Rajeshwar **Vanka**

STAFF SOFTWARE ENGINEER / TECHNICAL LEAD MANAGER

Cupertino, CA

☎ 919-607-6889 | ✉ rajeshwar.vanka@gmail.com | rvanka | rajeshwarvanka

Profile

Versatile Staff Software Engineer and Technical Lead Manager with expertise in databases, distributed systems, technical infrastructure and compilers. Adept at identifying and resolving complex engineering challenges, leading cross-functional teams, and consistently delivering high-quality, performant solutions that meet strategic objectives.

Work Experience

Cloud Spanner, Google

Sunnyvale, CA

STAFF SOFTWARE ENGINEER / TECHNICAL LEAD MANAGER *Dec 2021 - PRESENT*

- **TL/M for Yahoo® Mail's transition to Cloud Spanner. Led a team of 9 SWEs and TLs.**
- Alignment with Director, UTL, PM and Senior Managers to ensure on-time delivery.
- Designed Distributed Rate Capping and Token-bucket based resource management in Cloud Spanner that scales to 1000s of nodes.
- **TL/M for Cassandra Endpoint for Spanner. Led a team of 6 SWEs and TLs.**
- Go-to person for the project: Ideation, alignment, conflict resolution, prototyping, implementation, communication
- Brought multiple customers from negotiation to revenue commit with PoCs and interactions with internal and customer-facing engineers within and outside Google.
- Unblocked scalability limits within Cloud Spanner, improving scaling in Spanner's large regions.

Google-Wide Profiling/AutoFDO Technical Lead (Tech Infra), Google

Sunnyvale, CA

SENIOR SOFTWARE ENGINEER *Oct 2018 - Dec 2021*

- GWP collects and visualizes profiles from fleet to identify efficiency opportunities.
- AutoFDO uses LBR profiles from machines to improve efficacy of compiler optimizations.
- Overhauled the AutoFDO infrastructure with automatic profile generation, and improved stack unwinding accuracy.
- Designed, implemented and deployed fleet-wide context-sensitive profiles to deliver massive fleet-wide compute and RAM savings.
- Designed automatic, customized AutoFDO profiles that scales to all C++ binaries within Google.
- Multiple internal awards for delivering savings and performance gains.
- YouTube® CDN: Uncovered and fixed lock contention leading to 5x faster rollouts for new releases and recovery.

Matlab Core Execution Engine, The MathWorks

Natick, MA

SENIOR COMPILER ENGINEER *Feb. 2013 - Oct. 2018*

- Part of the team that designed and implemented the JIT that shipped in MATLAB® R2015b.
- Designed and implemented various optimizations at the language and builtin level in Interpreter and JIT compilers while maintaining current semantics.
- Designed scalable tools to analyze MATLAB® code within and outside the company, to drive optimization strategies.
- Mentored projects involving various parts of the compiler, each with a well-defined scope and delivery timeline.

Lawrence Livermore National Labs / Redpine Signals

INTERN

- **Redpine Signals**: Designed and implemented a CORDIC Co-Processor in Verilog, using MATLAB® and Synopsys® tools for verification.
- **LLNL**: Designed and implemented a source-level runtime error checking tool to detect array out of bounds, null pointer dereferences, memory leaks, dangling pointers and uninitialized variables for C and C++.

Education

North Carolina State University

Raleigh, NC

PH.D IN COMPUTER ENGINEERING *Aug. 2007 - Feb. 2013*

- Advisor: Dr. James M. Tuck
- Thesis: Efficient Data Dependence Profiling
- Developed a novel technique for quickly and accurately for gathering runtime data dependencies between may-alias loads and stores. This information is then fed back into the compiler to perform speculative optimizations.
- Implemented the profiler and feedback optimization system in LLVM as a standalone tool that works on bytecode (IR).

North Carolina State University

Raleigh, NC

M.S IN ELECTRICAL AND COMPUTER ENGINEERING *Aug. 2007 - May 2009*

- GPA: 3.86/4.0
- Courses: Code Generation and Optimization, Code Optimization for Scalar and Parallel Programs, Computer Design and Technology, Advanced Computer Microarchitecture, Parallel Systems

ICFAI Institute of Science and Technology

Hyderabad, India

B.Tech in Electronics and Communication Engineering

Aug. 2003 - July 2007

- GPA: 9.27/10
- Class Rank: 5/100

Publications

Lightweight runtime checking of C programs with RTC

CLSS 2016

Reed Milewicz, Rajeshwar Vanka, James Tuck, Daniel Quinlan, Peter Pirkelbauer

Journal Paper

- RTC is a tool for instrumenting C programs with run-time safety checks. It handles memory violations, arithmetic underflow/overflows, and type violations.
- The instrumented code can be compiled by any C99 compliant compiler. The runtime framework is extremely lightweight, making it useful for in situ analysis. Tool has an average run-time overhead rate of 9.7× and memory overhead rate of 3.5× across a range of benchmarks.

Runtime Checking C Programs

SAC 2015

Reed Milewicz, Rajeshwar Vanka, James Tuck, Daniel Quinlan, Peter Pirkelbauer

Conference Paper

- This paper presents RTC, a runtime monitoring tool that instruments unsafe code and monitors the program execution.
- In tests against known error detection benchmarks, RTC found 98% of all memory related bugs and had zero false positives.

Efficient and Accurate Data Dependence Profiling using Software Signatures

CGO 2012

Rajeshwar Vanka, James Tuck

Conference Paper

- A novel set-based profiler combined with an optimization driven approach for fast and accurate data dependence profiling.
- Software signatures used to perform quick membership and insertion operations on sets.
- Multiple knobs expose a large design space spanning various performance and accuracy tradeoffs.

Efficient Data Dependence Profiling for Speculative Load Hoisting

PLDI 2011

Rajeshwar Vanka, James Tuck

ACM SRC Poster

- Fast and accurate data dependence profiler using bloom filters.
- Accurate profiling with slowdowns of 22x on average versus an aggressive baseline. LLVM IR level instrumentation with feedback into unmodified optimization passes.

Enhanced Memoization using Function Splitting

PACT 2009

Rajeshwar Vanka, James Tuck

ACM SRC Poster

- Static function transformation to expose and eliminate hidden redundant calls.
- Functions exhibiting read-write dependences are split, to create new regions where redundancies can be exposed by existing memoization techniques.

Grants & Patents

Computing performance for execution of functions called by function handles

US Patent: 10949181

Rajeshwar Vanka, Ayon Basumallik, Brett Baker

Granted: Mar 16, 2021

- Lead inventor on the patent that scales a JIT compiler's ability to handle large programs with many function handles without imposing a runtime cost.
- Specializing the handles on types moves always-paid runtime cost to a smaller one-time compile cost.

A Practical Data Dependence Profiler for Program Characterization and Optimization.

NSF Grant 1320356

James Tuck

- I contributed to this proposal and my doctoral research work formed the basis for this project proposal.
- This project focuses on the design of a fast practical data dependence profiling that works effectively for a wide range of applications and for a wide range of program analysis needs.

Programmable CORDIC Processor

US Patent 8332451 B2

Phanimithra Gangalakurti et. al

- The internship work at Redpine Signals was subsequently patented.
- This patent covers the multi-stage programmable design of the CORDIC Co-Processor.

Skills

Technical Expertise	Distributed Systems, Databases (Spanner, Cassandra), Compilers, Computer Architecture, Profiling
Platforms	Linux, macOS™, Microsoft® Windows™, Google Cloud
Languages	C, C++, MATLAB®, Python, Perl, Java, Rust, Go, CUDA, OpenCL, OpenMP, Verilog, VHDL, Shell
Compilers	LLVM, GCC, ROSE
Profiling Tools	Instruments, PIN, oprof, Intel® VTune™, eBPF, LBR, pprof, Stackdriver
Development Environments	VSCode, Vim, Atom, Xcode, Eclipse, Microsoft® Visual Studio, Emacs, MATLAB®,

Teaching

Compilers 150 students. Graduate level course introducing parsing, IR generation and x86 code generation.

Computer Architecture 200+ students. Graduate level course introducing processor pipelining and branch prediction.

Embedded Systems 70 students. Graduate level course using Renesas®QSK26A.