

Data Processing on Modern Hardware

Assignment 5 – Synchronization

Handout: **31st May 2023**
Due: **7th June 2023 before Tutorial**

Introduction

The goal of this exercise is to compare different synchronization approaches introduced in this week's lecture. We analyze pessimistic and optimistic locking, exclusive and shared locks and vary the granularity from coarse- to fine-grained locking. As workload we use a list-based set provided in the code skeleton.

Part 1 - Implementation

Your task is synchronizing the given list-based set using the following approaches:

1. Coarse-Grained Locking
2. Coarse-Grained Locking with read/write lock
3. Lock Coupling
4. Lock Coupling with read/write locks
5. Optimistic
6. **(Bonus)** Optimistic Lock Coupling

For more information:

- you can see the examples provided in the **Assignment 5 - Synchronization Methods Examples** on Moodle
- you can consult this week's lecture slides
- for the Optimistic Lock Coupling method, you can find the original paper describing the method in the **Background Reading Material** folder on Moodle.

Part 2 - Analysis and Evaluation

Apply the different synchronization techniques for the list-based set to analyze and compare them:

- How expensive is locking for the different approaches?
- How do the approaches perform in regard to #clock-cycles, #instructions and IPC?
- Which approach provides the best performance?

Vary the number of threads from low to high contention. Further, investigate the effect of the domain parameter. Provide graphical plots for your results and normalize the results to make them comparable, e.g., using parallel speed-up as metric (Lecture 7, slide 6).

Submission guidelines

This homework has a duration of one week. Fork the repository and commit your changes in the git to hand in your homework.

The programming language of this homework is C++, or C. We provide you a code skeleton, you can add functions needed for the implementation. For performance measuring of the experiments, you can either use the provided `perfEvent.hpp` and the commented code blocks in C++, or use the tools you applied in the previous assignments.