Blueprints for Next-Generation Data Assimilation Systems, 8-10 March 2016, Boulder, CO

The Parallel Data Assimilation Framework PDAF:

Status and Future Developments

Lars Nerger

Alfred Wegener Institute for Polar and Marine Research
Bremerhaven, Germany

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

PDAF: A tool for data assimilation

PDAF Parallel Data Assimilation Framework

PDAF - Parallel Data Assimilation Framework

- program library for ensemble modeling and data assimilation
- provide support for ensemble forecasts
- provide fully-implemented filter and smoother algorithms
- easily useable with (probably) any numerical model
  (applied with NEMO, MITgcm, FESOM, MPIOM, HBM, NOBM)
- makes good use of supercomputers (Fortran, MPI, OpenMP)
- first public release in 2004; continued development
- ~170 registered users

Free & open source:
Code and documentation available at

http://pdaf.awi.de

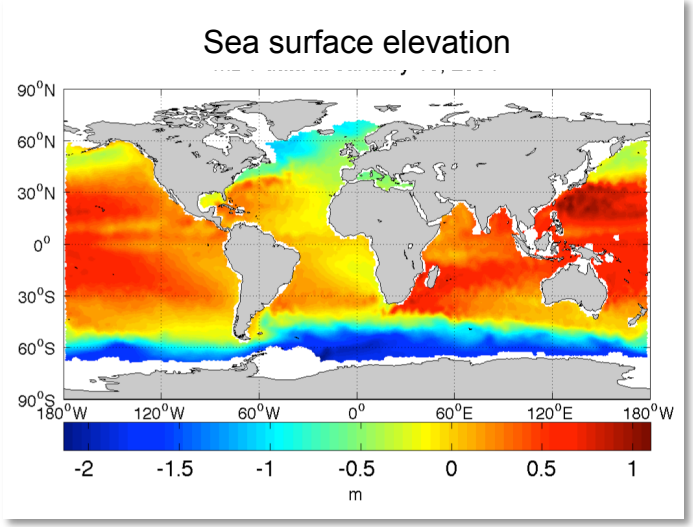
L. Nerger, W. Hiller, Computers & Geosciences 55 (2013) 110-118

AWI

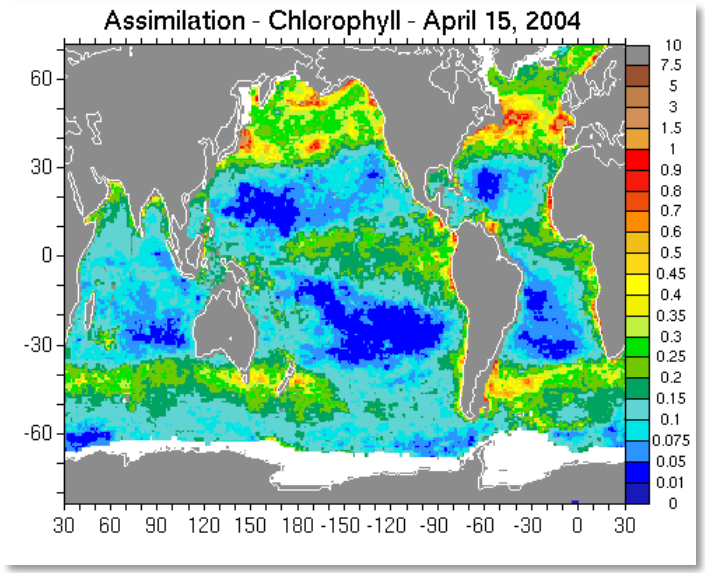
Application examples run with PDAF

PDAF Parallel Data Assimilation Framework

- Ocean state estimation by assimilation of satellite ocean topography data into global model



- Chlorophyll assimilation into global NASA Ocean Biogeochemical Model (with Watson Gregg, NASA GSFC)



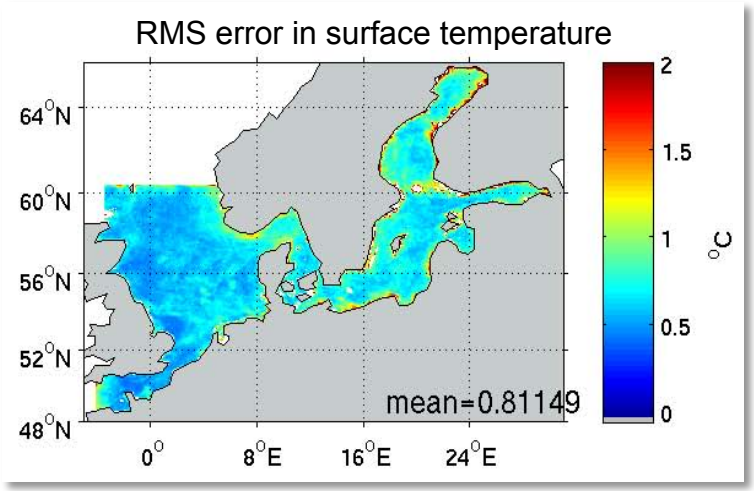
AWI

Application examples run with PDAF

PDAF Parallel Data Assimilation Framework

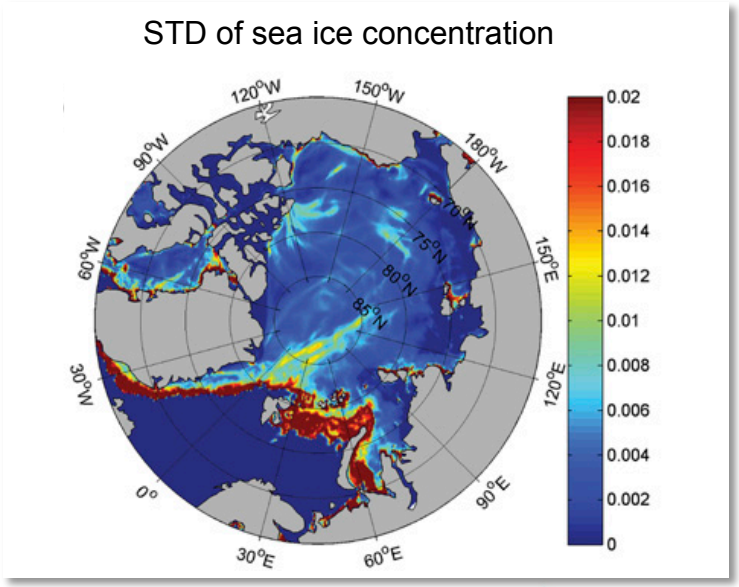
- Regional/coastal assimilation of SST and in situ data (project “DeMarine”, S. Losa)

- Improving sea-ice forecasts assimilating ice concentration and thickness (NMEFC Beijing, Q. Yang)



+ external applications & users, e.g.

- Geodynamo (IPGP Paris, A. Fournier)
- MPI-ESM (coupled ESM, IFM Hamburg, S. Brune/J. Baehr)
- CMEMS BAL-MFC (Copernicus Marine Service Baltic Sea)
- TerrSysMP-PDAF (hydrology, Jülich, Hendricks Franssen)



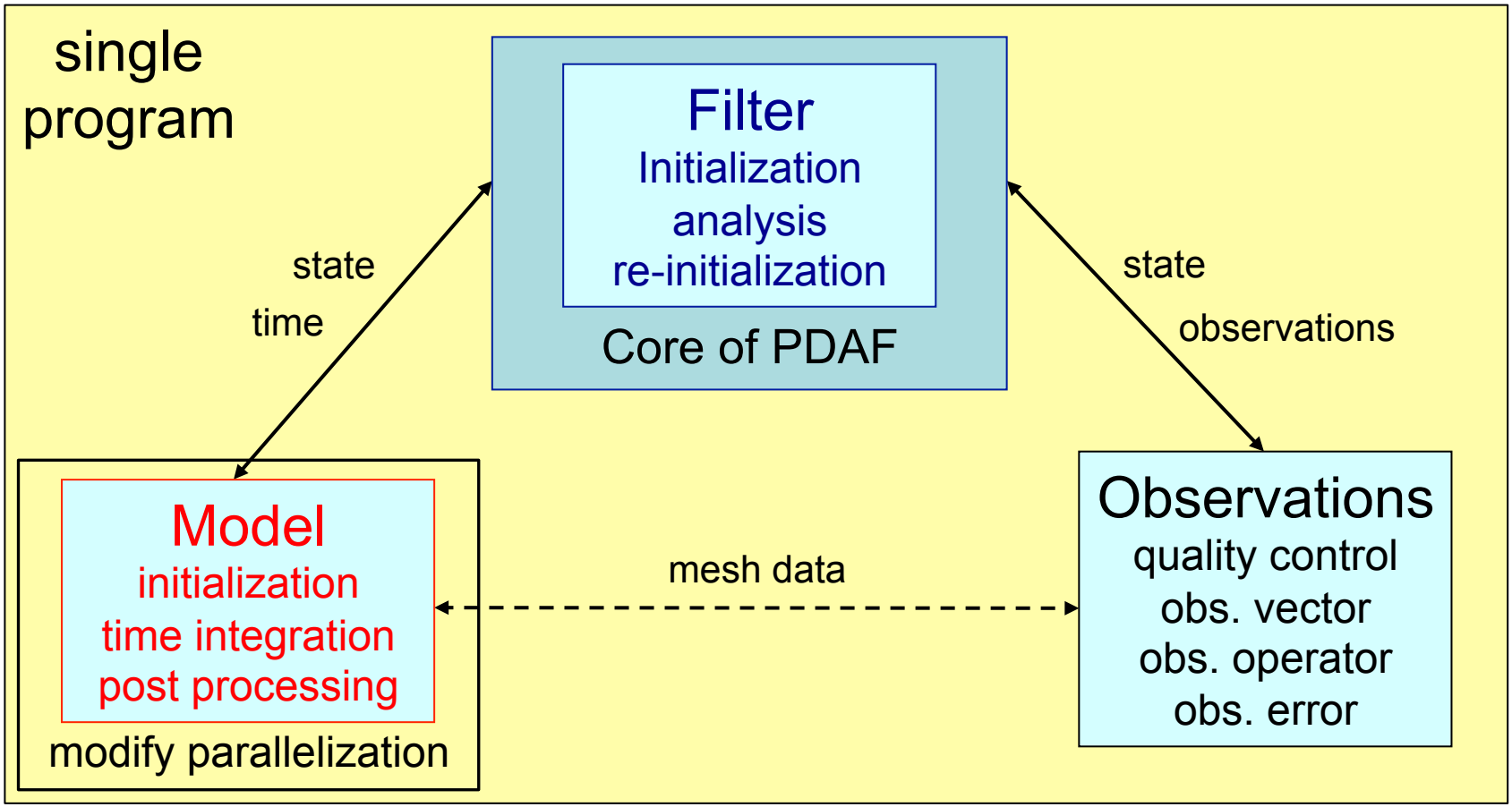
PDAF: Design Considerations

- Focus on ensemble methods
- direct (online/in-memory) coupling of model and data assimilation method (file-based coupling added later)
- minimal changes to model code
  when combining model with PDAF
- model not required to be a subroutine
- control of assimilation program coming from model
- simple switching between different filters and data sets
- complete parallelism
  in model, filter, and ensemble integrations

Implementation Concept

Logical separation of assimilation system

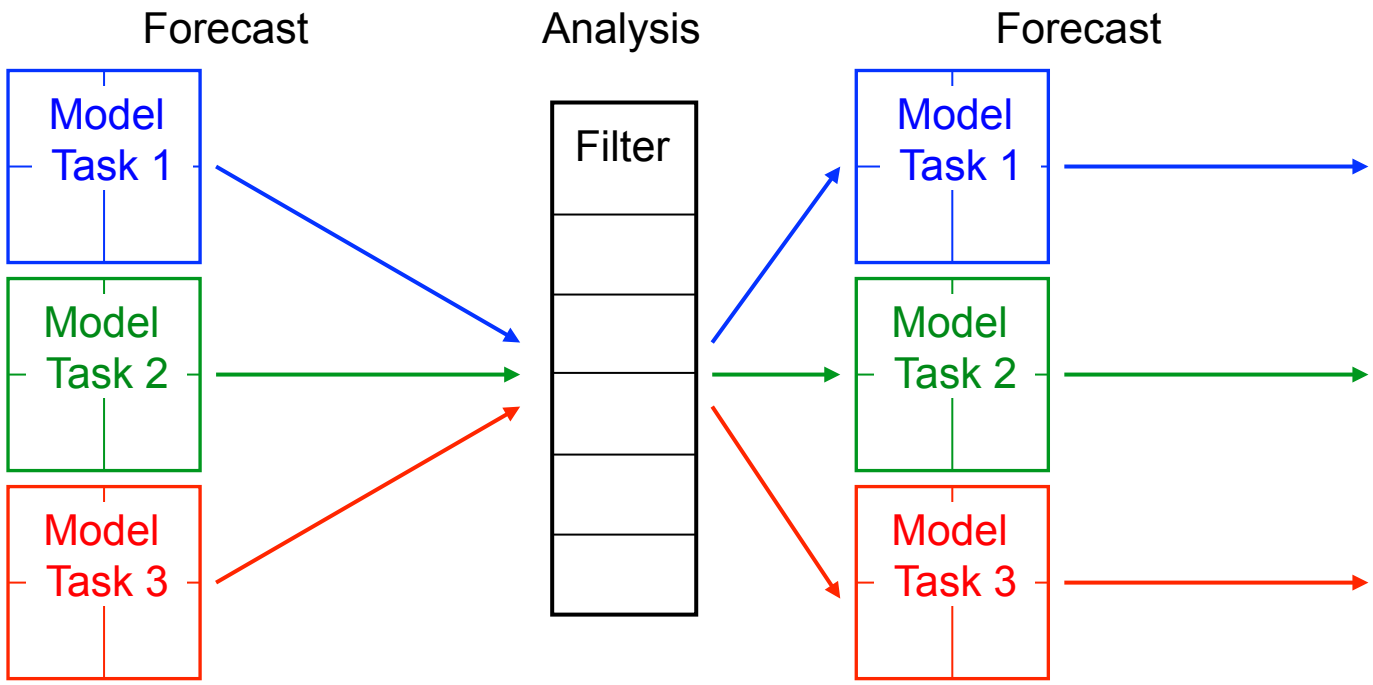
PDAF Parallel Data Assimilation Framework



Nerger, L., Hiller, W. (2013). Software for Ensemble-based DA Systems – Implementation and Scalability. Computers and Geosciences. 55: 110-118

AWI

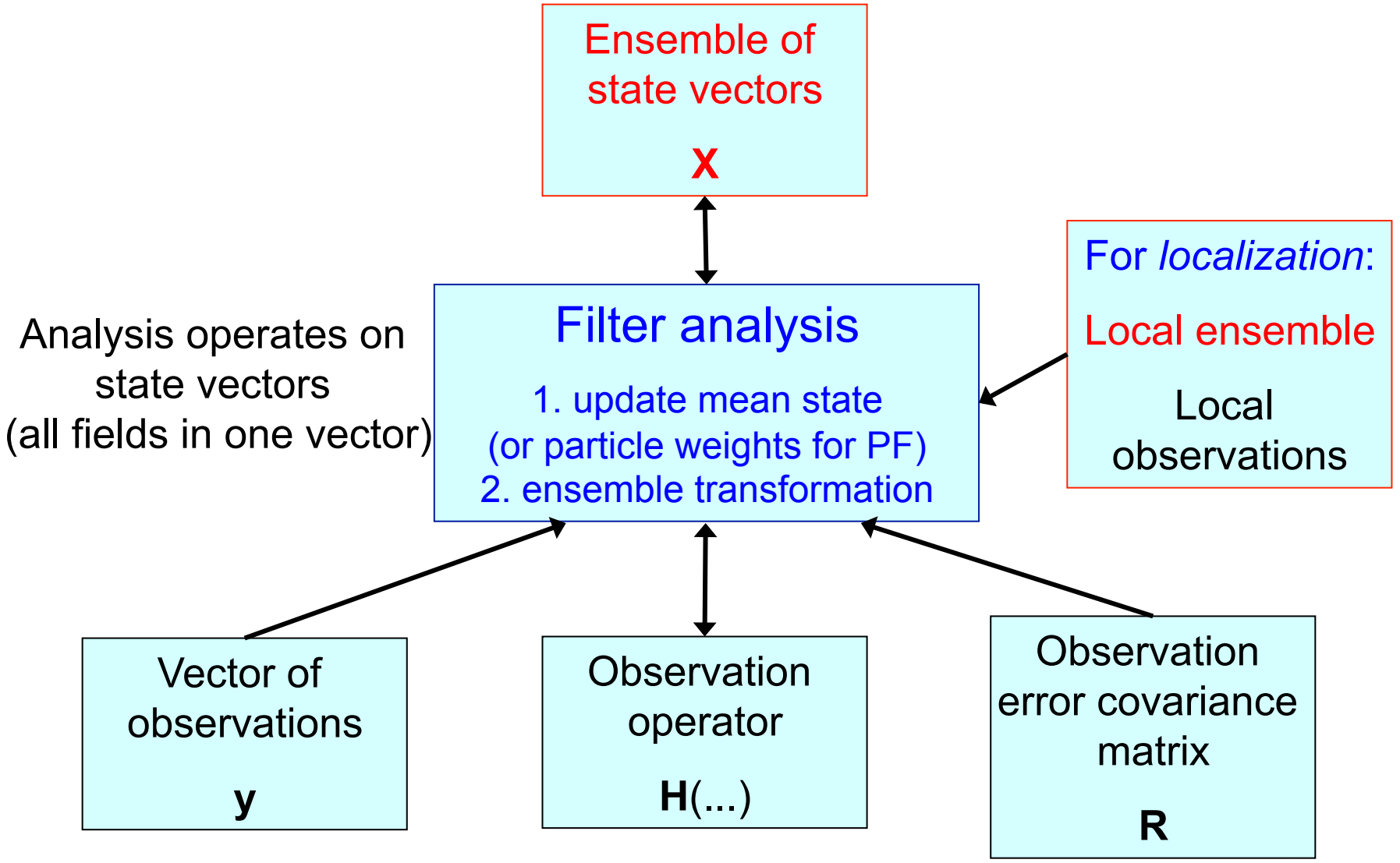
2-level Parallelism



1. Multiple concurrent model tasks
2. Each model task can be parallelized

➢ Analysis step is also parallelized

Ensemble filter/smoother analysis step



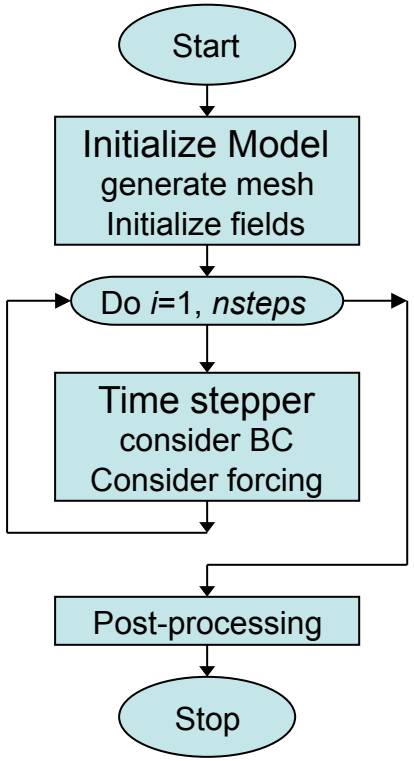
Filter analysis implementation

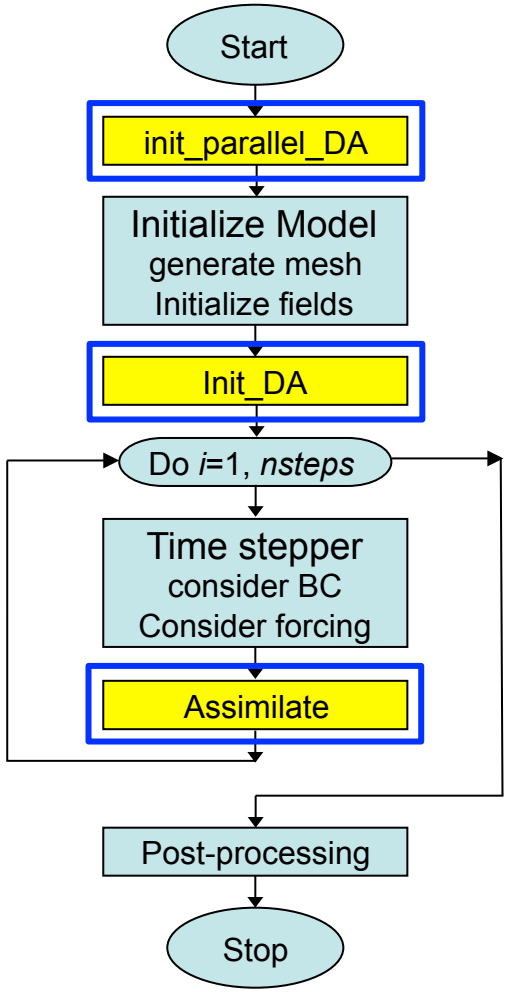
Operate on state vectors

- Filter doesn't know about 'fields'
- Computationally most efficient
- Call-back routines for
  - Transfer between model fields and state vector
  - Observation-related operations
  - Localization operations

Extending a Model for Data Assimilation

Model





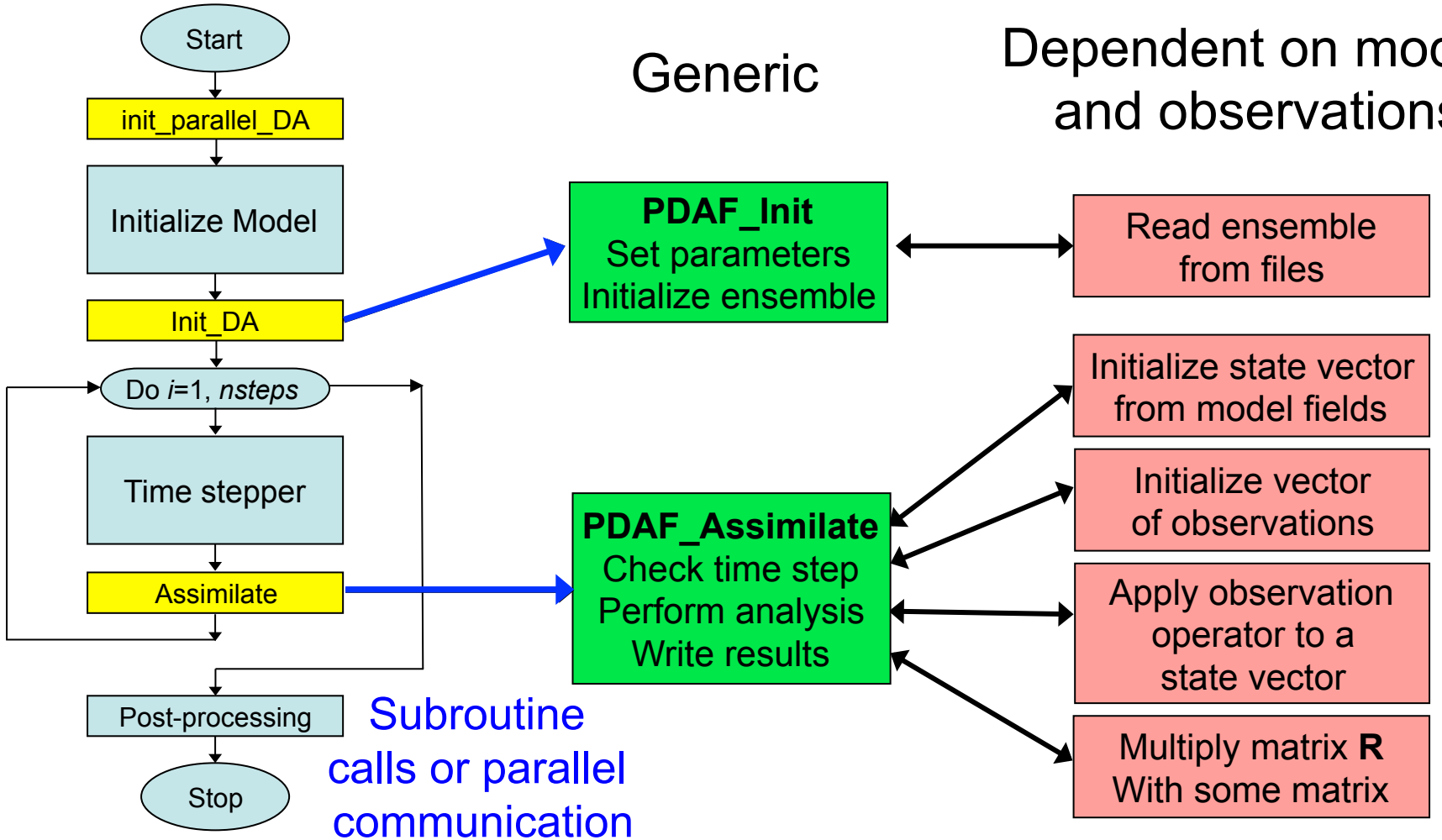
Extension for data assimilation

ensemble forecast enabled by parallelization

plus:
Possible model-specific adaption.

E.g. NEMO: Euler time step after assimilation

Framework solution with generic filter implementation



Model with assimilation extension

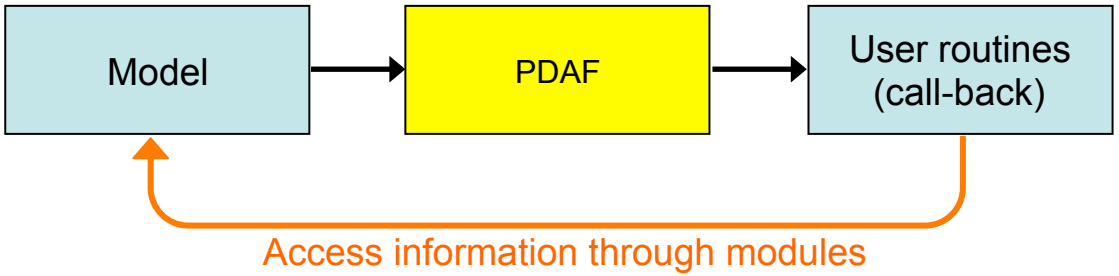
Core-routines of assimilation framework

Case specific call-back routines

PDAF interface structure

PDAF Parallel Data Assimilation Framework

- Defined calls to PDAF routines and to call-back routines
- Model und observation specific operations:
  elementary subroutines implemented in model context
- User-supplied call-back routines for elementary operations:
  - transfers between model fields and ensemble of state vectors
  - observation-related operations
  - filter pre/post-step to analyze ensemble
- User supplied routines can be implemented
  as routines of the model (e.g. share common blocks or modules)



Parallelization: MPI Communicators

Communicators define a group of processes for data exchange

3 communicator sets are required:

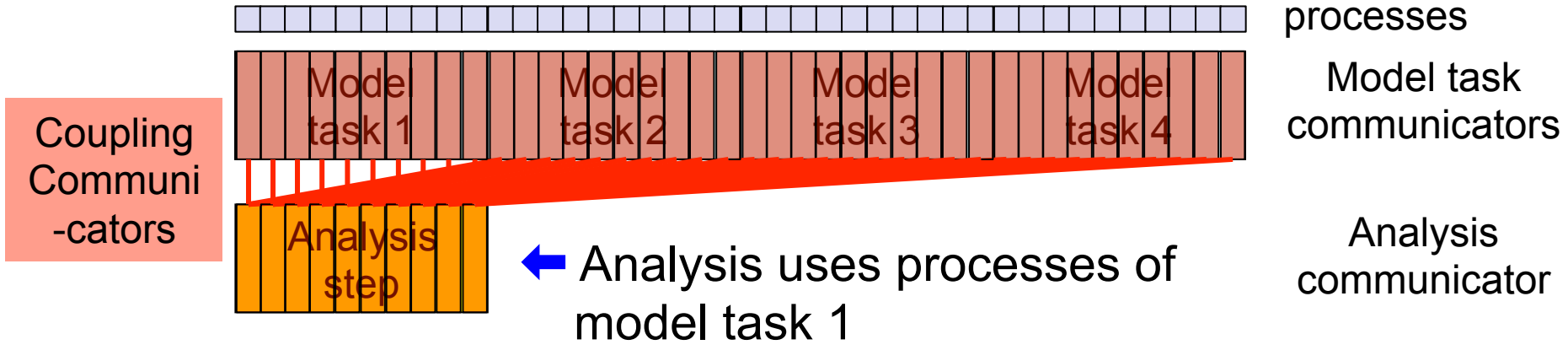
1. Model communicators (one set for each model task)
2. Filter communicator (a single set of processes)
3. Coupling communicators
   – to send data between model and filter
   (one set for each filter process and connected model processes)

Configuring the parallelization (MPI)

PDAF Parallel Data Assimilation Framework

- Assume *4* ensemble members
- Model itself is parallelized (like domain decomposition)
- Configuration of “MPI communicators” (groups of processes)

Variant 1:

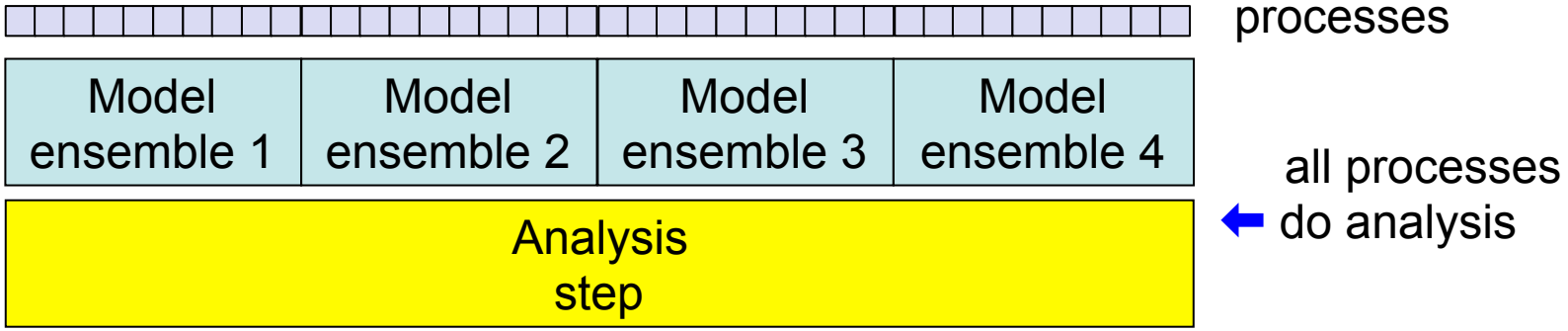


- Default communication variant of PDAF
- Default *init_parallel_pdaf* provides this configuration
- Reasoning: Convenience to use same domain decomposition for model and analysis (also efficient for ocean with satellite data)

Alternative Configurations

If you worry about idle processes

Variant 2:



Issues:

- Communication pattern more complicated
- More time in communications

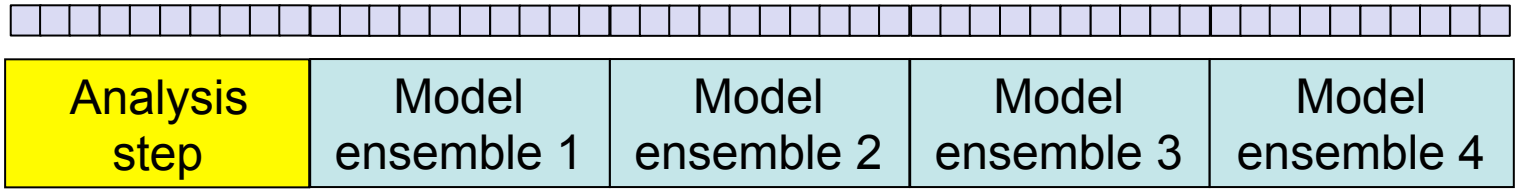
In domain-decomposed models:

- Need a decomposition of process sub-domains
  (didn't try this with our finite-element model FESOM needing partitioner METIS)

Alternative Configurations

PDAF Parallel Data Assimilation Framework

Variant 3: (just replace *init_parallel_pdaf*)



⬆ Separate set of processes

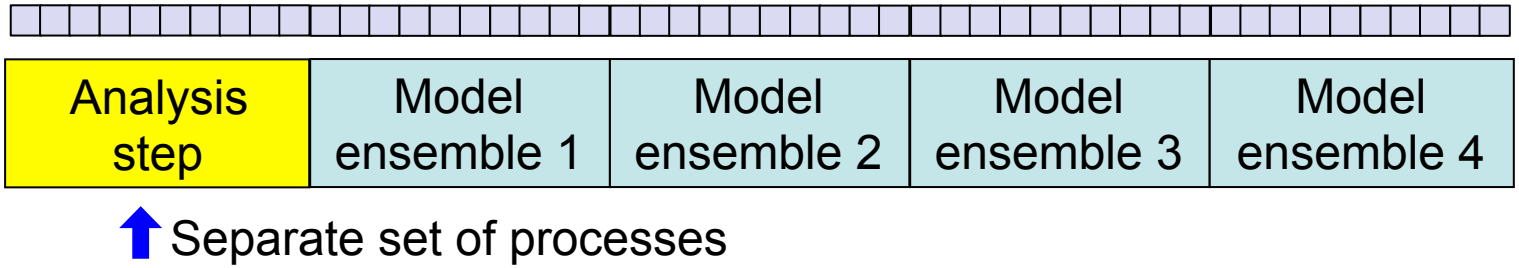
When memory is really limited

- Analysis processes might idle during forecast
- Might allow for observation preparations during forecast phase
- Also configurable: Separation into two programs

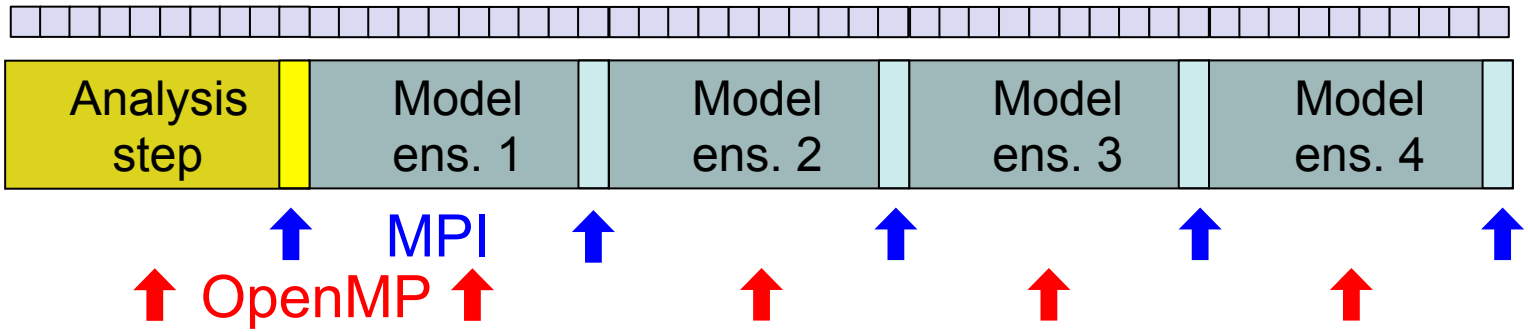
Alternative Configurations

PDAF Parallel Data Assimilation Framework

Variant 3: (just replace *init_parallel_pdaf*)



Variant 4: (supported since PDAF release V1.11)



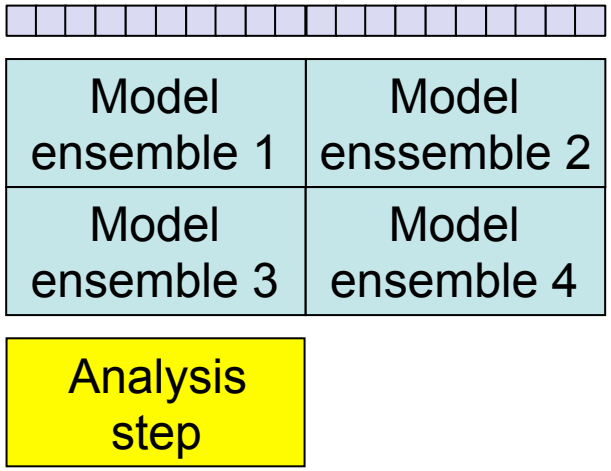
- Hybrid parallelization (MPI and OpenMP)
- Analysis on model task 1 or separate

AWI

Alternative Configurations

Issue: Configuration of coupling communicators is more complicated

Variant 5:

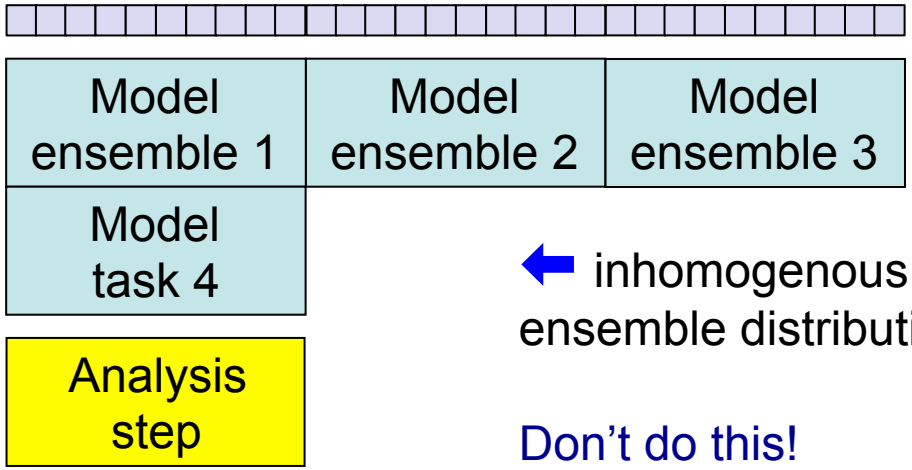


processes

⬅ less model tasks than ensemble members

Needs fully flexible implementation!

Variant 5b:

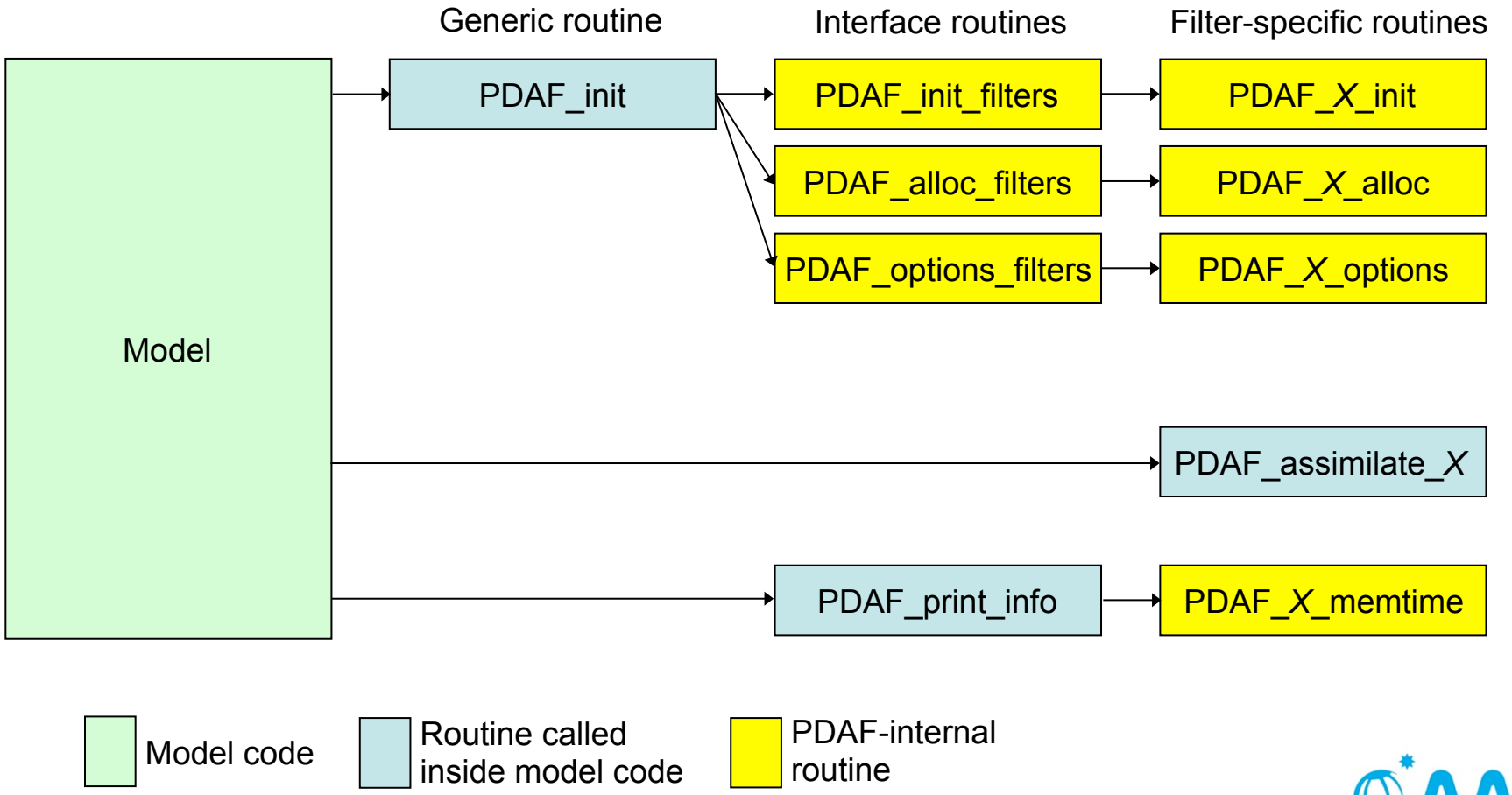


⬅ inhomogenous ensemble distribution

Don't do this!

Internal interface of PDAF

- PDAF has a framework structure for ensemble forecasts
- Internal interface to connect filter algorithms
  (Easy addition of new filters by extending interface routines)



Current algorithms in PDAF

PDAF Parallel Data Assimilation Framework

PDAF originated from comparison studies of different filters

Filters

- EnKF (Evensen, 1994 + perturbed obs.)
- ETKF (Bishop et al., 2001)
- SEIK filter (Pham et al., 1998)
- SEEK filter (Pham et al., 1998)
- ESTKF (Nerger et al., 2012)

Global filters

Not yet released:
- serial EnSRF
- Particle filter
- EWPF
- NETF

- LETKF (Hunt et al., 2007)
- LSEIK filter (Nerger et al., 2006)
- LESTKF (Nerger et al., 2012)

Localized filters

Smoothers for

- ETKF/LETKF
- ESTKF/LESTKF
- EnKF

Global and local smoothers

AWI

Compute performance of PDAF

Parallel Performance – with FESOM

Use between 64 and 4096 processors of SGI Altix ICE cluster (Intel processors)

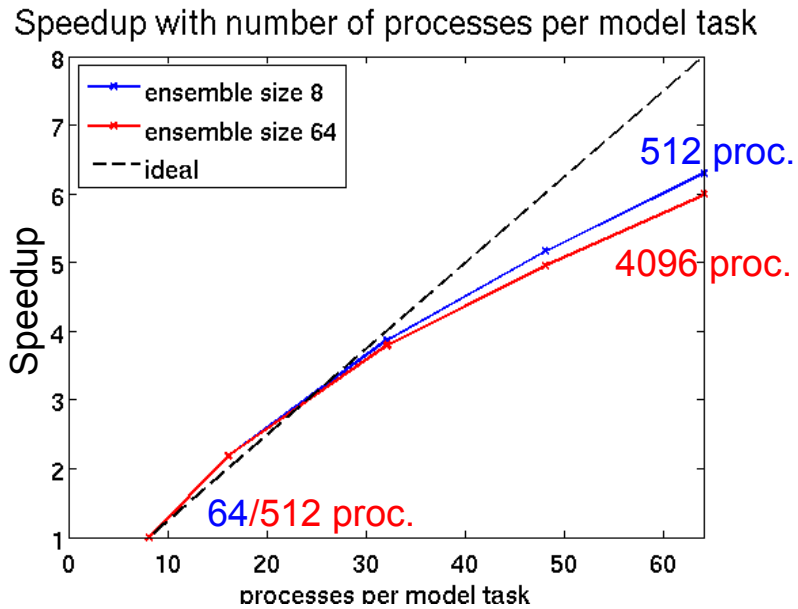
94-99% of computing time in model integrations

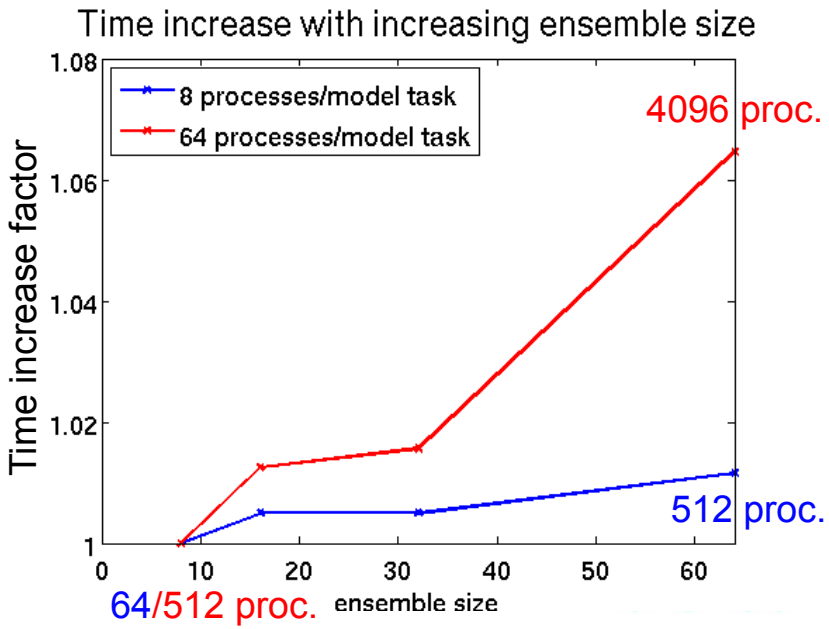
Speedup: Increase number of processes for each model task, fixed ensemble size

- factor 6 for 8x processes/model task
- one reason: time stepping solver needs more iterations

Scalability: Increase ensemble size, fixed number of processes per model task

- increase by ~7% from 512 to 4096 processes (8x ensemble size)
- one reason: more communication on the network

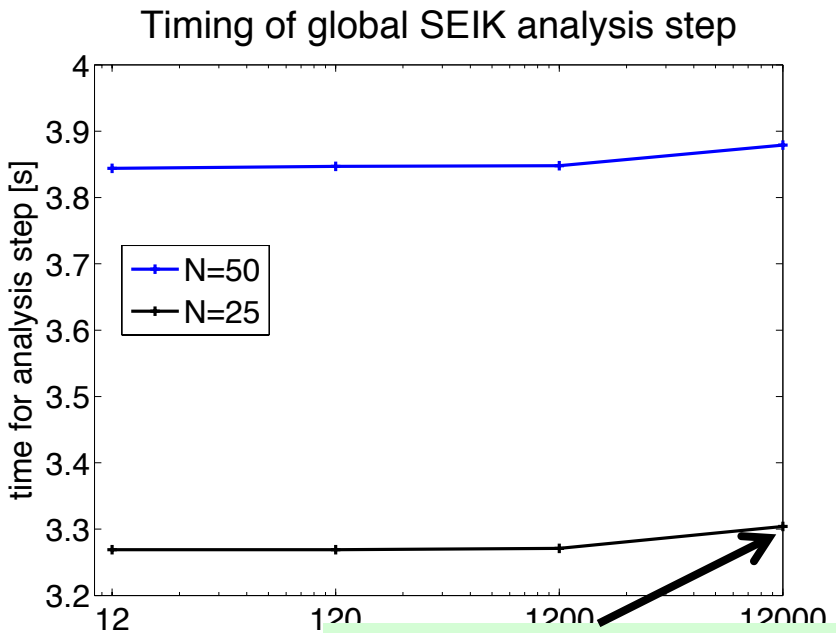




Very big test case

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



State dimension: 1.2e11

Observation dimension: 2.4e9

- Very small increase in analysis time (~1%)
- Didn’t try to run a real ensemble of largest state size (no model yet)

Requirements

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

- PDAF at least tested (often used) on various computers:
  - Laptop & Workstation: MacOS, Linux (gfortran)
  - Cray XC30/40 (Cray ftn and ifort)
  - NEC SX-8R / SX-ACE
  - SGI Altix & UltraViolet (ifort)
  - IBM Power 6 (xlf)
  - IBM Blue Gene/Q

Future developments

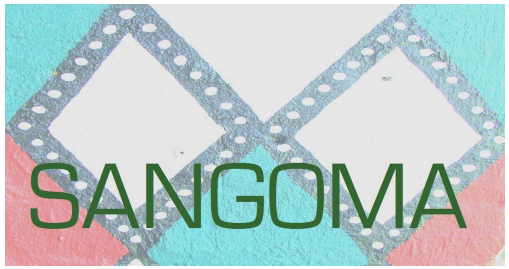
- Prepare model-specific routine packages
- Integrate more diagnostics
- Additional tools for observation handling

- Revision for Fortran 2003 standard
- GPGPU/Intel Phi support?

More Assimilation tools

- SANGOMA: Stochastic Assimilation for Next Generation Ocean Model Applications
- Project funded by European Union 2011-2015

- Different benchmark setups for ocean data assimilation
- Development of set of ~50 data assimilation tools
  - Large set of different diagnostics (beyond RMS errors)
  - Tools for ensemble generation
  - Simplified filter analysis steps



www.data-assimilation.net

PDAF: A tool for data assimilation

PDAF Parallel Data Assimilation Framework

PDAF - Parallel Data Assimilation Framework

- program library for ensemble modeling and data assimilation
- provide support for ensemble forecasts and provide fully-implemented filter and smoother algorithms
- makes good use of supercomputers (Fortran, MPI, OpenMP)
- separates development of DA methods from model
- easy to couple to models and to code case-specific routines
- easy to add new DA methods
  (structure should support any ensemble-based method)
- efficient for research and operational use

Free & open source:
Code and documentation available at

http://pdaf.awi.de

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

AWI