34. page 2, around line 20: the discussion focuses on performance aspects only. Mention of the implications for energy/power use would also be useful here, as the focus is on extreme-scale systems.

    Fixed, see page 2, line 18

35. page 3, line 12: “how it is achieved without increasing data movement” should, I think, be changed to “and how it is achieved with minimal impact on data movement”. This point is discussed further below [*].

    [*] The following point is my main concern with the paper as it stands. I believe the paper would benefit by being clearer on the use, and limitations, of shared memory threading to implement the concurrent execution of the radiation and the rest of the atmosphere model. It is clear that for a given multi-core processor there will be limits on the number of MPI processes per node and the number of threads each model may use within an MPI process without incurring potentially expensive data movement between caches. The example results given are based on two threads for each model. It would be good to make clear the rationale for this choice. Here are some thoughts on this and suggestions for possible changes which might help achieve this. Currently, the use of threads for executing the radiation in parallel with the rest of the atmosphere is described as not incurring any communication costs. While it is true there will be no MPI communication incurred between cores running the atmosphere and cores running the radiation, I believe there may well be some extra remote data accesses (i.e. cache misses) incurred between cores running the atmosphere and cores running radiation. The magnitude of this effect is, of course, architecture dependent and also depends on the number of threads used for each model and the mapping of the threads to cores. The results presented are for AMD Interlagos processors with two threads used for the atmosphere model and two threads used for the radiation model (within each MPI process). The Interlagos processor chip consists of eight 2-core modules. Two threads executing on the same module share an L2 cache. All 2 core modules share a large L3 cache. So, if one atmosphere and one radiation thread share a module, they can share data in the L2 cache (as well as the L3 cache). If two Atmosphere threads share a module and two radiation threads share a module, they will communicate through the L3 cache, which is more expensive in terms of cycles to access. If threads are on separate processor chips, there will be (even more expensive) data movement within the shared memory node.

    The cache behaviour in either of the above cases is likely to be different to that of a single thread running first the atmosphere and then the radiation. If more than two threads were used for each model, some sharing would have to take place via the L3 cache. Total thread numbers are clearly limited by the core count of a shared memory node. I would suggest to the authors that some clarification of these issues be made. For example: - in Figures 3 and 4, the images depicting MPI and OpenMP could be re-drawn to illustrate the relationship of threads within MPI in each case. In Figure 3, this would simply show multiple threads in an MPI task and multiple MPI tasks. In Figure 4, MPI tasks could be shown with both atmosphere and radiation threads or with ocean threads. For Figure 4, this might be something like: [AARR] [AARR]... [O] [O]... (where [] here represent MPI processes and letters represent threads and their models) - In Section 3 (perhaps?), a brief description of a multi-core processor (like the Interlagos) could be given along with the implications of thread-to-core mapping. This description would help to explain the benefit of using OpenMP to exploit parallelism between the atmosphere and radiation models (i.e. no MPI communication) and pave the way for a discussion of the potential for sharing data between caches in the specific configurations presented in the results section.

The concurrency software design was formulated to ensure the radiation component was truly independent from the remaining atmosphere components with a single data synchronization point (copy) at the end of each time step. The data copying, itself implemented with OpenMP threading, can be completely removed via index flipping, but this might result in non-local accesses or extra cache invalidations - so a simple data copy is preferable. Additionally, each thread immediately calls a subroutine (either atmosphere or radiation) and is working with either explicitly blocked data structures or pseudo-blocked data created via copy-in variables. So not only are the atmospheric and radiation components completely independent, but within each component the data for each thread is isolated as well. If a radical paradigm shift in computing occurs, this forethought in the design allows us to quickly re-cast the radiation threads as separate MPI processes with a needed MPI communication synchronization point replacing the current data copy. See also (5) above.

36. page 5, Figure 1: The role of A_t should be depicted in the top figure (to be consistent with equations 1 and 2), I feel. Also, in many models, the atmosphere is executed on more processors than the ocean (because it scales better). Is this diagram consistent in this respect with the FMS model being described? Also, the bottom figure in Figure 1 implies that the ocean is executed on fewer processors in the concurrent set up. Is the intention to simply show a deployment utilising the same number of processors in total? If so, that should be made clear in the caption and text.

The relative balance of PE count between atmosphere and ocean depends on many factors, and within the FMS system, we have examples of both. However, in Figure 1 this is mostly schematic, indicating that some PEs may idle in serial processing.

We have adjusted the numbering of the component timesteps to be consistent with Eqs. 1-4.

37. page 9, line 2: “chosen to offer optimal load balance” could be extended to “chosen to offer optimal load balance and data sharing”, for example. [I generally have a concern over the use of the work “optimal”, which has a formal sense of “provably best”. The word “good” might be better unless the load balance is provably optimal?]

Fixed, see page 10, line 8

38. page 11, line 9: The above arguments are tied up with the statement that “All runs use the optimal processor/thread layout for a given PE count”. Some explanation about what this layout is and how it was chosen could be added.

Done, see page 14, line 9

39. page 1, line 4: I suggest changing “based on marginal increases in clock speed” to, for example, “based on, at best, marginal increases in clock speed” since it is likely clock speeds may decrease in future in some systems.

Done, see page 1, line 3

40. page 1, line 14: Define the acronym CCC here.

Done, see page 1, line 16

41. page 1, line 15: is a little ambiguous about what is running in parallel ("and all other atmospheric physics components". I would suggest making it clear that there are only two concurrent components (i.e. not all "other atmospheric components" are executed in parallel with each other!

Done, see page 1, line 18

42. page 2, line 3: perhaps provide a reference to the IPPC assessments.

See page 2, line 14.

43. Section 2: page 4, line 4: needs a closing bracket after "example".

Done, see page 4, line 31

44. page 8, line 24: "Individual" should be "individual".

Done, see page 9, line 29

45. page 9, Figure 4: In this figure, the Land and Ice models are shown as executing concurrently but this is not mentioned in the text. This should be explained (or made consistent with Figure 3).

Fixed. It should indeed have been consistent with Fig. 3. See new Fig. 4.

46. page 10, line 9: "that that" should be "that".

Done, see page 11, line 14

47. page 10, line 11: This sentence would benefit from having a reference added.

Done, see page 11, line 16

48. page 10, line 13: Expand the acronym GPCP here as a definition.

Done, see page 11, line 19

49. page 10, lines 15-17: The point here is, I think, that this result is counter intuitive. If that is correct, it would be worth stating.

As noted above in (25) and (28), the differences between the runs are small and within the bounds of the tuning process. Differences should not be over-interpreted as they are likely to vanish upon tuning. See page 14, line 2 also.

50. page 11, line 3: "less expensive as..." should be "less expensive as the...".

Done, see page 14, line 1

51. page 12, line 4: the figures for processor count and SYPD given in this line are rounded versions of those in Table 1. Those in the previous sentence are not rounded. Please use the precise figures for consistency.

    Done, see page 14, line 21

52. page 13, line 18: It would be worth giving the definition of CCC again here to remind the reader.

    Done, see page 15, line 20