Rajan Sawhney

URL: https://rajan3012.github.io

CONTACT INFORMATION:
Phone: (541)-954-4110
Email: rsawhney@cs.uoregon.edu
GitHub: https://github.com/rajan3012
LinkedIn: https://www.linkedin.com/in/sawhneyrajan

SUMMARY & INTERESTS:

3+ years of Industry and Graduate Research experience

Distributed Systems, Cloud Engineering, Data Science, Statistical Modelling, Machine Learning, Computer Vision, Web Development, Mobile Development, User Interfaces and API development

EDUCATION:

M.S in Computer Science, University of Oregon (June 2017)

Relevant Coursework: Algorithms and Complexity, Advance Data Structures, Big Data and Data Science, AI, Probabilistic Methods in AI, Parallel Computing, Distributed Systems, Software Engineering and User Interfaces

B.E Information Technology, University of Pune, India (May 2013)

TECHNICAL SKILLS:

Programming and Scripting Languages: Python, C, C++, C#, Java, JavaScript, SQL, MATLAB, R, OCaml, Bash

Libraries, Frameworks and Development Platforms: ReactJS, NodeJS, Express, REST API, AWS, Weka, Unity, Docker Sci-kit learn, SciPy, NumPy, Pandas, MPI, OpenMp, TBB, Cilk, Android Studio, Hadoop, Arduino

PROFESSIONAL EXPERIENCE:

Graduate Research Fellow, HPCL, University of Oregon July 2016 – June 2017

- Worked as a research member, at the High-Performance Computing Lab (HPCL), on a project funded by the Department of Energy(DoE) to auto-tune performance of simulation of large-scale scientific experiments.
- Used Python and Shell scripting for tracing and sampling data from experiments conducted on Edison-NERSC supercomputer and used Machine Learning techniques to improve performance and speed-up by 4x

Graduate Teaching Fellow, University of Oregon Dec 2015 – June 2016

- Lab Instructor for Introduction to Web Programming - Taught HTML/CSS and JavaScript
- Lab Instructor for Hands on with Internet of Things - Taught Python programming with Raspberry Pi

Software Engineer, Accenture, India Oct 2013 – Dec 2014

- Performed analysis and code changes in ABAP programming to correct functionality and usability issues related to the system. Successfully resolved over 50 critical system related issues affecting the client's business

Computer Vision Intern, Aeron Systems, India Dec 2012 – June 2013

- Emotion Based Music Player - Developed an application in MATLAB that utilizes real-time emotion recognition to play music depending on the facial expression displayed by the user

PROJECTS:

Full Stack Development – Personal Kan Ban

- Building a Web and Mobile productivity application (Personal Kan Ban) using the MERN stack (**MongoDB**, **Express**, **ReactJS**, **NodeJS**) with AWS, Google Calendar and Speech API integration

Internet of Things – Cluster controlled autonomous vehicles

- Developed an IoT system that allows Arduino based Ringo robots to work in communication with a central server (**MQTT**). Used Ricart-Agrawala and Supervisor-worker algorithms to develop the system
- Developed using **Python**, Raspberry PIs to form the cluster and **Arduino** for robot programming to simulate autonomous vehicles

Virtual Buttons – AR Android app

- Developed an Android application using **C#**, **Unity** and **Vuforia** to create Virtual Buttons to interact in an Augmented-Reality (AR) setting (download here: https://github.com/rajan3012/Virtual-Buttons-in-AR)

YouTube Data Analyzer using Hadoop

- Developed a Big Data analyzer for YouTube videos using **Hadoop MapReduce** to obtain top viewed and top categories from around 4,000,000 records in under 10 seconds

Compiler Development

- Built a compiler from scratch in **C++** to translate from Quack, an object oriented strongly types language, to C
- Developed a type checker for contra/covariance, recursion support, control flow with short circuit evaluation, and full polymorphism including dynamic dispatch mimicking C++ Virtual Method Tables

Simulation of Random Walk using MPI

- Designed and implemented a parallel Random Walk Simulation in **C++** using that calculated an iteration of multiple random walks over a big graph data set with 10,000,000 edges in under 20 seconds