Jason Sun's Resume

sunapi386@gmail.com sunapi386.ca linkedin.com/in/sunapi386 +1 (408) 599-0428 27/03/2019

Profile

- Full-stack autonomous system software engineer. Big data, mapping, localization, vehicle remote vehicle control, hybrid human-vehicle interactions, cloud to vehicle API. A generalist architect role who lead efforts to design and integrate multiple tech stacks to form new features, working in C++, Python, and Javascript.
- Driven and self-motivated personality with many hobbies (see blog at *blog.sunapi386.ca*).
- Over a decade of experience in coding, starting with scripting since 2004 on *Slackware Linux, Arch Linux, Debian/Ubuntu*, ***OS*** *X*.
- Experienced working in startup environments, at Velocity Residence and Velocity Garage. Conferences: *DEFCON (#25), DEFCON (#22)*, *2600 Hope 9*, PennApps 2013, MHacks 2014, HackMIT 2014, and HackZurich 2014.

Education

- *University of Waterloo* Bachelor of Computer Science, 2015.
- Computer Science Exchange Student at *EPFL (Swiss Federal Institute of Technology at Lausanne).*

Work Experience

- Autonomous Systems Engineer *AutoX Technologies Inc.* in August 2017 – present San Jose, California

Automating the mapping infrastructure for large scale map building used in lidar localization, with Hadoop, GraphQL, Node.js, React, GeoMesa DB.

Lead efforts to build API backend for location-based delivery service AutoX apps. General architect role, lead team efforts to design and integrate tech stacks for product feature development using C++, Python, Javascript.

Full-stack autonomous system integration, low/high-speed delivery vehicles. Experienced in SLAM, LOAM, lidars, cameras, radars, ultrasonics. Created features like: remote control (networked video streaming and throttle/steering control), hybrid human-and-AI decision making for unhandled driving scenarios (in patent process), design and implement vehicle-to-cloud APIs (LTE networked), embedded systems with touchscreen UI. Cluster computing automation using Docker, Kubernetes, ROS (Protobuf).

- Software Engineer *Apple Inc.* in November 2015 – January 2017 Cupertino, California

Improved existing product reliability tools, conceived and built full-stack website (Rails API & Ember.js & nginx & mySQL) for managing iOS devices. Created client-sided task runners for delegating test-work, where work jobs were queued from the website. Also worked on internal iOS and macOS apps (Swift & Objective-C).

- Undergrad Research Assistant at University of Waterloo in May 2015 – September 2015 Waterloo, Ontario

Worked with my AI professor and a 6 person team to develop a data tool for analysis of chat logs, and built a search engine prototype (using TF-IDF in Apache Lucene). Prototyped a search engine for customer support chat dialogues with Apache Lucene using TF-IDF indexing for feature recognition and searching.

- Software Engineer Intern *Shutterfly Inc.* in July 2014 – August 2014 Redwood City, California

Design and implemented a distributed REST API service, in Java/Scala based upon the Apache Cassandra database. The API was designed by myself, with guidance from full-time employee members of the web infrastructure platform team, to be used by other Shutterfly services. Create additional network load tests for our distributed image hosting service.

- Software Developer at *Encircle Inc.* in May 2014 – June 2014 Kitchener, Ontario

Worked on the website (CoffeeScript & Python Tornado server) and Android app, at a 3 engineer startup in the *Velocity Garage*, which is a University of Waterloo startup incubator.

- Software Tools Developer Intern at RIM (BlackBerry) in September 2013 – December 2013 Ottawa, Ontario

Built internally used features to GitLab (Ruby on Rails) and helped with database migration from Github to Gitlab. Developed a testing framework for integration and regression testing website user interfaces using the Selenium Webdriver.

- Physics Teaching Assistant at *Wilfrid Laurier University* in September 2011 – April 2012 Waterloo, Ontario

Developed a spectrometer reading program in Python, using the pySerial library, to automate the reading of lab samples, and generate a spreadsheet file. Supports multiple spectrometer readings in parallel.

Related Classes

- Real-time Operating Systems (CS 452): One project for the entire semester, which is to build a real-time microkernel and write user programs to control model trains on a track. The course is famously called the Trains course.
- Artificial Intelligence (CS 486): Worked on projects involving machine learning, learning probabilistic models, Bayesian networks, search and constraint satisfaction problems.
- Advanced Algorithms (EPFL CS 450): A graduate course in algorithms, learned theoritical techniques and their applications to solve problems. Interesting techniques such as network flow, randomization, dynamic programming.
- Computer Graphics (EPFL CS 440): Half the course was about graphics rendering techniques, and the second half about animation techinques. Implmented rendering methods into a ray-tracer (called *Nori*), and explored animation modeling and some fluid mechanics.
- AI: Intelligent Agent class (EPFL CS 430): Developed intelligent agents to pickup and deliver parcels in a simulated environment, with intelligent behaviours: reactive, deliberative, centralized, decentralized, and auctioning.
- Concurrency & Parallel Programming (CS 343): Multithreaded quicksort, implementation of well known concurrency control mechanisms, such as monitors. Language is in ***uC++***, a concurrent dialect of C++, developed at University of Waterloo.
- Distributed Systems (CS 454): Built a remote procedure call library in C++ on top of TCP, both client and server side. Implemented the go-back-N reliable transmission protocol, over UDP using Java.
- Computer Security (CS 458): Created exploits in **C** using techniques such as buffer overflow, and format strings. Implemented an intrusion detection program which parses output from tcpdump to detect spoofed packets, malicious hosts, and worms.

- Compilers (CS 251): Built a MIPS compiler in C++, parsing a subset of C keywords and generating assembly (MIPS) code.
- Computer Architecture (CS 450): Designed a pipelined CPU in Verilog, supporting 8 instructions for computer architecture class. This is sufficient to run machine code produced by the MIPS compiler, from the CS 251 compilers class.

Awards and Achievements

- *HackZurich 2014* at ETH Zürich, received Tamedia Digital Award (all-inclusive team trip to visit startups in Berlin, Germany) with *.GIFMeIt*: An iOS app that lets a user easily capture and share GIF images.
- *PennApps 2013* at University of Pennsylvania, received Twilio's Communication Award ($500) with *Marmoset*: A chatbot to respond to your chosen Facebook friends without them knowing.