Reply to reviewer 1 on "Interactive comment on "Scientific Workflows Applied to the Coupling of a Continuum (Elmer v8.3) and a Discrete Element (HiDEM v1.0) Ice Dynamic Model"

We are very grateful for the reviewer's comments towards improving our manuscript, and would like to respond to the main points below:

Reviewer:

"UNICORE may be a contribution to solving difficulties arising here, and with the portability of solutions, although, telling from the manuscript, the installation of UNICORE on a supercomputer seems to be far from trivial ("Maintaining a server-side deployment is not trivial because it needs a dedicated server that manages workflows and atomic jobs" (p. 23 ll 22), "The authors are grateful to [. . .] for his support and patience in making the UNICORE services available of CSC's computing resources").

Reply:

We would like to disagree on this point. While the UNICORE workflow deployment certainly is a prerequisite to our development, installation and configuration of any middleware - just like the time-intensive maintenance of HPC systems in general - is not the task of the user, but rather that of administrators at HPC centres (to whom we expressed our gratitude). The most challenging part was to study and analyse the coupling scenario as a whole, and then segregate the tasks from the available set of scripts and group them in different workflow elements so that any addition of new tasks in the future should not hurt the existing structure. UNICORE middleware in our application acts as an abstraction, i.e. a general-purpose HPC middleware solution to abstract multiple kinds of resource management systems with varying computing architectures. In other words, even if its installation requires one-time work by HPC administrators, the users have multiple benefits, most notably, neither the necessity of rewriting job scripts for various HPC systems nor the requirement to familiarize themselves with low-level details (e.g. file system capabilities, locations, or available memory and CPUs/cores). On top of that middleware resides a pluggable workflow management system that exploits multiple HPC submission sites by running complex multi-task and distributed workflows. We think it is a novel contribution to the field of glaciological sciences to show how scientific workflow management systems can support science and greatly ease users' work on distributed HPC infrastructures.

We have acknowledged the resource administrators here as they helped us not only in the deployment part, but also the post-deployment efforts, such as site-specific configurations performed once for many users, or application-and-workflow-derived requirements. In our revision of the manuscript, we will states more clearly that UNICORE is a prerequisite, just like the installation of the operating system, the deployed models (Elmer and HiDEM) and their underlying numerical libraries -- all of which require administrative work (just like setting up UNICORE), which however needs to be done only once for many users.

Reviewer:
In the end, the authors have managed to replace 400 lines of shell script (including job-headers) with a dependency on a high-level workflow management system with some services hosted at yet another supercomputing center. I am not convinced that a thorough clean-up/rewrite of the shell script would not have solved the main problems of the old script with way less effort and overhead.

Reply:
It seems that we need to emphasize the main point of the article more clearly: the abstraction (and hence decoupling from specific platforms) of the workflow. We admit that “a thorough clean-up/rewrite” may alleviate the main problems in a single instance on a single HPC system, however, it would still restrict the execution to this single HPC system the script is tailored to and executed on. In other words, its applicability would remain confined to a static environment. Any change in the setup of the underlying system would inevitably induce changes in the workflow-script and force the user (usually not the administrator) to permanently maintain her workflow code. UNICORE, on the other side, allows to build an abstraction layer on top of the underlying systems (mind the plural). This means that workflows are

a) Largely decoupled from changes in the operations of the underlying hardware (i.e., HPC cluster, supercomputer) and system software (e.g. job submission system, resource management system specifics);
b) Capable of running dedicated elements (such as Elmer and HiDEM) on computer systems that are tailored to the task (thereby increasing efficiency), and are specifically optimized and configured by HPC experts of the system once for many users;
c) Easy to maintain for the user, as the abstraction layer of UNICORE removes a large part of the maintenance effort associated with system updates from the user and shifts it towards system administrators. This helps to avoid error-prone manual script edits by users in time-consuming debug sessions, in order to understand underlying system changes.

We apologize if we missed to communicate these important points in sufficiently detail, and will clarify the manuscript with regard to the main motivation/advantage of deploying UNICORE in lieu of a single-instance script environment.

Reviewer:

The manuscript itself is somehow entangled between the scientific aspects of the ice dynamical problem, and the technical aspects of the software solution for job control on a set of supercomputers. It lacks a clear focus on either side, as can easily be seen from the mixture of plots that describe the glaciological problem (Figs. 1, 2, 3, 9) and the software solution for abstracting the scripting (Figs. 4-8, with Fig. 7 duplicating the right half of Fig 6). While we are introduced to the basic reasons why one would run the different glaciological models for the different sub-problems, and get some insights on the model grids, and which piece of code solves which sub-problem, there is no interpretation of the results shown in fig 8, or direct glaciological relevance of re-writing the coupling. On the other hand, for the description of how the authors took a shell script and turned it into something more high-level, the details of the glaciological problem are largely irrelevant.

Reply:

Again, we must respectfully disagree. The focus of the manuscript <u>IS</u> the application of abstract workflow management to a glaciological problem (so it focuses on both aspects and not a single one), namely, the one of combining long-term, continuum flow models to short-term, discrete element models that are capable of describing the physics of crack propagation and material failure. Naturally, as space is limited, we cannot elaborate the scientific part in full detail, and rather focus on the completely new aspect of the computational implementation using middleware. The results of the mentioned use-case have been described in detail in a previous publication (Vallot et al., 2018) that we regrettably missed to reference in the initial manuscript, but will include in the revision.

Reviewer:

I find it hard to image that somebody would successfully use ELMER on a supercomputer without being able to read and manipulate a shell script. The fact that all sub-steps in their new solution are done in shell or python scripts, is ignored. Similarly I don't see a problem with the user having to take care of the code that copies files from one supercomputer to the other, or to adjust a job header for a new computer/queuing system. These tasks are usually adjusted within a day, with individual tasks being on the order of 10 minutes for anybody used to the supercomputer.

Reply:

It is important to note that our case study is not just to "use ELMER", but rather a combination of Elmer and HiDEM, combining two different applications, and also invoking mesh and conversion routines. As a matter of fact, we had a Master's student in computer science working on improving that workflow for his MSc thesis, where his first task was to understand the 400-line shell script. After a few weeks, that student gave up because it was impossible to understand that shell script independently. Only after explanation by the shell script author, it was possible to improve the script over time. According to the scientific user involved in this endeavor, despite having some Bash shell scripting expertise, UNICORE has proven to be a tremendous improvement for automating the sequential/parallel execution of these scientific workflows. It should be noted that those shell and Python scripts that still remain in our UNICORE-based solution are in fact very short, perform exactly one task each, and are thus modular and easy to understand -- in contrast to the original 400-line shell script.

Furthermore, as the middleware provides an abstract way to combine these models in an application-agnostic way, it may thus open such simulations to the wider glaciology community. This will not only be limited to the Elmer-HiDEM combination, but can be used to integrate other applications as well. This contributes not only to proper reproducibility of science in the light of current discussions on the topic data preservation and data management plans, but also encourages the uptake of our solution for similar research. Further, UNICORE provides a possibility to run across platforms, which would not be possible with a single Bash script running on a front node. The scripts were initially tested on a low-end VM, and could then be run on supercomputers without having to change a single line of code. This is a considerable improvement for scientific communities working on complex problems involving a composite mix of different applications and sharing results in a collaborative manner.

Reviewer:

Sadly, we never learn, how much resources each of these bits requires and whether there is a point in increasing the complexity of the problem to two supercomputers with remote file transfers for gaining computational efficiency, or whether simply letting the smaller model run in a separate (slightly inefficient) job on the same computer would have been faster in the end.

Reply:

Section 3.3 specifies resource requirements for each workflow task (or job). Naturally, as we just applied changes in the workflow, and the main computation time is consumed by the unaltered models that are coupled therein, the resource requirements have not changed at all. However, we would like to point out that by an improved error handling in the middleware (which is difficult, if not impossible to implement in a Bash script) and the opportunity to exploit system availability across several platforms (hence minimizing queuing times), UNICORE represents a solution with a lot of potential to optimize the overall workflow development and management lifecycle. Admittedly, it is difficult to quantify these gains.

We would also like to point out that -- depending on how “slightly” the inefficiency is interpreted -- some HPC centres demand some minimum code performance, which could render the reviewer’s suggestion to run inefficient jobs on the same platform infeasible. Even though running the smaller model on the same computer -- as suggested -- might be faster for the individual scientist, it would result in a sub-optimal resource utilisation. The goal of the HPC community is not only to give individual users their results as fast as possible, but also to utilise the existing hardware as well as possible, to allow running more jobs in the same time. As HPC systems are usually ~99.9% full of jobs on a daily basis, computation time is a precious resource that all scientists should strive to use in the most efficient way.

Finally, regarding file transfers, they can be easily be crafted in scripts if the types of data sources and sinks (http(s), gridftp, scp, sftp...) are known and fixed. But it is unclear how we can manage them if these types are not known in advance while delivering our workflow template to other users, unless we have written a very well-thought-out script. Experience shows that these factors are not always considered when developing scientific applications. Furthermore, the available transfer protocols and their security configurations are changing from HPC site to site and thus can be nicely viewed as another clear benefit of using UNICORE. In UNICORE, we have shown that such parameters can be changed without affecting any scripts.

Reviewer:
While the authors claim a focus of their work was on improving performance, (p.10 l26 “focused in particular on improving overall runtime [,. . .]”), they never provide any information about the results of this endeavor in terms of reduction of overall runtime, or similar metrics.

Reply:
Thank you for pointing to this potentially misleading statement. Rather than improving performance, we should state more clearly that the major aim of our research is to reduce the overall time and effort required for the workflow design, development, workflow-wide and group-wise iterative and conditional constructs, shared and confined scopes, and monitoring and debugging in a platform independent manner. While we indeed mention runtime on page 10, line 26, this was meant in the context of the attributes just mentioned, and will be clarified in a revision. Furthermore, in Section 8 we explicitly state that “this discussion doesn’t cover the application performance”, to emphasize which aspects we are focusing on.

Reply to reviewer 2 on “Scientific Workflows Applied to the Coupling of a Continuum (Elmer v8.3) and a Discrete Element (HiDEM v1.0) Ice Dynamic Model” by Shahbaz Memon et al.

We are grateful for the reviewer’s time and the comments towards improving our manuscript. Our response to the suggested edits and improvements is stated below:

Page 4, line 1, suggest to find an easier name for "the glacier coupling and calving use case“ – and in other places use this name, it will make reading easier

We have now adopted the term “glacio-coupling use case” for this.

Page 12, line 32, can you state how much "significantly reduced“ is?

This line is now more streamlined and better describes the main advantages of our approach. It includes the code usability and separation of each of the defragmented workflow steps into a set of interlinked reusable tasks. Furthermore, there are supporting statements being added, that answer: why the workflow code is reusable and what benefit it yields in terms of extensibility and resource heterogeneity.

Page 18, line 28, missing year in reference

These are actually not references, but the code names of the processor microarchitectures produced by Intel, now clarified this in the text.

Title: suggest to put "model“ in plural

As both the phases, coupling and calving incorporate one single model separately therefore we think “model” being singular would be more suitable here.

Page 4, line 2, add "a“ in front of "part“
Page 4, line 14, suggest to replace "largely“ with "extensively“
Page 4 line 24, suggest to edit "the later-on extruded footprint“ is not clear
Page 4, line 30 suggest to replace "long-time“ with "long-term“
Page 5, line 11, missing r in through

Page 16, line 17, missing t in "It also..."
Page 19, line 5, is a repeat, it has already been stated, maybe it is possible to combine these sentences?
Page 23, line 29, suggest to replace "huge" with a quantitative statement
Page 25, line 1, suggest to replace "verify" with "validate" and "glaciology" with "glaciological"

We appreciate the careful proofreading and have incorporated these corrections and stylistic improvements in the text.

One minor change:
In the last manuscript version there was no citation on the actual glacio-coupling use case which describes the underlying scientific significance and methods. The reference is now added to the latest draft on Page 4, line 8-9.

The modified manuscript is attached as a supplement to this reply.

References

Vallot, D., J. Åström, T. Zwinger, R. Pettersson, A. Everett, D.I. Benn, A. Luckman, W.J.J. van Pelt, F. Nick, and J. Kohler, 2018. ***Effects of undercutting and sliding on calving: a global approach applied to Kronebreen, Svalbard***. The Cryosphere, **12**, 609-625, doi:10.5194/tc-12-609-2018