Dear Editor, Dear Referee,

We would like to thank you very much for your positive feedbacks and detailed suggestions to improve our revised manuscript.

We reply to these comments individually below and will also address them where possible in the newly revised version (marked-up version, line numbers mentioned below refer to this version). Reviewer comments are reproduced in **bold**, with our explanations in *italic*. We also listed the further changes regarding to language improvements at the end.

Wenkui He

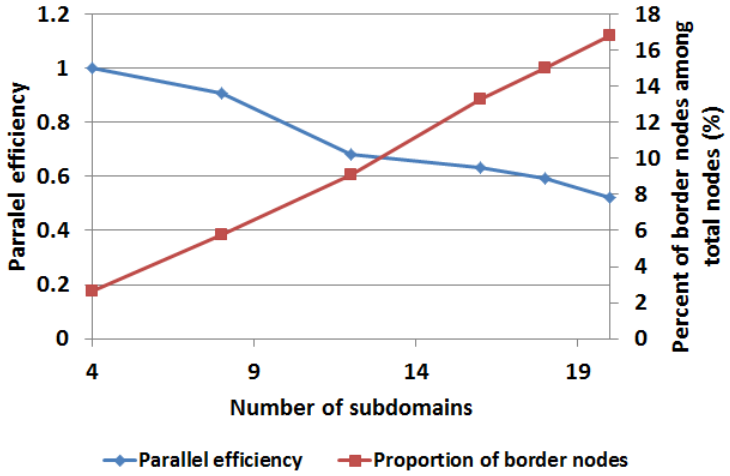
On behalf of all authors

Response to Glenn Hammond

1	**Thank you for your responses to the previous questions. I am satisfied with the responses. I apologize for the additional questions below, but there is so much new content in the revised paper that I have additional questions/concerns.**
	Thank you very much for your positive feedback and careful review. We improved our manuscript further based on your comments. We address these comments individually below.
2	**The grammar in the paper is awkward in many locations. I highly recommend that a native English speaker edit/revise some of the wording. Several situations are listed below, but there are many more. This should not be too difficult.**
	Thank you for improving the language. Besides correcting the sentences mentioned by you, we revised the whole manuscript with the help of a native English speaker. These changes are listed at the end.
3	**(Page) 2: Line 4-5: "…and more straightforwardly on high performance computers" – needs to be revised as this does not make sense. Easier?**
	We changed "more straightforwardly" as "easier" (2: Line 4).
4	**2: Line 10: "almost any chemical reaction"? How about "many chemical reactions" or "numerous chemical reactions"?**
	Thank you, we think "numerous chemical reactions" is better (2: Line 7).
5	**3:Line 8: "coupling of different codes…is an indispensable choice" – this is solely opinion.**
	We deleted "an indispensable choice and" (3: Line 10).
6	**4: Line 7: Since Hammond et al., 2012 is cited but not included in the references, you may want to review all citations to ensure that the references exist at the end.**
	Thank you very much for the careful check. We apologize for our careless mistakes. We added the reference for this citation, and checked all the other citations and references. *We corrected the citation "Kosakowski and Watanabe, 2013" as "Kosakowski and Watanabe, 2014" in 2:Line 25; 3: Line 15, 24 and 6: Line 22. We corrected the reference in 27: Line 4.*
7	**5:Line 4: "therefore a speedup for flow/transport is not given anymore". Should this be "is not experienced anymore".**
	We changed "not given anymore, if…" as "no longer experienced when …" (5: Line 7).
8	**5:Line 12: "flexible distribution of different amount of computer resources" -> amounts?**

	We corrected "amount" as "amounts" (5: Line 18).
9	**6:Line 25: "It provides a well-defined series of methods" How about "it provides an API (Application Programming Interface)".**
	Thank you for the nice suggestion, we changed the sentence as "it provides an application programming interface (API)" (7: Line 7).
10	**7: Line 18: The reference to CMake is out of the blue. How about calling it the build system? Otherwise, you need to explain CMake.**
	We changed "the CMake file" as "the build system" (8: Line 1).
11	**9: Line 19: Lasaga type rate law -> it is referred to as "transition state theory" (and then cite Lasaga)**
	Thanks, we adopted your suggestion (10: Line 3).
12	**10: Line 5: "IPhreeqc interface and the overhead involved in calling IPhreeqc are 12.7% and 3.8%". Can you point me to where you explain/define the difference between the interface and overhead? I assume that the initialization of the thermodynamic database is in the overhead….**
	In Sect. 2.3 (9: Line 11) we defined the compositions of the overhead involved in calling IPhreeqc. Yes, the loading of the thermodynamic database is included in the overhead. To make it clearer to the reader, we added • *explanation of the time consumption for the interface as "(including the preparation of input for IPhreeqc and the processing of output from IPhreeqc)", after "IPhreeqc interface" (10: Line 18);* • *"(described in Sect. 2.3)" after "the overhead involved in calling IPhreeqc" (10: Line 19).*
13	**11: Line 12: The description of domain decomposition leads me to believe that you are using Richardson iteration (i.e. where one solves the local subdomain and then updates the ghost/halo values), but I find that hard to believe. With domain decomposition, the linear/nonlinear system is solved across the entire domain for each iteration. If you are truly iterating between subdomains, I highly recommend using the full implicit solve as it would converge much faster. 11: Line 24: Again, this doesn't sound like traditional domain decomposition.**
	We are using the domain decomposition for partitioning the computing tasks of • *assembly of subdomain matrices and vectors* • *solving global equation system parallel with iterative Krylov subspace equation solver* *With the current approach, the stiffness matrices and vectors are assembled within subdomains, and the Krylov subspace solver collects the norms of the production of the subdomain matrices and vectors to compute a converged solution.* *More details of the DDC approach can be found in Wang et al. (2009).* *To avoid misunderstanding we renamed the section title as "Application of the DDC approach of OGS" and rewrote the whole part as:* *"The domain decomposition approach (DDC) is applied to partition the computational tasks of the global assembly and the linear solver implemented in OGS (Wang et al, 2009).* *For the current DDC approach, METIS is used as a preprocessing tool to partition mesh in order to balance the node quantities and minimize the border nodes among subdomains efficiently. With the partitioned mesh data, the stiffness matrix and the right-hand side vector of the system of linear equations are only assembled within subdomains by individual compute cores. Then these assembled subdomain matrices and vectors are taken to compute a converged solution with iterative solvers. By this way, the computational tasks of the global assembly and the linear solver are parallelized straightforwardly."*
14	**12: Line 21: "a single input string will be prepared" – is there a separate string for each cell (all concentrations in each cell) or is there a single string for all concentrations in all cells being**

	solved on a core?
	In a serial simulation, only one input string will be prepared for all nodes (cells) during each time step. In a parallel simulation, one input string will carry the concentration values for the nodes being solved on a specific compute core. We modified the sentence "a single input string will be prepared" as "only one input string will be prepared for all finite element nodes" (13: Line 15). *We rewrote the sentence ", whose number is equal … employed." as ". Each string carries the information for the nodes being solved on a specific compute core." (13: Line 18).*
15	**13: Line 11: Are the concentrations in the strings full (double) precision?**
	For the presented simulations, the output string is set to scientific notation with precision 12. We changed this in the meanwhile to 16.
16	**13: Line 28: An MPI_Allreduce() creates a copy of the full buffer for each process. So, you are generating a full global concentration vector for each core? Are you sure that this is not a gather or scatter operation? That seems more reasonable.**
	Yes, we used MPI_Allreduce(). The reviewer is right that it will generate a full global concentration vector for each core. In the current implementation, the grouping of the nodes for solving geochemical reactions is independent of DDC (described in 14: Line 13). We use MPI_Allreduce() as a straightforward solution to collect concentrations from different local buffers and then return the concentrations back to different cores for the following calculation of mass transport. In this way, we do not need to consider the complex tracking of indices when updating concentration values in different cores. *Nevertheless, we are aware that more sophisticated ways should be developed to minimize the memory usage and communication among different cores. We are grateful for the suggestions of the reviewer and will try to use gather or scatter operation to improve our code.* *In the manuscript, we added the following text after "MPI_Allreduce method" (14: Line 24):* *"(it's a straightforward solution for the current implementation. A more sophisticated approach, however, should be implemented to minimize the inter-processor communication and memory usage)"*
17	**15: Line 21: To me it seems that from Figure 8b, the best (fastest) combination would be 18 subdomains (best flow and transport speedup) + 20 geochemistry cores (best chemistry speedup). Just a comment.**
	Thank you for the careful observation, that's right. In 16: Line 25, we changed "when the number of compute cores exceeds 16" into "when 18 DDCs are applied".
18	**15: Line 28: My experience is that parallel efficiency can degrade significantly well before the number of border (ghost) nodes approaches the total number of nodes.**
	In response to this comment, we evaluated the parallel efficiency (for flow and transport) and the number of border nodes for different DDCs for this 2D example. Based on the results we obtained (see the figure below), we think the reviewer is right and modified the whole sentence as follows (16: Line 27): *"In this example, the parallel efficiency for solving flow and mass transport degrades already when more than 8 DDCs are employed, for which the border nodes only account for around 6% of the total nodes. Further increase of the number of DDCs up to 20, yielding 17% of border nodes, decreases the parallel efficiency down to 0.5 almost linearly."*

	
19	**16: Line 14: Averagely -> On average?**
	Yes, it should be "On average," (17: Line 19).
20	**16: Line 16: Let the reader come to this conclusion on their own as some may disagree.**
	We removed the whole sentence (17: Line 21).
21	**18: Line 9: The addition of the uranium leaching problem, an example that clearly illustrates the benefit of this algorithm, greatly improves the paper.**
	Thank you for the comment.
22	**20: Line 14: I believe that where this new algorithm will pay off is with massively parallel computation (hundreds of thousands of cores) where linear solver performance is poor. In those cases, one would want to use fewer cores to solve flow and transport, as conventional parallel preconditioners tend to degrade in performance at extremely large core counts.**
	Thank you for the comment. Due to the limited computational resources available (maximum 300 cores for each user) we could not test our approach for massively parallel computation in the present study. Nevertheless, your argument is a strong encouragement for our future researches and developments, when more computational resources become available. We added the following sentence in order to be more precise (21: Line 17): *"and a continuous speedup can be obtained (with the compute cores that are available) for the calculation of flow and mass transport," before "then using the conventional DDC approach …"*

Additional changes

Positions	*Changes*
5: Line 1	*"optimum amount" -> "optimum amounts"*
5: Line 15	*"reactions system" -> "reaction system"*
3: Line 9; 5: Line 21; 6: Line 17; 11: Line 9; 13: Line 4; 16: Line 12, 19, 27; 19:Line 29; 20:Line 22	*add the article "the"*
3: Line 10	*"applies" -> "applied to"*
3: Line 22	*remove the ","*

3: Line 26	*remove "scales"*
4: Line 27	*remove "additionally"*
4: Line 28	*"…open source, i.e." => "…open source; thus," "could" => "can"*
5: Line 5	*"is" => "are", "so" => "meaning"*
5: Line 8	*"an efficient … more compute cores." => "whereas the computation of the chemical system can see a further speedup with the addition of more compute cores, the computation of the transport problem may already reach a point of optimization, rendering the addition of further compute cores beyond this point inefficient."*
5: Line 14; 6: Line 13	*"applied for" => "applied to"*
5: Line 25	*remove "in the following"*
5: Line 28	*"apply … on" => "apply … to"*
6: Line 10	*"different kind of" => "different kinds of"*
6: Line 12	*"two phase" => "two-phase"*
6: Line 20	*"kinetically controlled biogeochemical-reactions" => "kinetically-controlled biogeochemical reactions"*
7: Line 3	*"widely used" => "widely-used"*
7: Line 5	*"Beside" => "Besides"*
9: Line 11	*"calling of functions" => "steps"*
9: Line 27	*"describes" => "shows"*
10: Line 4	*remove "the" before "dolomite"*
10: Line 13	*"Dolomite" => "dolomite"*
10: Line 30, 19: Line 17	*"Totally" => "In total"*
11: Line 13	*Remove "as much as"*
12: Line 24	*"decomposition procedure" -> "DDC procedure"*
14: Line 3	*"As a special case" => "In special cases"*
14: Line 14	*"independent from" => "independent of"*
15: Line 13	*"Of course jobs…" => "Jobs…"*
16: Line 2	*"differ with" => "differ from"*
17: Line 8	*", hence …" => "; hence,"*
17: Line 12	*add "," after "With a DDC=4"*
17: Line 18	*add "and" before "handing output…"*
18: Line 9	*"lower number of" => "fewer"*
18: Line 12	*add "this" after "The reason behind"*
18: Line 15	*"comparing" => "compared"*
18: Line 25	*"big" => "significant"*
19: Line 8	*"more close" => "closer"*
19: Line 21	*"A total …" => "The total…"*
19: Line 28	*"Best speedups" => "The best speedups"*
19: Line 25	*"between 8 to 16" => "between 8 and 16"*
20: Line 12	*"…coupling, when…"=> "…coupling when a"*
20: Line 13	*"in-" => "input-"*
21: Line 8	*"Particularly" => "In particular"*
21: Line 13	*"compute resources" => "computational resources"*
21: Line 21	*"become" => "becomes"*
21: Line 22	*"middle sized" => "middle-sized", "system" => "systems"*

21: Line 24	*“when further increase” => “when a further increase”*
22: Line 3	*“for large number of” => “for a large number of”*
22: Line 9	*add “in” after “involved”*
22: Line 14	*“remain as” => “remain”*
22: Line 15	*“resource was” => “resources were”*
23: Line 8	*“supports” => “support”*
23: Line 9	*“helps” => “help”*