Reply to Referee #1.

We would like to thank Moritz Hanke for his careful review and thoughtful comments. We will reply to the comments below (in green text)

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.

We have added the following clarifications:

- " A separate top-level driver to control system sequencing is not required "
- " all coupling fields passed through a separate central hub coupler component"
- source and destination are implicitly defined in the introduction
- "Each parallel field in the source model was gathered to a single process on the hub where operations such as mapping and time averaging were executed, and the field was then scattered to the destination model",
- MCT router is an MCT datatype. We have updated the text and the only place where "router" appears in text is in the following sentence where it is defined, " Data communication and mapping rearrangement is handled internally in OASIS3-MCT via MCT routers.".
- "CONSERV" is clearly defined in section 2.4, "The CONSERV operation computes global sums of the source and destination fields and applies corrections to the decomposed mapped field in order to conserve area-integrated field quantities."

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.

To be honest, the original title of the paper was " Development and performance of a new version of the OASIS coupler, OASIS3-MCT ", but the editor encouraged us to be more specific with regard to the version in the title prior to formal submission. The paper is written at a time when OASIS3-MCT_3.0 is the current release, and so we feel it is reasonable to include that information in the title. It is true that this paper takes a slightly broader approach by summarizing changes since OASIS3 including features added before OASIS3-MCT_3.0 (see details in Appendix A). It even includes some information about what is coming in the version 4.0 release of OASIS3-MCT. We made a few changes in the text to further clarify the scope of the paper but feel the current title is reasonable. In particular, we have added " This paper describes the development of OASIS3-MCT from OASIS3 to the current version 3.0 release and will also introduce some new features expected in the version 4.0 release." to the introduction.

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

We have updated the text so all references to figures and tables in the text are lower case unless they occur at the start of the sentence.

"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 reviewer's comments are correct. It doesn't fundamentally matter whether multiple components are run as a single executable or as multiple executables in Oasis3-MCT. I think the main point of including that statement is to make it clear that both modes are supported. We have added a sentence at the end of the second paragraph in section 2.5 to emphasize that point and address the reviewer's concerns.

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.

This is a reasonable point. However, Figure 5 is useful in that it shows the scaling of mapping across a broader range of pe counts which some readers might find useful. The other problem is that while it's relatively easy to time the mapping separately with appropriate barriers, it's much harder to time the transfer in these cases as there is significant load imbalance, puts are non-blocking, and some of the performance is associated with overlapping transfer and mapping work. We believe the information in table 1 is consistent with the reviewers request, and figure 5 provides additional insight into the mapping performance that goes beyond what could be done with a table. A final point is that it's not correct to suggest the map timing is independent of grid type. In fact, the grid decomposition, number of weights, whether mapping is done on the src or dst side, number of pes in play, and distribution of the weights have a large impact on the map timing. We have updated figure 5, so the symbols and symbol key are clearer. We do agree about the comment that amount of data exchanged is important and we have added the statement, " Another point is that if there is a large disparity in the number of grid cells in the two mapped grids, it should be better to exchange the coupling fields expressed on the grid with the fewest grid cells and perform the remapping on the other component tasks."

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

We have added a sentence in the first paragraph of the introduction to define the OASIS project.

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.

We have changed the sentence to read, "parallelization of the coupling communication and run time grid interpolation " to emphasize parallelization of the interpolation at run time, which is unrelated to the process of weights .

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

We have rewritten this sentence as "In particular, OASIS4 included a library that performed a parallel calculation for generation of the mapping weights and addresses needed for the interpolation of the coupling fields."

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

This is a good point. We have changed this sentence to "Third, the OASIS hub coupler was deprecated and is no longer needed or implemented."

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?

This is a very good point. We have clarified this sentence as follows, " Compared to OASIS3, which required an all-to-one communication, interpolation on the single hub process, and a one-to-all communication to couple fields, OASIS3-MCT requires just one parallel all-to-all communication between the source and destination processes and one parallel mapping which includes a rearrangement of the data on the source or destination processes. " We have also changed some of the wording in the document to provide more consistency, clarifying the terms redistribution, communication, coupling, and mapping.

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.

We have added a new sentence to further define the implementation, "In OASIS3-MCT, the weight files are read serially on the root process and distributed to other processes in reasonable chunks.

That chunk size is currently set to 100,000 weights at a time to limit memory use on the root process."

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).

This is a good comment. We have changed this sentence to "Users also have an additional option to set the implementation of the underlying mapping algorithm."

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.

We have added the word "optional" in the first sentence of section 2.4 to reiterate the fact that CONSERV is an optional transform. We have also updated this section to reflect some new features.

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

We recognize that the common keywords are not ideal and are working to differentiate them in future releases. We have added a sentence in section 2.4 to clarify, "Note that both the CONSERV operation and the underlying mapping algorithm setting share a common flag, *bfb*, but that these two settings are completely independent."

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.

We have removed this section and Figure 2 from the paper. This information is in the user guide and we agree that this does not need to be duplicated in the paper.

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?

This is a good question and something we've been trying to clarify in the implementation and user guide. To answer the question, each put is non-blocking but waits for the completion of the put of the same coupling field at the previous coupling timestep before it executes. Therefore, you cannot queue up a bunch of puts before executing a get on overlapping or non-overlapping pes. We have tried to clarify this paragraph in section 2.5 by adding, "In OASIS3-MCT, puts are generally non-blocking while gets are blocking. More specifically, a put waits for the completion of the put of the same coupling field at the previous coupling timestep before proceeding in order to prevent puts from queuing up in MPI and using excess memory. In other words, for a specific put-get pair, the last put can never be more than one coupling period ahead of the equivalent get in OASIS3-MCT. This means that the puts and gets have to be interleaved when coupling on overlapping tasks. It is not possible to queue up a series of puts over multiple coupling periods before executing the equivalent gets."

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

We are constantly struggling whether to use MPI tasks, processes, cores, or pes as a way to describe parallelism. We have tried to be consistent in the paper. We have changed the text from "16,000 cores" to "16,000 MPI tasks".

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?

To address this comment, the end of the last paragraph in section 3.1 has been updated as follows, "The initialization uses MPI heavily to initialize the coupling interactions, read in the mapping files, and setup the communication for the mapping rearrangement and coupling communication. In general, the initialization is not expected to scale well, but the initialization overhead is what allows the model to run efficiently during the actual run phase. There is clearly some concern that as core counts continue to increase, the initialization time will continue to grow. OASIS developers continue to monitor and analyze both the runtime and initialization costs and make improvements. "

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?

We had an error in the description, the weights are based on five-nearest-neighbor interpolation and the ORCA025 grid has masked points. 4.5 million weights for T799->ORCA025 is the equivalent of 61% unmasked points on the ocean grid, 3.0 million weights for ORCA025->T799 is 71% of the maximum number of weights if the grids were unmasked. We have corrected that section and it now reads, "Each coupling of data between a pair of components consists of a mapping operation that interpolates the masked data via a five-nearest-neighbor algorithm that includes both floating point operations and rearrangement, and then a communication operation that transfers the data between concurrent sets of MPI tasks in the different components. So there are four distinct MPI operations in a single ping-pong. There are 4.5 million different links (weights) between the T799 grid points and the ORCA025 grid points and 3 million weights for the mapping in the other direction."

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?

This paper does not mention nor include an analysis or comparison to the IS-ENES2 benchmark. Having said that, the comment is interesting, and we are currently looking at the benchmark results in the context of these timing tests to better understand the timing differences. The curves in the IS-ENES benchmark show roughly the same behavior although the absolute timing is quite different. These differences are likely related to the different resolutions, different mapping files, and different machines used in the two 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)

This is a good point that we should clarify. The pipo time is done without any barriers while the mapping timing is done as a separate test run with barriers around the mapping. In general, those barriers will slow the model down because any overlap in mapping and data transfer due to load imbalance will be lost with the barriers. Timing parallel kernels in a consistent way is always tricky.

We have updated section 3.4, combined tables 2 and 3, and added some new timing information for the pipo time when barriered. We hope this significantly clarifies the timing information.

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.

We have updated this sentence to "It is possible that the MPI memory footprint is accounting for most of this behavior (Balaji et al, 2008, Gropp, 2009), but further investigation will need to be carried out in the future to confirm." We hope that is a reasonable response.

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.

We have removed the line between the measurement points in Figure 4.

P21 Figure 5
Are these single measurements or averages?

We have added 1 sentence in section 3.3 to answer this question, "Two trials were carried out and the results shown are for the best times with variability generally much less than 5% between runs."

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?

We have rerun the tests with additional timing information, combined tables 2 and 3, updated the table with some additional results, and updated section 3.4 to clarify these results. The barriered pipo time is now shown to compare with the sum of the map time for an apples and apples comparison. Compared to the unbarriered pipo time, this also better demonstrates the amount of load imbalance and overlapping work between the mapping and communication in the unbarriered case and the text has been revised to reflect that.

P22 Table 2 and 3

(5.95 + 6.02) > 10.56 => Are the measurements correct?

See comment above.

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?

Again, this comes down to the barrier around map timing which we now describe in the text. See the comment above with regard to P22, Table 2 and 3. We have added some text in section 3.4 to explain the timing numbers better. The old timing information did not provide insight into the load imbalance. In fact, the mapping time does go up but you cannot immediately assume the communication time is decreased. This is hopefully clarified in the text.

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?

It's not clear that you can make simple conclusions like this from the timing information. The timing of the pipo is complicated by load imbalance, dependencies in the communication between tasks, and other issues. In addition, the order of operations for src+bfb and dst+bfb are quite different and depending where in the sequencing the global sums are carried out, this can have an impact on the load imbalance and overall pipo time. We have updated Table 4 to reflect some new results and we have added some additional information in the discussion in Section 3.5.

Technical Comments

P1L14 “separate hub coupler **process**”

We have implemented this change to the text.

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

We have not made this change, we feel the current wording is ok.

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?

We have updated this sentence consistent with the review.

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 [...]”?

We have updated this sentence as suggested by the reviewer.

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

We have updated the spelling of neighbor

P2L23-26 check use of Oxford comma

We have added a comma as suggested

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

We put abbreviations everywhere.

P2L26-28 Maybe add a reference?

We added Hollingsworth et al., 2008

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

We have updated the sentence as suggested by the reviewer

P3L8 “Transform**ation**s are carried out” P3L21 “section 4 provides a summary”
Section 4 is called “Conclusion”

We have updated this sentence as follows, "... and section 4 provides conclusions and a summary."

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?

It will always be done in parallel. Even if the bfb option is used to compute the global sums, the corrections are applied in parallel on the decomposed fields after broadcasting those global sums to all tasks. We have added a word, "decomposed" to "...applies corrections to the decomposed mapped fields..." to make it clearer the correction is happening in parallel.

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

I assume the reviewer was asking us to check the capitalization of "F"? We have corrected this throughout the paper and changed all references to figures and tables to small letters unless figure or table are the first word of a sentence.

P9L10 “**there are**”

We have removed "there are" in that compound sentence.

P10L5 “(1.91**s** vs 4.70**s**)”

Units have been added

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

We have modified table 4 and used the word unset consistently.

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

We have updated the second sentence starting with "Table 4 shows" to improve readiblity.

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

We have added an equivalent reference as requested by the reviewer

P12L19 "**10stens** of thousands"

We have changed the wording from 10s to tens as suggested

P12L27-28 "**fastestbest** performance"

We have changed the wording of fastest to best as suggested

P16L5-7 "Valcke [...] 2012a"

We have removed this reference and changed 2012b to 2012.

P16L12-14 "Valcke [...] 2015"
Could not find references of these papers in the text.

We have added this reference in Section 1 near the end of the section.

P19 Figure 3
P20 Figure 4
P21 Figure 5
x-axis: maybe use logarithmic base 2 instead of 10 y-axis label: "**secondsTime in s**"

We have renamed the y-axis label, but left the x-axis scale as is.

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

We believe the key to this figure is not the time at the lower core counts, but the time at the higher core counts. Switching the y-axis to log will make that information less clear. We have not changed the y-axis scale.

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.

Thanks for catching that, we have corrected that problem by eliminating a redundant point.

P21 Figure 5

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

This is a good point. We have changed the symbols and updated the symbol table to make the data more readable. None of the symbols are filled anymore.

P22 Table 4 "**pescores**" P22L23 "**taskscores**"

We have changed both the pes and tasks wording to cores as suggested.

P23 Figure 6

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

We have not changed the x or y axis scales.

P23 Figure 6

MB or MiB? per core?

MB is typically used when discussing memory use. I don't think it adds to the paper to differentiate between MB and MiB. They differ in definition by less than 5% and that difference has no impact on the plot or discussions. In fact, the scaling of the memory use is more important than the absolute memory use numbers in the plot.

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?

In the underlying implementation of the new "bundle" feature, the 3d fields are treated under the covers are multiple 2d fields. We count multi-level 3d fields as multiple 2d fields. The requirement for using 2d bundled field is the same as the requirement for coupling multiple fields in a single namcouple statement, i.e. those fields have to share the same grids, masks and will use the same mapping file.

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

Several checks were added like making sure time didn't go backwards, making sure a coupling period wasn't skipped, and others. Some of the new checks had to be removed or deprecated to support sequential coupling on overlapping pes. In general, the new capabilities are not adequate to prevent deadlocks.

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

There are no plans for non-blocking gets. In general, we presume that users execute a get when the data is needed. A non-blocking get would require users add a wait in their code before they could use the data which we think adds complexity with little gain. There is lack of symmetry with respect to put and get in systems such as this. If the community requests non-blocking gets, they could probably be implemented but with some additional burden on users and the user implementation.

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.

This is true. But right now, Oasis3-MCT does not provide an on-line parallel weights computation capability. Several offline tools do provide that capability. In addition, those offline tools have experts in grid and weights generation that cannot (and maybe should not) be duplicated within Oasis. The complexity associated with generating weights on (for instance) complex unstructured grids, and for many different types of gridding options (bilinear, conservative, higher order, gradient preserving, nearest neighbor, and so forth) are probably best dealt with by specialized tools outside Oasis, and these tools do already exist and exceed any capability that Oasis could build. Having said that, if future requirements, such as time evolving grids impose new requirements on Oasis for fast, parallel weights generation, Oasis will consider incorporating additional external tools into the infrastructure. This section of text in the paper was updated to reflect these ideas.

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 map- ping 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.

Reply to Anonymous Referee #3.

We would like to thank referee #3 for taking the time to review our paper. Our replies to the three main comments are below (in green).

1) There is no discussion of OpenMP as an alternative to MPI. Future hardware will require going to more shared memory and less message passing.

We acknowledge that there is no discussion of OpenMP in the current paper. OpenMP parallelization is not currently explicitly supported in Oasis3-MCT, and OASIS developers are aware of this shortcoming in the current implementation, especially in regards to possible future architectures. As indicated in the final section of the paper, OpenMP parallelization and performance of Oasis3-MCT on new architectures is something that is currently being explored by the development team. We hope to provide support in the next year (or so) and will have results at that time to share with the community.

2) There is no discussion of GPUs, MICs, etc and plans to port OASIS to novel architectures.

The OASIS development team is actively pursuing access and testing of OASIS on newer architectures and hope to have some results in the next year to share with the community. We recognize this is an important issue moving forward.

3) I am somewhat taken aback by the extreme cost of providing bfb (bit for bit) reproducing algorithms. In other similar codes this cost ratio is somewhat lower (which could of course mean that the non-reproducing modes in other codes are too slow!) This may require some work.

With regard to the cost of the bfb conservation computation, we were also quite shocked at the cost of the bfb operation. We have revised the discussion and results of the CONSERV transform in the paper significantly, adding global sum options that have been recently added to the OASIS infrastructure and that will be released in OASIS3-MCT_4.0. The OASIS3-MCT_3.0 timings showed a clear problem with the bfb CONSERV performance. OASIS3-MCT_4.0 will provide additional options, including an option called "reprosum" that produces bit-for-bit results on different core counts and decompositions while performing significantly better than the current "bfb" option. Please see revised section 2.4, 3.5, and table 3 in the paper.

Development and performance of a new version of the OASIS coupler, OASIS3-MCT_3.0

Anthony Craig[1], Sophie Valcke[1], Laure Coquart[1]

[1]CECI, Université de Toulouse, CNRS, CERFACS, 42 Av. G. Coriolis, 31057 Toulouse Cedex 01, France

Correspondence to: S. Valcke (valcke@cerfacs.fr)

Abstract. OASIS is coupling software developed primarily for use in the climate community. It provides the ability to couple different models[1] with low implementation and performance overhead. OASIS3-MCT is the latest version of OASIS. It includes several improvements compared to OASIS3 including elimination of a separate hub coupler process, parallelization of the coupling communication and run time grid interpolation, and the ability to easily reuse mapping weights files. OASIS3-MCT_3.0 is the latest release and includes the ability to couple between components running sequentially on the same set of tasks as well as to couple within a single component between different grids or decompositions such as physics, dynamics, and I/O. OASIS3-MCT has been tested with different configurations on up to 32,000 processes, with components running on high-resolution grids with up to 1.5 million grid cells, and with over 10,000 two dimensional coupling fields. Several new features will be available in OASIS3-MCT_4.0 and some of those are also described.

Sophie Valcke 6/29/17 5:15 PM
Deleted: , Sciences de l'Univers au CERFACS, URA1875

Tony Craig 6/16/17 10:36 AM
Deleted: full

Tony Craig 6/16/17 10:37 AM
Deleted: l

Tony Craig 6/16/17 10:39 AM
Deleted: s

1 Introduction

OASIS is coupling software developed primarily for the climate community. OASIS was originally an abbreviation for "Ocean Atmosphere Sea Ice Soil", but the capabilities provided by OASIS are not restricted to just those kinds of models, so the name OASIS now represents a project to develop general coupling software. It is in relatively wide use especially in European based modeling efforts. It is one of a number of coupling infrastructure packages (Valcke et al., 2016) that is focused on standard and reusable methods to support coupling requirements like interpolation and communication of data between different models and different grids. OASIS is maintained and managed by the Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (CERFACS) and the Centre National de la Recherche Scientifique (CNRS) in France. It is a portable set of Fortran 77, Fortran 90 and C routines. Low-intrusiveness, portability and flexibility are key OASIS design concepts. The current version of the software, OASIS3-MCT, is a coupling library that is compiled and linked to the component models. Its

[1] Within the text, we use "model" in the sense of a "numerical model"

primary purpose is to interpolate and exchange the coupling fields between or within components to form a coupled system. OASIS3-MCT supports coupling of fields on grid types commonly used in climate science via a put/get approach, which means components make subroutine calls to send (put) or receive (get) data from within the component code directly. A separate top-level driver to control system sequencing is not required to use OASIS3-MCT, but a handful of subroutine calls must be added to the code to initialize the coupling, define grids, define decompositions (partitions), define coupling fields, and to put and get variables between components. OASIS3-MCT leverages a text input file called the *namcouple* file to configure the interactions between components. Mapping (also known as remapping, regridding, or interpolation), time transformations, and the ability to read or write coupling data from disk are supported in OASIS3-MCT.

OASIS development began in 1991 and the first version, OASIS1, was used two years later in a 10-year coupled integration of the tropical Pacific (Terray et al., 1995). In the intervening decades, OASIS2 and OASIS3 were released. The history of OASIS development is well documented (Valcke, 2013). With OASIS3, the coupled models always had to run concurrently as separate executables on different MPI tasks and all coupling fields passed through a separate central hub coupler component that also ran concurrently. OASIS3 allowed parallel coupling of parallel models on a per-field basis by gathering each parallel field in the source model to a single process on the hub where operations such as mapping and time averaging were executed, and the field was then scattered to the destination model. OASIS3 generated mapping weights on a single process at initialization using the SCRIP library (Jones, 1999) from the grid information specified by the component models.

A first attempt to design and develop a fully parallel coupler was started in the framework of the EU FP5 PRISM and FP7 IS-ENES1 projects (see https://is.enes.org), and that led to the development of OASIS4 (Redler et al. 2010). In particular, OASIS4 included a library that performed a parallel calculation for generation of the mapping weights and addresses needed for the interpolation of the coupling fields. This version had several other features such as the use of an xml file for specifying the configuration information. OASIS4 was used by Météo-France, ECMWF, KNMI and MPI-M in the framework of the EU GEMS project for 3-D coupling between atmospheric dynamic and atmospheric chemistry models (Hollingsworth et al., 2008); it was also used by SMHI, AWI and by the BoM in Australia for ocean-atmosphere 2-D regional and global coupling. But OASIS4 had limited success and its development was stopped in 2011 after a performance analysis determined some fundamental weaknesses in its design, in particular with respect to the support of unstructured grids.

With OASIS3-MCT, a different approach was taken to improve the parallel performance and to address new requirements. It extends the widely used and distributed OASIS3 version of the model. This paper describes the development of OASIS3-MCT from OASIS3 to the current 3.0 release and also introduces some new features expected in the next 4.0 release. The initial requirements of OASIS3-MCT were to

Tony Craig 6/7/17 4:32 PM
Deleted: relatively arbitrary grids…. This approach ... [1]

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

Tony 7/1/17 1:00 AM
Deleted: only …: ... [2]

Sophie Valcke 6/29/17 5:28 PM
Deleted: .

Tony 7/1/17 12:57 AM
Deleted:

Sophie Valcke 6/29/17 5:28 PM
Deleted: E

Tony 7/1/17 12:59 AM
Deleted: was gathered

Sophie Valcke 6/29/17 5:30 PM
Deleted: , PHASE 1

Tony Craig 6/16/17 10:45 AM
Deleted: performing a fully…of…source neighbour weights and addresses…map ... [3]

Sophie Valcke 6/28/17 10:36 PM
Deleted: ,…, and…the Alfred Wegener Institute for Polar and Marine Research (…in Germany), …Bureau of Meteorology ... [4]

Tony Craig 6/15/17 11:25 AM
Comment: Sophie, can we rephrase this so it's clearer. Also, Moritz asked about why we mixed abbreviations and long names. And is there a reference we can point to?

Tony Craig 6/7/17 4:45 PM
Deleted: OASIS3-MCT…ed ... [5]

Tony 6/26/17 12:58 PM
Deleted: .

Sophie Valcke 6/29/17 5:52 PM
Deleted: version …will …version ... [6]

Tony 6/26/17 12:58 PM
Deleted:

improve the parallel performance of the coupling, implement an ability to read in mapping weights to mitigate the cost of weights generation, support next generation grids such as high resolution unstructured grids running on high processor counts, and to add those features while retaining the basic OASIS3 application programming interfaces (APIs) and *namcouple* file to support backwards compatibility.

To accomplish these requirements, a number of changes were made. First, a portion of the underlying communication implementation was replaced with the Model Coupling Toolkit (MCT) software package (Larson et al., 2005) developed by the Argonne National Laboratory. This implementation is transparent to the user, as MCT methods and datatypes are only used within the OASIS3-MCT infrastructure to support parallel mapping and parallel redistribution. Second, the ability to specify pre-defined mapping files was added. Mapping files can now be generated offline using a diverse set of packages, such as SCRIP, ESMF (Theurich et al, 2016), or any locally developed methods. Third, the OASIS3 hub coupler was deprecated and is no longer needed or implemented. Transforms are carried out on the component processes, and data is transferred directly between components via MCT. These features were released in OASIS3-MCT_1.0 in 2012 (Valcke et al, 2012) and because of backwards compatibility, OASIS3 users could upgrade easily to OASIS3-MCT.

With the release of OASIS3-MCT_3.0 in 2015 (Valcke et al, 2015), several new features were added to the coupler. 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 an I/O module. OASIS3-MCT_3.0 also allows a component to define grids, partitions, and coupling fields on subsets of its tasks, and it comes with a Graphical User Interface (GUI) to generate the *namcouple* file.

The next section, titled Implementation, describes these features in greater detail. Section 3 provides performance and memory scaling results from OASIS3-MCT_3.0 as well as some initial results for features expected in OASIS3-MCT_4.0, and section 4 provides conclusions and a summary.

2. Implementation

As discussed in the introduction, OASIS3-MCT development started with the objective to keep the OASIS3 general design. The requirements of OASIS3-MCT were focused on improved parallel performance including parallel mapping and parallel data coupling, the ability to efficiently support unstructured grids, the ability to specify pre-defined mapping files to mitigate the serial cost of generating mapping weights on-the-fly, and backwards compatibility in usage of both the *namcouple* file and the

Sophie Valcke 6/29/17 5:52 PM
Deleted: improve reuse and

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

Tony Craig 6/16/17 10:50 AM
Deleted: required

Tony 6/26/17 12:51 PM
Deleted: source or destination

Tony Craig 6/15/17 11:40 AM
Deleted: b

Tony 7/1/17 1:05 AM
Deleted: the

Tony 6/26/17 2:37 PM
Deleted: s

Sophie Valcke 6/29/17 5:56 PM
Deleted: running

Tony 6/26/17 2:37 PM
Deleted: even within a component,

Sophie Valcke 6/29/17 5:56 PM
Deleted: model

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

Sophie Valcke 6/29/17 5:57 PM
Deleted: i

Sophie Valcke 6/29/17 5:58 PM
Deleted: an

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

OASIS3 APIs. A summary of the changes between OASIS3 and OASIS3-MCT_3.0 is provided in Appendix A as well as an initial list of features expected in OASIS3-MCT_4.0.

Tony 6/26/17 1:21 PM
Deleted: .

2.1 General Architecture

To accomplish these tasks efficiently and in a timely manner, the Model Coupling Toolkit (MCT) developed by the Argonne National Laboratories (Larson et al 2005) was incorporated into OASIS3 to support parallel matrix vector multiplication and parallel distributed exchanges. Its design philosophy, based on flexibility and minimal invasiveness is consistent with the approach taken in OASIS. MCT has proven parallel performance and is one of the underlying coupling software libraries used in the National Center for Atmospheric Research Community Earth System Model (NCAR CESM) (Jacob et al., 2005, Craig et al., 2012).

MCT handles two primary tasks in OASIS3-MCT. The parallel transfer of data from a source model to a destination model, and interpolation of fields between decomposed grids. At the present time, these two steps are independent and both are largely performance limited by MPI communication cost at moderate to high processor counts due to the data rearrangement in both. Data communication and mapping rearrangement is handled internally in OASIS3-MCT via MCT routers.

Tony Craig 6/16/17 11:20 AM
Deleted: upling

Tony Craig 6/16/17 11:19 AM
Deleted: communication

Another significant change in the OASIS3-MCT implementation compared to OASIS3 is that a separate hub coupler executable running on its own processes is no longer needed. Accumulation, temporal lagging, mapping, and other transforms are carried out in the OASIS3-MCT coupling layer on the model processes in parallel using temporary memory to store data as needed. Compared to OASIS3, which required an all-to-one communication, interpolation on the single hub process, and a one-to-all communication to couple fields, OASIS3-MCT requires just one parallel all-to-all communication between the source and destination processes and one parallel mapping which includes a rearrangement of the data on the source or destination processes. In addition, the memory needed in the infrastructure in OASIS3-MCT is much more scalable.

Tony Craig 6/16/17 10:57 AM
Deleted: two data rearranges

Sophie Valcke 6/28/17 8:41 PM
Deleted: a

Sophie Valcke 6/28/17 8:43 PM
Deleted: in order to pass through the OASIS3 hub

Sophie Valcke 6/28/17 8:55 PM
Deleted: redistribution

Tony Craig 6/16/17 10:56 AM
Deleted: rearrange

Sophie Valcke 6/28/17 8:43 PM
Deleted: to move

Sophie Valcke 6/28/17 8:43 PM
Deleted: between two components

2.2 Coupling

OASIS3-MCT fundamentally supports coupling of 2-D logically rectangular fields but 3-D fields and 1-D fields are also supported using a one-dimension degeneration of the grid structure. If the user provides a set of pre-calculated weights, OASIS3-MCT will be able to interpolate any type of 1-D, 2-D, or 3-D fields, but the capability to calculate the mapping weights by the coupler is only available for 2-D fields on the sphere.

Another new feature is the option to couple multiple fields as a single coupling operation. This is supported for fields for which the coupling options defined in the *namcouple* file are identical. This can

Tony 6/29/17 12:50 PM
Deleted: namcouple

Tony 6/29/17 12:50 PM
Formatted: Font:Italic

improve performance because rather than mapping and coupling fields one at a time, the mapping and coupling can be aggregated over multiple fields. Coupling multiple fields at once is accomplished by specifying a list of colon-delimited fields in the *namcouple* file on both the source and destination side. In this implementation, the get and put calls in the model are still individual calls on individual fields, but the coupling layer will aggregate the multiple fields specified in the *namcouple* file into a single step. On the put side, the multiple fields are not mapped or sent until all of the individual put calls are made. On the get side, the multiple fields are received and mapped on the first get call and then subsequent get calls just copy in fields that were received earlier. A user can quickly switch between coupling single and multiple fields just by changing the *namcouple* input file.

One additional feature available in the current development version and that will be released with the next official version, OASIS3-MCT_4.0, is the ability to couple a bundle of 2-D fields via extensions to the OASIS calling interfaces. An extra dimension is supported in the variable definition and in the get and put field arrays. In this case, a user can treat a bundled 2-D field as a single field in the system, while the underlying implementation treats it just like a multiple field coupling.

2.3 Interpolation

Mapping weight files can either be read directly or generated at run-time, on one processor, using the same serial method based on SCRIP as existed in OASIS3. In OASIS3-MCT, the weights are read serially by the root process and distributed to other processes in reasonable chunks, currently set to 100,000 weights at a time to limit memory use on the root process. For the interpolation, OASIS3-MCT creates a simple one-dimensional decomposition of the source grid on the destination processes or vice-versa. Fields are then either remapped to the destination grid on the source processes and then sent to the destination processes or sent to the destination processes and then remapped to the destination grid. The user is able to specify whether the source or destination processes are used for remapping via an optional setting in the *namcouple* file. That choice will generally be made based on mapping performance and depends on the relative size of the grids, the number of weights, and the process counts of the source and destination models. In OASIS3-MCT_4.0, a new option is expected that may reduce the mapping rearrangement cost by choosing a more efficient decomposition of the source grid on the destination processes (or vice-versa) compared to the current default one-dimensional decomposition.

Users also have an additional option to set the implementation of the underlying mapping algorithm. The *bfb* option will enforce an order of operations that will be bit-for-bit identical on different process counts. It does this by distributing the mapping weights on the destination decomposition and then redistributing the source coupling field grid point values to the destination processes before applying the mapping weights. This ensures operation order is independent of decomposition. The *sum* option does the opposite.

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

Sophie Valcke 6/29/17 6:05 PM
Deleted: they can be

Tony 6/29/17 12:02 PM
Deleted: the former case,

Sophie Valcke 6/29/17 6:06 PM
Deleted: files

Sophie Valcke 6/29/17 6:06 PM
Deleted: on

Sophie Valcke 6/29/17 6:07 PM
Deleted: . That chunk size is

Sophie Valcke 6/29/17 6:08 PM
Deleted: interpolated

Sophie Valcke 6/29/17 6:08 PM
Deleted: interpolated

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

Sophie Valcke 6/29/17 6:09 PM
Deleted: to be released

Tony 6/26/17 1:25 PM
Deleted: will

Sophie Valcke 6/29/17 6:10 PM
Deleted: in mapping

Sophie Valcke 6/29/17 6:11 PM
Deleted: the mapped field

Tony Craig 6/16/17 11:51 AM
Deleted: pecify

Tony Craig 6/16/17 11:48 AM
Deleted: type of

Tony Craig 6/16/17 11:48 AM
Deleted: to be carried out

It distributes the mapping weights on the source decomposition and then computes partial sums of the destination field on the source decomposition, before rearranging them to the destination decomposition and adding up the partial sums. This does not guarantee identical order of operations on different process counts and decompositions. In both approaches, the same number of floating operations are carried out as defined by the mapping weights. The main difference between the *bfb* and *sum* strategies is that in *bfb* mode, the source field is rearranged onto the destination distribution before the mapping weights are applied while in *sum* mode, the mapping weights are applied on the source decomposition to form partial sums of the destination field and then the partial sums are rearranged. From the performance point of view, it's generally better to use the method that rearranges the field on the grid that contains the fewest grid cells to minimize the communication cost. But of course, if bit-for-bit reproducibility on different core counts is required, then the *bfb* mode should be chosen.

Sophie Valcke 6/29/17 6:12 PM
Deleted: redistributing …or…;… those are target ... [7]

Tony 6/29/17 12:54 PM
Deleted: grid

2.4 Conservation

With OASIS3-MCT, the optional CONSERV transform has been refactored. In OASIS3, this operation was always performed on a single process. In OASIS3-MCT, this operation is now performed in parallel on the source or destination processes. The CONSERV operation computes global sums of the source and destination fields and applies corrections to the decomposed mapped field in order to conserve area-integrated field quantities. There are two options for computing the global sums in OASIS3-MCT_3.0. The first, *bfb*, gathers the fields onto the root process to compute the global sums in an ordered fashion that guarantees bit-for-bit identical results regardless on the number of cores or decomposition of the field. (Note that both the CONSERV operation and the underlying mapping algorithm setting share a common flag, *bfb*, but these two settings are completely independent.) The second CONSERV option, *opt*, carries out a local double precision sum of the field and then does a scalar reduction to generate the global sums. This will typically introduce a round off difference in the results when changing process counts or decomposition but is much faster. However, the *opt* option will be bit-for-bit reproducible if the same number of processes and decomposition are used between different runs.

In the OASIS3-MCT_4.0 release, three new options (*lsum16*, *ddpdd*, and *reprosum*) will be added to compute the global sums in CONSERV. At the same time, *opt* will be renamed *lsum8* while *bfb* will be renamed *gather*. The rest of this paper will use the OASIS3-MCT_4.0 naming convention for CONSERV options. The first new global sum method, *lsum16*, works just like *lsum8* but uses quadruple precision to compute the local sums and to carry out the scalar reduction. The cost will be higher than *lsum8* but there is greater chance that results will be bit-for-bit for different decompositions than *lsum8*. The *ddpdd* is a parallel double-double algorithm using a single scalar reduction (He and Ding, 2001). It should behave between *lsum8* and *lsum16* with respect to performance and reproducibility. The third new algorithm, *reprosum*, is a fixed point method based on ordered double integer sums that requires two scalar reductions

Sophie Valcke 6/29/17 6:21 PM
Deleted: and is an inherent part of the mapping operation

Tony Craig 6/16/17 6:58 PM
Deleted: currently

Tony 7/1/17 1:19 AM
Deleted: T…however ... [8]

Sophie Valcke 6/29/17 6:22 PM
Deleted: is

Tony Craig 6/16/17 7:31 PM
Formatted ... [9]

Tony 7/1/17 1:20 AM
Deleted: on different task counts

Tony Craig 6/16/17 7:05 PM
Formatted ... [10]

Tony 7/1/17 1:21 AM
Deleted: and…a few ... [11]

per global sum (Mirin and Worley, 2012). The cost of reprosum will be higher than some of the other methods, but it is expected to produce bit-for-bit results on different task counts except in extremely rare cases, and the cost should be significantly less than the *gather* method.

2.5 Concurrency, Process Layout, and Sequencing

The ability to couple fields within one executable running on partially overlapping tasks was added in OASIS3-MCT_3.0. A number of new capabilities had to be implemented to support this feature including the ability to define grids, partitions, and coupling fields on subsets of component tasks. There also had to be a major update in the handling of MPI communicators within the infrastructure. These changes are transparent to the user. This allows, within a single model, different sets of MPI tasks to define multiple grids, multiple decompositions (partitions), and different coupling fields. These new features and updates provide the flexibility needed to couple fields between components or within a component.

Figure 1 provides a schematic of the type of coupling that can be carried out between and within components in OASIS3-MCT_3.0. Executables are defined as separate binaries that are launched independently at startup, components are defined as separate sets of tasks within an executable, and grids can be defined on all tasks or on a subset of tasks within a component. Each task will be associated with only one executable and one component in any application, but multiple grids and decompositions can exist across overlapping tasks within a component. While OASIS3-MCT supports both single and multiple executable configurations, the coarsest level of concurrency in the system is the component.

In figure 1, an example schematic is presented that shows how two executables, exe1 and exe2, are running concurrently on separate sets of MPI tasks (0-5 for exe1 and 6-37 for exe2). Executable exe1 includes only one component comp1 that has coupling fields defined on only one grid, grid1 (decomposed on all 6 tasks). Executable exe2 includes 3 components, comp2, comp3, and comp4 running concurrently on tasks 6-11, 12-33 and 34-37 respectively. Component comp2 participates in the coupling with fields defined on only one grid, grid2 (decomposed on all 5 tasks) while comp4 does not participate in the coupling. Component comp3 exchanges coupling fields defined on 3 different grids, grid3 (tasks 12-21), grid4 (tasks 22-30) and grid5 (tasks 12-26, overlapping with both grid3 and grid4). Finally, comp3 has 3 tasks (31-33) not involved in the coupling. Different coupling capabilities are indicated by the different lettered arrows in figure 1. Coupling is supported between components in separate executables, within a single executable between different components, and between overlapping, non overlapping, or partially overlapping grids in a single component. In OASIS3, only coupling between separate executables was supported; in OASIS3-MCT_3.0, a functional and highly flexible coupled system can now be designed and implemented as either a single executable or with multiple executables.

Tony Craig 6/16/17 7:34 PM
Formatted: Font:Italic

Tony 6/26/17 1:30 PM
Deleted: .

Tony 6/26/17 1:30 PM
Formatted: Font:Bold

Tony 6/26/17 2:35 PM
Deleted: between components

Sophie Valcke 6/29/17 6:24 PM
Deleted: In OASIS3-MCT_3.0,

Sophie Valcke 6/29/17 6:24 PM
Deleted: e

Tony 6/29/17 12:06 PM
Deleted: being

Tony Craig 6/7/17 5:54 PM
Deleted: F

Tony Craig 6/7/17 5:51 PM
Deleted: .
Figure 2 shows how the OASIS3-MCT_3.0 API calls should be executed across different tasks to support the coupling shown in figure 1. Each MPI tasks has to call the oasis initialization routine (oasis_init_comp) once with the name of its component. Comp4 is not participating in coupling, so that component calls the oasis initialization routine with the argument coupled=.false. and then that component does not need to call any other OASIS3-MCT routine. Since some of comp3 tasks are participating in the coupling, all comp3 tasks have to call the routines to initialize the coupling (oasis_init_comp), retrieve a local communicator for all component processes (oasis_get_localcomm), create a coupling communicator (oasis_create_couplcomm), finalize the definition phase (oasis_enddef), and terminate the coupling (oasis_terminate), and these are the only routines that have to be called by comp3 tasks 31-33 since those tasks are not participating in the coupling. To initialize the coupling exchanges, the tasks that participate in coupling have to define the decomposition of the grids (oasis_def_partition) and declare the coupling fields (oasis_def_var). Finally, the tasks exchanging coupling fields have to call the sending (oasis_put) and receiving (oasis_get) routines accordingly. .

Within OASIS, it has always been mandatory for a user to establish a set of configuration inputs that are consistent with the get and put sequencing in the components such that the coupled system will not deadlock. OASIS3-MCT provides some new capabilities to detect potential deadlocks before they occur, but it is still largely up to the user to make sure this does not happen. This is even more important for coupling components on overlapping tasks as there is almost no way to detect a deadlock ahead of time. Specifically, a field put routine must be called before the matching get (taking into account any lags specified in the configuration file) when coupling on overlapping tasks. In OASIS3-MCT, puts are generally non-blocking while gets are blocking. More specifically, a put waits for the completion of the put of the same coupling field at the previous coupling timestep before proceeding in order to prevent puts from queuing up in MPI and using excess memory. In other words, for a specific put-get pair, the last put can never be more than one coupling period ahead of the equivalent get in OASIS3-MCT. This means that the puts and gets have to be interleaved when coupling on overlapping tasks. It is not possible to queue up a series of puts over multiple coupling periods before executing the equivalent gets.

Tony Craig 6/16/17 12:35 PM
Deleted: always

Sophie Valcke 6/28/17 9:31 PM
Deleted: same

Sophie Valcke 6/28/17 9:32 PM
Deleted: last

Tony Craig 6/16/17 12:34 PM
Deleted: .

2.6 Other Features

There are several additional features in OASIS3-MCT relative to OASIS3. The grid writing routines have been extended to support parallel calls from all component processes. However, even when the parallel interface is used, the grid information is still aggregated onto the root processor within the OASIS3-MCT layer and then written serially to disk.

OASIS3-MCT also now includes a GUI, which is an application of OPENTEA (Dauptain, 2014), the graphical interface developed at CERFACS. The OASIS3-MCT GUI helps users produce the *namcouple* configuration file for a specific run, without worrying about the format syntax of the file.

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

3. Performance

This section summarizes the performance of various aspects of OASIS3-MCT_3.0 at low and high process counts and at moderate to high resolution. The performance and scaling of initialization, coupling, mapping, conservation and other features will be presented. Memory usage will also be shown.

3.1 Initialization

Figure 2 shows the initialization cost for a T799-ORCA025 test case on up to 16,000 MPI tasks per component, with the two components running concurrently (32,000 tasks total) on Curie at CEA TGCC. Curie consists of 5040 nodes with 2 eight-core Intel Sandy Bridge EP (E5-2680) 2.7GHz processors per node connected with an InfiniBand QDR Full Fat Tree network. These tests were run with simple toy

Tony 6/26/17 1:55 PM
Deleted: 3

Tony Craig 6/16/17 12:51 PM
Deleted: cores

Sophie Valcke 6/28/17 10:54 PM
Deleted: 's

Tony 6/29/17 12:08 PM
Deleted: ®

Tony 6/29/17 12:09 PM
Deleted: processors

Sophie Valcke 6/29/17 12:40 PM
Deleted: 16 Intel Sandy Bridge

Sophie Valcke 6/29/17 12:40 PM
Deleted: processes

models that define grids, couple test data, but have practically no model initialization or run-time overhead. This configuration was chosen because it demonstrates OASIS3-MCT's ability to support high-resolution climate configurations. The T799 is a global atmospheric gaussian reduced grid with a ~25km resolution and 843,490 grid points. The ORCA025 grid is a tripolar grid with 1442 x 1021 (~1.47 million) grid points and is one of the grid configurations used by the NEMO ocean model (http://www.nemo-ocean.eu/). The OASIS3-MCT initialization consists of several steps including setting up the partitions, reading in and distributing the mapping weights, computing the mapping rearrangement, communication patterns, and computing the coupling communication patterns. Most of these operations rely heavily on MPI to define the interactions, reconcile the coupling fields and decompositions, and setup the mapping and coupling interactions. Multiple runs were performed for each number of cores with little variability in timing measured. Based on the results in figure 2, the total initialization time for Oasis3-MCT is likely to be reasonable for most applications, even at high numbers of cores. Below 2000 MPI tasks per component, the OASIS3-MCT initialization time is less than one minute. At 16,000 tasks per component, for this relatively high-resolution configuration, the initialization time is below 7 minutes. The initialization uses MPI heavily to initialize the coupling interactions, read in the mapping files, and setup the communication for the mapping rearrangement and coupling communication. In general, the initialization is not expected to scale well, but the initialization overhead is what allows the model to run efficiently during the actual run phase. There is clearly some concern that as task counts continue to increase, the initialization time will continue to grow. OASIS developers continue to monitor and analyze both the runtime and initialization costs.

3.2 Coupling

Figure 3 shows the cost of a ping-pong coupling for the same configuration as figure 2. The times are per single ping-pong coupling but the test was done by running and averaging 1000 ping-pongs. In a ping-pong test, data is passed back and forth between the two components sequentially. In other words, data is sent from model 1 and received by model 2, followed by different data being sent from model 2 to model 1. Each coupling of data between a pair of components consists of a mapping operation that interpolates the non-masked data via a five-nearest-neighbor algorithm that includes both floating point operations and rearrangement, and a communication operation that transfers the data between the concurrent sets of MPI tasks of the two components. So there are four distinct MPI operations in a single ping-pong. There are 4.5 million different links (weights) between the T799 grid points and the ORCA025 grid points and 3 million weights for the mapping in the other direction. In this case, scaling is good to about 400 cores per component as the MPI cost is relatively small and the floating point operations associated with the mapping dominate the cost. Between 400 and 4000 cores per component, the ping-pong cost is relatively constant and 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. At 8000 cores per component,

Sophie Valcke 6/28/17 10:58 PM
Deleted: based on a …from …r ... [12]

Tony 6/26/17 1:55 PM
Deleted: 3

Sophie Valcke 6/28/17 11:02 PM
Deleted: cores (…)…cores ... [13]

Tony Craig 6/16/17 12:56 PM
Deleted: There is however clearly some concern that as core counts continue to increase, the initialization time will continue to grow. But one has to consider that in many ways, the time spent setting up complex MPI interactions during the initialization is coupling ... [14]

Sophie Valcke 6/28/17 11:03 PM
Deleted: core

Tony 6/26/17 1:55 PM
Deleted: 4

Tony Craig 6/16/17 11:21 AM
Deleted: mmunication

Tony 6/26/17 1:55 PM
Deleted: 3

Sophie Valcke 6/28/17 11:05 PM
Deleted: ing

Tony Craig 6/16/17 11:22 AM
Deleted: mmunication

Tony 6/29/17 12:12 PM
Deleted: : 1-

Tony Craig 6/16/17 3:20 PM
Deleted: two

Tony 6/29/17 12:12 PM
Deleted: 2-

Sophie Valcke 6/28/17 11:08 PM
Deleted: then

Tony 6/29/17 12:12 PM
Deleted:

Tony Craig 6/16/17 11:13 AM
Deleted: redistribution

Sophie Valcke 6/28/17 11:07 PM
Deleted: in…different ... [15]

Tony Craig 6/7/17 5:06 PM
Deleted: ,…there are ... [16]

decompositions are getting relatively sparse with just 100 to 200 grid points per core. In addition, timing variability between runs (not shown) above 1000 cores and the jump in cost at 8000 cores suggests that interconnect contention is likely a problem at these core counts. Equivalent timings from OASIS3.3 are also shown in figure 3 (Valcke, 2013), and the ping-pong time is about an order of magnitude better in OASIS3-MCT for a large range of core counts.

3.3 Interpolation

One of the features of OASIS3-MCT is the ability to map data on either the source or destination side as described in Section 2.3. Figure 4 shows the timing of the mapping portion of coupling which includes both the floating point application of weights and the necessary rearrangement of the data on either the source processes (*src*) or the destination processes (*dst*) but not the communication between the source and destination processes. Two trials were carried out, and figure 4 shows the best times with variability generally much less than 5% between runs. This test was run using the T799-ORCA025 toy model on a Lenovo Xeon based cluster at CERFACS consisting of over 6000 2.5 GHz cores connected by an Infiniband FDR. Mapping is about half the total cost of the ping-pong (not shown) in these cases. Figure 4 shows timing data for both mapping directions and for mapping done on the source (*src*) or destination (*dst*) side. In all cases, the *bfb* algorithm is used. The mapping in this case scales well to several hundred cores. In general, the cost of the T799 to ORCA025 mapping is more expensive than the reverse, largely due to the fact that there are more mapping weights (4.5 vs 3.0 million) to apply.

Table 1 documents the ping-pong time for 1000 trials for the same T799-ORCA025 toy model test on Lenovo. In this case, the total number of cores is held at 360, but the relative distribution of cores to each model is varied in three test configurations. The ping-pong tests were carried out with the mapping done on the source, the destination, the ORCA025, or the T799 sets of cores. In these trials, the *bfb* map algorithm was used. In this case, the best performance is when the mapping is done on the model with the highest core count because in this range of core counts, the mapping and communication are still scaling. At higher core counts or with different grids, the optimum performance may be different. For the current cases, the best time is a factor of up to 2.5 times better (1.91s vs 4.70s) compared to the default setting of *src* and by an even greater factor compared to the slowest setting. Another point is that if there is a large disparity in the number of grid cells in the two grids, it should be better to exchange the coupling fields expressed on the grid with the fewest grid cells and perform the remapping on the other component tasks. In general, the number of processes per component is going to be determined by the relative cost of the scientific models, but the above analysis shows that for a given task layout, there may be ways to reduce the coupling cost by mapping on the tasks that provide the greatest performance.

Tony Craig 6/7/17 5:54 PM
Deleted: F

Tony 6/26/17 1:55 PM
Deleted: 4

Sophie Valcke 6/29/17 12:44 PM
Deleted: the

Tony 6/26/17 1:55 PM
Deleted: 5

Sophie Valcke 6/29/17 3:17 PM
Deleted: target

Sophie Valcke 6/29/17 3:21 PM
Deleted: the results

Sophie Valcke 6/29/17 3:21 PM
Deleted: n are for

Tony 6/26/17 1:55 PM
Deleted: 5

Tony 6/26/17 12:18 PM
Deleted: In addition, t

Tony 6/26/17 12:18 PM
Deleted: T

Tony 6/29/17 12:19 PM
Deleted: was

Sophie Valcke 6/29/17 3:39 PM
Deleted: mapped

Tony 7/1/17 1:38 AM
Formatted: Font:Times New Roman, Font color: Auto

Sophie Valcke 6/28/17 8:23 PM
Deleted: couple

Sophie Valcke 6/28/17 8:24 PM
Deleted: s'

Tony 6/26/17 12:16 PM
Deleted: not

Tony 6/26/17 12:16 PM
Deleted: by the coupling cost but

3.4 Field Aggregation

OASIS3-MCT provides a new feature, as described in Section 2.2, that allows users to aggregate coupling of multiple fields into a single coupling operation by specifying coupled fields via colon delimited field names in the *namcouple* file. Table 2 shows unbarriered and barriered ping-pong and barriered mapping timing for the T799-ORCA025 configuration on Lenovo using single and multiple fields. For the barriered case, MPI barriers were added before the send and before the mapping in each component in both directions of the coupling to strictly enforce serialization of operations and to be able to time the mapping cost cleanly. Times are in seconds for the slowest task over the entire run. The fastest time from two test runs is shown. Variability between runs is less than 2%. The columns in table 2 are for a configuration with 180 cores per component using *src+bfb* map settings for a single field, 10 fields coupled via 10 coupling calls, 10 fields coupled via a single coupling communication, and 10 fields bundled into a single variable. The bundled fields option will be available in the OASIS3-MCT_4.0 release. The barriered pipo (ping-pong) time in table 2 is about 50% greater than the unbarriered time. The significant performance penalty with barriers suggests that there is normally some overlap of coupling communication and mapping in these timing runs when running without barriers.

The unbarriered pipo time in table 2 shows that coupling 10 fields performs proportionally better than coupling a single field. More specifically, the case with 10 fields coupled with 10 coupling calls performs best, likely because there is a greater chance to overlap mapping and coupling communication in this case since each fields is mapped and sent independently. The barriered pipo time further suggests that the case with 10 fields coupled with 10 coupling calls has the greatest amount of overlapping work because that case has the largest performance degradation when barriers are turned on.

In contrast, the mapping time for 10 fields coupled via a single operation is faster than mapping 10 fields one at a time. This is expected as the underlying implementation aggregates the mapping rearrangement and coupling communication cost when fields are bundled. But in this case, that mapping advantage is offset by the ability to overlap less work. This simple test case carries out coupling without any real model work between calls. In a real model, the coupling performance will depend on the sequence of the coupling calls within the model, how much work can be overlapped with coupling, and the relative core counts and grid sizes of the different coupling fields.

3.5 Conservation

Table 3 shows the timings of a ping-pong test of the T799-ORCA025 case on the Lenovo cluster for four different configurations (48 and 180 cores with *src* or *dst* mapping) with CONSERV unset and CONSERV set to *lsum8* (equivalent to *opt* in OASIS3-MCT_3.0), *lsum16*, *ddpdd*, *reprosum*, and *gather* (equivalent to *bfb* in OASIS3-MCT_3.0). The CONSERV implementation and a description of the different options for

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

Tony 6/21/17 2:19 AM
Deleted: s… and 3…mapping and … shows the mapping time on … [17]

Tony 6/21/17 2:22 AM
Formatted … [18]

Tony 6/21/17 2:52 AM
Deleted: In general, the time to couple 10 fields is proportionally less than the time to couple 1 field, and a clear advantage is seen in the 10 field mapping cost when done as a single aggregated operation. The time for mapping a bundle field of dimension 10 is similar to the time for the 10 fields coupled via a single coupling; this is expected because the underlying implementation is basically the same.

Table 3 shows the ping-pong times for the same cases as table 2. In this case, the advantage of aggregating multiple coupling fields is less clear. For the *dst+bfb* case, the single operation performs only marginally better while for the *src+bfb* case, the single operation performs slightly worse. …In this case, t…e…coupling calls are happening sequentially … [19]

Sophie Valcke 6/29/17 5:07 PM
Deleted: a

Tony 6/21/17 2:53 AM
Deleted: the

Tony 6/26/17 1:53 PM
Deleted: 4

Tony 7/1/17 2:29 AM
Formatted … [20]

Tony Craig 6/16/17 6:49 PM
Deleted: and cases …, … [21]

Tony 6/26/17 9:02 AM
Deleted: as well as

Tony Craig 6/16/17 6:50 PM
Formatted … [22]

Tony Craig 6/16/17 6:48 PM
Deleted: CONSERV set to *opt*, and CONSERV set to *bfb*

the computation of the global sums are described in Section 2.4. Times are accumulated over 1000 ping-pongs for a single coupling field in each direction. Two trials of each case were carried out and the minimum time is shown. Differences between trials were less than 2% except for the gather case where variations in time of up to 10% were observed. The CONSERV operation increases the pipo time by at least 50% regardless of the method used compared to CONSERV off (unset), and the *gather* option is at least an order of magnitude more expensive than other CONSERV methods. When OASIS3-MCT_4.0 is available, *lsum8* will still be the fastest CONSERV method while *reprosum* will be the best bit-for-bit option. The cost of *reprosum* is only slightly higher than *lsum16*, but reproducibility characteristics are significantly better. When using CONSERV, it is important to test the performance of various methods and consider carefully the requirements of the science. Of course, when possible, mapping weights that are inherently conservative such as area overlap conservative (Jones, 1999) should be used to avoid use of the CONSERV operation all together.

3.6 Memory

Figure 5 shows the memory use per core for the T799-ORCA025 test case on Curie, the same test case as in figures 2 and 3. Memory use was determined by calls into the gptl (http://jmrosinski.github.io/GPTL/) interface, included in the OASIS3-MCT release, which queries memory usage through C intrinsics. At 16,000 cores, the infrastructure is using a bit more than 1GB per core, which while not tiny, is generally acceptable for many applications and hardware. Memory is increasing on a per core basis at higher core counts. It is possible that the MPI memory footprint is accounting for most of this behavior (Balaji et al, 2008, Gropp, 2009), but further investigation will be carried out in the future to better understand this behavior.

4. Conclusions

OASIS3-MCT was implemented largely to address limitations in parallel performance of OASIS3 and to provide a framework for use at higher resolutions. With OASIS3-MCT, the widely used OASIS3 model interfaces (APIs) and configuration file have largely been preserved, and this explains the wide adoption of OASIS3-MCT within the OASIS user community. Since its release in May 2015, about 250 downloads of OASIS3-MCT_3.0 were registered from most major climate modeling groups in Europe as well as from groups in North and South America, Asia, Australia, and Africa. In the last two years, the OASIS3-MCT coupler was used in many state-of-the-art coupled systems including high resolution climate models and systems that couple 3-D atmospheric fields between global and regional models frequently among others. Other examples of coupled model applications that use OASIS3-MCT can be found on the OASIS3-MCT

Tony Craig 6/16/17 6:48 PM
Deleted: its *bfb* and *opt* options …Table 4 shows that …t … [23]

Sophie Valcke 6/29/17 5:09 PM
Deleted:

Tony 7/1/17 1:55 AM
Deleted: and

Tony Craig 6/16/17 7:43 PM
Formatted: Font:Not Bold, Italic, Font color: Auto

Tony 7/1/17 1:55 AM
Deleted: stands out …with respect to cost…will be … [24]

Tony Craig 6/16/17 7:44 PM
Formatted … [25]

Tony Craig 6/16/17 6:54 PM
Deleted: , even when the global sum option is set to *opt*, adds a cost to the coupling. However, beyond that, the difference in cost between the *opt* and *bfb* CONSERV settings are significant for this high resolution case because the *bfb* option gathers the entire field on the root process while the *opt* routine uses a more parallel algorithm and only gathers a scalar from each task. … therefore…both *opt* and *bfb* and decide whether it's absolutely necessary to use *bfb*… (… [26]

Tony 6/26/17 1:55 PM
Deleted: 6

Sophie Valcke 6/29/17 5:12 PM
Deleted: F

Tony 6/26/17 1:56 PM
Deleted: 3…4 … [27]

Sophie Valcke 6/29/17 5:13 PM
Deleted: and is included in the OASIS3-MCT release… … [28]

Tony 6/29/17 12:22 PM
Deleted: ,…'s likely … [29]

Sophie Valcke 6/29/17 5:13 PM
Deleted: need to

Tony 6/26/17 11:46 AM
Deleted: . The fundamental datatypes and arrays in OASIS3-MCT, such as fields and partitions, are generally quite memory scalable as implemented, and we do not believe the memory increase with core count is coming primarily from OASIS3-MCT.

coupled model page[2].

The underlying software was refactored significantly in OASIS3-MCT to improve parallel performance and coupling capabilities. MCT serves as a key part of the OASIS3-MCT implementation and provides parallel capabilities for coupling operations. OASIS3-MCT_3.0 also provides new capabilities to couple fields within a single component running on concurrent, overlapping, or partially overlapping processes. This increases the flexibility of OASIS3-MCT significantly and provides a mechanism for coupling data between different decompositions or grids within a single model among many other things. OASIS3-MCT can now be used as a coupling layer for components running sequentially, concurrently or both; for single or multiple executable execution; to exchange coupling fields defined on a subset of the component tasks; and to support features like a separate I/O component included in the executable but not involved in the coupling. This provides significant flexibility to layout models on parallel tasks in relatively arbitrary ways to optimize overall performance and to build new features into a model beyond model coupling. OASIS3-MCT has been tested at high resolution, at high processor counts, and with a large number of coupling fields successfully.

There are other benefits in the OASIS3-MCT implementation. OASIS3-MCT still supports mapping weights generation on-the-fly via SCRIP using a single processor just like OASIS3. However, mapping files can also be generated offline, read in directly relatively efficiently, more easily reused, and the cost associated with generating the mapping files can be moved to a preprocessing step using more sophisticated tools. If online weights generation needs to be upgraded in OASIS in the future to support, for instance, time evolving grids, OASIS will consider incorporating more sophisticated external tools into the infrastructure. There are new features that support creating grid data using a parallel interface, that couple multiple fields in a single operation, and that generate the *namcouple* file offline via a GUI. The requirement for an OASIS3 hub coupler has been removed and all communication and mapping is done in parallel and performance is significantly improved.

The scaling and performance results in Section 3 demonstrate the ability of OASIS3-MCT to support high-resolution model coupling on large core counts. However, as core counts get well into the tens of thousands and beyond, there are questions and concerns about the cost of both the initialization and coupling exchanges in OASIS3-MCT. The operations in OASIS3-MCT are ultimately constrained by MPI performance at those core counts, and developers will continue to pursue performance improvements in the underlying implementation. However, for the near term future, say the next 5 years, OASIS3-MCT is likely to adequately meet the needs of the climate modeling community.

[2] https://portal.enes.org/oasis/oasis-dedicated-user-support-1/survey-on-coupled-models-using-oasis-march-2016/coupled-models-using-oasis

Tony 6/26/17 2:40 PM
Deleted: or between components

Tony 7/1/17 2:01 AM
Deleted:

Tony Craig 6/15/17 11:57 AM
Deleted: M

Tony Craig 6/15/17 11:57 AM
Deleted: be

Sophie Valcke 6/29/17 6:42 PM
Deleted: where

Sophie Valcke 6/29/17 6:42 PM
Deleted: are available

Tony Craig 6/15/17 11:52 AM
Deleted: , but

Tony Craig 6/15/17 11:57 AM
Deleted: OASIS3-MCT still supports mapping weights generation on-the-fly via SCRIP using a single processor just like OASIS3

Tony Craig 6/15/17 11:58 AM
Deleted: .

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

Tony Craig 6/16/17 11:17 AM
Deleted: oupling

Tony Craig 6/7/17 5:26 PM
Deleted: 10

Tony 6/26/17 9:06 AM
Deleted: need to

Sophie Valcke 6/29/17 6:40 PM
Deleted: https://verc.enes.org/oasis/oasis-dedicated-user-support-1/some-current-coupled_-models/some-oasis3-mct-coupled-models

The flexibility and relative cost of OASIS3-MCT to map fields by various approaches was shown. A general recommendation is to test different approaches and to choose the approach that yields the best performance. While it is always first recommended to use conservative mapping weights to avoid the use of the global CONSERV transformation, the performance of the different options of this transformation were shown for a high-resolution case. If the CONSERV transformation is needed, the more efficient *lsum8* (*opt* in OASIS3-MCT_3.0) option, implemented using partial sums, is recommended unless bit-for-bit reproducible results on different core counts are absolutely required. The partial sum option will produce bit-for-bit reproducible results for a configuration with fixed process counts and decomposition and will introduce no more than roundoff level differences when changing process counts or decomposition. CONSERV options planned for the OASIS3-MCT_4.0 release were also included in the results in section 3. In OASIS3-MCT_4.0, the new *reprosum* option will significantly improve the performance of the bit-for-bit CONSERV option compared to the currently available *gather* (*bfb* in OASIS3-MCT_3.0) option.

The ability to couple multiple fields via a single coupling operation was demonstrated. While not shown in this study, OASIS3-MCT has been used to successfully couple over 10000 fields in some coupled systems within the community. Those tests were carried out both with single field coupling and multiple field coupling with success. In that case, multiple field coupling significantly reduces the size of the *namcouple* file. Multiple field coupling was shown to reduce the mapping time compared to coupling the same number of fields individually. The performance benefit of using the multiple field feature in the overall coupling time is less clear and will depend on the sequencing and design of each coupled system.

A number of future extensions are being considered for OASIS3-MCT. In theory, it should be possible to combine the mapping and coupling steps to eliminate a field rearrangement and further reduce communication cost. As a first step, decomposition strategies that could reduce the rearrangement cost in the mapping operation are being developed for release in OASIS3-MCT_4.0. There are also many opportunities in OASIS3-MCT to improve the I/O performance. In the current version, I/O is done via a gather and/or scatter to/from a root task and data is written in serial from the root task. This is likely to eventually lead to memory and performance issues. Finally, better support within OASIS3-MCT for shared memory threading (i.e. OpenMP) and on various multi-core architectures is likely to become more important in the future.

In summary, OASIS3-MCT_3.0 is the latest released version of the OASIS coupler. OASIS3-MCT extends the well-used OASIS software with backwards compatibility with regard to usage, but has an entirely new implementation internally. It provides the functional capability to couple high resolution structured or unstructured grids at high core counts successfully and should serve the community well for the next several years. The underlying implementation continues to be improved, and OASIS3-MCT_4.0 is expected to be ready for release in 2018.

Tony Craig 6/7/17 5:27 PM
Deleted: fastest

Tony Craig 6/16/17 7:46 PM
Formatted: Font:Not Bold, Italic, Font color: Auto

Tony 6/26/17 2:19 PM
Formatted: Font:Not Italic

Tony 6/26/17 9:08 AM
Deleted: nevertheless

Tony 7/1/17 2:04 AM
Deleted: An update of the

Tony 7/1/17 2:05 AM
Deleted: implementation

Tony 6/26/17 9:07 AM
Deleted: expected with

Tony 7/1/17 2:05 AM
Deleted: as

Tony Craig 6/16/17 7:47 PM
Deleted: s is planned in the future to improve performance

Tony 6/26/17 9:09 AM
Formatted: Font:Not Bold, Italic, Font color: Auto

Tony 6/26/17 9:09 AM
Formatted: Font:Not Bold, Italic, Font color: Auto

Tony 6/26/17 1:47 PM
Formatted: Font:Italic

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

Tony 7/1/17 2:08 AM
Deleted: If that is technically difficult

Tony 7/1/17 2:07 AM
Deleted: there are several

Tony 7/1/17 2:08 AM
Deleted: be introduced to

Tony 6/29/17 1:00 PM
Deleted: The performance and memory scaling of initialization and coupling may become a larger problem as resolutions and core counts continue to grow.

Tony 6/26/17 1:49 PM
Deleted: It

Code Availability

The OASIS3-MCT source code is available for use and testing after registration at https://portal.enes.org/oasis/download. The SVN command line to download OASIS3-MCT_3.0 is "svn checkout https://oasis3mct.cerfacs.fr/svn/branches/OASIS3-MCT_3.0_branch/oasis3-mct". The OASIS3-MCT_3.0 source code is also available as a tar file at ftp://ftp.cerfacs.fr/pub/globc/exchanges/distrib-oasis/oasis3-mct.tar.gz.

Appendix A

The following list provides a history of changes to OASIS3-MCT since OASIS3 up to OASIS3-MCT_3.0. It also includes an initial list of some features expected in the next release, OASIS3-MCT_4.0.

OASIS3-MCT_1.0 (2012):

- requirement for separate coupler processes and hub removed
- use of MCT in underlying coupling layer for regridding and communication
- parallel remapping
- fully parallel communication
- ability to couple a single field to multiple destinations
- extended ability to read mapping file
- improved deadlock trapping
- only MPI1 job launching supported
- ability to couple on a subset of processes
- support for one-dimensional coupling field arrays
- support for prism_ and oasis_ interface names
- restart files for LOCTRANS operations
- coupling multiple fields through a single *namcouple* entry

OASIS3-MCT_2.0 (2013)

- support for bicubic interpolation given the field gradient is specified in the interface arguments
- coupling support on a subdomain of the full grid
- update to timing and debugging capabilities
- parallel interface to grid writing

OASIS3-MCT_3.0 (2015)

- improved memory use, initialization cost and scaling
- updated mapping file reading algorithm
- ability to implement a coupled system within a single executable

Sophie Valcke 6/29/17 6:47 PM
Deleted: https://verc.enes.org/oasis/download.

Tony 7/1/17 2:10 AM
Deleted: get the

Tony 7/1/17 2:11 AM
Deleted: sources

Tony 7/1/17 2:11 AM
Deleted: s are

Tony 6/29/17 12:25 PM
Deleted: and

Tony 6/29/17 12:24 PM
Deleted: up to the future

Sophie Valcke 6/29/17 6:48 PM
Deleted: 3

Tony 6/29/17 12:51 PM
Deleted: namcouple

Tony 6/29/17 12:51 PM
Formatted: Font:Italic

- ability to couple sequentially and on partially or completely overlapping processes

OASIS3-MCT 4.0 (2018?)

- support for bundled coupling fields
- additional CONSERV global sum methods and improved CONSERV bit-for-bit performance
- a new option for decomposing the mapped field to reduce communication cost
- an update to a newer version of MCT that may improve initialization performance

Tony Craig 6/16/17 10:41 AM
Deleted:

Unknown
Formatted: Bullets and Numbering

Tony Craig 6/16/17 10:41 AM
Formatted: Normal, No bullets or numbering

Tony 6/26/17 1:50 PM
Deleted: TBD

Tony Craig 6/16/17 10:42 AM
Formatted: Bullets and Numbering

Tony 7/1/17 2:12 AM
Formatted: Bullets and Numbering

Acknowledgements

This research was supported by the ESiWACE H2020 European project grant agreement No 675191 (www.esiwace.eu), the IS-ENES2 FP7 European project contract number 312979 (https://verc.enes.org/ISENES2), and the CONVERGENCE project funded by the French National Research Agency ANR-13-MONU-0008.

References

Balaji, P., D Buntinas, D. Goodell, W.D. Gropp, S. Kumar, E.L. Lusk, R. Thakur, and J.L. Traff: MPI on a Million Processors. Recent Advances in Parallel Virtual Machine and Message Passing Interface, Volume 5759 of the series Lecture Notes in Computer Science, pp 20-30, doi:10.1007/978-3/642-03770-2_9, 2008.

Craig, A. P., Vertenstein, M., and Jacob, R.: A New Flexible Coupler for Earth System Modeling developed for CCSM4 and CESM1, Int. J. High Perf. Comp. App., 26(1), 31–42, doi:10.1177/1094342011428141, 2012.

Dauptain, A.: OpenTEA Super-User Guide, Dec, 16, 2014. http://oasis3mct.cerfacs.fr/svn/trunk/oasis3-mct/util/oasisgui/OpenteaSUG.pdf.

Gropp, W.: MPI at Exascale: Challenges for Data Structures and Algorithms, Proceedings of the 16th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface, editors M. Ropo, J. Westerholm, and J. Dongarra, published by Springer-Verlag, Berlin, doi: 10.1007/978-3-642-03770-2 3. 2009.

He, Y. and Ding, C.H.Q.: Using Accurate Arithmetics to Improve Numerical Reproducibility and Stability in Numerical Applications, The Journal of Supercomputing, 18:259, doi:10.1023/A:1008153532043, 2001.

Hollingsworth, A., Engelen, R. J., Textor, C., Benedetti, A., Boucher, O., Chevallier, F., Dethof, A., Elbern, H., Eskes, H., Flemming, J., Granier, C., Kaiser, J. W., Morcrette, J.-J., Rayner, P., Peuch, V. H., Rouil, L., Schultz, M. G., Simmons, A. J., and The GEMS Consortium: Toward a Monitoring and Forecasting System For Atmospheric Composition: The GEMS Project, B. Am. Meteorol. Soc., 89, 1147–1164, 2008.

Jacob, R., J. Larson, E. Ong, MxN Communication and Parallel Interpolation in CCSM3 Using the Model Coupling Toolkit.: Int. J. High Perf. Comp. App.,19(3), 293-307, doi:10.1177/1094342005056116, 2005.

Jones, P.: Conservative remapping: First-and second-order conservative remapping, Mon. Weather Rev., 127, 2204-2210, 1999.

Larson, J., R. Jacob, and E. Ong: The Model Coupling Toolkit: A New Fortran90 Toolkit for Building Multiphysics Parallel Coupled Models. Int. J. High Perf. Comp. App., 19(3), 277-292, doi:10.1177/1094342005056116, 2005.

Tony 6/26/17 2:06 PM
Deleted: 9

Tony 6/19/17 10:44 PM
Deleted: -1

Mirin, A.A. and P.H. Worley, Improving the Performance Scalability of the Community Atmosphere Model, Int. J. High Perf. Comp. App.,26(1), 17-30, doi:10.1177/1094342011412630, 2012.

Redler, R., S. Valcke and H. Ritzdorf: OASIS4 - A Coupling Software for Next Generation Earth System Modelling, Geosci. Model Dev., 3, 87-104, doi:10.5194/gmd-3-87-2010, 2010.

Terray, L., Thual, O., Belamari, S., Déqué, M., Dandin, P., Lévy, C. and Delecluse, P.: Climatology and interannual variability simulated by the arpege-opa model, Clim. Dynam., 11, 487–505, 1995.

Theurich, G., Deluca, C., Campbell, T., et al.: The Earth System Prediction Suite: Toward a Coordinated U.S. Modeling Capability. Bull. Amer. Meteor. Soc., http://journals.ametsoc.org/doi/abs/10.1175/BAMS-D-14-00164.1, 2016.

Valcke, S., A. Craig, R. Dunlap, and G. Riley: Sharing Experiences and Outlook on Coupling Technologies for Earth System Models, Bull. Amer. Meteor. Soc., Vol 97 No 3, ES53-ES56, doi:10.1175/BAMS-D-15-00239.1, 2016.

Valcke, S., The OASIS3 coupler: a European climate modelling community software. Geosci. Model Dev., 6, 373–388, doi:10.5194/gmd-6-373-2013, 2013

Valcke, S., T. Craig, L. Coquart: OASIS3-MCT User Guide, OASIS3-MCT_3.0, Technical Report, TR/CMGC/15/38, CERFACS/CNRS/SUC URA No1875, Toulouse, France, 2015. http://www.cerfacs.fr/oa4web/oasis3-mct_3.0/oasis3mct_UserGuide.pdf

Valcke, S., T. Craig, L. Coquart: OASIS3-MCT User Guide, OASIS3-MCT 1.0, Technical Report, TR/CMGC/12/49, CERFACS, Toulouse, France, 2012. http://www.cerfacs.fr/oa4web/papers_oasis/oasis3mct_UserGuide_1.0.pdf

Tony 6/19/17 10:43 PM
Deleted: DOI

Tony Craig 6/7/17 5:30 PM
Deleted: Valcke, S., V. Balaji, A. Craig, C. Deluca, R. Dunlap, R. Ford, R. Jacob, J. Larson, R. O'Kuinghttons, G. Riley, M. Vertenstein: Coupling technologies for Earth System Modelling, Geosci. Model Dev., 5, 1589-1596, doi:10.5194/gmd-5-1589-2012, 2012a

Tony Craig 6/15/17 11:40 AM
Deleted: b

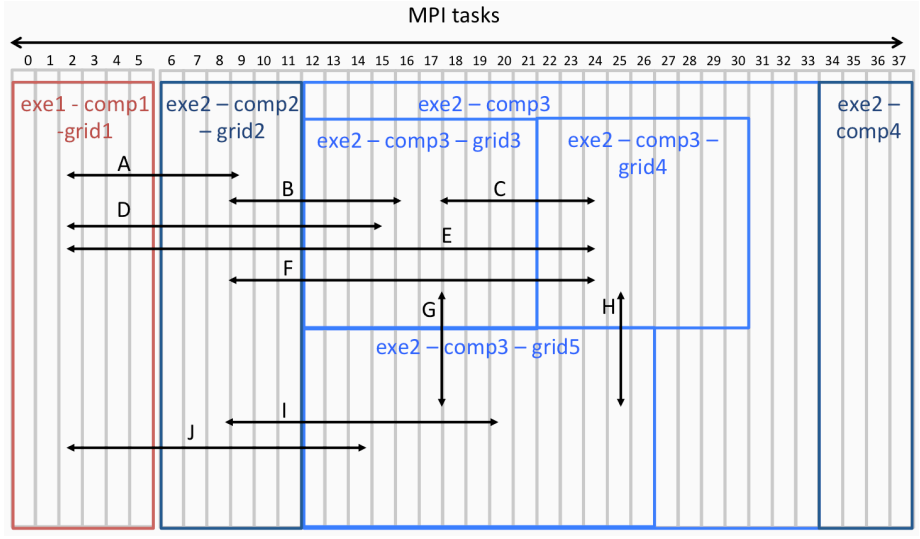


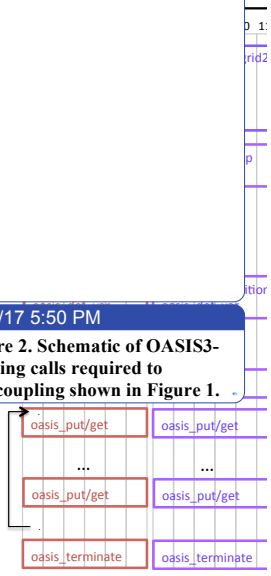
Figure 1. A schematic of the coupling capability in OASIS3-MCT_ 3.0. In this example, there are 2 executables, exe1 and exe2. Executable 2 has 3 components, comp2, comp3, and comp4, and comp3 has 3 grids, grid3, grid4, and grid5; comp4 is not involved in any coupling in this case. The executables, components, and grids are laid out across different tasks. Arrows indicate different coupling capabilities; A), D), E), and J) between different components in different executables; B), F), and I) in a single executable between different components with different grids; C) between different grids in a single component on non-overlapping tasks; G) between different grids in a single component on partially overlapping tasks; and H) between different grids in a single component on partially overlapping and partially non overlapping tasks.

Tony Craig 6/7/17 5:50 PM

Deleted:

Tony Craig 6/7/17 5:50 PM

Deleted: Figure 2. Schematic of OASIS3-MCT_3.0 coupling calls required to implement the coupling shown in Figure 1.



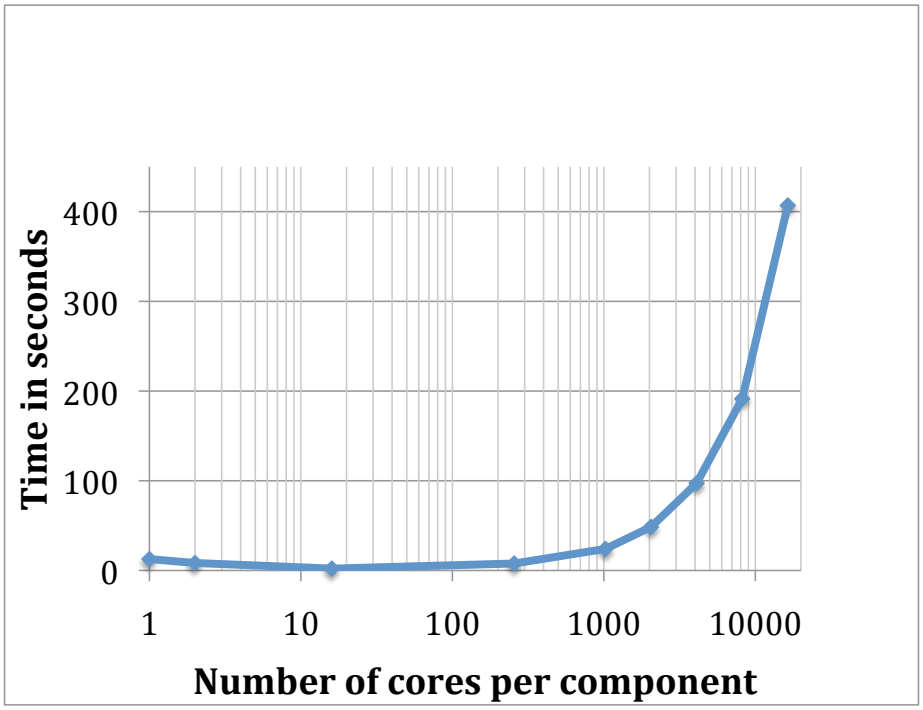


Figure 2. Initialization cost for the T799-ORCA025 toy model using OASIS3-MCT_3.0 on Curie Bullx.

Tony 6/26/17 1:54 PM
Deleted: 3

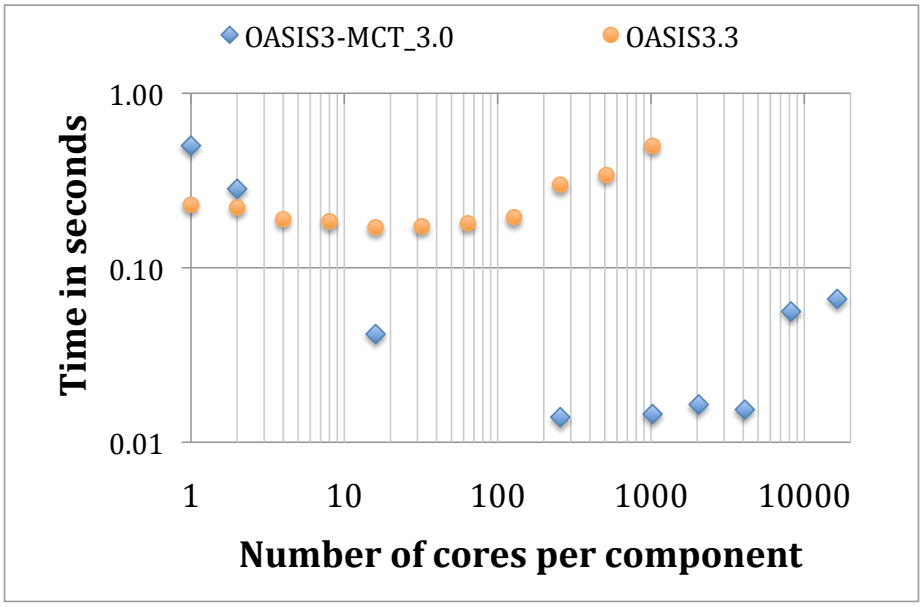


Figure 3. Comparison of the ping-pong (pipo) time for the T799-ORCA025 toy model for OASIS3.3 and OASIS3-MCT_3.0 on Curie Bullx. The time is averaged for a run where 1000 ping-pongs were carried out.

Tony 6/26/17 1:54 PM
Deleted: 4

Tony 6/26/17 8:55 AM
Deleted: P

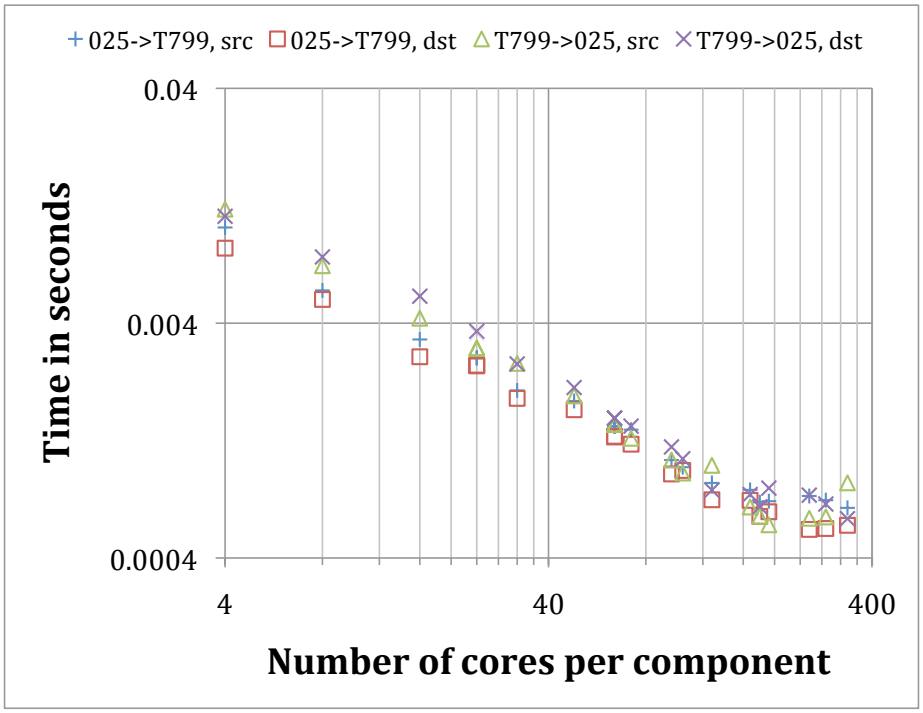


Figure 4. OASIS3-MCT_3.0 T799-ORCA025 mapping time versus core count per component on Lenovo. *src* and *dst* mapping are shown for both mapping directions using the *bfb* algorithm based on tests where 1000 ping-pongs were run.

Tony 6/26/17 1:54 PM
Deleted: 5

Tony 7/1/17 2:33 AM
Deleted: .

a) ORCA 025 cores	b) T799 cores	c) pipo time for mapping on *src* cores (s)	d) pipo time for mapping on *dst* cores (s)	e) pipo time for mapping on ORCA025 cores (s)	f) pipo time for mapping on T799 cores (s)
24	336	5.10	5.48	7.29	3.79
180	180	1.29	1.54	1.36	1.36
336	24	4.70	4.93	1.91	6.69

Table 1. Comparison of the ping-pong (pipo) time for the T799-ORCA025 toy model on Lenovo on 360 cores with both the relative core-count/component and the mapping location varied. The time is in seconds for 1000 ping-pongs. Columns a and b define the core-count used for each component of the toy model. Columns c-f are

the pipo times for 4 different mapping approaches c) mapping always on the source cores, d) mapping always on the destination cores, e) mapping on the ORCA025 cores, and f) mapping on the T799 cores.

time (seconds) mapping = *src+bfb*	1 field, 1 coupling	10 fields, 10 couplings	10 fields, 1 coupling	10 fields, 1 bundle
pipo time, no barriers	1.29	10.52	11.93	12.29
pipo time, with barriers	1.87	17.63	16.56	17.48
map ORCA025->T799, with barriers	0.67	5.48	4.61	4.68
map T799->ORCA025, with barriers	0.56	5.28	4.76	4.81

Table 2. Comparison of unbarriered and barriered ping-pong (pipo) and barriered mapping time for the T799-ORCA025 toy model on Lenovo on 180 cores per component for coupling of 1 field, coupling of 10 fields one at a time, coupling 10 fields using OASIS3-MCT multiple-coupling-field capability, and coupling of 10 fields by a single 3-D bundle. All times are for *src+bfb* mapping for 1000 ping-pongs. For barriered times, MPI barriers were introduced in both components before the send and before the mapping to force serialization of work and to time the mappings separately.

cores mapping	CONSERV unset	CONSERV *lsum8*	CONSERV *lsum16*	CONSERV *ddpdd*	CONSERV *reprosum*	CONSERV *gather*
48, *src+bfb*	4.00	8.27	16.78	10.65	17.34	117.72
48, *dst+bfb*	4.39	8.02	16.59	10.42	16.98	142.12
180, *src+bfb*	1.25	2.21	4.59	2.87	4.85	126.91
180, *dst+bfb*	1.56	2.26	4.62	2.92	4.90	130.01

Table 3. Comparison of ping-pong (pipo) times for the T799-ORCA025 toy model on Lenovo on 48 and 180 cores per model with the CONSERV option off (unset), set to *lsum8* (*opt* in OASIS3-MCT 3.0), *lsum16*, *ddpdd*, *reprosum* and *gather* (*bfb* in OASIS3-MCT). Times are accumulated over 1000 ping-pongs for a single coupling field in each direction.

Deleted: mapping

Tony 6/21/17 2:06 AM
Deleted: 9 ... [30]

Tony 6/21/17 2:07 AM
Deleted: 60 ... [31]

Tony 6/26/17 8:39 AM
Deleted: …on ... [32]

Tony 6/26/17 8:53 AM
Deleted: .

Tony 6/26/17 8:53 AM
Deleted: pipo time ... [33]

Tony 6/26/17 8:53 AM
Deleted: Table 3. Comparison of p ... [34]

Tony Craig 6/7/17 5:36 PM
Deleted: pes

Tony Craig 6/7/17 5:08 PM
Deleted: off

Tony Craig 6/16/17 3:40 PM
Formatted ... [35]

Tony Craig 6/16/17 3:37 PM
Deleted: *opt*

Tony Craig 6/16/17 3:40 PM
Formatted ... [36]

Tony 6/26/17 8:42 AM
Deleted: 3.67

Tony Craig 6/16/17 3:44 PM
Deleted: 4.04

Tony 6/26/17 8:43 AM
Deleted: 5.16

Tony Craig 6/16/17 3:44 PM
Deleted: 7.56

Tony 6/26/17 8:43 AM
Deleted: 1…1.40 ... [37]

Tony 6/26/17 8:42 AM
Deleted: 2.96

Tony Craig 6/16/17 3:44 PM
Deleted: 4.41

Tony 6/26/17 8:43 AM
Deleted: 5.39

Tony Craig 6/16/17 3:43 PM
Deleted: 7.33

Tony 6/26/17 8:43 AM
Deleted: 1…2.15 ... [38]

Tony 6/26/17 8:42 AM
Deleted: 1….02 ... [39]

Tony Craig 6/16/17 3:44 PM
Deleted: 1.29

Tony 6/26/17 8:43 AM
Deleted: 1.57

Tony Craig 6/16/17 3:43 PM
Deleted: 2.11

Tony 6/26/17 8:43 AM
Deleted: 3.47 ... [40]

Tony 6/26/17 8:42 AM
Deleted: 1.04

Tony Craig 6/16/17 3:44 PM
Deleted: 1.61

Tony 6/26/17 8:43 AM
Deleted: 1.39

Tony Craig 6/16/17 3:43 PM

Tony 6/26/17 8:43 AM
... [41]

Tony 6/26/17 1:52 PM

Tony Craig 6/7/17 5:37 PM
... [42]

Tony Craig 6/16/17 3:39 PM

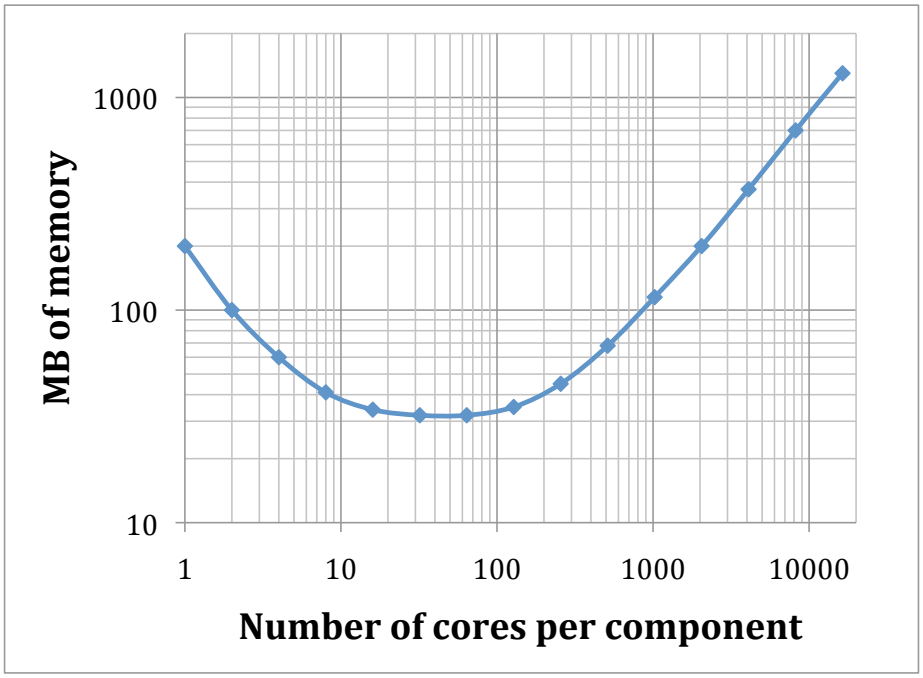


Figure 5. OASIS3-MCT_3.0 memory use on Curie Bullx for the T799-ORCA025 toy model as a function of cores per component.

Tony 6/26/17 1:54 PM
Deleted: 6

Tony 7/1/17 2:28 AM
Deleted:

Tony 7/1/17 2:28 AM
Deleted: -count

Tony 7/1/17 2:28 AM
Deleted: /