Dear Reviewer,

We would like to sincerely thank you for your second thorough and constructive review of our manuscript (An Effective Communication Topology for Performance Optimization: A Case Study of the Finite Volume WAve Modeling (FVWAM)). Your insightful comments have been invaluable in improving the quality of our work. Please find below our detailed responses to each of the comments you raised.

Sincerely,

Renbo PANG, on behalf of the co-authors

The revised paper is greatly improved. I especially welcome the addition of Open MPI results. I believe the article is nearly ready for publication in GMD. I have just a few minor comments.

Reply： Thank you very much for your positive comments!

Minor comments 1: *Following my suggestion, the authors renamed grid IDs to cell IDs in the text. However, figures 4 and 5 still refer to grid IDs. This needs to be fixed.*

Reply： The “grid IDs” in Figures 4 and 5 have been replaced by “cell IDs”.

Track changes: Please refer to Figures 4 and 5 between Lines 208 and 209 in the revised manuscript with tracked changes.

Minor comments 2: *In Section 4.1 the authors show new performance results of halo exchange with Intel MPI and Open MPI in small-scale parallel experiments. They compare the graph-based and point-to-point implementations for each MPI library. They conclude that there is a large speed-up from using the distributed graph topology only with the Intel library. However, I don’t see any discussion of the fact that the point-to-point method using Open MPI is significantly faster than the same method with Intel MPI for inter-node communication. This suggests that the large speed-up from using the graph implementation with the Intel library is partly because the point-to-point implementation is performing poorly. I think this should be mentioned in the paper.*

Reply： Discussion between Open MPI and the Intel MPI have been added as following in the manuscript. Different MPI implementations may utilize varying communication patterns, process binding and scheduling

strategies, buffer management techniques, and support for remote direct memory access (RDMA). These differences can lead to performance disparities among MPI implementations. For intra-node communication, both the distributed graph communication topology and the point-to-point communication method using the Intel MPI library significantly outperformed their counterparts implemented with the Open MPI library. Liu et al. (2003) stated that the Intel MPI library enhances performance by leveraging shared memory mechanisms for intra-node communication. For inter-node communication, the distributed graph communication topology exhibited similar performance with both the Intel MPI and Open MPI libraries. However, the point-to-point communication method with the Open MPI library demonstrated significantly superior performance compared to the Intel MPI library. Rashti (2010) identified high overhead for small messages with the Intel MPI library, which was attributed to semantics miscorrelation between MPI and the user-level library. A further potential cause could be the lack of RDMA support or poor performance with buffer management for small messages in the point-to-point communication method using the Intel MPI library.

Track changes: Please refer to lines 286-297 in the revised manuscript with track changes.

Minor comments 3: *The added plots of communication data volume are interesting, but, for me at least, there are too many of them. I don't think it is necessary to show the process versus process plot for every process count. Most of them are very similar and don't add much. It would be sufficient to show them only for the smallest and the largest number of processes.*

Reply： Only the smallest, middle (for aesthetic purposes to maintain an even number of images), and largest numbers of processes were retained; the others have been removed.

Track changes: Please refer to Figure 6 between Lines 262 and 263, and Figure 7 in Page 26 in the revised manuscript with track changes.

Minor comments 4: *In Table 2 it would be good to include compilation options again. Otherwise, it looks like only the Intel runs used compiler optimization. Also, why is the version of NetCDF included in Table 1 ? I don't see how it is relevant, considering the paper doesn't discuss I/O performance.*

Reply： As suggested, the version of NetCDF has been removed.

Track changes: Please refer to Table 1 in Page 11 in the revised manuscript with track changes.

Reference:

Rashti, M. J.: Improving Message-Passing Performance and Scalability in High-Performance Clusters, Ph.D. thesis, Queen's University, 2010.

Liu, J., Chandrasekaran, B., Wu, J., Jiang, W., Kini, S., Yu, W., Buntinas, D., Wyckoff, P., and Panda, D. K.: Performance comparison of MPI implementations over InfiniBand, Myrinet and Quadrics, in: Proceedings of the 2003 ACM/IEEE conference on Supercomputing, p. 58, 2003.