

2-norm computation

1 2-norm computation

The 2-norm of a vector x of size n is simply defined as

$$\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

A sequential code for computing this value is straightforward and basically consists of a single loop over all the coefficients of the array that stores the vector.

Note that computing and summing the squares may overflow the maximum value imposed by the chosen arithmetic even if the final value of the norm does not. Therefore in practice more sophisticated methods have to be used. We will not cover this case.

2 Package content

In the `norm2` directory you will find the following files:


- `main.c`: this file contains the main program that creates a vector x of size n and then computes its 2-norm using first the sequential routine `dnorm2_seq` and then the two parallel routines `dnorm2_par_red` and `dnorm2_par_nored` that have to be implemented as described below. **Only this file must be modified.**


The code can be compiled with the `make` command: just type `make` inside the `norm2` directory; this will generate a `main` program that can be run like this:

```
$ ./main n
```

where n is the size of the vector whose norm has to be computed.

3 Assignment

-  At the beginning, the `dnorm2_par_red` and `dnorm2_par_nored` are a copy of `dnorm2_seq`. Modify these two routines in order to parallelize them in the following way

- `dnorm2_par_red`: parallelize this routine making use of the `reduction` clause of the OpenMP constructs
- `dnorm2_par_nored`: parallelize this routine **without** making use of the `reduction` clause of the OpenMP constructs, that is to say, doing the reduction operation by hand.
-  Analyze and compare the performance of the parallel code using one, two and four threads. Did you observe any speedup (reduction of the execution time) using 2 and 4 threads instead of 1? How can you interpret these results? Report your answer in the form of comments at the bottom of the `main.c` file.

Advice.

- in the `dnorm2_par_nored` parallel routine the reduction operation has to be computed by hand without relying on the OpenMP `reduction` clause. This can be achieved through the use of a shared array.