# Efficient Computing in a Safe Environment

10

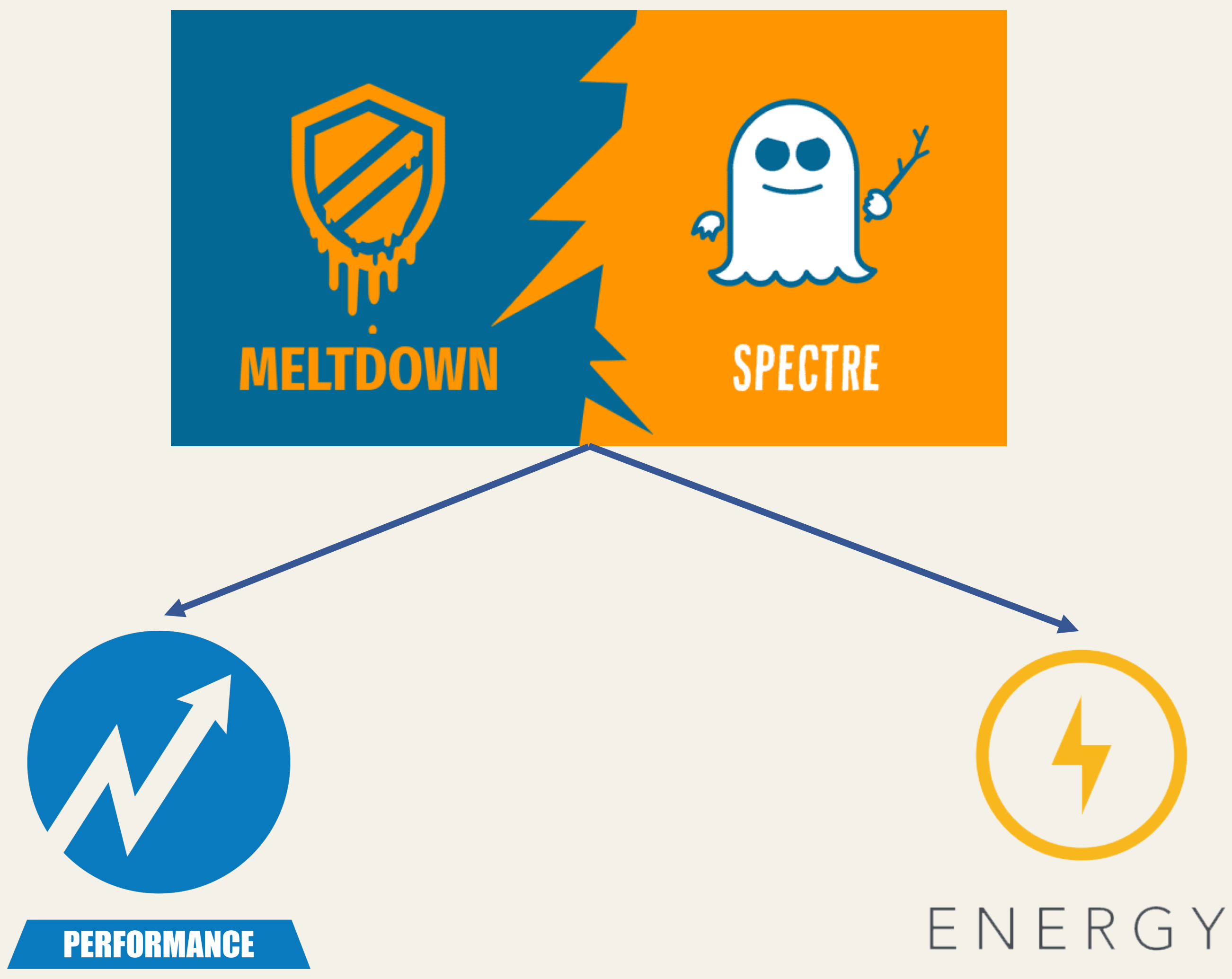
Michail Loukeris

loukerismichalis@gmail.com

Athens University of Economics and Business

## Introduction

A Study on Energy and Run-Time Performance impact of Spectre and Meltdown mitigation patches to determine whether it is worth it disabling them in a safe environment.



*"When performance goes down by 50% on some loads, people need to start asking themselves whether it was worth it."*

*Linus Torvalds*

## Research Questions

**RQ1:** *What are the energy and run-time performance implications of Meltdown and Spectre mitigation mechanisms?*

**RQ2:** *Which application type's energy and run-time performance are affected more from Meltdown and Spectre mitigation mechanisms?*

## Benchmarks

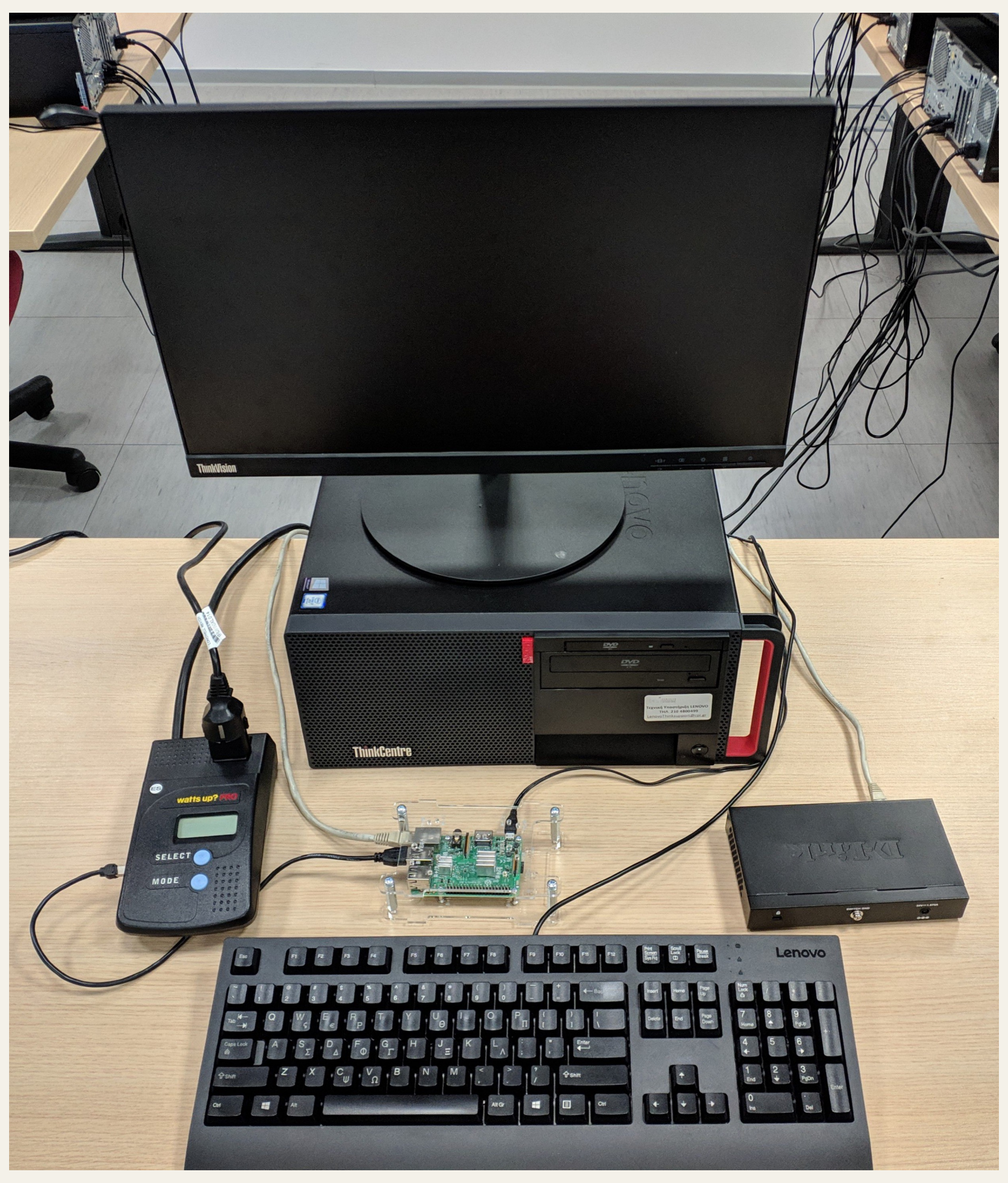
We selected benchmarks from the ***Phoronix Test Suite*** to stress a different functionality in a computer system.

**Phoronix Benchmarks**

| Benchmark Type | Operation(s) |
|---|---|
| Apache & Nginx | AB Apache Command |
| OpenSSL | aes, blowfish, camellia, cast, idea, dsa, ecdsa, ghash, hmac, whirlpool |
| OS Bench | create files, create processes, create threads, launch programs, mem_alloc |
| CacheBench | memcpy, memset, mixed, read, write |
| MC Perf | add, append, delete, get, prepend, replace, set |

## Setup - Methods

### Experimental Platform



We performed our experiments on:

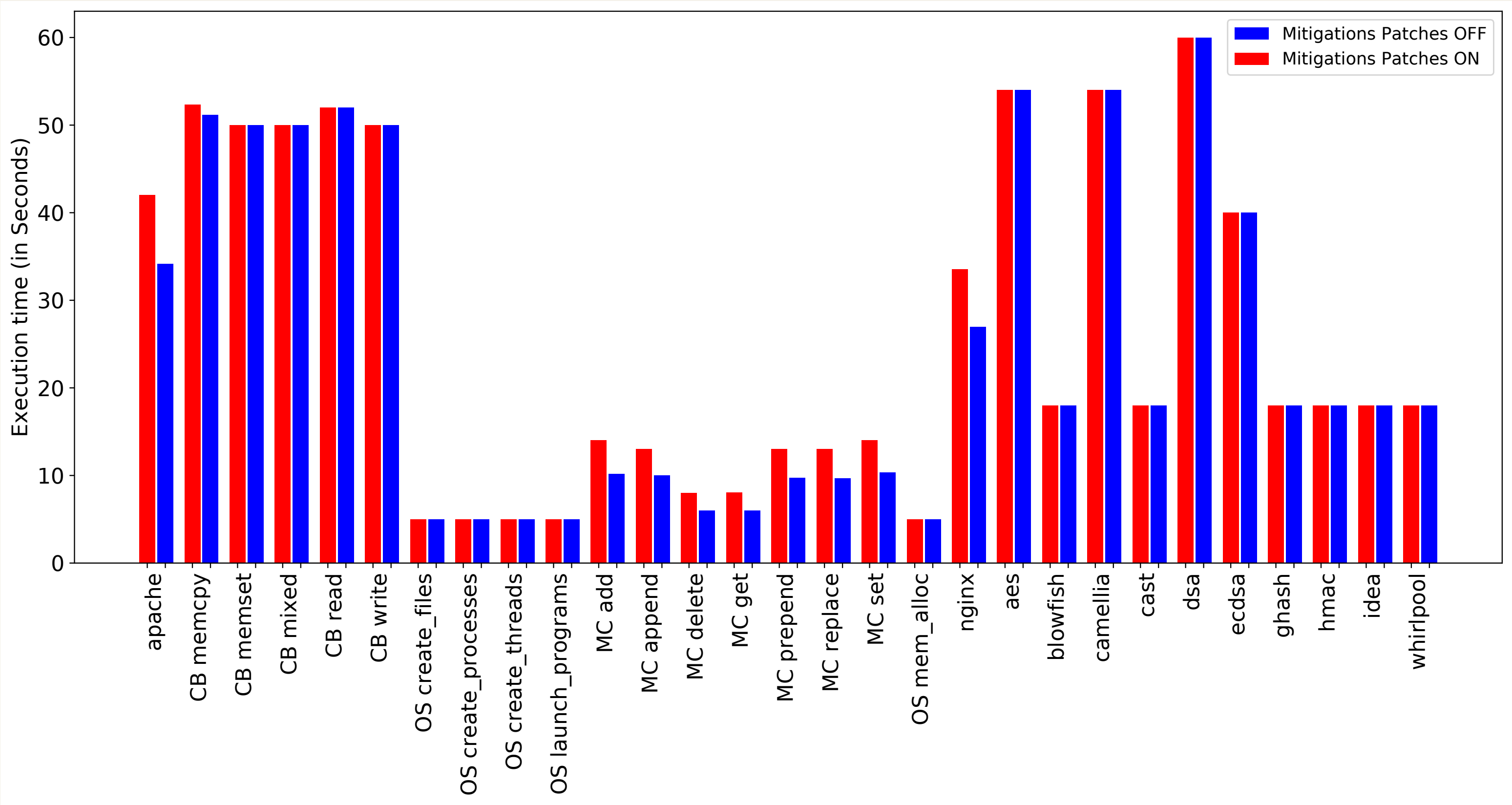
- Lenovo ThinkCentre M910t.
- Fedora 28 and Linux Kernel 5.0.9-100.
- Watts Up Pro (WUP).
- Raspberry Pi 3B.

### Running Experiments

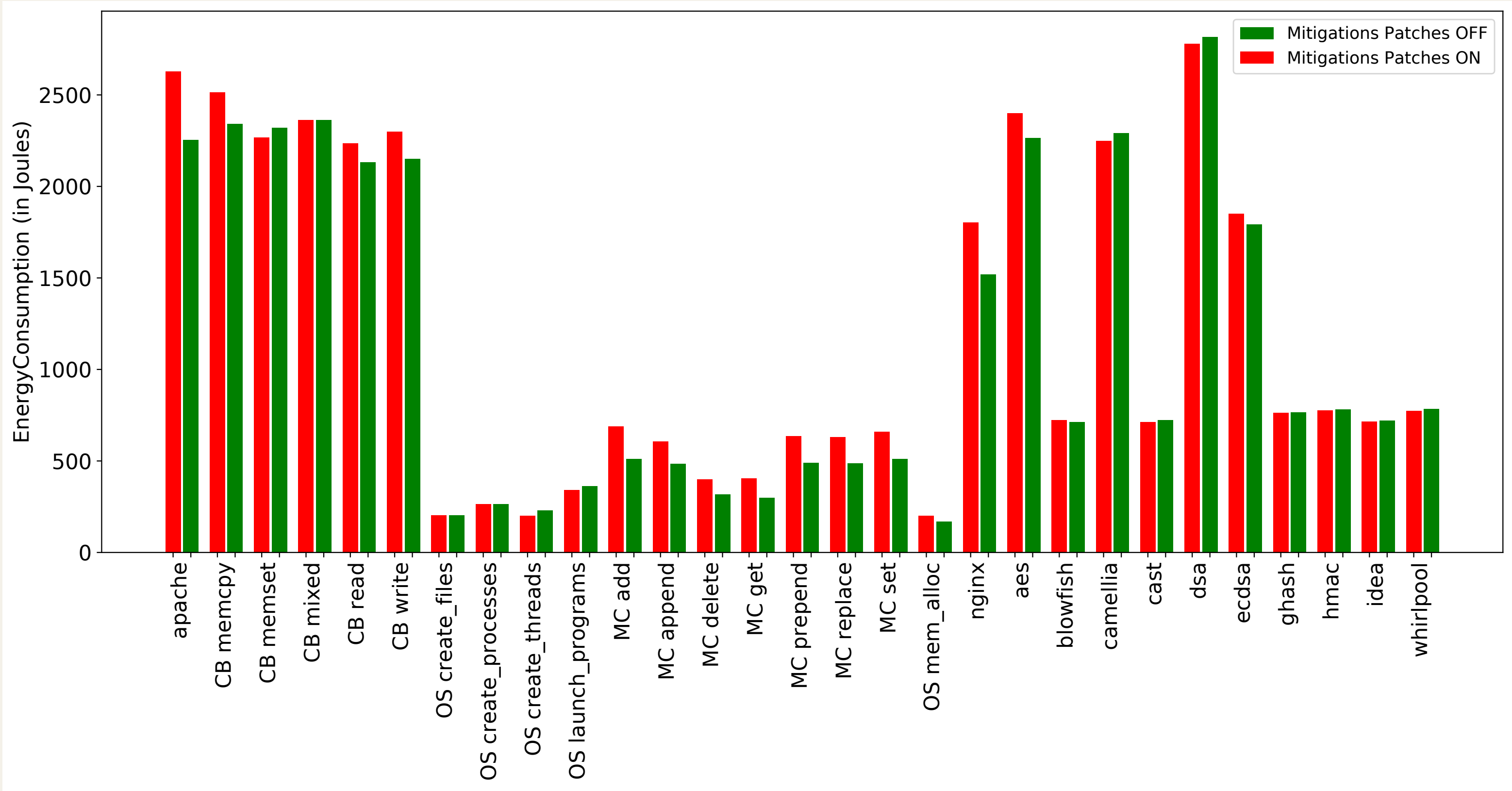
Before running our experiments we took a number of precautions to ensure the validity of our results:

- We shut down background processes and daemons.
- We let a small time window of 30" between each test to avoid power tails.
- We executed each test case 20 times to be statistically correct.

## Results



**[ Meltdown & Spectre benchmarks execution time ]**



**[ Meltdown & Spectre benchmarks energy consumption ]**

## Conclusion

- **RQ1 →** Higher energy consumption and run-time performance overhead of up to **26%** and **27%** respectively.
- **RQ2 →** *Apache* and *Nginx* were both affected by high energy consumption and run-time performance overhead similarly to memory-like operations such as *memcpy, memset, read, write, add, append, replace, set and mem_alloc*. By examining the cryptographic algorithms we experience up to **17%** of increased throughput. Finally, we observe that processes, files and thread creation were not affected by Spectre and Meltdown.

## Acknowledgements

The research described has been carried out as part of the CROSSMINER Project which has received funding from the European Union's Horizon 2020 Research and Innovation Programme under agreement No. 732223

27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE), 26 – 30 August 2019, Tallinn, Estonia