Xiaofan Hu's Resume

Summary

- Name: Xiaofan Hu
- Email: xiaofan.xhu@gmail.com
- Phone: +86 15224019502
- GitHub: https://github.com/bom-d-van
- Blog: https://www.xhu.buzz/

I have almost 10 years of experiences being a software engineer, covering from front end all the way to the back end systems and are familiar with scaling and developing highly concurrent, highly available distributed systems.

My current focus in mainly in high available infrastructure, distributed systems, database kernel development, and system programming.

- Programming languages: Go, C, C++, Bash, Perl, Java, JS, Ruby, Python, Rust.
- Tech: Linux, TSDB (Graphite), SQL DB (MySQL/PostgreSQL), Redis, Ruby on Rails, EBPF, SLO, SRE, uWSGI, Puppet, Protobuf/GRPC, ReactJS/jQuery/Backbone/ExtJS/etc.

October 2017 - July 2022 (Booking.com Amsterdam)

- Title: Developer/Senior Developer
- Company scale: 2000 Tech/10,000+ Total
- Tech: Time Series Database, Go, Perl, Large Scale Distributed Systems, Site Reliability Engineering, Kubernetes, Helm, System Programming, Ansible, Puppet, Java

Achievements:

- Scaling large scale distributed time series systems (Graphite)
  - Scale: over 1k servers, over 1 PB SSD storage, 500 million uniq metrics, 2k QPS, 30m data points per second on ingestion.
  - By adopting Facebook Gorilla compression algorithm, design and implement a new file format for compression that reduces disk space usage from 30% - 70%.
  - Design and implementing a new index algorithm by using NFA+DFA algorithms introduced by Russ Cox that is able to support 10 - 40 millions uniq metric paths with low indexing overhead and low tail latencies.
  - Introducing a quota subsystem for reliability and control that are able efficiently
  - Design and implement a lockless and concurrent trie indexing that were able to reduce memory usage and supports real time indexing
  - Design and implement a higher performant Quota subsystems that able to reduce the noisy neighbor effect in a multi-tenant environment and achieves efficient resource management and control
  - Optimize the rebalance tool for the systems, by introducing a self-regulated mechanism using health check and jitters, the changes has produced faster and adjustable sync rate and enhances observability.
- Scaling and maintaining a large scale distributed config management system
  - Scale: 8k RPS, over 1300 backend systems and over 60k end points
  - Scaling the API backend to support high concurrency and high availability
  - Define and implement SLI and SLO metrics, including Availability, Propagation Latency, Error Rate, Request distributions across different backends and roles, etc.
  - Optimize Perl clients
- Scaling large scale distributed config management system
  - Scale the API backend to support 60k endpoints
  - Define and implement SLI and SLO for monitoring propagation latencies, usage and scale RPS on a per role basis.
- Site Reliability Engineering
  - Maintain multiple medium and small backend roles (servers ranging from 10 to 100s), debug production issues and being on-call.
  - Design and implement an internal auto-capacity testing system targeting envoy based systems by interacting with an internal control plane API.
  - Design and implement an uWSGI timeout callback mechanism for logging automation that's called Graceful Harakiri
  - Research and implement EBPF based tooling for debugging production issues
  - Debug and resolve a storage leakage Bug in Hashicorp Vault production system.
- Develop and scale a new fast partner signup channel/product.

September 2016 - August 2017 (UCloud Shanghai)

- Title: Web Development Engineer
- Company scale: 500+
- Tech: Go, Bash, C++, Linux, TC, Networking, DPDK

Achievements:

Maintain and develop the traffic control system which is an important part of networking control, using C++, Go, Bash etc.
Responsibilities and Experiences included:

- Develop an accounting system for bandwidth usage monitoring and operations
- Use Quagga and Keepalived to implement Redis high availability across different data centers in the same region
- Develop new ingress traffic control feature in
- System problems on-call, writing up scripts for data consistency checking and log error monitoring

May 2013 - August 2016 (The Plant Hangzhou/Tokyo)

- Title: Developer
- Company scale: 50+
- Tech: Go, Bash, Ruby on Rails, Linux

Achievements:

- Mainly Worked and maintained two EC projects in Go and a few other smallish projects like product recommendations (by using Mahout), and a react project. Responsibilities and Experiences included:
  - 3D bin packing algorithm
  - Order and Payment(first GMO, then migrated to Stripe) User register/login, products management
  - Memory leak problem fixes
  - Unit/Integration tests
  - System deployment and maintenance etc
- Worked on Qortex, a communication web application designed and made for high performance teams. Experiences and Personal highlights included:
  - VirtualBox Packaging with auto-updates support for Enterprise users Go package management (Pak)
  - Deployment/migration automation script (developed later into Harp) Email processing/sending (Beanstalkd, SMTP)
  - Chatting (Ejabberd Integration) API maintenance

May 2010 - Feb 2013 Student Developer in Wando Lab, HuiZhou University

Involved in the design and development of two ERP (Enterprise Resource Planning) systems in RoR and Ext JS. It's a great learning experience and an excellent environment for helping improve programming and collaboration skills.

Open Source Projects

- Go-Carbon: Graphite Storage in Go.
- Harp: A Go application deployment tool.
- 3D Bin Packing: A Golang 3D Bin Packing Implementation
- AssetTube: A tool for fingerprinting and serving asset files for Go Web applications.
- CHTTP: A stupid and incomplete http/http2 C implementation, built for learning C.
- plperf: A tracing program for uwsgi+perl environment, using ebpf and perl dtrace, in Go.
- Pak: A Go package version management tool.