

# Emil Tsalapatis

\* *Website*: <https://watchedkettle.com>

\* *Github Account*: <https://www.github.com/etsal>

\* *E-mail*: [emil@etsalapatis.com](mailto:emil@etsalapatis.com)

## Currently Working at:

### University of Waterloo

PhD Researcher

September 2018 - June 2024 (Expected)

Waterloo, ON, Canada

- Led the Aurora operating system project that totals 3 PhD researchers & 5 undergraduate students
- Published papers at top venues (SOSP, ASPLOS, HotOS)
- Thesis Topic: OS Support for Application Persistence. Aurora (*SOSP 2021*) is a FreeBSD kernel extension that implements automatic application crash recovery. The system eases development by letting developers write applications as if they never crash.
- Followup projects include a serverless invoker for  $\mu$ s cold starts (under submission); an OS persistence API for databases (*ASPLOS 2024*); and checkpoint-based distributed fault tolerance. Please see *Selected Projects* for details.
- Wrote 40KSLOC of FreeBSD kernel code over 5 years, spanning the kernel's memory management, system call, and file system/buffer cache subsystems.

## Selected Projects

(Lead designer and developer unless noted otherwise)

### Aurora Single Level Store

2018-Ongoing

- FreeBSD kernel subsystem for continuous application checkpoint/restore
- Transparently persists running applications at a high-frequency (100x/s)
- 25 KSLOC spanning the kernel's virtual memory manager and buffer cache
- Aurora has 10x less stop time than userspace systems (CRIU)

### MemSnap: An OS API for Efficient Database Persistence

2022-2023

- OS API and kernel code to efficiently persist memory-mapped data
- Addresses the correctness and scalability problems of `mmap-and-fsync`
- API compatible with existing applications, retains their ACID guarantees
- RocksDB and SQLite have 2x higher throughput when using MemSnap
- 3 KSLOC of virtual memory-related kernel code, 1 KSLOC of database code

### Microsecond-Latency Serverless Invoker

2021-Ongoing

- Serverless invoker that minimizes serverless function cold start latency and resource usage
- Invokes functions using snapshot images for fast cold starts of 1 ms on average
- Uses paravirtualization to safely share common dependencies, lowering memory usage
- Aimed towards private clouds, supports both OS containers and virtual machines
- Supports multiple languages including Python, JavaScript, Rust, Ruby, and C

### In-kernel copy-on-write (COW) File System

2019-2021

- File system w/ COW snapshotting similar to ZFS and Btrfs, creates 100s of snapshots per second
- Targets applications that require millisecond-granularity data versioning, uses NVMe SSDs
- Achieves performance parity with production file systems, e.g., ZFS, for write-heavy workloads
- Designed the first version of the FS, worked on the core data structures of later versions

## Education

### Computer Science

PhD Degree

Focus Areas: Operating Systems, File Systems, Cloud Computing

University of Waterloo

September 2018 - June 2024 (Expected)

Thesis Topic: Efficient Fine-grained Application Persistence with a Single Level Store OS

## Electrical and Computer Engineering

Joint BSc & MSc Degree

Final Grade: 8.56/10 (top 7%)

Thesis Topic: utmem: Towards memory elasticity in cloud workloads

National Technical University of Athens (NTUA)

September 2012 - May 2018

## Past Work Experience

---

### University of Waterloo

Sessional Lecturer/Teaching Assistant

September 2019 - December 2023

Waterloo, ON, Canada

- Lecturer for the CS350 Operating Systems course (300 students) during the Winter 2021 semester
- Organized a team of 12 teaching assistants to support students with office hours
- Designed and authored the course assignments on top of the CastorOS educational OS
- Designed and coded the assignment submission system, acted as system administrator

### Computing Systems Lab, NTUA

Undergraduate Researcher

September 2016 - May 2018

Athens, Greece

- Designed a system on the KVM hypervisor for virtual machine memory elasticity w/o ballooning
- Designed a Linux OS API to integrate utmem with applications running inside VMs
- Integrated Redis with the utmem mechanism to evaluate utmem's performance
- Presented a peer-reviewed publication on the work at the International Supercomputing Conference

### Christian-Albrechts Universität Kiel

Programmer (Internship)

June 2017 - August 2017

Kiel, Germany

- Designed and implemented a CPLEX-derived Java framework for linear programming
- Framework used by the Computer Science department for research on online bin-packing algorithms

## Collaborator In:

---

### Record-Replay & Checkpointing for Distributed Fault-Tolerance

2023-Ongoing

### Capability-Based OS API for Secure Containers

2022-2024

### Intel SGX-based Decentralized Key-Value Store

2021-2022

## Selected Publications

---

E. Tsalapatis, R. Hancock, R. Hossain, and A. J. Mashtizadeh, "MemSnap: A Data Single Level Store for Fearless Persistence," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS 2024, Association for Computing Machinery, 2024.

E. Tsalapatis, R. Hancock, T. Barnes, and A. J. Mashtizadeh, "The Aurora Single Level Store Operating System," in *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles*, SOSP '21, (New York, NY, USA), p. 788–803, Association for Computing Machinery, 2021.

E. Tsalapatis, R. Hancock, T. Barnes, and A. J. Mashtizadeh, "The Aurora Operating System: Revisiting the Single Level Store," in *Proceedings of the Workshop on Hot Topics in Operating Systems*, HotOS '21, (New York, NY, USA), p. 136–143, Association for Computing Machinery, 2021.

## Technical skills

---

### Programming Languages

C, Python, Bash Scripting, Familiarity with C++ and Rust

### Kernel Development

Virtual Memory, File Systems, Hypervisors

### Operating Systems

Linux, FreeBSD, Plan9

### System Software & Databases

RocksDB, Redis, SQLite, KVM, Docker