Matthew Do

Code	github.com/minhnhdo
Email	m@minhdo.org

Work Experience

Hopper Inc. (hopper.com) – Software Developer – Growth **March 2020 – Present**

- Designed and implemented major pieces of software infrastructure for Growth services
- Investigated bottlenecks and optimized the core service for Growth resulting in reduction of request processing time by two orders of magnitude
- Leveled up other engineers via knowledge transfer at multiple level of abstractions (what the code is, what was intended, what the long term vision is)
- Volunteered to work across multiple business verticals to help deliver results

Technologies used: Scala, Python, PostgreSQL, Kafka, Pub/Sub

Viki Inc. (viki.com) – DevOps Engineer **May 2016 – June 2017**

- Developed a development environment (codenamed nDev) capable of running the whole Viki infrastructure on one machine, based on Docker in Docker and PPTP for VPN
- Maintained uptime of all Viki infrastructure

Leadbook Pte Ltd (leadbook.com) – Software Engineer **Jun 2015 – Apr 2016**

- Simplified codebase by removing overdesigned and redundant subsystems
- Implemented optimizations
  - Reduced page load time for list view of data by 4 times by not rendering redundant data
  - Reduced request serving time for bulk operations by two orders of magnitude by using bulk update and partial record/document update APIs

Viki Inc. (viki.com) – Platform Engineer **Jun 2014 – May 2015**

- Containerized services for speedy recovery measures, easy deployment and better security
- Worked with Amazon S3, Redis, RabbitMQ, PostgreSQL, Ruby and CoffeeScript to deliver high performance backend API to increase viewer engagement (see viki.com/celebrities for an example)
- Maintained and improved uptime of the backend API
- Developed integration testing tool for multi-service black box testing

Projects

PGo – distributed system compiler for PlusCal model checking language **Oct 2017 – May 2019**

- PGo (github.com/UBC-NSS/pgo) translates PlusCal specifications to distributed Go programs

doco – documentation generation using static and dynamic analysis **Mar 2018**

- doco generates, as code comments, pre- and post-conditions for Java methods using static analysis, dynamic analysis, and Rust

Education

University of British Columbia (MSc in Computer Science) **Sep 2017 – May 2019**

- Teaching assistant
  - Functional and Logic Programming: helped students learn to program in Prolog and Haskell
  - Distributed Systems: helped students build distributed systems (toy blockchain, Raft, and Paxos implementations), and built automated grader for assignments
- Research assistant
  - PGo: a distributed system compiler from PlusCal to Go

Nanyang Technological University (BEng in Computer Science) **Aug 2010 – May 2014**

- Teaching assistant (developed lab manual materials)

Skills

Programming Languages	Scala, Rust, Go, Ruby, JavaScript, Clojure, Python, C, C++, Java
Tools	vim, git, Redis, RabbitMQ, Kafka, Pub/Sub, PostgreSQL, docker, linux