Exercise 01

1 About Myself

My name is Simon Hinterseer. I am currently studying CSE at TU Wien. The only experience with parallel computing, that I have, is from this master's course. This means, we did some multi-threading (with OpenMP) and some multi-processing (with MPI) on x86 machines. In that sense, I have no experience with computational science on many core architectures.

2 Expectations

I hope to learn how to do calculations on a GPU. In this regard, I am especially curious about experiencing SIMD. So far, I only learned, what it means in theory. It would also be an interesting side-track to see, how one can make use of SIMD on an x86 architecture. I think, I heard someone say, that it is used when benefiting from vectorization, but I never really used that concept, in my code.

3 Summing random numbers in a lecture hall

To simplify the problem, I would like to re-formulate it slightly. Each student has one memory position, that can hold an integer. Each student can each turn do one of three things:

1. exchange numbers with a neighbour and add the received number to the number in their memory.

2. exchange numbers with a neighbour and overwrite the memory with the received number. The neighbour does not change their memorized number.

3. be idle

This document will refer to these operations later.

3.1 One student knows the sum

It takes at least 6 turns to make one student hold the sum of all initial random numbers in their memory. It is not possible, to make that happen in 5 or fewer turns, since there is no student *A*, such that every student has a distance[1] of 5 or less to student *A*. This can be checked by examining the next figure.

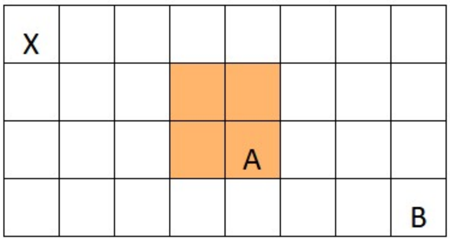

*Figure 1 - field of students – students in the orange part have a maximum distance to other students of 6, which is the distance between student **X** and student **A**. Students outside the orange part have a larger maximum distance. The distance between student **X** and student **B** is 10.*

Find a strategy that makes the four students in the orange field hold the sum of random integers in their memory **in 6 turns**:

1. make each of the four students know the sum of their quadrant in 4 turns:

Figure 2 shows the top left quadrant of the solution. The problem is symmetrical, so this solution can be applied to the other quadrants in a mirrored fashion.

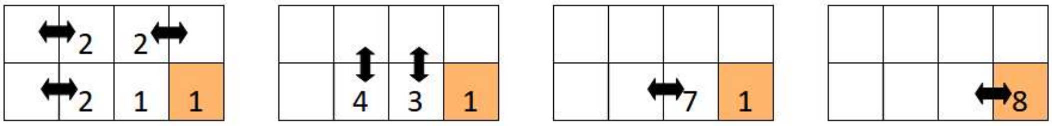

Figure 2 – This strategy ends up in the orange student holding the sum of random integers after 4 turns. The numbers indicate the number of summands in the respective students' memory at the end of each turn. The double arrows indicate operation 1: exchange and store the sum.

2. make the four central (orange) students know the sum of the whole field in 2 additional turns

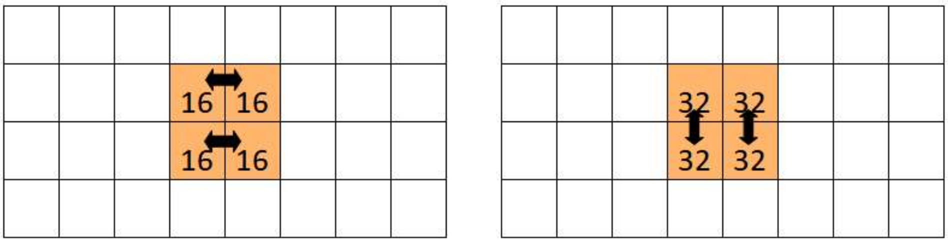

Figure 3 – The final two turns of the first problem. After this, the four central students know the sum of the whole field.

[1] The distance is the minimum number of turns, it takes for these two students to exchange any information. Therefore, the distance is the number of moves from the one student's position to the other's.

3.2 All students know the sum

Continuing from the final state of the last problem, it takes 4 additional turns to make all students know the sum. The problem is again symmetrical, since each of the four central students just “pushes” their value into their respective quadrant:

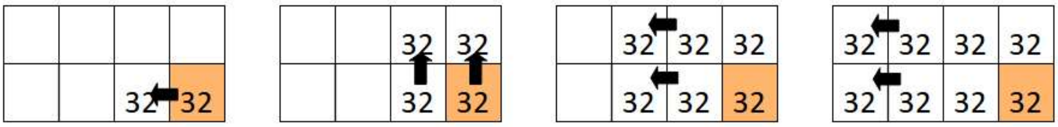

Figure 4 – The 4 additional turns, that make all the students know the sum of the whole field. The single arrow stands for operation 2: exchange and overwrite value for the student, that is at the tip of the arrow.

3.3 Bonus Point - Arbitrary communication

If arbitrary communication is possible, it takes five turns for every student to know the sum. It is not possible in fewer turns, because a student can only learn numbers from one other student in each turn. Therefore, the maximum number of summands in the stored sum is at most the sum of the number of summands for the two students. So, in each turn the students can hold a sum composed of this number of summands at most:

Table 1: maximum number of summands in each student’s memory per turn

	maximum number of summands in the stored number	
	before the turn	after the turn
turn 1	1	2
turn 2	2	4
turn 3	4	8
turn 4	8	16
turn 5	16	32

This strategy makes every student hold the sum of all random integers in their memory after five turns.

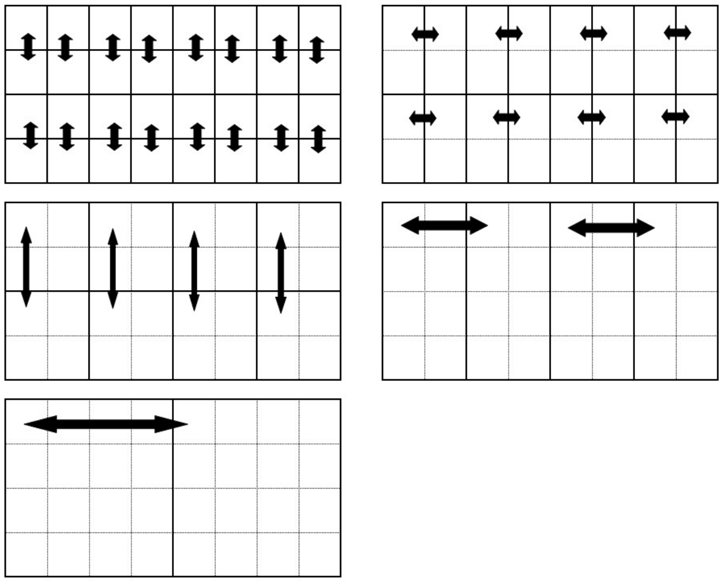

Figure 5: strategy for arbitrary communication. The double arrow indicates operation 1 (exchange and store sum). The thick lines indicate groups of students, that all hold the sum of the original random numbers of the whole group in their memory. In each turn 16 pairs of students communicate but for purposes of overview, only one communication per student group is displayed.