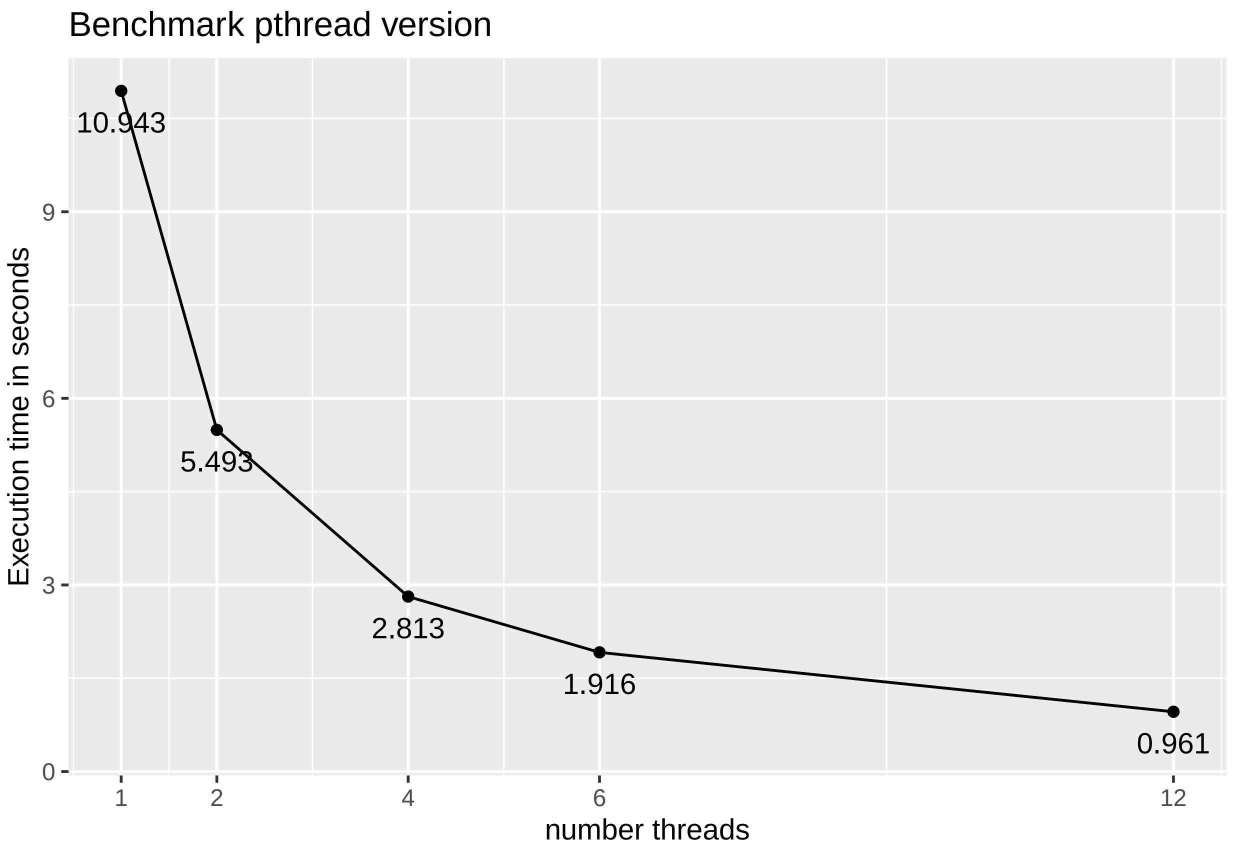
PS-Parallele Programmierung Exc 4

Marco Fröhlich, Johanna Backer and Camillo Zanolin

Exercise 1

All measurements where done on the LCC3 and the result of the mean of 9 executions.

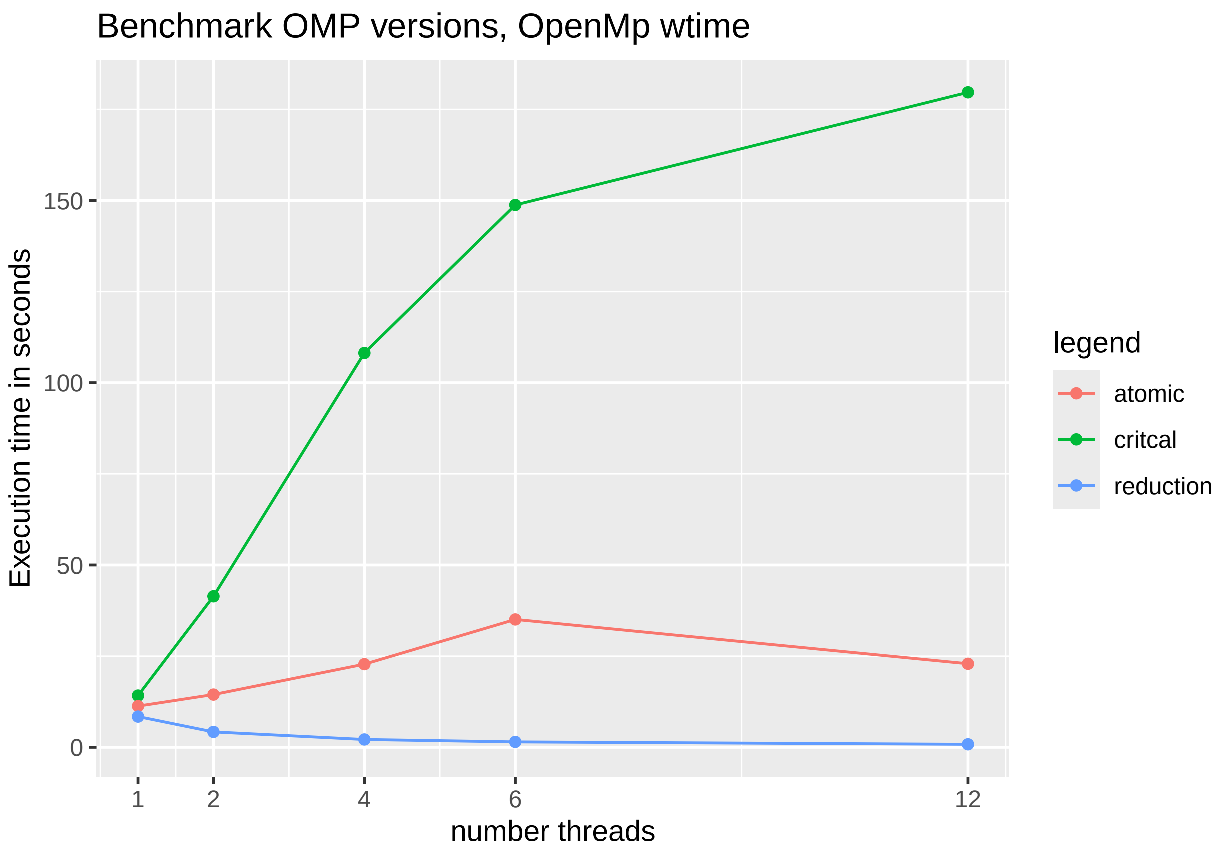
Task 1: Benchmark pthread version



Observation:

More threads = less execution time

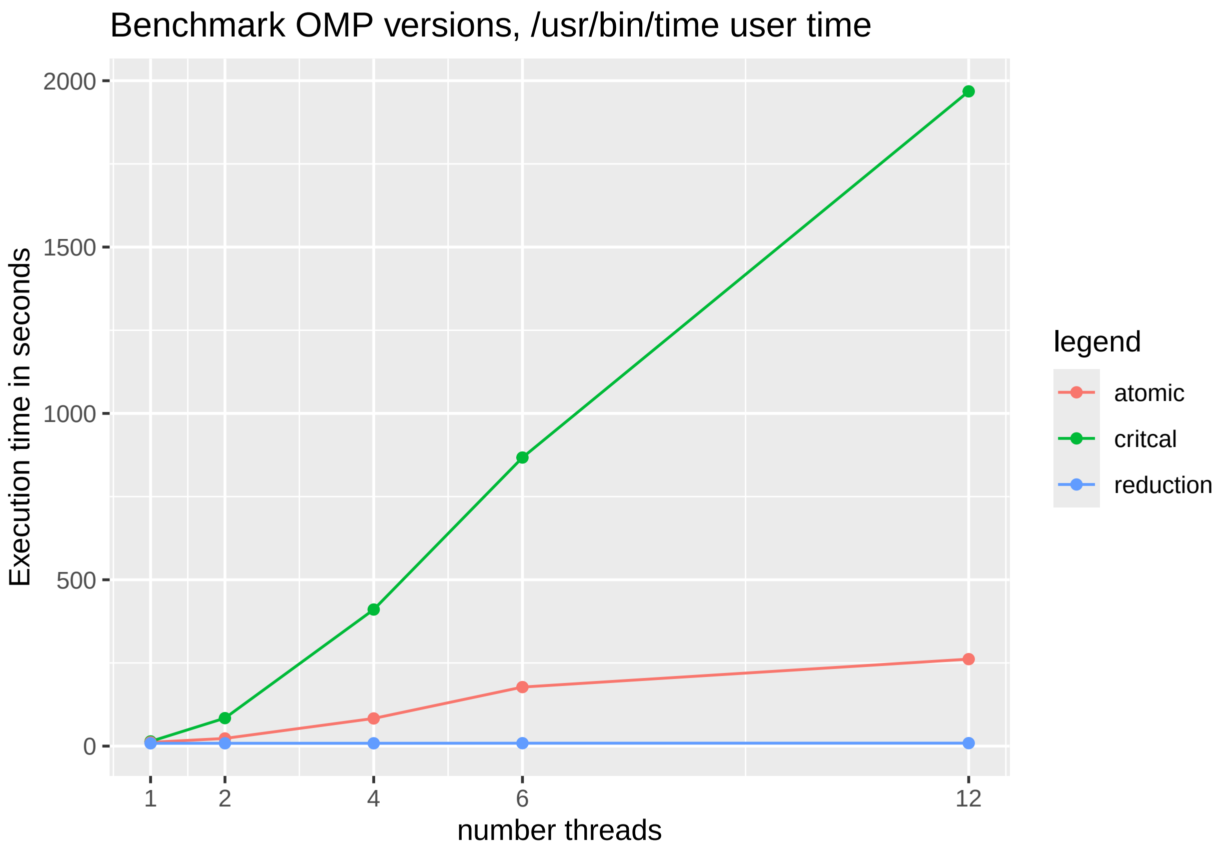
Task 4: Benchmarking different OpenMP versions with OpenMP wtime

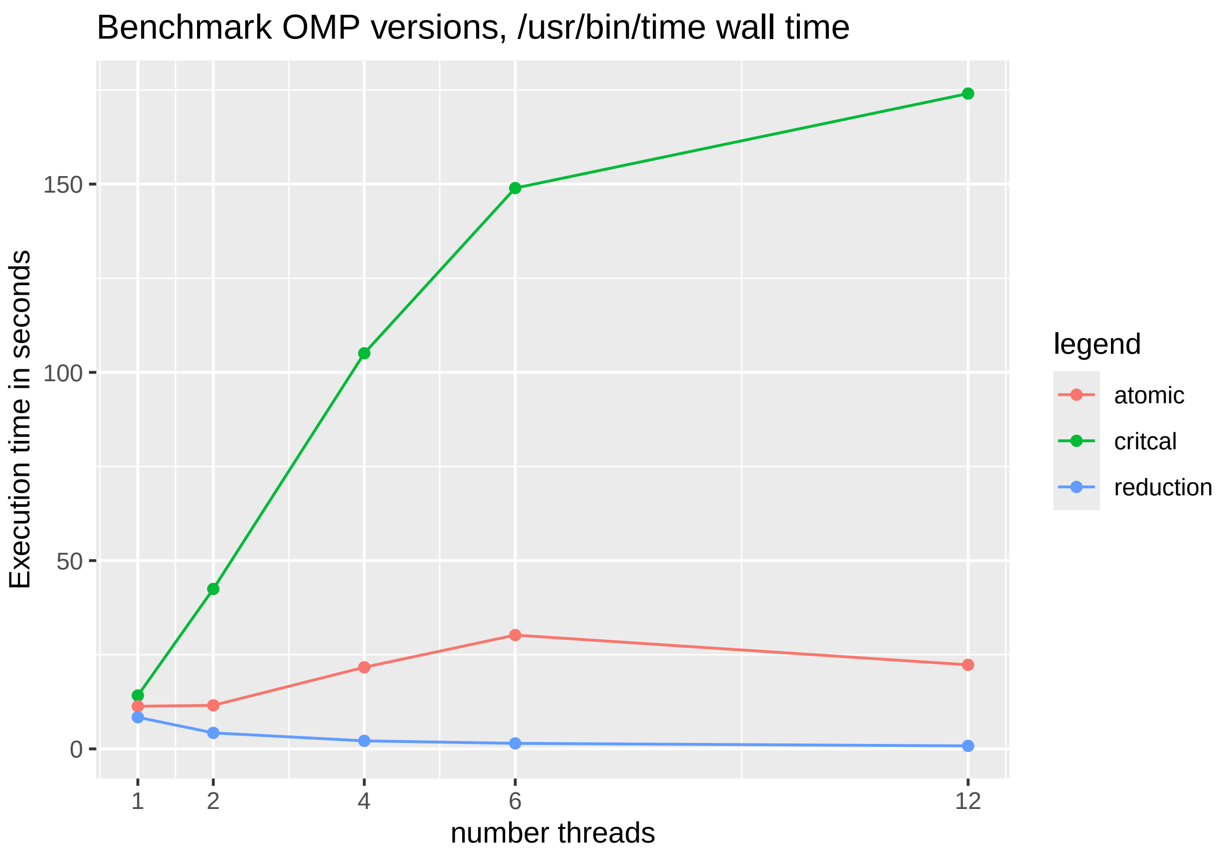


Observations:

- Critical causes very long run time. Because each in each iteration only one thread at a time can enter the critical section, therefore the thread count increases the run time. This is against the pthread results.
- Atomic version increases slightly with more threads.
- Reduction version is very fast, even faster then the pthread version.

Task 5: Benchmarking differtent OpenMP versions with /usr/bin/time





Observations:

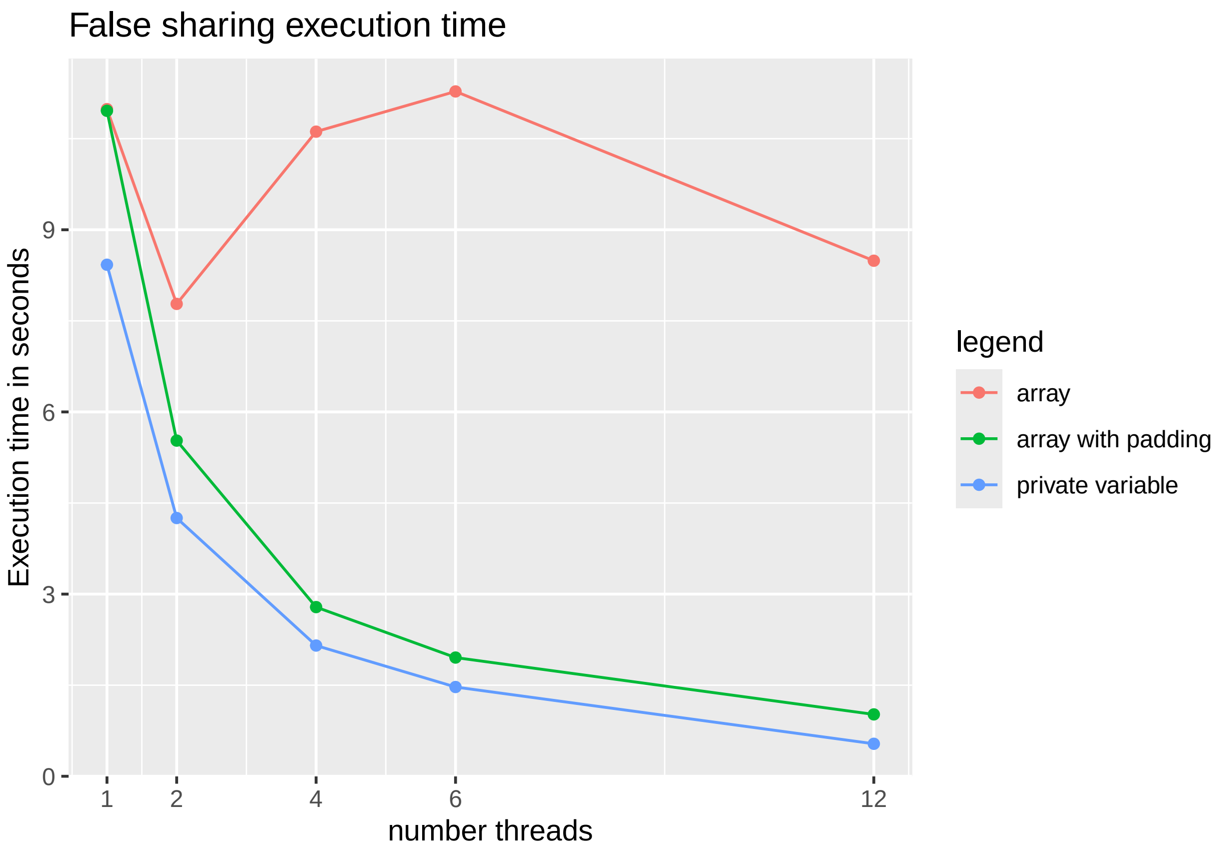
`/usr/bin/time` wall time is the same as the wtime measurements of OpenMP. Whereas the `/usr/bin/time` user time sums up the execution time of all involved threads.

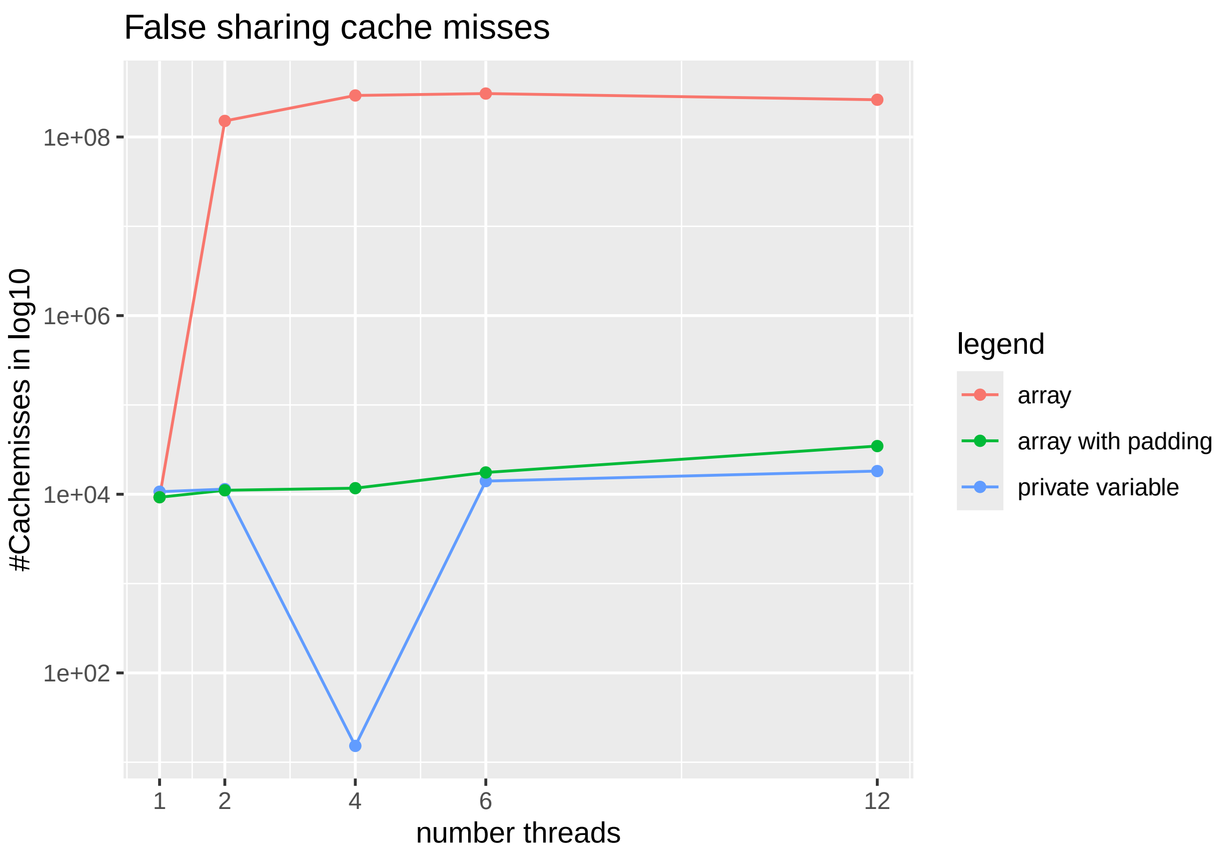
Exercise 2

Task 3

We found the ideal padding size to be 64 Bytes. We found this by executing this command: `getconf -a | grep CACHE` and look at the cache line size value.

Task 4


False sharing execution time
Execution time in seconds
0
3
6
9
1
2
4
6
12
number threads
legend
array
array with padding
private variable



Observations:

The version with an array without padding has terrible cache miss count. The other two version have very similar cache miss counts, with private version slightly better. In terms of execution times private variable is best with padding being slightly slower. Both of those improve with more threads, whereas the normal array version get slower with more threads.