

Madhav Jivrajani

Bangalore, India | Email: madhav.jiv@gmail.com

[GitHub](#) | [LinkedIn](#) | [Blog](#)

EDUCATION

PES University

Bangalore, India

Bachelor of Technology in Computer Science and Engineering (CGPA: 9.36/10)

Aug. 2018 – May. 2022

- With Specialization In *Systems And Core Computing*

Relevant Coursework: Performance Engineering, Cloud Computing, Database Management Systems and Technologies, Compiler Design, Heterogeneous Parallelism, Generic Programming, Big Data, Operating Systems, Computer Architecture, Design and Analysis of Algorithms

Awards: CNR Rao Merit Scholarship (awarded to the top 20% in the department)

SKILLS

Programming Languages: Golang, Python, C, C++

Technologies: Kubernetes, Containerd, Prometheus, eBPF, etcd, gRPC

Technical Skills: Distributed Systems, CRDTs, Systems Modelling and Verification (Queuing Theory and TLA+), Performance Debugging Go Systems, git

Non Technical Skills: Open Source Governance, Community Management

EXPERIENCE

Member of Technical Staff - {1, 2} (K8s Maintainer, SIG ContribEx TL)

Aug. 2021 – Present

VMware

Bangalore, India

- Improve Scalability of The Kubernetes Storage Layer
 - * Successfully understood and identified bottlenecks in the **Kubernetes** API Server Watch Cache.
 - * Implemented a solution to reduce lock contention in the watch cache and reduce the API Server's memory footprint and make it more predictable.
 - * Implemented a solution to evolve the watch cache to serve paginated LIST calls by implementing a BTree based cache, hence saving a trip to etcd.
 - * Investigating de-duplicating serialisation/de-serialisation of objects in the storage layer to significantly reduce CPU cycles.
- Improve Reliability of Kubernetes Storage Layer
 - * Unify testing of the caching layer and etcd for the Kubernetes API Server.
 - * Fix and harden bugs discovered in the process related to Kubernetes *resourceVersion* semantics.
- Shepherd Kubernetes Releases
 - * Ensure Kubernetes is released securely, reliably and on time, by tracking releasing blocking failures/flakes and regressions.
 - * Work across cross-cutting technical areas within the Kubernetes project (api-machinery, scalability, release, testing) to resolve CI failures and regressions.
 - * Work with open source communities that Kubernetes depends on (most notably: *the Go project*) to resolve CI failures and regressions.
- Act as Technical Lead For SIG Contributor Experience
 - * Provide technical guidance internally and in the community, on developing GitHub automation for the Kubernetes project.
 - * Drove implementation of automation to democratise the code approval process in the Kubernetes project to increase reliability, PR velocity and sustainability of the project.
 - * Developed tools for other maintainers to help gauge PR impact and project health.

Software Development Engineering Intern

May. 2021 – Aug. 2021

Akamai Technologies India Pvt Ltd.

Bangalore, India

- Redesigned a legacy python microservice to be extensible and testable.
- Migrated microservice application to Golang and implemented a pluggable communication media to switch between REST and gRPC protocols as and when needed.
- Added end-to-end testing for the microservice, exposing a REST API, covering interactions with a PostgreSQL database.
- Added resiliency measures to the Go client in forms of exponential back-off and jitters.

SELECTED TALKS AT CONFERENCES AND MEETUPS

- **The Kubernetes Storage Layer: Peeling The Onion:** *KubeCon + CloudNativeCon, Nov. 2023* [talk][slides]
- **Keep CALM and CRDT On:** *Papers We Love Bangalore, Oct. 2023* [slides]
- **Reliably Absorbing A Go Release: Learnings From The K8s Community:** *GopherCon, Sep. 2023* [slides]
- **Using eBPF To Debug the Performance of The Go Scheduler:** *Go Bangalore, July 2023* [talk][slides]
- **Control Theory and Concurrent Garbage Collection: The Go GC Pacer:** *GopherCon 2022* [talk][slides]
- **Queues, Fairness, and The Go Scheduler:** *GopherCon 2021* [talk][slides]
- **Adventures In Optimizing The Kubernetes API Server:** *Go Bangalore, September 2022* [talk][slides]
- **Control Theory, Controllers and Kubernetes:** *OpenInfra Days Asia 2021* [talk][slides]
- **Imperative, Declarative, and Kubernetes:** *Kubernetes Community Days Bangalore 2021* [talk][slides]

A full list of my talks can be found [here](#).

SELECTED AWARDS

- **Google Open Source Peer Bonus Award, 2023:** For contributions to the Kubernetes project
- **Kubernetes Contributor Award, 2021:** For contributions to SIG Architecture

CONFERENCE VOLUNTEERING

- **Paper Review Committee:** *GopherCon 2022, GopherCon 2023*
- **Program Committee and Track Chair:** *KubeCon + CloudNativeCon NA 2022*

SELECTED PROJECTS

- gse:** *Go Scheduler Exporter* [code][talk] Oct. 2021 – Dec. 2021
- Implemented a way to export the Go runtime's scheduler traces to **Prometheus**.
 - Implemented a Prometheus exporter to export scraped and processed metrics from the Go runtime.
 - Implemented detection of Goroutine preemption using both the **Linux Tracing Subsystem** and **eBPF**.
 - Used this tool to demo and explain the internals of the Go scheduler using **Grafana** at GopherCon 2021.
- graph-based-auto-scaling:** *A service graph aware auto-scaling algorithm* [code] Jan. 2020 – Dec. 2021
- Used concepts from **queuing theory** to design an auto-scaling algorithm for microservices that was aware of the service graph and optimised for end-to-end throughput, with Kubernetes as the choice of orchestrator.
 - Used **Istio** as the service mesh along with the **Kiali** plugin to access service graph information along with topology and network metrics.
 - Deployed and used the Kubernetes **metrics-server** and subsequently developed a Go client for it, to scrape resource usage metrics to feed into our queuing model.
- btree-indexer:** *A BTree backed Kubernetes client-go Indexer* [code] Feb. 2022 – Present
- Implemented a BTree based cache defined by Kubernetes client-go's Store interface.
 - Implemented Indexing on top of the cache to implement a BTree backed Indexer interface.
 - Used this cache in optimizing the Kubernetes API Server Watch Cache in how it serves LIST calls.
- locknt:** *Concurrent and Lock-Free Data Structures In Go* [code] Jan. 2021 – May. 2021
- Implemented concurrent and lock-free data structures in Go.
 - Wrote benchmarks to try and understand bottlenecks in the implementation.
 - Profiled the code using **pprof** as well as **perf** to try and understand the effects of false sharing and the cost of mitigating these effects.
- grofer:** *A TUI based system and resource monitoring tool written in Go* [code] Jun. 2020 – Present
- Implemented a generic way to scrape metrics from various sources and feed them into the desired sinks.
 - Implemented methods to scrape low-level metrics (system-wide and per-process) from the **/proc** file system.