

COL380 Assignment 1

Sayam Sethi

January 2022

Contents

1	Code Analysis and Plan of Action	1
---	----------------------------------	---

1 Code Analysis and Plan of Action

The following observations were made after looking at the original code and analysing using various tools:

1. Both threading loops had issues of cache misses. This was because of the loop having the property of skip-by-numt instead of a linear loop. This leads to lesser spacial locality and more cache misses as a result.
2. `counts[v].increase(tid)` was leading to false sharing. This was due to the increase of adjacent array elements in different threads.
3. The second threading loop was not dividing the work equally among all the threads. This was because each bin won't have an equal number of elements, therefore, the work is not fairly divided across all threads.
4. Additionally, in the second loop, for each range, the code was looping over the entire D array, this led to excessive read and write misses, apart from the wasted time complexity.

`perf` was initially used to compare results with different modifications, however this was discarded in exchange of `cachegrind` tool of `valgrind` since `perf` had very high fluctuations across multiple runs of the same code. This led to inconsistent deductions and difficulty in removing noise from the results of the tool.

Running `gprof` on the original code signified that `readRanges` took the most amount of time. This analysis wasn't helpful. Additionally, running `gprof` on the optimised code gave some absurd results that could not be inferred at all. Therefore, the output of `gprof` tool was discarded. However, the functioning of the tool has been very well understood.