Solution to the 2021-2022 exam in HPPS

January, 2022

1 Context

The reference solution code is in the directory `exam-solution`. This document contains reference answers to the questions posed in section 2 of the exam text *relative to the reference solution code*. It is possible that a student submits differing-but-correct code, and hence also provides differing-but-correct answers. However, given the quite fixed task, it is unlikely that any major divergence is going to be correct.

a)

Using `T=8`.

$n, m =$	1024	2048	4096
`transpose`	$8ms$	$31ms$	$302ms$
`transpose_blocked`	$2ms$	$12ms$	$87ms$
`transpose_parallel`	$2ms$	$16ms$	$100ms$
`transpose_blocked_parallel`	$1ms$	$6ms$	$50ms$

b)

c)

d)

e)

The `matmul` function has somewhat horrible spatial locality, as we access array `B` with a stride of `k`. This means we likely have many cache misses.

In contrast, `matmul_locality` accesses all arrays with unit stride, ensuring perfect spatial locality. Finally, `matmul_transpose` also accesses all arrays with unit stride, made possible by first transposing `B`. Since transposition is asymptotically (and in practice) much faster than matrix multiplication, this preprocessing does not add much to the runtime, but allows a more efficient subsequent memory access pattern.

f)

g)

For `matmul_parallel` and `matmul_transpose_parallel`, I decided to parallelise only the outer two loops with `omp pragma parallel for collapse(2)`. This is because the inner loop has a dependency on the accumulator, and while it can be parallelised using a `reduce` clause, it is likely not worth the overhead—the two outer loops provide sufficient iterations for most practical workloads and machines.

I use OpenMP's default static scheduling, because the different iterations should are naturally load-balanced.

h)

`matmul_locality_parallel()` only parallelises the outer (`i`) loop. This is because different iterations of the `j` loop write to the same `i` row of the output matrix, and hence the iterations of the `j` loop are not independent. Hence `matmul_locality_parallel()` is less parallel than `matmul_transpose_parallel`.

For most workloads, the number of iterations in the outer loop (`n`) is going to exceed the number of cores in the machine, and so this difference will not matter.

i)

j)