Daniel Duclos-Cavalcanti

Computer Engineer
55th Street New York, New York 10019
XXX-XXX-XXXX | U.S. Citizen | daniel@duclos.dev | www.duclos.dev | linkedin/duclos-cavalcanti | 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** (2023 – 2024)
- Master Thesis: **"VM Selection Heuristic for Financial Exchanges in the Cloud"** – (*TreeBuilder*)
- **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 <= 250 µs, with a difference < 1 µs for 1K receivers.
- Achieves better scalability and around 50% lower latency than the multicast service provided by AWS.
- Enhanced performance using kernel-bypass (DPDK) to reach up to a 35K multicast packet rate per second.

Experience

Research Assistant **Jul 2022 – Oct 2022**
TU Munich *Munich, Germany*

- Collaborated on TensorDSE, a Design-Space Exploration framework to accelerate machine learning model deployments.
- Assessed the performance metrics of multiple ML models across GPUs, CPUs and TPUs with TensorFlow Lite.
- Generated cost analysis reports on Google's Coral Edge TPU via USB traffic analysis (PyShark) during inference.
- Established that on average up to 57% of total inference time consists of data transmission with the external TPU.
- TensorDSE consumed reports to map a model's deployment optimally onto an available set of hardware devices.

Embedded Software Engineer Intern **Aug 2021 – Jan 2022**
Molabo GmbH *Ottobrunn, Germany*

- Increased test coverage (GTest) up to 25% on safety-critical motor controller features, identifying and resolving bugs.
- Developed state simulation tooling with Linux's virtual CAN interface to validate motor functionality in real-time.
- Extended the firmware update system for over 18 clients, enhancing reliability of partial updates via CAN bus.
- Automated build and testing workflows via Jenkinsfiles, Makefiles, and CMake, supporting a team of over 10 engineers.

Tutor (Embedded Systems Programming Lab) **Apr 2021 – Aug 2021**
TU Munich *Munich, Germany*

- Mentored over 12 students on designing and developing low-level embedded FreeRTOS applications in C.
- Conducted 30+ sessions on best practices in software engineering, concurrency, performance, and real-time scheduling.

Technical Skills

Languages: C++, Python, Golang, Java, 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**

- Optimally selects K out of N VMs in a cluster to form a multicast tree of depth D and fan-out F, minimizing latency.
- Deployed UDP-based probe jobs on VMs, gathering network performance data for informed heuristic selection (JSON).
- Integrated Terraform for cloud state management, ZMQ for node communication, and Protocol Buffers for serialization.
- Improved multicast latency up to 24% for a cluster of 25 VMs and multicast tree of depth 3 and fan-out 2.

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).
- Leveraged MPI to iteratively distribute workload across an HPC cluster, executing 52% faster than in single-threaded.

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.
- Error-injected packets sent between host and SoC via UART and offloaded to the FPGA for detection/correction.