

Gaurav Jain

SUMMARY

Software engineer with 20 years of experience across the stack — machine learning infrastructure, distributed storage, operating system kernels, and embedded firmware. Currently building secure sandboxed execution environments and fine-grained access controls for LLM agents at Snowflake.

HIGHLIGHTS

- Scaled Snowpark from 3% of product revenue to a \$100M+ run rate.
- Top 50 all-time TensorFlow contributor.
- Founding engineer at Rubrik, a 10-person startup that grew into a 1,000+ person rocketship.
- Founding iPhone team member — built its Bluetooth stack ("Best of the Breed" award).

WORK EXPERIENCE

Principal Engineer, Snowflake

2022 – Present

Cortex — CoWork / Snowflake Intelligence — *Go, ML Platform Infrastructure, LLM Agent Infrastructure, Secure Sandbox Containers, Databases*

- Built zero-latency secure sandboxes for agent code execution — model-generated code starts instantly, fully isolated, without leaving the customer's data boundary.
- Designed strong access controls enabling fine-grained database access policies, giving customers precise control over the scope of agent capabilities.

Snowpark — *Python, C++, ML Platform Infrastructure, gVisor, Operating Systems, Secure Sandbox Containers, Databases*

- Scaled Snowpark from 3% of product revenue to a \$100M+ run rate — over 100 million queries per day from more than half of Snowflake's customers.
- Rebuilt sandbox infrastructure on top of gVisor, strengthening isolation and program compatibility at zero performance cost — 1.5% faster than the legacy in-house sandbox on full benchmarks.
- Optimized the gVisor runtime — cut virtual-memory bookkeeping (VMA entries) by 182x on memory-heavy workloads, and improved ARM compatibility for ML libraries with better CPU virtualization.
- Built a secure pipeline for Python package delivery with sub-second dependency solves — a 99.95% solver-cache hit rate that cuts query initialization latency by ~85% in production.
- Publications:
 - [SEE++: Evolving Snowpark Execution Environment for Modern Workloads](#)
 - [Snowpark: Performant, Secure, User-Friendly Data Engineering and AI/ML Next To Your Data](#)

Tech Lead / Engineering Manager (TLM), Google Brain — Mountain View, CA

2018 – 2022

TensorFlow — *Python, C++, Performance Optimization, Machine Learning, Deep Learning, AI*

- Delivered core functionality for the next-generation TensorFlow 2.0 framework, including development of GPU/TPU accelerator management, missing kernel functionality, as well as API testing and consistency.
- Optimized the core eager runtime, achieving a 2x training speedup for dynamic models, matching compiled graph mode performance on ResNet50 and improving TPU startup time by 50x.

ML for Systems — *Python, C++, Machine Learning, AI, Mixed-Integer Programming, Databases*

- Built and evaluated applied-ML techniques that replaced hand-tuned heuristics in Google's backend systems.
- Projects included ML-guided query planning — Kepler, a 2.41x geometric-mean speedup over PostgreSQL's optimizer with near-zero regressions — ML-optimized block caching, and ML-tuned compiler optimization passes.
 - Published: [Kepler: Robust Learning for Faster Parametric Query Optimization](#)
 - Patented: [Autonomous Column Selection for Columnar Cache](#)

Founding Engineer / Engineering Manager, Rubrik — Palo Alto, CA

2014 – 2018

- Early engineer as Rubrik grew from 10 people to more than 1,000 within 4 years.

- Patented work across the distributed file system and data management stack:
 - [Cluster-Based Network File Server](#) — a highly available file system optimized for ingesting backups from virtualized environments.
 - [Throttling Network Bandwidth Using Per-Node Network Interfaces](#)
 - [Chunk Allocation](#)

Polaris — *Go, Python, Kubernetes, Microservices, GCP, Machine Learning, Leadership*

- Led backend development for Rubrik's ML-powered cloud data management platform.
- Designed key components and service interactions, such as RPCs and database schema management.
- Built remote communication protocol with transport fault tolerance and agent discoverability.

Jarvis — *Python, C++, Graphite, ElasticSearch, Stats/Log Analysis, Monitoring, AWS, REST API*

- Implemented cross-language performance & monitoring framework for comprehensive system analysis.
- Enabled proactive customer support with real-time log & stats alerting engine using AWS stack.

Forge — *Python, C++, Scala, Cassandra, Salesforce, Networking, Linux, Manufacturing*

- Built cluster management layer, enabling API-based cluster operations and node self-healing.
- Designed and implemented API-based network configuration such as bonding, VLANs, routing & failover.
- Utilized Salesforce to provide an integrated asset tracking system for operations, sales & support.

Senior Software Engineer – Kernel, Facebook

2012 – 2014

Skills: C, C++, Lua, Python, Linux Kernel, Tracing, Networking, Performance/Latency, iOS, Obj-C

- Designed and built kernel-based tracing tool enabling complete system-level overview during abnormal events such as high latency, lock contention and application sluggishness.
- Developed an IPVS scheduler enabling consistent hashing of network packets across load balancers.
- Prototyped networking patches to reduce network latency for latency sensitive applications.
- Built performance framework to identify inefficiencies in image fetching logic within iPhone app.

Senior Software Systems Engineer – Core OS, Blue Coat Systems — Waterloo, ON

2009 – 2012

Skills: C, C++, x86/x86_64 Assembly, RTOS, Concurrent & Parallel Computing, File Systems

- Enhanced kernel and file system from 32-bit, uni-processor to 64-bit Multi-Processor aware.
- Re-architected components of the core operating system to scale across multiple processors.
- Prototyped, designed and implemented performance improvements to the file system which included CPU optimizations, disk defragmentation, file cache efficiency and disk I/O reductions.
- Refactored codebase to be highly object-oriented and unit testable. In addition, built a testing framework to enable developer tests to be run on development machines as well as target devices.
- Mentored junior team members through design reviews and technical knowledge transfer.

Senior Embedded Software Engineer, Logitech — Mississauga, ON

2008 – 2009

Skills: C, C++, iPhone, ARM/Thumb, RTOS, QNX, NAND, RF4CE, Cortex-M3

- Developed and debugged the high-end remote software platform, including areas such as NAND file system, RF4CE drivers and system middleware.
- Designed a graphics framework and scalable data model for the next generation Harmony remote platform, offering enhanced functionality at a reduced Bill Of Materials cost.
- Developed cross-platform canvas platform to develop apps across embedded devices & iPhone.

Software Engineer – iPod Touch/Nano & iPhone, Apple — Cupertino, CA

2006 – 2008

iPod — *C, C++, ARM, RTOS, EFI, Power Management, Nordic, DMA, I2S, I2C, JTAG*

- Developed 3rd party accessory communication framework for inbuilt accessories such as Nike+.
- Brought up new hardware platforms through driver and boot loader implementation.
- Optimized components of the system to cater to better power saving methodologies. Subsystems included audio engine, video engine, application frameworks, drivers and core operating system.
- Developed middleware and frameworks to support the use of new technologies for applications running on a real-time operating system.

iPhone — *C, C++, Objective-C, Bluetooth, WiFi, Darwin/Mac OS X, iOS, Quality of Service (QoS)*

- Brought up iPhone embedded Bluetooth stack. Achieved Bluetooth "Best of the Breed" Award.

- Developed a cross-platform code base to enable debugging of the software stack on multiple platforms including Mac OS X, Windows, Linux and an RTOS environment.
- Worked closely on improving Bluetooth/WiFi coexistence to enable a high user experience.
- Designed other Apple wireless technologies such as Nike+ and the iPhone Bluetooth Headset.
- Patented: [Group Formation Using Anonymous Broadcast Information](#) — sharing anonymous "tokens" over existing wireless protocols.

WLAN & VoIP Software Developer, BlackBerry — Waterloo, ON

2005

Skills: C, Java, TCP/IP, RTP, SIP, WiFi, DSP, GPRS, SDIO, ARM

- Developed and debugged RTP and SIP stack protocols for the first WiFi BlackBerry devices.
- Ported drivers for a new wireless network card chipset to interact with existing code base.
- Re-designed code segments into a more modular architecture to enable rapid development.

SKILLS

- **Programming Languages:** C++, C, Python, Go, Objective-C, Scala, Java, x86/x86_64 Assembly, ARM
- **ML & Agent Infrastructure:** TensorFlow, GPU/TPU Accelerators, ML for Systems, LLM Agent Infrastructure, Secure Sandboxing (gVisor)
- **Systems:** Embedded Systems, File Systems, Distributed Systems, Concurrent and Parallel Computing, Networking, Protocol Architecture, Bluetooth, WiFi, Power Management, Performance Tuning, Kernel Tracing
- **Operating Systems:** Linux, RTOS (RTXC, Unison), Mac OS X, iOS, QNX
- **Infrastructure & Tools:** Kubernetes, GCP, AWS, Git, Graphite, ElasticSearch

OPEN SOURCE PROJECTS

python-diamond — Maintainer & Developer

Skills: Python, Monitoring

- Maintain Diamond, a Python daemon that collects system and application metrics and publishes them to Graphite and other backends.
- Review and merge community contributions, and develop new collectors and core improvements.

CodeWeavers/WineHQ — Developer & QA Tester

Skills: C, Win32 API, Git, Scripting

- Provided patches to the WineHQ project to improve compatibility on the Mac platform.
- Developed installation scripts for the operation of Windows games on CrossOver for Mac.
- Tested and validated applications during CrossOver beta & release cycles.

Pygments — Developer

Skills: Python, Objective-C, C, C++, Unicode, RTF, Mercurial

- Improved Objective-C, C & C++ parsing and highlighting.
- Bug fixes to various components such as the RTF formatter.

EDUCATION

Master of Applied Science (MASC), University of Waterloo

2011 – 2013

- Research: RR-TM, a runtime for eager software transactional memory (C++, LLVM, x86 assembly) — designed a lower-overhead STM API, an LLVM pass rewriting shared-memory accesses into runtime calls, and a path-sensitive alias-analysis optimization.

Bachelor of Mathematics — Honors Computer Science (Software Engineering Option), University of Waterloo

2002 – 2006