Scientific Computing - Exercise Sheet 1

Jonathan Hellwig, Jule Schütt, Mika Tode, Giuliano Taccogna

19.04.2021

1 Exercise

Notation: In the following we use **for** as a short hand to simplify the notation of repeated instructions. Since **for** is not included in the instruction set each **for** is to be interpreted as a simpel repetition of all instructions inside the loop. We use $\leftarrow$ to denote saving values into registry that are reused in subsequent computations.

(a) **Input**: $a, b, k \in \mathbb{R}$

1: **load**(k)
2: **load**(a)
3: $l_1 \leftarrow$ **mult**(a, k)
4: **load**(b)
5: $l_2 \leftarrow$ **add**(l_1, b)
6: **store**(l_2)

(b) **Input**: $a, b \in \mathbb{R}^d$, $k \in \mathbb{R}$

1: **load**(k)
2: **for** $i \leftarrow 1 : d$ **do**
3: **load**(a_i)
4: $l_{2i-1} \leftarrow$ **mult**(k, a_i)
5: **load**(b_i)
6: $l_{2i} \leftarrow$ **add**(l_{2i-1}, b_i)
7: **store**(l_{2i})
8: **end for**

Number of instructions: $5d + 1$
In particular, we have 51 instructions for $d = 10$.

(c) **Input**: $a, b \in \mathbb{R}^d$, $k \in \mathbb{R}$

1: **load**(k)
2: **load**(a_1)
3: $l_1 \leftarrow$ **mult**(k, a_1); **load**(b_1)
4: $l_2 \leftarrow$ **add**(l_1, b_1); **load**(a_2)

5: **for** $i \leftarrow 2 : (d-1)$ **do**
6: $\quad l_{2i-1} \leftarrow \textbf{mult}(k, a_i)$; **store**$(l_{2i-2})$; **load**$(b_i)$
7: $\quad l_{2i} \leftarrow \textbf{add}(l_{2i-1})$; **load**$(a_{i+1})$
8: **end for**
9: $l_{2d-1} \leftarrow \textbf{mult}(k, a_d)$; **store**$(l_{2d-2})$; **load**$(b_d)$
10: $l_{2i} \leftarrow \textbf{add}(l_{2d-1})$
11: **store**(l_{2d})

Number of instructions: $2d + 3$
In particular, we have 23 instructions for $d = 10$.

Optimality of the algorithm:
In total there are $1 + d + d = 2d + 1$ to load into registry. Only when all values are in registry all necessary calculations can be made. Additionally, one cycle to perform the last calculation and one cycle to store the result into main memory are required. therefore, an algorithm using the proposed instruction set contains at least $(2d+1)+1+1 = 2d+3$ instructions. Thus, the proposed algorithm has minimal number of instructions for the given architecture.

Serial vs pipelined Version:
Both algorithms - pipelined and serial - have a computation time that depends linearly on d. However, the serial algorithm has a factor of 5 while the parallel one has a factor of 2. Therefore, for large d a significant performance increase can be achieved by the parallel algorithm.

2 Exercise

(a) Sketch of the 4×4 ring:

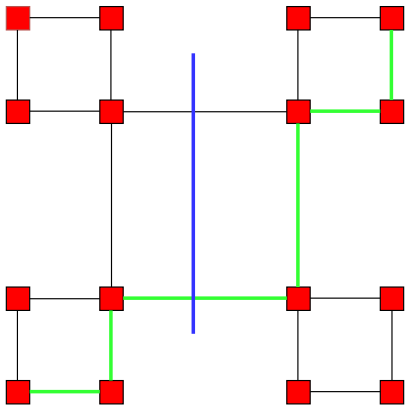

Figure 1: Ring network

(b) The network has got $4 \cdot 4 = 16$ processors. (Red nods.)

(c) The bisection bandwidth is 2. (Blue cut.) There is no opportunity to split the network while cutting just one connection.

(d) The longest communication part is 6. It is the number of hops from one 'outside' nod to the diagonal 'outside' nod. (Green path.)