

# 胡小帆简历

## 基本信息

- 姓名: 胡小帆
- 邮箱: xiaofan.xhu@gmail.com
- 电话: +86 15224019502
- GitHub: <https://github.com/bom-d-van>
- 博客: <https://www.xhu.buzz/>

近十年的互联网工作经验, 涵盖了前中后端, 高可用, 高流量分布式系统的开发和维护。

目前工作重心主要放在高可用基础架构服务, 分布式系统, 数据库内核开发, 和系统编程。

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

## 主要工作经验

### 2017年10月 - 2022年7月 (Booking.com 荷兰阿姆斯特丹)

- 职位: 研发工程师/高级研发工程师
- 公司规模: 2000研发/1万多非研发
- 技术栈: Time Series Database, Go, Perl, Large Scale Distributed Systems, Site Reliability Engineering, Kubernetes, Helm, System Programming, Ansible, Puppet, Java

主要经验:

- 研发和维护高可用大型分布式时序数据库Go-Graphite (存储和查询方向)
  - 规模: 1千多台服务器, PB级数据量(SSD), 5亿Unique Metrics (3 replicas), 2k+ QPS, 3千万data point每秒的Ingestion
  - 设计和实现了基于[Facebook的Gorilla论文](#)中描述的Gorilla[时序数据压缩算法的新存储文件结构](#), 达到了30%-70%的压缩率。
  - 设计和实现了基于Russ Cox的DFA算法的[Globbing查询算法的前缀树索引](#), 实现了高性能的索引效率和低延迟的查询, 支持单台服务器高效索引10-40 Millions Metrics。
  - 设计和实现了[Lockless and concurrent 前缀树索引](#), 减少了数据库内存消耗, 支持实时索引新数据
  - 设计和实现了高性能的[Quota子系统](#), 减少了多租户系统中的Noisy Neighbour和实现有效资源管理和控制
  - 优化了集群Rebalancing tool, 引入了[基于health check的自调节速率机制](#), 在避免导致服务器负载过高的情况下优化了sync的速率以及提高了系统可监控性。
- 扩展和维护高可用配置分发系统 (Scaling large scale distributed config management system)
  - 规模: 8k RPS, 支持1300多个后端系统和6万多个客户端
  - 拓展API后端支持高并发和高可用
  - 定义和实行SLI和SLO, 包括Availability, Propagation Latency, Error Rate, 后端系统请求分布情况等
  - 优化Perl客户端
- Site Reliability Engineering
  - 运维管理多个中小型后端系统(规模从10到100多台服务器), debug和oncall各种生产环境问题
  - 实现了针对公司内部envoy控制面的自动压测系统
  - 实现了uwsgi上超时回调机制用于收集线上系统[Graceful Harakiri](#)
  - 研发和使用EBPF工具debug[在线上产系统问题](#)
  - Debug和解决了Hashicorp Vault生产系统的[存储泄漏的Bug](#)
- 研发和拓展新fast partner signup channel

### 2016年九月 - 2017八月 (UCloud 中国上海)

- 职位: 研发工程师
- 公司规模: 500+
- 技术栈: Go, Bash, C++, Linux, TC, Networking, DPDK

主要经验:

- 维护和研发公司云平台的基于Linux tc的流量控制系统。
- 研发了带宽操作的对账系统。
- 使用Quagga和Keepalived实现了Redis的跨机房高可用
- 研发新的ingress流量控制下发服务
- 研发数据一致性检测的脚本和错误日志监控

### 2013年五月 - 2016八月 (The Plant 中国杭州/日本东京)

- 职位: 研发工程师

- 公司规模: 50+
- 技术栈: Go, Bash, Ruby on Rails, Linux

主要经验:

- 研发和维护了两个电子商务相关的项目
  - 调研和实现了一个3D bin packing算法, 自动化了打包
  - 集成基于Mahout的产品推荐算法
  - 对GMO和Stripe的订单和支付系统的集成
  - 用户注册登陆, 产品管理模块的研发
  - 生产系统的内存泄露问题的调查和解决
  - 单元测试和集成测试
  - 基于React的前端开发
- 办公写作系统Qortex的部分功能的开发
  - 基于Virtual Box的企业版系统的打包和部署的自动化
  - 开发和部署的自动化
  - 基于SMTP和Beanstalkd的邮件系统的集成和处理
  - 基于Ejabberd聊天子系统的集成和开发

## 2010五月 - 2013二月 (惠州大学Wando实验室)

参与两个ERP系统的设计和研发。主要技术栈是Ruby on Rails和ExtJS。

## 开源项目

- [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 ebp and perl dtrace, in Go.
- [Pak](#): A Go package version management tool.

# 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 is mainly in high available infrastructure, distributed systems, database kernel development (both TSDB and general), 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.