Spencer Scorcelletti

> Full-stack developer, dabbles in systems engineering, just here to learn and make cool stuff.

1300A Kobbe ave, San Francisco, CA 94129 | 415.316.2003 | asimpletune@icloud.com

Work Experience

MTS Systems Engineer, Salesforce.com, May 2015 - Jan 2016

- Worked on an in-memory, distributed, graph-based database to model state of next-generation data centers
  - Designed method for continuing development ahead of having customer requirements, using an intermediary query language and client-specified schemas
  - Helped design/code MVP using go, protobuf, and embedded consul.io consensus library
  - MVP handled multiple-thousand requests per second, guaranteeing strong consistency, while tolerating benign faults
- Owned a platform for integrating internal, legacy development/bug-tracking tools with GitHub enterprise (**architecture and engineering - personal project**)
  - Handled authentication across platforms
  - Elegantly supported how to grandfather-in bugs tracked prior to this system, requiring no user intervention
  - Devised a way of extending new functionality through meta-data, for future-proofing further development
- Evangelized Docker development
  - Consulted with cloud architects on canonical Docker development practices
  - Reviewed Docker related code outside of team
  - Developed curriculum for in-house docker training material

MTS Software Engineer, Salesforce.com, Oct 2014 - Apr 2015

- Worked on a next-generation, internal CI system
  - Product scope was build/test/reporting across three platforms: web (Linux), iOS, and Android
  - Featured auto-scaling elastic compute resources, heavily parallelized builds/test-runs, self-service support and customized compute environments
  - Owned design/development of a templating system that allowed teams to customize and maintain their test environments as code in GitHub
- Owned an OS X elastic compute service (**architecture and engineering - personal project**)
  - Services run on top of highly available, gossip protocol layer, which provides consensus primitives
  - Clean API served authenticated, parameterized CRUD operations on compute nodes
- Served as team scrum master, mentored intern, moved team from Vagrant to Docker, and wrote team site
  - Advocated creating a team site as a top priority, which reduced time team spent on support by ~25%

AMTS Software Engineer, Salesforce.com, Jan 2014 - Sept 2014

- Named as an inventor on a patent with provisional status
- Full stack development of new features for Salesforce “Knowledge” (product)
- Fixed bugs, refactored old code, and wrote test automation

Software Engineering Co-op, Intel Corporation, May 2012 - Jan 2014

- Owned development of automatic robotic part handlers for post silicon validation (**architecture and engineering - personal project**)
  - Developed reliable pathfinding, self-calibration, and part handling capabilities for two $25K+ 6-axis robots
  - Abstracted work out to a robotic SDK that allowed different classes of robots (*6-axis, cartesian, etc…*) to be used interchangeably
  - Integrated with interlocks, emergency switches, part-reading camera, vacuum suction cups, and part actuators to run and record results of automated test in a database
- Wrote GUI app for analogue test engineers to interface, via Modbus protocol, with thermal controllers

Software Engineering Intern, Salesforce.com, May 2013 - August 2013

- Wrote majority of the front end development for Salesforce Knowledge’s redesign which shipped into production
- Extracted separate device views into a single, responsive view across devices, which saved time spent on development and maintenance in the future

Student Technology Assistant, UT Austin, May 2010 - Jan 2011

- Code PHP, CSS, JS, HTML for departmental websites at UT Austin

Education

University of Texas at Austin, December 2013

College of Natural Sciences, **Bachelor of Arts in Computer Science**

- **Thesis**: *Using DSP to Determine Volume of Containers*
- Selected Projects:
  - Implementation of Multi-Paxos
  - Program in asm that deletes everything in memory, including itself
  - Demonstration of how Bellman-Ford can beat Dijkstra by exploiting techniques learned in graduate-level performance class

Skills

Languages:

Node/JS, C#, Java, go, Swift, Objective-C/C, Python, Perl, x86

Tools/Technologies:

Docker, Consul, Serf, Packer, Vagrant, VirtualBox, Mongo, Digital Ocean, EC2 (to name a few important ones)

Areas of work/academic interest:

Linux Containers, Distributed Systems, Software Design, Operating Systems and Computer Architecture, Virtualization, developer tools and automation, continuous integration

Personal Interests

Making things, VW buses (1969-1979), music, reading, art, and nature