We would like to thank the anonymous referee's constructive comments. An item-by-item response to the comments is presented below. The manuscript was also revised according to the referee's technical suggestions. As a consequence, in our opinion the revised manuscript is improved with respect to the original one.

*==

Responses to the second anonymous referee's comments:

Anonymous Referee #2
Received and published: 6 May 2015

Summary:
This paper presents the GPU porting and optimization process for a sub-process of the WRF model. The authors describe the mathematical background of the existing model sub-process, then delve into the wide variety of optimizations made to achieve performance from the GPU version of the code. The authors compare the results of each successive optimization to the performance of the original CPU code.

General Comments:
This paper does a good job presenting the various optimization techniques used to achieve a clearly excellent speedup result. The optimizations used are described well, and will likely be useful in other domains beyond the scope of accelerating the YSU PBL scheme.

Thank you for the positive remarks.

* ==

Questions:
(1) Is there a specific reason that the Yonsei University PBL scheme was chosen for acceleration instead of any other scheme (none of which were named)?

We have implemented several schemes of the Weather Research and Forecasting (*WRF*) model. Yonsei University PBL (YSU PBL) scheme is one of them. Our goal is to have a GPU-based accelerated WRF model.

(2) Is this the most popular model, or perhaps the most amenable to GPU acceleration? I believe some discussion of other schemes may be warranted, at least as motivation for why the YSU scheme was chosen.

The physical feature of no interaction among horizontal grid point (i.e., *i* and *j* in the code) makes all schemes of WRF physical (not dynamical) models very favorable for parallel processing. Among all PBL schemes, YSU PBL is one of the popular schemes used by WRF users, and thus we first chose to work on this scheme among all PBL schemes.

(3) The paper describes a direct mapping from the benchmark dataset's spatial grid size to the implementation's use of thread blocks. Will the accelerated code be easy to apply to other domains or resolutions and other datasets?

Yes. Our code is designed for any kind of data domain and resolution so that the users do not need to adjust the code. The users only need to provide the data dimension, i.e., (i, j, k), where (i, j) is the horizontal grid-point dimension while k is the vertical levels. For example, in our case, (i, j) = (433, 308) and k = 35.

(4) Has some other test dataset been used to examine speedups for a different test case?

For this scheme, we have not tried some other test dataset. Nevertheless, it is expected that the speedup should be about the same regardless of which dataset or data size is used.

(5) Is this work intended to be incorporated in some future release of WRF?

Yes.

(6) Is there possibly a timeline for when an accelerated (or partially accelerated) version of WRF is available?

This will depend on the funding and man power though we aim for finishing the implementation of all schemes of the WRF model as soon as possible.

(7) How much does the impressive speedup obtained for the PBL scheme improve the performance of the full WRF model?

Based on those schemes that we have finished the GPU-based implementation, it was found that different scheme has different speedup. Since we have not completed all schemes yet, at this moment it is hard to examine how much this GPU-based YSU PBL scheme would improve the performance of the full WRF model.

(8) It would be good to see some timing data for the full model with the accelerated PBL scheme incorporated.

Once all schemes of the WRFL model have been implemented on GPUs, we will definitely investigate the performance of the GPU-based WRF model with the accelerated YSU PBL scheme incorporated using timing data.

* ==

Technical Corrections:

Page 8033, line 18: "GPU-accelerated longwave radiation scheme of the rapid radiative transfer model for general circulation models"

Page 8034, line 8: "which is one of the physical models in WRF."

Page 8041, line 12: "built in the WRF model."
Page 8042, line 15: "The driver, in the C language,"
Page 8042, line 20: "into the memory of the CPU."
Page 8042, line 21: "From the viewpoint of CUDA Programming,"
Page 8043, line 10: "contiguous data, and are aligned in memory."
Page 8043, line 24: "Three major reasons for doing this in this way are"
Page 8043, line 26: "CUDA C programs in a short time."
Page 8044, line 3: "accross the entire US. The WRF domain is"
Page 8045, line 6: "one way to do this is to call"
Page 8045, line 20: "to a parallel GPU basis in the next section."
Page 8046, line 6: "one CPU core of an Intel Xeon E5-2603."
Page 8047, line 22: "to the structure of the WRF model,"
Page 8049, line 3: "per thread at 63."
Page 8049, line 15: "make execution more efficient."
Page 8049, line 16: "of the GPU architecture."
Page 8051, line 19: "In the plot, x starts from value"
Page 8062, Table 6a: Column headings need some label relating to what they represent, as opposed to just the description in the caption.
Page 8063, Table 6b: Same as for Table 6a, needs column headings.
Page 8076, Figure 12: Include labels for the x-axis of the graph, instead of just the range mentioned in the paper.

Thank you for the technical corrections. We have made these corrections marked in "red" color in the revised manuscript that has been submitted to the GMD journal. In addition, Fig. 12 has been also modified with adding the x-axis labels for the graph (see below).