# Part 1: Responses to Anonymous Referee #1

1. First, please increase the dpi for the figures, it's hard to read them.

Response: We have increased the dpi for the figures in the revised manuscript.

2. Line 21: please properly introduce the acronyms: "...system GSI/EnKF and the coupled model FIO-AOW".

Response: The acronyms corresponding to GSI/EnKF and FIO-AOW have been added. Please refer to L54 and L311.

3. Lines 26-31: Here you are talking about two widely used DA algorithms: ensemble techniques derived from the Kalman filter and variational data assimilation. But there are methods that you don't mention. Please reformulate the paragraph accordingly. Besides, I would suggest to remove the sentence ", can be viewed as a special case of ensemble-based methods with only one member in the ensemble when we attempt to design and develop a software framework for data assimilation." (lines 30-32).

Response: Some DA methods such as Optimal Interpolation (OI) and Ensemble OI has been added. Please refer to L31~L36. Currently, we do not remove the sentence ", can be viewed as a special case …", and add a new discussion at L471~L474. We will further modify the manuscript if this modification is improper.

4. Line 34: Please add the Environment and Climate Change Canada's (ECCC) hybrid 4D-EnVar DA used operationally in the weather prediction system: Buehner, M., McTaggartâ˘ A˘RCowan, R., Beaulne, A., Charette, C., Garand, L., Heilliette, S., Lapalme, E., Laroche, S., Macpherson, S.R., Morneau, J. and Zadra, A. (2015) Implementation of deterministic weather forecasting systems based on ensembleâ˘ A˘Rvariational data assimilation at Environment Canada. Part I: the global system. Monthly Weather Review, 143, 2532–2559

Response: We have modified manuscript accordingly. Please refer to L38.

5. Lines 36-37: With the rapid development of science and technology, numerical forecasting systems with only an individual component model (such as an atmospheric model) have reached a predictability limit. Please reformulate it or add appropriate references.

Response: We have reformulated this statement with new references. Please refer to L40~L42.

6. Lines 37-38: Coupled models have been widely used in numerical forecasting to break the bottleneck of the limited predictability... I would also reformulate it in a way that it was established that coupled models may provide better results with respect to uncoupled models. Besides, you may wish to add references for coupled models that used operationally in several operational centres (Met Office, ECMWF, ECCC, etc.)

Response: We have reformulated this statement with new references. Please refer to L40~L42.

7. Lines 42-43: add references to coupled data assimilation (CDA) systems developed in JAMSTEC, NOAA, Met Office, ECMWF, ECCC: Sugiura et al., 2008, Mochizuki et al., 2016, Yang et al., 2013, Zhang et al., 2014, Lea et al., 2015, Laloyaux et al. 2016 and 2018, Browne et al., 2019, Skachko et al. 2019

Response: The above references have been added into the revised manuscript. Please refer to L45~L48.

8. Lines 45-49: It is not clear what do you mean. Reformulate or simply remove this sentence.

Response This sentence has been reformulated. Please refer to L49~L52.

9. Line 49: You don’t mention the NASA GMAO system, for example. I would suggest changing “most” by “several”.

Response: The manuscript has been modified accordingly. Please refer to L52.

10. Lines 49-60: The ensemble size of current global operational systems used in NWP generally exceeds 256 members. The global models used in the ensemble are run on grids with horizontal sampling of several tens of kilometers. So, the I/O operations are currently necessary. If the aim of the paper is not to discuss real systems, please state it more clearly in the abstract and the beginning

of introduction. It will help to understand what kind of CDA system you are developing and its future use.

Response: Thanks a lot for your suggestions. Our statements may not be clear enough to cause misunderstandings. DAFCC1 can handle the original I/O operations for data exchange between model ensemble and DA algorithms via MPI, while the I/O operations of operational systems for outputting results are still kept. Regarding the evaluation in this manuscript, the corresponding I/O operations of WRF as well as other component models are still kept. In the revised manuscript, we try to use the word "extra" for distinction. Please refer to L59~L61.

11. Line 62-63: reformulate the meaning of WCDA. Not the models are assimilated independently, but data from model components are assimilated independently by two separate DA systems.

Response: We have modified the corresponding statement. Please refer to L70~L71.

12. Line 76-78: It's obviously more efficient. However, you don't mention the feasibility of this approach when a model (or several models and couplers) is called in the PDAF Fortran code is a subroutine.

Response: We have modified the corresponding statements. Please refer to L87~L89.

13. Line 332: again, GSI/EnKF has not been defined yet. What is GSI?

Response: We have a brief introduction in Sec 1 L54: "… Grid Point Statistical Interpolation (GSI; Shao et al., 2016) combined with EnKF (Liu et al., 2018a), …". In Sec 5.1, more details about the system structure, operation, integration and driver by DAFCC1 of GSI/EnKF are introduced.

14. Lines 330-343: It's difficult to understand this section. First, you have an atmosphere-ocean-wave coupled model. Second, you have a DA system that you call GSI/EnKF. Does this system compute atmospheric analyses? How ocean analyses are computed? Do you compute also analysis for the waves?

Response: It has been stated and discussed that the example DA system only computes atmospheric analyses corresponding to WRF currently. Please refer to L319~L320, L478~L480.

15. Line345: What do you mean by "combines"? Is it a hybrid system? Or you may choose between two options: a variational (which one?) and ensemble technique? You may also state explicitly that both options share parts of codes, for example, observation operators.

Response: GSI/EnKF is composed of a variational DA sub-system (GSI) and an ensemble DA sub-system (EnKF), which can be used as variational, pure ensemble or hybrid DA systems sharing the observation operator in GSI codes. In this paper, we focus on the pure ensemble DA system for developing the sample DA system, where GSI is only used as the observation operator that calculates the difference between model variables and observations on the observation space and EnKF calculates analysis increments and updates model variables. It provides two options for calculating analysis increments for ensemble DA; i.e., a serial Ensemble Square Root Filter (EnSRF) algorithm and a Local Ensemble Kalman Filter (LETKF) algorithm, while EnSRF is used in this paper. The manuscript has been modified accordingly. Please refer to L324~L326.

16. Lines 345-350: What DA method is used in your experiments: EnKF, EnSRF or ETKF? Please clarify. Instead of using "pure ensemble DA", you may simply say that the EnKF (or ETKF or whatever) mode was chosen.

Response: EnSRF is used in our experiments. This has been stated in the revised manuscript. Please refer to L329~L330.

17. Line 354-371: atmospheric DA? So, do you use EnKF to perform atmospheric analyses?

Response: Yes, we use EnKF to perform atmospheric analyses both in the ensemble DA sub-system of WRF and the example DA system of FIO-AOW. This has been stated in the revised manuscript. Please refer to L330.

18. L373: what step you are talking about? An atmospheric model time step?

Response: "Step" means the flowchart for running the pure ensemble DA system in a DA window shown in Figure 7a and further summarized in Sec4.1.1. We have revised the statements to make it clearer. Please refer to L331~L333 and L354.

19. l.374-395: As shown... It's not clear.

Response: We have revised the related statements accordingly. Please refer to L356~L357.

20. L397: where it was used? And who has upgraded the coupled model to the c-coupler 2.0? After I read the section 5, I realized that there is no ocean data assimilation in your system. So I wonder why you call your prototype WCDA? Your system is an atmospheric DA using a coupled model to compute background states.

Response: FIO-AOW has been used for some typhoon simulations and researches in the referenced works (Zhao et al., 2017 and Wang et al., 2018; please refer to L311~L312). It has been upgraded to C-Coupler2 by us (please refer to L380). It is true that our system is an atmospheric DA using a coupled model to compute background states. We call it WCDA following the WMO report (Penny et al., 2017): "for WCDA, the direct impacts of observations on the analysis are limited to the domain in which the observations reside, while cross-domain impacts are produced as a secondary effect by the integration of the coupled model forecast", which does not restrict whether one-component or multiple-component data are assimilated.

# Part 2: Responses to Anonymous Referee #2

1. The motivation for developing your own system rather than using PDAF I found lacking. There were 2 main points I could find in section 2, namely "PDAF [... imposes] a precondition of process layout such that each ensemble member uses the same number of processes with successive IDs in the MPI_COMM_WORLD" and "[PDAF] only makes the processor cores of the first ensemble member available to the DA algorithm and forces the processor cores used by other ensemble members to idle when running the DA algorithm". The second statement I think is untrue, but Dr Lars Nerger has posted a short comment on PDAF so I trust he will ensure the correctness there. The first point I find to be obscure as I cannot think of a situation where you would not have ensemble members using sequential MPI process IDs. If you have a specific situation where this is the case, please elaborate on it so the reader can understand why this is important. One overarching question which is not addressed is why would you design from the outset a "weakly coupled" data assimilation system? Why not design a strongly coupled system and then simplify it? I suppose the answer here is to still be able to piggyback on existing observation processing systems and to allow for different observation frequencies, but this should be clearly set out in the article.

Response: Based on the discussions with Dr Lars Nerger, we rewrite the motivation for developing DAFCC. Please refer to L80~L106 in Section 2. A statement has been added for the question why we would design from the outset a "weakly coupled" DA system. Please refer to L69~L70.

2. I would like to see more clarity in relation to the comparisons that you make. There are a number of places where the comparison is against an system that uses I/O and reading/writing files from/to disk rather than MPI communications. In such a case phrases like "accelerating the DA system" should be qualified. There are other relations made where it is unclear what the comparison is with. For example in the abstract you state that the new methodology "enables the DA method to utilize more processor cores in parallel execution" but I cannot see the baseline for such a statement. Moreover would such a statement hold with a different baseline?

Response: The manuscript has been modified accordingly. Please refer to L20~L22, L25, L437, L438 and L471.

3. The article gives a reasonable overview and references for general data assimilation concepts. However the article should point the reader to some of the latest examples of operational weakly coupled data assimilation. Good references for this include, with the first using PDAF:

Goodliff, M., Bruening, T., Schwichtenberg, F., Li, X., Lindenthal, A., Lorkowski, I., & Nerger, L. (2019). Temperature assimilation into a coastal ocean-biogeochemical model: assessment of weakly and strongly coupled data assimilation. Ocean Dynamics, 69(10), 1217–1237. https://doi.org/10.1007/s10236-019-01299-7

Skachko, S., Buehner, M., Laroche, S., Lapalme, E., Smith, G., Roy, F., . . . Garand, L. (2019). Weakly coupled atmosphere-ocean data assimilation in the Canadian global prediction system (v1). Geoscientific Model Development, 12(12), 5097–5112. https://doi.org/10.5194/gmd-12-5097-2019

Browne, P. A., de Rosnay, P., Zuo, H., Bennett, A., & Dawson, A. (2019). Weakly Coupled Ocean-Atmosphere Data Assimilation in the ECMWF NWP System. Remote Sensing, 11(234), 1–24. https://doi.org/10.3390/rs11030234

Response: We have added the recommended references. Please refer to L45~L48.

Specific comments:

4. Lines 12,25. "better" than what?

Response: We have modified the corresponding statement. Please refer to L13.

5. Line 47: "how to conveniently (1) achieve an ensemble run of a coupled model" What is your measure of convenience here? This is a task which is regularly done at many centres around the world, do they all have inconvenient methods for running ensembles of coupled models?

Response: The corresponding statement has been modified. Please refer to L50.

6. Line 55: On the use of disk files, this is also a robust strategy when it comes to massively parallel computing, as this risk of random task failures increases with the size of the coupled models and the

number of ensemble members. This should be noted as positive reason for using disk files, as well as the potential to use a larger ensemble than can be run at a single time on an HPC machine.

Response: We make a corresponding discussion in the revised manuscript. Please refer to L496~L506.

7. Line 57. PDAF is indeed *the* standard for ensemble based DA frameworks. Others also exist. For example EMPIRE (https://pbrowne.bitbucket.io/empire) Browne, P. A., & Wilson, S. (2015). A simple method for integrating a complex model into an ensemble data assimilation system using MPI. Environmental Modelling \& Software, 68, 122– 128. https://doi.org/10.1016/j.envsoft.2015.02.003

Response: Thanks a lot for introducing this pioneer work. Brief introductions and discussions about EMPIRE have been added. Please refer to L62~L65 and L80~L87.

8. Line 93: "How to compile the code of DA methods with the model". This is not necessary. In particular if you run (using MPI) in MPMD mode then the model and the DA could be compiled independently.

Response: As how to compile DA may be not a critical problem, we have reduced the related statements in the revised manuscript.

9. Line 108: "Although PDAF enables a DA algorithm to run in parallel, it only makes the processor cores of the first ensemble member available to the DA algorithm and forces the processor cores used by other ensemble members to idle when running the DA algorithm." This is not my understanding of PDAF. I see that Dr Lars Nerger has already submitted comments in relation to PDAF, so I am assured that he will have given you the latest and correct information in relation to this. You need to discuss other parallel strategies such as that used by P.A. Browne, S. Wilson, 2015.

Response: Based on the discussions with Dr Lars Nerger, we rewrite the motivation for developing DAFCC, where EMPIRE has also been briefly discussed. Please refer to L80~L106 in Section 2.

10. Line 115: what are such preconditions? Can you give examples where these exist, and if they do, why they are a problem?

Response: We have removed the statements about preconditions.

11. Line 122: You should make clear this is because the MPI processes from all ensemble members are available. Or are there even more available?

Response: We have modified the corresponding statement, please refer to L103~L104.

12. Line 131: Are you suggesting a coupled model which uses a different coupler, such as OASIS, would then be put into C-Coupler2.0 for the DA component?

Response: Yes, C-coupler2.0 can handle this case.

13. Figure 1: This has no explanation. I fail to see the usefulness of this figure.

Response: For explanations of Figure 1, please refer to L117~L131.

14. Line 143: What is the alternative to DLL?

Response: DLL is the unique choice. We have revised the manuscript accordingly. Please refer to L123.

15. Line 146: "The ensemble component manager is responsible for generating and managing the communicator of ensemble members of a component model." Does this mean you have a separate ensemble component manager for every component of your coupled model, such as atmosphere, ice, land, composition, etc? If so please state this to help the reader.

Response: The ensemble component manager is a common software module shared by all components. We modified “a component model” into “each component model”. Please refer to L125~L127.

16. Line 155/Figure 3: Is a restriction that the components in each ensemble members run on the same number of MPI processes? Surely there is a restriction enforced by the DA algorithms that the component model is on the same grid for every ensemble member, or has some very exotic DA methodology been implemented? In the case they have, how do you then establish which DA algorithms are applicable given the difference in the ensemble members?

Response: We have deleted this part in the revised manuscript.

17. Line 159/160: "execution of a DA algorithm in a component model does not force the processes of other component models to be idled". This must relate to the time stepping procedure of the coupled model. In fact here are you for the first time enforcing that all components of the model must have separate MPI processes? This is not the case in, for example, the ECMWF earth system model (Mogensen, K., Keeley, S. and Towers, P., 2012. Coupling of the NEMO and IFS models in a single executable. Reading, United Kingdom: ECMWF.)

Response: We have revised the corresponding statement to make it clearer. Please refer to L134~L135.

18. Line 165). Point 4. Please give an examples of such a DA algorithm procedures. Do you mean, for example, that DA algorithm 1 procedure 1 would be calculation of model equivalents ($H(x)$) and DA algorithm 1 procedure 2 would be something like an SVD of the ensemble perturbation matrix?

Response: Yes. For example, for the GSI/EnKF system, GSI can be used as DA algorithm 1 procedure 1, which is used as the observation operator that calculates the difference between model variables and observations on the observation space; EnKF can be used as the DA algorithm 1 procedure 2, which calculates analysis increments and updates model variables of all ensemble members. Please refer to the example shown in L359~L362.

19. Line 169: "Scripts are allowed to conduct necessary process control". This comes out of the blue, and it is not clear how this fits within the methodology of having everything using MPI communication. Can you give examples in section 4.4.

Response: This statement is incorrect. We have revised it. Please refer to L144~L145.

20. Line 188: The "weakly coupled" component of your methodology then relies on using the C-Coupler2.0 to control the coupling of the model then?

Response: It does not rely on using the C-Coupler2.0 for model coupling. C-Coupler2 and DAFCC can only handle the coupling between the model ensemble and the DA algorithm, while the coupling among component models in each ensemble member can also be handle by the original coupler that can be not C-Coupler2.

21. Figure 5: Why is there no red within the DA_CCPL_RUN subroutine to indicate data exchange between the model and the DA?

Response: We have added more introductions to the DA_CCPL_RUN subroutine. Please refer to L192~L195.

22. Section 4.4, 2). Terms such as periodic timer, period_unit, period_count and lag are introduced with no context. These should be defined as well as an explanation of why they are needed for data assimilation.

Response: The periodic timer is used to enables users to flexibly set the frequency as well as the model time of running the corresponding DA algorithm (L290~L291). The introduction to periodic timer can be found in L291~L295.

23. Section 4.4) I fail to see why any of this is relevant to data assimilation. What is an example of statistical processing in a DA context?

Response: Two kinds of statistical processing of input model fields are supported in DAFCC1: the statistical processing in each time window named “time_processing” and the statistical processing among ensemble members named “ensemble_operation”. “Time_processing” specifies whether the field values from the model to the DA algorithm are instantaneous or time averaging. Time averaging model values may be required in some interdecadal DA experiment. “Ensemble_operation” refers to whether the fields transferred from the model to the DA algorithm is the value of a single member or the values of all ensemble members. For example, in the flowchart for running GSI/EnKF (Sec 4.1.1), the model is required to provide the mean values of all ensemble members to GSI in the third main step and the aggregated values of all ensemble members to EnKF in the firth main step. The manuscript is modified accordingly. Please refer to L296~L299.

24. Line 330) "A sample weakly" - do you mean "An example weakly coupled ensemble DA system"? I don’t know what "sample" refers to here. This is used many times throughout the manuscript - please clarify.

Response: “sample” has been replaced by “example” throughout the paper.

25. Section 6.1) The details of the EnSRF (i.e. localization radius and inflation factors) are not useful without a description of the model.

Response: Descriptions of the model parameters are given in the following part of Sec 5.1 (for example, horizontal resolution can be referred to Table 1). Some details of the EnSRF in this part are introduced for the reproducibility of the experiment, although the DA experiment in this paper is mainly used to verify DAFCC1 and we do not pay too much attention to the evaluation of actual DA effect.

26. Line 410) "We evaluate the effectiveness of DAFCC1 in developing a weakly coupled ensemble DA system". I don't see the justification for this statement. I can see you have implemented the system and shown how it performs computationally with various parameters, as well as a very simplistic verification that the data assimilation is implemented correctly. You should state a measure for effectiveness - was it simply to have a functioning system? Compare this with Browne and Wilson, 2015, where they "propose a simple implementation strategy which does not focus on maximum efficiency of the code. Instead the focus is on the speed of implementation."

Response: In fact, we focus on the code speed and correctness of DAFCC1 in developing a weakly coupled ensemble DA system. "effectiveness" has been replaced by "correctness" throughout the paper.

27. Line 424) Why were 3200 cores used when each node has 24 processors? mod(3200,24) != 0.

Response: Although our account can use a maximum number of 3600 cores, we can only use about 3200 cores actually (there are may be some errors in the computer system).

28. Line 436) "variables used for DA" -> "are the prognostic, or analysed, variables in the data assimilation".

Response: Thank you for your suggestions. We have revised the statement, please refer to L416~L418.

29. Section 6.2 could be a simple statement saying that WRF-GSI/EnKF with DAFCC1 is bit-identical to the original offline WRF-GSI/EnKF.

Response: We have added introduction to the validation standard. Please refer to L433~L434. We still keep most original content in Section 5.2 because we think that we should show why we employ the bit-identical standard for validation.

30. Line 460/Figure 11c) Why does the offline timing of GSI vary with different numbers of ensemble members? On line 458 you state that you run all ensemble members of the offline system concurrently, so I would expect a constant value of time for the model run as you change the number of ensemble members. This clarification will be essential in understanding the rest of the figures here, as otherwise it seems like the comparison may be unfair. Could it be i/o related? With every member trying to write output files at the same time your system slows? If this is the case it should be explicitly accounted for in the final paragraph of this section. Furthermore, you should detail what file system architecture is used at BSCC in section 6.1. Is it something like lustre?

Response: We add a discussion about the above issues in Section 5.3 (L456~L460). The Lustre file system is used at BSCC (L402~L403).

31. Section 6.4) This is not a measure of the effectiveness in developing a weakly coupled ensemble DA system. Figure 15 shows results for northern hemisphere (25N-90N) and tropics (25S-25N) but I understood this was a limited area system, running from 0N-50N. You should update the figure to reflect this. All this figure appears to show is that the DA system is producing increments which have the correct sign.

Response: “Effectiveness” has been replaced by “correctness” throughout the paper. There were mistakes in the title in Figure 15, which has been corrected (L781~L782).

Part 3: a marked-up manuscript version

# Developing a common, flexible and efficient framework for weakly coupled ensemble data assimilation based on C-Coupler2.0

Chao Sun[1], Li Liu[1,3], Ruizhe Li[1], Xinzhu Yu[1], Hao Yu[1], Biao Zhao[1,2,~~3~~4], Guansuo Wang[2,4], Juanjuan ~~Liu~~[4] Liu[5], Fangli Qiao[2,~~3~~4], Bin Wang[1,~~4~~5]

[1] Ministry of Education Key Laboratory for Earth System Modeling, Department of Earth System Science, Tsinghua University, Beijing, China
[2] First Institute of Oceanography, Ministry of Natural Resources, Qingdao, China
[~~3~~3] Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai), China
[4] Key Lab of Marine Science and Numerical Modeling, Ministry of Natural Resources, Qingdao, China
[~~4~~5] State Key Laboratory of Numerical Modeling for Atmospheric Sciences and Geophysical Fluid Dynamics (LASG), Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing, China

*Correspondence to*: Li Liu (liuli-cess@tsinghua.edu.cn), Bin Wang (wab@tsinghua.edu.cn)

**Abstract.** Data assimilation (DA) provides~~ better~~ initial states of model runs by combining observational information and models. Ensemble-based DA methods that depend on the ensemble run of a model have been widely used. In response to the development of seamless prediction based on coupled models or even ~~earth~~Earth system models, coupled DA is now in the mainstream of DA development. In this paper, we focus on the technical challenges in developing a coupled ensemble DA system, ~~which have not been satisfactorily addressed to date.~~especially how to conveniently achieve efficient interaction between the ensemble of the coupled model and the DA methods. We first propose a new DA framework DAFCC1 (**D**ata **A**ssimilation **F**ramework based on **C**-**C**oupler2.0, version 1) for weakly coupled ensemble DA, which enables users to conveniently integrate a DA method into a model as a procedure that can be directly called by the model ensemble. DAFCC1 automatically and efficiently handles data exchanges between the model ensemble members and the DA method without global communications, and ~~enables~~does not require users to develop extra codes for implementing the ~~DA method to utilize more processor cores in parallel execution.~~data exchange functionality. Based on DAFCC1, we then develop ~~a sample~~an example weakly coupled ensemble DA system by combining ~~the~~an ensemble DA system ~~GSI/EnKF~~ and ~~the~~a regional atmosphere-ocean-wave coupled model~~ FIO-AOW~~. This ~~sample~~example DA system and our evaluations demonstrate the ~~effectiveness~~ correctness of DAFCC1 in ~~both~~ developing a weakly coupled ensemble DA system and the effectiveness in accelerating an offline DA system that uses disk files as the interfaces for the ~~DA system~~data exchange functionality.

## 1 Introduction

Data assimilation (DA) methods, which provide better initial states of model runs by combining observational information and models, have been widely used in weather forecasting and climate prediction. The **en**semble **K**alman **f**ilter (EnKF; Houtekamer and Mitchell, 1998; Evensen, 2003; Lorenc, 2003a; Anderson and Collins, 2007; Whitaker, 2012) is a widely used DA method

**Formatted:** Superscript

**Formatted:** Superscript

**Formatted:** Font: Bold

**Formatted:** Font: Bold

**Formatted:** Font: Bold

that depends on an ensemble run of members. Other DA methods that ~~only rely on~~can be performed a single model run, such as the Nudging method (Hoke and Anthes, 1976; Vidard et al., 2003), Optimal Interpolation (OI; Gandin, 1965), Ensemble OI (EnOI; Oke et al., 2002; Evensen, 2003), three-dimensional variational analysis (3D-Var; Anderson et al., 1998; Courtier et al.1998; Gauthier et al., 1999; Lorenc, 2000) and four-dimensional variational analysis (4D-Var; Courtier et al., 1994; Kalnay, 2003; Lorenc, 2003b; Rabier et al., 2007), can be viewed as a special case of ensemble-based methods with only one member in the ensemble when we attempt to design and develop a software framework for data assimilation. Moreover, hybrid DA methods, such as hybrid Ensemble/3D-Var (Hamill, 2000; Etherton and Bishop, 2004; Wang et al., 2008, 2013; Ma et al., 2014) and ~~the ECMWF~~ ensemble-based 4D-Var ~~scheme~~schemes (Fisher, 2003; Bishop and Hodyss, 2011; Bonavita et al., 2012, 2016; Buehner et al., 2015), also depend on an ensemble run of members from the same model.

With the rapid development of science and technology, numerical forecasting systems ~~with~~are involving from only an individual component model (such as an atmospheric model) ~~have reached a~~ to coupled models that can achieve better predictability ~~limit. Coupled models have been widely used in numerical forecasting to break the bottleneck of the limited predictability,~~(Brown et al., 2012; Mulholland et al., 2015), and earth system models are being used to develop seamless prediction that spans timescales from minutes to months or even decades (Palmer et al., 2008; Hoskins, 2013). Along with the use of coupled models in numerical forecasting, common and flexible DA methods for coupled models are urgently needed (Brunet et al., 2015; Penny et al., 2017). Coupled DA technologies have already been investigated widely and DA systems have been constructed (Sugiura et al., 2008; Fujii et al., 2009, 2011; Saha et al., 2010, 2014; Sakov et al., 2012; Yang et al., 2013; Tardif et al., 2014, 2015; Lea et al., 2015; Lu et al., 2015a, b; Mochizuki et al., 2016; Laloyaux et al., 2016, 2018; Browne et al., 2019; Goodliff et al., 2019; Skachko et al. 2019), in which ensemble based DA methods have already been applied (e.g., Zhang et al., 2005, 2007; Sluka et al., 2016).

~~In addition to~~ To develop a coupled ensemble DA system, besides the ~~inherent~~ scientific challenges ~~of~~regarding DA methods, there are also technical challenges ~~in developing a coupled ensemble DA system, which have not yet been satisfactorily~~to be addressed~~. These include~~, such as how to ~~conveniently (1)~~ achieve an ensemble run of a coupled model, ~~(2)~~how to conveniently integrate the software of a coupled model and the software of ensemble DA methods into a robust system, and ~~(3)~~how to conveniently achieve efficient interaction between the ensemble of the coupled model and the DA methods~~, especially as model resolution improves. Most~~. The existing coupled ensemble DA systems such as the Data Assimilation Research Testbed (DART; Anderson et al., 2009), the ensemble coupled data assimilation system (ECDA; Zhang et al., 2005, 2007), and the Grid ~~Point~~point Statistical Interpolation (GSI; Shao et al., 2016) combined with EnKF (Liu et al., 2018a), employ disk files as the interfaces of data exchange between the model ensemble members and the DA methods, and iteratively switch between the run of the model ensemble and DA using software-based restart functionality that also relies on disk files. Such an implementation (called offline implementation hereafter) can guarantee software independence between the models and the DA methods, so as to achieve flexibility and convenience in software integration; however, the extra I/O accesses of disk files as well as the extra initialization of software modules introduced by the data exchange and the restarts are time-consuming and can be a severe performance bottleneck ~~for improved~~under finer model resolution (Heinzeller et al., 2016; Craig et al., 2017).

**Formatted:** Font: +Headings (Times New Roman)

**Formatted:** Font: +Headings (Times New Roman)

The Parallel Data Assimilation Framework (PDAF; Nerger et al., 2005; Nerger and Hiller, 2013; Nerger et al., ~~2019) provides a set of application programming interfaces (APIs)~~2020) and the Employing Message Passing Interface for ~~transforming the DA software modules into internal procedures of the model and achieves~~ Researching Ensembles (EMPIRE; Browne and Wilson, 2015) framework have shown that MPI (Message Passing Interfaces)-based data ~~exchange~~exchanges between the ~~models~~model ensemble members and DA procedures~~. It~~ can ~~therefore achieve~~produce better performance ~~as it does~~for DA systems, because they do not require disk files or the restart ~~functionality. However, it still has some limitations that will be described further in Sect. 2.~~operations.

~~To fully overcome~~Noting that most existing couplers for Earth system modeling have already achieved flexible MPI-based data exchanges between component models in a coupled system, we design and develop a common, flexible and efficient framework for coupled ensemble data assimilation, based on the ~~technical challenges, we first consider~~latest version of the Community Coupler (C-Coupler2.0; Liu et al., 2018b). Considering that existing observation processing systems can introduce different observation frequencies corresponding to different component models, we take consideration of weakly coupled ensemble DA where the data from different component models are assimilated independently ~~from each other~~by separate DA methods (Zhang et al., 2005, 2007; Fujii et al., 2009, 2011; Saha et al., 2010, 2014~~),~~) in this work, and in further work will then target strongly coupled ensemble DA, which generally uses a cross-domain error covariance matrix to account for the impact of the same observational information on different component models cooperatively (Tardif et al., 2014, 2015; Lu et al., 2015a, b; Sluka et al., ~~2016). In this paper, we design and develop a common, flexible and efficient framework for weakly coupled ensemble data assimilation, based on the latest version of the Community Coupler (C-Coupler2.0; Liu et al., 2018b), that is a flexible coupler which can exchange data efficiently between different component models in a coupled system.~~2016).

The remainder of this paper is organized as follows. Section 2 introduces the ~~motivation for developing~~overall design of the new DA framework named DAFCC1 (**D**ata **A**ssimilation **F**ramework based on **C**-**C**oupler2.0, version 1). The~~ overall design and~~ implementation of DAFCC1 are described in ~~Sect. 3 and 4, respectively.~~ Section ~~5~~3. Section 4 introduces the development of ~~a sample~~an example weakly coupled ensemble DA system based on DAFCC1. Section ~~6~~5 evaluates DAFCC1. Finally, Section ~~7~~6 contains a discussion and conclusions.

## ~~2 Motivation~~

## 2 Overall design of the new framework

The ~~experience~~experiences gained from PDAF ~~shows~~and EMPIRE show that, a framework with an online implementation where all ensemble members of the model and all procedures of DA methods are combined into a single MPI program is essential for ~~efficient~~improving the interaction between the model and the DA software. ~~As a coupled model and its component models often consist of general purpose software that can serve~~ There can be different ~~applications other than DA systems, a DA framework should try to minimize the change to a coupled model system when setting up its ensemble run in a single MPI~~



**Formatted:** Font: NimbusRomNo9L-Regu

~~program. Besides creating a working directory~~strategies for ~~each ensemble member of the coupled model, such an ensemble run requires addressing how to generate distinct local communicators from the global communicator *MPI_COMM_WORLD* of an MPI program and how to make the coupled model use a local communicator rather than the global communicator for each ensemble member. Slight modifications to the model code will be required when the original model code uses the global communicator but not a local communicator as the default. PDAF addressed the problem of generating local communicators by imposing a precondition of process layout such that each ensemble member uses the same number of processes with successive IDs in the *MPI_COMM_WORLD*. For example, if there are 10 members in an ensemble run and each ensemble member uses 50 processes, there should be 500 processes (#0~#499) in the *MPI_COMM_WORLD*, and processes #0~#49 will be used for the first ensemble member. Such a solution can facilitate the corresponding code implementation, but requires the user to know and guarantee the process number precondition when submitting the MPI job of an ensemble run.~~

~~In~~ the online implementation~~, DA methods should be directly driven by the model, which means that DA methods should be integrated into the model as procedures that are directly called by the model driver. There are at least three issues to be addressed: 1) how to compile the code of DA methods with the model; 2) how to initialize and run the DA methods in the model execution; and 3) how to pass the corresponding information and variables between the model and DA methods. Although model developers can fully address these questions by developing all the required code themselves, a framework should seek to minimize the required code development and to maintain .~~. In EMPIRE, a DA method is compiled into a standalone executable running on the processes distinct from the model ensemble, and global communications of *MPI_gatherv* and *MPI_scatterv* are used for exchanging data between the model ensemble and the DA method. Such an implementation can maintain the independence between the DA software and the model~~. PDAF provides several APIs those formulate how a model initializes and runs DA methods and how the model and a DA method exchange information and variables. However, the user is still required to perform many tasks; e.g., developing the call back functions for gathering the same model field from all ensemble members~~, while the global communications are generally inefficient and there are idle processes almost at any time because the model ensemble and the DA method generally work sequentially but not simultaneously. In PDAF, a DA method is transformed into a ~~vector of the DA method and distributing the data of a field in a vector to all ensemble members, according to the~~ native procedure that is called by the corresponding models via the PDAF application programing interfaces (APIs). Thus, a DA method can share the processes of the model ensemble. The code releases of PDAF (http://pdaf.awi.de/trac/wiki) provide implementations of data ~~structures of the model and the DA implementation. The user is recommended to make the DA implementation using~~exchanges for a default case where a DA method shares all processes of the first ensemble member of the corresponding model and keeps the same parallel decomposition (grid domain decomposition for parallelization) ~~as the model, and needs to develop new code when the model and DA implementation use~~with the corresponding model. For a case different from the default (e.g., a DA method does not use the same process set with the first ensemble member or uses a parallel ~~decompositions. Moreover, efforts should be made to enable the software compilation system of the model to compile the code of the DA methods.~~decomposition different from the corresponding model), users will be required to develop extra codes for implementing the corresponding data exchange functionality following the rules of PDAF (e.g., the PDAF

**Formatted:** Indent: First line: 0 ch

communicators of COMM_COUPLE, COMM_FILTER, COMM_CPLMOD and COMM_MODEL, and the formats of PDAF APIs).

Most DA software consists of parallel programs that generally can be accelerated by using more processor cores. When running an ensemble DA algorithm for a component model in an ensemble run, all ensemble members of the component model are synchronously waiting the result of the DA algorithm. ~~The DA algorithm therefore can be accelerated by using~~Therefore, all the processor cores corresponding to all ensemble members of the component model~~. Although PDAF enables a DA algorithm to run in parallel, it only makes~~ can be used to accelerate the ~~processor cores of the first ensemble member available to the DA algorithm and forces the processor cores used by other ensemble members to idle when running the DA algorithm.~~

## ~~3 Overall design of the~~ DA algorithm. To develop a new framework

~~The motivation outlined in Sect. 2 shows that further work is needed to overcome the technical challenges in developing a coupled ensemble DA system. Besides using the online implementation, our new common framework that targets~~ for weakly coupled ensemble ~~DA as a first step~~data assimilation, we should ~~achieve the following functionalities.~~

~~1) It can either generate the local communicators without any preconditions of process layout, or use existing local communicators that have already been generated outside the framework.~~

~~2) It automatically handles the~~ target an improved implementation of the data exchange ~~between the component models ensemble members and the DA algorithm to reduce the user's code development, and enables a DA algorithm to use its own compiling system for greater independence of the model and the DA algorithm.~~

~~3) It enables a DA algorithm to use~~functionality, which does not use global communications, enables a DA method to share almost all the processes of ~~all ensemble members of the corresponding component models for acceleration.~~

the model ensemble and does not require users to develop extra codes in any case. When a DA algorithm uses ~~the processes of all ensemble members rather than just the first~~a process set different from a model ensemble member, the DA algorithm ~~should~~will use a parallel decomposition that differs from the model ensemble members~~ for higher parallelism~~. Thus, the data exchange between the DA algorithm and a component model ensemble member will introduce a challenge of transferring fields between ~~the whole set of processes and a subset of processes~~different process sets with different parallel decompositions.

Fortunately, such a challenge has already been overcome by most existing couplers ~~for earth system modelling~~ (Craig et al., 2012; Valcke, 2012; Liu et al., 2014; Craig et al., 2017; Liu et al., 2018b~~).~~), each of which can transfer data between different process sets with different parallel decompositions without global communications. We therefore use the C-Coupler2.0 (Liu et al., 2018b), the latest version of the Community Coupler (C-Coupler), as the foundation for developing ~~our new framework.~~DAFCC1. Moreover, C-Coupler2.0 has more functionalities that ~~the new framework~~DAFCC1 can benefit from. For example, C-Coupler2.0 can handle data exchange of 3-D or even 4-D fields where the source and destination fields can have different dimension orders (e.g., vertical+horizontal at the source field, and horizontal+vertical at the destination field). It will be convenient to combine ensemble members of a coupled model into a single MPI program based on C-Coupler2.0

because each ensemble member can be registered as a component model of C-Coupler2.0~~, and C-Coupler2.0 can use existing local communicators of ensemble members or generate the local communicators of ensemble members without any preconditions on process layout~~. Most operations for achieving data exchange can be generated automatically because C-Coupler2.0 can generate coupling procedures between two process sets~~ of processes~~ even when the two sets are overlapping.

~~We based the design of the~~The architecture of ~~our new framework~~ DAFCC1 based on C-Coupler2.0~~, as~~ is shown in Fig. 1. It includes a set of new managers (i.e., DA algorithm integration manager, ensemble component manager, ensemble data exchange operation manager, online DA procedure manager, and ensemble DA configuration manager) and the new APIs corresponding to these managers. The DA algorithm integration manager enables the user to conveniently develop driving interfaces for a DA algorithm based on a set of new APIs that enables the DA algorithm to register its input and output fields and to obtain various information from the model. A DA algorithm can include a set of procedures such as observation operators and analysis modules, each of which can be called by the model separately. The framework ~~recommends~~uses the dynamic link library (DLL) technique for the connection of a DA algorithm program to a model, so that the original configuration and compilation systems of a DA algorithm can generally be preserved for greater independence of the DA algorithms from the models, and for less work in integrating a DA algorithm. The ensemble component manager is responsible for generating and managing the communicator of ensemble members of ~~a~~each component model. The online DA procedure manager provides several APIs that enable the ensemble members of a component model to initialize, run and finalize a DA algorithm cooperatively, and automatically handles the data exchanges between the ensemble members and the DA algorithm with a set of operations. The ensemble DA configuration manager enables the user to flexibly specify the DA algorithm, DA frequency and the operations for the data exchange in a DA simulation through a configuration file.

Guided by the architecture in Fig. 1, we implemented the new framework (see ~~Sect. 4~~Section 3 for detailed implementation), which enables a coupled ensemble DA system to achieve the following features~~ that indicate that all our targets have been fully achieved.~~:

~~1) There is no restriction of process layout among ensemble members of a coupled model or among component models in each coupled model ensemble member. For example, given the three ensemble members of the coupled model that consists of four component models in Fig. 2, the first component model can take processes #0–#2, #2–#4, and #4–#6 in the first, second, and third ensemble members of the coupled model, respectively.~~

1) Each component model can use different instances of DA algorithms online independently~~;~~, and the execution of a DA algorithm in ~~a~~ the MPI processes of a component model does not force ~~the~~other MPI processes ~~of other component models~~ to be idled. For example, components 1, 2, and 4 in Fig. 2 use DA algorithms at different frequencies, while component 3 does not use DA.
2) Given a common DA algorithm, it can be used by different component models under different instances with different configurations; e.g., the fields assimilated, the observational information used, and the frequency. In Fig. 2 for example, components 2 and 4 use different instances of the same DA algorithm 2 independently.

3) An instance of a DA algorithm can either use the processes of all ensemble members of the same component model cooperatively or use the processes of each ensemble member separately. For example, each DA algorithm instance in Fig. 2 uses the processes of all ensemble members of the corresponding component model cooperatively, except procedure 1 of DA algorithm 1 that uses the processes of each ensemble member of component 1 separately.

4) Besides employing the DLL technique for integrating DA algorithm programs, ~~scripts are allowed to conduct necessary process control, which further increases~~a configuration file is designed for increasing the flexibility and convenience in ~~integrating~~using a DA algorithm (see ~~Sect. 4~~Section 3.4 for detailed implementation).

## ~~4~~3 Implementation of DAFCC1

In this section, we will detail the implementation of DAFCC1 in terms of the ensemble component manager, DA algorithm integration manager, online DA procedure manager, and ensemble DA configuration manager. Moreover, we will provide an example of how to use DAFCC1 to develop a DA system.

### ~~4~~3.1 Implementation of the ensemble component manager

In C-Coupler2.0, the model coupling resources, including MPI communicators, time steps, timers, model grids, parallel decompositions, coupling field instances, and coupling interfaces, are associated with each component model that is registered to C-Coupler2.0 via the API *CCPL_register_component*. When running an ensemble of a model in a single MPI run, each ensemble member should be used as a separate component model. In C-Coupler2.0, model names are the keywords to distinguish different component models. To distinguish different ensemble members of a model that generally share the same code or executable, we update the API *CCPL_register_component* to implicitly generate different names of ensemble members by appending the ID of each ensemble member to the model name (the parameter list of the API *CCPL_register_component* is unchanged). The ID of an ensemble member is given as the last argument (formatted as "CCPL_ensemble_*{ensemble numbers}_{member ID}*") of the corresponding executable when submitting an MPI run (see Fig. 3 as an example), where "*ensemble numbers*" marks the number of ensemble members and "*member ID*" marks the ensemble member ID of the current component.

Given an ensemble run of a coupled model, all ensemble members of the component models of the coupled model can be organized as one level of models (see Fig. 4a), although we recommend constructing two hierarchical levels of models with the first level corresponding to all ensemble members of the coupled model and each ensemble member including the component models at the second level (Fig. 4b), because the hierarchical organization retains the original architecture of the coupled model through a simple additional registration of the coupled model to C-Coupler2.0.

As a DA algorithm that handles ensemble fields can run on the MPI processes of all ensemble members of a component model (Fig. 2), an ensemble-set component model that covers all ensemble members of the component model is required to use the DA algorithm (Fig. 4b). The ensemble component manager provides the capability to generate an ensemble-set component model, which does not introduce global synchronization and only involves the ensemble members of the corresponding component model.

### 43.2 Implementation of the DA algorithm integration manager

A pair of a model and a DA algorithm have essentially the relationship between a caller and a callee in a program, where the callee generally declares a list of arguments that includes a set of input and output variables, while a caller should match the argument list of the callee when calling the callee (herein, the model is referred to as the host model of the DA algorithm). For a caller and a callee that are in the same native code, a corresponding compiler can guarantee the consistency of the argument list between them, regardless of the data structure of each argument. However, compilers cannot guarantee such consistency between a host model and a DA algorithm that is enclosed in a DLL but not in the native code of the host model.

To address the above challenge, we designed and developed a new solution for passing arguments between a host model and a DA algorithm, and tried to make such a solution as flexible as possible to improve the flexibility of DAFCC1 in serving various DA algorithms. There are three driving subroutines for initializing, running, and finalizing a DA algorithm; their subroutine names share the name of the DA algorithm as the prefix and are distinguished by different suffixes. We tried to make the explicit argument list of each driving subroutine as simple as possible (e.g., only a few integer arrays), and developed a set of C-Coupler APIs for flexibly passing implicit arguments between the host model and the DA algorithm. Based on these APIs, the DA algorithm can obtain the required information from the host model and the grids via C-Coupler2.0 and can also declare any field instances that the DA algorithm has registered to C-Coupler2.0 as implicit input or output arguments, at the initialization stage of the DA algorithm. Figure 5 shows an example of the driving subroutines where the running and finalization driving subroutines are very simple. In the initialization driving subroutine, besides the original functionalities of the DA algorithm such as determining parallel decompositions, allocating memory space for variables and other operations for initialization, there are additional operations for obtaining information from the host model and grids using C-Coupler2.0, registering the parallel decompositions, required grids, and field instances to C-Coupler2.0, and declaring the field instances as implicit input or output arguments. In the running driving subroutine *DA_CCPL_RUN*, there are no explicit calls for data exchange, because the data from the model ensemble to the DA algorithm is transferred automatically and implicitly by DAFCC1 before running *DA_CCPL_RUN*, while the data from the DA algorithm to the model ensemble is transferred automatically and implicitly after running *DA_CCPL_RUN*.

The use of DAFCC1 requires some native code of a DA algorithm to be further updated accordingly. For example, the original communicator of the DA algorithm needs to be replaced with the communicator of the host model that can be obtained through the corresponding C-Coupler API, and the original I/O accesses for the model data in the DA algorithm can be turned off.

### 43.3 Implementation of the online DA procedure manager

To enable different component models to use the same DA algorithm but with different configurations, a component model can use a distinct instance of a DA algorithm with the corresponding configuration information. Corresponding to the three driving subroutines of a DA algorithm, there are three APIs (*CCPL_ensemble_procedures_inst_init*, *CCPL_ensemble_procedures_inst_run*, and *CCPL_ensemble_procedures_inst_finalize*) that enable a host model to initialize, run, and finalize the DA algorithm instance, and handle the data exchanges between the host model and the DA algorithm instance automatically. When a component model initializes, runs, or finalizes a DA algorithm instance, all ensemble members of this component model should call the corresponding API at the same time.

#### 43.3.1 API for initializing a DA algorithm instance

The API *CCPL_ensemble_procedures_inst_init* for initializing a DA algorithm instance is designed and implemented with the following steps.

1) Determine the host model of the DA algorithm instance according to the corresponding information in the configuration file. If the DA algorithm instance is an individual algorithm that operates on the data of each ensemble member separately (e.g., Procedure 1 of DA algorithm 1 in Fig. 2), each ensemble member will be a host model. Otherwise (i.e., the DA algorithm instance is an ensemble DA algorithm that operates on the data of the ensemble set; e.g., Procedure 2 of DA algorithm 1 in Fig. 2), the host model will be the ensemble-set component model that will be generated automatically by the ensemble component manager.
2) Prepare information from the host model, such as model grids, parallel decompositions, and field instances, which the initialization driving subroutine of the DA algorithm can obtain via the corresponding APIs.
3) Initialize the corresponding DA algorithm instance according to the corresponding algorithm name and DLL name specified in the corresponding configuration file, where the corresponding DLL will be linked to the host model and the corresponding initialization driving subroutine in the DLL will be called. This implementation enables the user to conveniently change the DA algorithms used in simulations via the configuration file without modifying the code of the model.
4) Set up data exchange operations according to the input or output fields of the DA algorithm instance declared in the initialization driving subroutine via the corresponding APIs. The data exchange is divided into two levels: the data exchange between the ensemble members and DAFCC1, and the data exchange between DAFCC1 and the DA algorithm. The data exchange between DAFCC1 and the DA algorithm instance is simply achieved by the import/export interfaces of C-Coupler2.0, which flexibly rearrange the fields in the same component model between different parallel decompositions. If the DA algorithm instance is an ensemble algorithm, the data exchange between the ensemble members and DAFCC1 is also handled by the import/export interfaces of C-Coupler2.0, which flexibly transfer the same fields between different component models (each ensemble member and the ensemble set are different component models).

Otherwise, the data exchange between the ensemble members and DAFCC1 is simplified to a data copy. DAFCC1 will hold a separate memory space for each model field relevant to the DA algorithm, which enables a DA algorithm instance to use instantaneous model results or statistical results (i.e., mean, maximum, cumulative, and minimum) in a time window, and enables an ensemble DA algorithm instance to use aggregated results or statistical results (ensemble-mean, ensemble-anomaly, ensemble-maximum, or ensemble-minimum) from ensemble members. The sets of data exchange operations for the input and output fields of the DA algorithm instance are generated separately.

Consistent with the functionalities in the above steps, the API *CCPL_ensemble_procedures_inst_init* includes the following arguments.

- The *ID* of the current ensemble member that calls the API, and the common full name of the ensemble members, which is used for determining the host model of the DA algorithm. When registering a component model to C-Coupler2.0, its *ID* is allocated and its unique full name formatted as "*parent_full_name@model_name*" is generated, where "*model_name*" is the name of the component model, and "*parent_full_name*" is the full name of the parent component model (if any). Given that the names of the coupled model and the component model 1 in Fig. 4 are "*coupled*" and "*comp1*", respectively, in the one-level model hierarchy in Fig. 4a, the full names of ensemble members of the component model 1 are "*comp1_1*" to "*comp1_N*" and the common full name is "*comp1_**" where "*" is a wildcard, while in the two-level model hierarchy in Fig. 4b the full names of ensemble members of the component model 1 are "*coupled_1@comp1*" to "*coupled_N@comp1*" and the common full name is "*coupled_*@comp1*".
- The name of the DA algorithm instance, which is the keyword of the DA algorithm instance and also specifies the corresponding configuration information. Different DA algorithm instances can correspond to different DA algorithms or the same DA algorithm. For example, the component models 2 and 4 use different instances of the same DA algorithm in Fig. 2.
- A list of model grids and parallel decompositions, which are optional arguments that enable the DA algorithm instance to obtain grid data and use the same parallel decompositions as the host model.
- A list of field instances, which specify the model fields that can be used for assimilation. This list should cover all input or output fields of the DA algorithm.
- An optional integer array of control variables that can be obtained by the DA algorithm instance via the corresponding APIs.
- An annotation, which is a string giving a hint for locating the model code of the API call corresponding to an error or warning, is recommended but not mandatory, and should be provided by the user.

### 43.3.2 API for running a DA algorithm instance

The API *CCPL_ensemble_procedures_inst_run* is responsible for running a DA algorithm instance with the following steps.

1) Executing the data exchange operations for the input fields of the DA algorithm instance. This step automatically transfers the input fields from each ensemble member of the corresponding component model to DAFCC1 and then from DAFCC1

to the DA algorithm instance, where the statistical processing regarding the time window or the ensemble is done at the same time.

2) Executing the DA algorithm instance through calling the running driving subroutine of the DA algorithm.

3) Executing the data exchange operations for the output fields of the DA algorithm instance. This step automatically transfers the output fields from the DA algorithm instance to DAFCC1 and then from DAFCC1 to each ensemble member of the corresponding component model.

Each DA algorithm instance has a timer specified via the configuration information, which determines when the DA algorithm instance is run. The *CCPL_ensemble_procedures_inst_run* can be called for the DA algorithm instance at each time step, while the above three steps will be executed only when the corresponding timer is on. To store the input data such as the observational information, a DA algorithm instance can either share the working directory of its host model or use its own working directory specified via the configuration information. The API *CCPL_ensemble_procedures_inst_run* will change and then recover the current directory for calling the running driving subroutine of the DA algorithm, if necessary.

#### 43.3.3 API for finalizing a DA algorithm instance

The API *CCPL_ensemble_procedures_inst_finalize* is responsible for finalizing a DA algorithm instance through calling the finalization driving subroutine of the DA algorithm.

### 43.4 Implementation of the ensemble DA configuration manager

The configuration information of all DA algorithm instances used in a coupled DA simulation is contained in an XML configuration file (e.g., Fig. 6), and each DA algorithm instance has a distinct XML node (e.g., the XML node "da_instance" in Fig. 6, where the attribute "name" is the name of the DA algorithm instance and also the keyword to match the name of the DA algorithm instance in API "*CCPL_ensemble_procedures_inst_init"*), which enables the user to specify the following configurations.

1) The DA algorithm specified in the XML node "external_procedures" in Fig. 6, where the attribute "dll_name" specifies the dynamic link library, and the attribute "procedures_name" specifies the name of the DA algorithm, which will be used to choose the driving subroutines. When the user seeks to change the DA algorithm used by a component model, it is only necessary to modify the XML node "external_procedures" in most cases.

2) The periodic timer specified in the XML node "periodic_timer" in Fig. ~~6.~~6, which enables users to flexibly set the frequency as well as the model time of running the corresponding DA algorithm. Besides the attribute "period_unit" and "period_count" for specifying the period of the timer, the user can specify a lag via the attribute "local_lag_count". For example, given a periodic timer <"period_unit"="hours", "period_count"=6, "local_lag_count"=3>, its period is 6 hours, and it will not be on at the $0^{th}$, $6^{th}$, and $12^{th}$ hours, but instead on at the $3^{rd}$, $9^{th}$, and $15^{th}$ hours due to the "local_lag_count" of 3.

3) Statistical processing of input fields specified in the XML node “field_instances” in Fig. 6, where the attribute “time_processing” specifies the statistical processing in each time window determined by the periodic timer and the attribute “ensemble_operation” specifies the statistical processing among ensemble members. For an individual DA algorithm, the attribute “ensemble_operation” should be set to “*none*”. Besides the default specification of statistical processing shared by all fields, a field can have its own statistical processing specified in a sub node of the XML node “field_instances”.

4) The working directory and the scripts for pre- and post-assimilation analysis (e.g., for processing the data files of observational information) optionally specified in the XML node “processing_control” in Fig. 6. When the working directory is not specified, the DA algorithm instance will use the working directory of its host model. The script specified in the sub XML node “pre_instance_script” will be called by the root process of the host model before the API *CCPL_ensemble_procedures_inst_run* calls the DA algorithm, and the script specified in the sub XML node “post_instance_script” will be called by the root process of the host model after the DA algorithm run finishes.

## ~~5 A sample~~4 An example weakly coupled ensemble DA system based on DAFCC1

To provide further information on how to use DAFCC1 and for validating and evaluating DAFCC1, we developed ~~a sample~~an example weakly coupled ensemble DA system by combining the ensemble DA system GSI/EnKF (Shao et al., 2016; Liu et al., 2018b) and a regional **F**irst **I**nstitute of **O**ceanography **A**tmosphere-**O**cean-**W**ave (FIO-AOW) coupled model ~~which is referred as FIO-AOW~~ (Zhao et al., 2017; Wang et al., 2018). GSI/EnKF mainly focuses on regional numerical weather prediction (NWP) applications coupled with the Weather Research and Forecasting (WRF) model (Wang et al., 2014), while FIO-AOW consists of WRF, the Princeton Ocean Model (POM; Blumberg and Mellor 1987; Wang et al., 2010), the MArine Science and NUmerical Modeling wave model (MASNUM; Yang et al., 2005; Qiao et al, 2016), and all the above three model components are coupled together by using C-Coupler (Liu et al., 2014, 2018b). There are two main steps in developing the ~~sample~~example system.

1) We developed an ensemble DA sub-system of WRF by adapting GSI/EnKF to DAFCC1. This sub-system helps validate DAFCC1 and evaluate the improvement in performance obtained by DAFCC1 (~~Sect. 6~~Section 5).

2) We merged the above sub-system and FIO-AOW to produce the ~~sample~~example DA system~~, which~~ that only computes atmospheric analyses corresponding to WRF currently. This system demonstrates the ~~effectiveness~~correctness of DAFCC1 in developing a weakly coupled ensemble DA system.

### ~~5~~4.1 An ensemble DA sub-system of WRF

#### ~~5~~4.1.1 Brief introduction to GSI/EnKF

GSI/EnKF combines a variational DA sub-system (GSI; Shao et al., 2016) and an ensemble DA sub-system (EnKF; Liu et al., 2018a~~).~~), which can be used as a variational, a pure ensemble or a hybrid DA system sharing the same observation operator in

**Formatted:** Justified

**Formatted:** Font: Bold

**Formatted:** Font: Bold

**Formatted:** Font: Bold

the GSI codes. It provides two options for calculating analysis increments for ensemble DA; i.e., a serial Ensemble Square Root Filter (EnSRF) algorithm (Whitaker et al., 2012) and a Local Ensemble Kalman Filter (LETKF) algorithm (Hunt et al., 2007). ~~It can be used as a~~In this paper, we use the pure ensemble DA system without using variational DA, where GSI is used as the observation operator that calculates the difference between model variables and observations on the observation space and ~~EnKF calculates analysis increments and updates model variables. In this paper, we focus on the pure ensemble DA system for developing the sample DA system~~EnSRF is chosen for calculating atmospheric analyses and updating atmosphere model variables.

Figure 7a shows the flowchart for running the pure ensemble DA system of the WRF model in a DA window. It consists of the following main steps that are driven by scripts, while the data exchanges between these main steps are achieved via data files.

1) Ensemble model forecast. An ensemble run of WRF is initiated or restarted from a set of input data files, and then is stopped after producing a set of output files (called model background files hereafter) for DA and for restarting the ensemble run in the next DA window.
2) Calculating the ensemble mean of model DA variables. A separate executable is initiated for calculating the ensemble mean of each DA variable based on the model background files, and then outputs the ensemble mean to a new background file.
3) Observation operator for the ensemble mean. GSI is initiated as the observation operator for the ensemble mean. It takes the ensemble mean file, files of various observational data (e.g., conventional data, satellite radiance observations, GPS radio occultations, and radar data) and multiple fixed files (e.g., statistic files, configuration files, bias correction files, and CRTM coefficient files) as input, and produces an observation prior (observation innovation) file for the ensemble mean and files containing observational intermediate information (e.g., bias correction and thinning).
4) Observation operator for each ensemble member. GSI is initiated as the observation operator for each ensemble member. It takes the background file of the corresponding ensemble member, the fixed files and the observational intermediate information files as input, and produces an observation prior file for the corresponding ensemble member.
5) EnKF for calculating analysis increments. EnKF is initiated for calculating analysis increments of the whole ensemble. It takes the model background files, the observation prior files and the fixed files as input, and finally updates model background files with the analysis increments. The updated model background files are used for restarting the ensemble model forecast in the next DA window.

#### ~~5~~4.1.2 Adapting GSI/EnKF to DAFCC1

~~To adapt~~When adapting GSI/EnKF to DAFCC1, ~~we used the WRF~~ an ensemble-set component model ~~for~~derived from the ensemble forecast of WRF (corresponding to the first main step ~~of the ensemble forecast~~in Section 4.1.1) is generated as the host model that drives the DA algorithm instances corresponding to the remaining main steps. As shown in Fig. 8, ~~there are~~ three DA instances corresponding to the last three main steps in Section 4.1.1 (i.e., observation operator for the ensemble mean, observation operator for each ensemble member, and EnKF for calculating analysis increments) are enclosed in DLLs, without ~~the~~a DA algorithm instance corresponding to the second main step in Section 4.1.1. This is because the online DA procedure manager of DAFCC1 enables a DA algorithm instance to automatically obtain the ensemble mean of model DA variables (~~Sect. 4~~Section 3.3). Although both the third and fourth main steps correspond to the same GSI, they are transformed into two different DA algorithm instances, because the third is an ensemble algorithm (i.e., it operates on the data of the ensemble set) and the fourth is an individual algorithm (i.e., it operates on the data of each ensemble member). Moreover, we compiled the same GSI code into two separate DLLs, each of which corresponds to one of these two instances, to enable these two instances to use different memory space.

For each DA algorithm instance, three driving subroutines and the corresponding configuration were developed~~.~~ (Fig. 8). In fact, the two instances corresponding to GSI share the same driving subroutines but use different configurations (especially regarding the specification of "ensemble_operation"). To enable the GSI code and EnKF code to be used as DLL, we made the following slight modifications to the code.

1) We turned off the MPI initialization and finalized and replaced the original MPI communicator with the MPI communicator of the host model that can be obtained via DAFCC1.
2) We obtained the required model information and the declared input/output fields via DAFCC1, and turned off the corresponding I/O accesses.

To drive the DA algorithm instances, the WRF code was updated with the new subroutines for initializing, running, and finalizing all DA algorithm instances. Moreover, the functionality of outputting model background files can be turned off, because the data exchanges between WRF and the DA algorithm instances are automatically handled by DAFCC1 and the WRF ensemble can be run continuously throughout DA windows without stopping and restarting. As a result, DAFCC1 saves sets of data files and the corresponding I/O access operations, while only the observation files, fixed files, and the files for the data exchanges among the DA algorithm instances are reserved (compare Fig. 7b and Fig. 7a).

### ~~5~~4.2 ~~Sample~~Example ensemble DA system of FIO-AOW

FIO-AOW, which previously used C-Coupler1 (Liu et al., 2014) for model coupling, has already been upgraded to C-Coupler2.0 by us. As GSI/EnKF and FIO-AOW share WRF, the development of the ~~sample~~example ensemble DA system of FIO-AOW can significantly benefit from the DA system of WRF, and it only took the following steps to construct the ~~sample~~example ensemble DA system.

1) Using the ensemble component manager, set up the two hierarchical levels of models shown in Fig. 9; i.e., the first level corresponds to all ensemble members of FIO-AOW while each member includes its three component models at the second level.
2) Merge the model code modifications, the DA algorithm instances, and configurations in the DA system of WRF into the ~~sample~~example ensemble DA system FIO-AOW.

As well as being described by the flowchart involving the WRF and the DA algorithm instances in Fig. 7b, the ~~sample~~example ensemble DA system of FIO-AOW follows the process layout in Fig. 10, which is essentially a real case of the process layout in Fig. 2.

## ~~6~~5 Validation and evaluation of DAFCC1

In this section, we evaluate the ~~effectiveness~~correctness of DAFCC1 in developing a weakly coupled ensemble DA system based on the ~~sample~~example ensemble DA system (referred to as the full-~~sample~~example-DA-system hereafter) described in ~~Sect. 5~~Section 4, and will also validate DAFCC1 and evaluate the impact of DAFCC1 in accelerating DA based on the sub-system with WRF and GSI/EnKF (WRF-GSI/EnKF hereafter).

### ~~6~~5.1 Experimental setup

The ~~sample~~example ensemble DA system used in this validation and evaluation consists of WRF Version 4.0 (Wang et al., 2014), GSI version 3.6 and EnKF version 1.2, and the corresponding versions of POM and MASNUM used in FIO-AOW (Zhao et al., 2017; Wang et al., 2018). In EnKF version 1.2 the default settings are used; i.e., the EnSRF algorithm is used to calculate analysis increments for ensemble DA, the inflation factor is 0.9 without smoothing, and the covariance is localized by distance correlation function with horizontal localization radius of 400 km and vertical localization scale coefficient of 0.4. The ~~sample~~example ensemble DA system is run on a supercomputer of the Beijing Super Cloud Computing Center (BSCC~~).~~) with the lustre file system. Each computing node on the supercomputer includes two Intel Xeon E5-2678 v3 CPUs (Intel(R) Xeon(R) CPU), with 24 processor cores in total, and all computing nodes were connected with an InfiniBand network. The codes were compiled by an Intel Fortran and C++ compiler at the optimization level O2, using an Intel MPI library. A maximum 3200 cores are used for running the ~~sample~~example ensemble DA system.

The WRF-GSI/EnKF integrates over an approximate geographical area generated from a Lambertian projection of the area 0°–50°N, 99°–160°E with center point at 35°N, 115°E. Initial fields and lateral boundary conditions (at 6 hour intervals) for the ensemble run of WRF are taken from the NCEP Global Ensemble Forecast System (GEFS) (at 1° × 1° resolution) (https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-ensemble-forecast-system-gefs). To configure WRF, an existing physics suite 'CONUS' (https://www2.mmm.ucar.edu/wrf/users/ncar_convection_suite.php) and 32 vertical sigma layers with the model top at 50 hPa are used. One-day integration on June 1st, 2016 is used for running the WRF-GSI/EnKF. NCEP global GDAS Binary Universal Form for the Representation of meteorological data (BUFR; https://www.emc.ncep.noaa.gov/mmb/data_processing/NCEP_BUFR_File_Structure.htm) and Prepared BUFR (https://www.emc.ncep.noaa.gov/mmb/data_processing/prepbufr.doc/document.htm), including conventional observation data and satellite radiation data, are assimilated every 6 hours (i.e., at 0000, 0600, 1200, and 1800 UTC). The air temperature (T), specific humidity (QVAPOR), longitude and latitude wind (UV), and column disturbance dry air quality (MU) are the variables ~~used for DA.~~analyzed in the data assimilation. The WRF-GSI/EnKF experiments are classified into four sets, where variations of horizontal resolution (and the corresponding time step), number of ensemble members and process number (each process runs on a distinct processor core) are considered (Tables 1 and 2).

All component models of the full-~~sample~~example-DA-system integrate over the same geographical area (0°–50°N, 99°–150°E) with the same horizontal resolution of 0.5° × 0.5° but different time steps (100 s for WRF and 300 s for POM and MASNUM, coupled by C-Coupler2.0 at 300 s intervals). More details of the model configurations can be found in Zhao et al. (2017). The configuration of initial fields, lateral boundary conditions, and observations of WRF for the ensemble run of the full-~~sample~~example-DA-system are the same as for WRF-GSI/EnKF. The full-~~sample~~example-DA-system integrates over 3 days (June 1st to 3rd, 2016), while the first model day is considered as spin-up, and DA is performed every 6 hours in the last two model days with T, UV and MU as DA variables.

### ~~6~~5.2 Validation of DAFCC1

To validate DAFCC1, we compare the outputs of the two versions of WRF-GSI/EnKF: the original WRF-GSI/EnKF (hereafter offline WRF-GSI/EnKF; https://dtcenter.org/community-code/gridpoint-statistical-interpolation-gsi/community-gsi-version-3-6-enkf-version-1-2) and the new version of WRF-GSI/EnKF with DAFCC1 (hereafter online WRF-GSI/EnKF) introduced in ~~Sect. 5~~Section 4.1. As DAFCC1 improves only the data exchanges between a model and the DA algorithms, the simulation results of an existing DA system should not change when it is adapted to use DAFCC1. We therefore employ a ~~bitwise identical~~ validation standard ~~for validating DAFCC1, which means~~that the ~~online~~ WRF-GSI/EnKF ~~achieves exactly the same~~ with DAFCC1 keeps bit-identical result ~~as~~with the original offline WRF-GSI/EnKF. DAFCC1 passes the validation test with all experimental setups in Table 2, where the binary data files output by WRF at the end of the 1-day integration are used for the comparison.

### ~~6~~5.3 Impact in accelerating an offline DA

WRF-GSI/EnKF is further used to evaluate the impact of DAFCC1 in accelerating an offline DA, by comparing the execution time of the offline and online WRF-GSI/EnKF under each experimental setup in Table 2. Considering that all ensemble members of the online WRF-GSI/EnKF are integrated simultaneously, we run all ensemble members of the offline WRF-GSI/EnKF concurrently through a slight modification to the corresponding script, in order to make a fair comparison.

The impact of varying the number of ensemble members is evaluated based on Set 1 in Table 2. DAFCC1 obviously accelerates WRF-GSI/EnKF, and can achieve higher performance speedup with more ensemble members (Fig. 11a). This is because DAFCC1 significantly accelerates the DA for both GSI and EnKF (Fig. 11b-d). Similarly, DAFCC1 significantly accelerates the DA as well as WRF-GSI/EnKF under different process numbers (Fig. 12, corresponding to Set 2 in Table 2) and resolution (Fig. 13, corresponding to Set 3 in Table 2). Considering that more processor cores are generally required to accelerate the model run under higher resolution, we also make an evaluation based on Set 4 in Table 2, where concurrent changes in resolution and process number are made to achieve similar numbers of grid points per process throughout the experimental setups. This evaluation also demonstrates the ~~effectiveness~~correctness of DAFCC1 in accelerating the DA as well as WRF-GSI/EnKF (Fig. 14).

The performance speedups observed from Figs. 11–14 result mainly from the significant decrease in I/O accesses. Although the online WRF-GSI/EnKF still has to access the observation prior files (~~Sect. 5~~Section 4.1.1 and Fig. 7b), most I/O accesses correspond to the model ensemble background files and model ensemble analysis files, and these I/O accesses have been eliminated by DAFCC1 (Table 3). Moreover, more I/O accesses can be saved under higher resolution or more ensemble members.

~~6~~We note that, the execution time of the offline GSI in Fig. 11c increases when using more ensemble members. This is reasonable, because more ensemble members introduce more I/O accesses, as shown in Table 3. We also note that, the execution time of the offline and online EnKF in Fig. 11d and Fig. 12 increases when using more ensemble members. This is because the current parallel version of EnKF does not achieve good scaling performance, and thus longer execution time can be observed when EnKF uses more processor cores.

### 5.4 ~~Effectiveness~~Correctness in developing a weakly coupled ensemble DA system

We have successfully run the full-~~sample~~example-DA-system with ten ensemble members, which enables us to investigate the model variables before and after DA. We find that changes to the model variables resulting from DA can be observed; e.g., the bias regarding T is slightly decreased and the bias regarding UV is more obviously decreased after using DA, as shown in Fig. 15.

## ~~7~~6 Conclusions and discussion

In this paper, we propose a new common, flexible and efficient framework for weakly coupled ensemble data assimilation based on C-Coupler2.0, DAFCC1. It provides simple APIs and a configuration file format to enable users to conveniently integrate a DA method into a model as a procedure that can be directly called by the model, while still guaranteeing software independence between the model and the DA method. The ~~sample~~example weakly coupled ensemble DA system in ~~Sect. 5~~Section 4 and the evaluations in ~~Sect. 6~~Section 5 demonstrate the ~~effectiveness~~correctness of DAFCC1 in both developing a weakly coupled ensemble DA system and accelerating ~~the~~an offline DA system. The development of a DA system that only employs a single model run but not an ensemble run can also benefit from the advantages of DAFCC1, while the functionality of data exchanges will be automatically simplified without generating ensemble-set component models for saving extra overhead.

DAFCC1 is able to automatically handle data exchanges between a model ensemble and a DA algorithm because its design and implementation significantly benefit from C-Coupler2.0, which already has the functionalities of automatic coupling generation and automatic data exchanges between different component models or within the same component model. DAFCC1 will therefore be an important functionality of the next generation of C-Coupler (C-Coupler3) ~~which~~that is planned to be released no later than 2022. ~~We~~Although the example ensemble DA system of FIO-AOW developed in this work only computes atmospheric analyses currently, the future work similar to adapting GSI/EnKF to DAFCC1 can be conducted to further enable the computation of ocean or wave analyses. Moreover, we have considered software extendibility when designing and implementing DAFCC1, which will enable us to conveniently achieve upgrades either for strongly coupled ensemble DA systems or for more types of data exchange operations in the future. As shown in Fig. 7, the I/O accesses to the observation prior files for the data exchanges between DA algorithms are still retained after using DAFCC1. Although they are not currently a performance bottleneck (Table 3), we will investigate how to avoid these types of I/O accesses when further upgrading DAFCC1.

Regarding the evaluations in ~~Sect. 6~~Section 5, we can only use at most 3200 processor cores, which limits the maximum number of cores per ensemble member. Consequently, we use relatively coarse resolutions of WRF and FIO-AOW. However, the results in Fig. 14 from the experiment Set 4 in Table 2 indicate that DAFCC1 will also obviously accelerate the DA system when using a finer resolution and more processor cores, because it will also significantly decrease I/O accesses. DAFCC1 can tackle the technical challenges in developing or accelerating a DA system, but cannot contribute to improvements in simulation results that generally depend on scientific settings which must be determined in the research environment (e.g., the DA algorithm configuration, the inflation factor, localization settings, initial states of the model ensemble run). Consequently, we did not examine the improvements in simulation results resulting from the full-~~sample~~example-DA-system based on various variables in ~~Sect. 6~~Section 5.4, but only made a simple comparison of simulation results demonstrating that the full-~~sample~~example-DA-system can successfully run and produce simulation results.

The offline implementation of a DA system that relies on disk files and restart functionalities of models and DA algorithms can be a robust strategy when it comes to massively parallel computing where the risk of random task failures generally increases with more processor cores used by a task, because a failed task that corresponds to an ensemble member can be resumed from the corresponding restart files. The online implementation that unifies all ensemble members into a task enables to significantly increase the number of cores used by a task. At the same time of enlarging the risk of random task failures, the online implementation can decrease such risk because it can significantly reduce disk file accesses that are generally an important source of task failures. The robustness of an online implementation can be further improved through developing the restart capability of the DA system based on the restart capabilities of the model and C-Coupler2, while users are enabled to flexibly set the restart-file-writing frequency for the online implementation that can be lower than the corresponding frequency for an online implementation generally determined by observation data frequencies. Moreover, the impact of the overhead of writing restart files in an online implementation can be further decreased via asynchronous I/O support.

*Code availability.* The source code of DAFCC1 can be viewed via https://doi.org/10.5281/zenodo.3739729 (please contact us for authorization before using DAFCC1 for developing a system). The original source code and scripts corresponding to WRF and GSI/EnKF can be download from https://www2.mmm.ucar.edu/wrf/users/downloads.html and https://dtcenter.org/com-

GSI/users/downloads/index.php respectively. For the source code of FIO-AOW, please contact the authors of (Zhao et al., 2017; Wang et al., 2018). The additional codes, configurations, scripts and guidelines for developing and running the ~~sample~~example weakly coupled ensemble DA system can also be download from https://doi.org/10.5281/zenodo.3774710.

*Author contributions.* CS was responsible for code development, software testing and experimental evaluation of DAFCC1 with the ~~sample~~example DA system, contributed to the motivation and design of DAFCC1 and co-led paper writing. LL initiated this research, was responsible for the motivation and design of DAFCC1, co-supervised CS, and co-led paper writing. RL, XY and HY contributed to code development and software testing. BZ, GW, JL and FQ contributed to the development of the ~~sample~~example DA system. BW contributed to scientific requirements and the motivation, and co-supervised CS. All authors contributed to improvement of ideas and paper writing.

*Competing interests.* The authors declare that they have no conflict of interest.

*Acknowledgements*. This work was jointly supported in part by the National Key Research Project of China (grant no. 2017YFC1501903), the National Natural Science Foundation of China（grant no. 41875127) and the National Key Research Project of China (grant no. 2016YFA0602204).

**References**

Andersson, E., Haseler, J., Unden, P., Courtier, P., Kelly, G., Vasiljevic, D., and Thepaut, J.: The ECMWF implementation of three-dimensional variational assimilation (3D-Var). III: Experimental results, Q. J. Roy. Meteor. Soc., 124, 1831-1860, 1998.

Anderson, J., Hoar, T., Raeder, K., Liu, H., Collins, N., Torn, R., and Arellano, A.: The Data Assimilation Research Testbed: A Community Facility, Bull. Am. Meteorol. Soc., 90, 1283–1296, 2009.

Anderson, J. ~~L.~~ and Collins, N.: Scalable implementations of ensemble filter algorithms for data assimilation, J. Atmos. Ocean Technol., 24, 1452-1463, 2007.

Bishop, C. ~~H.~~ and Hodyss, D.: Adaptive ensemble covariance localization in ensemble 4D-VAR state estimation, Mon. Weather Rev., 139, 1241–1255, 2011.

Blumberg, A. ~~F.,~~ and ~~G. L.~~ Mellor, G.: A description of a three-dimensional coastal ocean circulation model, in Three-Dimensional Coastal Ocean Models, edited by N. S. Heaps, pp. 1–16, AGU, Washington, D. C, 1987.

Bonavita, M., Isaksen, L., and Holm, E~~. V~~.: On the use of EDA background-error variances in the ECMWF 4D-Var, Q. J. R. Meteorol. Soc., 138, 1540~~–~~1559, 2012.

Bonavita, M., Holm, E., Isaksen, L., and Fisher, M. A.: The evolution of the ECMWF hybrid data assimilation system, Q. J. Roy. Meteor. Soc., 142, 287-303, 2016.

Brown, A., Milton, S., Cullen, M., Golding, B., Mitchell, J., and Shelly, A.: Unified modeling and prediction of weather and climate: a 25 year journey, B. Am. Meteorol. Soc., 93, 1865–1877, https://doi.org/10.1175/BAMS-D-12-00018.1, 2012.

Browne, P., de Rosnay, P., Zuo, H., Bennett, A., and Dawson, A.: Weakly coupled ocean-atmosphere data assimilation in the ECMWF NWP system, Remote Sens., 11, 1-24, 2005.

Browne, P., and Wilson, S.: A simple method for integrating a complex model into an ensemble data assimilation system using MPI, Environ. Modell. Softw., 68, 122-128, 2015.

Brunet, G., Jones, S., and Ruti, P~~. M~~.: Seamless prediction of the Earth System: from minutes to months, Tech. Rep. WWOSC-2014, World Meteorological Organization, 2015.

Buehner, M., McTaggart-Cowan, R., Beaulne, A., Charette, C., Garand, L., Heilliette, S., Lapalme, E., Laroche, S., Macpherson, S.R., Morneau, J. and Zadra, A.: Implementation of deterministic weather forecasting systems based on ensemble-variational data assimilation at Environment Canada. Part I: the global system, Mon. Weather Rev., 143, 2532–2559, 2015.

Courtier, P., Andersson, E., Heckley, W~~. A~~., Vasiljevic, D., Hamrud, M., Hollingsworth, A., and Pailleux, J.: The ECMWF implementation of three-dimensional variational assimilation (3D-Var). I: Formulation, Q. J. Roy. Meteor. Soc., 124, 1783-1807, 1998.

Courtier, P., Thepaut, J., and Hollingsworth, A.: A strategy for operational implementation of 4D-Var, using an incremental approach, Q. J. Roy. Meteor. Soc., 120, 1367-1387, 1994.

Craig, A~~. P~~., Vertenstein, M., and Jacob, R.: A new flexible coupler for earth system modeling developed for CCSM4 and CESM1, Int. J. High Perform. Comput. Appl., 26, 31-42, 2012.

Craig, A., Valcke, S., and Coquart, L.: Development and performance of a new version of the OASIS coupler, OASIS3-MCT_3.0, Geosci. Model Dev., 10, 3297-3308, https://doi.org/10.5194/gmd-10-3297-2017, 2017.

Etherton, B. ~~J.~~ and Bishop, C~~. H~~.: Resilience of hybrid ensemble/3DVAR analysis schemes to model error and ensemble covariance error, Mon. Weather Rev., 132, 1065-1080, 2004.

Evensen, G.: The ensemble kalman filter: theoretical formulation and practical implementation, Ocean Dyn., 53, 343-367, 2003.

Fisher, M.: Background error covariance modelling, in: Proceedings of Seminar on Recent Developments in Data Assimilation for Atmosphere and Ocean, Reading, UK, 8-12 September 2003, 45–63, 2003.

Fujii, Y., Kamachi, M., Nakaegawa, T., Yasuda, T., Yamanaka, G., Toyoda, T., Ando, K. and Matsumoto, S.: Assimilating ocean observation data for ENSO monitoring and forecasting, in: Climate Variability - Some Aspects, Challenges and Prospects, edited by: Hannachi, A., InTechOpen, Rijeka, Croatia, 75-98, 2011.

Fujii, Y., Nakaegawa, T., Matsumoto, S., Yasuda, T., Yamanaka, G., and Kamachi, M.: Coupled climate simulation by constraining ocean fields in a coupled model with ocean data, J. Clim., 22, 5541-5557, 2009.

Gandin, L.: Objective analysis of meteorological fields. Jerusalem: Israel program for scientific translations, 1965

Gauthier, P., Charette, C., Fillion, L., Koclas, P., and Laroche, S.: Implementation of a 3D variational data assimilation system at the Canadian Meteorological Center. Part I: The global analysis, Atmos. Ocean, 37, 103-156, 1999.

Goodliff, M., Bruening, T., Schwichtenberg, F., Li, X., Lindenthal, A., Lorkowski, I., and Nerger, L.: Temperature assimilation into a coastal ocean-biogeochemical model: assessment of weakly and strongly coupled data assimilation, Ocean Dyn., 69, 1217-1237, 2019.

Hamill, T. M.: A hybrid ensemble kalman filter-3D variational analysis scheme, Mon. Weather Rev., 128, 2905, 2000.

Heinzeller, D., Duda, M. G., and Kunstmann, H.: Towards convection-resolving, global atmospheric simulations with the Model for Prediction Across Scales (MPAS): an extreme scaling experiment, Geosci. Model Dev., 8, 6987-7061, 2016.

Hoke, J. and Anthes, R.: The initialization of numerical models by a dynamic initialization technique, Mon. Weather Rev., 104, 1551–1556, 1976.

Hoskins, B.: The potential for skill across the range of the seamless weather-climate prediction problem: a stimulus for our science, Q. J. Roy. Meteor. Soc., 139, 573-584, 2013.

Houtekamer, P. L.,., and Mitchell, H. L.: Data assimilation using an ensemble kalman filter technique. Mon. Weather Rev., 126, 796-811, 1998.

Hunt, B. R., Kostelich, E. J., and Szunyogh, I.: Efficient data assimilation for spatiotemporal chaos: A local ensemble transform kalman filter, International Symposium on Physical Design, 230, 112-126, 2007.

Kalnay, E.: Atmospheric modeling, data assimilation and predictability, Cambridge University Press, Cambridge, UK, pp. 364, 2002.

Laloyaux, P., Thepaut, J., and Dee, D.: Impact of scatterometer surface wind data in the ECMWF coupled assimilation system, Mon. Weather Rev., 144, 1203-1217, 2016.

Laloyaux, P., Frolov, S., Benjamin Ménétrier, and Bonavita, M.: Implicit and explicit cross-correlations in coupled data assimilation, Q. J. R. Meteorol. Soc., 144, 2018.

Lea, D., Mirouze, I., Martin, M., King, R., Hines, A., Walters, D., and Thurlow, M.: Assessing a new coupled data assimilation system based on the met office coupled atmosphere–land–ocean–sea ice model, Mon. Weather Rev., 143, 4678-4694, 2015.

Liu, H., Hu, M., Ge, G., Stark, D., Shao, H., Newman, K., and Whitaker, J.: Ensemble Kalman Filter (EnKF) User's Guide Version 1.3, Developmental Testbed Center, available at: https://dtcenter.org/community-code/ensemble-kalman-filter-system-enkf/documentation, 80 pp, 2018a.

Liu, L., Yang, G., Wang, B., Zhang, C., Li, R., Zhang, Z., Ji, Y., and Wang, L.: C-Coupler1: a Chinese community coupler for Earth System Modeling, Geosci. Model Dev., 7, 2281-2302, doi: 10.5194/gmd-7-2281-2014, 2014.

Liu, L., Zhang, C., Li, R., Wang, B., and Yang, G.: C-Coupler2: a flexible and user-friendly community coupler for model coupling and nesting, Geosci. Model Dev., 11, 3557-3586, https://doi.org/10.5194/gmd-11-3557-2018, 2018b.

Lorenc, A~~. C~~., Ballard, S~~. P~~., Bell, R~~. S~~., Ingleby, N~~. B~~., Andrews, P~~. L~~., Barker, D~~. M~~., Bray, J~~. R~~., Clayton, A~~. M~~., Dalby, T., Li, D., Payne, T~~. J~~., and Saunders, F~~. W~~.: The Met. Office global three-dimensional variational data assimilation scheme, Q. J. Roy. Meteor. Soc., 126, 2991-3012, 2000.

Lorenc, A~~. C~~.: The potential of the ensemble Kalman filter for NWP-A comparison with 4D-VAR, Q. J. Roy. Meteor. Soc., 129: 3183-3203, 2003a.

Lorenc, A~~. C~~.: Modelling of error covariances by 4D-Var data assimilation, Q. J. Roy. Meteor. Soc., 129, 3167-3182, 2003b.

Lu, F., Liu, Z., Zhang S., and Liu Y.: Strongly coupled data assimilation using leading averaged coupled covariance (LACC). Part I: Simple model study, Mon. Weather Rev., 143, 3823–3837, doi: 10.1175/MWR-D-14-00322.1, 2015a.

Lu, F., Liu, Z., Zhang, S., Liu, Y., and Jacob, R.: Strongly coupled data assimilation using leading averaged coupled covariance (LACC). Part II: GCM Experiments, Mon. Weather Rev., 143, 4645-4659, doi: 10.1175/MWR-D-14-00322.1, 2015b.

Ma, X~~. L~~., Lu, X., Yu, M~~. Y~~., Zhu, H~~. J~~., and Chen, J.: Progress on hybrid ensemble-variational data assimilation in numerical weather prediction, J. Trop. Meteorol., 30, 1188-1195, 2014.

Mochizuki, T., Masuda, S., Ishikawa, Y., and Awaji, T.: Multiyear climate prediction with initialization based on 4D-Var data assimilation, Geophys. Res. Lett., 43, 3903-3910, 2016.

Mulholland, D., Laloyaux, P., Haines, K., and Balmaseda, M.: Origin and impact of initialization shocks in coupled atmosphere–ocean forecasts, Mon. Weather Rev., 143, 4631–4644, doi: 10.1175/MWR-D-15-0076.1, 2015.

Nerger, L. and Hiller, W.: Software for Ensemble-based Data Assimilation Systems - Implementation Strategies and Scalability, Comput. Geosci., 55, 110–118, 2013.

Nerger, L., Hiller, W., and Schröter, J.: PDAF - The Parallel Data Assimilation Framework: Experiences with Kalman filtering., in: Use of High Performance Computing in Meteorology - Proceedings of the 11. ECMWF Workshop, edited by Zwieflhofer, W. and Mozdzynski, G., pp. 63–83, World Scientific, 2005.

Nerger, L., Tang, Q., and Mu, L.: Efficient ensemble data assimilation for coupled models with the Parallel Data Assimilation Framework: ~~Example~~example of AWI-CM~~,~~ (AWI-CM-PDAF 1.0), Geosci. Model Dev~~. Discuss.,~~., 13, 4305–4321, https://doi.org/10.5194/gmd-~~2019-167, in review, 2019~~13-4305-2020, 2020.

Oke, P., Allen, J., Miller, R., Egbert, G., and Kosro, P.: Assimilation of surface velocity data into a primitive equation coastal ocean model. J. Geophys. Res, 107, 2002.

Palmer, T~~. N~~., Doblas-Reyes, F~~. J~~., Weisheimer, A. and Rodwell, M~~. J~~.: Toward seamless prediction: Calibration of climate change projections using seasonal forecasts, Bull. Am. Meteorol. Soc., 89, 459-470, 2008.

Penny, S~~. G~~., Akella, S., Alves, O., Bishop, C., Buehner, M., Chevalier, M., Counillon, F., Drper, C., Frolov, S., Fujii, Y., Kumar, A., Laloyaux, P., Mahfouf, J.-F., MArtin, M., Pena, M., de Rosnay, P., Subramanian, A., Tardif, R., Wang, Y., and Wu, X.: Coupled data assimilation for integrated Earth system analysis and prediction: Goals, Challenges and Recommendations, Tech. Rep. WWRP 2017-3, World Meteorological Organization, 2017.

Qiao, F., Zhao, W., Yin, X., Huang, X., Liu, X., Shu, Q., Wang, G., Song, Z., Liu, H., Yang, G., and Yuan, Y.: A highly effective global surface wave numerical simulation with ultra-high resolution, in: Proceedings of the International

Conference for High Performance Computing, Networking, Storage and Analysis (SC '16), IEEE Press, Piscataway, NJ, USA, doi: 10.1109/SC.2016.4, 2016.

Rabier, F., Jarvinen, H., Klinker, E., Mahfouf, J., and Simmons, A~~. J~~.: The ECMWF operational implementation of four-dimensional variational assimilation. I: Experimental results with simplified physics, Q. J. Roy. Meteor. Soc., 126, 1143-1170, 2007.

Saha, S., Moorthi, S., Pan, H., Wu, X., Wang, J., Nadiga, S., Tripp, P., Kistler, R., Woollen, J., Behringer, D., Liu, H., Stokes, D., Grumbine, R., Gayno, G., Wang, J., Hou, Y., Chuang, H., Juang, H. H., Sela, J., Iredell, M. T. R., Kleist, D., van Delst, P., Keyser, D., Derber, J., Ek, M., Meng, J., Wei, H., Yang, R., Lord, St., Van Den Dool, H., Kumar, A., Wang, W., Long, C., Chelliah, M., Xue, Y., Huang, B., Schemm, J., Ebisuzaki, W., Lin, R., Xie, P., Chen, M., Zhou, S., Higgins, W., Zou, C., Liu, Q., Chen, Y., Han, Y., Cucurull, L., Reynolds, R. W., Rutledge, G., and Goldberg, M.: The NCEP Climate Forecast System Reanalysis, Bull. Am. Meteorol. Soc., 91, 1015-1057, 2010.

Saha, S., Moorthi, S., Wu, X., Wang, J., Nadiga, S., Tripp, P., Behringer, D., Hou, Y., Chuang, H., Iredell, M., Ek, M., Meng, J., Yang, R., Mendez, M., Van Den Dool, H., Zhang, Q., Wang, W., Chen, M., and Becker, E.: The NCEP Climate Forecast System Version 2, J. Clim., 27, 2185–2208, http://dx.doi.org/10.1175/JCLI-D-12-00823.1, 2014.

Sakov, P., Counillon, F., Bertino, L., Lisæter, K. A., Oke, P. R., and Korablev, A.: TOPAZ4: an ocean-sea ice data assimilation system for the North Atlantic and Arctic, Ocean Sci., 8, 633-656, 2012.

Shao, H., Derber, J., Huang, X. Y., Hu, M., Newman, K., Stark, D., Lueken, M., Zhou, C., Nance, L., Kuo, Y. H., and Brown, B.: Bridging Research to Operations Transitions: Status and Plans of Community GSI. Bull. Amer. Meteor. Soc., 97, 1427-1440, doi: 10.1175/BAMS-D-13-00245.1, 2016.

Skachko, S., Buehner, M., Laroche, S., Lapalme, E., Smith, G., Roy, F., Surcel-Colan, D., Bélanger, J., and Garand, L.: Weakly coupled atmosphere-ocean data assimilation in the Canadian global prediction system (v1), Geosci. Model Dev., 12, 5097-5112, 2019.

Sluka, T~~. C~~., Penny, S~~. G~~., Kalnay, E~~.~~., and Miyoshi, T.: Assimilating atmospheric observations into the ocean using strongly coupled ensemble data assimilation, Geophys. Res. Lett., 43, 752-759, 2016.

Sugiura, N., Awaji, T., Masuda, S., Mochizuki, T., Toyoda, T., Miyama, T., Igarashi, H., and Ishikawa, Y.: Development of a 4-dimensional variational coupled data assimilation system for enhanced analysis and prediction of seasonal to interannual climate variations, J. Geophys. Res., 113, C10017, doi: 10.1029/2008JC004741, 2008.

Tardif, R., Hakim, G~~. J~~., and Snyder, C.: Coupled atmosphere–ocean data assimilation experiments with a low-order climate model, Clim. Dyn., 43, 1631-1643, doi: 10.1007/s00382-013-1989-0, 2014.

Tardif, R., Hakim, G~~. J~~., and Snyder, C.: Coupled atmosphere~~–~~-ocean data assimilation experiments with a low-order model and CMIP5 model data, Clim. Dyn., 45, 1415-1427, doi: 10.1007/s00382-014-2390-3, 2015.

Valcke, S., Balaji, V., Craig, A., Deluca, C., Dunlap, R., Ford, R. W., Jacob, R., Larson, J., Okuinghttons, R., Riley, G., and Vertenstein, M: Coupling technologies for Earth System Modelling, Geosci. Model Dev., 5, 1589-1596, 2012.

Vidard, P. A., Le Dimet, F. X., and Piacentini, A.: Determination of optimal nudging coefficients, Tellus, 55A, 1–15, 2003.

Wang, G., Qiao F., and Xia C.: Parallelization of a coupled wave-circulation model and its application, Ocean Dyn., 60(2), 331–339, doi:10.1007/s10236-010-0274-6, 2010.

Wang, G., Zhao, B., Qiao, F., and Zhao, C.: Rapid intensification of Super Typhoon Haiyan: the important role of a warm-core ocean eddy, Ocean Dyn, 68, 1649-1661, 2018.

Wang, W., Barker, D., Bray J., Bruye`re C., Duda M., Dudhia J., Gill D., and Michalakes J.: WRF Version 3 Modeling System User's Guide, available at http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_V3/contents.html, 2014.

Wang, X., Barker, D., Snyder, C., and Hamill, T~~. M~~.: A hybrid ETKF-3DVAR data assimilation scheme for the WRF model. Part I: Observing system simulation experiment, Mon. Weather Rev., 136, 5116–5131, 2008.

Wang, X., Parrish, D~~. D~~., Kleist, D., and Whitaker, J~~. S.~~.: GSI 3DVar-based ensemble-variational hybrid data assimilation for NCEP global forecast system: Single-resolution experiments, Mon. Weather Rev., 141, 4098 – 4117, 2013.

Whitaker, J.~~ S.~~ and Hamill, T. M. Ensemble Data Assimilation without Perturbed Observations, Mon. Weather Rev., 130, 1913-1924, 2012.

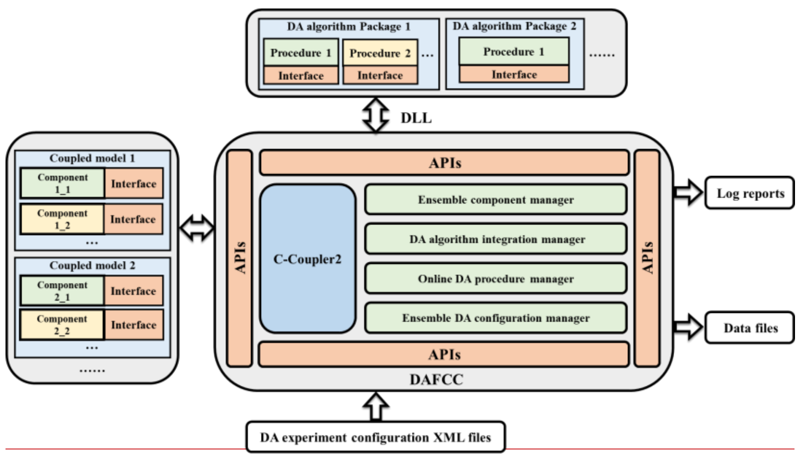
Yang, ~~Y. Z., F. Qiao, W. Zhao, Y. Teng, and Y. Yuan:~~X., Rosati, A., Zhang, S., Delworth, T., Gudgel, R., Zhang, R., Vecchi, G., Anderson, W., Chang, Y., DelSole, T., Dixon, K., Msadek, R., Stern, W., Wittenberg, A., and Zeng, F.: A predictable AMO-like pattern in the GFDL fully coupled ensemble initialization and decadal forecasting system, J. Clim., 26, 650-661, 2013.

Yang, Y., Qiao, F., Zhao, W., Teng, Y., and Yuan, Y.: MASNUM ocean wave numerical model in spherical coordinates and its application, Acta Oceanol. Sin., 27, 1–7, 2005.

Zhao, B., Qiao, F., Cavaleri, L., Wang, G., Bertotti, L., and Liu, L: Sensitivity of typhoon modeling to surface waves and rainfall, J. Geophys. Res. Oceans, 122, 1702–1723, doi:10.1002/2016JC012262, 2017.

Zhang, S., Harrison, M~~. J~~., Wittenberg, A~~. T~~., Rosati, A., Anderson, J~~. L~~., and Balaji, V.: Initialization of an ENSO forecast system using a parallelized ensemble filter, Mon. Weather Rev., 133, 3176–3201, 2005.

Zhang, S., Harrison, M~~. J~~., Rosati, A., and Wittenberg, A.: System design and evaluation of coupled ensemble data assimilation for global oceanic climate studies, Mon. Weather Rev., 135, 3541–3564, doi: 10.1175/MWR3466.1, 2007.



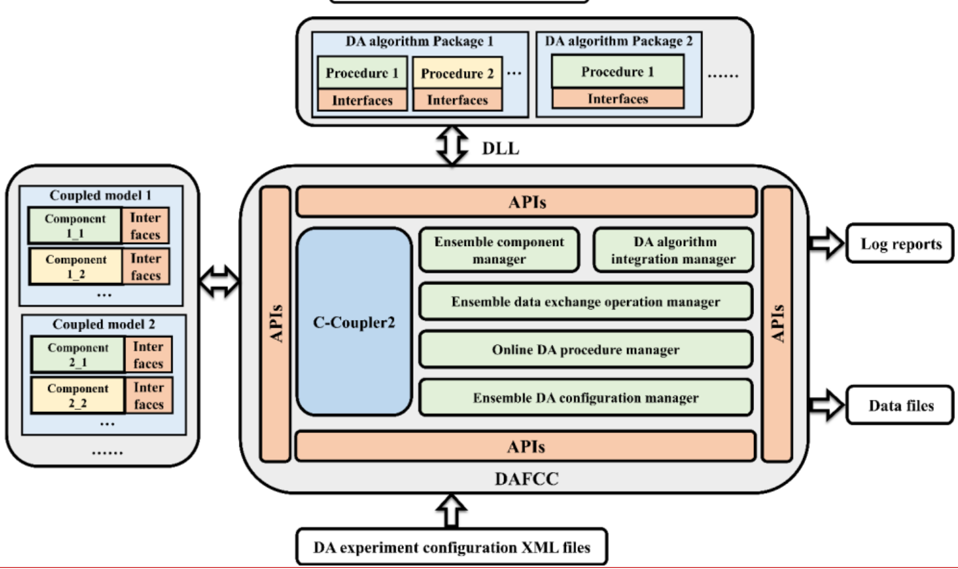
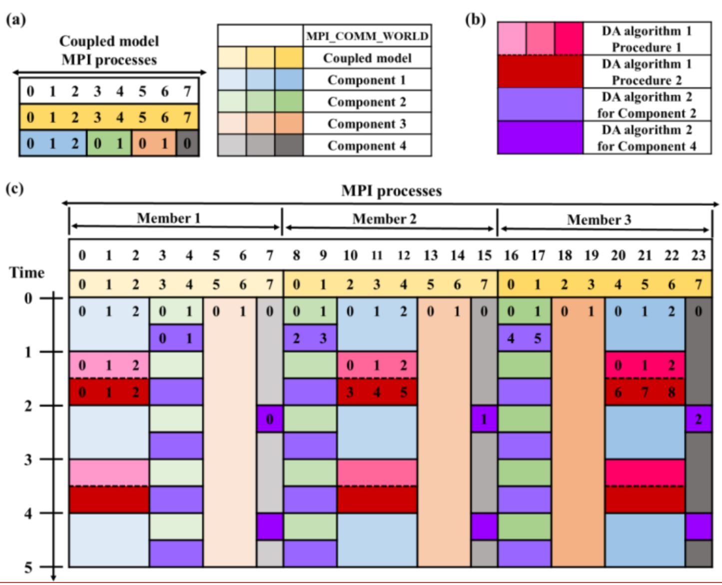




**Figure 1. Architecture of DAFCC1.**


(a)
Coupled model
MPI processes
MPI_COMM_WORLD
Coupled model
Component 1
Component 2
Component 3
Component 4
(b)
DA algorithm 1
Procedure 1
DA algorithm 1
Procedure 2
DA algorithm 2
for Component 2
DA algorithm 2
for Component 4
(c)
MPI processes
Member 1
Member 2
Member 3
Time

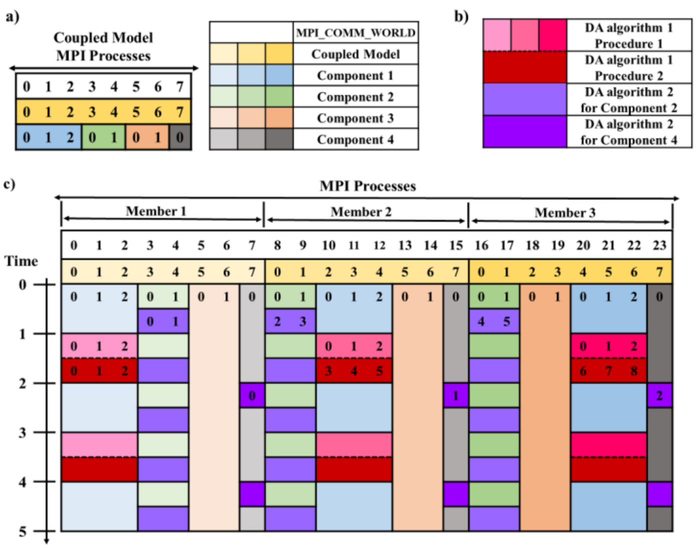
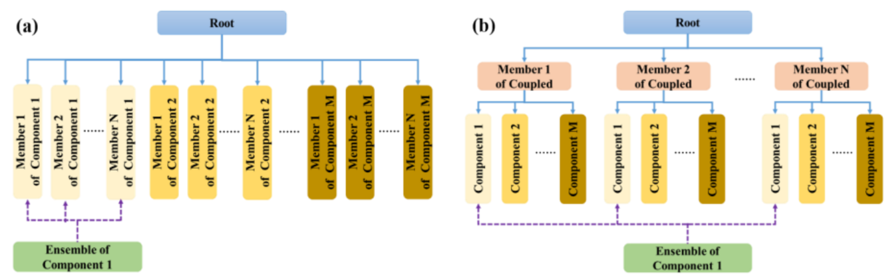



**Figure 2. Example of running a DAFCC1-based weakly coupled ensemble DA system with three ensemble members. (a) Each ensemble member of the coupled model (yellow series) uses 8 MPI processes, where component 1 (blue series) uses three MPI processes, component 2 (green series) uses two MPI processes, component 3 (orange series) uses two MPI processes, and component 4 (grey series) uses one MPI process. (b) DA algorithm 1 and two instances of DA algorithm 2 (purple series) are used in this DA system, where DA algorithm 1 includes procedure 1 (pink series) and procedure 2 (red). (c) Execution of the DA system: the process layout of ensemble members of component models, the process layout of DA algorithms, and the alternative execution of a DA algorithm and the corresponding component model. Each number in the colored box in (a) and (c) indicates the process ID in the corresponding local communicator of a member of the coupled model, a member of a component model, or all members of a component model.**

~~mpirun –np N1_1 Comp1 namelist CCPL_ensemble_3_1 : –np N1_2 Comp2 namelist CCPL_ensemble_3_1 : –np N2_2 Comp2 namelist CCPL_ensemble_3_2 : –np N2_1 Comp1 namelist CCPL_ensemble_3_2 : –np N3_1 Comp1 namelist CCPL_ensemble_3_3 : –np N3_2 Comp2 namelist CCPL_ensemble_3_3~~

```
mpirun –np N1_1 Comp1 namelist CCPL_ensemble_3_1 : –np N1_2 Comp2 namelist
CCPL_ensemble_3_1 : –np N2_2 Comp2 namelist CCPL_ensemble_3_2 : –np N2_1 Comp1
namelist CCPL_ensemble_3_2 : –np N3_1 Comp1 namelist CCPL_ensemble_3_3 : –np N3_2
Comp2 namelist CCPL_ensemble_3_3
```

**Figure 3. Example of the command for submitting an MPI run of three ensemble members of a coupled model that consists of Comp1 and Comp2. Comp1 can be before Comp2 at the second ensemble member, and the process numbers N1_1, N2_1, and N3_1 of Comp1 at different ensemble members can be different~~, because C-Coupler2.0 does not introduce any preconditions on the process layout of ensemble members of models~~.**



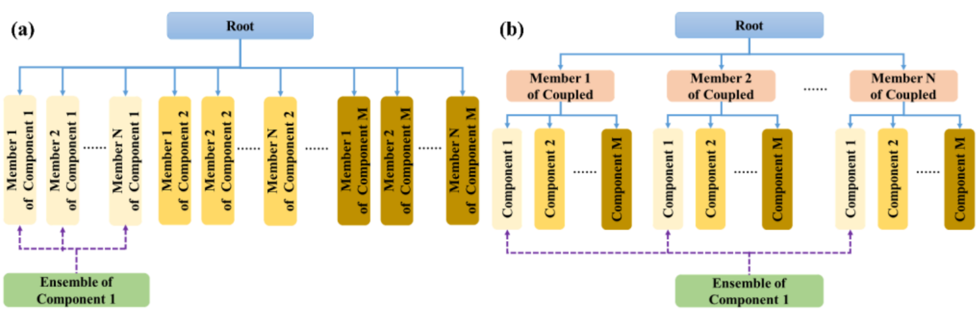




**Figure 4. Two examples of the organization of N ensemble members of a coupled model consisting of M component models. (a) Single-level organizational architecture of all ensemble members of the component models in the coupled model. (b) Two-hierarchical-levels organizational architecture. All ensemble members of the coupled model are organized as the first level with all component models from each ensemble member of the coupled model at the second level. An ensemble that covers all ensemble members of component model 1 is generated as an example for using the DA algorithm in ensemble component manager.**

(a)
```
Subroutine DA_CCPL_INIT
    Obtain model information and grids from C-Coupler2
    Initialize grid and parallel decomposition
    Allocate memory space of variables
    Initialize other operations
    Register the parallel decompositions, required grids
    and field instances to C-Coupler2
    Declare field instances as implicit input or output arguments
End subroutine DA_CCPL_INIT
```

(b)
```
Subroutine DA_CCPL_RUN
    DA analysis
End subroutine DA_CCPL_RUN
```

(c)
```
Subroutine DA_CCPL_FIN
    DA finalize
End subroutine DA_CCPL_FIN
```

a)
```
Subroutine DA_CCPL_INIT
    Obtain model information and grids from C-Coupler2
    Initialize grid and parallel decomposition
    Allocate memory spaces of variables
    Initialize other operations
    Register the parallel decompositions, required grids
    and field instances to C-Coupler2
    Declare field instances as implicit input or output arguments
End subroutine DA_CCPL_INIT
```

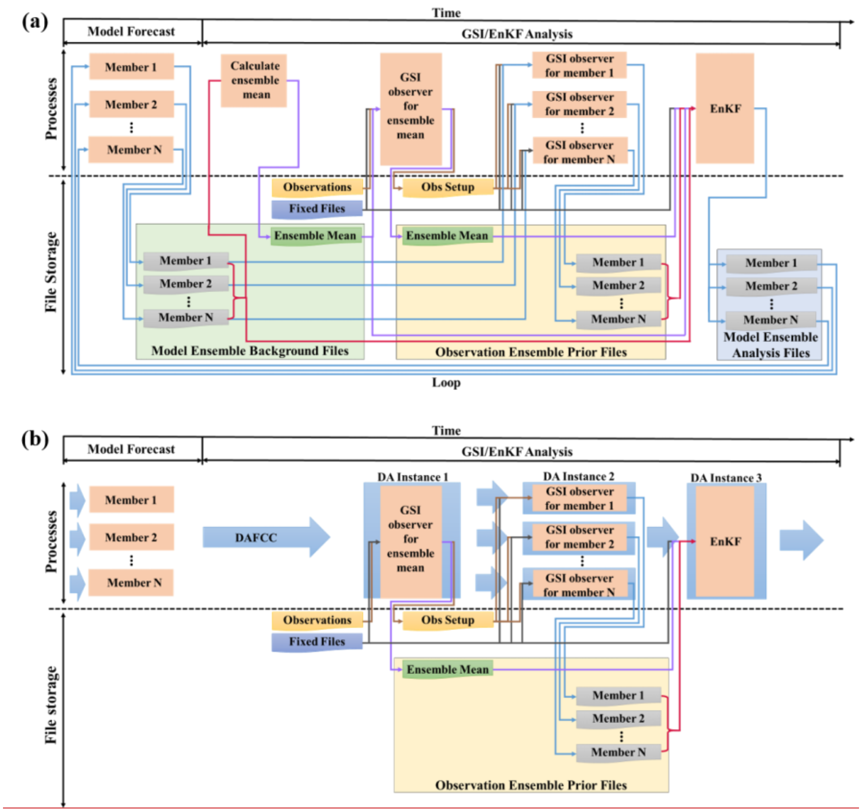
b)
```
Subroutine DA_CCPL_RUN
    DA analysis
End subroutine DA_CCPL_RUN
```

c)
```
Subroutine DA_CCPL_FIN
    DA finalize
End subroutine DA_CCPL_FIN
```

**Figure 5. Example of the driving subroutines in a DA algorithm. (a) Initialization driving subroutine. (b) Running driving subroutine. (c) Finalization driving subroutine. The name of the DA algorithm "DA" is used as the prefix of the three driving subroutines; different suffixes are used for distinction. Black font indicates original functionalities of the DA algorithm, while red font indicates additional operations to perform online data exchanges between the model and DA algorithm.**

```xml
<root>
<da_instance name="DA_algorithm1_procedure1" status="on">
    <external_procedures status="on" procedures_name="algorithm1_procedure1" dll_name="lib_da1_p1.so"/>
    <periodic_timer status="on" period_unit="seconds" period_count="21600" local_lag_count="0"/>
    <field_instances status="on" time_processing="inst" ensemble_operation="none"/>
    <processing_control status="on" >
        <working_directory status="off" path=""/>
        <config_scripts status="on">
            <pre_instance_script status="on" name="da1_p1_online_run.sh"/>
            <post_instance_script status="off" name=""/>
        </config_scripts>
    </processing_control>
</da_instance>
<da_instance name="DA_algorithm1_procedure2" status="on">
    <external_procedures status="on" procedures_name="algorithm1_procedure2" dll_name="lib_da1_p2.so"/>
    <periodic_timer status="on" period_unit="seconds" period_count="21600" local_lag_count="0"/>
    <field_instances status="on" time_processing="inst" ensemble_operation="aver">
        <field name="XLAT" time_processing="inst" ensemble_operation="mem_1"/>
        <field name="XLONG" time_processing="inst" ensemble_operation="mem_1"/>
    </field_instances>
    <processing_control status="on" >
        <working_directory status="on" path="./experiment/da1"/>
        <config_scripts status="on">
            <pre_instance_script status="off" name=""/>
            <post_instance_script status="on" name="da1_p2_online_run.sh"/>
        </config_scripts>
    </processing_control>
</da_instance>
<da_instance name="DA_algorithm2" status="on">
    <external_procedures status="on" procedures_name="algorithm2" dll_name="lib_da2.so"/>
    <periodic_timer status="on" period_unit="seconds" period_count="21600" local_lag_count="0"/>
    <field_instances status="on" time_processing="inst" ensemble_operation="gather"/>
    <processing_control status="on" >
        <working_directory status="on" path="./experiment/da2"/>
        <config_scripts status="on">
            <pre_instance_script status="on" name="./scripts/da2_pre_online_run.sh"/>
            <post_instance_script status="on" name="./scripts/da2_post_online_run.sh"/>
        </config_scripts>
    </processing_control>
</da_instance>
</root>
```

**Figure 6. Example of the XML configuration for a DA experiment.**


(a)
Time
Model Forecast
GSI/EnKF Analysis
Processes
Member 1
Member 2
Member N
Calculate ensemble mean
GSI observer for ensemble mean
GSI observer for member 1
GSI observer for member 2
GSI observer for member N
EnKF
File Storage
Observations
Fixed Files
Ensemble Mean
Obs Setup
Model Ensemble Background Files
Observation Ensemble Prior Files
Model Ensemble Analysis Files
Loop
(b)
DAFCC
DA Instance 1
DA Instance 2
DA Instance 3
File storage

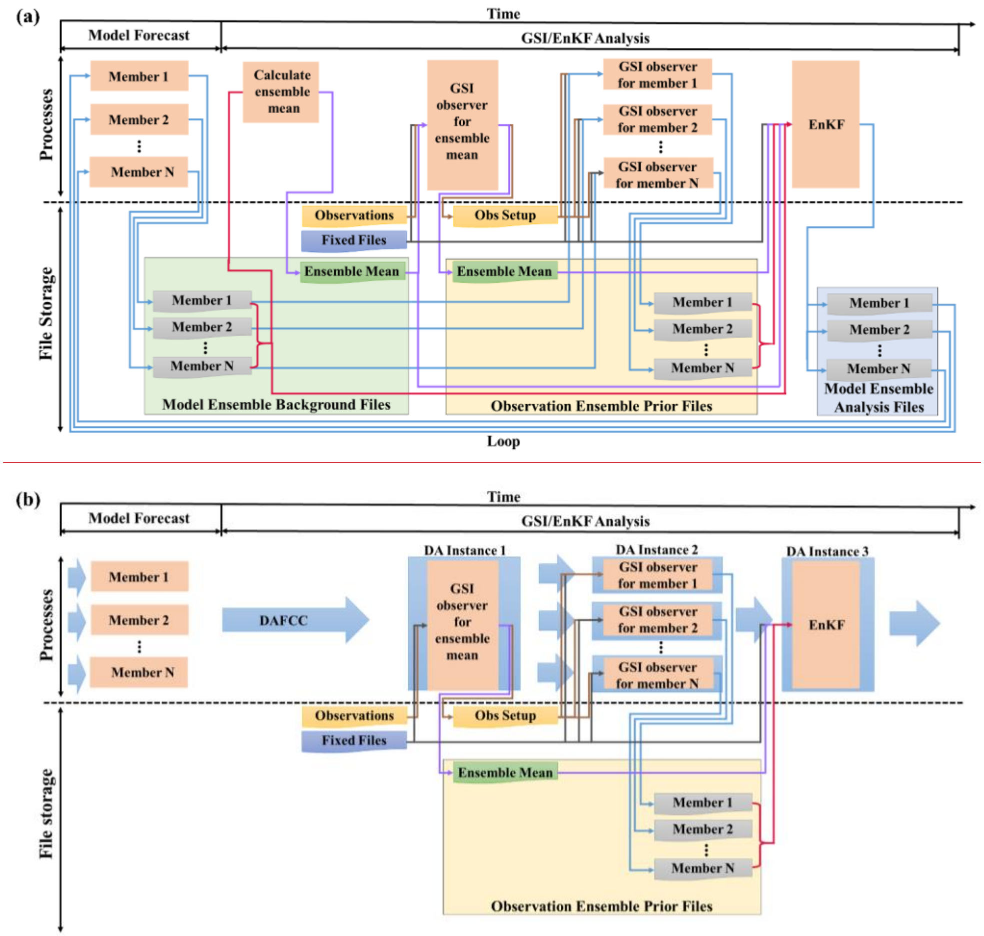
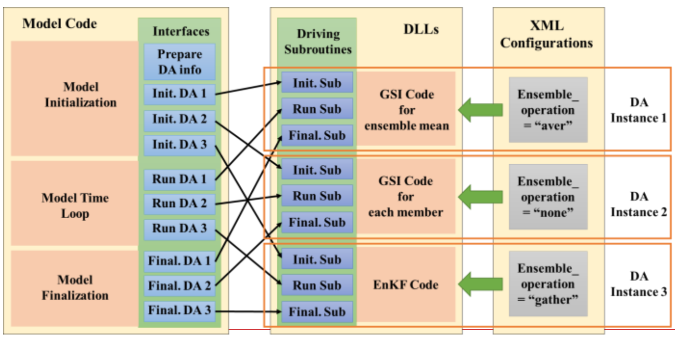



**Figure 7. Running processes and data scheduling for (a) original GSI/EnKF used as a pure ensemble DA system, and (b) modified GSI/EnKF based on DAFCC1. Orange rectangles in the Processes panel indicate different running processes, while thick blue arrows mark data scheduling based on DAFCC1. Rectangles of various colors with a curved lower edge in the File Storage panel indicate different files, while arrows of different colors indicate the scheduling of corresponding files.**



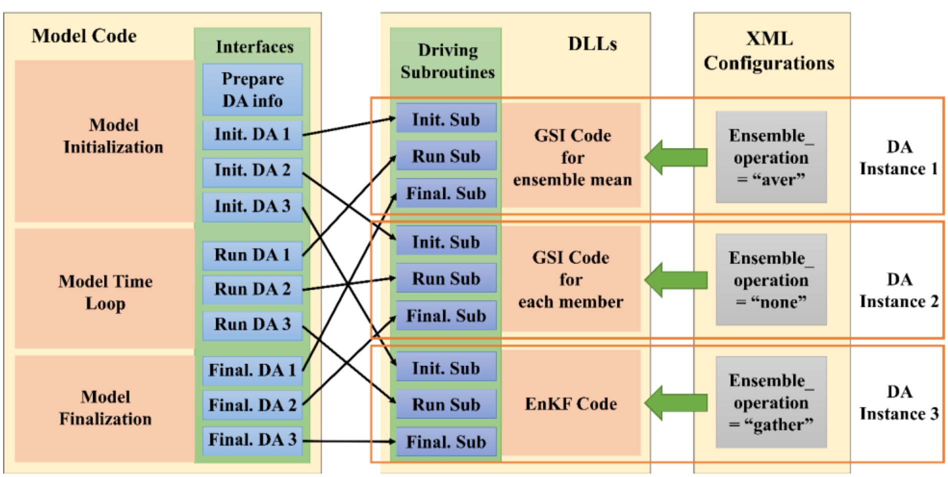
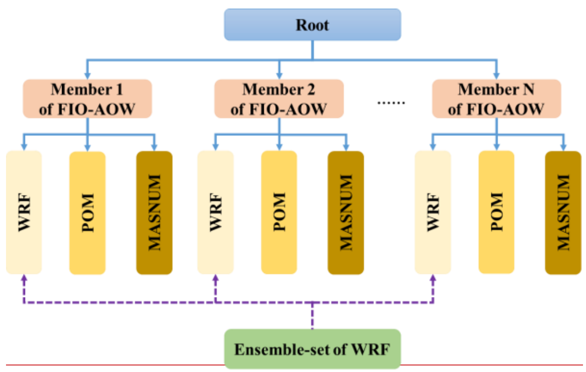




**Figure 8. Modifications of model code and the invoking of relationships to the DA algorithm in the ~~sample~~example ensemble DA system.**



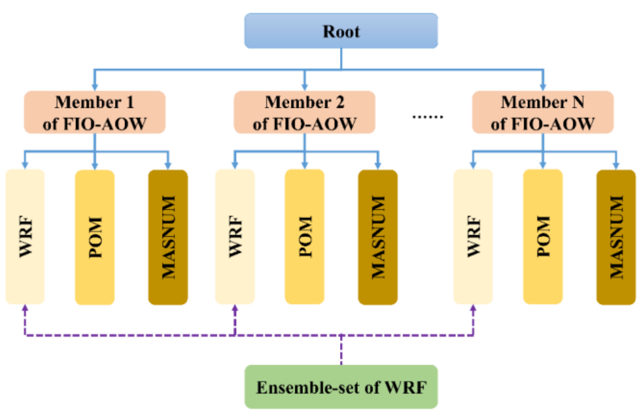
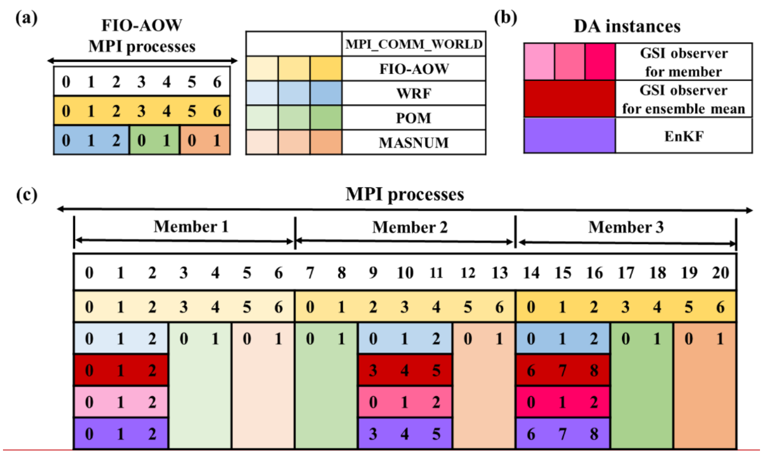




**Figure 9. Two-hierarchical-level organizational architecture for N ensemble members of FIO-AOW consisting of WRF, POM, and MASNUM. All ensemble members of FIO-AOW are organized as the first level with all component models in each ensemble member at the second level. An ensemble-set that covers all ensemble members of component model WRF is generated by the ensemble component manager.**



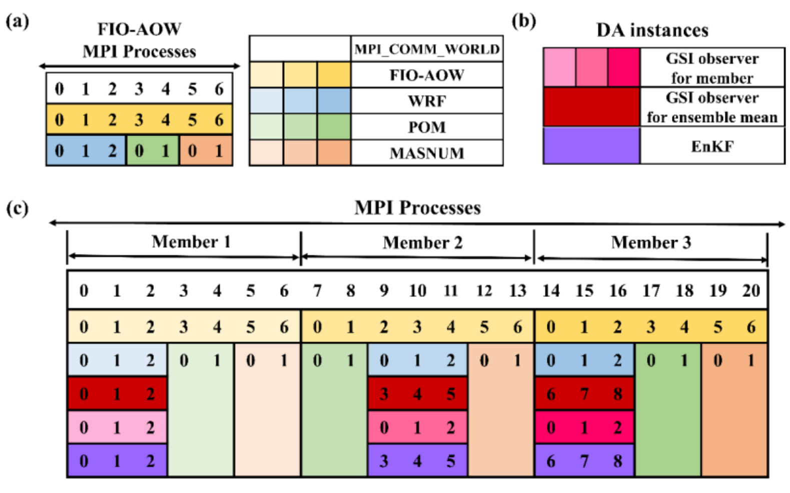
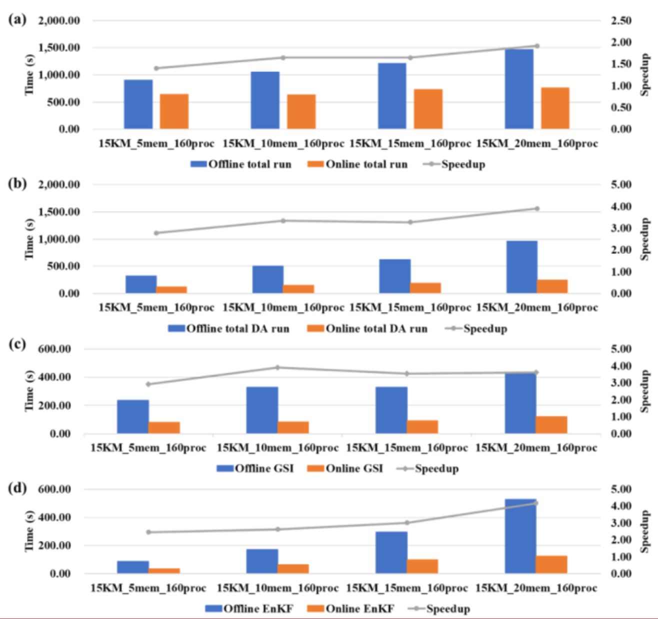
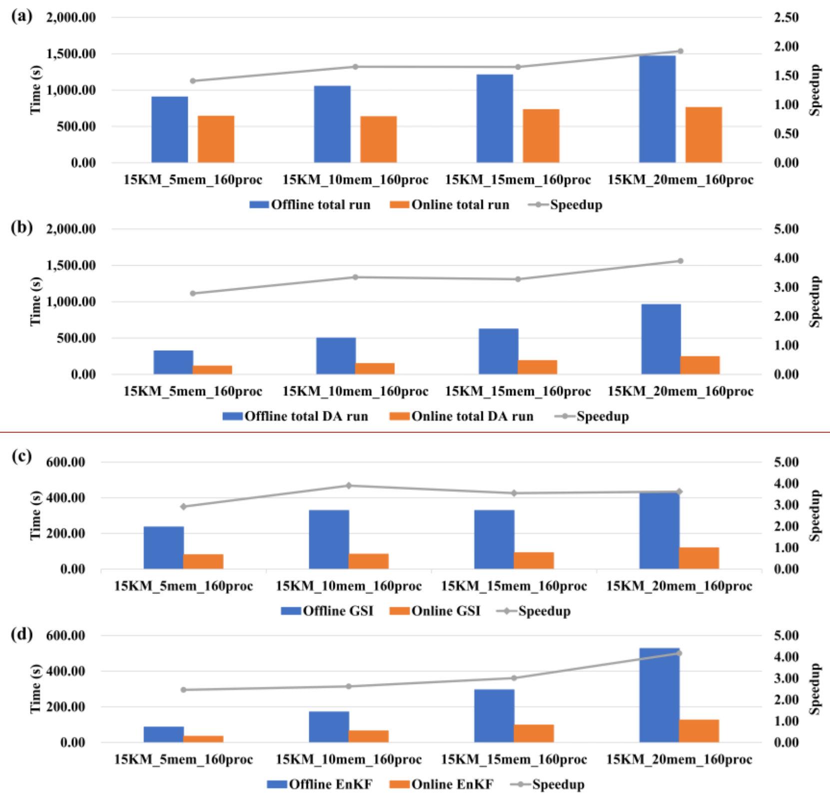




**Figure 10. Example of the process layout of the ~~sample~~example ensemble DA system FIO-AOW. (a) Each ensemble member of FIO-AOW (yellow series) uses 7 MPI processes, where WRF (blue series) uses 3 MPI processes, POM (green series) uses 2 MPI processes, and MASNUM (orange series) uses 2 MPI processes. (b) Two DA algorithm instances of GSI are adopted for each member (pink series) and ensemble mean (red) respectively following another DA algorithm instance of EnKF in this DA system. (c) Process layout of the DA system: the process layout of ensemble members of component models and the process layout of DA algorithms. Each**

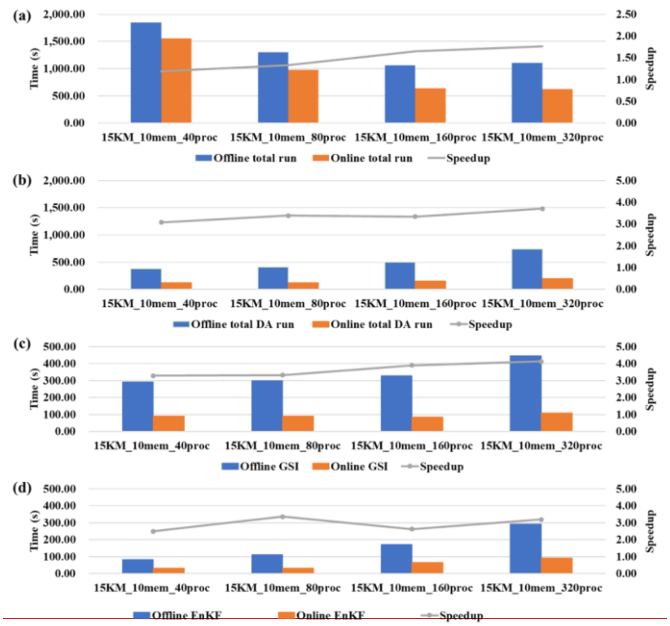
number in the colored boxes in (a) and (c) indicates the process ID in the corresponding local communicator of a member of the coupled model, a member of a component model, or all members of a component model.**






845

**Figure 11. Execution time (colored bars) corresponding to the online and offline WRF-GSI/EnKF and the corresponding speedup (gray line, ratio of offline execution time to online execution time) from experiment set 1 in Table 2. (a) Total run (including model run and DA algorithms run). (b) DA algorithms (including GSI and EnKF) run. (c) GSI run. (d) EnKF run.**


(a)
Time (s)
Speedup
15KM_10mem_40proc
15KM_10mem_80proc
15KM_10mem_160proc
15KM_10mem_320proc
Offline total run
Online total run
Speedup
(b)
Offline total DA run
Online total DA run
Speedup
(c)
Offline GSI
Online GSI
Speedup
(d)
Offline EnKF
Online EnKF
Speedup

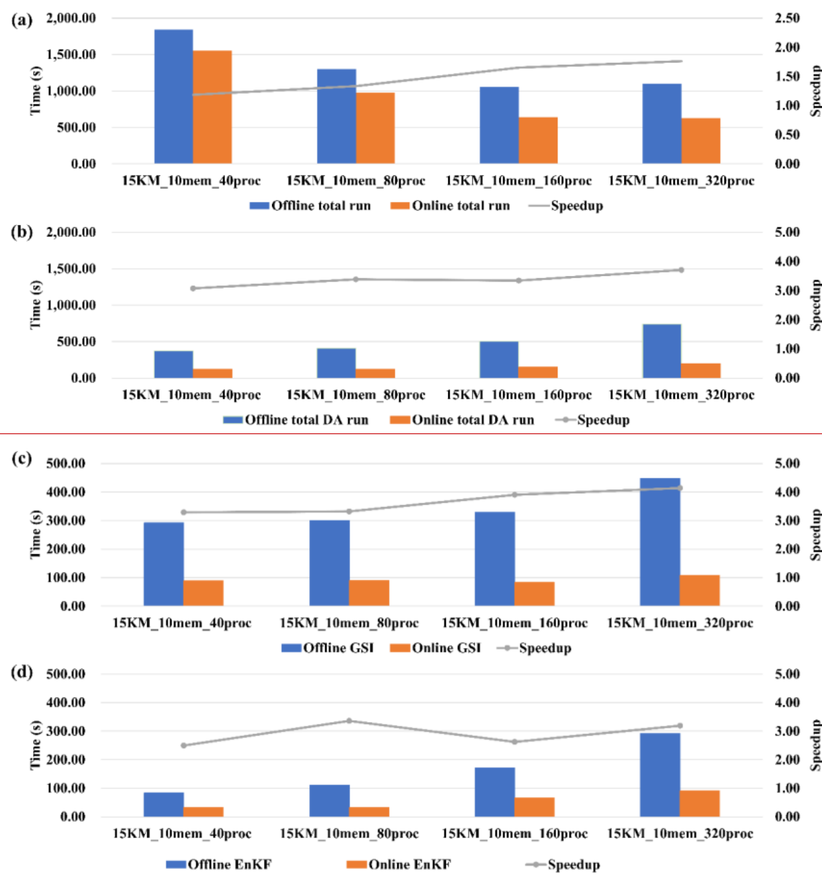
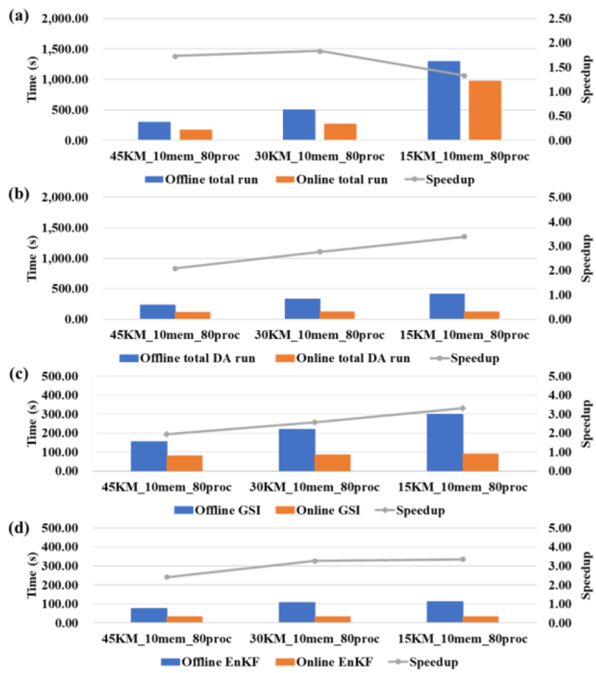



**Figure 12. As in Fig. 11, but from experiment set 2 in Table 2.**




**Formatted:** English (United States)

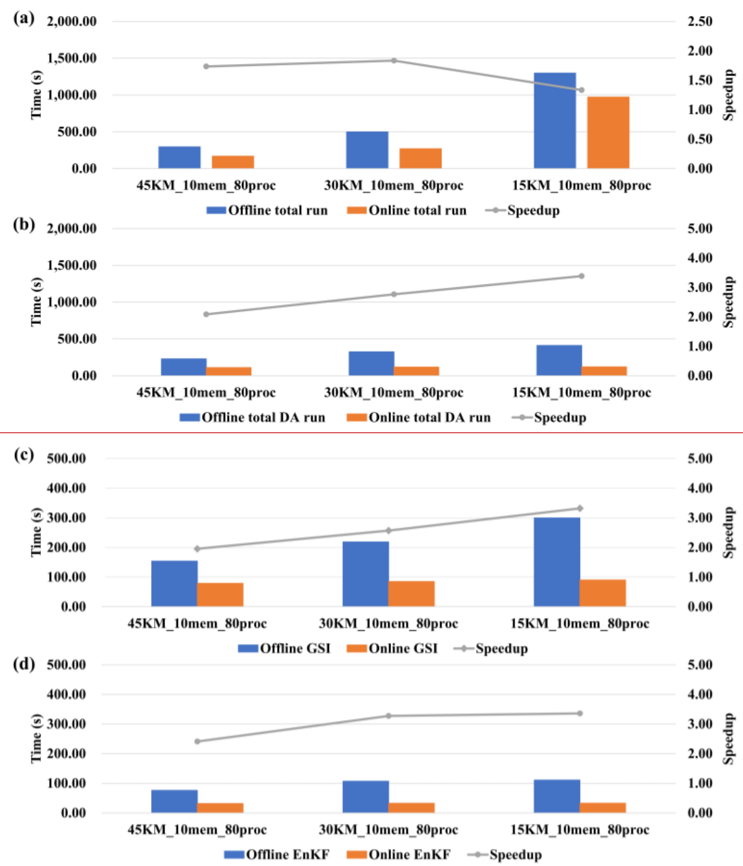
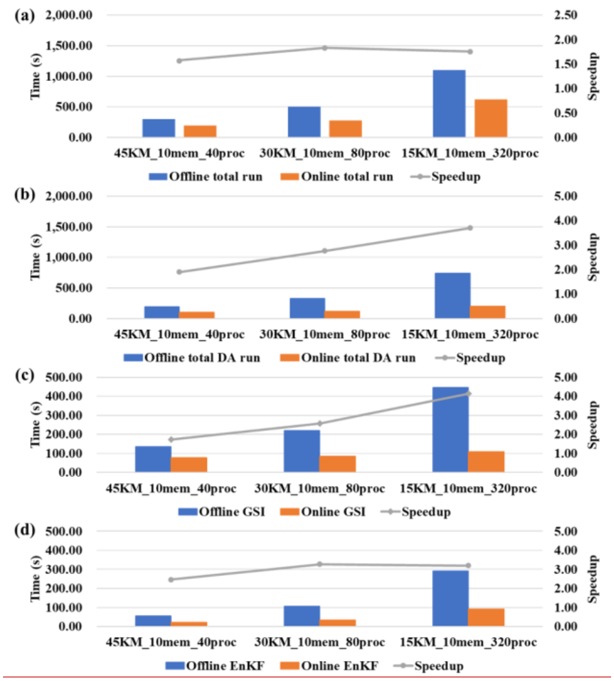



**Figure 13. As in Fig. 11, but from experiment set 3 in Table 2.**


(a)
Time (s)
Speedup
45KM_10mem_40proc
30KM_10mem_80proc
15KM_10mem_320proc
Offline total run
Online total run
Speedup
(b)
Offline total DA run
Online total DA run
Speedup
(c)
Offline GSI
Online GSI
Speedup
(d)
Offline EnKF
Online EnKF
Speedup

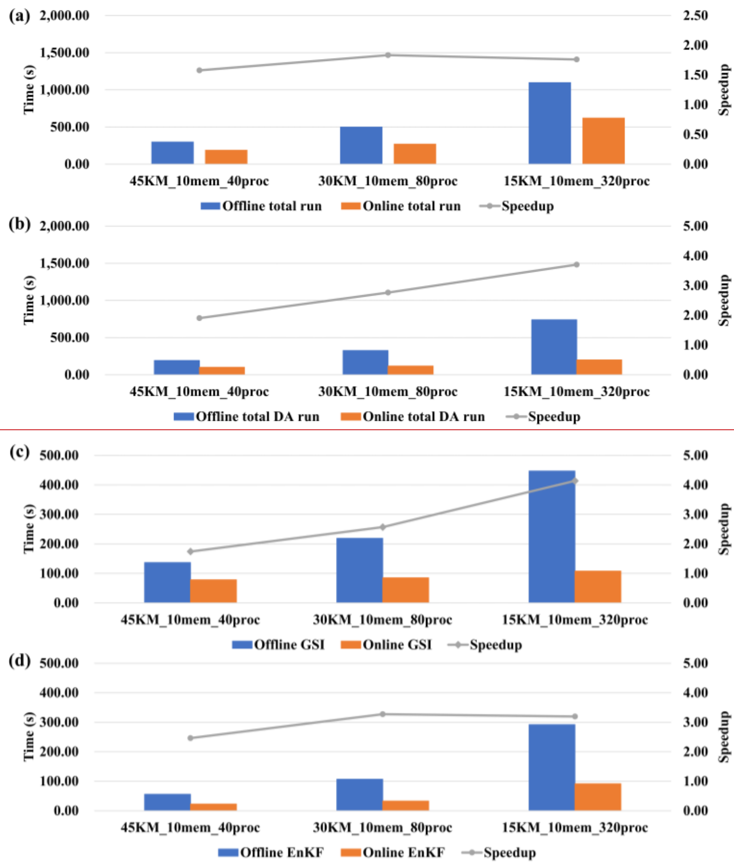
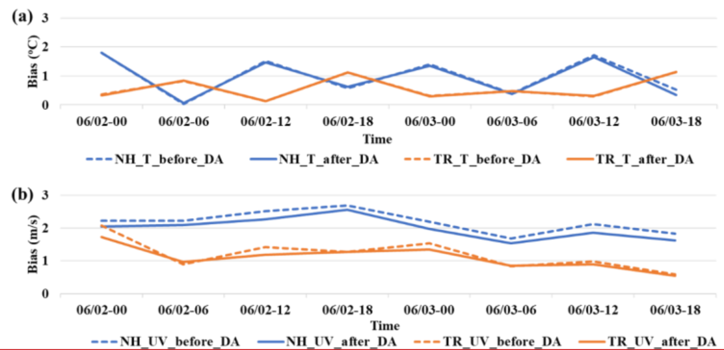



**Figure 14. As in Fig. 11, but from experiment set 4 in Table 2.**

865



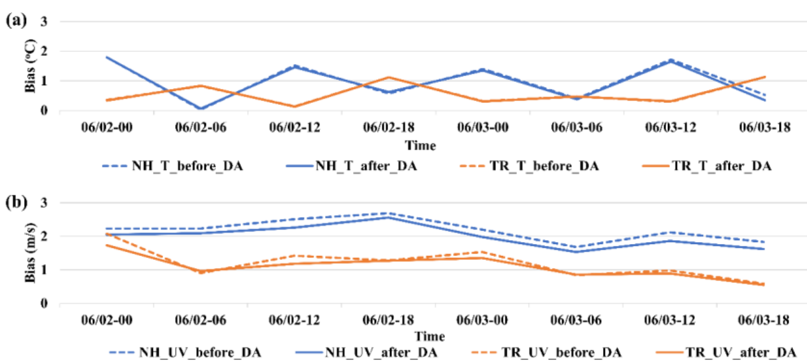




**Figure 15. Total bias of assimilated variables relative to corresponding observations before and after DA for (a) T and (b) UV at each DA time from the EnKF standard output file. The dotted lines indicate the bias of assimilated variables before DA and the solid lines indicate the bias of assimilated variables after DA. Blue lines are the bias in the ~~tropics (25° S–~~area of 0°–25° N~~)~~,, and orange lines are the bias in the ~~Northern Hemisphere~~ area ~~(~~of 25°–~~90~~50° N~~)~~..**

**Table 1. Horizontal resolutions and time steps of WRF.**

| Horizontal Resolution | Total Horizontal Grid Points | Time Step |
|---|---|---|
| 45 km | 160×120 | 180 s |
| 30 km | 240×180 | 120 s |
| 15 km | 480×360 | 60 s |

**Table 2. Setup of four experiment sets in terms of horizontal resolution, number of ensemble members and number of processes**

| Experiment set | Horizontal resolution | Number of ensemble members | Processes for each ensemble member | Label marks |
|---|---|---|---|---|
| **Set 1** | 15 km | 5 | 160 | 15KM_5mem_160proc |
| | | 10 | | 15KM_10mem_160proc |
| | | 15 | | 15KM_15mem_160proc |
| | | 20 | | 15KM_20mem_160proc |
| **Set 2** | 15 km | 10 | 40 | 15KM_10mem_40proc |
| | | | 80 | 15KM_10mem_80proc |
| | | | 160 | 15KM_10mem_160proc |
| | | | 320 | 15KM_10mem_320proc |
| **Set 3** | 45 km | 10 | 80 | 45KM_10mem_80proc |
| | 30 km | | | 30KM_10mem_80proc |
| | 15 km | | | 15KM_10mem_80proc |
| **Set 4** | 45 km | 10 | 40 | 45KM_10mem_40proc |
| | 30 km | | 80 | 30KM_10mem_80proc |
| | 15 km | | 320 | 15KM_10mem_320proc |

**Table 3. I/O access statistics corresponding to WRF-GSI/EnKF**

| Horizontal resolution | Number of ensemble members | Number of observation prior files | Total I/O accesses to observation priors | Number of model ensemble background & analysis files | Total I/O accesses to model ensemble background & analysis files |
|---|---|---|---|---|---|
| 15 km | 5 | 12 | 0.11 GB | 324 | 129.13 GB |
| 15 km | 10 | 22 | 0.21 GB | 624 | 251.30 GB |
| 15 km | 15 | 32 | 0.30 GB | 924 | 373.48 GB |
| 15 km | 20 | 42 | 0.39 GB | 1224 | 495.65 GB |
| 30 km | 10 | 22 | 0.18 GB | 624 | 62.86 GB |
| 45 km | 10 | 22 | 0.17 GB | 624 | 27.96 GB |