Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2017-242-RC2, 2017

Geoscientific Model Development Discussions

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 #2

Received and published: 28 November 2017

The paper presents the some of the challenges of geoscientific modelling on HPC resources, with emphasis on profiling and processing workflows. In order to address provenance, portability and profiling best practices, a run control framework (RCF) based on JUBE is described. The demonstration of the RCF is conducted in a weak scaling experiment in ParFlow, an integrated watershed model. Presentation of the tests results constitutes a notable - and interesting - part of the paper.

The paper is relatively easy to read in sections, but difficult to read as a whole. It covers several domains of expertise such as HPC, geoscientific modelling, software engineering, run harness and profiling. More work is required to present a coherent and uni-

formly detailed experiment in its full context. The profiling element is also substantially more detailed than the others to the point of overshadowing them. Furthermore, profiling results could be much better linked and/or mapped to the very informative "health check" section. As for layout of the "health check", typical diagnosis tools appended for each item does not convey the specific (overlapping or not) role or features of each software very well.

The authors list in section 3.4 the outcomes of the profiling case study. This entices the reader to believe that profiling will also advance provenance and portability. The article presents only a few elements of future work to support the full scope of the study, at least as the title describes it. It is also difficult for a reader to consider the tests results - the most detailed element of the paper - as sufficient proof of application of best practices in profiling, portability and provenance.

The test case experimental design for ParFlow reads like a completely disjoint section to the rest of the paper. It is only very lightly linked with previous or subsequent sections, for instance on the motivations behind that particular test case and how it leverages the RCF, the workflows or the HPC resources. It is unclear looking at the profiling results what are the implications for the test case or to ParFlow itself. This section offers a great potential to present ParFlow software and models (graphically), to show alternative test configurations or to contrast with real-world scenarios.

The paper mentions two other workflow engines (p.3 33) before rapidly shifting to JUBE (p.4 25). What are the advantages, limitations, similarities of JUBE compared to these other solutions? As is, considering that the results analysis of ParFlow provided by the RCF is a large (and interesting) part of the paper, a reader might be tempted to think that JUBE was selected first and pitted against other solutions a posteriori. It might be sufficient to cite major findings for the associated publication.

Specific Comments

1. Some sentences are too long (p.1 8-10, p.3 8-10, p.17 6-10). The messages con-

veyed by these long sentences is important for the coherence of the whole.

2. Figure 9 could be changed to a textual list without loss of content. Is there an overlap or interrelation between those developments? Is there development to do outside the scope of ParFlow, for instance in the run harness or workflows? What does “towards exascale” refers to exactly (cite or describe)? At what scale is the system operating at right now? What are the hurdles from terascale/petascale onto exascale that the presented work will limit or remove?

3. A very large part of the relevant information related to Figure 2 is in its legend. The reader might not care very much about the screen layout of the output. The reader might be interested in numerical values for each result on some occasions, but most values aren’t described or introduced previously in the article. Most of all, the reader will most probably be interested in the metrics themselves and how they relate to profiling - or to some extend to portability or provenance, if applicable.

4. Section 3.1 is very short compared to others at this level, which diminishes its impact. It does not help with readability and flow. The section would benefit from an extension of concepts or reinforcement of links with other sections. It may also be merged elsewhere.

5. Some of the claims in the article are very lightly substantiated, insufficiently nuanced or lacking details. For instance (p.1 16 and p.19), the author claims that RCF is less time consuming and more robust, but less/more than what? Than without use of an RCF? The article concludes by claiming a more efficient use of HPC resources that was not clearly demonstrated; it reads like this is only implied because best practices were followed. If that is the case, it is suggested to better define and highlight these best practices.

6. The paper also mention in a few places costs notions (“invested effort”, “paid off”, “cost in resources”) without providing any data or basic financial analysis. What was the approximate amount invested by articles cited? How much money or energy was

saved? This comment is provided without diminishing the (substantial) presented work or assuming that this information is essential.

7. The paper briefly mentions hybrid and heterogeneous architectures, but do not mention cloud computing. While a very different architecture than HPC, a reader might interested to see if any of the work presented can be applied in cloud computing environments (worfklows? packaged code? profiling? tools? models?). Commercial and scientific offers in HPC-as-a-service might prove an interesting option for the RCF. Absence of cloud computing discussion is not seen here as a limitation of the article, only a potential topic of interest.

8. There are also almost no mention of standards, except a brief sentence on (p.10 3-5). A reader might expect that such large scale systems with claims on portability and provenance do indeed follow standards instead of reinventing the wheel.

9. The code profiling section (p.6 6) makes it difficult to the reader to separate author contribution - by means of the RCF - to outputs from ScoreP and Scalasa. There is a long list of what software can “examine” those outputs. Why are those outputs compatible with all these software? Is it a standardized file format or structured data?

10. Alinea Performance Reports and Intel Vectorization Advisor are present in each item of the health examination, but both software weren’t used. Still, they are recommended by the authors. Is the toolset of the experiment sufficient? What is the additional insight offered by Alinea and Intel’s products that the other tools can’t?

11. Interoperability is largely undefined throughout the text. There is only a single mention of interoperability (p.10 6) for “extra” features. Interoperability between what and what exactly? Was interoperability a critieria either in the conception of the test, the run harness or the workflow?

12. The subsequent paragraph - a very long sentence - mention download and rerun. Results on reruns would be welcomed if possible, either on JUQUEEN or better, on

other infrastructures.

13. The paper states that platform.xml can be easily extended or altered to include new systems. It is always easy to modify XML files, but not trivial to know what constitutes a valid modification and to successfully deploy it on other systems. Is there any tools to help a user, a developer, an administrator? Most discussion on portability revolves around XML files, compiler/linker flags and use of Python language. The paper concludes that the RCF using a workflow engine leads to code that can be ported easily. These conditions are important, but insufficient. A more thorough description of "environment preparation setup" might help a reader to better assess how close this particular run harness is compared to his own environment(s).The article would benefit from a better definition of portability. Ported from where to where? Any example of a second HPC infrastructure in your network? Precise what future work will advance portability. Other topics that could help a reader - this reviewer in particular - to assess portability could include virtual environments, software containers, software repositories and continuous integration frameworks.

14. Table 2 presents efficiencies measured during the weak scaling experiment. The authors states that more in-depth analysis is needed, but no strategy, best practices or future work is offered to the reader. Is this analysis to be conducted by a specialist, is it tool-assisted, what is the state of the art?

15. There is an assumption that the directory structure (Figure B1) "allows for run time provenance tracking" and "such that the model can be rerun without using any other external tools". The directory structure presented is most probably correct as a part of the RCF implementation. Still, it is unclear this is sufficient to insure provenance tracking or rerun. Is a tool set available to explore these directories and/or rerun models? Is it indexed in some form? Is there some other semantic information available that can be used?

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-242, 2017.