Sahiti Bommareddy

+1 404 786-1767
sahiti@jhu.edu
cs.jhu.edu/~sahiti

Education

2019– now **Ph.D., Computer Science**, *Johns Hopkins University*, Baltimore, USA
Advisor: Professor Yair Amir (https://www.cs.jhu.edu/~yairamir/)

2017–2019 **M.S., Computer Science**, *Duke University*, Durham, NC

2007–2011 **B.Tech, Electronics and Communication Engineering**, *Jawaharlal Nehru Technological University*, Hyderabad, India

Professional Experience

Jan 2013–Jan 2016 **Software Engineer**, *Aktrix Technologies Pvt Ltd*, Hyderabad,India

May 2011–Nov 2012 **Performance Engineering Analyst**, *Deloitte*, Hyderabad,India

Specialized in application performance evaluation, network performance analysis and optimization.

Reduced application transaction latency to bring it within the Service-Level-Agreement(SLA) window by identifying performance bottlenecks. Received *Applause Award* in recognition of my application performance optimization efforts at Deloitte for both optimization and performing Root Cause Analysis (RCA) with traffic profiling, CPU, and memory utilization analysis in production environment.
Ensured guaranteed performance (backed by SLAs) in geo-distributed systems. Initiated and led prototype development of on an automated performance monitoring system to enable machine learning-based analysis tool, that reduced cost and time spent on RCA of performance degradation and bottleneck identification by 2x to 5x in each of the issue instances.

Technical Skills

Languages Python, C, Java, SQL

Tools & Frameworks Kubernetes, LoadRunner, HP Performance Center, Mininet, Spark, MongoDB

Systems *NIX (CentOS, Ubuntu, Kali), Windows

Dev Tools Vim, Git, VS Code, Eclipse, PyCharm

Teaching Experience

Spring 2022 Advanced Distributed Systems (JHU EN.601.717), *Special Help*

Fall 2021 Distributed Systems (JHU EN.601.417/617), *Special Help*

Spring 2021 Software for Resilient Communities (JHU EN.601.310), *Special Help*

Fall 2020 Intermediate Programming (JHU EN.601.220), *Special Help*

Fall 2019 Distributed Systems (JHU EN.601.417/617), *Teaching Assistant*

Spring 2019 Introduction to Artificial Intelligence (Duke CS270), *Teaching Assistant*

Research Areas, Projects and Publications

My research interests include distributed systems and networks, resilient critical infrastructure and dependable systems (Lab Website: https://cnds.jhu.edu). My publications are:

Real-Time Byzantine Resilience for Power Grid Substations, Accepted to be published at 41st Interna-

tional Symposium on Reliable Distributed Systems 2022

Data-Centric Analysis of Compound Threats to Critical Infrastructure Control Systems, in 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Baltimore, MD, June 2022, pp. 72-79. URL:https://ieeexplore.ieee.org/abstract/document/9833853

Open Source Software Releases / Products

Spire: Spire is an open-source intrusion-tolerant Supervisory Control And Data Acquisition (SCADA) system for the power grid. It is designed to withstand attacks and compromises at both the system level and the network level, while meeting the timeliness requirements of power grid monitoring and control systems (on the order of 100-200ms update latency).
In the latest release of Version 1.3 in December 2021, I optimized latency of the system in single site configuration to around 30ms i.e., 3x latency performance improvement and added a Machine Learning-based network intrusion detection system. Currently, I am working on applying the intrusion-tolerance principles to achieve Byzantine resilience in power grid substations that have exact real-time constraints on latency (4 milliseconds) among other rigid design constraints. This system is delivered for deployment at Pacific Northwest National Labs (PNNL), Siemens, General Electric and Hitachi Energy.

Web page: https://www.dsn.jhu.edu/spire/

Spines: Many applications including critical infrastructure require high demanding combinations of latency, reliability, resilience, and processing even in presence of malicious actors at network level. To support such applications, new overlay dissemination protocols are developed and used to provide the necessary timeliness and reliability. Spines is a generic messaging infrastructure, that provides automatic reconfiguration and network flexibility required for research and production deployments.
In December 2020, the latest version 5.5 of the Spines release was led by me. It includes the cryptography and needed algorithm support for Byzantine resilient critical infrastructure SCADA systems.

Web page: http://www.spines.org

Select Academic Projects

Tweet Irony Detection: Implemented Irony Detection in tweets by modeling features from Context, Behavioural and Word embeddings. Explored a number of classifiers and improved F1-score by nearly 10% compared to Baseline.

Loosely Coupled Key-Value Store with Fault Tolerance: Built a sharded, replicated key-value store that utilizes the Raft consensus protocol. The objective of this project was to study the tradeoffs offered by the CAP theorem and implications of data distribution schemes. The prototype built was Consistent, but not Available, under a network partition.

Fair Decision Making using Privacy-Protected Data: This research investigates if fairness and privacy are fundamentally incompatible, and if there exists some differentially private mechanism which can preserve fairness, or if fairness can only be preserved in some cases. This project extends previous work by exploring new privacy mechanisms, and by designing a new fairness metric. Different privacy preserving mechanisms are explored to reduce error and increase fairness while preserving privacy.

Cracking a Home Router: Evaluated the security vulnerabilities of a home router – compromised the router using several known methods, such as password cracking, DoS attacks, Evil twin exploit, SYN flood and MITM attacks. This helped in understanding the security vulnerabilities to which a networked system can be exposed. The successful attacks were demonstrated and identified.

Performance Evaluation of Fat-Tree: The fat-tree topology is one of the most commonly used network topologies. This project builds a fat-tree on the Mininet network emulator, and compares its performance with a standard tree topology. The impact of application distribution, scheduling schemes and topology on performance SLAs and Bandwidth Utilization based on traffic profile varied from 0.5x - 3x.