Homework 1:

Charlie Wang

1:

a: my\_first\_i = my\_rank \* (n/p), my\_last\_i = (my\_rank+1) \* (n/p)

b: my\_first\_i = my\_rank \*floor(n/p), my\_last\_i = [n if my\_rank == p-1 else (my\_rank+1)\*(n/p)]

2: Assumes there are N cores

a: receives = N-1, additions = N-1

b: receives = LogN additions = LogN

c:

|  |  |  |
| --- | --- | --- |
| Cores | Non-Tree | Tree |
| 2 | R = 1 A = 1 | R = 1 A = 1 |
| 4 | R = 3 A = 3 | R = Log3 A = Log3 |
| 8 | R = 7 A = 7 | R = 3 A = 3 |
| 16 | R = 15 A = 15 | R = 4 A = 4 |
| 1024 | R = 1023 A = 1023 | R = 10 A = 10 |

d: Receive is more expensive because communication is more expensive than computation

3: There can be too much overhead, synchronization, unbalanced load on cores that making the end result not worth it.Also, only when a program is mostly parallelized, does adding more processors help more than parallelizing the remaining.

rest"

4:

Task – Parallelism:

Needs more work by the developer because each core is executing many different functions across the same or different datasets.

(SIMD) Data – Parallelism:

Hardware is enough because each core is executing same function across the elements of a dataset

5: In a distributed system, the memory is associated with individual processors, and a processor is only able to address its own memory. Each processor can utilize its memory fully without other processors interfering. Second of all, because of the lack of a common bus, theoretically there is no limit to the amount of processors. However, because of the structure, communication between different processors become more complicated.