Computational Science on Many-Core Architectures
Exercise 1

Leon Schwarzäugl

October 2023

1 About Yourself

I am now in my third semester of the CSE masters program; I initially took this course in the first semester as well, but had to drop it after the first two weeks because of time constraints.

2 Expectations

I look forward to learning to program in multi-core environments and improving in code optimization. It seems like an interesting topic to me where I would maybe also consider to look for a master thesis later on.

3 2D-Arrays in C

Disclaimer: The answers for the next two topics are the same that I gave when I tried the lecture last year. With both of these I am still quite confident, so there is no real change compared to last year. I hope this is ok.

A 1D-array can be allocated by using a pointer to a memory block, in our case this block will have size $NM \cdot \text{sizeof}(\_\_\text{type}\_\_)$.

```
    __type__* __name__ = malloc((N * M) * sizeof(__type__);
```

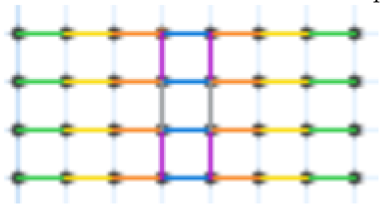
We can allocate a 2D matrix using an array of pointers of size N for which we can iteratively allocate memory:

```
    __type__* A[N];
    for(int i = 0; i < N; i++)
        A[i] = (__type__*)malloc (M * sizeof(__type__));
```

4 Summing Random Numbers in a Lecture Hall

4.1

The information should travel from the outer regions of the classroom towards the middle as shown in this simplified graph:



This way, we get 6 turns, at which point 4 students can know the sum. This must be the minimal time, since the information has to, at the least, travel to the center of the shape from each corner. Since the center in this case consists of 4 students, it needs to travel to the furthest of them.

4.2

For this, we retrace the same way back from the center, except for the last steps (because these students already know the sum). Doing this gives us 10 turns.

4.3

We define the number of columns as $C = \text{ceil}(\frac{S}{R})$, with S the number of students and R the number of rows. Then we get for the number of rounds until one student knows the sum:

$$\text{ceil}\left(\frac{R}{2}\right) + \text{ceil}\left(\frac{C}{2}\right), \quad R, C > 1$$

Of course, for a single student with one row and one column we receive 0 turns and for two students (who are arranged either in one column, two rows or vice versa) we receive 1 round.

The number of turns until all the students know the sum is for a number

$$2\left(\text{ceil}\left(\frac{R}{2}\right) + \text{ceil}\left(\frac{C}{2}\right)\right) - \left(2 - \text{ceil}\left(\frac{R \bmod 2 + C \bmod 2}{2}\right)\right), \quad R, C > 1$$

The first term accounts for the number of turns it takes to propagate all the information to one student and then back to all the other students. This however is suboptimal, since always at least two students must acquire the sum at the same time (since the sum is acquired during an exchange of information), with a possibility of four people acquiring the sum in the same time given the classroom has the right dimension. The second term is thus a correction which depends on the relation of rows and columns: if both are an even number, 4 students will acquire the sum first in the same turn, which saves two turns during the backpropagation, with the last ceil becoming 0. In all other cases (R even, C odd; C even, R odd; C odd, R odd), two students will acquire the sum first in the same turn, which will save one turn during the backpropagation, with the last ceil deducting 1.

4.4 Bonus task

Now every student can communicate with any other student. We can thus ignore the rows and columns and only focus on the number of students. For the time it takes one student and all students to know the sum in the case of 32 students, both take the same amount of turns, which is 5 turns - so all the students have the sum at the same time (reasoning further below).
The number it takes one student to have the sum is, for a general number of students S, given by:

$$\mathrm{ceil}(\log_2(S))$$

This number is explained due to the effect that as long as the number of students is given by potencies of two, every student can communicate with a new "cluster" each turn, doubling the information the student has. At the same time, the number of "clusters" (groups of students that have the same information) halves each turn. If the number of students is not given by a potency of two, we need one extra turn to exchange the information with "their" cluster (they always have at least one student who has all of their information since they must be less students than the number of students in the "main group").
For a general number of students, the time it takes for all the students to have the sum is for a number of students equal to a potency of two:

$$\log_2(S)$$

and else, when the number of students does not equal a potency of two:

$$\mathrm{floor}(\log_2(S)) + \mathrm{ceil}(\log_2(S)) - \mathrm{floor}(\log_2(S - 2^{\mathrm{floor}(\log_2(S))}))$$

The first term describes the number of turns needed in the "optimal" case of a number of students which is equal to a potency of two. The second term adds another turn if the number of students is not equal to a potency of two - this term describes the time it takes to pass the information to one "lone" student and following backpropagation. The last term is another correction that reduces the number of turns needed depending on the biggest potency of two found within the "lone" students.
Interestingly, the worst case is thus given by a number of students that is just by one higher than the previous potency of two. For example, 5 students need 5 turns for all of them to know the sum, while 7 students need only 4 turns.