Erik Alapää

Location: Zurich, Switzerland

linkedin.com/in/alapaa/ | Email: erik.alapaa@gmail.com | Mobile: +41 77 265 14 66

Hands-on, research-trained tech lead, software developer and near-metal C++ expert at Google

Software developer in Borg(*) SRE at Google. Specialist in Unix and hard real-time C/C++. Research degree (Licentiate) in Applied Mathematics from Chalmers, Gothenburg. M.Sc. in Engineering Physics from Luleå University of Technology. +20 years of industry experience, mostly on **near-metal programming in C and C++**. Worked on everything from **192k RAM** embedded controllers up to Google's world-scale infrastructure.
(*) *Borg is the cluster operating system that controls all Google datacenters and runs almost all infrastructure at Google.*

Roles I am looking for: Pure SWE, research, architect, tech lead.

Technical Skills

Languages	:	C++, C, Python
Databases	:	Google Cloud BigQuery
Dev Tools	:	Clang, GCC, GDB, Git, Valgrind (Memcheck, Cachegrind, Callgrind, Helgrind), Clang Thread Safety Analysis, Compiler Explorer (godbolt.org)
Frameworks	:	Boost Graph Library, Pthreads, gRPC, Google Protobuf (open source), Google Abseil C++ Common Libraries (open source)

Career highlights, examples of my expertise (full history at linkedin.com/in/alapaa/)

SRE, Borg Cluster Operating System — Feb 2019 – Present
Google — *Zurich, Switzerland*

- Single-handedly developed automation for the machine management part of Borg turnup, removing a very time-consuming process for my team's oncallers. Turned up 50 Google Edge clusters in Q1 2021, dramatically improving the sharding, redundancy and reliability of Google's Edge cluster network.
- As the **Edge cluster turnup lead** of a small team of SREs developed and deployed improved automation that **entirely removed cluster controller machine management toil and costs** (by moving to a model where cluster controllers do not need dedicated machines).
- Designed and implemented a solution for turndown cleanup in the Aristotle (C++ based) machine management system. This saved Google very large amounts of money by not blocking subsequent large cluster turnups, and also eliminated hours of toil every week for me and other oncallers in the SRE team. I also designed safety checks preventing the cleanup to be run on production cells, which could have caused large outages.
- Modified handling of open-source Protocol Buffer extension code (C++), saving a couple of bytes per message. This **saved 3 TB of RAM across the Google datacenter fleet**.
- **Taught all 3 Google C++ courses** to hundreds of Google engineers during 2021-2023.

TSE, Technical Solutions Engineer — Aug 2017 – Jan 2019
Google — *Zurich, Switzerland*

- 3rd line expert support of Google Cloud Big Data products (e.g. BigQuery, Dataflow, Pub/Sub, Tensorflow) .
- Specializing in the architecture of BigQuery, an extremely fast, mostly-read SQL database, handling **terabytes** of queries in seconds.
- Worked with premium customers, developers and SRE to **debug the C++ shuffler part of the database engine**, solving slowdowns and failures of large queries and speeding up query execution by a factor of **10-20 times**.

C and C++ SW developer, tech lead — Jan 2014 – Jun 2017
Netrounds — *Luleå, Sweden*

- Occasional **Linux kernel patches** in CPE:s (home routers/WiFi access points). One prominent example was when I added support to the Broadcom kernel and user-space API for tapping the built in layer 2 hardware (the switch), making packets visible to the host kernel. This enabled our measurement software to access the packets in user-space.
- Research and hands-on development in C++ of hardware timestamping of network packets, **improving accuracy and precision of measurements by a factor of 1000**. I did all the research, wrote the requirements specification and wrote the time-critical core measurement functions and also lead a small team of developers in this effort.

- Extensive study of the Linux kernel networking code, for example to **understand the path MTU detection logic** and use it fully in our custom IP measurement software.
- Developed a solution using SR-IOV for **near bare metal latency RTT measurements** by mapping the hardware queues of a network interface card directly to a virtual machine, bypassing the host kernel.

C++ SW developer, consultant 2012-2013
Ericsson *Stockholm, Sweden*

- **Advanced debugging of multi-threaded software** in the mobile base station radio software. (Pthreads and OSE RTOS processes) . One example was finding an initialization bug caused by an older C++ compiler not emitting locks around static initialization at the startup of the system.
- C++ development in an extremely constrained **hard real-time** environment, **64K data RAM and 128k instruction RAM**, the system for linearization in the mobile radio base station transmitter . Using my research background to interact with mathematicians and signal processing experts to realize the implementation. Working hands-on in the lab verifying the implementation directly on the target RISC CPU.

Tech lead, SW developer (pure C), consultant 2010-2011
Combitech/Ericsson *Stockholm, Sweden*

- Shared tech lead of a project to develop a distributed, scalable, telecom-grade load balancer using the built-in Linux kernel load balancer for **+50 million connections**, including termination of hardware-encrypted IPSec connections.
- I wrote the requirements specification for the **Fault Tolerance and High Availability** design of the load balancer system.
- **Developed a coroutines solution in C** for the control plane of the load balancer to avoid spawning new threads when operators logged in to configure the system using a Cisco-style router CLI.
- Worked with Ericsson legal experts on **open-source licensing** to deploy the coroutines solution.
- The load balancer project contributed fixes to the then-new **Linux kernel container/network namespace code** (for example fixing issues in the VETH interface logic), and most prominently, the HMARK patch in the kernel firewall (Netfilter) code that is foundational for load balancing in Linux.

Education

Chalmers University of Technology Gothenburg, Sweden
*Licentiate **(research degree) in Applied Mathematics**. Specialization in wavelets and Fourier analysis. Aug 1999 – Nov 2004*

Thesis on partial differential equations and error-correcting codes for efficient transmission of multimedia over 4G mobile radio (Ericsson Resarch), implemented in C++.

Luleå Technical University Luleå, Sweden
Master of Science in Engineering Physics, specializing in Applied Mathematics *Aug 1993 – Aug 1999*

Thesis on surface reconstruction using laser-scanned point cloud samples, implemented in C++.

Certifications

- **C++ master level, Brainbench**.
- Achieved Google C++ readability, allowing me to submit changelists without separate readability review.

Additional information

- Nationality: Swedish, lived in Switzerland the last 6 years with my wife and 9-year old son. Grew up above the Arctic Circle.
- Got my first computer, Electrolux ABC 80, in 1982. **Started writing M68K assembler on my Amiga 500 1988**. First full-time position as SW developer at ABB Signal, Stockholm 1992, developing fail-safe hard real-time systems for trains. Started learning and using C++ professionally 1993.
- **Six languages**: Swedish (native), English (excellent), Russian (fluent spoken Russian), German (very good understanding, only rudimentary spoken German), Finnish (very basic), French (very basic)
- Interests: Political economy, economic history, technical history, energy technology including solar, nuclear and hydrogen storage. Avid e-biker.