[GMDD]

Interactive comment on "Best practice regarding the three P's: profiling, portability and provenance when running HPC geoscientific applications" *by* Wendy Sharples et al.

Anonymous Referee #1

Received and published: 21 November 2017

Title: Best practice regarding the three P's: profiling, portability and provenance when running HPC geoscientific applications

Author(s): Wendy Sharples et al.

This paper presents a utility for controlling the execution and initial evaluation of an application (the ParFlow model) running in a (primarily) HPC framework. There are two levels to the framework described: the "run control framework" (or RCF) which itself utilises a more generic JUBE benchmark framework as a workflow engine. Essentially these provide a method of systematically defining, running and analysing some benchmarks - the authors also suggest it would be suitable for use for production simulations

as well.

The framework isolates multiple runs using parameters configured at set up time and keeps all the data produced along with a set of reports, allowing parameter sweeps with automated isolation of the various results (termed "provenance tracking" in the paper). A case study is presented utilising the framework to examine weak scaling (increasing the domain size) for the ParFlow model.

This paper is a difficult read, because there are three different strands within it: motivation, tooling, and the results of using the tools. They are well isolated by the sections, but each is somewhat unsatisfying on its own, and the links between are not as strong as I would like to see in a GMD journal paper. As it stands I do not think it is fit for full publication in GMD, but I think it could be made so with some reworking to make the material more accessible and relevant to the GMD audience.

There is some good material in the motivation, but it falls uncomfortably between being either a complete description of the portability, performance and reproducibility issues associated with geo-scientific modelling or an introduction to those elements for which the tools discussed later are either well suited or applied. It would be stronger if it were the latter.

The discussion of the tooling itself is incomplete in important details, yet full of detail (like the XML files in the appendix) which cannot be easily consumed by the reader because of a lack of appropriate explanation. There is no discussion of why this tool is any different from any other tool (e.g. what are the strengths and weaknesses with respect to the two workflow tools introduced in section 1)?

The results of the analysis of ParFlow are probably the strongest and most interesting parts of the paper, but because of the layout of the paper, introductory material such as the definitions of load balance, are mixed in with results and interpretation. I would rather this section had been organised to correspond to the (very useful) list of "health checks" which begins on the bottom of page 6. It might have been that the relevant

definitions (equations 1 to 4) could have appeared in section 2, since that's where these issues are first introduced. The results and scientific consequences could then be clearly identified in 3.4.

Specific Comments

1. I do not believe the title fairly reflects the material of the paper. The paper is not about best practice, it is primarily about one workflow/benchmarking application, although it does list elements of best practice and motivate some of the issues.

2. The paper begins with motivation with a selective list of references for how increased HPC might be used. The list is somewhat different from that usually presented which normally now includes increased use of data assimilation alongside increasing complexity, domain (spatial or temporal), resolution and ensemble size. It would be good to see this list inclusive of data assimilation and temporal extent and without quite so many references which don't add much value (there are so many it appears to be an attempt to be exhaustive, but it is clearly not exhaustive, better to have few or no exemplars than three or four each, because one is left wondering "why *these* ones"?).

3. There is then some material on the upcoming difficulties with performance portability which adds to the motivation, but the implication is that these are issues which can be solved by optimisation. In particular the paragraph beginning on line4 of page 3 begins by recognising the massive investments required to get performance, then implies that this investment can leverage analysis of existing codes using benchmarking tools such as the RCF/JUBE one discussed here. I think this section would be stronger if there was a clean separation between the aspirations of parameter sweeping and performance analysis, which is primarily about optimisation, and that of massive structural reorganisation of code such as was involved in Leutwyler et al. This is not to denigrate the importance of the former, but just to realistically recognise the scope. As written, the paper overstates it.

4. In the context of scope it would also be useful to identify where the tool might

Interactive comment

have significant limitations, e.g, where it could interfere with other configuration and workflow managers (or be interfered with). This is not to suggest that the tool is not useful, or even powerful, for a particular class of problems; just that like all solutions, it almost certainly has limited applicability. It would be a service for potential users for the authors to provide some clarity on any known scope issues.

5. I think the paper confuses key issues around reproducibility. The implication of the discussions about reproducibility on page 3 and section 3.3 is that "if only the relevant parameters were documented, simulations would be reproducible". While this is undoubtedly *necessary*, it is far from *sufficient*, Baker et al. doi.org/10.5194/gmd-8-2829-2015 discuss the issue of ensuring that the science remains the same when hardware and software environments change. This paper would be stronger for identifying the distinction between these different issues of reproducibility and linking them to Irving's discussion and prior literature.

6. This might address the issue that there is only a cursory discussion of the issues associated with porting ParView to JUQUEEN - indeed, one might have expected the use of this framework to help with that process. "It was found that the optimal flags which did not compromise accuracy" with accuracy determined "to six figures". This reviewer has no idea what they meant by "accuracy" in this context, and the cursory argument suggests that important issues around solution stability were not investigated (despite the motivation being reproducibility).

7. In the discussion of the tools itself, the overall workflow is well described (Figure 1 etc and the excellent list provided for the "health examination"), but the discussion of the tool provides names of files, and then exemplar files (in XML, in the appendices). It's not clear at all how and why this framework is better than a bunch of scripts with input files - it would be considerably stronger if there was some discussion of how the tool exploits the XML files (is there a semantic structure inherent in the files beyond that provided by the use of XML to control syntax)? It might be that this is what the description JUBE reference provides, but I was unable to access the description of

JUBE hidden behind a paywall. Some kind of discussion about how the XML content links to JUBE actions would be helpful. In any case, I recommend removal of the XML files in the appendix, on their own they are inscrutable and provide little value. However, if they were provided in a repo with documentation as to function, they would provide useful complementary material.

8. The bulk of the case study shows the ability of the tools to generate information to understand the performance of the ParView model on this platform, and introduces some of the plans to alleviate performance bottlenecks (such as Adaptive Mesh Refinement). However, I did not fully understand the argument as to why this is the obvious next step from the current arrangement where all cells know about all other cells (why)?. This piece of the argument was another place where I felt that there was blurring together in this paper around issues of performance portability (optimising for a target architecture, but not changing the science), versus algorithmic improvements in performance (which involve changing the science).

9. Somewhere in the paper there needs to be some comparison to prior art and other similar tools which may address some part of the scope of these tools. (The description of GMD Development and Technical papers states: "Development and technical papers usually include a significant amount of evaluation against standard benchmarks, observations, and/or other model output as appropriate.") While, the key word may be *usually*, in the context of *this* paper I think there should be a section similar to the "Related Work" section that appears in many computer science and software engineering papers covering relevant strengths and weaknesses.

10. Finally, I note that the code is apparently available on request, but I think it will not get significant uptake without being both open source and having an open development process. Neither are required for GMD publication, I only mention these factors to encourage the authors to build a community around what looks like a useful tool for a certain class of problems.