Stefan Vercillo

✉ svercillo7@gmail.com ☎ (917)-488-2282 stefanvercillo.com svercillo svercillo

Education

University of Waterloo, *BASc Computer Engineering, Varsity Soccer* Sep 2018 – May 2023
Systems Programming & Concurrency (97%), Programming Fundamentals (95%), Programming for Performance (96%), Algorithms & Data Structures I/II, Distributed Computing, Computer Security, Database Sys Implementation, Operating Systems

Skills & Aptitudes

Languages: Java, C++, Python, Rust, C, SQL, GoLang, Groovy, JavaScript, TypeScript, C#, Bash, PHP
Tools: Kafka, Kubernetes, Gradle, Maven, Docker, Terraform, ArgoCD, Git CI/CD, Redis, Kinesis, AWS, GCP, Azure
Libraries/Frameworks: Apache Flink, Hadoop, Apache Spark, Spring, Django, Cuda, React, Vue.JS, UDP/TCP networking

Professional Experience

Palantir, *Software Engineer – Streaming Compute Team* Sep 2022 – present | New York, NY

- Reengineered the stream archiving and replay processes by optimally creating and merging virtual shards with a heuristic dp algorithm, using barrier-based synchronization of task managers across the reading of archive transactions to maintain per key ordering, achieving a **100x parallelism/throughput** improvement dropping replay time from months to hours in cases
- Implemented **AIP** logic board functionality allowing streams to interface **LLMs** to provide real-time intelligence enrichment
- Innovated streaming ingest method utilizing auto-scaling containers for flexible data population, enabling edge use cases such as live audio calls that seamlessly integrate with a live AIP-driven recommendation engine
- Developed a multi-job per **Flink** cluster streaming compute architecture, significantly enhancing resource utilization
- Optimized resource utilization of streaming architecture by supporting multiple streaming jobs simultaneously per cluster
- Created a feedback system for dynamic **Flink** scaling based on predictive usage profiling, leveraging metrics like TCP back pressure, and CPU utilization to determine the ideal pod instance count and adjust the Kubernetes cluster conf accordingly
- Developed an ETE test framework for streaming workflows, seeing a **80%** drop-off in P0s the first quarter implemented

Twitter, *Software Engineer – Kernel and Encryption Team* May 2022 – Aug 2022 | San Francisco, CA

- Refined, packaged and deployed **booster,** an improved initramfs image generator, to **10k+** Twitter production machines
- Created **Go** based **RPM** package, working with **Koji** and an internal artifact registry to build the **RPM** in a virtualized chroot
- Used **Puppet CI/CD** to deploy and configure booster across hardware and OS configs, improving boot speed by **15%**
- Enabled full-encryption at rest using multiple device mappers, **Clevis TPM**, and network **Tang** bindings with booster

Federato AI, *DevOps Engineer* Dec 2021 – May 2022 | San Francisco, CA

- Refactored infrastructure mono-repo into logical modules deployed on **Terraform Cloud** with **Github Action** triggers, enabling mass scaling of infrastructure, clients, and engineering power through tight isolation and code versioning
- Recreated **CD pipeline** into declarative **GitOps** CD to trigger deployments with **ArgoCD** on a pull basis within the cluster

TradeLogiQ / Omega ATS, *Quantitative Developer* Jan 2021 – May 2021 | Toronto, ON

- Interfaced the TSX, optimizing **C++** retransmission service for missing UDP sequences, lowering E2E latency by **70%**
- Developed a multithreaded **Python** service to monitor and validate the correctness of our real-time trading system, using both **UDP** for speed and **TCP** for reliability, whilst incorporting concurrency safeguards for logical consistency

Enso Connect, *Software Engineer* Dec 2020 – May 2021 | Toronto, Ontario

TD Bank, *Quantitative Developer* May 2020 – Sep 2020 | Toronto, ON

Sky View Suites, *Software Engineer* Sep 2019 – Jan 2020 | Toronto, ON

Ford Motor Company, *Software Engineer* Jan 2019 – May 2019 | Waterloo, ON

Projects

Database Systems Implementation, *C++*

- Created a relational query cost analyzer by evaluating materialization costs of temp tables, hash/tree index vs table scan selection costs, and considering byte reduction through projections and selection order using provided schema & statistics
- Developed a query tree visualizer and optimizer by intelligently exploring a subset of new indexes, rebalancing the query tree to prioritize pipelined joins, and maximizing byte reduction by strategically pushing selections and projections