General Comments

This paper introduces the new version of the coupling software OASIS and its latest revision OASIS3-MCT_3.0. It describes in detail the most important improvements and new features of this version. In addition, it provides performance data relevant for users of the software.

It has a clear structure and is written well. It gives users of older versions of the software a good understanding of the changes and helps to decide whether to switch to the latest version or not. For developers of other coupling solutions this paper gives an interesting insight on how the current version of OASIS works.

After some modifications and clarifications regarding the presented performance results, I would recommend this paper for publication.

Specific Comments

If you are not familiar with coupling software in general or with OASIS is particular, some parts of the paper may be difficult to understand, due to usage of domain-specific terms and concepts without further explanation for example "hub coupler" in abstract, “top-level driver” in introduction, the terms “source” and “destination”, “MCT router” in 2.1 General Architecture, or “CONSERV transform” in 2.4 Conservation. Depending on the target audience this might be an issue.

In the paper you talk about OASIS3-MCT and its improvements compared to older OASIS versions and about its latest revision OASIS3-MCT_3.0 in particular. However this is not reflected in the title of the paper. It implies that the paper is mainly about OASIS3-MCT_3.0.

You use lower and upper case when referencing figures or tables. This should be consistent.

“2.5 Concurrency, Process Layout, and Sequencing”
I do not see why there is a need to differentiate between different executables. Since each MPI process only has a single component, shouldn’t it be enough to start the differentiation at the component level? This might reduce the complexity of this paragraph. Or would there be any difference if comp2, comp3, and comp4 were run on three individual executables?

The main conclusion of section “3.3 Interpolation” is that the default option of performing the mapping on the processes of the source component might not always be the best choice and that explicitly setting OASIS to do it on the processes of the component with the most resources can deliver better results. However, to draw this conclusion the presented test cases and diagrams seem to be overly complicated. Since the mapping is done based on a “simple one-dimensional” decomposition, the performance should be independent of the grid types being used. Therefore you could draw the same conclusion from a table similar to the following one (only showing the results for a one directions data exchange), which I think is much easier to understand:

# src cores	# dst cores	Mapping on src		Mapping on dst	
		transfer	mapping	transfer	Mapping
24	336	* s	* s	* s	* s
180	180	* s	* s	* s	* s
336	24	* s	* s	* s	* s

In the discussion of section “3.3 Interpolation”, I would add that depending on where the mapping is executed, the amount of data that is exchanged between both components varies. This might be important in case both grids have a significantly different number of cells.

In the text it is nowhere mention what the abbreviation OASIS stands for.

P1L13-15 “It includes […] full parallelisation of the […] grid interpolation”
This may be interpolated as OASIS being able to generate interpolation weights on-the-fly in parallel.

P2L21 “source neighbour weights”
I do not know this term.

P3L7-8 “the hub coupler […] is no longer required”
This could be interpreted as: not required but still usable. Is that intended?

P4L15-18 “Compared to OASIS3 which required two data rearranges to couple fields in order to pass through the hub, OASIS3-MCT requires just one parallel rearrange to move data between two components.”
You are comparing the coupling of fields in OASIS3 with the moving of data between components in OASIS3-MCT, which seems unfair, because in the paragraph above it is said that OASIS3-MCT also requires two data rearranges for the full coupling. Or is there a misunderstanding?

P5L9 “Mapping weight files can either be read directly”
For big weight files it may be important to know whether this is done in serial or in parallel. Only in section “4. Conclusions” it is mentioned that I/O in general is done in serial.

P5L18 “Users also have an additional option to specify the type of mapping to be carried out.”
The term “type of mapping” is a little bit ambiguous. It could also refer to interpolation types (e.g. linear, nearest neighbour, or conservative interpolation).

P6L1-11 Maybe you should mention that there is the possibility to turn off the CONSERV transform. Which is important since this operation does not make sense for all field types.

P6L1-11 There is a bfb option for CONSERV transform and for mapping type. This can be confusing. Maybe clarify this.

P7L8-20 whole paragraph + Figure 2
This paragraph and the associated figure seem to be out of place. I would expect them to be part of a user manual.

P7L27 “a field put routine must be called before the matching get”
In case there are two components comp1 and comp2, if there is only a one directional data flow from comp1 to comp2, do all puts in comp1 actually have to be called before (in time) the respective gets in order to avoid a deadlock? Or do the gets wait until the respective put is called?

P8L11 “16,000 cores”
Maybe you should talk about MPI processes or specify that you are using one MPI process per core.

P8L28-29 “There is however clearly some concern that as core counts continue to increase, the initialization time will continue to grow.”
Did you analyse the cause for the increase? Can you add some discussion on this?

P9L8-10 With two-nearest-neighbour interpolation you should have two weights per point on the destination grid.
T799->ORCA025: 2 * 1442 * 1021 = 2,944,564 weights << 4.5 mio weights
ORCA025->T799: 2 * 843,490 = 1,686,980 weights << 3 mio weights
Did I misunderstand something or how do you explain the difference in the number of weights?

P9L13-14 “above 8000 cores per component, the timing is degraded relative to lower core counts. At higher core counts, the timing depends heavily on the MPI performance.”
Why do you not see this behaviour in the IS-ENES2 coupling technology benchmark? Is this due to the different grids used in both test cases?

P10L25 “while for the src+bfb case, the single operation performs slightly worse”
Are you sure that the measurements (10.56 vs 11.89), this statement is referring to, are correct?
(5.95 + 6.02) > 10.56 (taken from Table 2 and Table 3)

P11L13-14 “It's likely that the MPI memory footprint is accounting for most of this behavior (Balaji et al, 2008, Gropp, 2009).”
With a modern MPI implementation this should not happen. I have not seen this behaviour in similar measurements for the ICON model. You could verify this using for example the valgrind tool Massif.

P20 Figure 4
In this case, I would not use a trendline or and any line between the measurement points. The number of cores has a significant impact on the decomposition, which might lead to interesting result between the provided measurements. Therefore, a line between the points implies a continuity that might not reflect reality for this test case.

P21 Figure 5
Are these single measurements or averages?

P22 Table 2 and 3
(0.69 + 0.60) == 1.29 => Did the data exchange between the components only take a negligible amount of time?

P22 Table 2 and 3
(5.95 + 6.02) > 10.56 => Are the measurements correct?

P22 Table 2 and 3
(11.89 – (4.70 + 4.60)) > (12.15 – (4.86 + 4.97)) => time for mapping ↑ time for transfer ↓ => How do you explain this?

P22 Table 4
(2.11 – 1.29) >> (2.17 – 1.61)
I would assume that the cost for CONSERV is independent of the src/dst option. How do you explain the difference?

Technical Comments

P1L14 “separate hub coupler **process**”

P1L23 “OASIS is **a** coupling software”

P1L32-33 “OASIS-MCT supports coupling of fields on relatively arbitrary grids […].”
Is “OASIS-MCT supports coupling of fields defined on most grid types, commonly used in climate science, […].” better?

P1L33 “via a put/get approach. This approach means components make subroutine calls […]”
“via a put/get approach, which is based on components making subroutine calls […]”?

P2L20-21 “calculation of the ~~**source neighbour**~~ weights and addresses needed for the mapping”

P2L23-26 check use of Oxford comma

P2L25 Why did you use the long form for AWI while using the abbreviation for ECMWF, KNMI, and MPI-M?

P2L26-28 Maybe add a reference?

P2L29-30 “OASIS3-MCT extended the widely used and distributed OASIS3 version of the model.”
“It extends the widely used and distributed OASIS3.”?

P3L8 “Transform**ation**s are carried out”

P3L21 “section 4 provides a summary”
Section 4 is called “Conclusion”

P6L2-3 “In OASIS3-MCT, this operation **can** ~~**is**~~ now **be** performed in parallel on the source or destination processes”
If the bfb option is used, it will still be done in serial, or not?

P7L1-2 “are indicated by the different lettered arrows ~~**in Figure 1**~~.”

P9L10 “~~**there are**~~”

P10L5 “(1.91**s** vs 4.70**s**)”

P10L32 “CONSERV unset”
In Table 4 this is called off.

P10L31-33 “Table 4 shows [...]. Table 4 shows [...]”
Identical start of two consecutive sentences.

P11L4-5 “such as area overlap conservative”
Maybe place a reference to:
http://dx.doi.org/10.1175/1520-0493(1999)127%3C2204:FASOCR%3E2.0.CO;2

P12L19 “~~**10s**~~**tens** of thousands”

P12L27-28 “~~**fastest**~~**best** performance”

P16L5-7 “Valcke […] 2012a”
P16L12-14 “Valcke […] 2015”
Could not find references of these papers in the text.

P19 Figure 3
P20 Figure 4
P21 Figure 5
x-axis: maybe use logarithmic base 2 instead of 10
y-axis label: “~~**seconds**~~**Time in s**”

P19 Figure 3
y-axis: use logarithmic scale to better show behaviour for 1 to 1000 cores per component

P21 Figure 5
The data set “T799->025,dst” seems to have two data points at 24 core per component while all others only have one.

P21 Figure 5
For higher number of cores (> 40), the choice of the symbols for the individual data sets makes it hard to read.

P22 Table 4 “~~**pes**~~**cores**”

P22L23 “~~**tasks**~~**cores**”

P23 Figure 6
x-axis: maybe use logarithmic base 2 instead of 10

P23 Figure 6
MB or MiB? per core?

Questions not necessarily relevant for the paper

P1L19-20 “10,000 two dimensional coupling fields”
In case of 3d fields, would the different levels be counted as separate fields?

P7L24 “OASIS3-MCT provides some new capabilities to detect potential deadlocks before they occur”
Very interesting! Can you be more specific?

P7L28-29 “In OASIS3-MCT, puts are always non-blocking while gets are blocking.”
Are there plans for non-blocking gets?

P12L11-12 “the cost associated with generating the mapping files can be moved to a preprocessing step”
Which not necessarily has to be faster, if weight computation is done in parallel.