dan.luu@gmail.com

EXPERIENCE

Microsoft, Engineer O Found algorithmic simplification, reducing largest and most complicated part of the system to 30LOC This formed the core of a SIGIR 2017 paper that won the Best Paper Award Many other improvements, e.g., reduced the number of hash collisions with improved re-hashing method SmartNIC; multiple order of magnitude tail latency improvement Convaired team to use modern practices such as version control and pass/fail tests Google, Engineer O Convinced team to use modern practices such as version control and pass/fail tests Google, Engineer Thu (deep learning hardware accelerator) https://www.google.com/patents/WO2016186801A1 https://www.google.com/patents/WO2016186801A1 https://www.google.com/patents/WO2016186801A1 https://www.google.com/patents/US20160342880 Recurse Center, Sabbatical Centaur Technology (acquired by VIA), Member of Technical Staff O Frest one sample six-month project (adding an ARM front-end to our x86): Helped reverse engineer the ARMv7 ISA (this was pre-AArch64) Created test generator that found 90% of the first 1000 buys on the project Thurland of the standard and wrote associated microcode Ornated test generator that found 90% of the first 1000 buys on the project Thurland of the scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant Littrafast Optics and Fiber Communications Lab, Research Assistant Littrafast Optics and Fiber Communications Lab, Research Assistant BM, Intern; Austin, TX Semi-formal vonstrained random POWER6 completion unit functional verification Hard Mattabard Constrained random POWER6 completion unit functional verification WHDL Filesh product engineering / characterization. Automated previously manual tasks. Pert Spatial Systems Research Laboratory, Research Assistant EDUCATION BS Math & CMPE (Wisconsin, '90-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS Randomized algorithms can best LRU/pseudo-LRU caches: Randomized algorithms can best LRU/ps	Recurse Center, Sabbatical	2017 - Present
○ Found algorithmic simplification, reducing largest and most complicated part of the system to 30LOC ○ This formed the core of a SIGIR 2017 paper that won the Best Paper Award ○ Many other improvements, e.g., reduced the number of hash collisions with improved re-hashing method ○ SmartNIC; multiple order of magnitude tail latency improvement ○ Half the latency of Amazon "enhanced networking" ○ Convinced team to use modern practices such as version control and pass/fail tests Google, Engineer ○ TPU (deep learning hardware accelerator) ○ https://www.google.com/patents/W02016186801A1 ○ https://www.google.com/patents/W02016186801A1 ○ https://www.google.com/patents/US20160342889 Recurse Centeer, Sabbatical Centaur Technology (acquired by VIA), Member of Technical Staff ○ Her'e's one sample six-month project (adding an ARM front-end to our x86): ○ Helpod reverse engineer the ARM-7 ISA (this was pre-AArch64) ○ Created architectural simulator and got Android running on it ○ Implemented '/'y of the translator, and wrote associated microcode ○ Created test generator that found 90% of the first 1000 bugs on the project ○ Other projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. ○ Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant ○ Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x IBM, Intern; Austin, TX ○ Semi-formal / constrained random POWER6 completion unit functional verification ○ KIBD, Micron Technology, Intern; Boise, ID ○ Flash product engineering / characterization. Automated previously manual tasks. Pert Spatial Systems Research Laboratory, Research Assistant EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS ○ Randomized algorithms can beat LRU/pseudo-LRU caches: Attps://danhuu.com/zehoices-eviction/ ○ A fuzzer written in an hour that found -20 bugs in Julia Attps://da	Microsoft, Engineer	2015-2017
o This formed the core of a SIGIR 2017 paper that won the Best Paper Award o Many other improvements, e.g., reduced the number of hash collisions with improved re-hashing method SmartNIC; multiple order of magnitude tail latency improvement SystemVerilog o Half the latency of Amazon "enhanced networking" o Convinced team to use modern practices such as version control and pass/fail tests Google, Engineer 7 17PU (deep learning hardware accelerator) https://www.google.com/patents/WO2016186801A1 o https://www.google.com/patents/WO2016186801A1 o https://www.google.com/patents/US20160342889 Recurse Center, Sabbatical Spring 2013 Centaur Technology (acquired by VIA), Member of Technical Staff Pleped reverse engineer the ARM-7 ISA (this was pre-At-riched) Created errors engineer the ARM-7 ISA (this was pre-At-riched) Created architectural simulator and got Android running on it Implemented 1/2 of the translator, and wrote associated microcode Verilog / Templating language Created test generator that found 90% of the first 1000 bugs on the project Other projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x MATLAB and C IBM, Intern; Austin, TX Summer 2002 Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant 2001 - 2005 NON-WORK PROJECTS Randomized algorithms can beat LRU/pseudo-LRU caches: https://danhuu.com/zchoices-exiction/ A fuzzer written in an hour that found −20 bugs in Julia https://danhuu.com/zchoices-exiction/ A fuzzer written in an hour that found −20 bugs in Julia https://danhuu.com/zchoices-exiction/ O Combining AFL and QuickCheck for directed fuzzing https://danhuu.com/desvicy-formal-verification Formal verification of	♦ Bing (BitFunnel)	C++
o Many other improvements, e.g., reduced the number of hash collisions with improved re-hashing method o SmartNIC; multiple order of magnitude tail latency improvement SystemVerilog O Half the latency of Amazon "enhanced networking" o Convinced team to use modern practices such as version control and pass/fail tests Google, Engineer O TPU (deep learning hardware accelerator) o https://www.google.com/patents/WO2016186801A1 https://www.google.com/patents/WO2016186801A1 https://www.google.com/patents/US20160342889 Recurse Center, Sabbatical Spring 2013 Centaur Technology (acquired by VIA), Member of Technical Staff 2005 − 2013 Helped reverse engineer the ARMv7 ISA (this was pre-AArch64) o Created architectural simulator and got Android running on it I mplemented ½ of the translator, and wrote associated microcode o Created test generator that found 90% of the first 1000 bugs on the project O ther projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant O Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x MATLAB and C BM, Intern; Austin, TX Summer 2003 O Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant D Summer 2002 Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant D Summer 2003 Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant D Summer 2003 Flash product engineering / characterization. Subminated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant D Summer 2003 Flash product engineering / characterization. Subminated previously manual tasks. Perl Spatial Systems Research Laboratory, Research A	$\circ~$ Found algorithmic simplification, reducing largest and most complicated part of the system to 30LOC	
o SmartNIC; multiple order of magnitude tail latency improvement o Half the latency of Amazon "enhanced networking" o Convinced team to use modern practices such as version control and pass/fail tests Google, Engineer o TPU (deep learning hardware accelerator) o https://www.google.com/patents/WO2016186801A1 o https://www.google.com/patents/US20160342889 Recurse Center, Sabbatical Spring 2013 Centaur Technology (acquired by VIA), Member of Technical Staff O Here's one sample six-month project (adding an ARM front-end to our x86): O Herle's one sample six-month project (adding an ARM front-end to our x86): O Helped reverse engineer the ARMv7 ISA (this was pre-AArch64) O Created architectural simulator and got Android running on it C implemented ½ of the translator, and wrote associated microcode O Created test generator that found 90% of the first 1000 bugs on the project O their projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant Optics and Fiber Communications Lab, Research Assistant Discussional / Constrained random POWER6 completion unit functional verification WIDL Micron Technology, Intern; Boise, ID Flash product engineering / characterization. Automated previously manual tasks. Pert Spatial Systems Research Laboratory, Research Assistant DON-WORK PROJECTS Randomized algorithms can beat LRU/pseudo-LRU caches: A fluzzer written in an hour that found ~20 bugs in Julia Attys://danhuu.com/2choices-eviction/ A fluzzer written in an hour that found ~20 bugs in Julia Attys://danhuu.com/danhuu/Fuzzii Web performance benchmarks for slow/flaky connections Formal verification of a secure hypervisor model Combining AFL and QuickCheck for directed fuzzing Attys://danhuu.com/tenhacucom/tenhacucom/tenhacucom/tenhacucom/tenhacucom/tenhacucom/tenhacucom/tenhacucom/tenhacucom/tenhacucom/tenhacucom/tenhacucom/tenhacucom/tenhacu	$\circ~$ This formed the core of a SIGIR 2017 paper that we	on the Best Paper Award
o Half the latency of Amazon "enhanced networking" o Convinced team to use modern practices such as version control and pass/fail tests Google, Engineer o TPU (deep learning hardware accelerator) o https://www.google.com/patents/WO2016186801A1 o https://www.google.com/patents/WS20160342889 Recurse Center, Sabbatical Centaur Technology (acquired by VIA), Member of Technical Staff 2005 − 2013 o Here's one sample six-month project (adding an ARM front-end to our x86): o Helped reverse engineer the ARMv7 ISA (this was pre-AArch64) o Created architectural simulator and got Android running on it o Implemented ½ of the translator, and wrote associated microcode Verilog / Templating language o Created test generator that found 90% of the first 1000 bugs on the project o Other projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant Distributed Systems and Fiber Communications Lab, Research Assistant Distributed Systems (acquired by 40x MATLAB and Communications Lab, Research Assistant Distributed Systems (acquired by 40x MATLAB and Communications Lab, Research Assistant Distributed Systems (acquired by 40x MATLAB and Communications Lab, Research Assistant Distributed Systems (acquired by 40x MATLAB and Communications Lab, Research Assistant Distributed Systems (acquired by 40x MATLAB and Communications Lab, Research Assistant Distributed Systems (acquired by 40x MATLAB and Communications Lab, Research Assistant Distributed Systems (acquired by 40x MATLAB and Communications Communications Lab, Research Assistant Distributed Systems (acquired by 40x MATLAB and Communications)	$\circ~$ Many other improvements, e.g., reduced the number	of hash collisions with improved re-hashing method
o Convinced team to use modern practices such as version control and pass/fail tests Google, Engineer ○ TPU (deep learning hardware accelerator) ○ https://www.google.com/patents/WO2016186801A1 ○ https://www.google.com/patents/US20160342889 Recurse Center, Sabbatical Centaur Technology (acquired by VIA), Member of Technical Staff ○ Helre's one sample six-month project (adding an ARM front-end to our x86): ○ Helped reverse engineer the ARMv7 ISA (this was pre-AArch64) ○ Created architectural simulator and got Android running on it ○ Implemented 1/2 of the translator, and wrote associated microcode Verilog / Templating language ○ Created test generator that found 90% of the first 1000 bugs on the project ○ Uther projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. ○ Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant ○ Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x IBM, Intern; Austin, TX ○ Semi-formal / constrained random POWER6 completion unit functional verification WHDL Micron Technology, Intern; Boise, ID ○ Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant 2001 EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS ○ Randomized algorithms can beat LRU/pseudo-LRU caches: https://danluu.com/2choices-eviction/ ○ A fuzzer written in an hour that found ~20 bugs in Julia https://danluu.com/achiau/Fuzz.ji ○ Web performance benchmarks for slow/flaky connections ○ A fuzzer written in an hour that found ~20 bugs in Julia https://danluu.com/web-bloat/ ○ Combining AFL and QuickCheck for directed fuzzing ○ A fuzzer written in an hour that found ~20 bugs in Julia https://danluu.com/web-bloat/ ○ Combining AFL and QuickCheck for directed fuzzing ○ Terminal latency benchmarking ○ Filesystem error	$\diamond~$ SmartNIC; multiple order of magnitude tail latency imp	provement $SystemVerilog$
Google, Engineer TPU (deep learning hardware accelerator) https://www.google.com/patents/WO2016186801A1 https://www.google.com/patents/WO201603428899 Recurse Center, Sabbatical Centaur Technology (acquired by VIA), Member of Technical Staff 2005 – 2013 Here's one sample six-month project (adding an ARM front-end to our x86): Helped reverse engineer the ARMv7 ISA (this was pre-AArch64) Created architectural simulator and got Android running on it Created architectural simulator, and wrote associated microcode Verilog / Templating language Created test generator that found 90% of the first 1000 bugs on the project Other projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant 2003 – 2005 Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x MATLAB and C IBM, Intern; Austin, TX Summer 2003 Semi-formal / constrained random POWER6 completion unit functional verification WHDL Micron Technology, Intern; Boise, ID Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS Randomized algorithms can beat LRU/pseudo-LRU caches: https://danluu.com/2choices-eviction/ A fuzzer written in an hour that found ~20 bugs in Julia https://danluu.com/anluu/Fuzz.jl Web performance benchmarks for slow/flaky connections https://github.com/danluu/secvisor-formate-verification Combining AFL and QuickCheck for directed fuzzing https://danluu.com/ueb-bloat/ Formal verification of a secure hypervisor model Combining AFL and QuickCheck for directed fuzzing https://danluu.com/idensuscen-for-fpga https://danluu.com/idensuscen-for-fpga https://danluu.com/filesystem-errors/	• Half the latency of Amazon "enhanced networking"	
o TPU (deep learning hardware accelerator) o https://www.google.com/patents/WO2016186801A1 o https://www.google.com/patents/US20160342889 Recurse Center, Sabbatical Spring 2013 Centaur Technology (acquired by VIA), Member of Technical Staff 2005 − 2013 o Here's one sample six-month project (adding an ARM front-end to our x86): o Helped reverse engineer the ARMv7 ISA (this was pre-AArch64) o Created architectural simulator and got Android running on it C o Implemented ½ of the translator, and wrote associated microcode Verilog / Templating language o Created test generator that found 90% of the first 1000 bugs on the project o Uther projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. o Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant 2003 − 2005 o Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x MATLAB and C IBM, Intern; Austin, TX Summer 2002 o Semi-formal / constrained random POWER6 completion unit functional verification WHDL Micron Technology, Intern; Boise, ID Summer 2002 o Flash product engineering / characterization. Automated previously manual tasks. Pert Spatial Systems Research Laboratory, Research Assistant 2001 EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS o Randomized algorithms can beat LRU/pseudo-LRU caches: https://danluu.com/2choices-eviction/ o A fuzzer written in an hour that found ~20 bugs in Julia https://danluu.com/archoices-eviction/ o Formal verification of a secure hypervisor model https://github.com/danluu/secvisor-formal-verification o Combining AFL and QuickCheck for directed fuzzing https://danluu.com/ueb-bloat/ o Terminal latency benchmarking https://danluu.com/ilespystem-errors/	• Convinced team to use modern practices such as ver	rsion control and pass/fail tests
o https://www.google.com/patents/WO2016186801A1 o https://www.google.com/patents/US20160342889 Recurse Center, Sabbatical Spring 2013 Centaur Technology (acquired by VIA), Member of Technical Staff 2005 − 2013 o Here's one sample six-month project (adding an ARM front-end to our x86): o Helped reverse engineer the ARMv7 ISA (this was pre-AArch64) o Created architectural simulator and got Android running on it C created test generator that found 90% of the first 1000 bugs on the project F# o Other projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. o Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant 2003 − 2005 o Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x MATLAB and C IBM, Intern; Austin, TX Summer 2003 o Semi-formal / constrained random POWER6 completion unit functional verification VIHDL Micron Technology, Intern; Boise, ID Summer 2002 o Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant 2001 EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS o Randomized algorithms can beat LRU/pseudo-LRU caches: https://danluu.com/gchoices-eviction/ o A fuzzer written in an hour that found ~20 bugs in Julia https://danluu.com/gchoices-eviction/ o Formal verification of a secure hypervisor model https://github.com/danluu/secvisor-formal-verification o Combining AFL and QuickCheck for directed fuzzing https://danluu.com/testing/ o Terminal latency benchmarking https://danluu.com/testing/ o Filesystem error handling https://danluu.com/testing/ o Filesystem error handling https://danluu.com/filesystem-errors/	Google, Engineer	2013 - 2014
Recurse Center, Sabbatical Spring 2013 Centaur Technology (acquired by VIA), Member of Technical Staff 2005 – 2013 Here's one sample six-month project (adding an ARM front-end to our x86): Helped reverse engineer the ARMv7 ISA (this was pre-AArch64) Created architectural simulator and got Android running on it Implemented ½ of the translator, and wrote associated microcode Verilog / Templating language Created test generator that found 90% of the first 1000 bugs on the project Helped reverse engineer that found 90% of the first 1000 bugs on the project Helped the projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x MATLAB and C IBM, Intern; Austin, TX Summer 2003 Semi-formal / constrained random POWER6 completion unit functional verification WHDL Micron Technology, Intern; Boise, ID Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS Randomized algorithms can beat LRU/pseudo-LRU caches: Attps://danluu.com/zchoices-eviction/ A fuzzer written in an hour that found ~20 bugs in Julia Attps://danluu.com/cehoices-eviction/ Formal verification of a secure hypervisor model Attps://dihub.com/danluu/secor-formal-verification Combining AFL and QuickCheck for directed fuzzing Terminal latency benchmarking Attps://danluu.com/testing/ Attps://danluu.com/testing/ Attps://danluu.com/testing/	⋄ TPU (deep learning hardware accelerator)	
Centaur Technology (acquired by VIA), Member of Technical Staff Centaur Technology (acquired by VIA), Member of Technical Staff Here's one sample six-month project (adding an ARM front-end to our x86): Here's one sample six-month project (adding an ARM front-end to our x86): Here's one sample six-month project (adding an ARM front-end to our x86): Here's one sample six-month project (adding an ARM front-end to our x86): Here's one sample six-month project (adding an ARM front-end to our x86): Here's one sample six-month project (adding an ARM front-end to our x86): Here's one sample six-month project (adding an ARM front-end to our x86): Here's one sample six-month project (adding an ARM front-end to our x86): Here's one sample six-month project (adding an ARM front-end to our x86): Here's one sample six-month project (adding an ARM front-end to our x86): Here's one sample six-month project (adding an ARM front-end to our x86): Here's one sample six-month project (adding an ARM front-end to our x86): Here's one sample six-month project (adding an ARM front-end to our x86): Created earchitectural simulator and got Android running on it Created earchitectural simulator and secondary front and secondary on the project Here's of the translating language (P## Armotocode Verilog / Templating language (P## Armotocode Verilog / Verilog / Templating language (P## Armotocode Verilog / Verilog / Verilog / Templating language (P## Armotocode Verilog / Verilog	• https://www.google.com/patents/WO2016186801A1	
Centaur Technology (acquired by VIA), Member of Technical Staff O Here's one sample six-month project (adding an ARM front-end to our x86): O Helped reverse engineer the ARMv7 ISA (this was pre-AArch64) O Created architectural simulator and got Android running on it O Implemented '/2 of the translator, and wrote associated microcode O Implemented '/2 of the translator, and wrote associated microcode O Created test generator that found 90% of the first 1000 bugs on the project O Created test generator that found 90% of the first 1000 bugs on the project O The projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. O Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant A Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x O Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x O Semi-formal / constrained random POWER6 completion unit functional verification WHDL Micron Technology, Intern; Boise, ID O Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS Randomized algorithms can beat LRU/pseudo-LRU caches: Attps://danluu.com/2choiccs-eviction/ A fuzzer written in an hour that found ~20 bugs in Julia Attps://danluu.com/anduu/Fuzz.jl Web performance benchmarks for slow/flaky connections Attps://danluu.com/anduu/secoisor-formal-verification Combining AFL and QuickCheck for directed fuzzing Attps://danluu.com/danluu/sega-system-for-fpga Filesystem error handling	$\circ\ https://www.google.com/patents/US20160342889$	
 ♦ Here's one sample six-month project (adding an ARM front-end to our x86): Helped reverse engineer the ARMv7 ISA (this was pre-AArch64) Created architectural simulator and got Android running on it Implemented ¹/2 of the translator, and wrote associated microcode Verilog / Templating language Created test generator that found 90% of the first 1000 bugs on the project F# Other projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc.	Recurse Center, Sabbatical	Spring 2013
o Helped reverse engineer the ARMv7 ISA (this was pre-AArch64) o Created architectural simulator and got Android running on it C Implemented ¹/2 of the translator, and wrote associated microcode o Created test generator that found 90% of the first 1000 bugs on the project F# Other projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. o Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant o Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x MATLAB and C IBM, Intern; Austin, TX Summer 2003 o Semi-formal / constrained random POWER6 completion unit functional verification WHDL Micron Technology, Intern; Boise, ID o Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant 2001 EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS o Randomized algorithms can beat LRU/pseudo-LRU caches: https://danluu.com/2choices-eviction/ A fuzzer written in an hour that found ~20 bugs in Julia https://dianluu.com/achoiuv/Fuzz.jl Web performance benchmarks for slow/flaky connections https://danluu.com/web-bloat/ Formal verification of a secure hypervisor model https://danluu.com/testing/ Terminal latency benchmarking https://danluu.com/testing/ Terminal latency benchmarking https://danluu.com/testing/ Filesystem error handling	Centaur Technology (acquired by VIA), Member o	f Technical Staff 2005 – 2013
 ○ Created architectural simulator and got Android running on it ○ Implemented ¹/2 of the translator, and wrote associated microcode ○ Created test generator that found 90% of the first 1000 bugs on the project ○ Created test generator that found 90% of the first 1000 bugs on the project ○ F# ◇ Other projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. ○ Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant ○ Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x MATLAB and C IBM, Intern; Austin, TX Summer 2003 ◇ Semi-formal / constrained random POWER6 completion unit functional verification VHDL Micron Technology, Intern; Boise, ID Summer 2002 ⋄ Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant 2001 EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS ⋄ Randomized algorithms can beat LRU/pseudo-LRU caches: https://danluu.com/2choices-eviction/ ⋄ A fuzzer written in an hour that found ~20 bugs in Julia https://danluu.com/anluu/Fuzz.jl ⋄ Web performance benchmarks for slow/flaky connections https://danluu.com/web-bloat/ ⋄ Formal verification of a secure hypervisor model https://danluu.com/testing/ ⋄ Terminal latency benchmarking https://danluu.com/testing/ ⋄ Terminal latency benchmarking https://danluu.com/filesystem-errors/ 	♦ Here's one sample six-month project (adding an ARM f	front-end to our x86):
o Implemented ¹/2 of the translator, and wrote associated microcode o Created test generator that found 90% of the first 1000 bugs on the project F# ◇ Other projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. o Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant ◇ Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x MATLAB and C IBM, Intern; Austin, TX Summer 2003 ◇ Semi-formal / constrained random POWER6 completion unit functional verification VHDL Micron Technology, Intern; Boise, ID Summer 2002 ◇ Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant 2001 EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS ◇ Randomized algorithms can beat LRU/pseudo-LRU caches: Aftps://danluu.com/2choices-eviction/ ◇ A fuzzer written in an hour that found ~20 bugs in Julia Attps://github.com/danluu/secvisor-formal-verification ◇ Combining AFL and QuickCheck for directed fuzzing Attps://danluu.com/testing/ ◇ Terminal latency benchmarking Attps://danluu.com/testing/ ◇ Sega system on FPGA Attps://danluu.com/filesystem-errors/	• Helped reverse engineer the ARMv7 ISA (this was pre-AArch64)	
 ○ Created test generator that found 90% of the first 1000 bugs on the project F# ♦ Other projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. ○ Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant 2003 - 2005 ♦ Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x MATLAB and C IBM, Intern; Austin, TX Summer 2003 ♦ Semi-formal / constrained random POWER6 completion unit functional verification VHDL Micron Technology, Intern; Boise, ID Summer 2002 ♦ Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant 2001 EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS ♦ Randomized algorithms can beat LRU/pseudo-LRU caches: https://danluu.com/2choices-eviction/ ♦ A fuzzer written in an hour that found ~20 bugs in Julia https://github.com/danluu/Fuzz.jl ♦ Web performance benchmarks for slow/flaky connections https://danluu.com/web-bloat/ ♦ Formal verification of a secure hypervisor model https://github.com/danluu/secvisor-formal-verification ♦ Combining AFL and QuickCheck for directed fuzzing https://danluu.com/term-latency/ ♦ Sega system on FPGA https://danluu.com/filesystem-error/ ♦ Filesystem error handling https://danluu.com/filesystem-error/ 	$\circ~$ Created architectural simulator and got Android run	nning on it C
 ◇ Other projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc. ◇ Improved job scheduling system, improving machine utilization from 60% to 92% Ultrafast Optics and Fiber Communications Lab, Research Assistant	\circ Implemented $^1\!/_2$ of the translator, and wrote associa	tted microcode Verilog / Templating language
Ultrafast Optics and Fiber Communications Lab, Research Assistant 2003 − 2005 Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x MATLAB and C IBM, Intern; Austin, TX Summer 2003 Semi-formal / constrained random POWER6 completion unit functional verification VHDL Micron Technology, Intern; Boise, ID Summer 2002 Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant 2001 EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS Randomized algorithms can beat LRU/pseudo-LRU caches: https://danluu.com/2choices-eviction/ A fuzzer written in an hour that found ~20 bugs in Julia https://github.com/danluu/Fuzz.jl Web performance benchmarks for slow/flaky connections https://danluu.com/web-bloat/ Formal verification of a secure hypervisor model https://github.com/danluu/secvisor-formal-verification Combining AFL and QuickCheck for directed fuzzing https://danluu.com/term-latency/ Sega system on FPGA https://github.com/danluu/sega-system-for-fpga https://github.com/danluu/sega-system-for-fpga https://github.com/danluu.com/filesystem-errors/	$\circ~$ Created test generator that found 90% of the first 10°	000 bugs on the project $F\#$
Ultrafast Optics and Fiber Communications Lab, Research Assistant 2003 - 2005	♦ Other projects included adding fault tolerance to a distributed system, post-silicon debug, test tooling, etc.	
 Lab work, included speeding up parallel (256 wavelength) polarimeter by 40x MATLAB and C IBM, Intern; Austin, TX Semi-formal / constrained random POWER6 completion unit functional verification WHDL Micron Technology, Intern; Boise, ID Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant 2001 EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS Randomized algorithms can beat LRU/pseudo-LRU caches: https://danluu.com/2choices-eviction/ A fuzzer written in an hour that found ~20 bugs in Julia https://github.com/danluu/Fuzz.jl Web performance benchmarks for slow/flaky connections Formal verification of a secure hypervisor model Formal verification of a secure hypervisor model Combining AFL and QuickCheck for directed fuzzing https://danluu.com/testing/ Terminal latency benchmarking https://danluu.com/danluu/sega-system-for-fpga Filesystem error handling https://danluu.com/filesystem-errors/ 	\circ Improved job scheduling system, improving machine utilization from 60% to 92%	
IBM, Intern; Austin, TX Summer 2003 Semi-formal / constrained random POWER6 completion unit functional verification	Ultrafast Optics and Fiber Communications Lab, Research Assistant 2003 – 2005	
 ◇ Semi-formal / constrained random POWER6 completion unit functional verification		
Micron Technology, Intern; Boise, ID Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant 2001 EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS Randomized algorithms can beat LRU/pseudo-LRU caches: https://danluu.com/2choices-eviction/ A fuzzer written in an hour that found ~20 bugs in Julia https://github.com/danluu/Fuzz.jl Web performance benchmarks for slow/flaky connections https://danluu.com/web-bloat/ Formal verification of a secure hypervisor model https://github.com/danluu/secvisor-formal-verification Combining AFL and QuickCheck for directed fuzzing https://danluu.com/term-latency/ Terminal latency benchmarking https://danluu.com/term-latency/ Sega system on FPGA https://github.com/danluu/sega-system-for-fpga https://danluu.com/filesystem-errors/		
 ♦ Flash product engineering / characterization. Automated previously manual tasks. Perl Spatial Systems Research Laboratory, Research Assistant 2001 EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS ♦ Randomized algorithms can beat LRU/pseudo-LRU caches: https://danluu.com/2choices-eviction/ ♦ A fuzzer written in an hour that found ~20 bugs in Julia https://github.com/danluu/Fuzz.jl ♦ Web performance benchmarks for slow/flaky connections https://danluu.com/web-bloat/ ♦ Formal verification of a secure hypervisor model https://github.com/danluu/secvisor-formal-verification ♦ Combining AFL and QuickCheck for directed fuzzing https://danluu.com/testing/ ♦ Terminal latency benchmarking https://danluu.com/term-latency/ ♦ Sega system on FPGA https://github.com/danluu/sega-system-for-fpga ♦ Filesystem error handling https://danluu.com/filesystem-errors/ 	\diamond Semi-formal / constrained random POWER6 completion unit functional verification	
Spatial Systems Research Laboratory, Research Assistant 2001 EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS A Randomized algorithms can beat LRU/pseudo-LRU caches: A fuzzer written in an hour that found ~20 bugs in Julia Begin by://danluu.com/2choices-eviction/ Web performance benchmarks for slow/flaky connections Formal verification of a secure hypervisor model Combining AFL and QuickCheck for directed fuzzing Terminal latency benchmarking Begin by://danluu.com/term-latency/ Filesystem error handling Research Assistant 2001	Micron Technology, Intern; Boise, ID Summer 2	
EDUCATION BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS \$\leq\$ Randomized algorithms can beat LRU/pseudo-LRU caches: \$https://danluu.com/2choices-eviction/\$ \$\leq\$ A fuzzer written in an hour that found \$\leq\$ 20 bugs in Julia \$https://github.com/danluu/Fuzz.jl\$ \$\leq\$ Web performance benchmarks for slow/flaky connections \$https://danluu.com/web-bloat/\$ \$\leq\$ Formal verification of a secure hypervisor model \$https://github.com/danluu/secvisor-formal-verification\$ \$\leq\$ Combining AFL and QuickCheck for directed fuzzing \$https://danluu.com/testing/\$ \$\leq\$ Terminal latency benchmarking \$https://danluu.com/term-latency/\$ \$\leq\$ Sega system on FPGA \$https://github.com/danluu/sega-system-for-fpga\$ \$\leq\$ Filesystem error handling \$https://danluu.com/filesystem-errors/\$	\diamond Flash product engineering / characterization. Automated previously manual tasks.	
BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS \$\leq\$ Randomized algorithms can beat LRU/pseudo-LRU caches: \$https://danluu.com/2choices-eviction/\$ \$\leq\$ A fuzzer written in an hour that found \$\sigma 20\$ bugs in Julia \$https://github.com/danluu/Fuzz.jl\$ \$\leq\$ Web performance benchmarks for slow/flaky connections \$https://danluu.com/web-bloat/\$ \$\leq\$ Formal verification of a secure hypervisor model \$https://github.com/danluu/secvisor-formal-verification\$ \$\leq\$ Combining AFL and QuickCheck for directed fuzzing \$https://danluu.com/testing/\$ \$\leq\$ Terminal latency benchmarking \$https://danluu.com/term-latency/\$ \$\leq\$ Sega system on FPGA \$https://github.com/danluu/sega-system-for-fpga\$ \$\leq\$ Filesystem error handling \$https://danluu.com/filesystem-errors/\$\$	Spatial Systems Research Laboratory, Research As	sistant 2001
BS Math & CMPE (Wisconsin, '00-'03), MS EE (Purdue, '03-'05) NON-WORK PROJECTS \$\leq\$ Randomized algorithms can beat LRU/pseudo-LRU caches: \$https://danluu.com/2choices-eviction/\$ \$\leq\$ A fuzzer written in an hour that found \$\sigma 20\$ bugs in Julia \$https://github.com/danluu/Fuzz.jl\$ \$\leq\$ Web performance benchmarks for slow/flaky connections \$https://danluu.com/web-bloat/\$ \$\leq\$ Formal verification of a secure hypervisor model \$https://github.com/danluu/secvisor-formal-verification\$ \$\leq\$ Combining AFL and QuickCheck for directed fuzzing \$https://danluu.com/testing/\$ \$\leq\$ Terminal latency benchmarking \$https://danluu.com/term-latency/\$ \$\leq\$ Sega system on FPGA \$https://github.com/danluu/sega-system-for-fpga\$ \$\leq\$ Filesystem error handling \$https://danluu.com/filesystem-errors/\$\$	EDUCATION	
		Purdue, '03-'05)
	NON-WORK PROJECTS	
	♦ Randomized algorithms can beat LRU/pseudo-LRU cae	ches: https://danluu.com/2choices-eviction/
$ \begin{tabular}{lllllllllllllllllllllllllllllllllll$		
	_	
	♦ Combining AFL and QuickCheck for directed fuzzing	
$\diamond \ \ \text{Filesystem error handling} \\ \qquad \qquad \qquad \\ \text{$https://danluu.com/filesystem-errors/} \\$		
	TT	https://danluu.com/android-updates/
\diamond See $https://github.com/danluu/$ and $https://danluu.com$ for more!		

MISCELLANEOUS

 $\diamond\,$ Work Authorization: U.S. Citizen