Research Statement

I am a fourth year Ph.D candidate in the Programming Languages and Software Engineering (PLSE) Group at University of California, Los Angeles. My research interests lie at the intersection of software engineering and big data systems. 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. I am also a Google Ph.D Fellow for years 2017-19.

Education

Pursuing a Ph.D in Computer Science
University of California, Los Angeles
Advisor: Miryung Kim

Bachelor of Science in Computer Science
Lahore University of Management Sciences

Publications


M Interlandi, A Ekmejki, K Shah, M A Gulzar, S D Tetali, M Kim, T Millstein, T Condie. “Adding data provenance support to Apache Spark”. In The VLDB Journal (Special Issue on ”The Best Papers of” VLDB 2016).


Invited Talks

Automated Debugging in Data-intensive Scalable Computing
SoCC, Sept 2017

Debugging Big Data Analytics in Spark
Spark Summit, Jun 2017

Debugging Big Data Analytics with BigDebug
SIGMOD, May 2017

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
Databricks Inc., Aug 2016

Towards Big Data Debugging in Apache Spark

Awards and Honors

Google Ph.D Fellowship Award 2017-18 (2 years with 1 year extension)
SIGMOD Student Travel Award 2017
SoCC Student Travel Award 2017
Graduation with Distinction, Dean’s Honor List Award, National Mathematics Olympiad Finalist
**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**

**White-Box Data Sampling and Generation for Testing Big Data Workflows**  
Position: Graduate Student Researcher at University of California, Los Angeles  
Jun ’17 – Present

Big data developers lack high confidence about testing their application before submitting it to the cloud to ingest production data. I am working on a novel white-box testing and sampling approach which automatically analyzes the semantics of individual data flow operators in tandem with the symbolic execution result of each user defined function. Based on this, we select a small footprint of data from large scale data which enables developers to thoroughly test the application with sample data in a local workstation, improving testing efficiency without sacrificing fault detection capability.

**Automated Debugging in Data-Intensive Scalable Computing**  
Position: Graduate Student Researcher at University of California, Los Angeles  
Jan ’16 – Sep’17

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 Bicsift. Bicsift’s underlying algorithm combines data provenance and delta debugging to effectively and efficiently pinpoint the root cause of errors in large-scale distributed data processing. The optimization techniques of Bicsift intelligently leverages in-memory data processing, predicate pushdown and resource aware job scheduling to reduce fault localization time by several orders of magnitude.

**BigDebug: Debugging Primitives for Interactive Big Data Processing in Spark**  
Position: Graduate Student Researcher at University of California, Los Angeles  
Sep ’14 – Sep ’15

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 inspecting the entire computation across distributed worker nodes is too time consuming for an end user. BigDebug facilitates users 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.

**Dynamic Model Update in Large Scale Stream Processing**  
Position: Summer Research Assistant at NEC Labs America  
Summer ’16

Big data analytic programs often use meta-data to make logical decision while processing incoming data. This meta-data 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 in a cluster environment. 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**  
Position: Summer Research Intern at Lahore University of Management Sciences  
Summer ’13

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.