PDAF Parallel Data Assimilation Framework

ALFRED-WEGENER-INSTITUT
HELMHOLTZ-ZENTRUM FÜR POLAR- UND MEERESFORSCHUNG

PDAF – Features and Recent Developments

Lars Nerger

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

Joint ECMWF/OceanPredict workshop
on Advances in Ocean Data Assimilation, May 17-20, 2021

HELMHOLTZ
SPITZENFORSCHUNG FÜR GROSSE HERAUSFORDERUNGEN

PDAF – Parallel Data Assimilation Framework

PDAF Parallel Data Assimilation Framework

A tool for data assimilation …

- program library for ensemble data assimilation
- provide support for parallel ensemble forecasts
- provide filters and smoothers - fully-implemented & parallelized
  (EnKF, LETKF, LESTKF, NETF, PF … easy to add more)
- easily useable with (probably) any numerical model
  (coupled to e.g. NEMO, MITgcm, FESOM/AWI-CM, ICON, SCHISM/ESMF)
- run from notebooks to supercomputers (Fortran, MPI & OpenMP)
- usable for real assimilation applications and to study assimilation methods
- ~500 registered users; community contributions

Open source:
Code, documentation, and tutorial available at

http://pdaf.awi.de

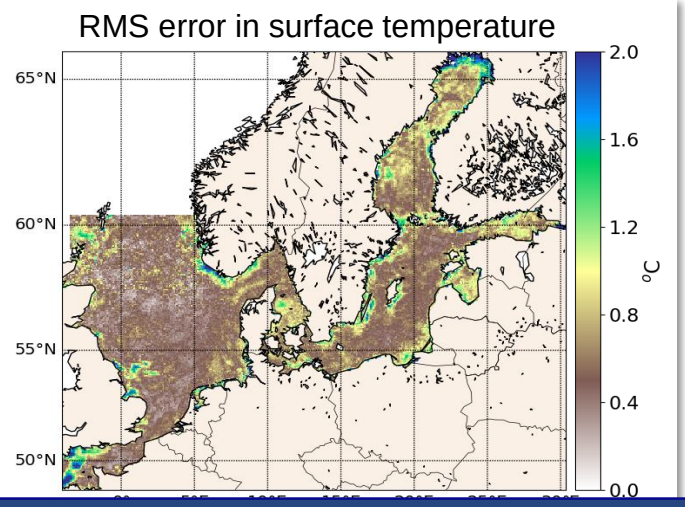
L. Nerger, W. Hiller, Computers & Geosciences 55 (2013) 110-118

AWI

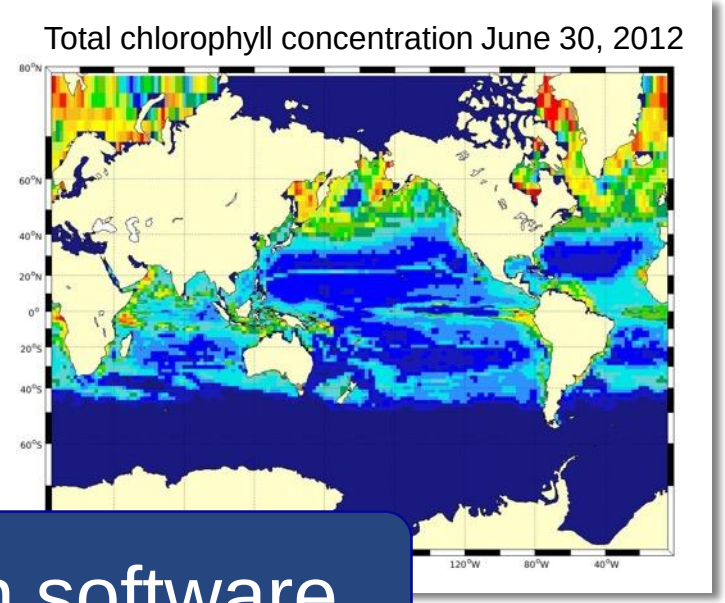
PDAF Application Examples

PDAF Parallel Data Assimilation Framework

HBM-ERGOM:
coupled physics/ biogeochemistry coastal assimilation
(Goodliff et al., 2019)

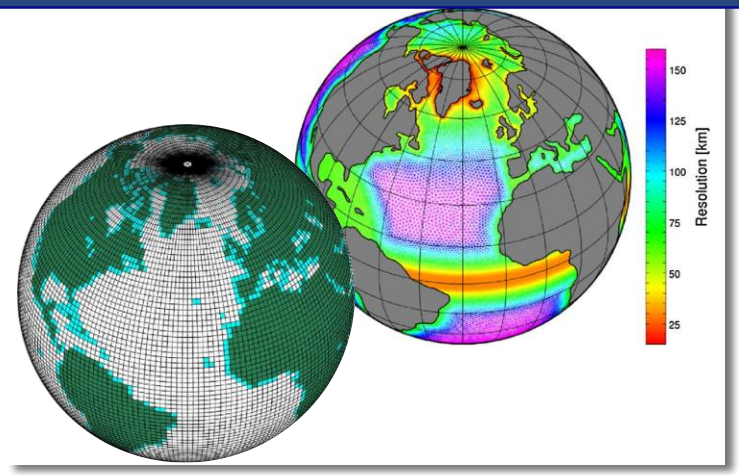


MITgcm-REcoM:
global ocean color assimilation into biogeochemical model
(Pradhan et al., 2019/20)



Different models – same assimilation software

AWI-CM:
coupled atmos.-ocean assimilation
(Tang et al., 2020
Mu et al., 2020
Nerger et al., 2020)



- MITgcm sea-ice assim (*operational*, NMEFC Beijing)
- CMEMS Baltic-MFC (*operational*, DMI/BSH/SMHI)
- NEMO (U Reading, P. J. van Leeuwen)
- SCHISM/ESMF (VIMS, J. Zhang)
- TerrSysMP-PDAF (hydrology, FZ Juelich, U Bonn)
- TIE-GCM (U Bonn, J. Kusche)
- VILMA (GFZ Potsdam)
- Parody geodynamo (IPGP Paris, A. Fournier)

PDAF: User-friendliness

Goal: Enable easy and fast setup of a DA system,
and allow for extension to fully featured system

Assumption: Users know their model

➜ let users implement DA system in model context

For users, model is not just a forward operator

➜ let users extend their model for data assimilation

operational centers might have other priorities – but the concept is still correct

Keep code simple for the user side:

- ➜ Define subroutine interfaces to DA code based on arrays
  (also simplifies interaction with languages like C/C++/Python)
- ➜ No object-oriented programming
  (most models don't use it; most model developers don't know it;
  many objects we would only have for observations – see later)
- ➜ Users directly implement case-specific routines
  (no indirect description (XML, YAML, ...) of e.g. observation layout)

PDAF: Design Considerations

PDAF Parallel Data Assimilation Framework

1. **Focus on ensemble methods**
2. **Efficiency:**
   - Direct (online/in-memory) coupling of model and data assimilation method (file-based offline coupling also supported)
   - complete parallelism in model, DA method, and ensemble integrations
3. **Ease of use:**
   - require just standard compilers and libraries, no containers, etc.
   - just add subroutine calls into model code when combining with PDAF
   - model time stepper not required to be a subroutine
   - model controls the assimilation program
   - case-specific routines implemented like model code
   - simple switching between different filters and data sets
   - Separation of concerns: model, DA methods, observations

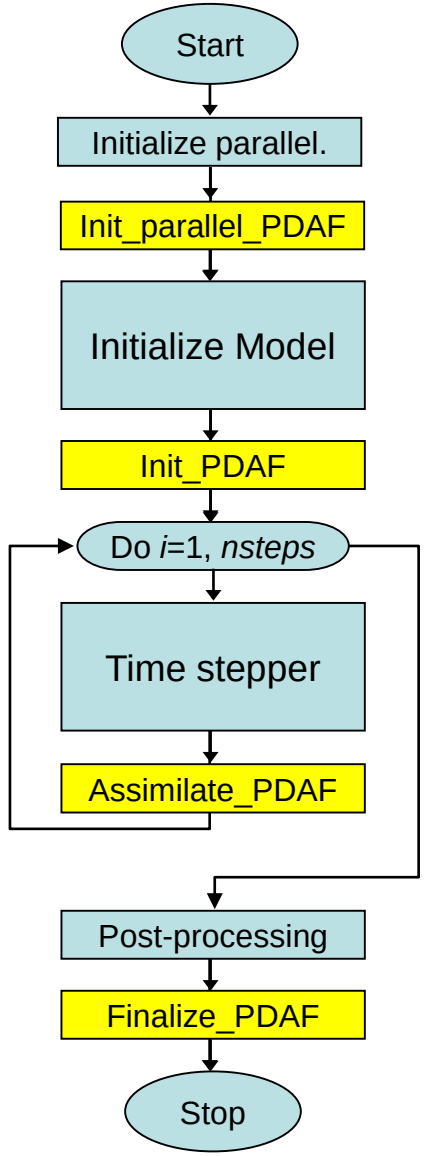
AWI

Online coupling - Augmenting a Model for Data Assimilation

PDAF Parallel Data Assimilation Framework

revised parallelization enables ensemble forecast

Data assimilation: run model with additional options



Model

Extension for data assimilation:

4 subroutine calls

plus:
Possible model-specific adaption

e.g. in NEMO or ECHAM: treat leap-frog time stepping

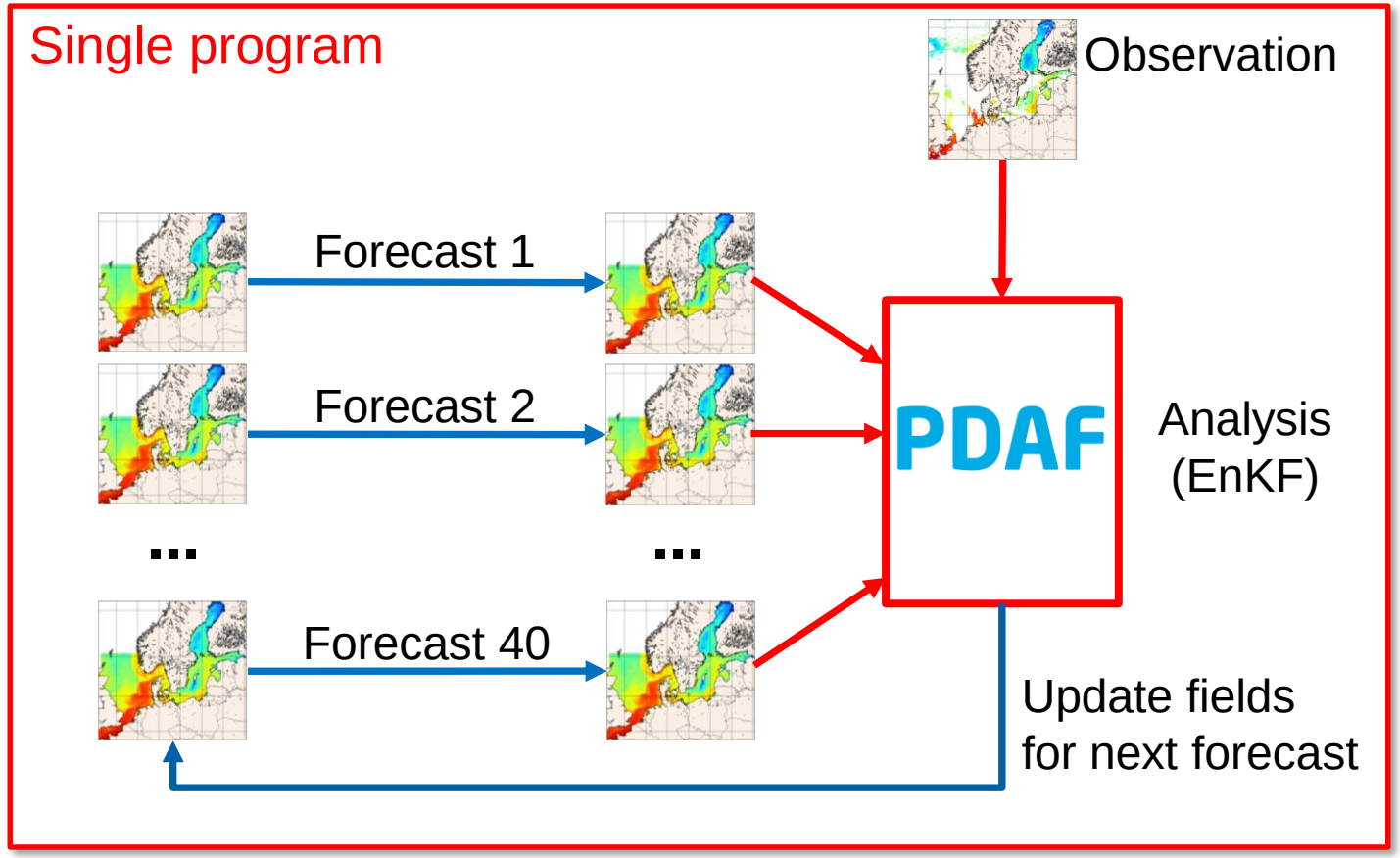
PDAF also supports file-based (offline) coupling of separate programs for model and DA
(but it is less efficient)

AWI

Assimilation-enabled Model

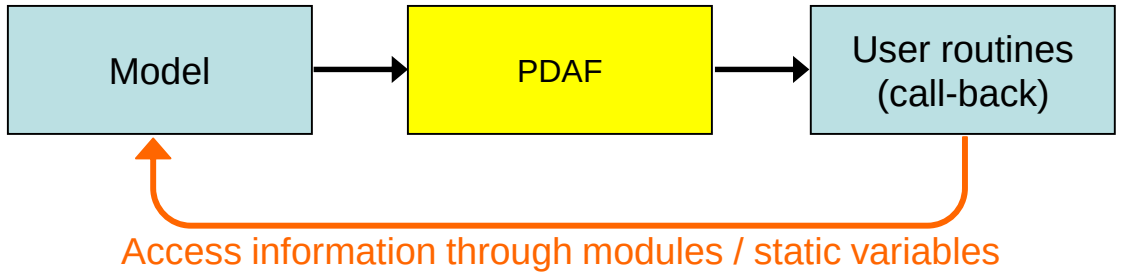
Couple a model with PDAF

- Modify model to simulate ensemble of model states
- Insert analysis step/solver to be executed at prescribed interval
- Run model as usual, but with more processors and additional options



PDAF interface structure

- **Model-sided API:** Defined calls to PDAF routines
- **Case-related API:** User-supplied call-back routines for elementary operations:
  - transfers between model fields and ensemble of state vectors
  - observation-related operations
- User supplied routines can be implemented as routines of the model and can share data with it (low abstraction level)

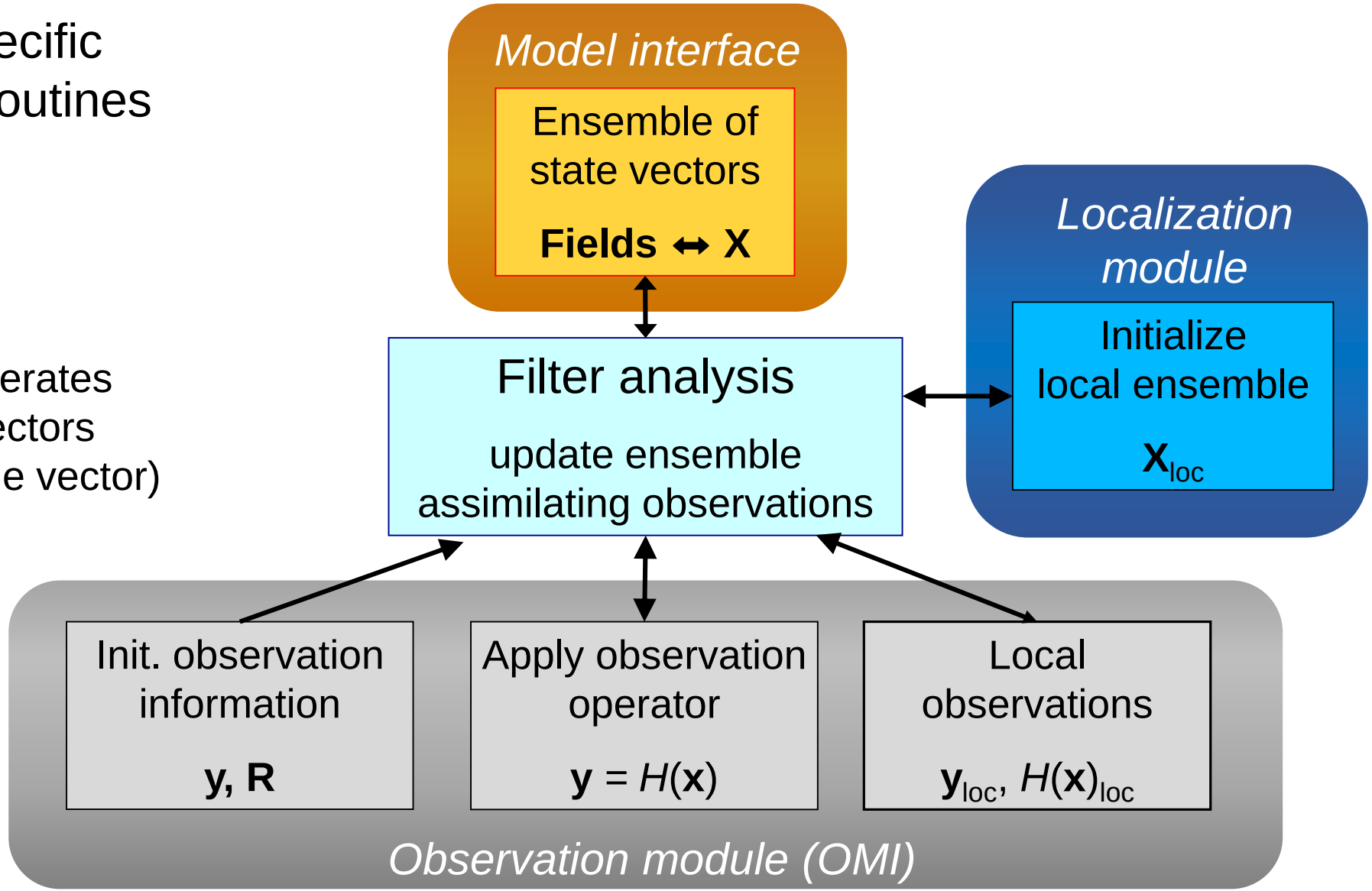


Implementing the Ensemble Analysis Step (Solver)

PDAF Parallel Data Assimilation Framework

case-specific
call-back routines

Analysis operates
on state vectors
(all fields in one vector)



DA Algorithms and models in PDAF

PDAF Parallel Data Assimilation Framework

PDAF originated from comparison studies of different filters

Filters and smoothers - *global and localized versions*

- EnKF (Evensen, 1994 + perturbed obs.)
- (L)ETKF (Bishop et al., 2001)
- ESTKF (Nerger et al., 2012)
- NETF (Toedter & Ahrens, 2015)
- Particle filter
- *EnOI mode*

Upcoming:
- Ensemble 3D-Var
- Hybrid 3D-Var
- Hybrid NETF/LETKF *(see my poster)*

Model bindings

- MITgcm
- AWI-CM / FESOM

Community provided:

SCHISM/ESMF

TerrSysMP-PDAF

Upcoming:
- NEMO 4 (U Reading)
- GOTM/FABM (BB ApS)

Toy models

- Lorenz-96 / Lorenz-63

Upcoming:
- Lorenz-2005 II/III

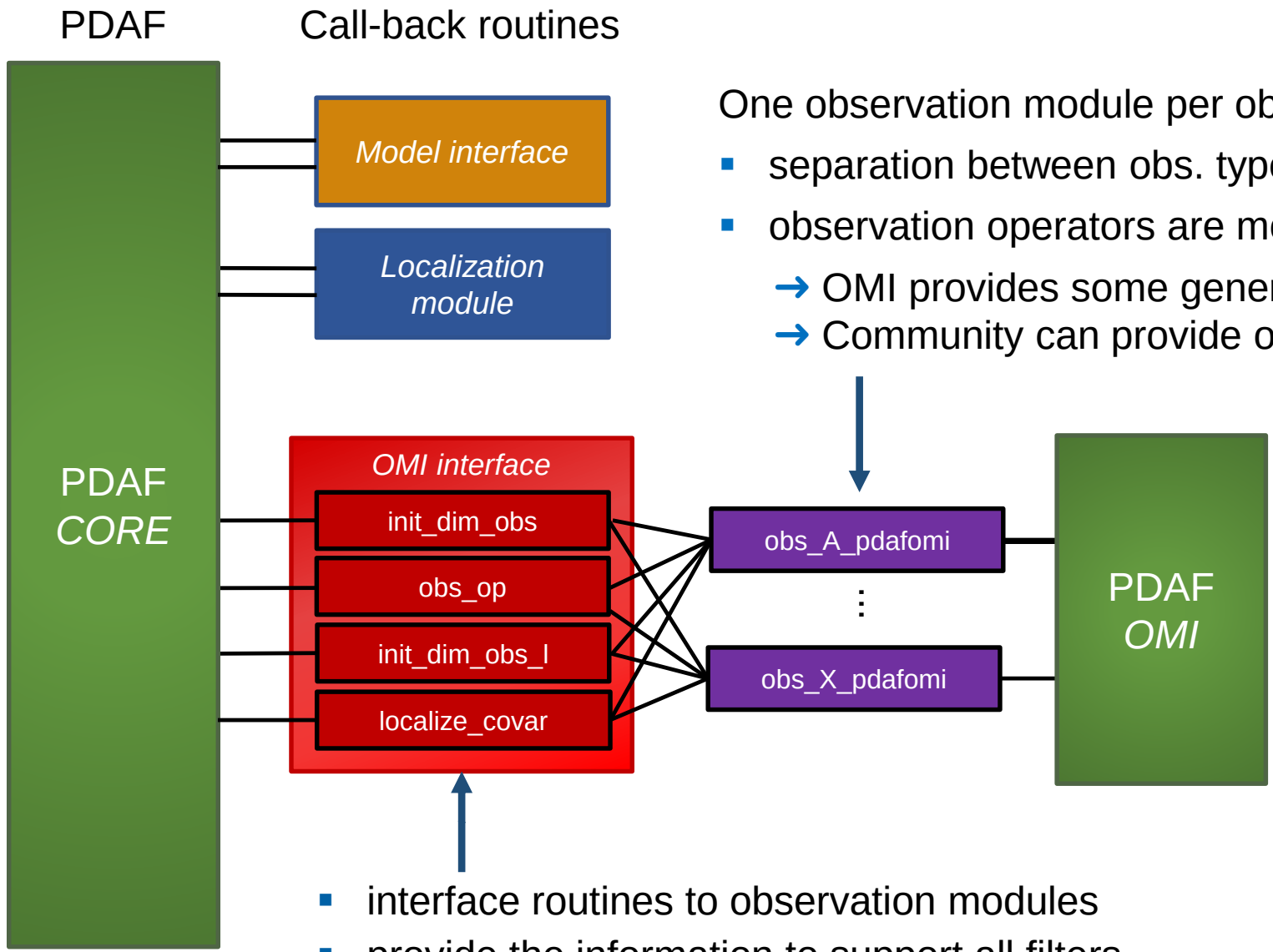
AWI

Recent and current developments

OMI: Code structure (Observation Module Infrastructure)

PDAF Parallel Data Assimilation Framework

Structure motivated by object-oriented programming. For sake of simplicity not implemented with OOP



One observation module per obs. type

- separation between obs. types
- observation operators are model independent
  - → OMI provides some generic operators
  - → Community can provide operators

Part of PDAF library

- interface routines to observation modules
- provide the information to support all filters

Part of PDAF V1.16

AWI

Strongly Coupled DA

PDAF Parallel Data Assimilation Framework

See Poster 18 by Qi Tang et al.

Strongly coupled DA:
Assimilate observation of component A into component B

PDAF supports strongly coupled DA:

achieved by adapting MPI communicator for the filter processes

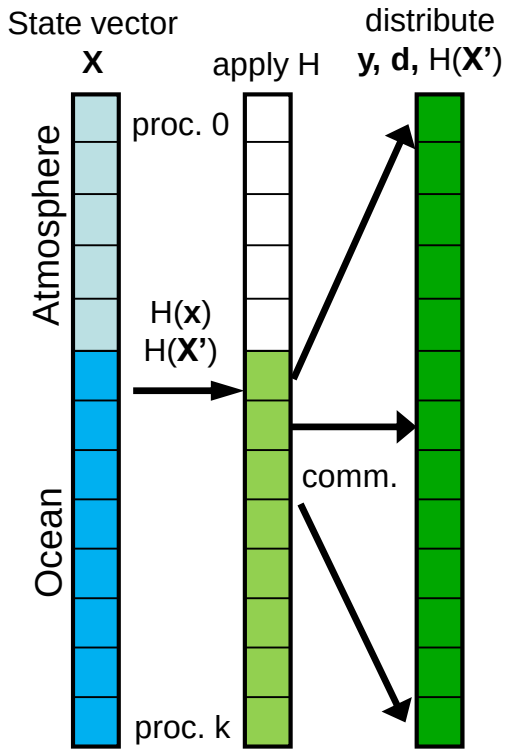
→ joint state vector decomposed over the processes

→ Provide observation operator that only performs MPI communication

need innovation **d** = H(**x**) – **y** and observed ensemble perturbations H(**X'**)

Observation operator H links different compartments

1. Compute part of **d** and H(**X'**) on process 'owning' the observation
2. Communicate **d** and H(**X'**) to processes for which observation is within localization radius



Observation handling in strongly coupled DA

Part of PDAF V1.16

Nerger, Tang, Mu (2020). GMD, 13, 4305–4321, doi:10.5194/gmd-13-4305-2020

Ensemble 3D-Var / Hybrid 3D-Var

Activity in EU-project SEAMLESS

1D Prototype (in development):

- GOTM/FABM + ecosystem models
- DA functionality provided by PDAF

- **Ensemble/Hybrid 3D-Var**
  - Some partners (PML, OGS) use 3D-Var
  - Integrate in PDAF analogous to EnKFs/PFs
  - Focus on infrastructure with optimizers as core
    - (Future PDAF release)

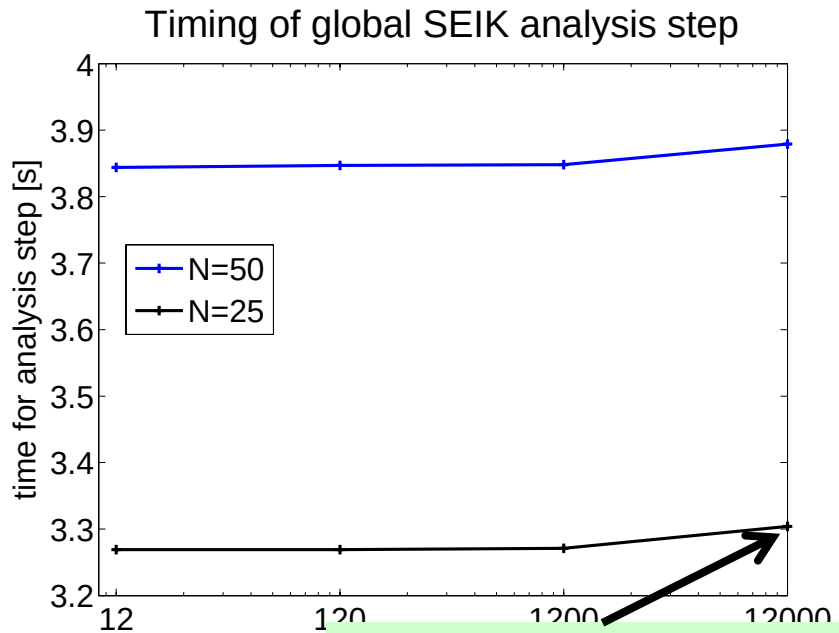
www.seamlessproject.org

Services based on Ecosystem data AssiMiLation: Essential Science and Solutions

PDAF Capability: Very big test case

PDAF Parallel Data Assimilation Framework

- Simulate a “model”
- Choose an ensemble
  - state vector per processor: 10^7
  - observations per processor: $2 \cdot 10^5$
  - Ensemble size: 25
  - 2GB memory per processor
- Apply analysis step for different processor numbers
  - 12 – 120 – 1200 – 12000
- Very small increase in analysis time (~1%) (Ideal would be constant time)



State dimension: 1.2e11

Observation dimension: 2.4e9

- Didn’t try to run a real ensemble of largest state size (no model yet)
- Latest test: analysis step using 57600 processor cores; state dimension 8.6e11

AWI

Summary - PDAF: A tool for data assimilation

PDAF Parallel Data Assimilation Framework

- a program library for ensemble modeling and data assimilation
- provides support for ensemble forecasts, DA diagnostics, and fully-implemented filter and smoother algorithms
- makes good use of supercomputers
- separation of concerns: model, DA methods, observations
- easy to couple to models and to code case-specific routines
- easy to add new DA methods
- efficient for research and operational use

PDAF adds DA functionality to models

Couple model and PDAF within days

Get DA capability in a month

Run DA in known environment

Access new DA methods by updating PDAF

Open source:
Code, documentation, and tutorial available at

http://pdaf.awi.de

L. Nerger, W. Hiller, Computers & Geosciences 55 (2013) 110-118

AWI

References

- **http://pdaf.awi.de**
- Nerger, L., Hiller, W. (2013). Software for Ensemble-based Data Assimilation Systems - Implementation Strategies and Scalability. Computers and Geosciences, 55, 110-118. doi:10.1016/j.cageo.2012.03.026
- Nerger, L., Tang, Q., Mu, L. (2020). Efficient ensemble data assimilation for coupled models with the Parallel Data Assimilation Framework: Example of AWI-CM. Geoscientific Model Development, 13, 4305–4321, doi:10.5194/gmd-13-4305-2020
- Tang, Q., Mu, L., Sidorenko, D., Goessling, H., Semmler, T., Nerger, L. (2020) Improving the ocean and atmosphere in a coupled ocean-atmosphere model by assimilating satellite sea surface temperature and subsurface profile data. Q. J. Royal Metorol. Soc., in press doi:10.1002/qj.3885
- Mu, L., Nerger, L., Tang, Q., Losa, S. N., Sidorenko, D., Wang, Q., Semmler, T., Zampieri, L., Losch, M., Goessling, H. F. (2020) Towards a data assimilation system for seamless sea ice prediction based o the AWI climate model. Journal of Advances in Modeling Earth Systems, 12, e2019MS001937 doi:10.1029/2019MS001937

Requirements

- Fortran compiler
- MPI library
- BLAS & LAPACK
- make

- PDAF is at least tested (often used) on various computers:
  - Notebook & Workstation: MacOS, Linux (gfortran)
  - Cray XC30/40 & CS400 (Cray ftn and ifort)
  - NEC SX-8R / SX-ACE / SX-Aurora TSUBASA
  - ATOS Bull Sequana X (ifort)
  - HPE Cray Apollo (ARM)
  - Legacy:
    - SGI Altix & UltraViolet (ifort) / IBM Power (xlf) / IBM Blue Gene/Q