Fundaments of HPC
Second Assignment

Nicola Domenis

December 6, 2019

1 Introduction

We present the second assignment in the course of FHPC. We will discuss about:

Exercise Zero
Objective 1 text

Exercise One
Objective 2 text

2 Exercise 0

We start by observing two slightly different codes: 01_array_sum.c and 04_touch_by_all.c . They implement the same algorithm: summing the first N natural integers. They both use the OpenMP standard to take on a parallel approach with multiple threads. The analysis of the programs can start by a strong scalability test.

2.1 (

Strong Scalability Test) We report the elapsed times versus the number of cores and also the speedups versus the number of cores:

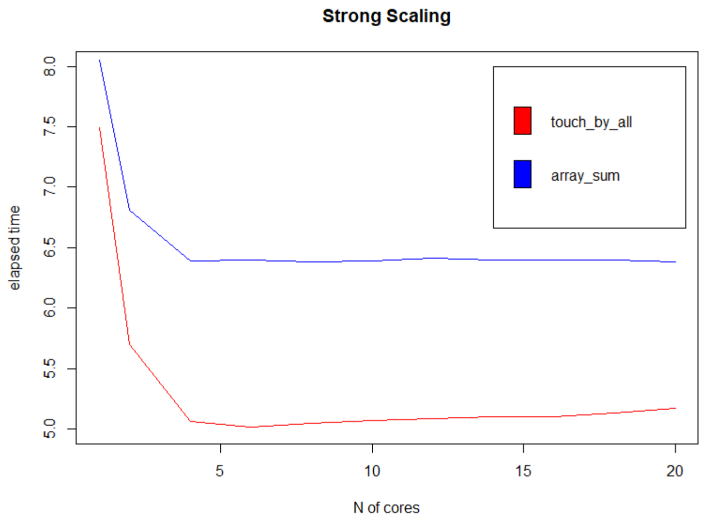

Figure 1: Elapsed times for $N = 10^9$

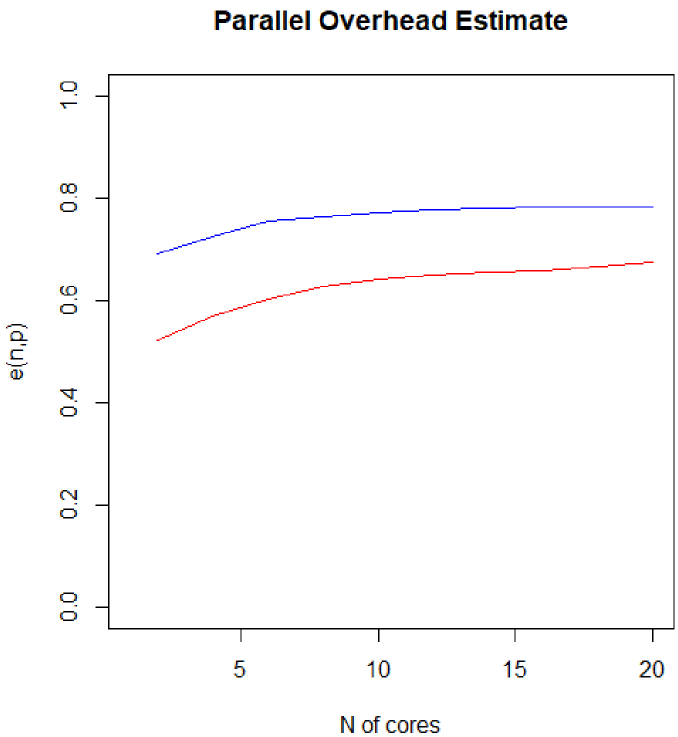

Figure 2: Speedup for $N = 10^9$

We see that the code scales for the same size of the problem and that the code that implements the "'touch by all"' policy is faster.

2.2 (

Parallel Overhead) Then we can see how to calculate the parallel overhead. We can use the formula $e(n,p) = \frac{\frac{1}{S_p(n,t)} - \frac{1}{p}}{1 - \frac{1}{p}}$ to estimate the parallel overhead. A plot with the measures we have would return:

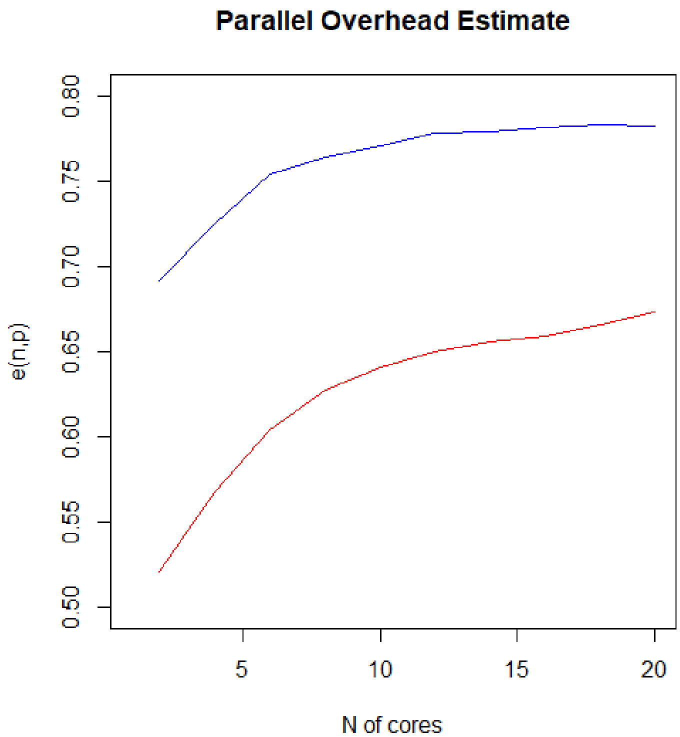

Figure 3: Overhead for $N = 10^9$

The touch_by_all code once again is better than the array_sum code, since it has a lower parallel overhead. But both lines are almost constant, showing that both codes have an almost constant parallel overhead while we increment the number of processors.

2.3 (

Performance evaluation) Now lets compare some valuable metrics for the two codes' executions:

Metric	measures					mean	
CPU cycles	array_sum	372889698	143797361	671828181	373809554	390581199	-
	touch_by_all	602487792	16412868	15030736	15901403	162458200	= 228122999
Cache misses	array_sum	9459	11492	49514	9484	19987.25	-
	touch_by_all	10356	10256	8417	12081	10277.5	= 9709.75
Elapsed time	array_sum	0,014486573	0,007287606	0,024476034	0,014344770	0.01514875	-
	touch_by_all	0,021872540	0,002893806	0,002676134	0,002677902	0.007530096	= 0.00761865

Even if the single performances, measured with Perf, are quite similar in their randomness, we see that the touch_by_all code outperforms in average the array_sum code in all the metrics.

3 Exercise 1

We need to rewrite the code mpi_pi.c by means of OpenMP directives. We had to solve all the problems we encountered in lesson, like avoiding race conditions and false sharing. In the end we produced the simple code openmp_pi.c.

3.1 (

Strong and Weak Scalability) We show the plots for the strong and weak scalability:

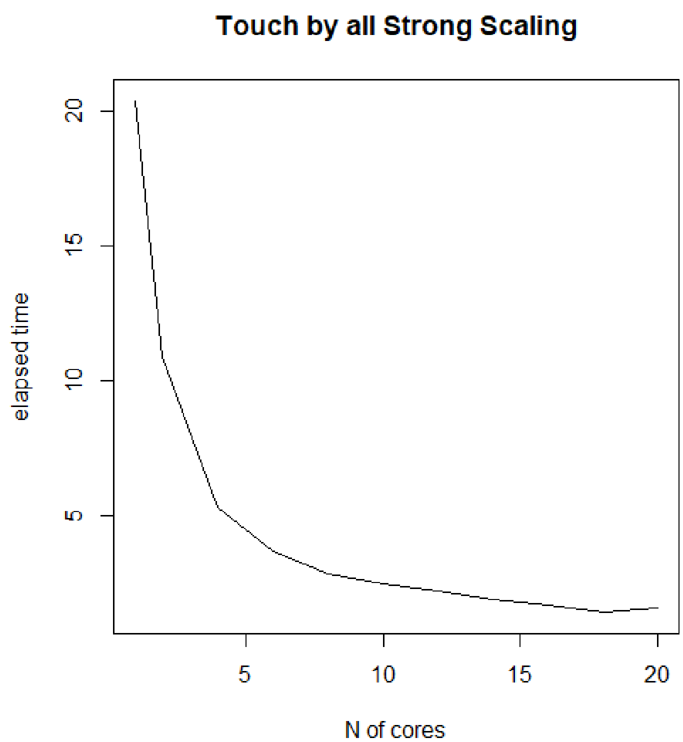

Figure 4: Strong scaling test for $N = 10^9$

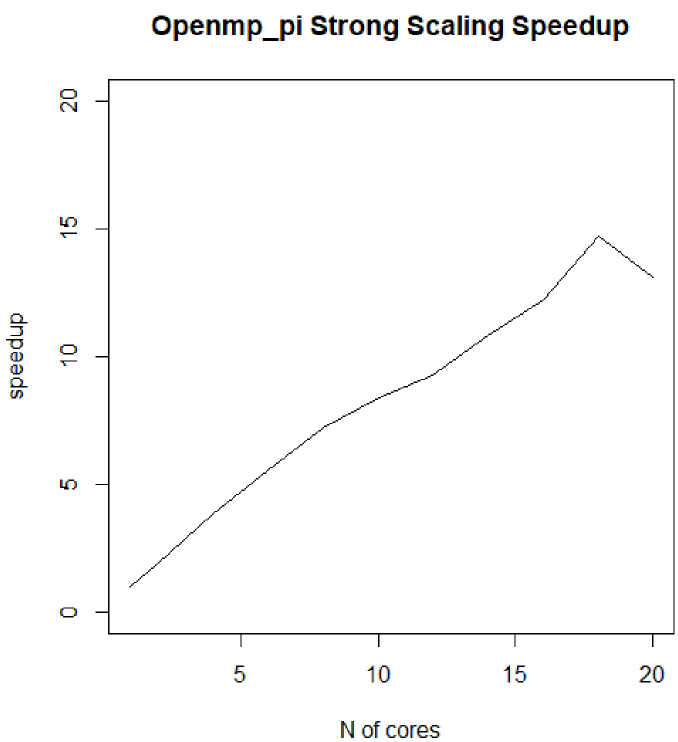

Figure 5: Speedup for $N = 10^9$

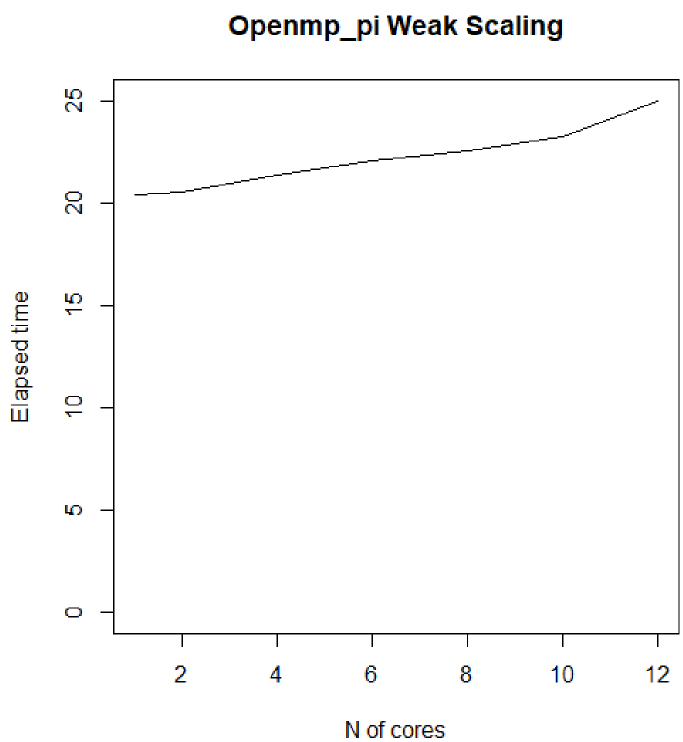

Figure 6: Weak Scaling test for $N = 10^4 * P$

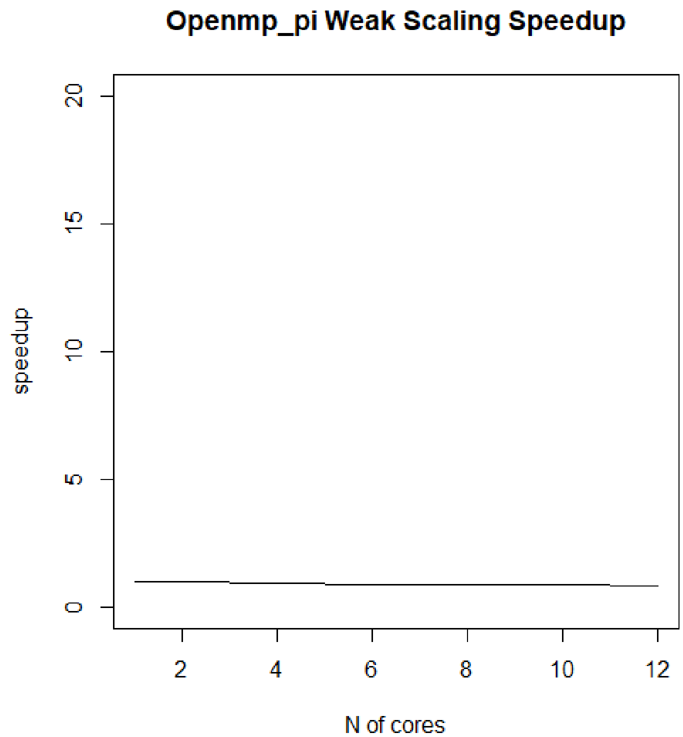

Figure 7: Weak Scaling speedup for $N = 10^4 * P$

In the strong scalability setting the code scales very well, the speedup is almost linear. But the weak scalability test shows that the program doesn't handle well problems of increasing size through an increase of threads.

3.2 (

Parallel overhead)
Now we estimate the parallel overhead, using again the formula to calulate $e(n,p)$ when we have an increasing number of threads on multiple cores:

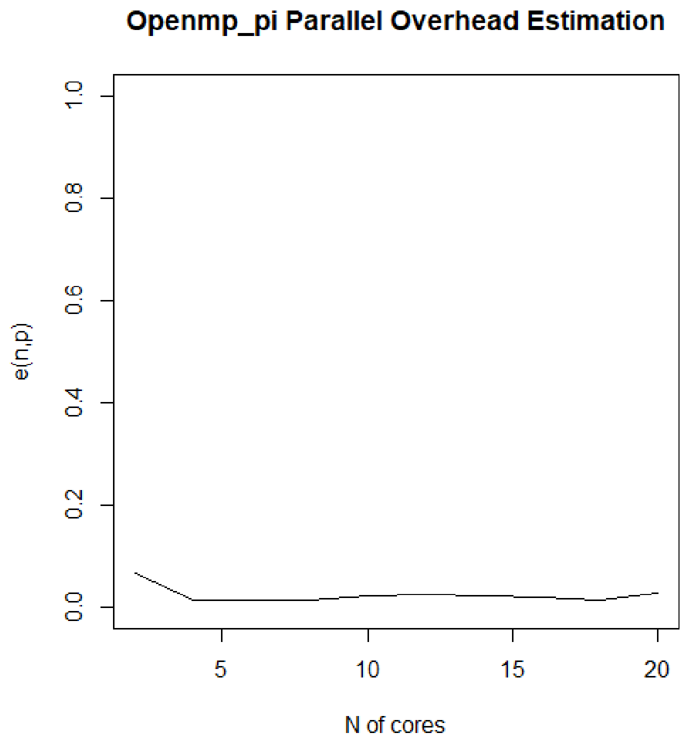

Figure 8: Overhead for $N = 10^4 * P$

We see that the overhead estimation remains almost constant, showing that the program's parallel overhead doesn't increase over the number of processors.

3.3 (

Comparison with mpi_pi.c) Now it's time to test the performance of mpi_pi.c on a single node and on multiple nodes. We started with 1 node with 20 cores and we finished with 20 cores having 1 node each. We see if splitting processors on multiple nodes gives a slower execution time

4 Results and Conclusions

The atomic weight of magnesium is concluded to be , as determined by the stoichiometry of its chemical combination with oxygen. This result is in agreement with the accepted value.

Figure 9: Figure caption.

5

The most obvious source of experimental uncertainty is the limited precision of the balance. Other potential sources of experimental uncertainty are: the reaction might not be complete; if not enough time was allowed for total oxidation, less than complete oxidation of the magnesium might have, in part, reacted with nitrogen in the air (incorrect reaction); the magnesium oxide might have absorbed water from the air, and thus weigh “too much.” Because the result obtained is close to the accepted value it is possible that some of these experimental uncertainties have fortuitously cancelled one another.

6 Answers to Definitions

1. The *atomic weight of an element* is the relative weight of one of its atoms compared to C-12 with a weight of 12.0000000..., hydrogen with a weight of 1.008, to oxygen with a weight of 16.00. Atomic weight is also the average weight of all the atoms of that element as they occur in nature.

2. The *units of atomic weight* are two-fold, with an identical numerical value. They are g/mole of atoms (or just g/mol) or amu/atom.

3. *Percentage discrepancy* between an accepted (literature) value and an experimental value is

$$\frac{\text{experimental result} - \text{accepted result}}{\text{accepted result}}$$