DARC, University of Reading, UK, June 13, 2018

Building a Scalable Ensemble Data Assimilation System for Coupled Models with PDAF

Lars Nerger, Qi Tang, Dmitry Sidorenko

Alfred Wegener Institute for Polar and Marine Research
Bremerhaven, Germany

ESM
Advanced
Earth System Modelling
Capacity

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

Motivation

The technical side of data assimilation:

How to build an efficient data assimilation system
– in a simple way?

Discussed here for a
coupled atmosphere-ocean model

Strategy:

1. Extend model to integrate an ensemble
   - mainly: adapt parallelization
2. Add analysis step to the model
   - just an update in between time steps

PDAF: A tool for data assimilation

PDAF - Parallel Data Assimilation Framework

- a program library for ensemble data assimilation
- provide support for parallel ensemble forecasts
- provide fully-implemented & parallelized filters and smoothers (EnKF, LETKF, NETF, EWPF … easy to add more)
- easily useable with (probably) any numerical model (applied with NEMO, MITgcm, FESOM, HBM, TerrSysMP, …)
- run from laptops to supercomputers (Fortran, MPI & OpenMP)
- first public release in 2004; continued development
- ~300 registered users; community contributions

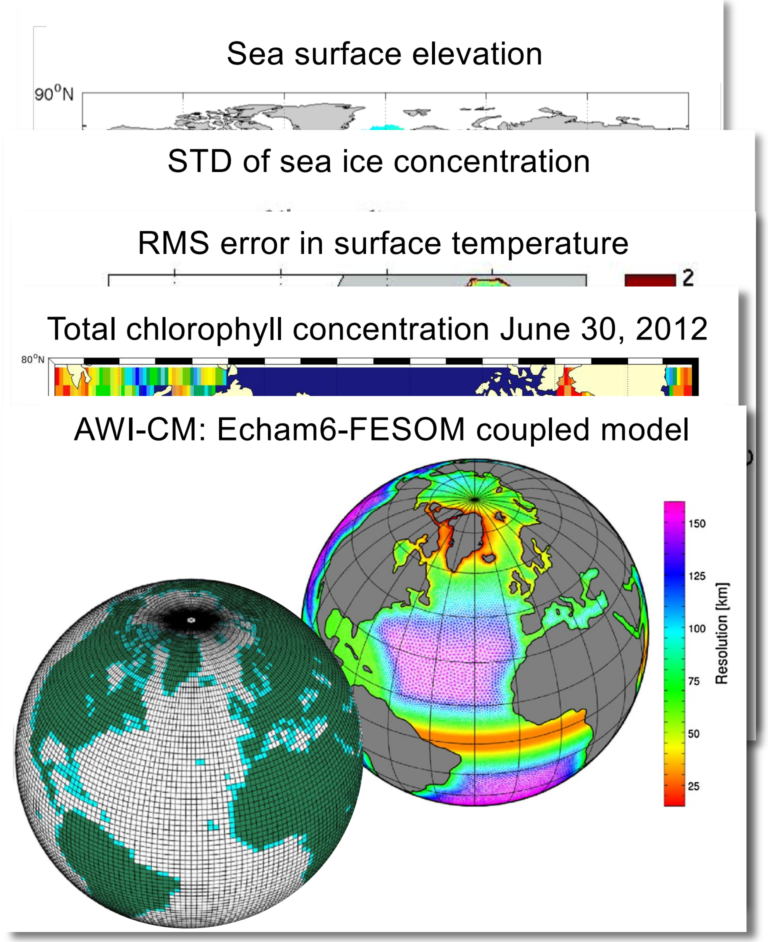
Open source:
Code, documentation & tutorials at

http://pdaf.awi.de

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

Application examples run with PDAF

- FESOM: Global ocean state estimation (Alexey Androsov)
- MITgcm: sea-ice assimilation (Q. Yang et al., NMEFC Beijing)
- HBM: Coastal assimilation of SST, in situ and ocean color (S. Losa, M. Goodliff)
- MITgcm-REcoM: ocean color assimilation for parameter estimation (Himansu Pradhan)
- AWI-CM: coupled atmos.-ocean assimilation (project ESM, Qi Tang)

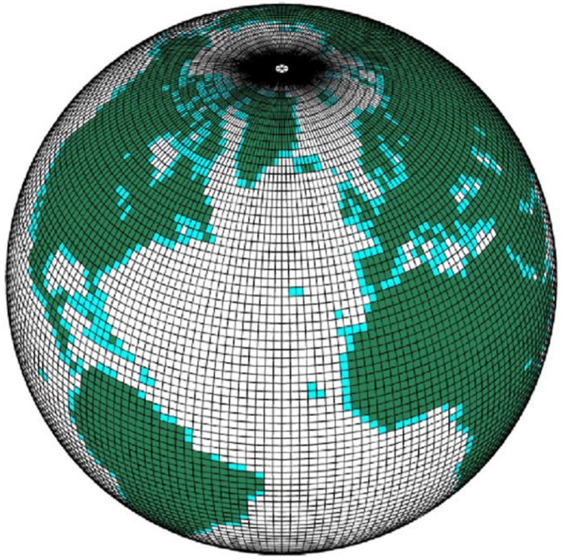


+ external applications & users, e.g.

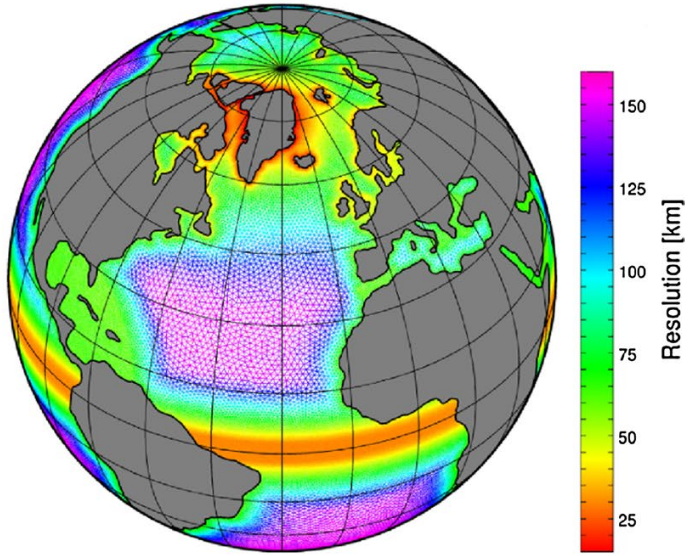
- Geodynamo (IPGP Paris, A. Fournier)
- TerrSysMP-PDAF (hydrology, FZJ)
- MPI-ESM (coupled ESM, IFM Hamburg, S. Brune/J. Baehr)
- CMEMS BAL-MFC (Copernicus Marine Service Baltic Sea)
- CFSv2 (J. Liu, IAP-CAS Beijing)

Example: ECHAM6-FESOM (AWI-CM)

Atmosphere



Ocean



OASIS3-MCT

fluxes →

← ocean/ice state

Atmosphere
- ECHAM6
- JSBACH land

Ocean
- FESOM
- includes sea ice

Coupler library
- OASIS3-MCT

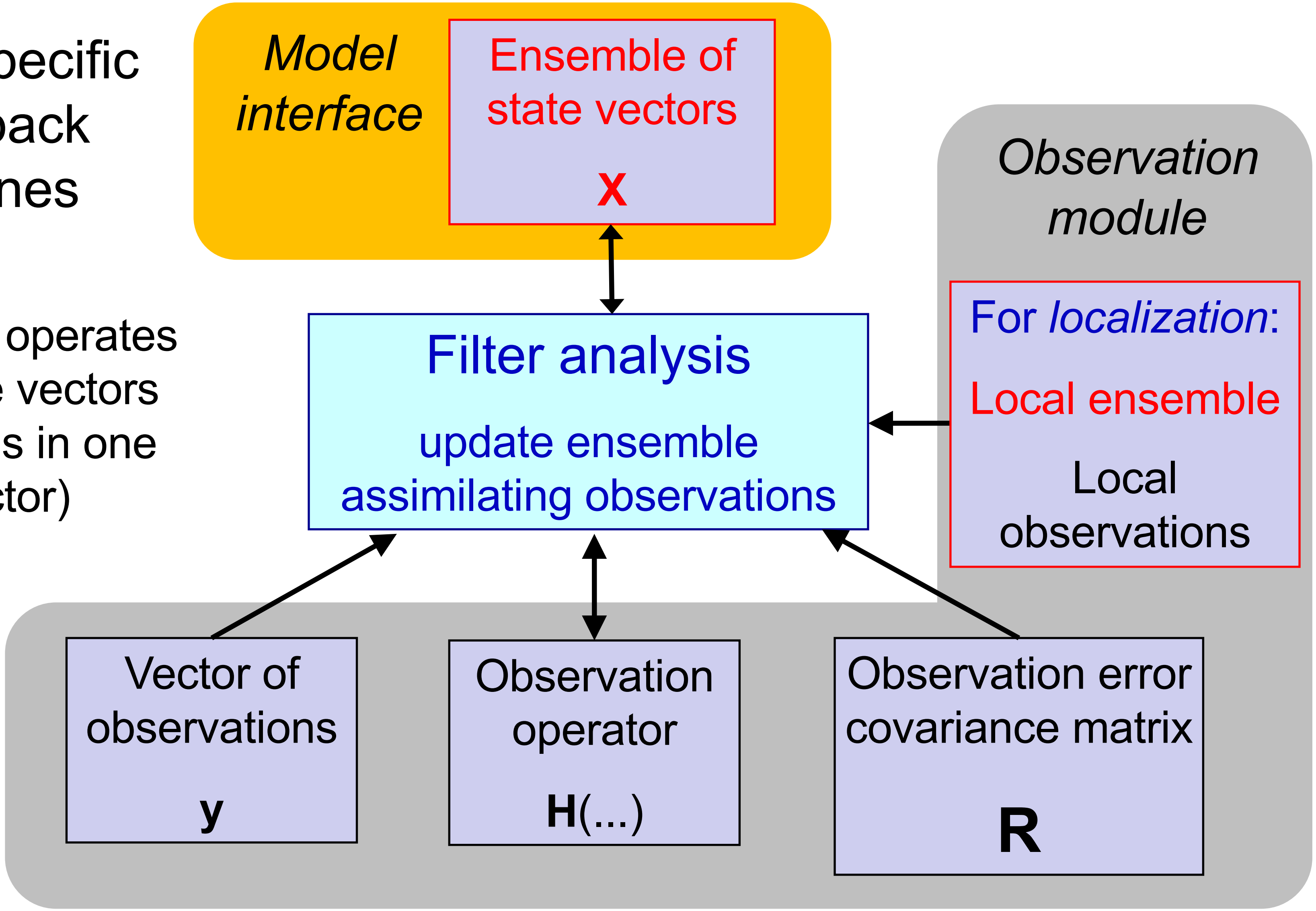
Two separate executables for atmosphere and ocean

D. Sidorenko et al., Clim. Dyn. 44 (2015) 757

Ensemble Filter Analysis Step

case-specific
call-back
routines

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

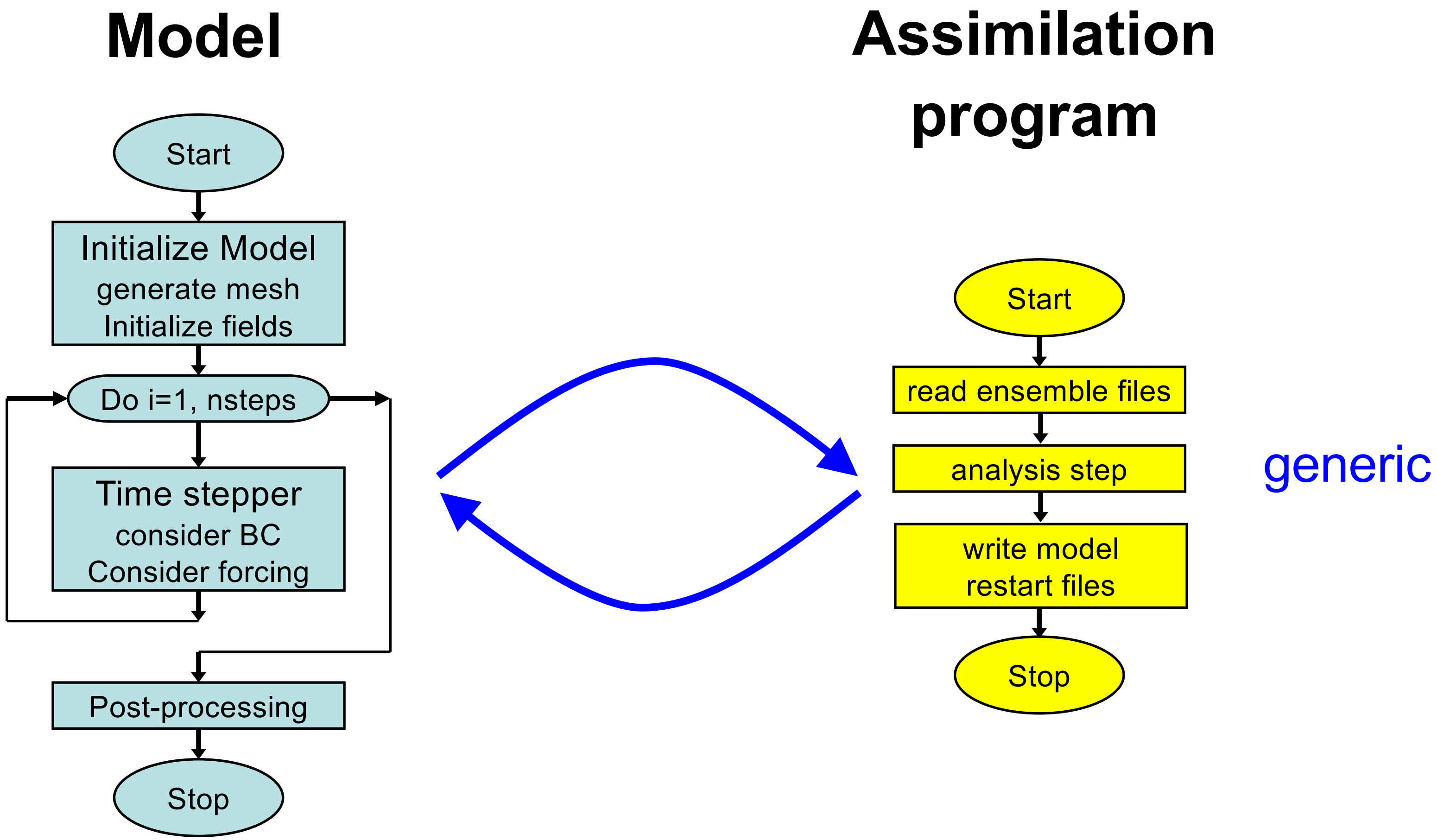


Filter analysis implementation

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

Offline mode – separate programs

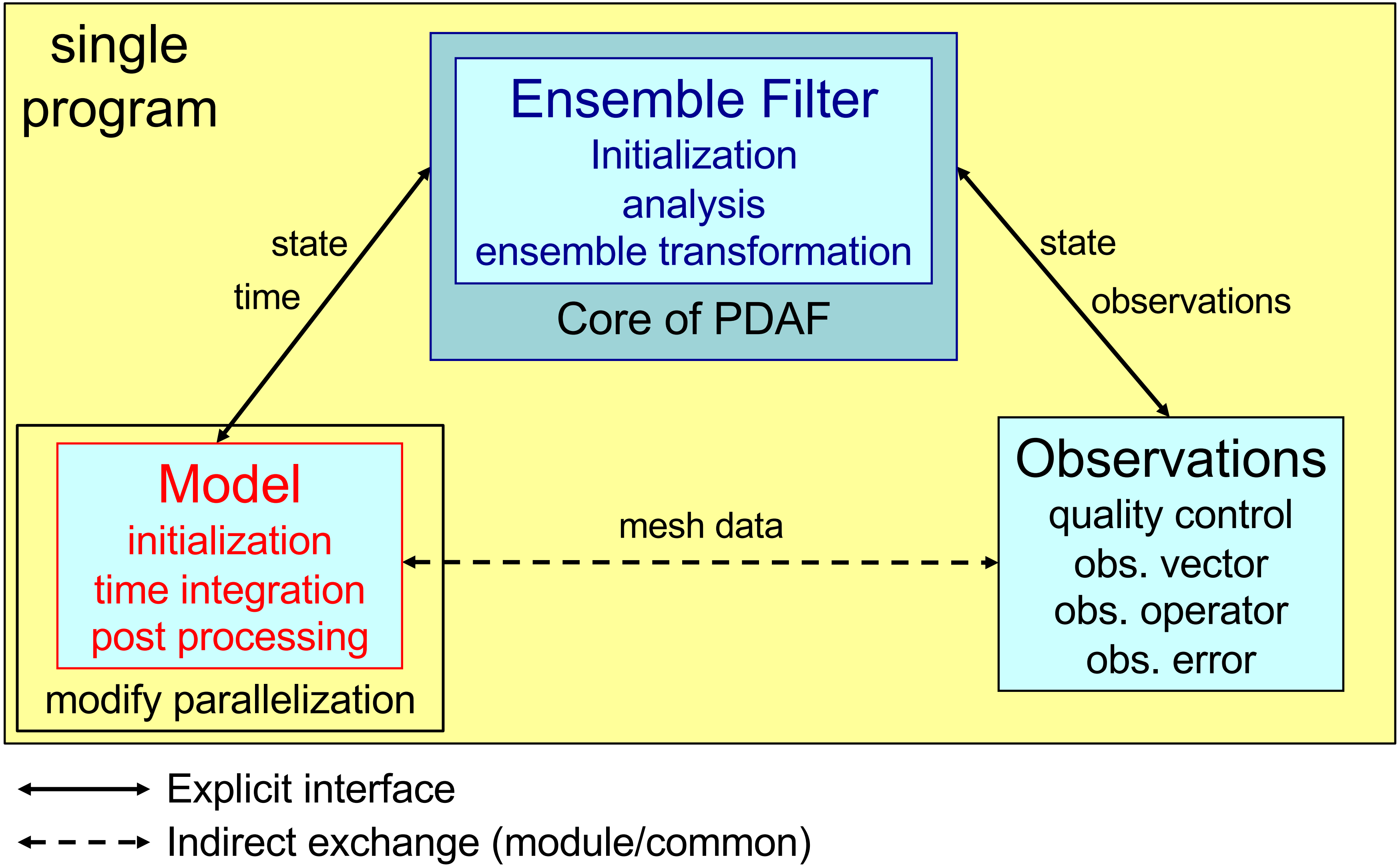


For each ensemble state

- Initialize from restart files
- Integrate
- Write restart files

- Read restart files (ensemble)
- Compute analysis step
- Write new restart files

Logical separation of assimilation system



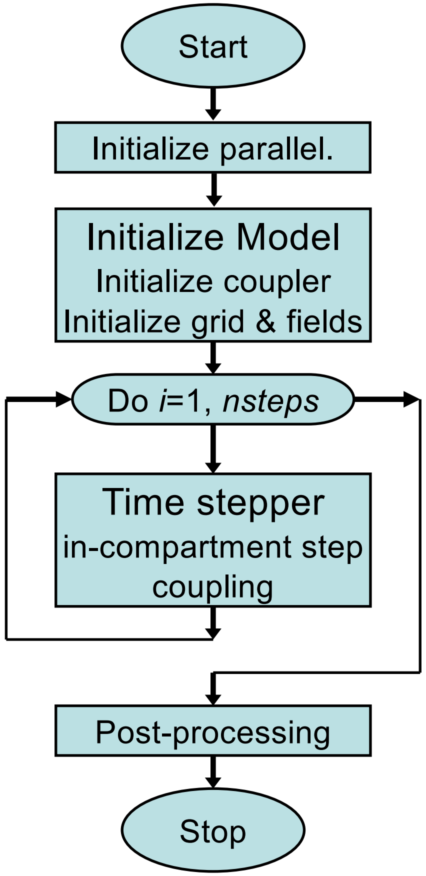
Nerger, L., Hiller, W. Software for Ensemble-based DA Systems – Implementation and Scalability. Computers and Geosciences 55 (2013) 110-118

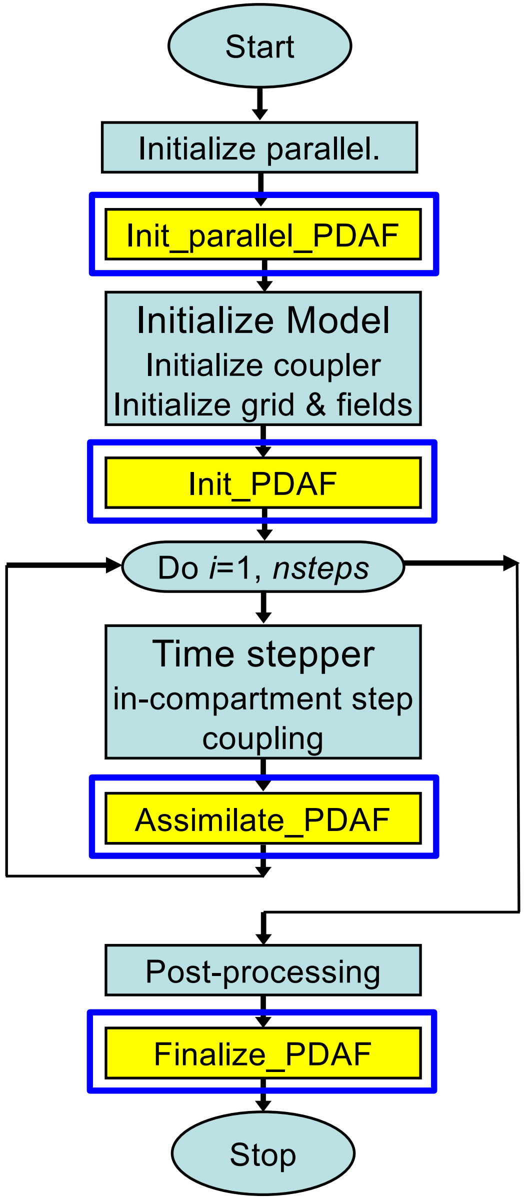
Extending a Model for Data Assimilation

Model

single or multiple executables

coupler might be separate program





Extension for data assimilation

plus:
Possible model-specific adaption

AWI-CM: adaption of coupler (e.g. OASIS3-MCT)

revised parallelization enables ensemble forecast

PDAF: User-friendliness

PDAF Parallel Data Assimilation Framework

Assumption: Users know their model

➜ let users implement DA system in model context

For users, model is not just a forward operator

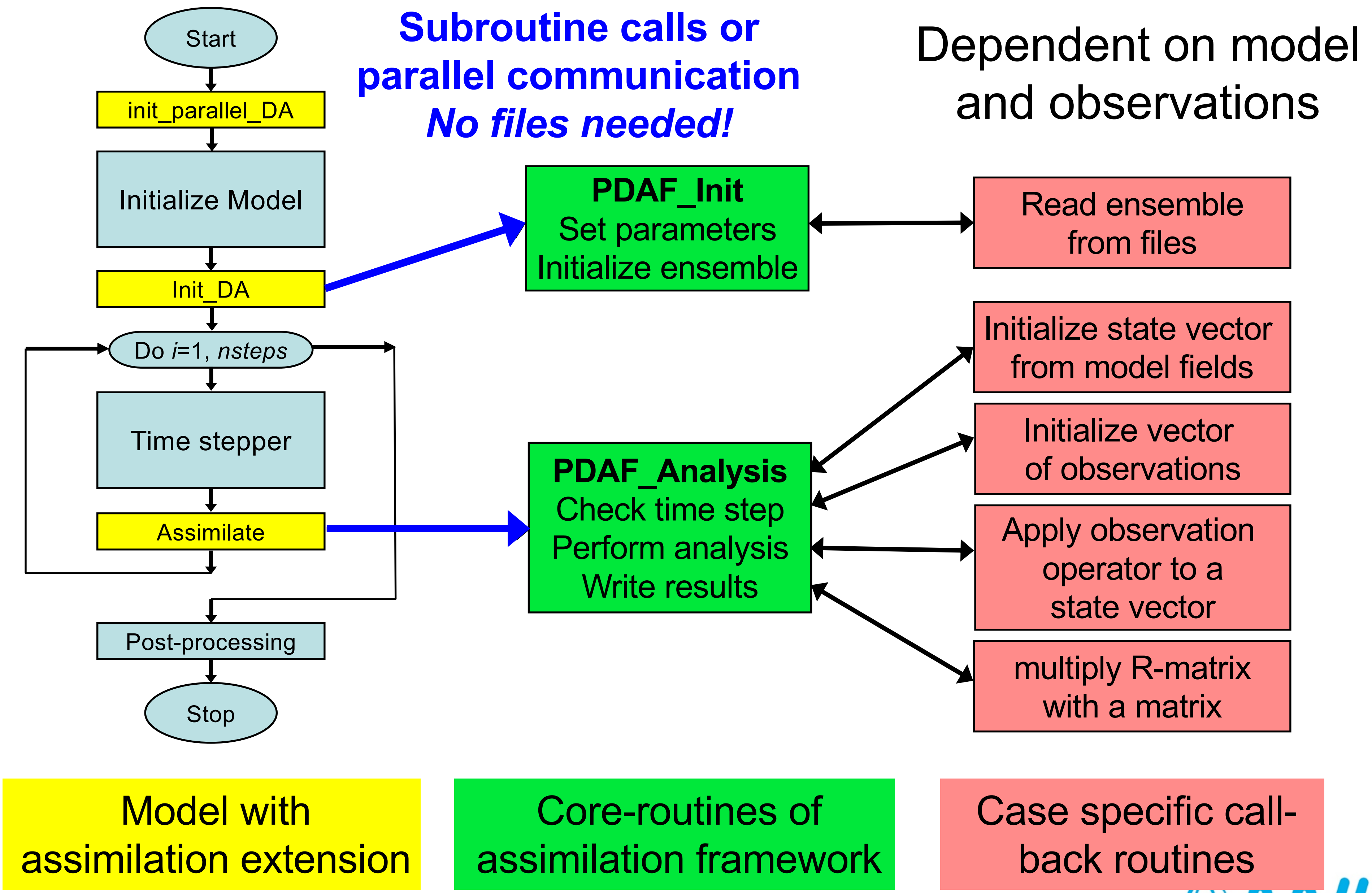
➜ let users extend they model for data assimilation

Keep simple things simple:

- Define subroutine interfaces to separate model and assimilation based on arrays
- No object-oriented programming
(most models don't use it; most model developers don't know it; not many objects would be involved)
- Users directly implement observation-specific routines
(no indirect description of e.g. observation layout)

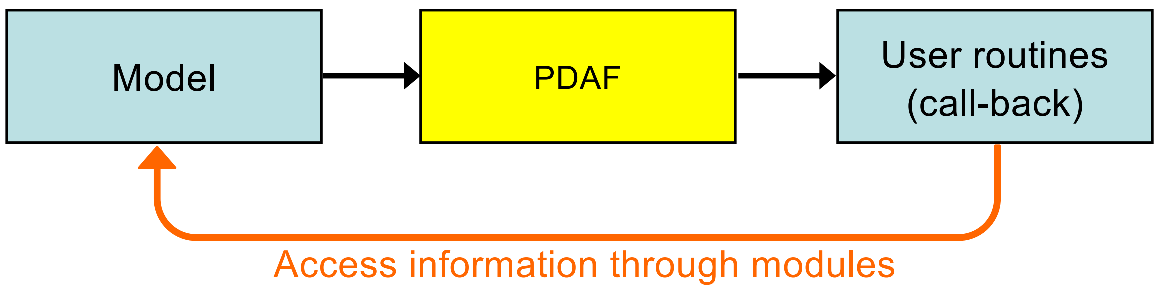
AWI

Framework solution with generic filter implementation

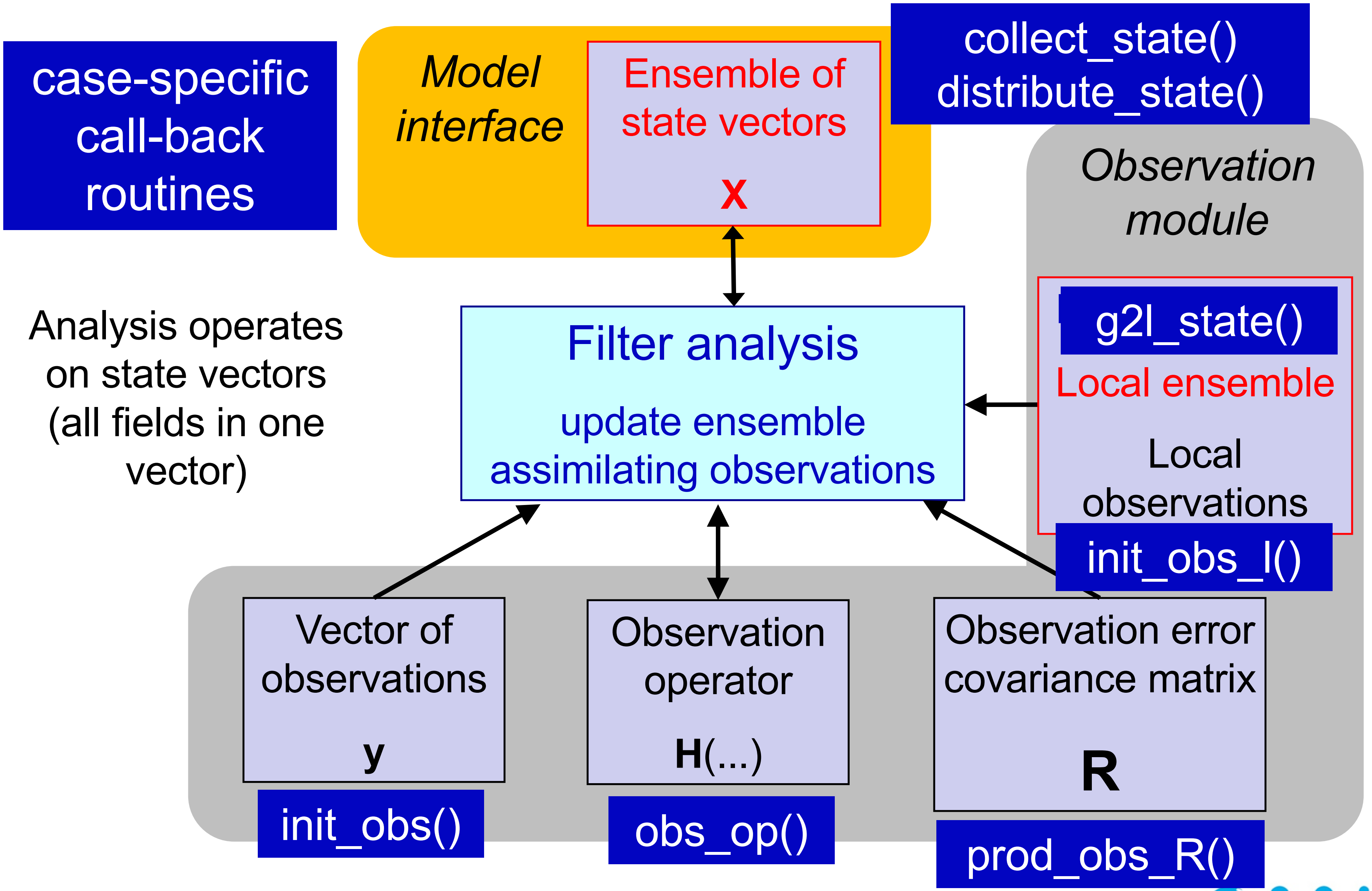


PDAF interface structure

- 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)



Ensemble Filter Analysis Step



Simple Subroutine Interfaces

Example: observation operator

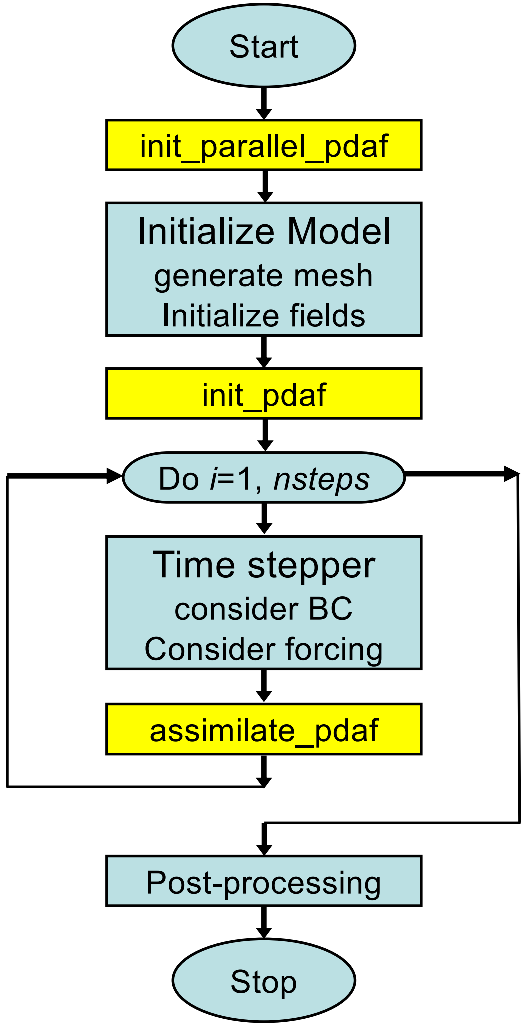
```
SUBROUTINE obs_op(step, dim, dim_obs, state, m_state)

  IMPLICIT NONE

! ARGUMENTS:
  INTEGER, INTENT(in) :: step           ! Current time step
  INTEGER, INTENT(in) :: dim            ! PE-local dimension of state
  INTEGER, INTENT(in) :: dim_obs        ! Dimension of observed state
  REAL, INTENT(in)    :: state(dim)            ! PE-local model state
  REAL, INTENT(inout) :: m_state(dim_obs) ! Observed state
```

Features of online program

- minimal changes to model code when combining model with filter algorithm
- model not required to be a subroutine
- no change to model numerics!
- model-sided control of assimilation program (user-supplied routines in model context)
- observation handling in model-context
- filter method encapsulated in subroutine
- complete parallelism in model, filter, and ensemble integrations

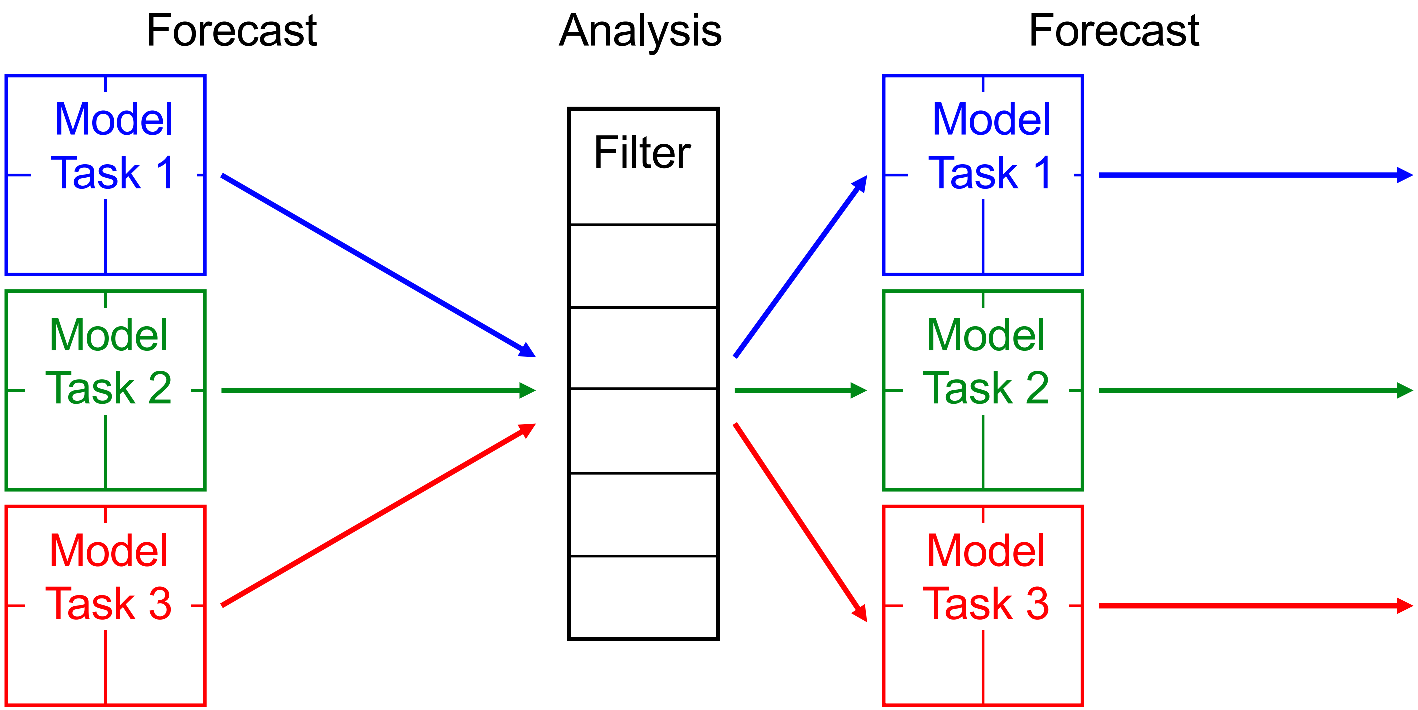


Building the Assimilation System

Problem reduces to:

1. Insert assimilation subroutine calls to model codes
2. Configuration of parallelization
   (MPI communicators)
3. Implementation of compartment-specific user routines
   and linking with model codes at compile time

2-level Parallelism

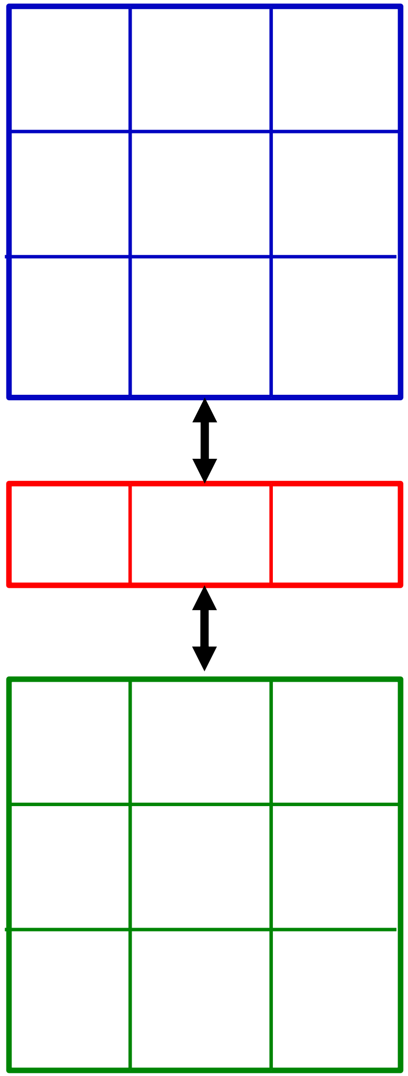


1. Multiple concurrent model tasks
2. Each model task can be parallelized

- Analysis step is also parallelized
- Configured by "*MPI Communicators*"

Example: Coupled ocean-atmosphere model

Atmosphere



Coupler

Ocean

Acess atmospheric model

- Fields, parameters and grid information

One might be able to access fields and grid information through coupler

Link atmosphere and ocean information using MPI

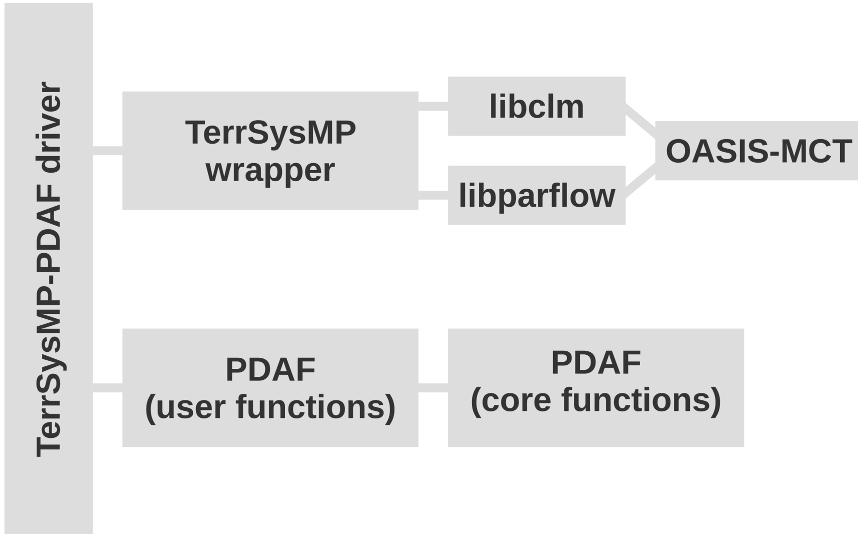
Acess ocean model

- Fields, parameters, and grid information

Example: TerrSysMP-PDAF (Kurtz et al. 2016)

TerrSysMP model

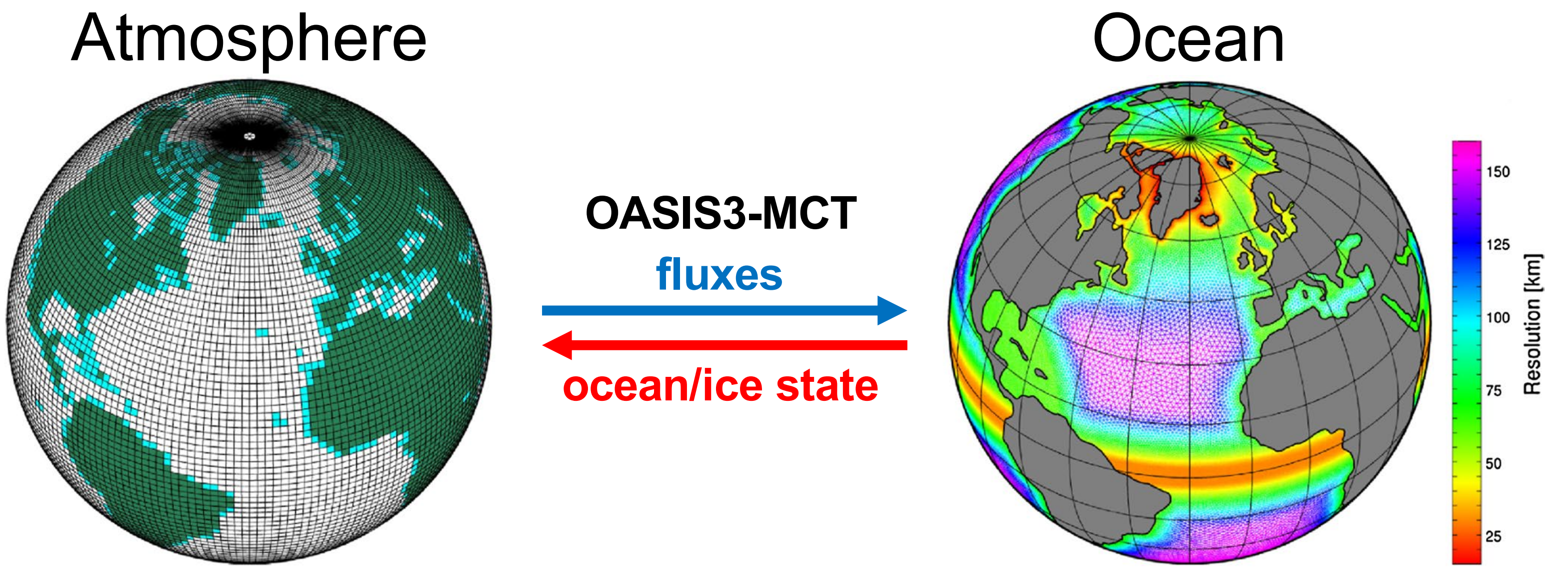
- Atmosphere: COSMO
- Land surface: CLM
- Subsurface: ParFlow
- coupled with PDAF using wrapper
- single executable
- driver controls program



- Tested using 65536 processor cores

W. Kurtz et al., Geosci. Model Dev. 9 (2016) 1341

Example: ECHAM6-FESOM



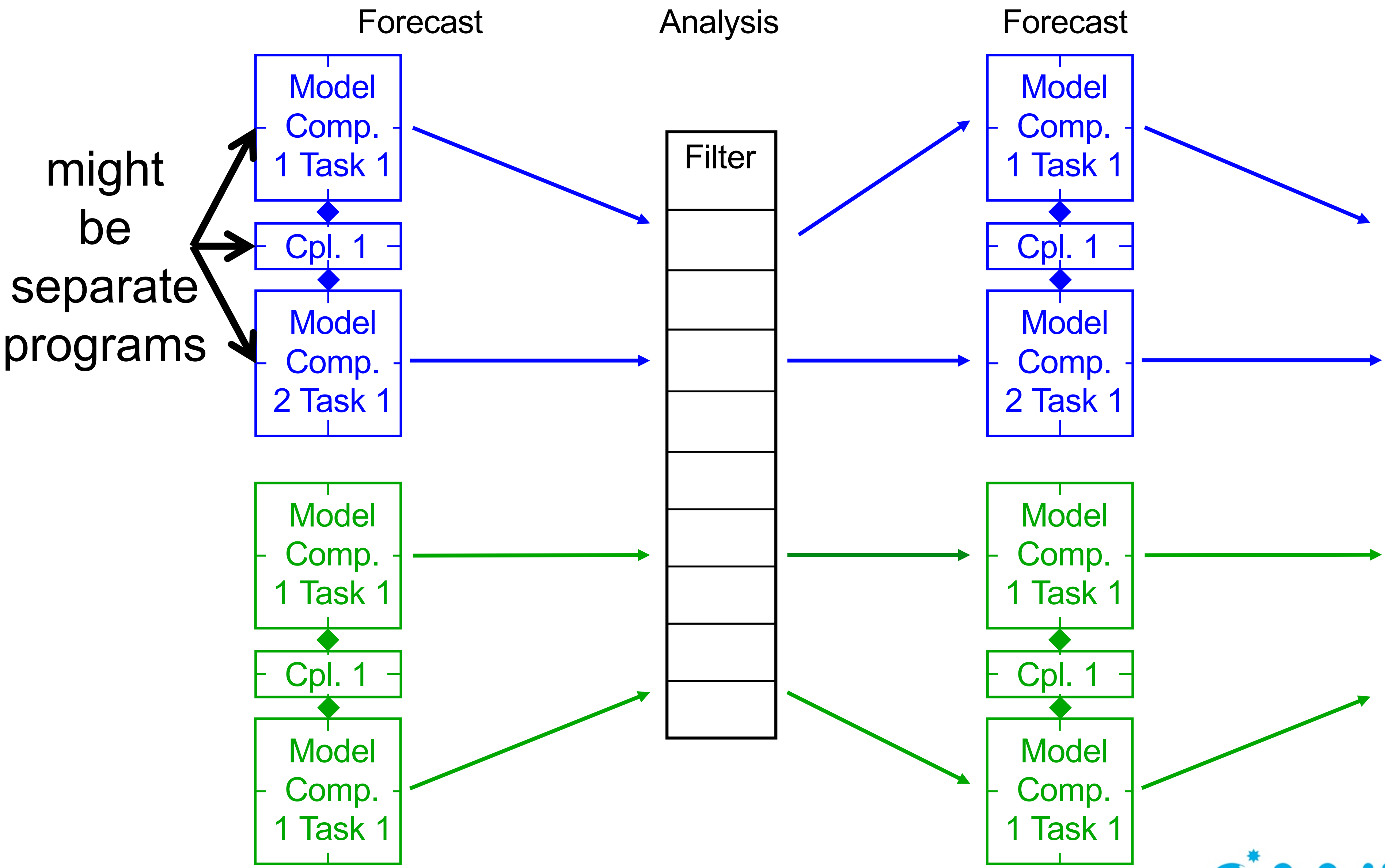
2 executables ECHAM and FESOM – do all coding twice

- add subroutine call into both models
- adapt model communicator (distinct names in the models)
- replace MPI_COMM_WORLD in communication routines for fluxes

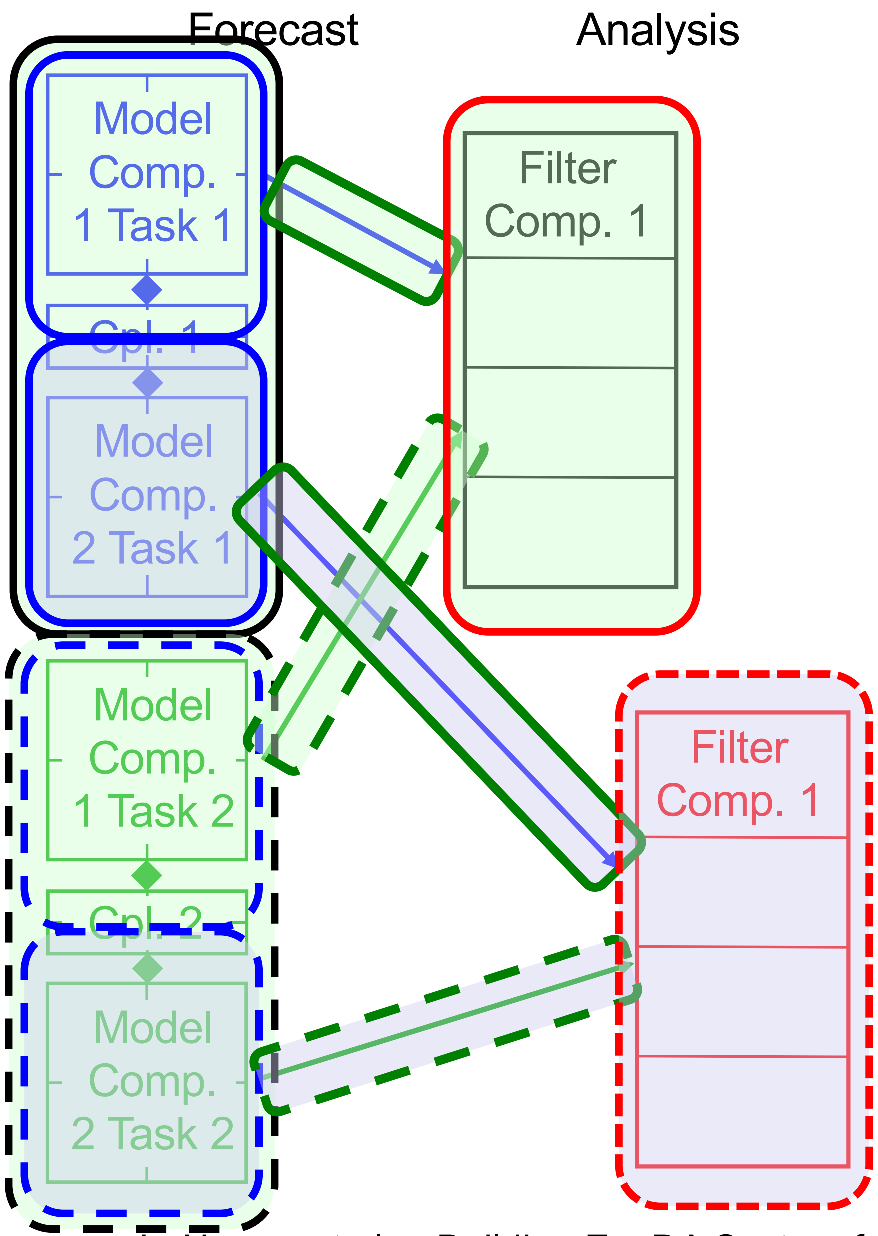
In OASIS-MCT library

- Replace MPI_COMM_WORLD in OASIS coupler
- Let each model task write files with interpolation information

2 compartment system – strongly coupled DA



Configure Parallelization – weakly coupled DA

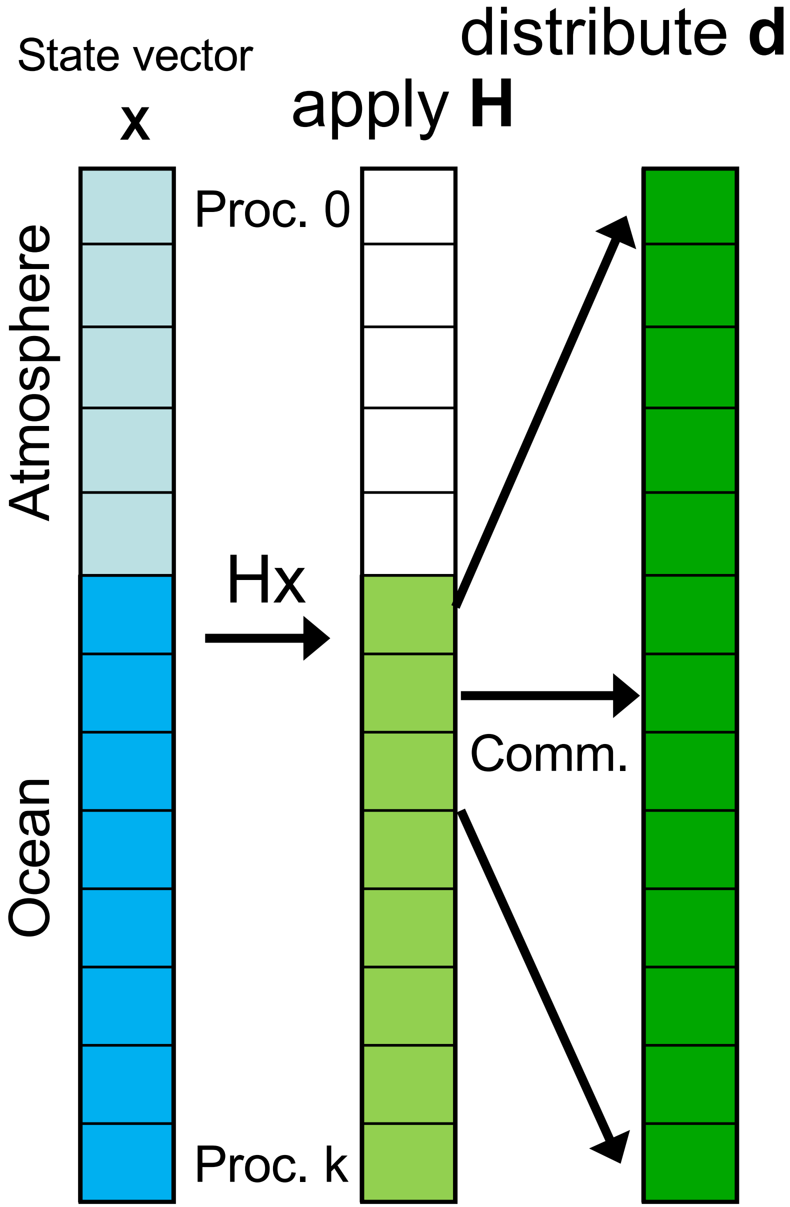


Logical decomposition:

- Communicator for each
  - Coupled model task
  - Compartment in each task (init by coupler)
  - (Coupler *might want to split MPI_COMM_WORLD*)
  - Filter for each compartment
  - Connection for collecting ensembles for filtering

- Different compartments
  - Initialize distinct assimilation parameters
  - Use distinct user routines

Strongly coupled: Parallelization of analysis step



We need innovation: **d = Hx - y**

Observation operator links different compartments

1. Compute part of **d** on process 'owning' the observation

2. Communicate **d** to processes for which observation is within localization radius

Assimilation of Sea Surface Temperature

- Daily assimilation of SST from Copernicus (L3 product)
  - Weakly coupled DA for year 2016
  - Assimilate into ocean compartment;
    atmosphere influence via model coupler

- Work in progress, but some insights
  - Initial model SST quite far away from observations (because there is no forcing)
  - High ensemble variance and difference to observations in Equatorial region (big assimilation corrections)
  - Sensitive at ice edge

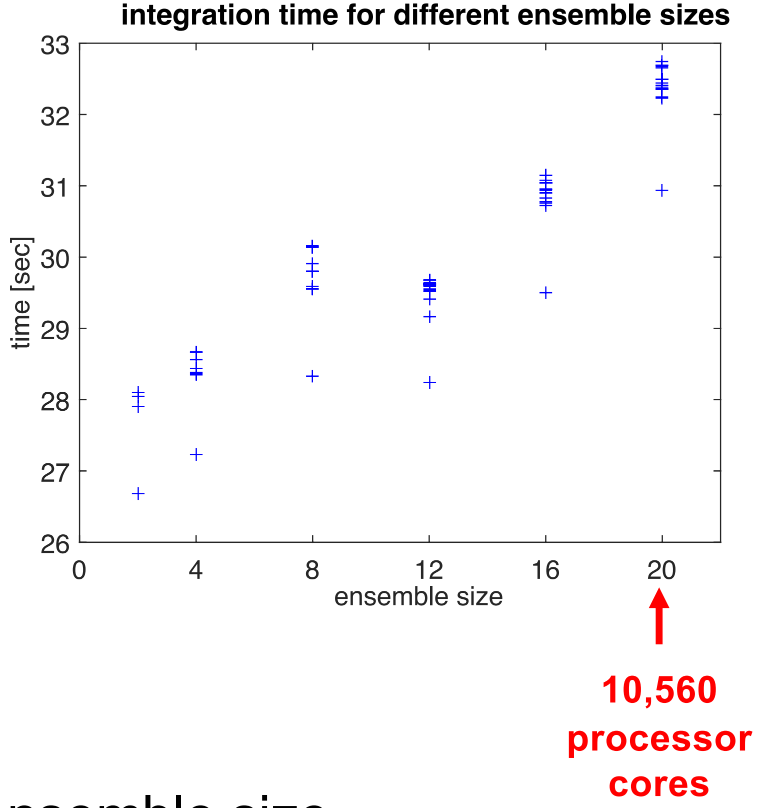
Execution times (weakly-coupled, DA only into ocean)

MPI-tasks

- ECHAM: 144
- FESOM: 384

Timings (1 day):

- Ens. forecast: 27 – 23 sec
- Analysis step: 0.5 – 0.9 sec

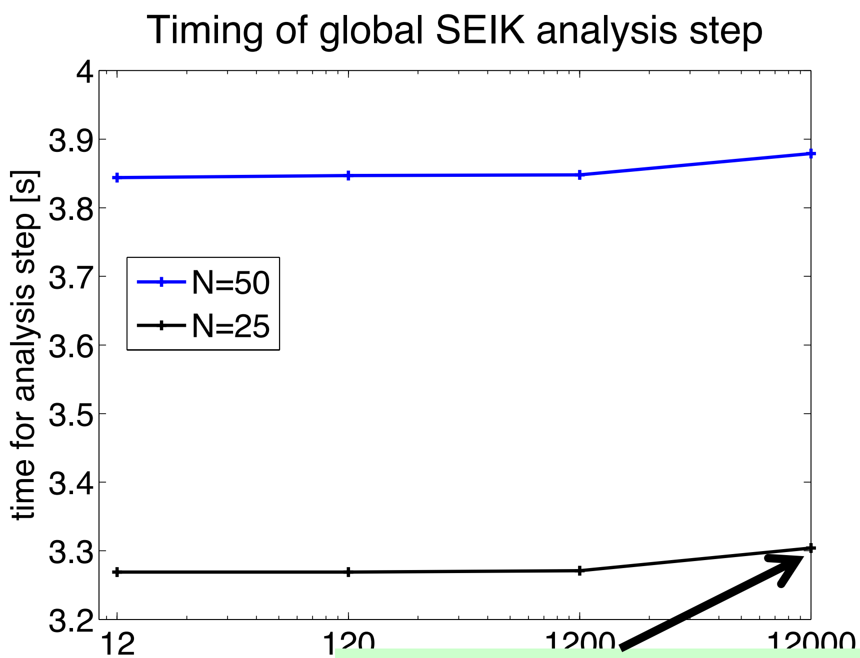


A remaining issue:

- Increasing integration time with growing ensemble size (only 16% due to more parallel communication)
- some variability in integration time over ensemble tasks
- Need optimal distribution of programs over compute nodes/racks (here set up as ocean/atmosphere pairs)

Very big test case

- 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)

Current algorithms in PDAF

PDAF originated from comparison studies of different filters

Filters and smoothers

- EnKF (Evensen, 1994 + perturbed obs.)
- ETKF (Bishop et al., 2001)
- SEIK filter (Pham et al., 1998)
- ESTKF (Nerger et al., 2012)
- NETF (Toedter & Ahrens, 2015)

Not yet released:

- serial EnSRF
- particle filter
- EWPF

All methods include

- global and localized versions
- smoothers

Model bindings

- MITgcm, Lorenz96

Not yet released:

- NEMO

PDAF features

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 (at least) any ensemble-based method)
- efficient for research and operational use

Future developments:

- Prepare model-specific routine packages (apart from MITgcm)
- Integrate more diagnostics
- Additional tools for observation handling
- Nonlinear filters
- Revision for Fortran 2003 standard

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

Summary

- AWI-CM/PDAF: Coupling completed;
  currently working on sea surface temperature assimilation
- Software framework simplifies building data assimilation systems
- Efficient online DA coupling; minimal model code changes
- Setup of data assimilation with coupled model
  1. Configuration of communicators
  2. Add routines for initialization & analysis step
  3. Implementation of case-specific user-routines
- Size of computing problem and communication layout might lead to tuning requirements

References

- http://pdaf.awi.de
- Nerger, L., Hiller, W. *Software for Ensemble-based DA Systems – Implementation and Scalability*. Computers and Geosciences 55 (2013) 110-118
- Nerger, L., Hiller, W., Schröter, J.(2005). *PDAF - The Parallel Data Assimilation Framework: Experiences with Kalman Filtering*, Proceedings of the Eleventh ECMWF Workshop on the Use of High Performance Computing in Meteorology, Reading, UK, 25 - 29 October 2004, pp. 63-83.

Thank you !