Solution submissions

Aymane Ghanam
12:11
Hello,
Can someone clarify where exactly should we submit our solutions for the different sheets?

Mira Salmensaari
15:03
You should have a directory only you and course staff has access to in:
/scratch/courses/programming_parallel_supercomputers_2023/username

you can get to the folder after logging into Triton with:
cd /scratch/courses/programming_parallel_supercomputers_2023
There you should find a directory with your username where for each exercise-sheet you make a folder for it and include the returnables in the folder you just created

Sheet 3 Ex1

The tips for Sheet3 Ex1 discussed in the lecture were: look at the example programs MPI_loctime.c and synchro.c. The former implements a skeleton of the exercise, and is timing some communication, but not exactly in the correct way (to time a matching send and receive). You should figure out how to improve. This example also checks whether the MPI library used is synchronizing clocks globally, that is, between the nodes. For all MPI libraries I have tested so far, no global sync is provided. The way of doing that yourself is presented in synchro.c program.

Vili Kohonen
14:08
For the first exercise communication-time measurement, do we have to send data both ways between the two processes and not just unilaterally? I guess the latter would produce less interesting benchmarks.

Rheinhardt Matthias
16:25
Hi Vili,
unilateral is OK, and I see no reason why bilateral should be more interesting. But you are free to include MPI_Sendrecv in your choices.
We may discuss this in more detail on Friday.

Henri Södergård
22:09

Just want to make sure that I've understood the assignment correctly: "three different combinations out of the spectrum of the MPI send/recv routines" - I assume this relates to the different ways of sending/receiving messages, i.e. "regular" blocking, synchronous blocking and buffered blocking, etc.?

Rheinhardt Matthias
22:23
Hi Henri, yes that's right.

You subscribed to stream Exercises
Exercises
>
Sheet3 ex1
NOV 10
Vili Kohonen
14:28
I was able to get this working unilaterally. At points I got very large time values with synchronous send but that was fixed by having unique tags for each send/recv pair.

Now the problem is that I still get -nan values for times seemingly randomly. Any idea where this issue might generate from?

Mokeev Danila
17:55
Hi, I was wondering if implementing it with a bilateral approach is still correct (and then dividing the time by 2)?

Rheinhardt Matthias
04:09
Hi Vili,
interesting that the messages really got messed up without unique tags. The NaNs should not appear, though. I would recommend to track them down systematically starting with the return values of MPI_Wtime.

04:19
Hi Danila,
bilateral is possibly OK, but if there is any concurrency of the two send-receive pairs (which is likely) you will have a hard time to infer from the measured total time to the time of one such pair alone. So unilateral is cleaner.

Sheet 3 Ex2
08:13

The tips for Sheet3 Ex2 discussed in the lecture were: First, we inspected (by drawing) the type of a stencil that is used in the problem. We inspected how the solution looks like (the initial condition is a sine wave; when it is advected, it should retain its shape and amplitude, but move to some direction depending on the velocity given). We discussed the division to subdomains, mapping to processes, and finding the neighbors, one possibility presented in the example program Comm_1.c, where a Cartesian communicator in torus geometry is used. We discussed the possible RMA communication schemes that one can choose (for active, see One_sided_1.c, for passive see One_sided_3.c). In the case of passive RMA scheme, the further tip was to see Split2.c, where the needed communicator splitting and process grouping is presented. In addition, one needs to decide what kind of datatype would be handy to use; a vector data type usage is exemplified in Datat_1.c. The basis for grading is that the code is functional, and produces a correct solution. How you accomplish it (which datatype you choose, which communication scheme, how you find neighbors,..., and how performant they are) are not in the focus of this exercise.

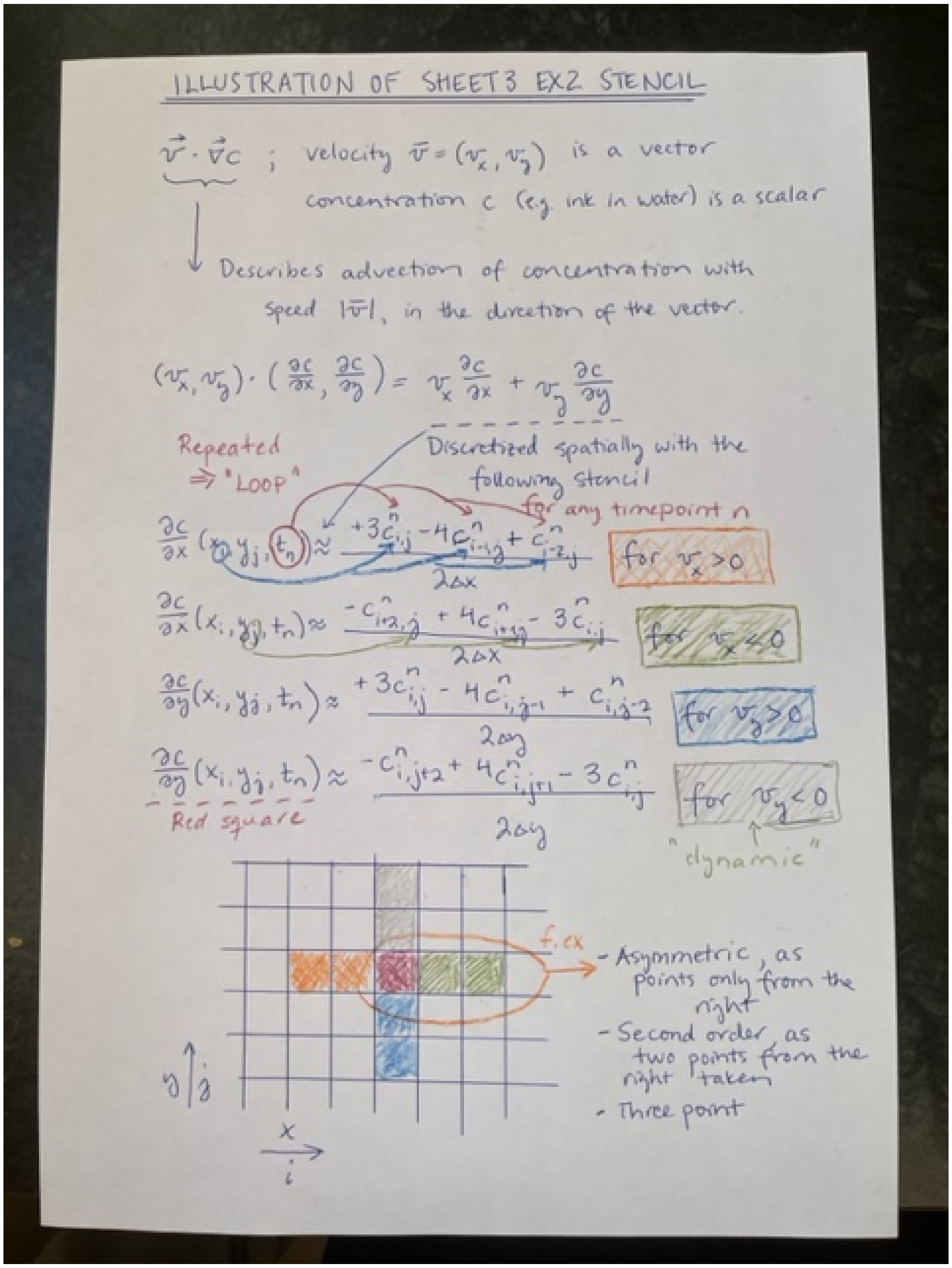

ILLUSTRATION OF SHEET3 EX2 STENCIL

$\underbrace{\vec{v}\cdot\vec{\nabla}c}$; velocity $\vec{v}=(v_x, v_y)$ is a vector

concentration c (e.g. ink in water) is a scalar

↓ Describes advection of concentration with speed $|\vec{v}|$, in the direction of the vector.

$$(v_x, v_y)\cdot\left(\frac{\partial c}{\partial x}, \frac{\partial c}{\partial y}\right) = v_x\frac{\partial c}{\partial x} + v_y\frac{\partial c}{\partial y}$$

Repeated ⇒ "LOOP"

Discretized spatially with the following stencil

for any timepoint n

$$\frac{\partial c}{\partial x}(x_i, y_j, t_n) \approx \frac{+3c^n_{i,j} - 4c^n_{i-1,j} + c^n_{i-2,j}}{2\Delta x} \quad \boxed{\text{for } v_x > 0}$$

$$\frac{\partial c}{\partial x}(x_i, y_j, t_n) \approx \frac{-c^n_{i+2,j} + 4c^n_{i+1,j} - 3c^n_{i,j}}{2\Delta x} \quad \boxed{\text{for } v_x < 0}$$

$$\frac{\partial c}{\partial y}(x_i, y_j, t_n) \approx \frac{+3c^n_{i,j} - 4c^n_{i,j-1} + c^n_{i,j-2}}{2\Delta y} \quad \boxed{\text{for } v_y > 0}$$

$$\frac{\partial c}{\partial y}(x_i, y_j, t_n) \approx \frac{-c^n_{i,j+2} + 4c^n_{i,j+1} - 3c^n_{i,j}}{2\Delta y} \quad \boxed{\text{for } v_y < 0}$$

Red square

"dynamic"

- Asymmetric, as points only from the right
- Second order, as two points from the right taken
- Three point

Perhaps the last tip for Sheet 3 Ex2 was to look at Sheet5 implementation. It is using a symmetric and static stencil, a two-sided communication scheme, and solving for the diffusion equation, but the basic functional principles of this program are the same that we are after in Ex2 implementation. We will go through this code in detail next week in the lecture and exercise session.

To measure the level of concurrency, do we typically want to change the number of tasks to check how the level of concurrency change or do we just need one fixed number of task for that (The latter makes sense to me but there is no explicit question in the description)

Rheinhardt Matthias
03:13
One choice for the task number is sufficient in the concurrency test, but it should be a decent one: The number of points in the halo, which determines the size of the communicated data packages should be markedly smaller than the number of points in the whole subdomain of a process. You are of course free to try out some variations.

Accuracy of EDP model

Henry Virgile
11:07
Hello,
How accurate is the pde model supposed to be ? with 64 grid points, I start with an average error of 0.002, but after 50 iterations it goes up to to 0.3. I'm not sure if this is due to the lack of precision of the simulated model, or to my mistakes.

Rheinhardt Matthias
18:34
Hi,
in numerical mathematics, all is about convergence. So, if you find that the growth of the error slows down at higher resolution (grid spacing and/or timestep) you are fine.

Initialization

de la Brassinne Bonardeaux Maxence
09:15
Hello,
In the initialization of the data array, I noticed the x and y start at halo_width. Therefore, the first 2 columns and rows are reserved for the halos. However, I don't understand why it does not depend on the sign of u_x and u_y as the halos will be on the right or left side of the domain depending on the sign of u_x.
Could you clarify this.

Thank you.

Maxence de la Brassinne

Rheinhardt Matthias
02:20
The halos don't need to be initialized. If they are, they will be anyway overwritten during the first halo exchange according to u_x/u_y.
And the initialization doesn't depend on the direction of the velocity.

Linear error

de la Brassinne Bonardeaux Maxence
17:32
Hello,
Is it normal that the error groes linearly with the number of iterations in sheet 3?
Thank you

Rheinhardt Matthias
02:21
Given that Euler's method is first order, I would say so.

Rheinhardt Matthias
18:38
Yes, for the time integration the rhs is needed and for this, in turn, the derivatives are.

Xuan Binh
18:52
@Rheinhardt Matthias
Regarding the RHS, if I understood correctly, I should use it to update d_data in each iteration, isnt it?
If that is the case, then how is it related to the analytic solution?

```
    for (ix = ixstart; ix < ixstop; ++ix)
       for (iy = iystart; iy < iystop; ++iy)
       {
          data[ix][iy] += dt*d_data[ix][iy];
       }
    t = t+dt;
```

This loop over here calculates the numerical solution, isn't it? I want to calculate data_true[ix][iy] as analytic solution, how do I approach it?

Rheinhardt Matthias
19:00
The analytical solution is defined in phys_appl.md as a function of x, y and t, employing the initial condition, which is a function of x and y. So, take what you have used as initial condition (e.g., sin(x)) and replace its argument x by (x-v_x t), and y analogously (if neeed) to obtain the analytical solution.

Xuan Binh
19:33
Hi, can I have some hint on the MPI_Get? Exactly what I should get the data for each process? And additionally, this is a rectangular grid, so must I pay attention to the grid border or the grid corner? Is this boundary problemall managed by the rhs function already? Or this field is a torus so the boundary actually refers to the other side?

Rheinhardt Matthias
03:31
I can't provide you the essence of the solution. Please analyse which points in the subdomain of a process need data from neighbouring processes in order to be able to evaluate the stencil for the u.grad operator at their positions. The domain is supposed to be periodic in x and y, so, indeed, it forms a torus. This topological property has to be expressed by proper neighbourhood relations between the processes.

Xuan Binh
11:28
Hi, I would love to know that what MPI functions should I use for exercise 3 besides the MPI win, fence and MPI_GET? :smiling_face_with_tear: Is there any need to use MPI_Cart_create or MPI_Type?

Maarit Korpi-Lagg
13:12
Hi Binh. The MPI functions for communication you propose sound right. No obligatory need in using Cartesian communicator and no preferred MPI datatype that you should use.

Xuan Binh
13:14
Thank you for your answer. But please help me professor :smiling_face_with_tear: ... I am now feeling confused about the periodicity and torus topology. Is your template code given by rheinhardt already takes care of it, or do I need to also manage periodicity and torus topology in my MPI_Get and updating the data matrix as well?

Maarit Korpi-Lagg
13:16
I do not think the torus topology is taken care of automatically, as this is one part of the learning outcomes of this exercise. Note that using Cartesian communicator to map the processes you can declare whether your communicator is a torus or not. Maybe this speaks for the usefulness to use this functionality.

13:18
Please use sheet5 code to understand how this functionality is used.... There you find an example for a static and smaller stencil, but the basic torus topology is the same.

Xuan Binh
13:18
Are you implying that if I use the Cartesian, it would help me taking care of the torus topology and I do not need to check the boundary by myself?

Maarit Korpi-Lagg

13:18
Yes, I am implying that.

Xuan Binh
13:19
I understand now. How about your opinion on periodicity? Is it already been assumed by the sin function and I dont need to care about this point?

Maarit Korpi-Lagg
13:19
Do not stress, just look at sheet5 example code (the one with MPI only).

13:19
sin function is periodic

13:20
should work out of the box if the neighbors that you get the data are chosen appropriately.

Xuan Binh
13:23
In a periodic domain, the neighbors of subdomains on the grid edges are the subdomains on the opposite edges. if we have a grid of processes in a 2D plane, the right neighbor of the rightmost process would be the leftmost process in the same row, and similarly for top-bottom neighbors.

If this is the case, the Cart would also take care of the periodicity, since it should correctly identify the neighbor, isnt it?

Maarit Korpi-Lagg
13:23
Meaning of this: if you chose the initial condition appropriately, that is, it is periodic which is needed for the torus mapping of neighbors to work, then if the correct neighbors exchange data, the solution should remain as a sine wave with the same amplitude, but only "move" in the direction you have asked it to move.

15:00
Xuan Binh said:

In a periodic domain, the neighbors of subdomains on the grid edges are the subdomains on the opposite edges. if we have a grid of processes in a 2D plane, the right neighbor of the rightmost process would be the leftmost process in the same row, and similarly for top-bottom neighbors.

If this is the case, the Cart would also take care of the periodicity, since it should correctly identify the neighbor, isnt it?

Yes, you are right.

15:12
But now I need to take a sauna break... Hopefully this sheet (3) will not cause too much misery for you. Ex2 has been very challenging for many students, and naturally we will adjust the grading scale to match the outcomes.

Xuan Binh
15:21
so professor @Maarit Korpi-Lagg , if my solution does not converge (my code has something flawed), but nevertheless has a structure, using all necessary MPI functions and prints out the error between numerical and analytic solution, will I get nonzero point and receive your referenced solution?

Maarit Korpi-Lagg
16:41
Sure, but note that diverging results are expected when low-order schemes are used. Linearly growing error should not be a showstopper, according to @Rheinhardt Matthias. Matthias has also been notified for the need of defining more concisely what is an acceptable solution or not. Sorry for this part of sheet3 having been so vague. we will improve this part in the future.

Xuan Binh
16:43
Hi, you are back. May I ask that can I initialize the halos of data as 0 at the beginning, and fill d_data with 0s ?

Additionally, how large should domain_nx and domain_ny be?

I'm determined to nail this exercise :sob:

Maarit Korpi-Lagg
16:44
Initial state of the halos is not important, as this will be overwritten during the next step.

16:45
Subdomains should be large enough to be in the computationally bound regime. So subdomains should not be too small.

Sheet 4:

Kähkönen Samu
15:57
Can we assume that the grading will be done using the job scripts provided in the source codes? So, for example in sheet4, qs_dist.sh has

```
#SBATCH --nodes=2 #Use two nodes
#SBATCH --ntasks=8 #Eight tasks
```

Will the grading also be done with 8 processes, or it will it be with another number?

Puro Touko
16:08
Kähkönen Samu said:

Can we assume that the grading will be done using the job scripts provided in the source codes? So, for example in sheet4, qs_dist.sh has

```
#SBATCH --nodes=2 #Use two nodes
#SBATCH --ntasks=8 #Eight tasks
```

Will the grading also be done with 8 processes, or it will it be with another number?

Hi,
based on my knowledge the grading would be done with the job scripts provided, this is true for at least sheet 4, but when using multiple nodes and processes your solutions shouldn't work for only a specific configuration (two nodes and eight tasks in this case) but should in theory work for all configurations.
I am having a hard time to come up with a example where knowing the number of processes a priori would even be that important since you can get that information at runtime.
Is there something specific on the job scripts that you worry about?

Kähkönen Samu
17:44
@Puro Touko You are correct that the exact amount of processes does not matter, but can we assume that the amount of processes used will be even? I can see there being some trouble with an odd amount of processes

Puro Touko
20:11
Kähkönen Samu said:

Puro Touko You are correct that the exact amount of processes does not matter, but can we assume that the amount of processes used will be even? I can see there being some trouble with an odd amount of processes

Oh I see good point, yes you can assume that the amount of processes is even

<u>Stability of quicksort</u>

Henry Virgile

13:28
Hello,
I'm struggling with the stable part. Quick sort seems to be inherently unstable, and attempts to make it stable either use another buffer, or are significantly slower and at this point it's better to use merge sort.
(I'm sorry if I'm not supposed to post links, I can remove this message but I'm struggling too much wit stabilizing quicksort).
How important is the stable part ? In our case we are sorting floats, so stability does not matter, and the solutions I'm considering is to either make a significantly slower sort, or use more space but then I'll have to change the function signature (as it is recursive, I don't want to allocate memory at each call) to pass another buffer around (or use this function to call another routine that actually performs the sort, but then again I'll change the signature).

Puro Touko
14:19
Hi,
You are allowed to use more space, since we hint about using prefix sums and that naturally requires more memory. I will make it more explicit in the instructions that you are allowed to use additional memory.
Do not change the function signature since the main file depends on it. It is okay and a good idea to have a different helper functions that you call from quicksort, naturally their signature does not matter (using helper functions was also hinted in the exercise instructions)
Would assume that all of the students would be able to google the links you provided, but since one of the links has an explicit solution for the first task could you please edit the message to remove them

GPU

Huynh Quang Long
18:21
Hello, can we assume the size of the array <1e6, or should we do an iterative reduction for the prefix sum?

Puro Touko

Hi,
You can assume the size of the array is <1e6, which means that the whole array will fit inside a single grid and when performing the recursive prefix sum on the block sums a single level of recursion should be enough (even though this probably would not affect how you structure the recursion).

09:28

For those having difficulties with the GPU exercise it is highly recommended to take a look at the article provided in the GitLab hints:
https://developer.nvidia.com/gpugems/gpugems3/part-vi-gpu-computing/chapter-39-parallel-prefix-sum-scan-cuda

There you will find an introduction to performing prefix sums on the GPU and a way to use them for performing radix sort, from which you can think about how to leverage prefix sums to perform quicksort

Regarding performance: usage of streams is optional and won't affect grading and usage of shared memory is only expected when performing the prefix sum algorithm.

Huynh Quang Long
12:01
Puro Touko said:

Huynh Quang Long said:

Hello, can we assume the size of the array <1e6, or should we do an iterative reduction for the prefix sum?

Hi,
You can assume the size of the array is <1e6, which means that the whole array will fit inside a single grid and when performing the recursive prefix sum on the block sums a single level of recursion should be enough (even though this probably would not affect how you structure the recursion).

I see. Tyvm!

Sheet 5

Perhaps the last tip for Sheet 3 Ex2 was to look at Sheet5 implementation. It is using a symmetric and static stencil, a two-sided communication scheme, and solving for the diffusion equation, but the basic functional principles of this program are the same that we are after in Ex2 implementation. We will go through this code in detail next week in the lecture and exercise session.

sheet 5 easy throughout, you just locate the green points of the subdomain and openMP this part, and then do scaling tests.

So you understand the stencil computation in task 5? In the baseline this computation is parallelized with only MPI processes and your job is to try a hybrid multithreading approach with MPI+OpenMP. In core.c there are the loops corresponding to the stencil computations that you have to parallellize with multithreading with OpenMP. After you have done that you should compare the MPI vs. MPI+OpenMP in terms of, memory used per process, in terms of performance, and in terms of scaling (weak and strong scaling) in zulip there are links to documentation of commands to measure memory usage (seff and sacct). To measure scaling you should vary the number of processes and the grid size, both for weak scaling and only the number of processes for strong scaling.

Puro Touko
15:14

Documentation about sacct and seff:
https://csc-training.github.io/csc-env-eff/hands-on/batch_resources/tutorial_sacct_and_seff.html

https://slurm.schedmd.com/sacct.html

15:14

Did not have time to mention on the exercise session, but it can be useful to turn the I/O off when doing your measurements for exercise 5

Maarit Korpi-Lagg
16:57

This can be done increasing
main.c: int image_interval = 500; //!< Image output interval
Sorry, hardcoded...

17:15

Tasks and tips for Sheet 5 (mentioned in the lecture):
you need to decide what is the suitable thread support level from MPI, and implement that.
find out the location in the code where the most intense computations are happening, and use threading there. Hint: look at core.c, and locate Lecture 3 stencil example green zones computation. You might think whether threading the computation of the red cells can give you a significant gain.
Remind yourself of the weak and strong scaling scenarios. To aid you with that, Lecture 5 in class.pdf has been added to MyCo. There are practical examples (data and plots) of both types of experiments.
Think of a suitable grid size w.r.t. the number of cores, giving you a reasonable subdomain size not to hit the communication-bound regime too early in your experiments. Hint: the default of 2000x2000 is in the right ballpark, but for strong scaling consider whether an even larger grid

would be beneficial, and for the weak scaling, whether a bit smaller grid to start with could be wise.
All PPC material and tricks on openMP are allowed, but nothing extra to these is required.
Use the seff and sacct tutorials provided by Touko.
Plots similar to Lecture 5 in class.pdf are expected + memory profiling.

Sheet 6

kernel signature

Rantanen Riku
10:21
Is it okay to modify the function signature for the reduce_kernel in exercise 6? It doesn't seem to be called anywhere during the testing.

Puro Touko
12:29

yes, it states in the instructions:
"Only the interface function int reduce(const int* arr, const size_t count) declared in src/reduce.cuh is used for grading. You can add additional helper functions if needed.", this implies that only the reduce function is being dependent on and called from other files. Will make it more explicit in the instructions

About the fourth hint. Isn't the default bottle.dat 200x200 instead of 2000x2000?

Maarit Korpi-Lagg
06:58
@Wu Tianxing The size of the data file is 200x200, but the default grid size is 2000x2000.
int rows = 2000; //!< Field dimensions with default values
int cols = 2000;

Conversation with Mr Puro Touko

Puro Touko: Hi,
It is getting harder to help you without seeing your code so I know what you are exactly doing, so if could send me your code for the task here as a private message so I can help you better?

Xuan Binh: HI sir :smiling_face_with_tear: Please help me this evening with this task. Im so lost

Xuan Binh: reduce-multi.cu

Puro Touko: No worries, I will help you!

Xuan Binh: I have a feeling I calculate the wrong number of blocks... I think that the number of blocks x number of threads per block x number of gpu should be the count passed to the function :smiling_face_with_tear:

Puro Touko: So first all you have tested that the reduce kernel works by running it on task 1, right?

Puro Touko: Can I see your code for it?

Xuan Binh: :smiling_face_with_hearts: Yes it did.

```
Position: 0, Model: 1, Candidate: 1, Correct? Yes
Position: 2301, Model: 2, Candidate: 2, Correct? Yes
Position: 71112, Model: 3, Candidate: 3, Correct? Yes
```

It says YES.

Xuan Binh: reduce-single.cu
Here it is sir

Xuan Binh: I proceed to put the code in single gpu in a for-loop in task (2) but now it fails :cry: I think I need to look at the int blocks one

Puro Touko: You see for task 1 you have the reduction across blocks and you don't have that anymore for task 2

Puro Touko: Furthermore, as it says in the instructions for task 1, you should apply the reduce kernel iteratively so you should do the reductions across the blocks also as a kernel call, not on the host side. You should move only the resulting scalar back to host

Xuan Binh: Puro Touko said:

Furthermore, as it says in the instructions for task 1, you should apply the reduce kernel iteratively so you should do the reductions across the blocks also as a kernel call, not on the host side. You should move only the resulting scalar back to host

Really? I thought that reduction on the host should be fast as there are only very few gpus... Or is that the point where I actually get wrong?

Puro Touko: so we are talking about task 1 now, so only a single gpu. Performing the reduction kernel iteratively on your data is in general better since you don't want to make assumptions about the size of the data you have been passed

Puro Touko: what I mean in your reduce-single.cu you have this loop:

```
int global_max = INT_MIN;
for (int i = 0; i < blocks; ++i) {
global_max = max(global_max, h_max[i]);
}
```

You should replace it with a call to your reduce_kernel, and finally only move a single int back from the device to the host

Xuan Binh: so it is reduced until the size of the array is only 1?
I dont understand well. Suppose we have 16 numbers and 1 gpu, what could happen?

Puro Touko: Sorry don't understand what you mean by what could happen?
But yes reduce the input array on the device until there is only a single element left, which is the maximum you wanted

Puro Touko: The reason why is it better to reduce until a single element is because let's say you get passed an array of 100 million ints. After performing a single reduction on it, the resulting array would still be quite large, so it is better to do it on the device.
As a general coding principle the less assumptions you have to make about your input the better

Xuan Binh: oh! so it depends on number of blocks for reduction, right?

Puro Touko: What depends?

Xuan Binh: :smiling_face_with_tear: Let me see, 100 million ints, and by default we usually have 256 threads and 256 blocks for one kernel calling

Puro Touko: The 100 million ints, was just an example don't get too caught up in it. You can assume that the array is not larger than 1024*1024, so it will fit into a single grid

Xuan Binh: 100 million /256 blocks / 256 threads =approx 1525 ints left. Then we apply another kernel on 1525 integers and get 1 single value. Do you think what I say is correct?

Puro Touko: So let's go with a smaller example. 1024 array and you launch the kernel with 4 blocks and 256 threads. Then you have 4 values left that need to be reduced

Puro Touko: If you launch the kernel with n blocks then after it you have n values left to be reduced

Xuan Binh: wait a moment... in practice it is usually at most 2 reductions and it already returns 1 single value maxint, isn't it? Since the first reduction is very comprehensive already

Xuan Binh: I didnt see any assumptions that the passed data fits in a grid. Only that the initial_count is divisible by the number of gpus in task 2

Puro Touko: Xuan Binh said:

wait a moment... in practice it is usually at most 2 reductions and it already returns 1 single value maxint, isn't it? Since the first reduction is very comprehensive already

Well yes, two kernel launches is enough to reduce all arrays that size is less than nblock*nblock so if nblock=256 then it is enough to reduce twice for all arrays less than approx. 60 000 elements

Puro Touko: But yeah, since you are doing grid strided loops you don't need to worry about the data fitting inside a single grid. The grid sizes for current GPUs are anyways quite massive so that issue comes up quite rarely

Xuan Binh: ah, I see.. so each element you mention in 60000 elements example is actually an array, and it is the job of one thread to reduce that array. I was confused for a moment. Is this what you mean?

Puro Touko: No, apologies if my explanations are confusing so let's start from the beginning :upside_down:

Puro Touko: You have the loop over the blocks for task1:

```
int global_max = INT_MIN;
for (int i = 0; i < blocks; ++i) {
global_max = max(global_max, h_max[i]);
}
```

This reduction should happen on the device

Puro Touko: After each reduction call you will have blocks number of ints left to reduce, where blocks is the number of thread blocks you launched the kernel with.

Puro Touko: So lets's say you start with an array of ints size 1024 x 1024 and reduce this once with a kernel launch with n blocks. Then you have n ints left to reduce. If your block dim is 256 then n would be 1024 x 4. After reducing again with block dim of 256 you have 4 x 4 elements left to reduce. You do one final reduction after which you have only a single element left that you copy over to the host. My hint and suggestion would be to do this with a while loop

Xuan Binh: suppose that I manage to solve this problem. Do you think it will help me solve task (2) error as well? :pleading_face: or this is just an optimization improvement

Puro Touko: yes, since what was missing in task (2) was the reduction across blocks. You reduced only once, which is not enough because of the reasons we have discussed here. If you get the logic down of reducing until a single element on the device then you can simply copy and paste that to task (2).
Optionally you can copy and paste your code from task (1) that does the reduction across blocks on the CPU (the for loop across blocks). Be warned though you won't likely get full points then, but I understand you are pressed for time so getting a partial solution for tasks 1 and 2 might be better than getting stuck completely

Xuan Binh: Okay, I will try by myself now.

Before I go and leave you for rest, I want to confirm something... Are you the course staff? :smiling_face: If yes, may I ask if professor will only give us the model solutions to problem that we attempt? If we dont attempt or got 0 point for that task, the teacher wont reveal model solution of that particular task to us? :exploding_head: To be honest, I just started doing assignments 2.5 days ago and now I am panicking :sob:

Puro Touko: I am the course staff, and if I remember correctly you will only get model solutions only to exercises that you submitted answers, at least heard that it was like this last year (this is my first year running this course). Would also assume that half-assed attempts don't count i.e. you get 0 points. But assuming you put some effort to your submission you should get more than 0 points.

Puro Touko: I undestand your situation, have also tried to cram myself too much at the end of a course :upside_down:, so good luck! You can ask me help during the weekend and Monday, but during the weekend might not answer as fast as on Monday

Xuan Binh: no... I want to see all model solutions. Then I have to equally attempt all exercises to get above 0 points :sleepy: I can't rest in ease when I don't know the solutions since the CUDA programming has fewer materials on the net so I dont even know what I can miss out on. :smiling_face_with_tear:

Xuan Binh: Thank you so much for your confirm... I pray that I dont have anymore stupid question for you :face_holding_back_tears:

Puro Touko: Your questions aren't stupid so don't worry!

Puro Touko: If you are interested mainly on the CUDA part, then I would suggest focusing on exercises 4 and 6 since they are the only ones that have GPU parts

Puro Touko: And if seeing the model solutions for problems that you didn't return solutions to is important, please confirm what I said from Maarit

Xuan Binh: Hi Mr Touko :smiling_face_with_tear: i think I need to bother you again. I hope you could reveal for me introductory steps, as I also dont know how to start.

(1) In task 6 mpi, where should I start, should I copy the code from multi task 2 and modify it? Or task 3 should not look anything like task 2? And also is there a while loop that use the device to reduce like in task mpi as well?
(2) I just look briefly at task 5 and I have set up the working environment. However I would be very grateful if you can help me where to start and how to tackle this exercise? The instructions are unclear to me.
(3) Finally, the notorious task (3b). No matter how much I read, I dont have a vague idea of what it is about. Would you have any strategy for me to do this task to earn at least non zero points? :sob:

Puro Touko: Hi,
for task6 mpi I would say start with your solution to task1. You have the reduction code for one device which you can copy and paste and then you add the code for selecting the appropriate device for your MPI process and the code for reducing the final result across MPI processes.
So you understand the stencil computation in task 5? In the baseline this computation is parallelized with only MPI processes and your job is to try hybrid multithreading approach with MPI+OpenMP. In core.c there are the loops corresponding to the stencil computations that you have to parallellize with multithreading with OpenMP. After you have done that you should compare the MPI vs. MPI+OpenMP in terms of, memory used per process, in terms of performance, and in terms of scaling (weak and strong scaling) in zulip there are links to documentation of commands to measure memory usage (seff and sacct). To measure scaling you should vary the number of processes and the grid size, both for weak scaling and only the number of processes for strong scaling.
Yes task 3b is quite difficult :upside_down:, so let's try to tackle it in steps. Maybe you could start with the decomposition of the grid.
Do you understand what you are supposed to do in these steps:
"defining a mapping of the N MPI processes (ranks) to the N equally-sized subdomains, into which the computational domain is decomposed"
"figuring out the neighboring relationships of the MPI processes and implementing corresponding functions (see code template)"
So you have full grid 2d xy grid that you have to split to N equally-sized subdomains and each process should correspond to one subdomain. Then each process should know the process id of the process responsible for adjacent subdomains

Xuan Binh: Oh, it sounds so complex :smiling_face_with_tear: Let me process what you says and I will report back to you when I make some progress

Xuan Binh: Hi Mr Puro :smiling_face_with_tear: I am now working on assignment 4. May I ask if Im allowed to use <vector> or <algorithm> in my implementation of quicksort?

Puro Touko: Hi you are allowed to use std::vector and algorithms in your quicksort, but you have to do the sorting yourself i.e. don't just call std::stable_sort, though you can use it as a base case after a couple levels of recursion

Xuan Binh: by the way, how do I know that my implementation is correct in stability? SUppose I have sorted the array, then will the grader also verifies whether my algorithm is also stable?

Puro Touko: Hi, unfortunately no.
That would mean the sorting would have to keep track how the original indexes get mapped to the new indexes. (Could be a improvement we make for next year)
As long as your algorithm uses prefix sums I would assume you have done it in a stable manner.

Xuan Binh: I saw you mentioned about prefix sum a lot, which is a sum of cumulative elements. But how is that relevant to the quicksort? Should I consider only prefix sum for Task (2) and Task (3) only? Is task(1) in sheet 4 also requiring prefix sum?

Puro Touko: you are free to do task(1) and task(2) without prefix sums, but for task(3) by far the easiest way to do it is to use prefix sums.
For task(1) it is a nice bonus that doing it with prefix sums naturally lends it to being stable and if you use prefix sums for task(1) it helps you to understand how to use it for this problem when tackling task(3)
And prefix sums (or more general prefix scans) are a useful building block when doing parallel algorithms

Xuan Binh: I understood Mr Puro, but now, what should success output of task 4 looks like? For me, the prog.out file has these lines

```
Is sorted at rank 0
Incorrect at rank 0!!
```

I am very sure that my quicksort is correct as I have tested on my local machine with 20000 random numbers. At least right now I have not tested stability and only for correctness .

Xuan Binh: image.png

Puro Touko: How are you testing your code locally? Do you test your output against to output after calling std::stable_sort, or do you only is the array sorted or not?

Xuan Binh: I only test if the array is sorted or not, I did not test the stability :sob:

Puro Touko: Stability is probably not an issue (you can't really test if on floats without bookkeeping more information) but on your local test you should test that your quicksort gives the same result as calling std::stable_sort (or simply std::sort)

Puro Touko: If you only test is if the array sorted then you can f.e. example always return an array of all zeros and that would be sorted, although clearly incorrect

Xuan Binh: Is sorted at rank 0
Incorrect at rank 0!!

So how can I interpret this message?
Does it means that my result returns the correct sorting, but the order is not correct, right? And I assume rank 0 here refers to the single thread that runs the algorithm

Puro Touko: rank 0 is the process rank, and since we are not using MPI you have only process with rank 0.
What the output says that even though your resulting array is sorted, it doesn't match the output of calling std::stable_sort on the original array, which is the result we want

Xuan Binh: Oh, I understand what you mean now :smiling_face_with_tear: Okay, let me try again

Puro Touko: Good luck!

Xuan Binh: an important question: if I dont implement the stability (for example professor reads my code and found no stability), will I receive 0 point or just point reduction?

Xuan Binh: Now I know why i got it wrong...

I implement the quicksort and call it with

```
quicksort(data[0], 0, size - 1, data);
```

while the grader is

```
quicksort(data[0], 0, size, data);
```

Now I got it correct part 1 :sweat_smile:

Puro Touko: Great!

Puro Touko: I am responsible for the grading of exercise 4, and I would say that it will not be 0 points, but reduction in points

Xuan Binh: Hi Mr Touko, can you give me some hints on how prefix sum can help quicksort? :sob: how is the parallelization of sequential exclusive sums be helpful to quicksort? I cant relate well to the radix sort

Puro Touko: Prefix sums can be used to do a parallel partition of the array into elements less than and greater than the pivot. And of course performing this partition is one of the critical parts of quicksort

Xuan Binh: Hi Mr Touko, I think I need to focus on dealing with sheet 5 as soon as possible to move on to sheet 3. :smiling_face_with_tear:

Can I ask you briefly about measuring the performance?

First of all, I must run both baseline and hybrid version of MPI.

For MPI baseline, I intend to have these number of tasks

2,4,8,10,16,20,40, which are divisible by 2000

The number of MPI ranks has to be a factor of the grid dimension (default dimension is 2000).

WHat would happen if I choose MPI that is not divisible by 2000? (My Q1)

and in MPI baseline, I always set #SBATCH --cpus-per-task=1, is this true? (My Q2)

Then moving on to Hybrid one, I would always set
#SBATCH --ntasks-per-node=2

and

#SBATCH --cpus-per-task=1, 2, 4, 5, 8, 10, 20

and compare it with the MPI baseline 2, 4, 8, 10, 16, 20, 40 so that they have the same number of processes.

Do you think this approach is valid? (my Q3)

Puro Touko: (Q1) the grid size should be divisible nicely with the number of processes because the grid is decomposed into equal sized parts. The application should warn you if the number of processes doesn't align well with the grid dimensions
(Q2) for the pure MPI case, yes
(Q3) For measurements you won't probably get meaningful measurements if you try to use more cpus then there are on the node (24 per node), thus --cpus-per-task=20 and --ntasks-per-node=2 is too much. Same case if you have 40 tasks per node.

And as you indicated rememeber to vary the grid size as appropriate for weak and strong scaling tests

Xuan Binh: (Q1) I understood, thank you
(Q2) I undertsood, thank you
(Q3) so what should be the limit of the number of tasks per node and limit of processes? SHould I vary the processes or the number of thread per processes?
:smiling_face_with_tear: I still dont understand mr touko. I thought that performance measurement is separate from scaling tests?

Puro Touko: Well yes, but in a sense doing a scaling test if more thorough performance tests. So doing a performance test is kind of embedded in scaling tests. In performance test you are interested in the raw numbers but in scaling you are interested in the overall trend as you add more processing units (threads and processes in our case)

Puro Touko: For the performance test choose a number of cpus that you want to use per node. Then launch that many tasks per node to get the baseline performance. Then decrease the number of tasks and increase the number of cpus per task so that the overall number of utilized cpus is the same.

Puro Touko: As an example you could compare --ntasks-per-node=24 & --ncpus-per-task=1 vs. --ntasks-per-node=1 vs. --ncpus-per-task=24

Puro Touko: Remember task==MPI process, we can use the name that feels more natural to you if you wish

Xuan Binh: :joy: This is all I want to hear from you, phew, I understood. So I only need to measure MPI baseline 1 time, but do many experiments with Hybrid many combinations

Puro Touko: The limit should be 24 per node since you have only 24 cores per node

Xuan Binh: image.png

Xuan Binh: I did search number of processes per node but on triton web I dont know what is my partition, since I left partition empty in my script

Puro Touko: Xuan Binh said:

:joy: This is all I want to hear from you, phew, I understood. So I only need to measure MPI baseline 1 time, but do many experiments with Hybrid many combinations

No, sorry if I was unclear but you should have as many MPI baseline measurements as hybrid measurements. For each measurement what should stay constant is the number of cpus you utilize (with MPI you utilize then with processes and with hybrid you utilize thm with threads)

Puro Touko: I would suggest using the course partition since that is the one intended for the course and your measurements would be similar to other students that only have the courses and courses-gpu partitions

Xuan Binh: Not really, Mrs Maaria encourage me to use common partitions to avoid queueing and GPU problem :exhausted:

Wait I dont get your point.

So I have one MPI baseline as tasks = 24.
Then Hybrid I have many combinations, (1,24) (2, 12), (3, 8) and so on...

What does it mean to have many MPI base line as the hybrids? or for MPI base line I should also measure (1,24), (2,12), (3, 8), ... as well like thehybrid version?

or what you mean is a comparison of of 1 task, many threads vs many tasks, 1 thread version?

MPI Hybrid MPI
(2,1) vs (1,2)
(4,1) vs (1,4)
(8, 1) vs (1, 8)

and so on?

Puro Touko: Xuan Binh said:

Not really, Mrs Maaria encourage me to use common partitions to avoid queueing and GPU problem :exhausted:

Wait I dont get your point.

So I have one MPI baseline as tasks = 24.
Then Hybrid I have many combinations, (1,24) (2, 12), (3, 8) and so on...

What does it mean to have many MPI base line as the hybrids? or for MPI base line I should also measure (1,24), (2,12), (3, 8), ... as well like thehybrid version?

Regarding the partition, yes I would weight Maarit's suggestion more than mine :upside_down:.
To get the maximum number of cpus on your partition you can use:

sinfo -p <partition name here>, which will give you a nodelist and query the nodes on the nodelists with:
sinfo -Nel --nodes <nodelist here> /gives the amount of cores per node.

Puro Touko: So for performance one measurement only is enough but for scaling you need naturally multiple measurements both of MPI vs. MPI+OpenMP

Puro Touko: When doing weak scaling for each grid size have a pure MPI measurement and MPI+OpenMP measurement.
Same for strong scaling

Xuan Binh: Okay, let me confirm.

So in performance test, I only have one (24, 1) for baseline and (1,24) for Hybrid comparison and that's enough (2 measurements only)?
In memory test, is it also the same tests as performance but instead of time, we measure memory usage?

In strong scaling, lets say I have grid size fixed at 2400.

Now, what I need to conduct is running

base MPI as [1,2,4,6,8,10,12,16,20,24] tasks
Hyrbid MPI as 2 nodes and and cpu per tasks as [1,2,3,4,5,6,8,10,12] ?
Please help me with these first before I hope I can ask you about weak scaling in details :pleading_face:

Puro Touko: 2400 x 2400, grid is probably adequate but we have hinted at using a 4000 x 4000 grid before. The main point is that the grid should be large enough for there to be parallel work

Puro Touko: So you should have as many utilized cores per node for the measurement to be fair. So for hybrid it would be 2 tasks per node, not two nodes. What would be even simpler have one task per node and [1,2,4,8,16,24] cores per task. For each measurement I would have the node count as constant that is larger than 1

Xuan Binh: Puro Touko said:

So you should have as many utilized cores per node for the measurement to be fair. So for hybrid it would be 2 tasks per node, not two nodes. What would be even simpler have one task per node and [1,2,4,8,16,24] cores per task. For each measurement I would have the node count as constant that is larger than 1

oops sorry I mean tasks, not nodes

Can you help me confirmed about performance and memory usage? Should I have only 2 measurements?

Puro Touko: Since the instructions do not ask you for the scaling of memory usage, you can have only 2 measurements for memory usage (baseline vs. hybrid). Again make sure your grid is large enough
Would not be that hard to measure the scaling of memory usage since you are already measuring the scaling of performance, but is not required and I know you are pressed for time, so no it is not required

Xuan Binh: :smiling_face_with_tear: I think I spend much more time trying to understand what to do than actual coding.

Puro Touko: Relatable :upside_down:

Xuan Binh: Thank you for your confirm... Now lets consider weak scaling

Xuan Binh: Weak scaling as I think follows gustafson law

Xuan Binh: where scale up can be led to infinity

Xuan Binh: Now, you mentioned that we should have varying grid size and also varying number of tasks and threads, which has 3 varying variables

Xuan Binh: If I say, I would have a graph of time (y axis) vs (number of processors - x axis). On this graph, there are multiple lines corresponding to each grid size

Puro Touko: So for weak scaling the grid size is determined from the number of processing units (cores for our case). So let's say you start with 2 x 2 and one core. Then the next measurement would be a 4 x 2 grid and two cores, then 4 x 4 grid and 4 cores, 8 x 4 grid and 8 cores and so on

Puro Touko: x-axis is the number of cores and the y-axis is the time taken

Xuan Binh: oh! so the grid size also must increase linearly proportional to the number of processors as well? Can a heat mpi runs rectangular shape?

Puro Touko: Xuan Binh said:

oh! so the grid size also increases proportionally to the number of processors as well? Can a heat mpi runs rectangular shape?

Yes, for weak scaling you should have (number of grid points/number of cores) = constant
I think it should be possible, the only requirement that the dimensions are divisible.

Xuan Binh: I understood now.

So let's consider two cases for baseline and hybrid

Xuan Binh: lets say I have Baseline mode

I will run
tasks [1,2,4,6,8,10,12,16,20,24]
grid size [100, 200, 400, 600, 800, 1000,1200,1600, 2000, 2400]

then I will measure the running time for 10 combinations, and plot them against the tasks

Puro Touko: Yes you would measure with (tasks=1,grid size=100), (tasks=2,grid size=200) and so on.

Xuan Binh: I understood. Now how about Hybrid mode?

Xuan Binh: Should I also vary number of tasks, or I keep it fixed as either 1 or 2?

Puro Touko: To make it easier to get the difference of MPI vs. MPI+OpenMP I would say keep the num of tasks fixed, I would suggest only a single task per node since the extreme is the interesting case

Xuan Binh: Thank you for your confirm. Okay one last question for me regarding sheet 5 is this
To run a restart with a certain number of iterations, use: srun ./heat_mpi - N_ITERATIONS, with - as input filename

Xuan Binh: I want to run heat_mpi restart instead of continuing. However I dont know what is input filename?

Xuan Binh: I dont want to run the bottle sauna

Puro Touko: You should have a bunch of save states that the program has produced, you can pass any of them as the input filename, (I have to check what the filenames are since can't remember them from the top of my head).
But anyways for performance measurements there is not need to restart so why do you want to restart?

Puro Touko: You could simply give the grid size and num of iterations as in the code_usage.md:
Default pattern with given dimensions and time steps:

```
srun <options> ./heat_mpi 800 800 1000
```

Xuan Binh: image.png

It is HEAD_RESTART.DAT?

Xuan Binh: I dont know... I think restarting makes it more authentic? :smiling_face_with_tear:

Puro Touko: Yes, it should be that file.

Xuan Binh: I did run the measurement for a few cases, but it is very fluctuating. I think I should report a mean as well like Task (3a)?

Puro Touko: The measurements should be pretty much identical with or without restarting, given that you run with enough timesteps that the initialization cost is amortized to the whole simulation.

Xuan Binh: So what could be your recommended grid size and time step? I choose now 4800 and 2000 timesteps

Puro Touko: Sounds good, maybe test with 2000 timesteps and with 4000 timesteps and if you get similar results then the number of timesteps should be adequate

Puro Touko: The reason for the fluctuations is probably because of the I/O, which especially on computer clusters with multiple users have many moving parts. Maarit gave some instructions have to turn it off:
https://pps23.zulip.aalto.fi/#narrow/stream/2149-Exercises/topic/sheet5/near/180542

Xuan Binh: Ahhh, I see now. So I should set int image interval to a very large number right?

Xuan Binh: Okay, I will need to stop bothering you for a while. When I finish Sheet 5, I will continue to ask you today or tomorrow with Sheet 3 and 4 again :sob: I

Puro Touko: Reporting the mean of your results is always great! (another possibility is to report the 90th percentile of your measurements)
And even better is if you have error bars around the mean representing the spread of your measurements. This is not required of you but good to keep in mind if you are doing bench marking in the future.
Anyways if you can't get the fluctuation to disappear than reporting the mean would be a wise thing to do

Puro Touko: Xuan Binh said:

Okay, I will need to stop bothering you for a while. When I finish Sheet 5, I will continue to ask you today or tomorrow with Sheet 3 and 4 again :sob: I

Good luck!

Xuan Binh: Wow thank you so much for the instructions!

Xuan Binh: Okay mister, I have finished sheet 5, now I haveSheet 3, Task 4.2, 4.3 and 6.3 left. I hope you would give me some directions for Sheet 3, please :smiling_face_with_tear:

Puro Touko: Hi,
On what specifically do you want guidance and did you read me earlier discussion about the grid splitting and neighbours?

Xuan Binh: :hurt: Hi Mr Puro

Xuan Binh: Let me think.... Actually Im all stuck at all assignments. so I find it hard to say which one I would continue.

Mrs Maarit claims Task 4.2 and 6.3 to be easy, but I do not think so.

Xuan Binh: If I guess, the prefix sum technique still preserves the original running time. However it only becomes faster given parallel components

Puro Touko: Well, easy is a relative term and what is easy for some is hard for others and vice versa

Xuan Binh: I have trouble understanding the long article of nvidia prefix sum since there's so many code and ideas. If you would be kind, is there a specific section on that website where I should read carefully? :sob:

Puro Touko: Focus on implementing the prefix sum on the GPU. It is okay at first if it is the slower naive version in the article but that would be a good starting point. After you have gotten that down try the more complex algorithm for the prefix sums

Xuan Binh: :thinking: oh!

What you means is, I should build first a working implementation of prefix sum on GPU.

Given a datay array of size N, returns an exclusive prefix sum in C++ on multiple GPU?

Puro Touko: That is for 4.3, for 4.2 you should be able to use your quicksort implementation for 4.1 (sorry thought you were talking about 4.3 since you mentioned the Nvidia article).
For the MPI quicksort you should find some example codes by googling, but the basic idea is that when splitting the array into high and low parts you should split the prosess also into processes that handle high and low parts. Then you recurse until each process has it's own subarray to sort and finally combine all of them together.
As a hint communicators and communicator splitting is useful for this

Xuan Binh: :smiling_face_with_hearts: Oh, I understood..
Regarding task 4.2 MPI, is it true that I should use lecture 4, one sided MPI to do this assignment? such as MPI_Put, MPI_Get, Broadcast, etc

Puro Touko: Using one sided MPI is a good idea, yes!
But of course you can do it with twosided communication if you want

Xuan Binh: Mr Puro, until when today will you become offline? I'm so scared :sob:

Puro Touko: since today is a normal workday I will be online until around 19:00-20:00

Xuan Binh: Oh no, I have to be fast. I have 5 tasks left. Okay, I will ask you soon :sob:

Xuan Binh: #SBATCH --nodes=2 #Use two nodes
#SBATCH --ntasks=4 #four tasks

Hi Mr Puro, I choose 4 tasks in quicksort_distributed.
Now I run my code and the output has something like this.

```
Not sorted at rank 3!!
Incorrect at rank 3!!
Is sorted at rank 0
Correct at rank: 0
Not sorted at rank 1!!
Incorrect at rank 1!!
Not sorted at rank 2!!
Incorrect at rank 2!!

real 0m0.965s
user 0m0.023s
sys 0m0.013s
```

So is it sorted correctly or not?

:smiling_face_with_tear:

Puro Touko: As long as it is not sorted at all ranks it is not sorted properly.
What you have is that it is sorted at rank 0, but not at other ranks

Xuan Binh: wait how could it be..... When it escape the quicksort call, all tasks run the same (is_sorted). If rank 0 believes it is sorted correctly, why dont other tasks see so...

Puro Touko: they have different copies of the data, right?

Puro Touko: With MPI all data of process is private to it by default

Puro Touko: So at the start each MPI process has a copy of the same data to be sorted. In the end each process should have a sorted copy of the data

Xuan Binh: :thinking: wait a moment... I thought that all process must be sorting in parallel, then merge. The data array should be common data?

Xuan Binh: FIrst I use MPI scatter to distribute data to each local data

Xuan Binh: wait, task 2 didnt ask me to use MPI to sort the array faster. It just asks that I use all tasks and sorts its own local data. Is this true Mr Puro

Puro Touko: No,
it is definitely faster to sort using more MPI tasks. Thinks about like this: you have two tasks 1 and 2. 1 sorts the lower part and 2 the upper part then they exchange their data. The sorting of upper and lower is done in parallel

Xuan Binh: So it is true that I will use each MPI tasks to sort 1/n_tasks portion and then use merge operation

Xuan Binh: Okay I need to use BCast to cast the sorted from rank 0 back to others :sweat_smile:

Puro Touko: Xuan Binh said:

So it is true that I will use each MPI tasks to sort 1/n_tasks portion and then use merge operation

That would follow merge sort, you are asked to follow quicksort

Puro Touko: You should still follow the idea of splitting the array into lower and upper when doing it with multiple processes

Xuan Binh: Wait.... I dont understand. Is merge operation allowed in this implementation of MPI quicksort?

Puro Touko: If you simply split the array into n parts, sort them and then merge, this is merge sort not quicksort right?

Puro Touko: To do it in quicksort manner, you should recursively split the array into lower and upper parts until you have n parts of the array and sort them in parallel

Xuan Binh: I divide it into 1/n_tasks portion, use quicksort to sort each, then merge them. It is hybrid quicksort-mergesort :sob: will you give me 0 point

Puro Touko: I will not give you 0 points, but you understand that it is not correct?

Xuan Binh: okay, I will try again. No merge operation allowed, if that what you wants and I must use recursion. However, it also means the n_tasks must be a power of 2 as well?

Puro Touko: I would say you are allowed to merge after sorting the portions, but because of the nature of quicksort the merging is trivial, the portions are already in relative order so simply append them together

Puro Touko: Xuan Binh said:

okay, I will try again. No merge operation allowed, if that what you wants and I must use recursion. However, it also means the n_tasks must be a power of 2 as well?

Does not need to be, but it is easier for power of 2

Puro Touko: Think about the splitting like forming a binary tree, it does not need to be a perfect binary tree

Puro Touko: Don't worry about it for now, start with the power of 2 first

Xuan Binh: Puro Touko said:

I would say you are allowed to merge after sorting the portions, but because of the nature of quicksort the merging is trivial, the portions are already in relative order so simply append them together

How can I understand this better?

So instead of dividing the array into 1/n_tasks like what I'm doing, you prefer it to partition the array using quicksort, and it only carries out the sorting when number of partitions = n_tasks.

Puro Touko: Yes you got it correct!

Xuan Binh: Hi Mr Puro, in the shell script of quicksort distributed, can I use nodes = 1? The template gives me nodes = 2

Puro Touko: you can use a single node for debugging, as long as it works with multiple nodes

Xuan Binh: I understand your idea, but now starting to code, I feeling stuck. I saw many implementations on google have merge. Was I wrong in understanding this parallel quicksort? :speechless:

Puro Touko: but they are not doing any splitting?

Xuan Binh: Okay I have no more time to ask you quicksort. I need to move to Task (3) now :smiling_face_with_tear:

Xuan Binh: Mrs Maarit claims it to be difficult, and I am reading it seriously right now again. In your opinion, should this exercise 3b be overly difficult? Can I tackle it within today? :smiling_face_with_tear:

Puro Touko: I would say exercise 3b is probably for the vast majority of students the most difficult exercise yes. It can take some time to wrap your head around what the formulas in the instructions mean and so on. I would say from a technical perspective 4.3 is probably the hardest, but in 3b there are a lot of concepts you might not be familiar with

Xuan Binh: :sob: I'm desparate to know the answer...

Puro Touko: the example solution for it?

Xuan Binh: Yes :smiling_face_with_tear: So by all means I must score nonzero point

Puro Touko: I am here to help so please ask for help, and @Rheinhardt Matthias, is the one who made the exercise so you can ask for his help, especially details on the grading if you want

Xuan Binh: this course is even harder than the other parallel course, unbelievable Mr. Puro :smiling_face_with_tear:

Now, it asks me to fill in many parts in the .c file. If I guess, these are the parts I must fill in? Additionally, I have to find out the parameters pased to find_proc and find_proc_coords as well?

```
const float u_x = ??? ;
const float u_y = ??? ;
const float c_amp = ??? ;

int find_proc(int ipx, int ipy, ???) {
// Implement finding process rank from coordinates ipx, ipy in process grid!
???
}

int* find_proc_coords(int rank, ???) {
// Implement finding process coordinates ipx, ipy in process grid from process rank!
```

```
???
}

   // Find neighboring processes!

   int domain_nx = atoi(argv[3]),              // number of gridpoints in x direction
       subdomain_nx = ???                        // subdomain x-size w/o halos
       subdomain_mx = ???                      //                  with halos

   int domain_ny = atoi(argv[4]),              // number of gridpoints in y direction
       subdomain_ny = ???                       // subdomain y-size w/o halos
       subdomain_my = ???                        //                  with halos

MPI_Win_create(/*???*/)

for (unsigned int iter = 0; iter < iterations; ++iter)
   {
       // Get the data from neighbors!
       ???
       // Compute rhs. Think about concurrency of computation and data fetching by MPI_Get!
       ???
```

Do you think is there any other things I miss out to fill in?

Puro Touko: You should also have code for creating the MPI datatypes and for timing (I would recommend first get the code working without timing and then worry about it)

Xuan Binh: :smiling_face_with_tear: Okay, at least in the first step, what I should fill in to make the code run and produce some feedback?

I create a shell script and module load openmpi and gcc as usual to run this .c script?

Puro Touko: I think you should fill at least u_x, u_y and c_amp for the code to be sensible

Puro Touko: and the halo_width, and subdomain sizes...
I would just recommend trying to first compile it (I think the compilation won't be successful if you don't add some things in) and add things to make it compile

Xuan Binh: I understood, I will try it now

In a brief sentence, would you help me explain what I should try to achieve? Is it we must obtain a numerical solution solving the advection equation to preserve the rate of concentration over time? I also see the comparison to analytic solution, is it given by the program or we must calculate by hand?

Puro Touko: You should achieve a numerical solution close to the analytical one. Sadly you have to code the comparison against the analytical solution yourself. (will probably in the future add the comparison to the analytical solution to the code skeleton so you don't have to do it yourself)

Xuan Binh: Where should I calculate the analytical solution in the code? :sneezing: Would you drop me some hint

Puro Touko: for a given timestep calculate the analytical solution at the same place you calculate the numerical one

Puro Touko: I would advice first getting the numerical solution up and running and then worrying about the analytical one

Xuan Binh:

```
    // Update field in data using rhs in d_data (Euler's method):
    for (ix = ixstart; ix < ixstop; ++ix)
       for (iy = iystart; iy < iystop; ++iy)
       {
          data[ix][iy] += dt*d_data[ix][iy];
       }
    t = t+dt;
```

Shoul I leave this intact Mr Puro?

Puro Touko: Yes, as it says it is the update in Euler's method (you take a small step in the direction of time derivatives) no need to change it

Xuan Binh: HI Mr Puro, is the for loop

```
for (unsigned int iter = 0; iter < iterations; ++iter) {
```

calculating the numerical solution? Should I calculate the analytical solution outside of this loop? and how can I actually calculate the analytical solution though? Is there any formula for it as I only see the advection equation

Puro Touko: So did you get the numerical solution working?

Xuan Binh: not really, I just want to be sure that I dont need to care about the analytical solution in this for loop :smiling_face_with_tear:

Puro Touko: Well for each timestep you have to have the analytical values to compare against, so you need to calculate it again for each timestep

Xuan Binh: Okay, so it should be looking like this, isnt it

```
        // Update field in data using rhs in d_data (Euler's method):
        for (ix = ixstart; ix < ixstop; ++ix)
           for (iy = iystart; iy < iystop; ++iy)
           {
              data[ix][iy] += dt*d_data[ix][iy]; (numerical solution)
             data_analytic [ix][iy] += .... (something here)

           }
       print average error between numerical and analytic solution at time t here
        t = t+dt;
```

Xuan Binh: oh, and additionally, how many nodes, tasks and threads should I use to run Assignment 3? :smiling_face_with_tear:

Puro Touko: Num of nodes and tasks doesn't matter, as long as your code doesn't depend on the exact number. The application is not multithreaded so only a single thread per process

Xuan Binh: Okay I understood :smiling_face_with_hearts:

Xuan Binh: Besides MPI_Get, am I expected to use any other type of MPI operations in the for loop? :smiling_face_with_tear: I think I should also use MPI_Win_fence as well?

Xuan Binh: Hi Mr Puro :smiling_face_with_tear:

```
[pe84:7240 :0:7240] Caught signal 11 (Segmentation fault: address not mapped to object at address (nil))
[pe84:7241 :0:7241] Caught signal 11 (Segmentation fault: address not mapped to object at address (nil))
==== backtrace (tid:   7241) ====
 0 0x000000000004d455 ucs_debug_print_backtrace()  ???:0
 1 0x000000000003afe7 __GI_____strtoll_l_internal()  :0
 2 0x0000000000037900 atoi()  ???:0
 3 0x0000000000400fb5 main()  ???:0
 4 0x0000000000022555 __libc_start_main()  ???:0
 5 0x0000000000400b89 _start()  ???:0
=================================
==== backtrace (tid:   7240) ====
 0 0x000000000004d455 ucs_debug_print_backtrace()  ???:0
 1 0x000000000003afe7 __GI_____strtoll_l_internal()  :0
 2 0x0000000000037900 atoi()  ???:0
 3 0x0000000000400fb5 main()  ???:0
 4 0x0000000000022555 __libc_start_main()  ???:0
 5 0x0000000000400b89 _start()  ???:0
=================================
srun: error: pe84: tasks 0-1: Segmentation fault
```

```
srun: launch/slurm: _step_signal: Terminating StepId=25916339.0
```

I encounter this error and dont know how to debug this. What I simply do is minimally add the int keywords to make all words defined. And also implement the two functions. I have not done the rhs and MPI_Get yet. Is this error expected as I have not write the MPI_Get?

Xuan Binh: right now, this is my compiling commands

```
module load openmpi/3.1.4
mpicc -lm -o advec_wave_2D_skel advec_wave_2D_skel.c
time srun advec_wave_2D_skel
```

Xuan Binh: advec_wave_2D_skel.c
This is my script, without the computation of MPI get or RHS computation. Can you help me :smiling_face_with_tear:

Puro Touko: To make debuging easier I would suggest first to compile with debug information on (add the -g flag)

Xuan Binh: It seems I got error on line 80

```
 3 0x0000000000401095 main() /scratch/courses/programming_parallel_supercomputers_2023/nguyenb5/sheet3/advec_wave_2D_skel.c:80
 4 0x0000000000022555 __libc_start_main()  ???:0
 5 0x0000000000400c69 _start()  ???:0
``
```

Xuan Binh: which corresponds to

```
nprocx = atoi(argv[1]); nprocy = atoi(argv[2]);  // process numbers in x and y directions
```

Xuan Binh: I have a feeling I have to pass these arguments in my command line?

```
time srun advec_wave_2D_skel -<4 arguments here?>
```

Puro Touko: Xuan Binh said:

I have a feeling I have to pass these arguments in my command line?

```
time srun advec_wave_2D_skel -<4 arguments here?>
```

Yes, but there needs to be 5 arguments (also the number of iterations)

Xuan Binh: Do you suggest some values? Please help me... :smiling_face_with_tear:

```
nprocx = atoi(argv[1]); nprocy = atoi(argv[2]); // process numbers in x and y directions
int domain_nx = atoi(argv[3]); // Number of gridpoints in the x direction
int domain_ny = atoi(argv[4]); // Number of gridpoints in the y direction
```

```
unsigned int iterations = atoi(argv[5]); // number of iterations=timesteps
```

Xuan Binh: and also, I expect nprocx x nprocy = n_tasks, isn't it

Puro Touko: Xuan Binh said:

and also, I expect nprocx x nprocy = n_tasks, isn't it

Yes

Puro Touko: Iterations does not matter that much, just take some number like 100, grid size maybe 1000 x 1000 and would advice to choose ntasks to split evenly so if you have a single node maybe like ntasks = 16 and nprocx = nprocy = 4

Xuan Binh: Okay mr puro, my code has now run okay. It outputs dt = ... without errors

Xuan Binh: What could be my next step? :cry:

Puro Touko: Sounds great! Then I would suggest to time the concurrency

Xuan Binh: I haven't done the MPI_Get and RHS yet... Should I do this first?

Puro Touko: Oh, sorry yes

Puro Touko: Do the time integration first

Puro Touko: So in each timestep you have to do the halo exchange, code that first

Xuan Binh: Okay mr Puro, before the Euler update loop, this is my skeleton for the MPI get

```
    MPI_Win_fence(0, win);

    // Fetch the halo data from each neighbor using MPI_Get
    MPI_Get(recv_halo?, subdomain_nx, MPI_FLOAT, neighbor_rank?, displacement?, subdomain_nx, MPI_FLOAT, win);
    Some more MPI_Get here

    MPI_Win_fence(0, win);
```

Xuan Binh: Do you think Im heading in correct direction? :face_holding_back_tears:

Puro Touko: Yes, but please remember to use MPI datatypes you have defined yourself

Xuan Binh: If I guess, there are 4 neighbors, so there are actually 4 MPI_Get for Euler methods. Is this true?

Regarding the 2nd order scheme, is it optional or I must implement it?
And if I use the 2nd order, does it mean there are 8 MPI_Get lines to update the center red cell?

Puro Touko: Use the 2nd order scheme. For it the number of neighbours is the same, 4, but you need values from only two of them

Xuan Binh: Hi mr Puro, I think I got NaN for my values... Do I need to pay attention to the boundary condition? :sob:

Xuan Binh: Mr Puro... :sob: Where should I handle the corner or border case

Puro Touko: Hi I have to wrap it for the day, but the grid is periodic as you correctly asked in the chat so your halo exchange should reflect that

Xuan Binh: Nooo, you are about to be offline. Please save me one last time before you become offline :sob:

Xuan Binh: 1) I must not change the rhs function?
2) If yes, I must change these three lines for the border condition?

```
int xrange[2] = {halo_width, halo_width + subdomain_nx};
int yrange[2] = {halo_width, halo_width + subdomain_ny};
rhs(xrange, yrange, subdomain_my, data, d_data);
```

Puro Touko: 1) yes, don't change it
2) the xrange and yrange are the ranges where you apply the rhs function, they have nothing to do with the boundary condition

Xuan Binh: Lastly, besides MPI_Get and the MPI_Win_fence are the only MPI functions I would need in this exercise, and also MPI_Reduce for accumulating the errors, I wouldn't need any other MPI functions, isn't it? :smiling_face_with_tear: I must complete this assignment without your help from now tonight

Puro Touko: You need to also create the MPI datatype you use be it subarray or a vector datatype or something else