General comments

This paper designed a new framework for an air quality model to adapt the new Intel architecture to improve the computing performance, which is very useful for HPC, model developer. I therefore fell that the manuscript is suitable for Geoscientific Model Development and Major Revision is needed. Some comments and suggestions follow below. I hope this paper can be more useful through further revisions.

Response: Thank you for your encourage and sharing the time in this manuscript. The detailed responses to the comments are given point to point as followed.

Specific comments

1. Writing

The writing is not good. I think the authors should look for a native language person to polish it. Some sentences are hard to understand, especially in 3.3 Performance Test and figure captions.

Response: The authors appreciate your precious comments. We would take some measures, including inviting a native speaker, to improve the language of the transcript, especially the part in section 3.3 mentioned by the reviewer.

2. Experiments and analysis

a. The authors did not give the information about the floating precision of the model. Double precision, or single precision? The performance should be difference, I think. If possible, the authors had better show the results of model with both single and double precision.

Response: Currently, the floating precision of the model is single-precision. That is a good idea. We will follow your advice and conduct the relevant experiments to test the difference of the performance with single and double precision.

b. In this paper, the authors only give the results with one node. How about the model's captions? Is the performance (speedup) on KNL still better than CPU?

Response: The authors appreciate your constructive comments. As the reviewer mentioned, we only conducted the single node tests currently. We will follow your advice to do some tests to investigate and compare the model's captions on CPU and KNL.

c. In performance test, authors only test the MPI and OpenMP separately. It's very interesting that the performances are almost same. Can authors explain that? I just wonder how about the computing performance by using MPI/OpenMP hybrid parallel method?

Response: The authors thank your for your precious comments. The MPI and OpenMP in our code were used for the same parallel segment in CBM-Z module, their acceleration ratio is similar at the algorithm level. And we only took the tests on single node that there is no overhead of MPI on message passing cross-node through the network, which led to the similar performance of OpenMP and MPI. We will use the performance analyzing tools like Intel Trace Analyzer and Collector (ITAC) and Intel Vtune to further analyze the codes.

The MPI and OpenMP are both used for the main loop currently in the CBM-Z module, and it requires more time and specific technological details for us to add MPI and OpenMP into different levels of parallel segments of CBM-Z module. In this study, we would mainly focus on the exiting codes and results, and the hybrid parallel method could be a direction for our future work.

d. It takes 539.86s with one CPU core while 4481.10s with one KNL core. The authors blame the worse performance to lower frequency of KNL. It's difficult to understand because frequency difference between CPU and KNL is significantly and much less than the computing difference. Other factors, such as memory bandwidth, should also contribute the computing difference.

Response: The authors thank for your comments. As the reviewer mentioned, other factors like memory bandwidth could also contribute to the gap of the performance. We will conduct some relevant experiments and use the tools like Intel Vtune and ITAC to investigate the potential reasons for the phenomenon.