SANDEEP POLISETTY

Email: spolisetty@cs.umass.edu

Seeking a research internship position for summer 2020

Areas of Interests: Graph Mining, Knowledge Graphs, Graph Learning, Graph Mining using GPU, Systems for Machine Learning, Stream Processing, Data Mining and Databases

University	Degree
University of Massachusetts, Amherst	MS / PhD Computer Science, 2016 - Present
University of Massachusetts, Amherst	MS Computer Science, 2016 - 2018 (3.92/4)
Indian Institute of Technology, Kharagpur	Bachelors of Technology, 2008-2013

UNIVERSITY RESEARCH EXPERIENCE

Graph Mining (advised by Prof. Marco Searfini) (2018 - present)

- Identified bottlenecks in subgraph enumeration. The standard approach to subgraph enumeration is to expand a partially matched subgraph one vertex at a time by intersecting the adjacency lists of all incident vertices. However, list intersection suffers from high cost of branch misprediction.
- Observed that some intersections are repeated and that using same sized bit-vectors instead of variable length adjacency lists would minimize branch mispredictions. Also noted that using bit vectors can extract more throughput from vectorized operators such as SIMD.
- With these ideas, developed a prototype in **C++** which **outperforms the state of the art platforms,** both **SIMD** - (LIGHT) and **non-SIMD** (DUALSIM), by **a factor of 2 - 4**. This work is in submission to SIGMOD (top conference).

Distributed Systems (advised by Prof. Arun Venkataramani) (2017-2018)

- Incorporated transactions into a **massively replicated key-value** store that provides consistency through the **PAXOS** protocol.
- A key challenge is guaranteeing safe ACID transactions while being highly available. This challenge was overcome by replicating the transaction coordinator and coordinating the steps of the replicated coordinator with the PAXOS protocol. This nested transaction approach (combining **2PC+Paxos**) introduced by scatter allowed highly available ACID transactions with safety guarantees.
- The transaction protocol was formally written in **TLA+** and its safety properties proved in **TLAPS**. This project was written in **JAVA**.

Teaching Assistant
For Prof. Mark Corner **OPERATING SYSTEMS** - Spring 2019

- Introduced the MIT **xv6** toy kernel as a pedagogical tool for the first time at UMass Amherst OS course. The xv6 kernel allows students to immerse themselves with OS concepts and implementations close to real world scenarios.
- Using real kernel code from xv6, lead a class of 70 students through code walkthroughs and **GDB** steps, demonstrating key OS concepts.
- Created weekly kernel hacking tutorials and assignments in **C.**

INDUSTRIAL INTERNSHIP EXPERIENCE

S3 - LOG **AMAZON WEB SERVICES**
Summer 2018

- Inspected the internals of Apache Zookeeper, which is a widely used software that provides powerful distributed service functionalities to clients, such as PUSH notifications. **Studied over 10000 lines of zookeeper source code** to strategize sidestepping the need of deploying zookeeper consistency protocol on the server side. The **OBJECTIVE** instead was to use Amazon internal consistency services while providing for zookeeper clients.
- Created a Zookeeper-like server that uses Amazon internal consistency services, and passed the entire test suite of zookeeper with my modified tool. This plays a key role in consolidating services while providing for Apache Zookeeper clients.
- Documented a formal proof guaranteeing the properties of consistency.

After successful completion of this project, I was **awarded a return internship** offer which I could not take up to focus on research.

ALEXA INTERN **AMAZON DEVICES INTERN**
Summer 2017

- Identified service calls to test key components in Alexa-enabled Fire Phones by studying the android source code.
- Integrated these identified service calls into current testing infrastructure.
- This enhancement to current infrastructure allowed automating a much wider range of test scenarios. On successful completion of this project I was **awarded a return offer.**

SKILLS

Programming Languages: C/C++, Java, Python, Ocaml
Formal Languages for Distributed System Verification: TLA, TLC, TLAPS

Distributed Programming Models: Hadoop, Spark, Pregel
Databases: Postgres-sql, MySql, Neo4j