Harsha

https://harsha.ca

Email : devkhv129@gmail.com
Mobile : +1-604-679-7845

Education

- **University of British Columbia** Vancouver, BC, Canada
  Master of Applied Science in Computer Engineering; Percentage: 88.5% *Sept. 2017 – Present*
- **Manipal Institute of Technology** Manipal, India
  Bachelor of Technology in Computer Science; GPA: 8.86/10.0 *June. 2011 – July. 2015*

Experience

- **University of British Columbia** Vancouver, BC, Canada
  Research and Teaching Assistant *Sept 2017 - Present*
  - **Research Assistant - Distributed Cloud Native Systems**: Participated in research on cluster schedulers for **Microservices** applications under Dr.Julia Rubin. Involved in creation of models for inter-dependencies between services and strategies to improve performance and resource utilization in shared cluster deployments. Currently developing extensions to **Kubernetes** in **GOLANG** to integrate with DevOps pipelines in public clouds.
  - **Lead student - IBM CAS Fellowship Project**: Lead a team of three to work with IBM cloud for identifying reliability issues in their Microservices deployments.
  - **Teaching Assistant - Computer Networks and Engineering Design Studio**: Each of the courses has more than 80 third year undergrads enrolled. Involved in creating assignment, exams and conducting lab sessions.
- **Siemens** Bangalore, India
  Systems Software Engineer *July 2015 - July 2017*
  - **System Control Unit for Siemens Artis Zeego**: Part of a 15 member globally distributed (Germany, America and India)team working on real-time control software for x-ray systems. Control system contains more than 600k lines of **C++** code and had strict coding requirements to ensure safety standards of the product. I solved more than 60 production issues and implemented 13 new features in the product. Spot award was given as an appreciation for quick adaption into product domain.
  - **X-ray Generator Simulation**: Started as a personal side project and meant to provide a software based simulation to replace costly hardware solution used for x-ray testing. The simulator is built in **C++** and uses CANopen network stack, control state machine to simulate actions of an x-ray generator. It is now integrated into automated deployment process and used for daily testing replacing costly hardware simulator.
  - **CANOPEN network stack**: Control area network is primarily used for real time communication between safety critical components in a control system. CANOPEN is standard protocol specification for the CAN. I implemented CANopen network stack in **C++** for communication between x-ray switch and generator.
- **SanDisk** Bangalore, India
  Graduate Intern *Jan 2015 - June 2015*
  - **Firmware Validation Framework for Flash Translation Layer**: Implemented a test framework in c++ to simulate solid state drives for automated testing of SSD device firmware.

Awards

- **Innovation Ambassador - Siemens - 2016**: Awarded for proposing and implementing a prototype which has a business impact on the company (cost reduction).
- **Instant Purskar Award - Siemens - 2015**: Awarded as a recognition for quick adaption into working domain and contributions to the main project.

Publications

- **Supporting Microservice Evolution**: published in 33rd IEEE International Conference on Software Maintenance and Evolution. See **http://ieeexplore.ieee.org/abstract/document/8094458.**

Recent Projects

- **Energy Management System for Android Kernel**: This project is aimed at creating a system to enable assigning and enforcement of strict energy quotas for android apps. These fixed energy quotas are assigned by users to save energy used by fewer priority apps. This project is implemented as a **Android kernel module and completely in C**.
- **REST Framework Over Named Data Networking**: Named data networking(NDN) is a future Internet architecture which is aimed at routing based on content names rather than IP addresses. NDN can be an ideal network between microservices as it has in-built support for features such as service discovery, load balancing ...etc. In this project, we implemented a framework to enable existing microservice applications to be deployed on NDN with minimal effort. To achieve this we implemented REST support for NDN network and a new flow balancing algorithm to address needs of REST based applications. It is implemented in **C++ and Python**. See the code at **https://github.com/asystemsguy/Microservices-over-NDN.**
- **MPI Cluster Network Simulator Using Linux Traffic Control**: This project is to simulate network properties such as delay, jitter, packet loss in MPI applications deployed over a virtual cluster. It is achieved by using Linux Kernel Traffic module and implemented in **Python**, See the code at **https://github.com/asystemsguy/MPI-network-simulator.**

Skills

- **Programming Languages**: C/C++, Go, Python
- **Technologies**: Google cloud, Kubernetes, Docker