Research Statement

I am a third year Ph.D student in the Programming Languages and Software Engineering (PLSE) Group at University of California, Los Angeles. My research interests broadly span Software Engineering, Distributed Systems and Data Science. Specifically, I am interested in supporting interactive debugging in big data processing frameworks and providing efficient ways to perform automated fault localization in big data applications.

Education

Pursuing a Ph.D in Computer Science
   University of California, Los Angeles (3.70/4.0)
   Advisor: Miryung Kim

Bachelor of Science in Computer Science
   Lahore University of Management Sciences (3.73/4.0)

Publications


M A Gulzar, X Han, M Interlandi, S Mardani, S D Tetali, T Condie, T Millstein, M Kim. “Interactive Debugging for Big Data Analytics”. In The 8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 2016), Denver, CO, June 2016. USENIX Association


Invited Talks

Interactive Debugger for Big Data Analytics FSE, Nov 2016
BigDebug: Debugging Primitives for Interactive Big Data Processing in Spark ICSE, May 2016
Interactive Debugging for Big Data Analytics NEC Labs America, August 2016
Towards Big Data Debugging in Apache Spark Databricks Inc., August 2015

Research and Work Experience

Graduate Student Researcher
   University of California, Los Angeles Sep ‘14 – Present

Summer Research Assistant
   NEC Labs America, Princeton NJ Jun ‘16 – Sep ‘16

Teaching Assistant
   CS130: Software Engineering at UCLA Sept ‘15 – Dec ‘15

Research Assistant
   Lahore University of Management Sciences Aug ‘13 – Aug ‘14
Software Engineer
Train of Thought (PVT) Ltd.
Sep '12 – Sep '13

Research Intern
Koç University, Turkey
May ’12 – Aug ’12

SELECTED PROJECTS

**BigDebug: Debugging Primitives for Interactive Big Data Processing in Spark**
Sep ’14 – Sep ’15

**Position:** Graduate Student Researcher at University of California, Los Angeles

Apache Spark has become a key platform for Big Data Analytics, yet it lacks complete support for debugging analytics programs. As a result, debugging Spark programs can be a painstakingly long process. To address this challenge, we designed a set of interactive, real-time debugging primitives for big data processing in Apache Spark. This requires rethinking the notion of step-through debugging in a traditional debugger such as gdb, because pausing the entire computation across distributed worker nodes causes significant delay and naively inspecting millions of records using a watchpoint is too time consuming for an end user. Data scientists can leverage BigDebug interactive debugging capabilities to set breakpoints and watchpoints, localize and repair faults, trace forward/backward through a program execution and perform function hot-swapping at runtime. In our empirical evaluation, BigDebug provides time saving and improves fault localization accuracy.

**BigSift: Tool-Assisted Automated Fault Localization Service**
Sep ’15 – Present

**Position:** Graduate Student Researcher at University of California, Los Angeles

Errors are hard to diagnose for big data analytics. An error could occur due to a bug in program logic, or it could be due to a wrong assumption or anomalies in input data. For precise and automated fault localization of failure inducing inputs in data workflows, we have built BigSift. BigSift’s underlying algorithm combines data provenance and delta debugging to effectively and efficiently pinpoint (with high precision) and repair the root cause of errors in large-scale distributed data processing. Equipped with an optimization engine, this tool intelligently leverages in-memory data processing, partial incremental computation and resource aware job scheduling to reduce fault localization time by several orders of magnitude.

**Dynamic Model Update in Large Scale Stream Processing**
Summer ’16

**Position:** Summer Research Assistant at NEC Labs America

Big data analytic programs often use meta data to make logical decision while processing incoming data. This meta once packaged and shipped, is hard to update at runtime, especially in stream processing environments. I extended a large scale stream processing engine to support dynamic model broadcasting on the fly. This feature allows machine learning applications, written in Spark, to dynamically modify a model as more information is learned from incoming data. I also designed a heartbeat memory manager for stateful stream processing to avoid over consumption of memory due to unhandled expired states.

**OCCAM: Object Culling and Concretization for Assurance Maximization**
Summer ’13

**Position:** Summer Research Intern at Lahore University of Management Sciences

Feature intensive applications with large code bases can provide functionality to a wide range of users each with their own specific requirements. I contributed to a tool chain, built on LLVM that specializes programs to a certain specifications and configurations in order to gain performance and security benefits including improved cache performance, optimized storage space, and a reduced attack surface.

**BGP is high on SDN**
Fall ’13

**Position:** Research Assistant at Lahore University of Management Sciences

BGP convergence is notorious for its unpredictable and unbounded limits. We present a novel way where we dissipate BGP state change messages from multiple Autonomous Systems. This results in multiplication of BGP propagation effort. We used OpenFlow capable routers distributed across the Internet to optimally automate this process. Our approach also helps to mitigate IP prefix hijack and Multiple Origin Autonomous System (MOAS) conflicts. Experimental results showed that this scheme can significantly increase BGP propagation and alleviate BGP security issues.

**Awards and Honors**

Graduation with Distinction, Dean’s Honor List Award, National Mathematics Olympiad Finalist