Summary of Research Accomplishments by Karthik Pattabiraman's group

November 2019

This document summarizes the research we have done in my group for the last 10 years since I joined UBC in 2010. Most of this research was carried out in collaboration with my students and colleagues at UBC, and has been published in top-tier venues such as DSN, ICSE, ISSRE, SC etc. We have also publicly released, and actively maintain, the software artifacts and data-sets for many of the research papers in my group. In total, we have published a total of over 100 papers (conferences, journals, and workshops) since 2010.

I will summarize the six main research directions we have worked on in this time.

1. **Good Enough Dependability:** This has been the mainstay of my research programme at UBC over the last ten years. Together with my students and colleagues, I proposed the idea of "good enough" dependability, which involves trading off protection overhead for fault coverage in applications by systematically modeling error propagation, and reasoning about which errors are the most impactful. Prior to this approach, the default approach was "all-or-nothing" protection, which was used only in a few highly critical systems such as banking and healthcare, due to its high cost. In contrast, the good enough approach can be customized based on the application's needs. Our main contribution is to identify the important portions of an application using both analytical modeling and empirical techniques, without requiring any programmer support.

   The good enough approach has since become the mainstream approach for dependability, and has been adopted by many companies as well. We have published papers on this work at top-tier conferences such as IEEE/IFIP International Symposium on Dependable Systems and Networks (DSN), as well as journals such as the ACM Transactions on Embedded Computing Systems (TECS) [DSN'12][DSN'13][CASES'14][DSN'15][DSN'16][HPDC'17][DSN'18A][DSN'18B][ICS'19][TECS-1][TECS-2]. Our papers in this space have received best paper (and runner up) awards at DSN'18, and at European Dependable Computing Conference (EDCC), 2016 [EDCC'16]. Many academic groups elsewhere (e.g., UIUC, Georgia Tech, UT Austin, Purdue etc.) have done follow-up work based on our results in this area. I have also had collaborations with companies such as IBM Research, Nvidia Research, AMD Research, Los Alamos National Labs (LANL) in this area, and we have coauthored papers with researchers from these companies.

   My recent PhD graduate, Guanpeng Li, whose dissertation was in this area, will start a tenure--track position at the University of Iowa in the US (he also won the NSERC Postdoctoral fellowship), and another (co-supervised) PhD student Bo Fang, will soon start a postdoctoral position at Pacific Northwest National Labs (PNNL) in the US. Yet another PhD graduate of our group (co-supervised) in this area, Layali Rashid, is working at Microsoft in the AI hardware group.

2. **Web and Cloud Reliability Studies:** Along with my graduate students and colleagues at Microsoft and UBC, I pioneered the area of empirical studies of JavaScript-based web applications' reliability. Our papers in this area [ISSRE'11][ESEM'13][MSR'14][TSE-1] showed that even mature, production websites exhibit significant numbers of JavaScript errors, and that the majority of these errors are caused by the interaction of the JavaScript code with the Document Object Model (DOM). The latter result flies in the face of conventional wisdom which rules that JavaScript errors occur due to the loosely typed nature of the language. This work has shaped many of the web

application engineering tools we since developed and released. Since our paper's publication, many other groups have started working in this area, based on our results. The students who worked on these studies are working in industry (e.g., SAP).

We have also worked on characterizing and predicting failures in cloud systems based on empirical data – our group was one of the first to work with the publicly released “Google cluster” dataset of job failures in the cloud [ISSRE’14][ISSREW’14]. We found that many jobs exhibited significant numbers of failures, and that many of the failures were due to resource limitations. We also found that simple techniques such as software rejuvenation can mitigate many of the failures. Finally, we found that it was possible to predict whether a job would fail even half-way during its execution with high accuracy. Our paper on this at ISSRE’14 was recently recognized as one of the “highlights of ISSRE”, which was a collection of the 30 most influential papers in the 30-year history of the IEEE International Symposium on Software Reliability Engineering (ISSRE).

3. **Fault Injectors**: My group at UBC has developed many fault injectors to evaluate the error resilience of programs. One of our main products in this space is LLFI, a fault injector based on the LLVM compiler that is highly configurable and allows easy mapping of the fault propagation results to the program’s code. LLFI supports a wide variety of failure modes in hardware and software, and is used by researchers in both academia and industry (e.g., IBM Research, Pacific Northwest National Lab, Huawei). Further, Cisco Systems funded a project at UBC to consider adopting LLFI in their internal quality assurance process. We have also published a number of papers on LLFI at top-tier venues such as DSN and ISSRE [DSN’14][DSN’17][ISSRE’19]. Another fault injector developed by our group, GPU-Qin, to evaluate the error resilience of GPGPU applications, has been used by academic research groups, Los Alamos National Labs in the US, and industry research labs (e.g., Nvidia, AMD) [ISPASS’14][DATE’14][TPDS]. Finally, our tools have been used by other papers published in this area. Many of my Master's students who worked with me on these fault injection tools are all working in industry positions (e.g., Amazon, Huawei, Microsoft).

4. **Tools for web application engineering:** In collaboration with the SALT Lab at UBC, we have developed a number of tools and techniques for web applications’ engineering based on the insights from our empirical studies. These have been published in top-tier international conferences such as ICSE (IEEE/ACM International conference on software engineering), ASE (IEEE/ACM International Conference in Automated Software Engineering), and ICST (IEEE International Conference on Software Testing, Verification, and Validation). The tools are also being used by other researchers in the area. One such paper, Clematis, won the SIGSOFT distinguished paper award at the IEEE/ACM International Conference on Software Engineering (ICSE'14) [ICSE’14A], and was extended as a journal paper [TOSEM]. Another paper, AutoFlox, was nominated for a best paper award at the IEEE International Conference on Software Testing (ICST'12) [ICST’12]. Yet another paper, Mutandis, won the best paper runner up award at ICST'13 [ICST’13], and an extended version was published at the IEEE Transactions on Software Engineering [TSE-2]. More recently, we have published papers on tools for building reliable server-side web applications written in node.js [ICSE’16][ICSE’18], and for finding bugs in Model-View Controller framework-based web applications [ICSE’15][ASE’17]. We have also used techniques from program repair [ICSE’14B] and program synthesis [ASE’14][ASE’15] to fix bugs in web applications, and to help programmers in developing web applications respectively.

   My recent (co-supervised) PhD graduate Saba Alimadadi, whose dissertation was in this area, will soon start a tenure-track position at Simon Fraser University in Canada (she won the NSERC Postdoctoral fellowship). Two other PhD graduates of mine (also co-supervised) in this area are now working in industry (SAP, and Storefront.com), as are many masters' students (e.g., SAP, Amazon).

5. **Internet of Things (IoT) Reliability and Security:** In the recent past, I have been working on the foundations of building IoT systems that are both reliable and secure. The goal is to provide automated detection of critical errors and security attacks in such systems. Our work in this space has led to best paper awards at conferences such as EDCC (European Conference on Dependable Computing) [EDCC'15], and a Killam Research Fellowship awarded by UBC (awarded to 10 researchers across all of UBC in 2016). We have also released ThingsJS, an open-source middleware platform for programming IoT systems in a reliable and safe manner – this has attracted attention from companies such as Intel and Huawei [M4IoT'17][ECOOP'18][HotEdge'19]. Our papers in this area have appeared at FSE (IEEE/ACM International Conference on Foundations of Software Engineering) [FSE'17], European Conference on Object Oriented Programming (ECOOP) [ECOOP'18], and ACSAC (Annual Computer Security Applied Conference) [ACSAC'16]. We have also published our work on smart meter security in journals such as TECS (ACM Transactions on Embedded Computing) [TECS-3] – this work was recently featured by UBC on their website, and picked up by multiple news outlets as well. We have recently extended this work to robotic vehicle systems (e.g., drones, rovers), and have found security attacks that violate their safety invariants [ACSAC'19].

   My postdoctoral researcher, Julien Samson, who worked in this area, started a tenure track position at Ecole Technologie Superiore (ETS), Montreal (he also received the NSERC post-doctoral fellowship). Further, my recent PhD graduate, Farid Molazem Tabrizi, whose dissertation was in this area, is now working as a software engineer at Google in the US in the software security team. Many of my Masters students have also worked in this direction, and are working in industry now.

6. **Dependability of Machine Learning:** This is a recent area that I have been exploring with my students and colleagues at Nvidia and Los Alamos National Labs. We investigated the resilience of Machine Learning (ML) algorithms deployed on special purpose hardware accelerator platforms to soft errors. We found that contrary to conventional wisdom, ML algorithms are susceptible to soft errors, and these can cause safety violations in applications such as autonomous vehicles. This work was published at SC 2017, and was co-authored with researchers from MIT and Nvidia - this paper has been cited 40 times since its publication in 2017 [SC'17].

   We have also built a fault injection tool for ML programs written using the TensorFlow framework, called TensorFI, which has been used by other researchers [WoSoCER'18]. Our most recent work in this area has been to optimize the fault injection process for ML programs based on their mathematical properties, thereby achieving an order of magnitude speedup – we call this tool BinFI. BinFI was used to uncover safety-critical violations in ML programs deployed on autonomous vehicles at a fraction of the cost of exhaustive injection techniques. This work was done in collaboration with Los Alamos National Labs (LANL), and published at SC 2019 [SC'19].

Summary: I believe that the future will be shaped by cyber-physical systems connected together to form the IoT. Ensuring security and reliability of IoT systems is one of the most important problems facing us today as we become increasingly dependent on their correct operation. I also believe that as ML is adopted in more and more safety-critical applications (e.g., autonomous vehicles), it will become important to ensure the resilience of ML systems to both accidental errors and security attacks. Consequently, I plan to actively pursue these two directions in the next few years.

References (my student/postdoc names are bolded in the list below)

[DSN'12] *BlockWatch: Leveraging Similarity in Parallel Programs for Error Detection,* ***Jiesheng Wei*** and Karthik Pattabiraman, Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2012.

[DSN'13] *Error Detector Placement for Soft Computation,* ***Anna Thomas*** and Karthik Pattabiraman, Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2013.

[CASES'14] *SDCTune: A Model for Predicting the SDC Proneness of an Application for Configurable Protection*, **Qining Lu**, *Karthik Pattabiraman*, Meeta S. Gupta and Jude A. Rivers, International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES), 2014.

[DSN'15] *Fine Grained Characterization of Faults Causing Long Latency Crashes in Programs,* ***Guanpeng Li*****, Qining Lu,** and Karthik Pattabiraman, Proceedings of the IEEE/IFIP International Conference on Dependable Systems (DSN), 2015.

[DSN'16] *ePVF: An Enhanced Program Vulnerability Factor Methodology for Cross-Layer Resilience Analysis,* **Bo Fang, Qining Lu,** Karthik Pattabiraman, Matei Ripeanu and Sudhanva Gurumurthi, Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2016.

[HPDC'17] *LetGo: A Lightweight Continuous Framework for HPC Applications Under Failures,* **Bo Fang,** Qiang Guan, Nathan Debardeleben**,** Karthik Pattabiraman, and Matei Ripeanu, Proceedings of the ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC), 2017.

[DSN'18A] *Modeling Soft Error Propagation in Programs*, **Guanpeng Li,** Karthik Pattabiraman, Siva Hari, Michael Sullivan and Timothy Tsai, Proceedings of the IFIP/IEEE International Conference on Dependable Systems and Networks (DSN), 2018.

[DSN'18B] *Modeling Input-Dependent Error Propagation in Programs*, **Guanpeng Li, and** Karthik Pattabiraman, Proceedings of the IFIP/IEE International Conference on Dependable Systems and Networks (DSN), 2018.

[ICS'19] *BonVoision: Leveraging Spatial Data Smoothness for Recovery from Memory Soft Errors,* **Bo Fang,** Karthik Pattabiraman, Matei Ripeanu and Sriram Krishnamoorthy, Proceedings of the ACM International Conference on Supercomputing (ICS), 2019.

[TECS-1] *Error Detector Placement for Soft-Computing Applications,* **Anna Thomas** and Karthik Pattabiraman, ACM Transactions on Embedded Computing Systems (TECS), 2016.

[TECS-2] *Configurable Detection of SDC-causing Errors in Programs,* **Qining Lu, Guanpeng Li,** Karthik Pattabiraman, Meeta Gupta and Jude Rivers, ACM Transactions on Embedded Computing Systems (TECS), 2017.

[EDCC'16] *Finding Resilience-Friendly Compiler Optimizations Using Meta-Heuristic Search Techniques,* **Nithya Narayanamurthy,** Karthik Pattabiraman and Matei Ripeanu, Proceedings of the European Dependable Computing Conference (EDCC), 2016.

[ISSRE'11] *JavaScript Errors in the Wild: An Empirical Study*, ***Frolin Ocariza,*** Karthik Pattabiraman and Benjamin Zorn, Proceedings of IEEE International Symposium on Software Reliability Engineering (ISSRE), 2011.

[ESEM'13] *An Empirical Study of Client-Side JavaScript Bugs*, ***Frolin Ocariza*****, Kartik Bajaj,** Karthik Pattabiraman and Ali Mesbah, Proceedings of the IEEE Intl. Symposium on Empirical Software Engineering and Measurement (ESEM), 2013.

[MSR'14] *Mining Questions Asked by Web Developers,* **Kartik Bajaj,** Karthik Pattabiraman and Ali Mesbah, Proceedings of the 11th Working Conference on Mining Software Repositories (MSR), 2014.

[TSE-1] *A Study of Causes and Consequences of Client-Side JavaScript Bugs,* **Frolin Ocariza, Kartik Bajaj,** Karthik Pattabiraman and Ali Mesbah, IEEE Transactions on Software Engineering (TSE).

[ISSRE'14] *Failure Analysis of Jobs in Compute Clouds: A Google Cluster Case Study***,** **Xin Chen**, Charng-da Lu and *Karthik Pattabiraman*, Proceedings of the 25th IEEE International Symposium on Software Reliability Engineering (ISSRE), 2014.

[ISSREW'14] *Failure Prediction of Jobs in Compute Clouds: A Google Cluster Case Study,* **Xin Chen,** Charng-da Lu and Karthik Pattabiraman, International Workshop on Reliability and Security Data Analysis (RSDA), 2014.

[DSN'14] *Quantifying the Accuracy of High-Level Fault Injection Techniques for Hardware Faults,* **Jiesheng Wei, Anna Thomas, Guanpeng Li,** and *Karthik Pattabiraman*, Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2014.

[DSN'17] *One Bit is (Not) Enough: An Empirical Study of the Impact of Single and Multiple Bit Flip Errors,* **Behrooz Sangchoolie,** Karthik Pattabiraman and Johan Karlsson, Proceedings of the IEEE International Conference on Dependable Systems and Networks (DSN), 2017.

[ISSRE'19] *A tale of two injectors: An end-to-end comparison of IR-level and Assembly-level Fault Injection*, **Lucas Palazzi, Guanpeng Li, Bo Fang,** and Karthik Pattabiraman, Proceedings of the IEEE International Conference on Software Reliability Engineering (ISSRE), 2019.

[ISPASS'14] *GPU-Qin: A Methodology for Evaluating the Error Resilience of GPGPU Applications, **Bo Fang,*** Karthik Pattabiraman, Matei Ripeanu and Sudhanva Gurumurthi, Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2014,

[DATE'14] *GPUS: Combining high-performance with high-reliability,* L. Bautista Gomez, F. Cappello, L. Carro, N. DeBardeleben, **B. Fang**, S. Gurumurthi, K. Pattabiraman, P. Rech, M. Sonza Reorda, Embedded tutorial paper Proceedings of the International Symposium on Design Automation and Test in Europe (DATE), 2014

[TPDS] *A Systematic Methodology for Evaluating the Error Resilience of GPGPU Applications,* **Bo Fang,** Karthik Pattabiraman, Matei Ripeanu, and Sudhanva Gurumurthi, IEEE Transactions on Parallel and Distributed Systems (TPDS). 2015.

[ICSE'14A] *Inferring Hierarchical Motifs from Execution Traces,* **Saba Alimadadi,** Ali Mesbah and Karthik Pattabiraman, Proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE), 2018.

[TOSEM] *Understanding JavaScript Event-Based Interactions with Clematis,* **Saba Alimadadi,** Ali Mesbah and Karthik Pattabiraman, ACM Transactions on Software Engineering and Methodology (TOSEM), 2016.

[ICST'12] *AutoFlox: An Automatic Fault Localizer For JavaScript, **Frolin Ocariza***, Karthik Pattabiraman and Ali Mesbah, IEEE International Conference on Software Testing, Verification and Validation (ICST), 2012.

[ICST'13] *Efficient JavaScript Mutation Testing*, ***Shabnam Mirshokraie,*** Ali Mesbah and Karthik Pattabiraman, Proceedings of the IEEE International Conference on Software Testing, Verification and Validation (ICST), 2013.

[TSE-2] *Guided Mutation Testing for JavaScript Web Applications*, **Shabnam Mirshokraie,** Ali Mesbah and Karthik Pattabiraman, IEEE Transactions on Software Engineering (TSE), 41(5), 429-444 (2015).

[ICSE'16] *Understanding Asynchronous Interactions in Full-Stack JavaScript,* **Saba Alimadadi,** Ali Mesbah and Karthik Pattabiraman, Proceedings of the IEEE/ACM International Conference on Software Engineering (ICSE), 2016.

[ICSE'18] *Inferring Hierarchical Motifs from Execution Traces,* **Saba Alimadadi,** Ali Mesbah and Karthik Pattabiraman, Proceedings of the ACM/IEEE International Conference on Software Engineering (ICSE), 2018.

[ICSE'15] *Finding Inconsistencies in JavaScript MVC Applications*, ***Frolin Ocariza*****,** Karthik Pattabiraman and Ali Mesbah, Proceedings of the IEEE/ACM International Conference on Software Engineering (ICSE), 2015.

[ASE'17] *Detecting Unknown Inconsistencies in Web Applications,* **Frolin Ocariza**, Karthik Pattabiraman, and Ali Mesbah, Proceedings of the ACM/IEEE International Conference on Automated Software Engineering (ASE), 2017.

[ASE'14] *DOMpletion: DOM-Aware JavaScript Code Completion*, ***Kartik Bajaj,*** Karthik Pattabiraman and Ali Mesbah, Proceedings of the ACM International Conference on Automated Software Engineering (ASE), 2014.

[ASE'15] *Synthesizing Web Element Locators, **Kartik Bajaj,*** Karthik Pattabiraman and Ali Mesbah, Proceedings of the IEEE/ACM International Conference on Automated Software Engineering (ASE), 2015.

[EDCC'15] *Flexible Intrusion Detection Systems for Memory-Constrained Embedded Systems, **Farid Tabrizi*** and Karthik Pattabiraman, Proceedings of the 11th European Conference on Dependable Computing (EDCC), 2015.

[M4IoT'17] *ThingsJS: Towards a Flexible and Self-Adaptable Middleware for Dynamic and Heterogeneous IoT Environments,* **Julien-Gascon Samson, Mohammad Rafiuzzaman**, and Karthik Pattabiraman, Workshop on Middleware and Applications for the Internet of Things (M4IoT), 2017.

[ECOOP'18] *ThingsMigrate: Platform-Independent Migration of Stateful JavaScript IoT Applications,* Julien Gascon-Samson, Kumseok Jung, Shivanshu Goyal, Armin Rezalean-Asel, Karthik Pattabiraman, Proceedings of the European Conference on Object Oriented Programming (ECOOP), 2018.

[HotEdge'19] *OneOS: IoT Platform based on POSIX and Actors,* **Kumseok Jung, Julien Gascon-Samson,** and Karthik Pattabiraman, 2nd Usenix workshop on Hot Topics in Edge Computing (HotEdge), 2019.

[ACSAC'16] *Formal Security Analysis of Smart Embedded Systems*, **Farid Molazem Tabrizi** and Karthik Pattabiraman, Proceedings of the Annual Computer Security Applications Conference (ACSAC), 2016.

[TECS-3] **Farid Molazem Tabrizi** and Karthik Pattabiraman, *Design-Level and Code-Level Security Analysis of IoT Devices*, ACM Transactions on Embedded Computing Systems (TECS). 2019.

[SC'17] *Understanding Error Propagation in Deep Learning Neural Network (DNN) Accelerators and Applications,* **Guanpeng Li,** Siva Hari, Michael Sullivan, Timothy Tsai, Karthik Pattabiraman, Steve Keckler, and Joel Emer, Proceedings of the International Conference for High-Performance Computing, Storage and Networking (SC), 2017.

[WoSoCER'18] *TensorFI: A Configurable Fault Injector for TensorFlow Applications,* Guanpeng Li, Karthik Pattabiraman, and Nathan DeBardeleben, 8th IEEE International Workshop on Software Certification (WoSoCER), 2018.

[SC'19] *BinFI: An Efficient Fault Injector for Safety-Critical Machine Learning Systems*, **Zitao Chen, Guanpeng Li,** Karthik Pattabiraman, and Nathan DeBardeleben, To appear in the IEEE/ACM International Conference for High-Performance Computing, Storage and Networking (SC), 2019.