

# DANIEL DUCLOS-CAVALCANTI

## Computer Engineer

55th Street New York, New York 10019

516-912-7975 | U.S. Citizen | [me@duclos.dev](mailto:me@duclos.dev) | [www.duclos.dev](http://www.duclos.dev) | [linkedin/duclos-cavalcanti](https://linkedin/duclos-cavalcanti) | [github/duclos-cavalcanti](https://github/duclos-cavalcanti)

## Education

### Technical University of Munich

Oct 2020 – Oct 2024

*M.Sc. Electrical and Computer Engineering*

*Munich, Germany*

- Visiting Non-Degree Graduate Student: **New York University – GPA 4.0**
- Master Thesis: **VM Selection Heuristic for Financial Exchanges in the Cloud**
- **Related Coursework:** Operating Systems, Machine Learning Methods, High Performance Computing Lab

### Technical University of Munich

Oct 2016 – Sept 2020

*B.Sc. Electrical and Computer Engineering*

*Munich, Germany*

## Publications

### Design and Implementation of A Scalable Financial Exchange in the Cloud | (*Paper*)

Jan 2024 – Present

- Novel Cloud financial exchange achieving low latency of  $\leq 250 \mu s$ , with a difference  $< 1 \mu s$  for 1K receivers.
- Achieves better scalability and around 50% lower latency than the multicast service provided by AWS.
- Used kernel-bypass techniques (DPDK) to scale performance up to a 35K multicast packet rate.

## Experience

### Research Assistant

Jul 2022 – Oct 2022

*TU Munich*

*Munich, Germany*

- Worked on TensorDSE, a Design-Space Exploration framework to guide machine learning model deployments.
- Evaluated the performance of various ML models across GPUs, CPUs and TPUs with TensorFlow Lite.
- Generated cost analysis reports for Google's Coral Edge TPU via USB traffic analysis (PyShark) during inference.
- TensorDSE used reports to accelerate a model's inference/deployment optimally onto available hardware devices.

### Embedded Software Engineer – Internship

Aug 2021 – Jan 2022

*Molabo GmbH*

*Ottobrunn, Germany*

- Added unit-tests (GTest) and test coverage (lcov) to safety critical features of their motor's embedded controller.
- Developed tooling for state simulations of their electric motor via Linux's virtual CAN interface and mock APIs.
- Extended their firmware update system used by 18+ clients, consisting of partial updates via CAN bus.
- Automated build and testing workflows via Jenkinsfiles, Makefiles and CMake for a team of over 10 engineers.

### Tutor - Embedded Systems Programming Lab

Apr 2021 – Aug 2021

*TU Munich*

*Munich, Germany*

- Mentored over 20 students in developing low-level FreeRTOS applications in C for embedded systems.
- Taught software engineering best practices, focusing on concurrency, real-time scheduling, and performance optimization.

## Technical Skills

**Languages:** C++, Python, Golang, Rust, C, Bash, JavaScript, HTML, CSS, Lua, VHDL

**Cloud Services:** Google Cloud Platform (GCP), Amazon EC2 (AWS), Terraform, Packer, Vagrant

**Tools:** Linux, Unix Shell, Git, Github CI/CD, Jenkins, CMake, GNU Make, Bazel, Vim, VSCode

**Technologies:** Docker, ZeroMQ, DPDK, MPI, FreeRTOS, FPGA, IoT, TensorFlow, Scipy, NumPy, Pandas, OpenMP

**Verbal/Written:** German – Fluent, Portuguese – Fluent

## Projects

### Cloud-TreeBuilder | GCP, ZMQ, Terraform, Python, C++, Distributed Systems, Heuristic

Mar 2024 – Present

- Launches and selects K out of N VMs in a cluster to create an optimal multicast tree of depth D and fan-out F.
- Deploys UDP based probe jobs on VM subsets, collecting data regarding their network performance (JSON).
- Applies a developed heuristic on collected data to select VMs for a tree layer by layer.
- Uses terraform to manage cloud state, ZMQ for node communication and Protobufs for data serialization.

### Open-MPI Value Iteration | C++, Parallel-Computing, MPI, HPC

Mar 2022

- An HPC prototype that solves a stochastic navigation problem through Asynchronous Value Iteration (AVI).
- Used different MPI techniques to iteratively distribute workload across an HPC cluster and gather results.

### Hamming Code Error Detection (16,11) | C, VHDL, FPGA, SoC, UART

Feb 2021

- Implemented an error detection/correction algorithm for packet transmission on Microsemi's SF2 FPGA/SoC.
- Packets sent between host and SoC via UART are cross-referenced against erroneous bit flips.
- Microprocessor offloads error-injected data onto the FPGA for detection/correction via an APB3 Bus Matrix.