Scalable Sequential Data Assimilation with the Parallel Data Assimilation Framework PDAF

Lars Nerger, Wolfgang Hiller, and Jens Schröter

Alfred Wegener Institute, Helmholtz Center for Polar and Marine Research, Bremerhaven, Germany
Contact: Lars.Nerger@awi.de · http://www.awi.de

Introduction

Data assimilation applications with high-dimensional numerical models exhibit extreme requirements on computational resources. Good scalability of the assimilation system is necessary to make these applications feasible. Sequential data assimilation methods based on ensemble forecasts, like ensemble-based Kalman filters, provide such good scalability. This parallelism has to be combined with the parallelization of both the numerical model and the data assimilation algorithm.

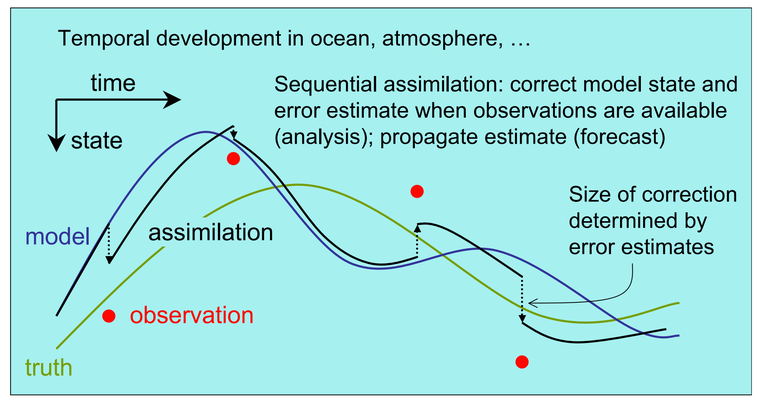
The Parallel Data Assimilation Framework PDAF has been developed to simplify the implementation of scalable data assimilation systems based on existing numerical models. PDAF is suitable for educational use with toy models but also for high-dimensional applications and operational use. PDAF is distributed as open source software.

Sequential Data Assimilation

PDAF is configured for sequential data assimilation with ensemble-based filters. A selection of filter and smoother algorithms is fully implemented and optimized in PDAF including parallelization, e.g.

- EnKF – Ensemble Kalman Filter [1]
- LESTKF – Local Error Subspace Transform Kalman Filter [2]
- LETKF – Local Ensemble Transform Kalman Filter [3]
- LSEIK – Local Singular Evolutive Interpolated Kalman filter [4]
- smoother extensions of the filters above

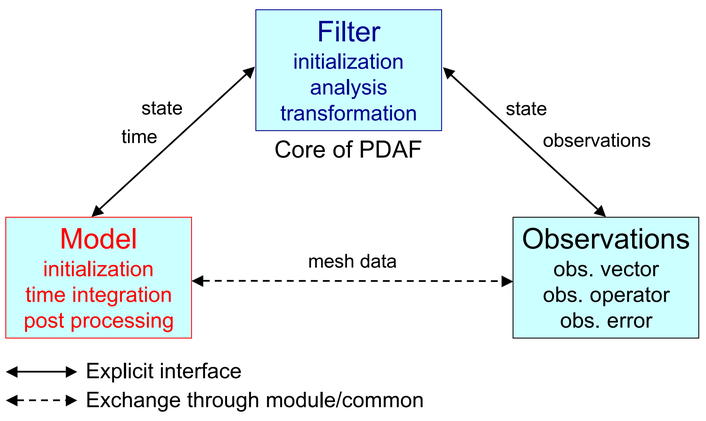
Common fixes and tuning options like covariance inflation are also implemented. Further, a selection of advanced localization options are available.



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.

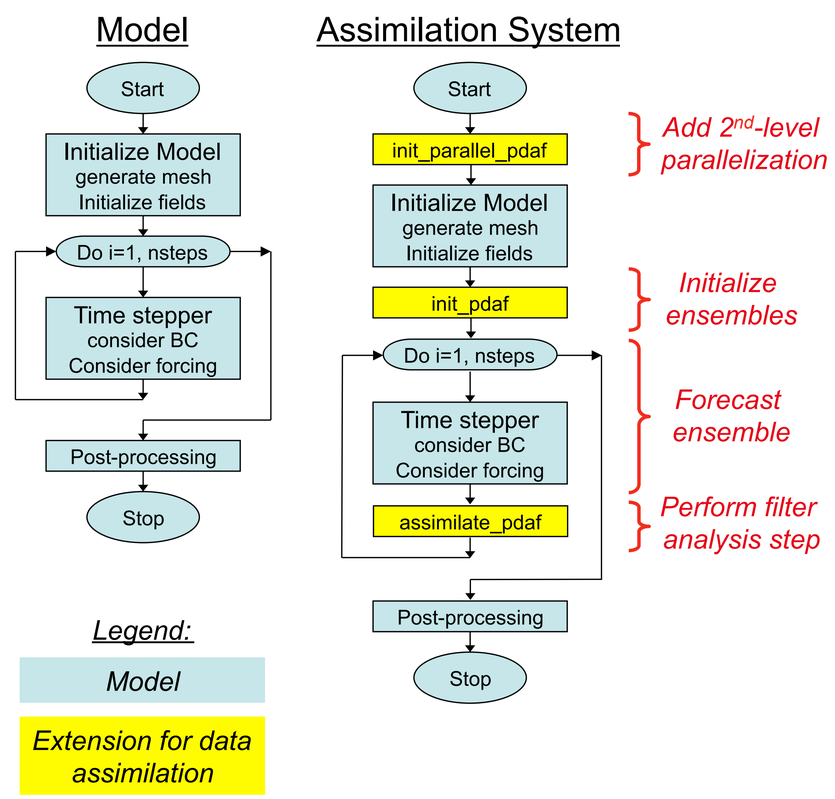
PDAF's Implementation Concept

Logical separation of the assimilation system

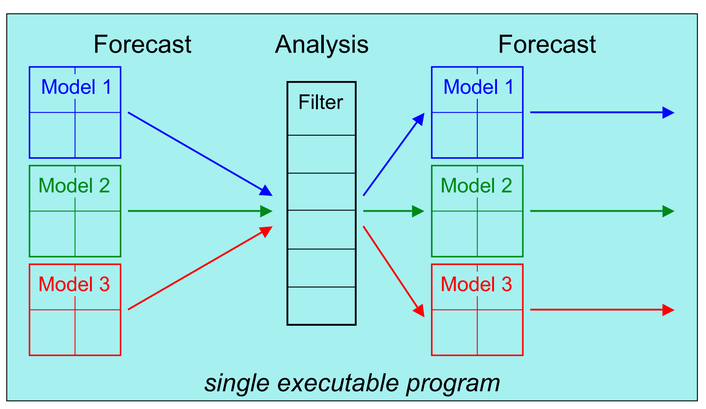


Left: PDAF uses a logical separation of the components of the data assimilation system: Model, filter algorithm, and observations. The filter algorithms are part of PDAF's core, while the model routines and routines to handle observations are provided by the user. A standard interface for all filter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

Right: The assimilation system is implemented with PDAF [5,6] by extending the model source code and utilizing parallelization. Three calls to subroutines are added. In contrast to other frameworks, the model does not need to exist as a separate subroutine. The ensemble forecast is controlled by user-supplied routines that are called through PDAF. Implementations using this online coupling have been performed for models like NEMO, FEOM, BSHcmod, MIPOM, NOBM, ADCIRC, and PARODY.



2-level parallelization of the assimilation system



Left: PDAF provides support for a 2-level parallelization for the assimilation system:

1. Each model task can be parallelized.
2. All model tasks are executed concurrently.

Thus, ensemble integrations can be done fully parallel. In addition, the filter analysis step uses parallelization. All components are combined 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

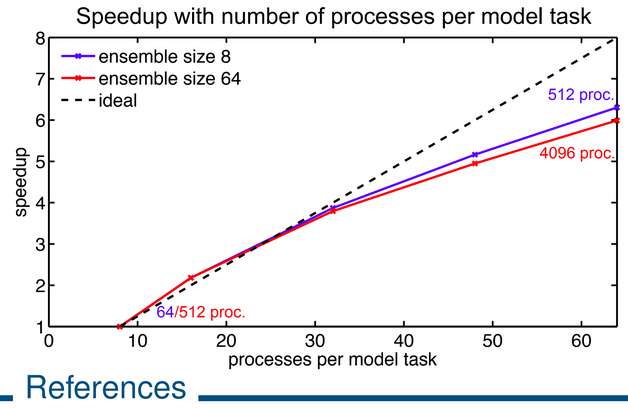
Parallel Performance

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.

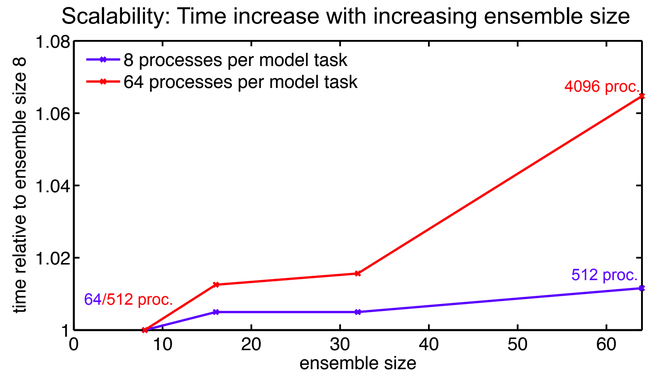
Speedup is accessed with a constant ensemble size. Due to the parallel properties of the model, a speedup of 6 is obtained when the number of processors is increased by a factor of 8 (left panel).

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).

Speedup with number of processes per model task



Scalability: Time increase with increasing ensemble size



Summary

- PDAF has been developed to simplify the implementation of data assimilation systems. It is aimed for large-scale data assimilation applications but can also be used to test or teach assimilation methods with small models.
- 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 also supported 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 with a variety of models. It is in pre-operational use for forecasting in the North Sea (see poster E-P10 by S. Losa et al.).

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] Nerger, L., T. Janjić, J. Schröter, J., and W. Hiller (2012). A unification of ensemble square root Kalman filters. *Mon. Wea. Rev.* 140, 2335–2345

[3] 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

[4] 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

[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.* 55, 110–118