Reply to Anonymous Referee #2.

We would like to thank referee #2 for taking the time to review our paper and for the thoughtful comments. We will reply to each comment individually below.

Specific comments:

1. The abstract (as well as some other parts in the context, such as P3 L14~L15, P6 L14~L15, and P11L31) mentions that "OASIS3-MCT_3.0 is the latest release and includes the ability to couple between components running sequentially on the same
set of tasks". It seems contradictory to P6 L24~L25 that "Each task will be associated with only one executable and one component in any application", which indicates that components cannot share any task. According to the API of "oasis_init_comp", I think the statement in P6 L24~L25 is true.

We have clarified the sentence in the introduction to "OASIS3-MCT_3.0 extends the ability to couple components running concurrently and adds support for coupling within a component for grids and fields defined on overlapping or partially overlapping sets of tasks, such as between physics and dynamics modules within an atmospheric model or to and from a model I/O module." We have clarified the description a bit in section 2.5. In particular, we have updated the first sentence to be " The ability to couple fields within one executable running on partially overlapping tasks was added in OASIS3-MCT_3.0". We have also added a sentence, "While OASIS3-MCT supports both single and multiple executable configurations, the coarsest level of concurrency in the system is the component." In the conclusions, we modified the sentence to " OASIS3-MCT_3.0 also provides new capabilities to couple fields within a single component running on concurrent, overlapping, or partially overlapping processes ". The reviewer makes a good point that we were implying that components could run on overlapping tasks and that's not true and that has been fixed in the text.

2. P1 L15~L18, P6 L18~L19, P6 L25~L27 and P12 L1~L2 may indicate that that there can be two different decompositions of the same grid within the same component and these two decompositions can have different subsets of the tasks (processes). To achieve this capability, the API "oasis_def_partition" has been extended with an addi- tional parameter "name". When I read the user manual at the first time, I guested that "name" means the name of the grid. After a careful consideration, I think that "name" should be the keyword of a decomposition but not the name of the corresponding grid, which means that the "name" corresponding to two different decompositions of the same grid within the same component should be different. If that point is true, please clarify it.

That is correct, the name associated with the "oasis_def_partition" call is the name given to the parition, not to the grid. We will clarify in the user guide.

3. The ability to define grids has been mentioned several times in the paper. What does it mean when only the API for writing grid data into files are introduced in the user manual. According to Figure 2, is the grid defined implicitly in the definition of decomposition?

The grid is something that does not depend on the decomposition and defines the grid center, corner, area, and mask information. At run time, OASIS reads this grid in a file that can be either produced by the user before the run or written through the API by the model. A partition is specific decomposition of a grid in the model. We have removed figure 2 from the revised draft as this better fits into the user guide and we will update the user guide to clarify.

4. Compared to OASIS3, OASIS3-MCT_3.0 have a new capability of pre-defined mapping files. After reading the paper as well as the user manual, it is still unclear for me that how to make OASIS3-MCT_3.0 know which mapping file should be used for a specific set of coupling fields (for example, users may want to use bilinear algorithm for state fields and use conservative algorithm for flux fields when coupling fields from an atmosphere model to an ocean model). Is there any restriction when users using the pre-defined mapping file. Concrete examples are welcome for this new capability.

We will clarify this information in the user guide. For a given entry in the namcouple file, the namcouple keyword MAPPING specifies the mapping file for those coupling fields. Each coupling field can be associated with a different mapping file rather arbitrarily and each mapping file can be generated via different algorithms.

5. P7 L28~L29. It is interesting to know how to make the puts non-blocking. In MCT, the data sending is blocking for example with the MPI_wait, which indicates that such
MPI_wait should be disabled for the non-blocking puts. It seems that OASIS3-MCT_3.0 does not use another MPI_wait out of MCT. So, one interesting question here is that how OASIS3-MCT_3.0 guarantees the puts constantly non-blocking (for example, we encountered the case that MPI_Isend was blocked when we sent a large message or many small massages) and how OASIS3-MCT_3.0 achieves safe non-blocking puts (for example, how to guarantee that next puts do not flush the data of previous puts in memory buffer).

MCT supports non-blocking MPI. The reviewer is correct that at some point, MCT will execute an MPI_Wait for a non-blocking MPI_ISend. On the put side, this happens before the next put of the same data at the next timestep. We define this as non-blocking MPI because the model does not wait for the actual put to occur and

the model can continue to advance. In fact, the put is only non-blocking in the sense that it can be only one coupling period ahead of the get at the most. While on the get side, the MPI is blocking at the time of the get. We have clarified the text in section 2.5 to reflect this information.

6. P6 L10~L11 states that "The opt option will however be bit-for-bit reproducible if the same number of processes is used between different runs". Given the same number of processes, bit-for-bit results may fail to be reproduced if the decomposition changed.

The reviewer is correct that if the decomposition changes, the sum will not be bit-for-bit reproducible. We have updated that sentence as follows, " The *opt* option will however be bit-for-bit reproducible if the same number of processes and decomposition are used between different runs ." We have also updated the conclusions.

7. One suggestion regarding Section 2.4 is that the opt option can use higher-precision of floating-point calculation to achieve faster bit-for-bit identical reduction. For example, using REAL8 when the coupling fields are REAL4 and using REAL16 when coupling fields are REAL8.

We have significantly revised section 2.4 to include some preliminary results of three new global sum algorithms including the algorithm suggested by the reviewer that are currently in the development version of OASIS3-MCT and expected in the OASIS3-MCT_4.0 release. The global sum calculation implemented in OASIS3-MCT_3.0 needed significant revision as indicated in the earlier version of the paper and this has already been undertaken.

8. Some results in Table 4 seem strange to me. Why the time for <10 fields, 10 couplings> is obviously smaller than 10 times of the time of <1 field, 1 coupling>? Why <10 fields, 1 coupling> is not much faster than <10 fields, 10 couplings>? The most significant reason may be the MPI message size of <1 field, 1 coupling> is big because the two components have similar decompositions and the core number is small relative to the big grid size. Given the same core number, more test cases with smaller grid size and different decompositions between the two components are welcome.

We have merged and updated the results in table 3 and 4 and added some new information. We have added a barriered ping pong time to compare with an unbarried time. This provides additional insights into the results that were not available in the initial version of the paper. In particular, 10 fields, 10 couplings is fastest in the unbarriered ping-pong time because it seems the amount of work that is overlapped between coupling and mapping is highest in that case. That case has the highest performance degradation when the send and mapping are barriered and

the mapping time of the 10 fields, 1 coupling is faster. These issues are now discussed in the paper in section 3.4.

9. The year of the first reference should be 2008.

We have changed 2009 to 2008, thanks.