The Parallel Data Assimilation Framework PDAF – a flexible software framework for ensemble data assimilation

Alfred Wegener Institute for Polar and Marine Research

Lars Nerger, Wolfgang Hiller, and Jens Schröter

Alfred Wegener Institute for Polar and Marine Research, Bremerhaven, Germany

Contact: Lars.Nerger@awi.de · http://www.awi.de

Introduction

Ensemble filter algorithms can be implemented in a generic way such that they can be applied with various models with only a minimum amount of re-coding. This is possible because ensemble filters can operate on abstract state vectors and require only limited information about the numerical model and the observations.

To build an assimilation system, the analysis step of a filter algorithm needs to be connected to the numerical model. Also, ensemble integrations have to be enabled. The Parallel Data Assimilation Framework PDAF has been developed to provide these features.

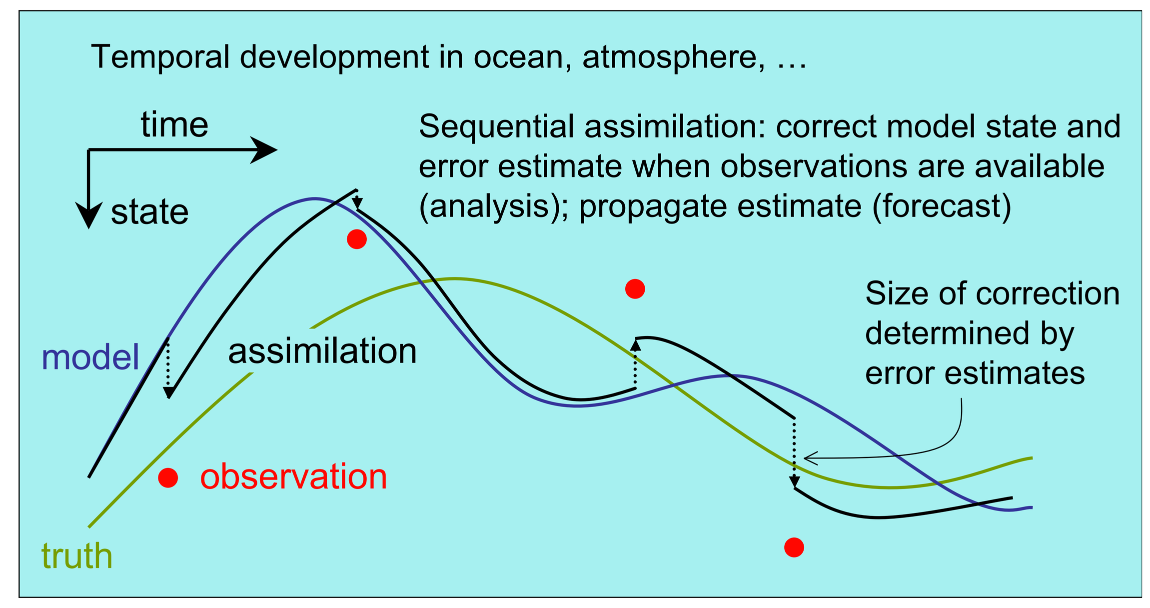
As the computational cost of ensemble data assimilation is a multiple of that of a pure forward model, the framework and the filter algorithms are parallelized and support parallelized models. Thus, data assimilation with high-dimensional numerical models is feasible.

Sequential Data Assimilation

PDAF is configured for sequential data assimilation with ensemble-based filters. A selection of important filter algorithms is fully implemented and optimized in PDAF including parallelization. Available are algorithms like

- EnKF – Ensemble Kalman Filter [1]
- LETKF – Local Ensemble Transform Kalman Filter [2]
- LSEIK – Local Singular Evolutive Interpolated Kalman filter [3]
- SEEK – Singular Evolutive Extended Kalman filter
- ESTKF – Error Subspace Transform Kalman Filter [4]

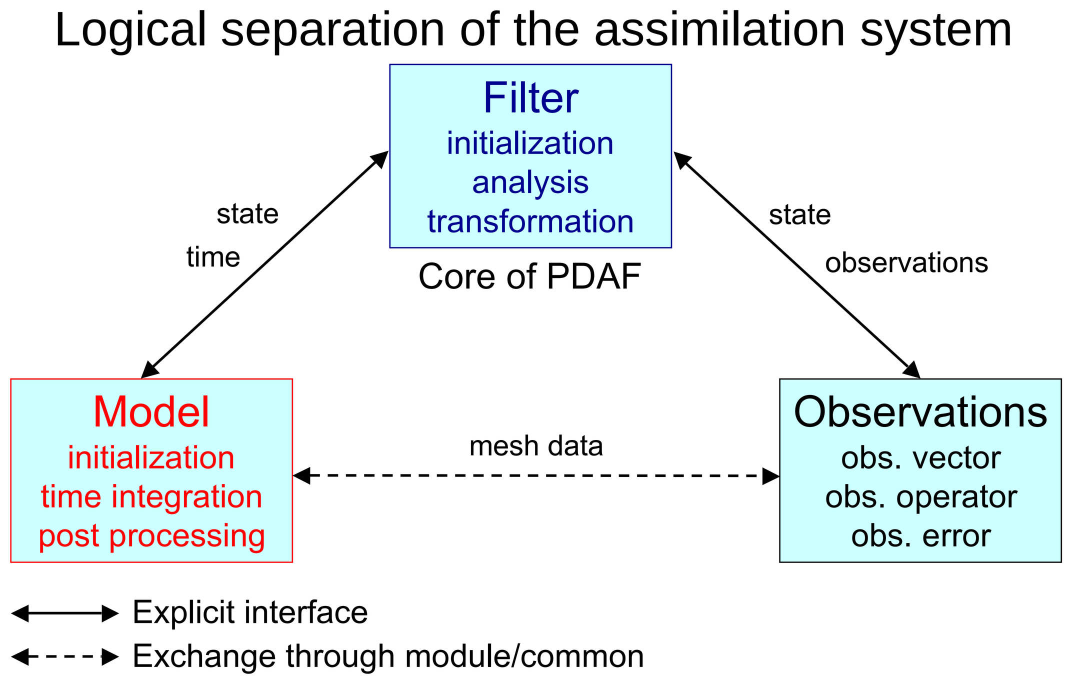
In addition to the filters, common fixes and tuning options like covariance inflation are implemented. Further, a selection of advanced localization options are available including observation localization.



Top: Principle of sequential data assimilation with a filter algorithm. The state estimate of the assimilation is given by the ensemble mean. The analysis estimate lies typically between the forecast estimate and the observation, hence closer to the true state.

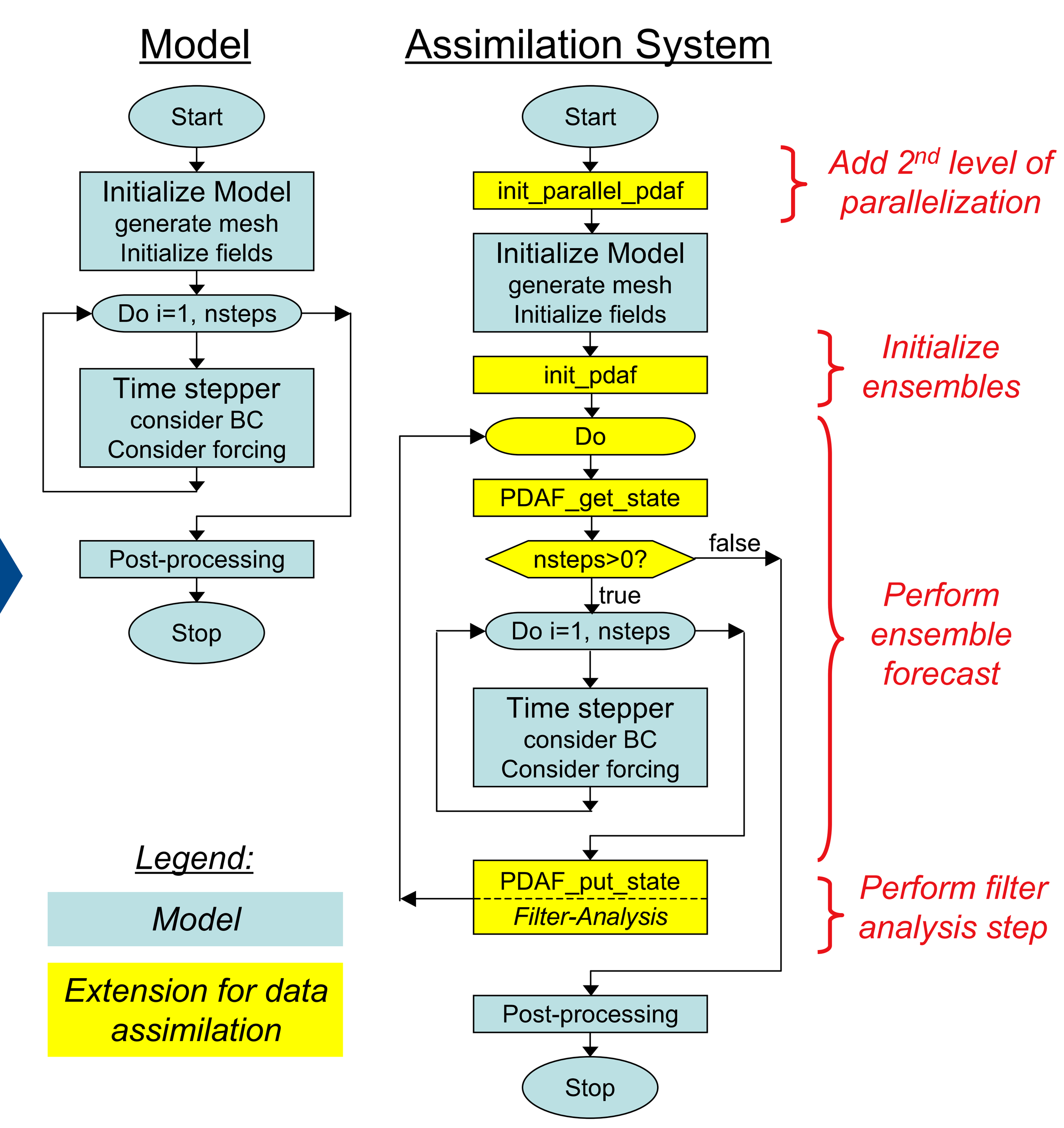
PDAF's Implementation Concept

Logical separation of the assimilation system

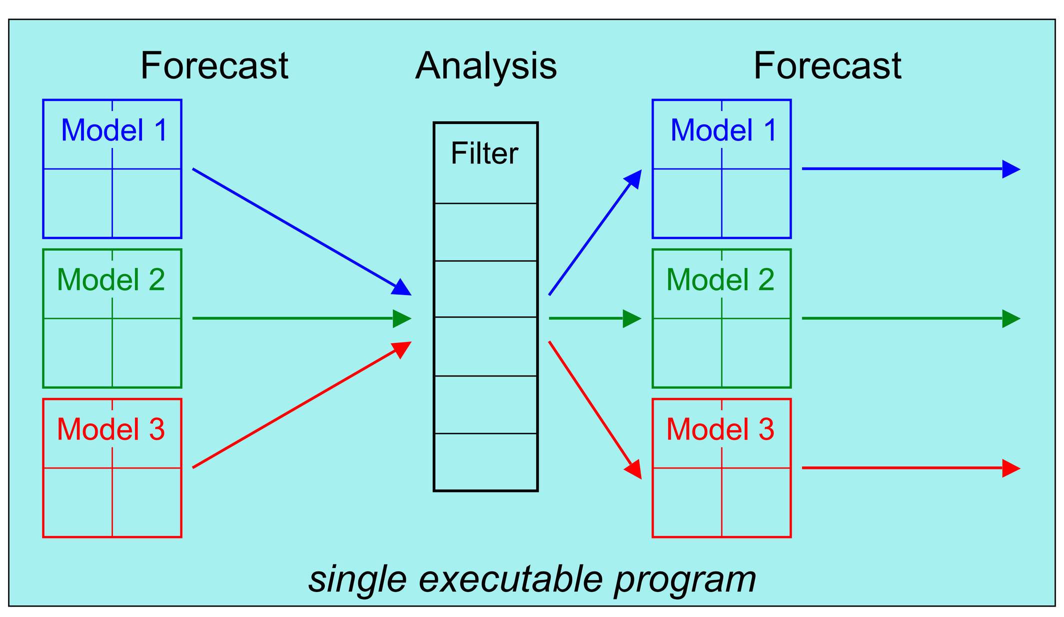


Left: PDAF is based on a consistent logical separation of the components of the data assimilation system: model, filter algorithm, and observations. The filter algorithms are part of PDAF's core, while the model routines and routines to handle observations are provided by the user. A standard interface for all filter algorithms connects the three components. All user-supplied routines exist in the context of the model and can be implemented like model routines.

Right: The assimilation system is implemented with PDAF [5,6] by extending the model source code. Four calls to subroutines are added. Further, an external loop enclosing the time stepping part of the model is required to perform ensemble integrations. In contrast to other frameworks, the model does not need to exist as a separate subroutine. The forecast phase is controlled by user-supplied routines that are called by PDAF_get_state. Implementations following this strategy have been performed for different models like FEOM, BSHcmod, MIPOM, NOBM, and ADCIRC.



2-level parallelization of the assimilation system



Left: PDAF not only provides fully implemented and parallelized ensemble filter algorithms, but also provides support for a 2-level parallelization for the assimilation system: 1. Each model task can be parallelized. 2. Several model tasks are executed concurrently. Thus, ensemble integrations can be done fully parallel. In addition, the filter analysis step uses parallelization. In the online-mode of PDAF, all components are included in a single program.

PDAF is coded in Fortran with MPI parallelization. It is available as free software. Further information and the source code of PDAF are available on the web site:

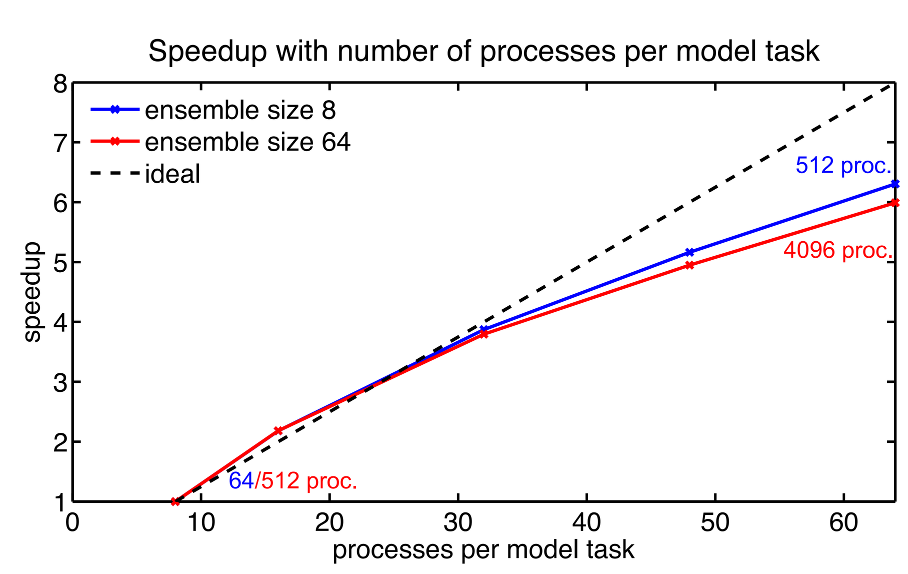
http://pdaf.awi.de

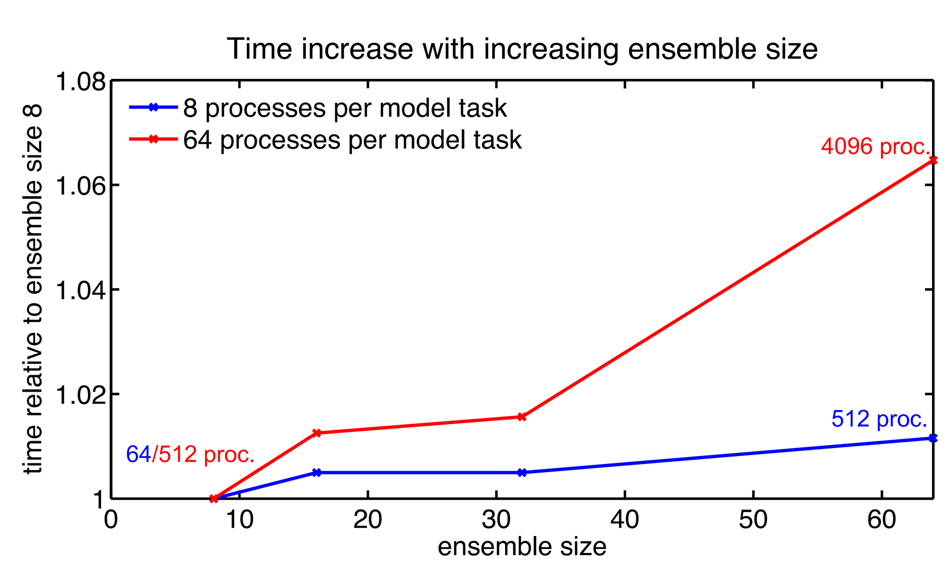
Parallel Performance

The parallel performance has been tested with an implementation of PDAF with the finite-element ocean model FEOM. About 94 to 99% of the computing time are used for the ensemble integrations.

Speedup is accessed with a constant ensemble size. Due to the parallel properties of the model, a speedup of 6 is obtained when the number of processors is increased by a factor of 8 (left panel).

The **scalability** of the assimilation system is visible when the number of processes per model task is kept constant. Increasing the ensemble size by a factor of eight results in a time increase between only 1% and 7% (right panel).





Summary

- The Parallel Data Assimilation Framework (PDAF) has been developed to simplify the implementation of data assimilation systems. It can be used to test assimilation methods, but is also applicable for realistic data assimilation applications.
- A very good scalability is provided through the complete parallelism of all parts of the assimilation system (ensemble integration, filter algorithms, and perhaps the model itself).
- Only minimal changes to the model source code are required when combining a model with PDAF in its online mode. An offline-mode is possible with separate programs for model and filtering. The offline mode avoids changes to the model code, but leads to a smaller computing performance.
- PDAF is currently used in several research projects and is in pre-operational use with BSHcmod. Also, it is distributed as free open-source software on the web site http://pdaf.awi.de.

References

[1] Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.* 99C: 10143

[2] Hunt, B.R., E.J. Kostelich, and I. Szunyogh (2007). Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D* 230: 112–126

[3] L. Nerger, S. Danilov, W. Hiller, and J. Schröter (2006). Using sea-level data to constrain a finite-element primitive-equation ocean model with a local SEIK filter. *Ocean Dynamics* 56: 634–649

[4] Nerger, L., T. Janjić, J. Schröter, J., and W. Hiller (2012). A unification of ensemble square root Kalman filters. *Mon. Wea. Rev.* In press. DOI:10.1175/MWR-D-11-00102.1

[5] Nerger, L., W. Hiller, and J. Schröter (2005). PDAF - The Parallel Data Assimilation Framework: Experiences with Kalman Filtering, in *Use of High Performance Computing in Meteorology - Proceedings of the 11th ECMWF Workshop* / Eds. W. Zwieflhofer, G. Mozdzynski. World Scientific, pp. 63–83

[6] Nerger, L. and W. Hiller (2012). Software for Ensemble-based Data Assimilation Systems – Implementation Strategies and Scalability. *Computers & Geosciences*. In press. DOI:10.1016/j.cageo.2012.03.026