Nick-Andian Tehrany

MSc Computer Science Student

Balthasar van der Polweg 1010, 2628ZJ Delft, The Netherlands

☎ +31 6 11077890 | ✉ nicktehrany1@gmail.com | 🏠 https://nicktehrany.github.io | nicktehrany | in nicktehrany

I'm currently a MSc Computer Science Student at the TUDelft with my main areas of interest in Storage & Memory Technologies, Operating Systems, and Distributed Systems.

Education

MSc Computer Science

Delft, The Netherlands

Delft University of Technology — *September 2020 - PRESENT*

- **Track - Software Technology**
- **Study Focus - Distributed Systems**

BSc Computer Science

Amsterdam, The Netherlands

Vrije Universiteit Amsterdam — *September 2017 - August 2020*

- **Minor - Deep Programming**
- **Thesis - ”Evaluating Performance Characteristics of the PMDK Persistent Memory Software Stack”**
  In this thesis I investigated the performance characteristics of persistent memories by analyzing the affects on performance of DAX-enabled file systems bypassing the page cache running on emulated persistent memories, compared to the performance of conventional file systems running on emulated RAM disk-block devices. Additionally, I explored the performance of the Persistent Memory Development Kit (PMDK) and identified software overheads within these libraries.

Honors & Awards

Amsterdam Data Science Thesis Award

Amsterdam, The Netherlands

Vrije Universiteit Amsterdam — *December 2020*

- I received the Amsterdam Data Science Thesis Award in 2020 in recognition of my BSc thesis on ”Evaluating Performance Characteristics of the PMDK Persistent Memory Software Stack”.

Honors High School Diploma

Woodland Hills, CA

El Camino Real Charter High School — *2015*

- I received an honorary High School diploma in 2015 at the El Camino Real Charter High School in Woodland Hills, CA.

President's Award for Educational Achievement

Woodland Hills, CA

El Camino Real Charter High School — *2015*

- I received the Presidential Award for Educational Achievement in recognition of my academic achievement throughout high school.

Skills

Propgramming	C, Java, C++, Bash, Python, Scala, LaTeX, Haskell, Lean, R, JavaScript
Frameworks & Tools	IJVM, LLVM IR, UNIX, Docker, PostgreSQL, Cassandra, Kubernetes, fio, perf, ftrace, filebench, PMDK, Kafka, RocksDB, QEMU
Languages	German, English

Projects

Developing, Deploying, and Benchmarking Micro-Services with Cassandra and PostgreSQL in a Kubernetes Cluster

Delft

Web-Scale Data Management Course — *2021*

- Develop micro-services for an online shopping service using Python Flask, with PostgreSQL and Cassandra as backends. Deploy the services and backends on Google Cloud as containerized Docker instances in a Kubernetes Cluster, and benchmark the different backends, comparing performance and scalability.

Automatically Optimizing Data Layout on Flash-based Storage Devices

Amsterdam

AtLarge Research - Massivizing Computer Systems

2021

- Voluntary project at the AtLarge Research group, in which we are building a framework for automatically optimizing data layout on flash-based storage devices, especially open-channel SSDs.

Wide Area Adaptive Streaming

Delft University of Technology

Distributed Systems Course

2021

- Implement a streaming system with an adaptive scheduler that will re-plan and re-deploy query tasks across servers based on current performance metrics of the system. Said metrics are collected from servers periodically and result in an optimized query planner solving an Integer Linear Program for finding optimal task placement, followed by logical and physical plan adaptation based on the newly calculated optimal task placement.

Distributed Algorithms

Delft University of Technology

Distributed Algorithms Course

2021

- Program three different distributed algorithms; an algorithm for causal message ordering on point-to-point messages, an algorithm for ensuring mutual exclusion in distributed systems, and an algorithm for creating minimum weight spanning trees in asynchronous networks using Java RMI calls.

membench - Benchmarking Memory and File System Performance

Vrije Universiteit Amsterdam

BSc Computer Science Thesis

2020

- During my BSc thesis for evaluating performance characteristics of persistent memory software, I developed several micro-benchmarks for evaluating several system performance metrics, including a micro-benchmark for measuring latency of memory accesses through in memory pointer chasing, a micro-benchmark for measuring and tracing mmap calls for different file systems, and a micro-benchmark for evaluating the required time for handling of page faults on different file systems.

Compiler for FenneC

Vrije Universiteit Amsterdam

Compiler Construction Course

2019

- Built a full compiler for FenneC, a C-like language, with a frontend for translating source code into LLVM IR, which is compiled into executable binary using Clang. The compiler also implements numerous LLVM passes for security checking and code optimizing.

Secure Chat

Vrije Universiteit Amsterdam

Secure Programming Course

2019

- Implement a secure chat client using RSA for key encryption, AES for message exchange, and Certificates for server and database authentication. The project is fully implemented in C and utilizes a server for registering clients and session keys for clients, a database for storing message history and salted passwords, and ssl-sockets connection for communication between different clients.

Multi-Core NDFS

Vrije Universiteit Amsterdam

Concurrency & Multithreading course

2019

- Implement a multi-threaded version of a nested depth first search (NDFS) algorithm to be deployed and performance benchmarked on a cluster.

Shell & Memory Allocator

Vrije Universiteit Amsterdam

Operating Systems Course

2018

- Built a shell for executing UNIX system commands, and implement a full heap allocator similar to jemalloc and ptmalloc2.

Emulator for IJVM Byte-Code

Vrije Universiteit Amsterdam

Project Application Development Course

2018

- Implement an emulator for executing IJVM byte-code in C. It supports stack memory, basic functions and control flow, as well as method invocation by parsing binary files and extracting IJVM headers.