Karan Joisher

https://karanjoisher.github.io
karanjoisher@gmail.com | +1 857-588-3119

Work Experience

Software Engineer 2 at Hewlett Packard Enterprise, Massachusetts, USA — Jun 2024 – Present

- Built NVGrid, a high-performance C++ RDMA library powering distributed object storage clusters, enabling east-west data transfer rates up to 400 Gb/s.
- Developed observability tooling to capture FlameGraphs, switch packet drops, and network statistics; to help diagnose performance bottlenecks.
- Tuned switch QoS policies, reduced lock contention and thread starvation, and balanced internode vs. disk I/O bandwidth, increasing cluster throughput from 128 Gb/s to 400 Gb/s.
- Replaced VMware ESXi with HPE VM Essentials and SR-IOV, enabling deployment of RDMA-capable Kubernetes clusters for development and testing, saving $3,500 per host annually in licensing costs.
- Developed a Producer–Consumer RDMA protocol to replace a Client–Server model, reducing internode operations per transaction from 4 to 3 and lowering latency by 5%.
- Developed an error injection API over gRPC that injects faults directly into the RDMA device context (CQ, async event queue), allowing validation of error-handling paths rarely triggered under organic workloads.
- Developed NAE scripts for Aruba CX 9300 switches to handle interface split and speed requirements, eliminating manual configuration and enabling factory units to support all HPE storage products.

Software Developer at Barclays, Pune, India — Jul 2019 – Jul 2022

- Created 12 RESTful APIs for a high-volume banking app (2500 TPM), streamlining loan payments for customers.
- Implemented performance monitoring library and parallelized API calls to improve performance by 30%.
- Led the full-stack development of a tool for bank agents that automated analysis for 60,000+ bank debt cases, saving 232 hours weekly (equivalent to 5.8 FTEs).
- Designed distributed and scalable AWS microservices to create repayment cases for delinquent accounts received from Apache Kafka, reducing the latency for case creation by 24 hours.
- Developed CloudFormation templates to provide one-click deployment of auto-scaling ECS clusters, MSK clusters, and RDS on the AWS cloud platform; achieving a 40% cost reduction through demand-based scaling.

Technical Skills

- Programming Languages: C, C++, Java, Python, Rust, Golang, C#, x86 and x64 assembly, GLSL
- Libraries: SpringBoot, Boost, gRPC, RDMA IBVerbs, OpenGL, Win32 SDK, CUDA
- Tech: AWS, Kafka, Redis, CMake, Bazel, GDB, MySQL, Docker, Kubernetes, Git

Education

- **Khoury College, Northeastern University, Boston, USA** — Sep 2022 – Apr 2024
  Master of Science in Computer Science – GPA 4/4
  Coursework: Operating Systems (Awarded Rockstar Programmer), Computer Networks

- **K.J. Somaiya College of Engineering, Mumbai, India** — Aug 2015 - May 2019
  Bachelor of Technology in Computer Engineering – GPA 8.56/10
  Coursework: Compilers, Distributed Systems

Projects (https://karanjoisher.github.io/projects)

Debugger

- Built a low-level Linux debugger supporting both x86 and x64 processes.
- Added support for custom breakpoints, giving users precise control to pause execution at arbitrary instruction addresses for debugging.
- Added execution control features: step-in, step-over, and continue; allowing users to manage program flow post-breakpoint.
- Enabled users to read and write CPU registers to instrument process state and control execution.
- Enabled visualization of process virtual memory in both hex and string views, providing deeper insight into program data and state.

Cross-Platform Multimedia Library

- Developed a cross-platform C++ library for Windows and Linux, designed to help developers create multimedia applications and games.
- Graphics: Implemented API to dynamically load the OpenGL DLL and resolve function pointers for cross-platform rendering support.
- Window Management: Added support for window creation, control, and image buffer access to facilitate custom rendering pipelines.
- Inputs: Implemented handling of keyboard and mouse events to enable interactive applications.
- Audio: Designed audio stream management and playback system with latency of 3ms for real-time applications.
- File: Provided a unified API to manage files and virtual memory pages across platforms.

3D Model Viewer

- Implemented a perspective projection camera from scratch to view the model from various angles.
- Built a Phong illumination model that integrates ambient, diffuse, and specular material properties to simulate realistic lighting effects.
- Added shadow-mapping support to simulate realistic light occlusion, allowing models to cast accurate shadows and improving visual depth.

Self-Driving Car Simulator

- Created a self-driving car simulator to accelerate prototyping of autonomous vehicle systems without requiring physical hardware.
- Added pedestrian and vehicle traffic using A* pathfinding and Reciprocal Velocity Obstacles (RVO) to simulate realistic driving environments.
- Created a plug-and-play interface for ML models, exposing simulation data and controls to train and test self-driving algorithms.