

5-2: Multithreading (Practice)

Artem Pavlov, TII, Abu Dhabi, 18.04.2025

Create new crate

- Create new branch in the repository **p52**
- Create new library crate **p52**
- Check that **p52** is listed as a member of the workspace in the root **Cargo.toml**

Sum 1

- Write function `map_sum1` in `lib.rs`
- The function accepts `Vec<u32>`, function pointer which maps `u32` to `u64` (i.e. `fn(u32) -> u64`), and const generic parameter `N`
- It should apply provided closure to each value in vector and sums return results using `N` threads
- Do not use `rayon`, channels, atomics, or scoped threads
- Each thread should return summation result of its part

Sum 2

- Write the following function which do the same works as `map_sum1` and have the same signature:
 - `map_sum2`: uses `AtomicU64` counter for summation results
 - `map_sum3`: uses MPSC channel to communicate mapping results from worker threads and does summation in the main thread
 - `map_sum4`: uses `rayon`

Sum 3

- Write `map_sum5` which accepts `&mut [u32]` instead of `Vec<u32>` and mapping function which maps `u32` to `u32`
- It should perform mapping and summation of results in `N` worker threads
- Results of mapping should be saved into the input buffer
- The function should return sum of mapped results
- Hint: use scoped threads