Homework

Shallow water eq_2D

ME 471/571, Spring 2019 **submission due: April 12th**

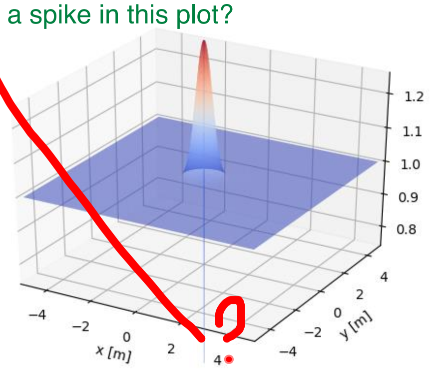
Homework done by: Arash Modaresi Rad

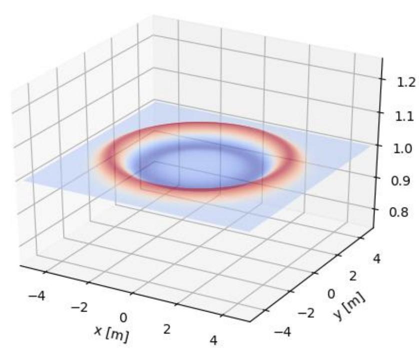
Task 1 - parallelize the computations using overlapping communication with computation

15 pts

For the serial process the original sw-2d.c algorithm was used and then the code was modified for parallel processing.

There seem to be a spike in this plot?





Is this result made using parallel code, or serial code? Either way, a comparison of serial and parallel would be nice, as it would prove that you get the same result both in serial and parallel.

The left most figure is the initial wave at t=0.5s and the most right figure is the final wave at t=1s.

t = 0?

Task 2 - Perform scalability study

10 pts

For strong scaling a fix problem size with Nx = 4000 and Dt = 0.0002 was considered and the results are in table 1, as it can be seen from Figure1 only 4 cores have a parallel efficiency above 60%. According to the results, 4 processors would be the optimum number for this code and strong scaling suggests that for this particular problem size more using more than 4 cores is inefficient.

Table1) strong scaling results

NPROC	NX	DT	TIME
1	4000	0.0002	5683.23
4	4000	0.0002	2169.53
16	4000	0.0002	674.11
64	4000	0.0002	175.09

strong

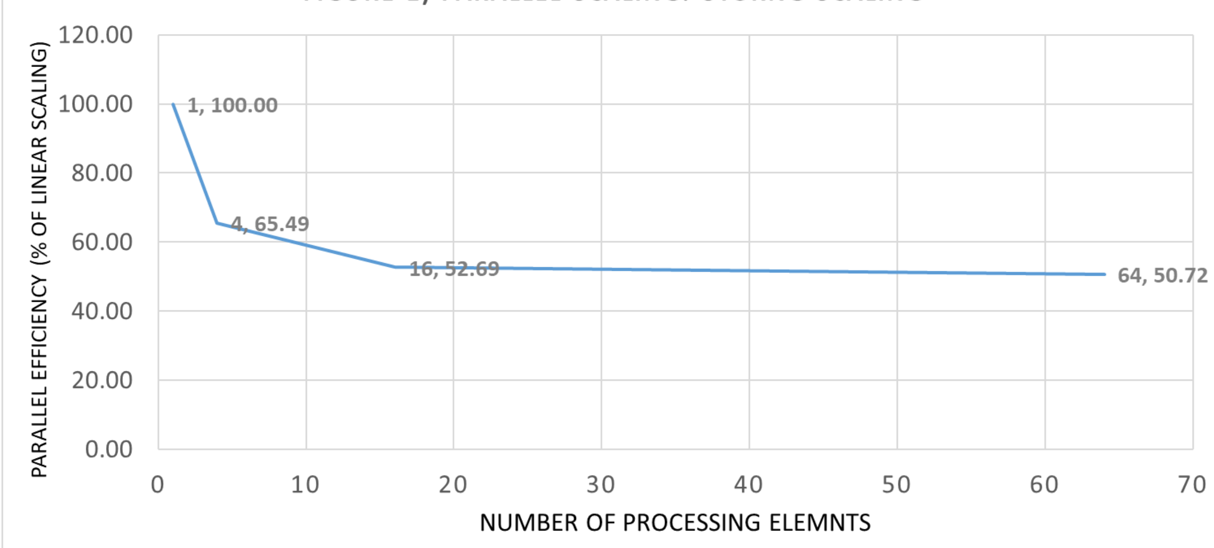


FIGURE 1) PARALLEL SCALING: STORNG SCALING

nx should be a product of a constant and sqrt(p) to get real weak scaling, as nx is the number of points in one direction, and p is the total number of processes in a 2d grid

The week scaling was setup in a way that Nx size would be the product of number of CPUs and a constant size of 80 and Dt = 0.000125. Results indicated that for week scaling parallel efficiency starts from 86% with 4 CPUs and goes up to 100% with 16 CPUs, however this might not be true as it should decrease or stay constant. The 64 CPUs goes to week scaling efficiency of 60%. It appears that this problem is better for week scaling than strong scaling.

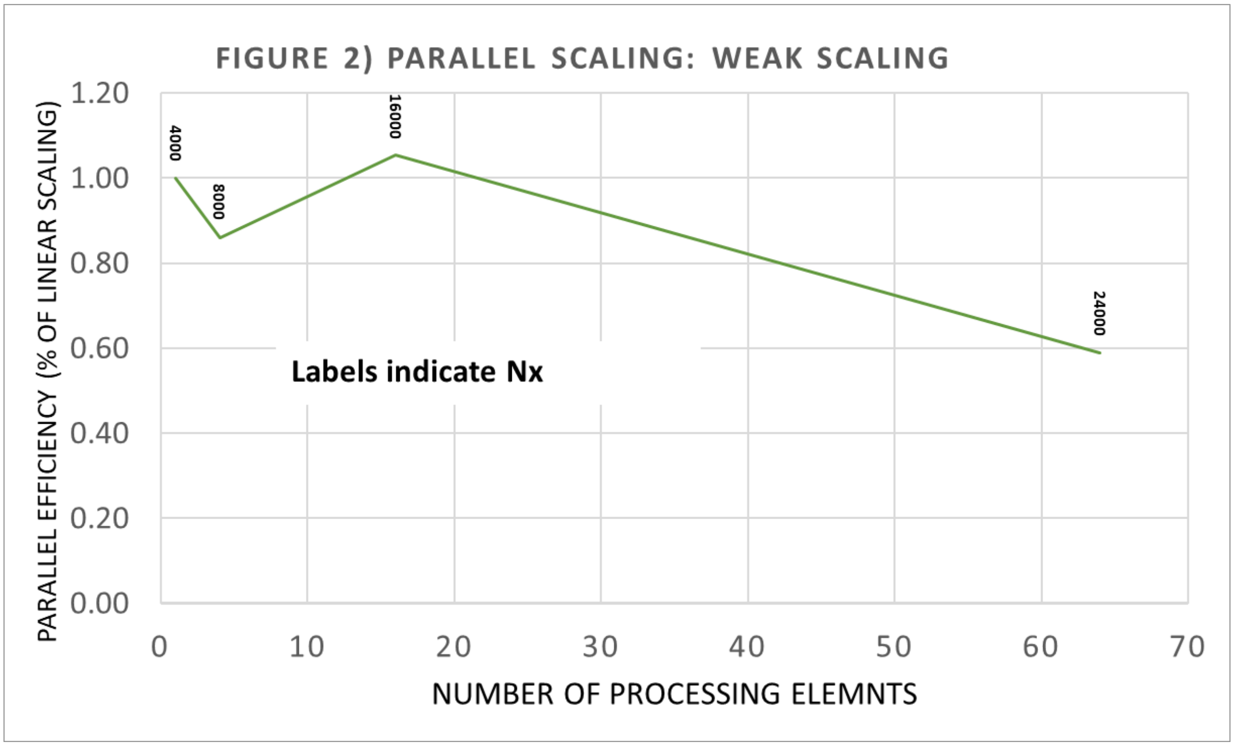
Figure 3 on the other hand, that the speedup of this algorithm for problem size of Nx= 4000 and Dt = 0.0002 is far from the ideal parallel process form the very beginning. This indicates that we could not completely achieve the communication and computation overlap and if all communication could be done by masking it in computation then we could have achieved the ideal parallel process.

weak

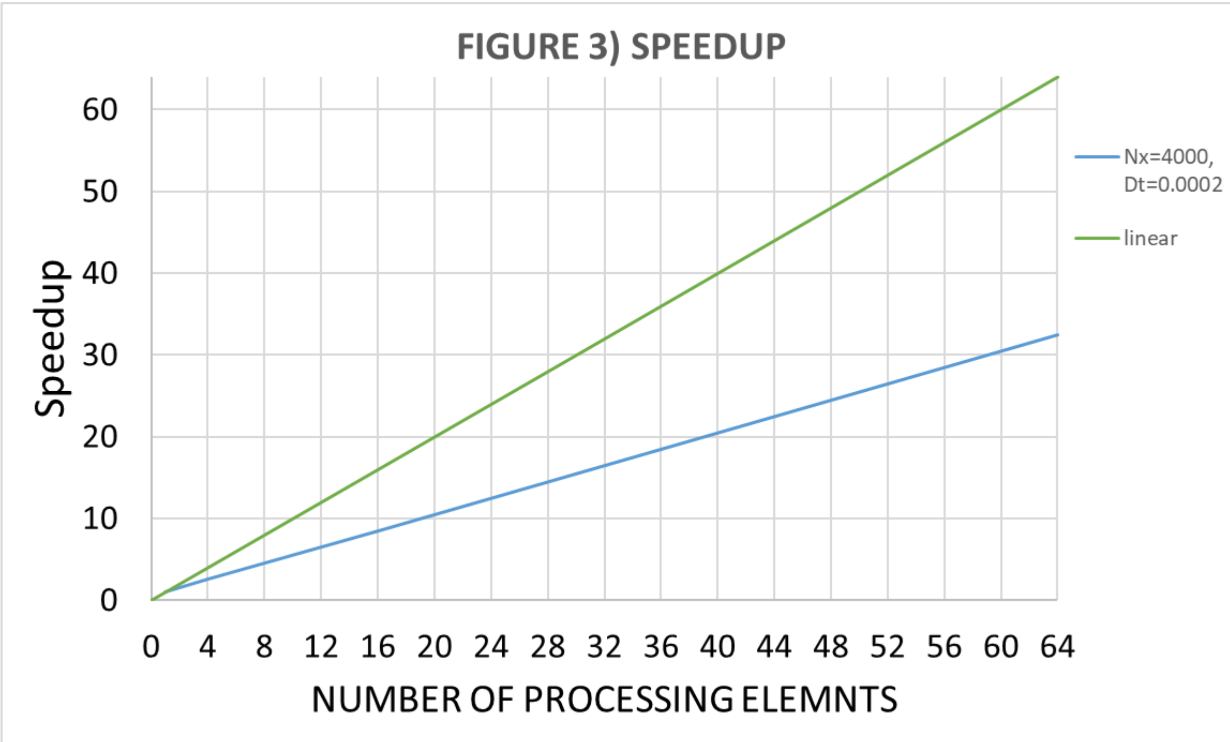
Table 2) week scaling results

NPROC	NX	L	TIME
1	4000	100	1001.24
4	8000	200	861.78
16	16000	400	1055.25
64	24000	600	590.20

this is a very good weak scaling result



use log log scale please



Task 3 - Analyze the theoretical performance

2 pts

You make some good analysis here, but I don't understand how you got the estimates for computation and communication. It would help if you explained where you got those numbers from, otherwise I need to guess.

Assuming the following parameters,

$nproc$ = number of processors, isn't p the number of processors too?

n_{x-loc} = number of elements in a single row of local matrixes

n_{y-loc} = number of elements in a single column of local matrixes

n_{loc} = number of elements in a local matrix

n = total number of elements

We can write,

it would be good to show where those numbers come from

$$T_1 = T_{comput} = \gamma.ops = \gamma.(16n^2 + 4 * 16n + 3 * 10n^2 + 4 * 3 * 10n) = \gamma.(46n^2 + 184n)$$

I think it is unlikely that the number of operations will be a square of n, where n = nx * ny (per your description). I assume you meant n to be nx - number of elements in one direction

$$T_p = \frac{\max\left(T_{comput,overlap}, \left(T_{latency} + T_{bandwidth}\right) + T_{comput,remain}\right.}{p} =$$

what is this? There is 4 messages being sent by each process

$$= \max\left(\gamma.\left(16n_{loc}^2\right), \left(\alpha * 4 * \left(nproc - nproc^{0.5}\right) + \beta * 2 * n_{x-loc}\right.\right.$$

what does this represent?

$$* \left(nproc - nproc^{0.5}\right) * 2 * n_{y-loc} * \left(nproc - nproc^{0.5}\right)\Big)\Big)$$

$$+ \gamma.\left(4 * 16n_{loc} + 3 * 10n_{loc}^2 + 4 * 3 * 10n_{loc}\right)/p =$$

$$= \max\left(\gamma.\left(16n^2\right), \left(4\,\alpha\left(nproc - nproc^{0.5}\right) + 4\beta n^{0.5}\left(nproc - nproc^{0.5}\right)^2\right)\right)$$

$$+ \gamma.\left(30n^2 + 184n\right)/p$$

I think we can overlap all O(n^2) computations, and only not overlap the computation at the processor edges, which is O(4n)

$$= \max\left(\gamma.\left(16n^2\right), 4\left(nproc - nproc^{0.5}\right)\left(\alpha + \beta n^{0.5}\left(nproc - nproc^{0.5}\right)\right)\right)$$

$$+ \gamma.\left(30n^2 + 184n\right)/p$$

It can be seen that although we attempted to overlap the computation with communication, still $\gamma.\left(30n^2 + 184n\right)$ more computations remain that are done after the communication part and therefore, this may not be the 100% overlap. The performance of our model can be expressed in terms of speed-up and efficiency in a way that,

$$S_p = \frac{T_1}{T_p}$$

$$= \frac{p.\gamma.\left(46n^2 + 184n\right)}{\max\left(\gamma.\left(16n^2\right), 4\left(nproc - nproc^{0.5}\right)\left(\alpha + \beta n^{0.5}\left(nproc - nproc^{0.5}\right)\right)\right) + \gamma.\left(30n^2 + 184n\right)}$$

In case computation is more time consuming,

$$S_p = \frac{p.\gamma.\left(46n^2 + 184n\right)}{\gamma.\left(16n^2\right) + \gamma.\left(30n^2 + 184n\right)} = p$$

Which we have completely hide the communication and the ideal way or the communication is more time consuming and,

$$S_p = \frac{p.\gamma.\left(46n^2 + 184n\right)}{4\left(nproc - nproc^{0.5}\right)\left(\alpha + \beta n^{0.5}\left(nproc - nproc^{0.5}\right)\right) + \gamma.\left(30n^2 + 184n\right)} < p$$

For efficiency we can write,

$$E_p = \frac{S_p}{p}$$

And therefore, we have to scenarios. One if we have complete overlap of communication and computation,

$$E_p = \frac{S_p}{p} = \frac{p}{p} = 1$$

So we achieve 100% efficiency, and if this is not the case,

$$E_p = \frac{S_p}{p} = \frac{\gamma.\left(46n^2 + 184n\right)}{4\left(nproc - nproc^{0.5}\right)\left(\alpha + \beta n^{0.5}\left(nproc - nproc^{0.5}\right)\right) + \gamma.\left(30n^2 + 184n\right)} < 1$$

and in this case, we will never achieve 100% efficiency.