Author's Point-by-point response to the reviewer

1. Figure 1 is a nice-looking figure, but I don't believe it adds anything to the discussion and is in fact misleading as it isn't really saying anything of substance at all. It doesn't reflect the point made in the text: "the appropriate programming model becomes a difficult task as not all of them support all types of accelerators" (lines 67-68), but the diagram just shows a wall with each brick representing a different programming model and arrows going indiscriminately to different hardware. It would instead be useful to show which programming models work on which hardware, either through a different diagram or a table.

We agree that the added value is not high enough and therefore removed it.

2. Figure 2 likewise looks very nice but doesn't really contain any substance about how the ICON code is currently and how ICON-C will change the code structure and apply the different programming models. I would rather for the authors show code snippets or pseudocode to illustrate how the modules could be ported to the new architectures using the different programming models, and how would this be done, e.g. manually or using a code writer? Other models, such as the LFRic weather and climate model (https://www.metoffice.gov.uk/research/approach/modelling-systems/lfric), are using tools such as PSyclone (https://psyclone.readthedocs.io) to try to achieve performance portability of the Fortran source code on different architectures.

We changed both illustrations to respect the given feedback. The switching in the current monolithic design is handled by various preprocessor macros for the respective architectures as illustrated in the updated left hand side of the figure. In contrast, the new ICON-C design now emphasizes that only one module of each kind is needed.

3. Figure 3 is quite confusing. The ideal strong scaling curves are very difficult to see as they are a light grey colour (the same as the gridlines behind). The point type used (left- and right-facing triangles) are difficult to differentiate and the fact that the colours are slightly different for the global and nested domains, but not different enough, makes it hard to unpick this. The different line styles are also not discussed. The authors could consider making the plot much larger, using very different point styles, mentioning the different line styles, and perhaps separating out the global and nested results into different sub-figures.

We revised the plotting style of this figure. Every line has now its distinct, unique combination of style and markers. The line style encodes the processor type while the markers denote the domain. Both are explained in the legend. Colors are used to emphasize the different line styles such that they identify the different processors in the same way as in the roof line plots later on.

4. Figure 4 is very interesting given the results in Figure 3 and the core numbers in Table 2. I would suggest highlighting the number of cores in the discussion around GPUs for Figure 3. It might also be helpful to highlight on Figure 4 the number of cores for each hardware type configuration, so it can be seen when the hardware becomes underutilised.

We modified the figure caption motivated by the reviewers comment and adjusted the figure style to match the previous figure. The new caption provides more explicit information on the compute capability of each respective processor for a single MPI process. The number of cores are discussed in line 186–205.

5. For Figure 5 I'm a bit unsure if the timing for B1 as given in the example will include to the end of the K41 kernel, i.e. outside the end of B1.

To clarify this, additional vertical lines were added in the top area. It should now be clear, that $K(B_1)$ extends from the beginning of $K3_1$ to the end of $K4_1$ and $K(C)$ from the beginning of $K3_2$ to the end of of $K1_2$.

6. The plots in Figure 6 are very hard to see as the points are large and the lines are close together and overlapping. The points are also all clustered around a small section of the graph, but the scale is much larger in X and especially Y, mainly to include the legend. I would suggest plotting each point on its own graph, making a large multi-figure plot, and zooming in as much as possible onto (0.1:100, 1:50,000) ranges to allow the relevant areas to be seen as clearly as possible. I would likewise do the same for Figure 7 by combining Figures 6 and 7 in a single multi-panel plot.

We agree that the previous versions made it hard to see the differences. But since we also wanted to express where the results are compared to the shape of the rooflines, we decided to add a zoom-in of the interesting region for each plot.

7. Figure 8 is also really interesting. Do you know the energy usage from the CPU runs?

We have no energy metrics for the CPU runs as we had no conceptual approach to measuring the energy consumption of a simulation on the CPU nodes at the time of the studies.

8. Line 17: I don't believe "System" should be capitalised here.

Corrected as suggested

9. Line 26: The term GPU is used without being defined (this occurs on line 40). Also, the discussion here is around x86 hardware being combined with GPUs, Vector, or ARM systems. Note that superchip hardware, such as NVIDIA's Grace-Hopper (https://resources.nvidia.com/en-us-grace-cpu/grace-hopper-superchip) is an ARM-GPU system, i.e. there is no x86 CPU, where the CPU itself is instead ARM-based.

The definition of GPU was put after the first use of the acronym as suggested. Regarding ARM-based systems, we are referring to the A64FX processor of the Fugaku supercomputer, where the CPU is ARM-based.

10. Line 345-6: I think this sentence needs to be rephrased: "Not only the ICON community is currently on the way to using current and upcoming Exascale systems for high-resolution simulations." - do you mean something like "ICON is not the only model that is currently on the way to using current and upcoming Exascale systems for high-resolution simulations."

Corrected as suggested

11. Line 356-7: Perhaps I'm missing something, but wouldn't halving the horizontal resolution result in a 4-fold increase in necessary resources, rather than 8-fold? I'm assuming that the level structure remains unchanged.

Halving the horizontal resolution results in an 4-fold increase in grid points and requires doubling of the required time steps as noted in Section 3 (lines 170–173 of the manuscript). Thus the required resources increase is 8-fold.

Additional editorial changes

- As requested during last submission the reference list is now compiled according to the GMD standard.