Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2018-310-RC2, 2019

Geoscientific Model Development Discussions

Interactive comment on "Optimization of the WRFV3.7 adjoint model" *by* Qiang Cheng et al.

Anonymous Referee #2

Received and published: 7 March 2019

This manuscript present a refactoring of some routines of WRFPLUS in order to decrease the execution time maintaining the total memory required.

1 General comments

- Authors state that they have used several techniques (push/pop-free method, IO analysis, use of adjoint locality) but my first impression is that the paper consist only on a code refactoring affecting some routines. This refactoring was basically done replacing superfluous subroutine calls with push/pop operations and modifying the way that push/pop operations are done. Even if they obtain a significant reduction in the execution time, I do not see in this work a real scientific contribution, but rather an engineering practice.

- The paper is difficult to read and understand. A general review in order to improve the quality of the writing is necessary.

2 Specific comments

- About the Input/output analysis. Usually, I/O refers to disk or network accesses, but no memory accesses. In the paper it is difficult to see if a real I/O analysis has been carried out.
- P2, L6-7: "To some AD applications such as 4DVAR, much more memory consumption must result in poor scalability both in computing grids and in different computer environments." Why is this true? What do you mean by "computing grids"?
- P2, L31-32: "... by a slight of hand several years ago", What does this expression mean?
- P2-3, L34-1: "Actually, the computational cost for these push/pop operations will turn out to be comparatively expensive when the local adjoint cost is reduced at a lower degree." I suppose that you are talking about the percentage of the push/pop operations over the total cost, aren't you? Please clarify it.
- P3, L8-10: "... the most difficult optimization may be associated with huge memory consumption stemming from the fact that the computational cost of the core adjoint procedure is almost evenly distributed across its calling subroutines..." Why is the memory consumption related with the cost balance of the subroutines? Please, clarify.
- P3, L16: "global/local IO", What do you mean by global and local IO in this context?

- P3, L26. You should specify that X and Y^* are in this context.
- P4, L18 and Fig. 1: Figure 1 does not help to understand the concept of “adjoint locality". A better explanation of this figure is necessary.
- P5, 2nd paragraph: It is said that required data are pushed into stack “. . . during each sweep of RP when running its adjoint subroutine Adj_RP. . . ". In the same paragraph: “However, the sweep of RP within Adj_RP will be completely removed for better adjoint performance. . . ” If the sweep of RP is removed, how are data pushed into the stack? Please, clarify.
- Figure 3: How can the second call to LL2JK be removed if P is not popped out from the stack?
- P5, L16: Please, indicate what NV is.
- P6, L19-21: “First, we use several or more stacks instead of only one, each of which is still a 1D data structure. It has been shown that this type of data structure has the advantages of smaller access cost and flexible expressions, either in the communications between procedures or in local program calculations.” Can you please add any reference to justify this assertion?
- P6, L23. “Although the cost of allocating/deallocating many dynamical stacks cannot be neglected compared with the costs of these adjoint procedures. . . ” Again, this assertion should be justified. Which memory allocator are you using?
- Section 3. In the paper, push/pop operations are replaced by copying the data into a vector (keepx) and then recovering them from that vector. I cannot see how different this data movement is from using a stack. In the paper it is said (P7, L14-16) “Note that the pushing operation PUSHREAL8ARRAY is replaced by a direct evaluation statement from x to keepx with approximately the same cost both in

run time and in memory.” So the only improvement is due to the replacement of the pop operation. Might not be simpler to optimize the use of this operation?

- P7, L15: “. . . a direct evaluation statement from x to keepx. . . ” Please, clarify what do you mean by “a direct evaluation", isn’t it a simple copy?
- P7, L28. “sulfurous calculations.” I suppose you mean “superfluous".
- Regarding the test results, more details about how the experiments were carried out should be included, e.g., compiler used, optimization level, MPI version, characteristics of the cluster (cores per processor, memory per core, etc).
- Table 1: You use until 256 processors, in a system with 5000 processors (250*20). It will be interesting to see values for a larger numbers of PEs and a larger resolution.
- Table 1: How many experiment have been done for each value of PE? Are the results in the table an average? What is the standard deviation?
- P8, L11-12: “At the same time, both aspects of improvements are slightly increased with the increase of the number of processors. . . ” What are those “both aspects"?
- P8, L18: “. . . uniformly different by no more than two last significant digits in double precision.” If you do not mean the last two bits in the double precision binary representation, this statement is meaningless. You should use the absolute difference between values.
- Test result section: It will be very interesting add some graph to show whether any change in scalability (both strong and weak) is caused by the proposed improvements.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2018-310, 2018.