.
- **RaceMob** is the first automated in-production data race detection technique that works always-on and provides high accuracy.
- **Portend** is a high-accuracy technique to classify data races according to their potential consequences under arbitrary memory models.
- **Bias-Free Sampling** is a technique which allows efficiently sampling rarely-executed code—where bugs often lurk—, without over-sampling frequently-executed code.

Intel

Santa Clara, CA, USA

Research Intern

Jul. 2015–Sep. 2015

Failure root cause diagnosis and security enhancements using hardware support

- I designed and implemented a gdb extension that allows developers to determine which program statements operate on a given data type at runtime. Using this extension reduces the number of statements to examine while debugging by an order of magnitude.
- I began investigating hardware support for enhancing system security. In particular, I have been looked into efficient path profiling for auditing and thwarting control flow hijack attacks.

VMware

Palo Alto, CA, USA

Research and Development Intern

Jun. 2014–Sep. 2014

Automated debugging and runtime control flow tracking

- I designed and implemented an infrastructure to remotely debug and profile VMware VCenter virtual machine management software, while it is running in production.
- I implemented a runtime for efficient control flow tracking in software. This work formed the basis of my **Gist** work on root cause diagnosis.

Microsoft Research

Redmond, WA, USA

Research Intern

Jun. 2013–Sep. 2013

Efficient runtime execution sampling technique and low overhead coverage measurement

- I worked on the design of the **Bias-Free Sampling** framework for efficient runtime sampling. I designed and implemented the bias-free sampling framework for managed code (i.e., C#). This tool is internally maintained and used at Microsoft.
- I designed and implemented a fault injection tool to detect resource leakage problems.

Siemens Corporate Technology

Istanbul, Turkey

Senior Software Engineer

Mar. 2008–May 2010

Embedded home and industrial automation software

- I designed and implemented a real-time embedded gateway software between Siemens communication processors and a building automation system.

Aselsan Electronic Industries

Software Engineer

Embedded motor control and functional testing infrastructure

Ankara, Turkey

May 2006–Mar. 2008

- I was responsible for a real time embedded control software for turret motor control. I also designed and implemented a full-system functional testing software.

Student Intern

Embedded software development

Jun. 2005–Jul. 2005

- I developed an embedded control software for a night vision camera on a PIC microcontroller.

PEER-REVIEWED PUBLICATIONS

- [1] Failure Sketching: A Technique for Automated Root Cause Diagnosis of In-Production Failures. Baris Kasikci, Benjamin Schubert, Cristiano Pereira, Gilles Pokam, and George Candea. *Symp. on Operating Systems Principles (SOSP)*, Monterey, CA, October 2015.
- [2] Failure Sketches: A Better Way to Debug. Baris Kasikci, Benjamin Schubert, Cristiano Pereira, Gilles Pokam, Madan Musuvathi, and George Candea. *Workshop on Hot Topics in Operating Systems (HotOS)*, Kartause Ittingen, Switzerland, May 2015.
- [3] Automated Classification of Data Races Under Both Strong and Weak Memory Models. Baris Kasikci, Cristian Zamfir, and George Candea. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, May 2015.
- [4] Efficient Tracing of Cold Code Via Bias-Free Sampling. Baris Kasikci, Thomas Ball, George Candea, John Erickson, and Madanlal Musuvathi. *USENIX Annual Technical Conf. (USENIX ATC)*, Philadelphia, PA, June 2014.
- [5] Lockout: Efficient Testing for Deadlock Bugs. Ali Kheradmand, Baris Kasikci, and George Candea. *5th Workshop on Determinism and Correctness in Parallel Programming (WoDet)*, Salt Lake City, UT, March 2014.
- [6] RaceMob: Crowdsourced Data Race Detection. Baris Kasikci, Cristian Zamfir, and George Candea. *Symp. on Operating Systems Principles (SOSP)*, Farmington, PA, November 2013.
- [7] Automated Debugging for Arbitrarily Long Executions. Cristian Zamfir, Baris Kasikci, Johannes Kinder, Edouard Bugnion, and George Candea. *Workshop on Hot Topics in Operating Systems (HotOS)*, Santa Ana Pueblo, NM, May 2013.
- [8] CORD: A Collaborative Framework for Distributed Data Race Detection. Baris Kasikci, Cristian Zamfir, and George Candea. *Workshop on Hot Topics in Dependable Systems (HotDep)*, Hollywood, CA, October 2012.
- [9] Data Races vs. Data Race Bugs: Telling the Difference with Portend. Baris Kasikci, Cristian Zamfir, and George Candea. *Intl. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, London, UK, March 2012.
- [10] Scalable Modeling of Software Product Line Variability. Baris Kasikci and Semih Bilgen. *Workshop on Scalable Modeling Techniques for Software Product Lines (SCALE)*, San Francisco, CA, August 2009.

TALKS

Automated Root Cause Diagnosis of In-Production Failures

- Symposium on Operating System Principles (SOSP) Oct. 2015
- Intel Corporation Sep. 2015
- Google Sep. 2015
- VMware Sep. 2015

Failure Sketches: A Better Way to Debug

- EcoCloud Annual Event Jun. 2015
- Hot Topics in Operating Systems (HotOS) May 2015

Efficient Tracing of Cold Code via Bias-Free Sampling

- USENIX Annual Technical Conference (USENIX ATC) Jun. 2014

Lockout: Efficient Testing for Deadlock Bugs

- Workshop on Determinism and Correctness in Parallel Programming (WoDet) Mar. 2014

RaceMob: Crowdsourced Data Race Detection.

- Symposium on Operating System Principles (SOSP) Oct 2013
- EPFL Systems seminar Oct. 2013

CoRD: A Collaborative Framework for Distributed Data Race Detection

- Workshop on Hot Topics in System Dependability (HotDep) Oct. 2012

Data Races vs. Data Race Bugs: Telling the Difference with Portend

- International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) Mar. 2012

How to Build Reliable Software?

- Seminar talk to the incoming undergraduate students at EPFL Sep. 2011

PROFESSIONAL SERVICE

Reviewer

- Transactions on Software Engineering 2015
- Transactions on Software Engineering and Methodology 2015

PC Member

- International Symposium on Software Testing and Analysis, Artifact Evaluation Committee 2014

Shadow PC Member

- EuroSys – EuroSys Conference on Computer Systems 2013, 2015

External Reviewer

- CIDR – Conference on Innovative Data Systems Research 2013
- DSN – Intl. Conf. on Dependable Systems and Networks 2011, 2013
- EuroSys – EuroSys Conference on Computer Systems 2011, 2012
- HotOS – Workshop on Hot Topics in Operating Systems 2011, 2013
- USENIX Annual Technical Conference 2011
- SOCC – Symposium on Cloud Computing 2012
- SOSP – Symp. on Operating Systems Principles 2011, 2013
- SPIN – Intl. SPIN Workshop on Model Checking of Software 2011

TEACHING ASSISTANTSHIP

Principles of Computer Systems (graduate level)	2014
Software Engineering (3 rd year undergraduate level)	2011, 2012, 2013
Informatics II (1 st year undergraduate level)	2010

RESEARCH MENTORING

Lisa Zhou (1st year master) Sep. 2015–

- Lisa and I are working on using hardware support for improving the security of software systems. In that regard, Lisa and Benjamin (see below) are building a framework for reproducing security bugs in large applications (e.g., Chrome).

Benjamin Schubert (3rd year undergraduate) Feb. 2015–

- Benjamin and I have been working on a framework that allows reliably reproducing failures in systems software like Apache and MySQL. We used this framework to evaluate my **Gist** work on root cause diagnosis. We are now extending this framework to encompass security vulnerabilities.

Ali Kheradmand (3rd year undergraduate) Jul. 2013–Sep.2013

- Ali and I worked on the **Lockout** project and developed a technique to systematically perturb program executions (without modifying program semantics) to increase the probability of deadlock manifestation. Ali is currently pursuing his Ph.D. in UIUC.

Radu Coman (Master's thesis) Jan. 2012–Sep. 2012

- Radu and I surveyed common concurrency bug patterns in open source software. After we identified data races as a common bug pattern among the 100 bugs we looked at in Google Code, we built a static data race detector. Radu is currently a senior software engineer at Ixia.

LANGUAGES

English: fluent
French: fluent
Turkish: native
German: beginner