Introductory Rust (2026): Performance and Profiling

Extension Lecture

Lukáš Hozda

Winter 2026/27

MFF CUNI

Performance Basics

First Principles

- Correctness before optimization.
- Measure before changing code.
- Reproducible experiments only.

Typical Traps

- Benchmarking debug builds.
- Measuring tiny inputs with high noise.
- Interpreting one run as truth.

Simple Benchmark Harness

```rust
use std::time::Instant;

let start = Instant::now();
let sum: i64 = (0..1_000_000).sum();
println!("{} {:?}", sum, start.elapsed());
```

Algorithm Beats Micro-Optimization

- O(n log n) -> O(n) usually dominates any tiny tweak.
- Data layout and cache locality matter.

Allocation Awareness

- Reuse buffers where practical.
- Avoid hidden allocations in tight loops.

Practical Rule

- Baseline -> change -> re-measure.
- Keep short notes for each optimization attempt.