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

HELMHOLTZ
SPITZENFORSCHUNG FÜR
GROSSE HERAUSFORDERUNGEN

WCRP-WWRP Symposium on Data Assimilation and Reanalysis, online, September 13-18, 2021

PDAF – Parallel Data Assimilation Framework

A universal tool for ensemble data assimilation …

- provide support for parallel ensemble forecasts
- provide assimilation methods (solvers) - fully-implemented & parallelized
- provide tools for observation handling and for diagnostics
- easily useable with (probably) any numerical model
- a program library (PDAF-core) plus additional functions
- run from notebooks to supercomputers (Fortran, MPI & OpenMP)
- usable for real assimilation applications and to study assimilation methods
- open for community contributions

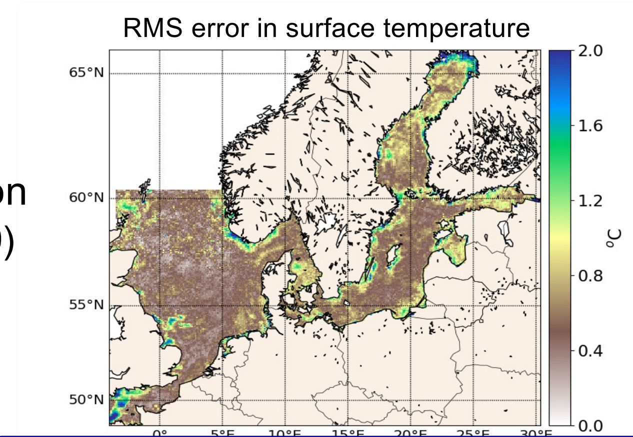
Open source:
Code, documentation, and tutorial available at

http://pdaf.awi.de

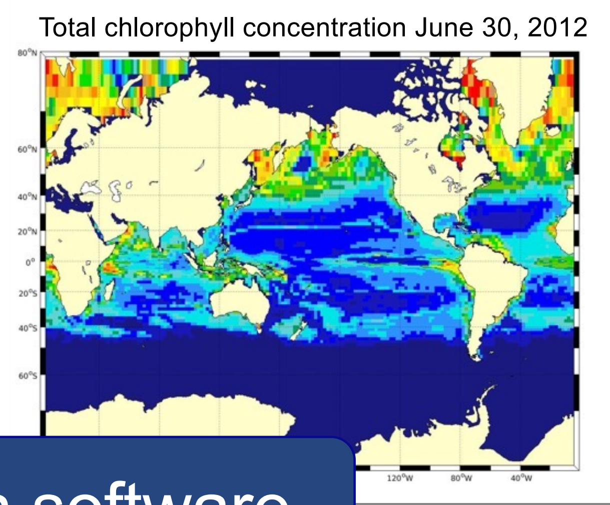
L. Nerger, W. Hiller, Computers & Geosciences 55 (2013) 110-118

PDAF Application Examples

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

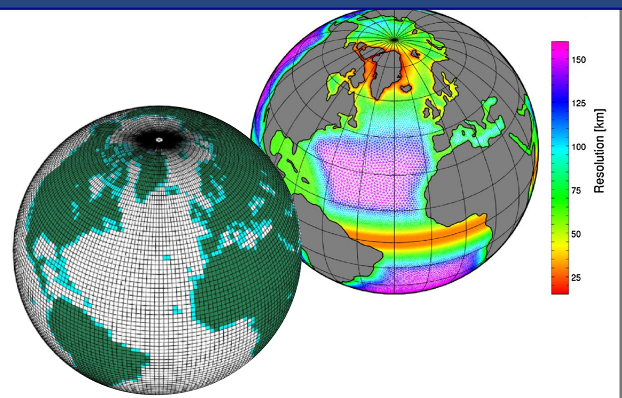


MITgcm-REcoM:
global ocean color
assimilation into
biogeochemical
model
(Pradhan et al., 2019/20)



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

PDAF Parallel Data Assimilation Framework

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

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

Assumption: Users know their model
➜ let users implement DA system in model context

For users, model is not just a time-stepping operator
➜ let users extend their model for data assimilation

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)

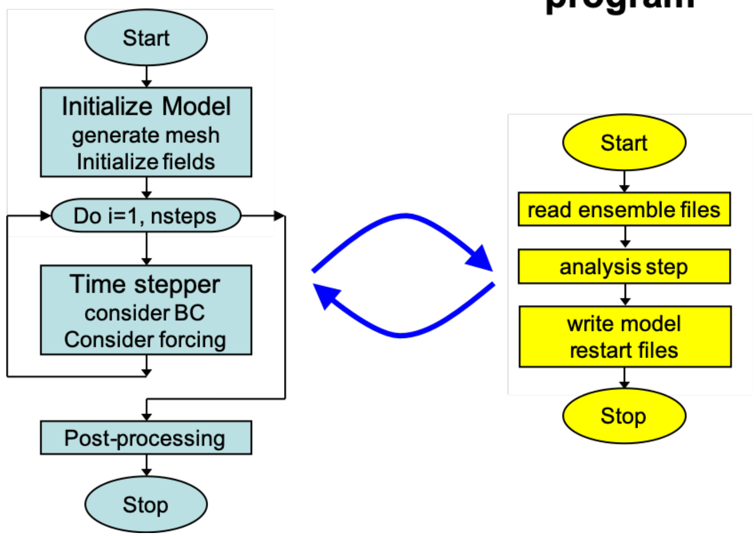
➜ Users directly implement case-specific routines
(no indirect description (XML, YAML, ...) of e.g. observation layout)

Coupling Model and Assimilation Code: 2 Variants

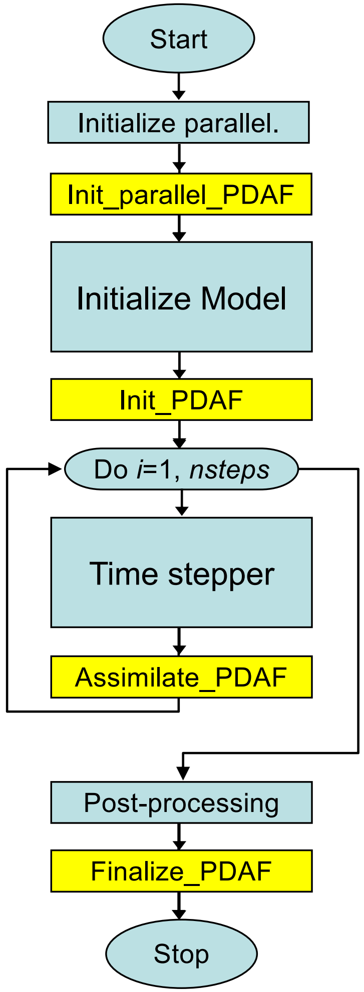
PDAF Parallel Data Assimilation Framework

Offline

Model **Assimilation program**



- Separate programs for model and DA
- Flexible to run
- Needs frequent model restarts and file output (less efficient than online coupling)



Online

- Augment model with DA functionality
- Insert 4 subroutine calls
- Very efficient & highly scalable

Model code	DA code

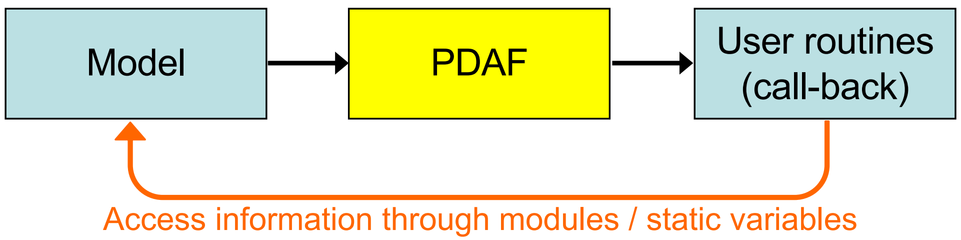
PDAF interface structure

- **Model-sided Interface:** Defined calls to PDAF routines
  (called by driver program for offline coupling)

- **Case-related Interface:** User-supplied call-back routines for elementary operations:
  - transfers between model fields and ensemble of state vectors
  - observation-related operations

- **Internal Interface:** Connect to data assimilation methods

- User supplied routines can be implemented as routines of the model and can share data with it (low abstraction level)

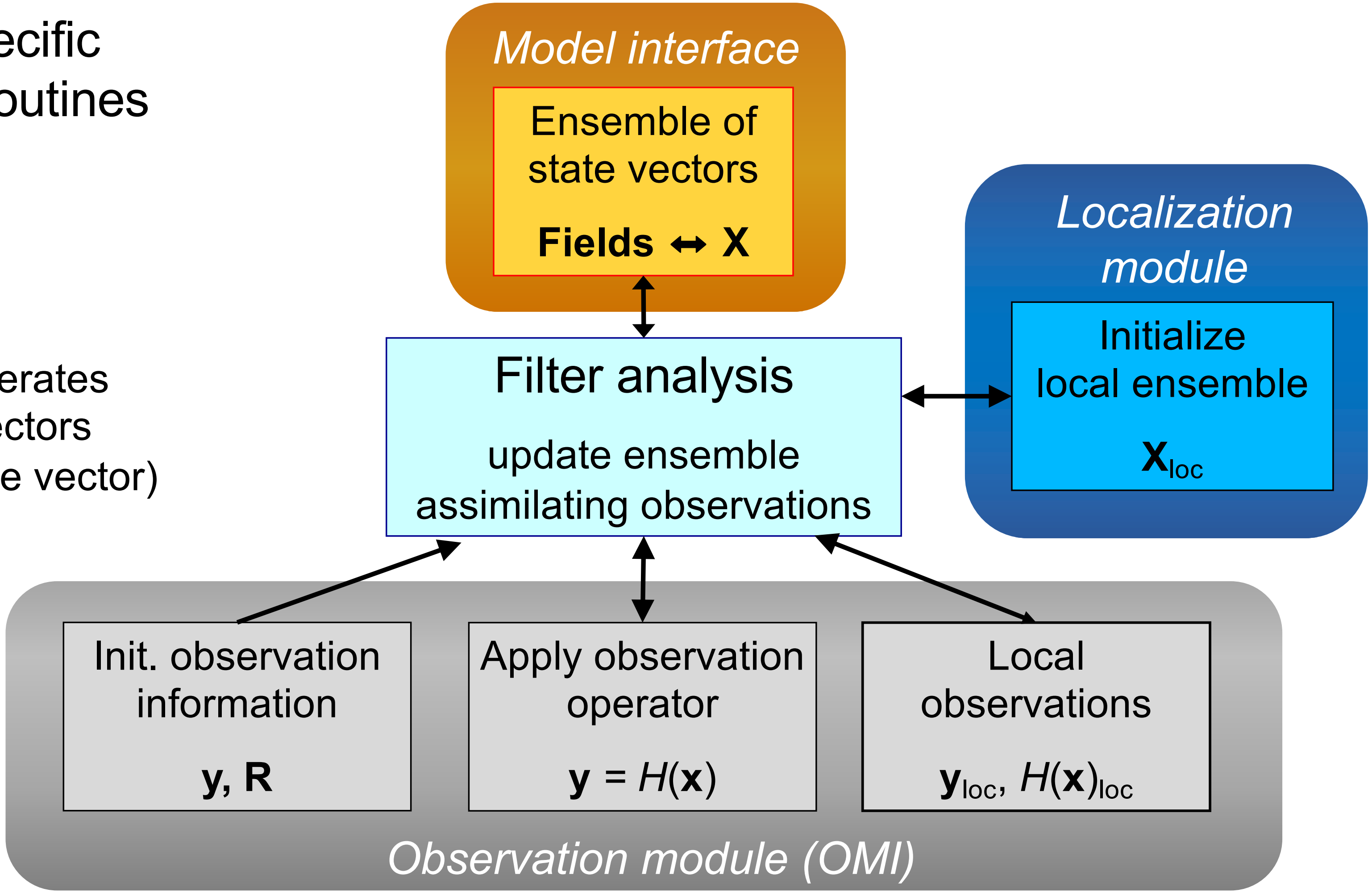


Implementing the Ensemble Analysis Step (Solver)

PDAF Parallel Data Assimilation Framework

case-specific call-back routines

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



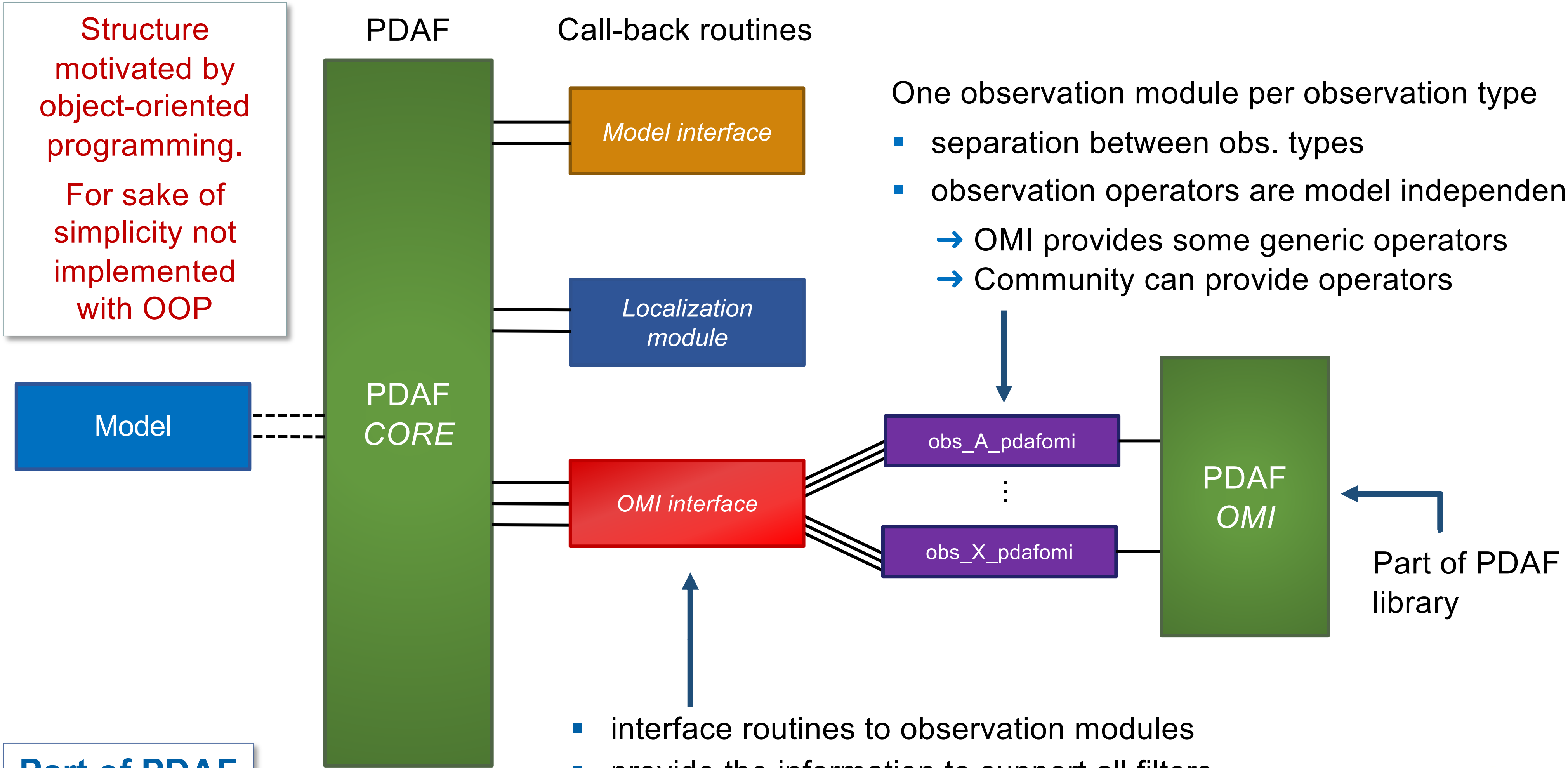
Recent and current developments

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 observation type

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

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

Part of PDAF V1.16

Strongly Coupled DA

Talk by Qi Tang et al.

PDAF Parallel Data Assimilation Framework

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

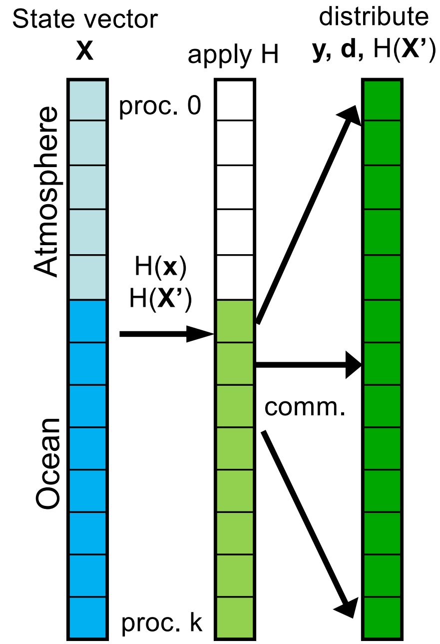
→ joint state vector decomposed over the processes

→ 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

- Some partners (PML, OGS) use 3D-Var and intent to step to EnVar
- Integrate in PDAF analogous to EnKFs/PFs
- Focus on infrastructure with optimizers as core
  → Future PDAF release

- Extension to Ensemble 4D-Var planned for later

www.seamlessproject.org

Services based on Ecosystem data AssiMiLation: Essential Science and Solutions

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/Hunt et al. 2007)
- 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 tomorrow)

Model bindings

- MITgcm
- AWI-CM / FESOM

Toy models

- Lorenz-96 / Lorenz-63

Community provided:

SCHISM/ESMF

TerrSysMP-PDAF

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

Upcoming:
- Lorenz-2005 II/III

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
   - Provide common DA infrastructure with generic components
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 can be implemented like model code
   - simple switching between different filters and data sets
   - *Separation of concerns:* model, DA methods, observations

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 excellent 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
- community code for DA methods and observations

Open source:
Code, documentation, and tutorial available at

http://pdaf.awi.de

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

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

PDAF Parallel Data Assimilation Framework

- 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

Extra Slides

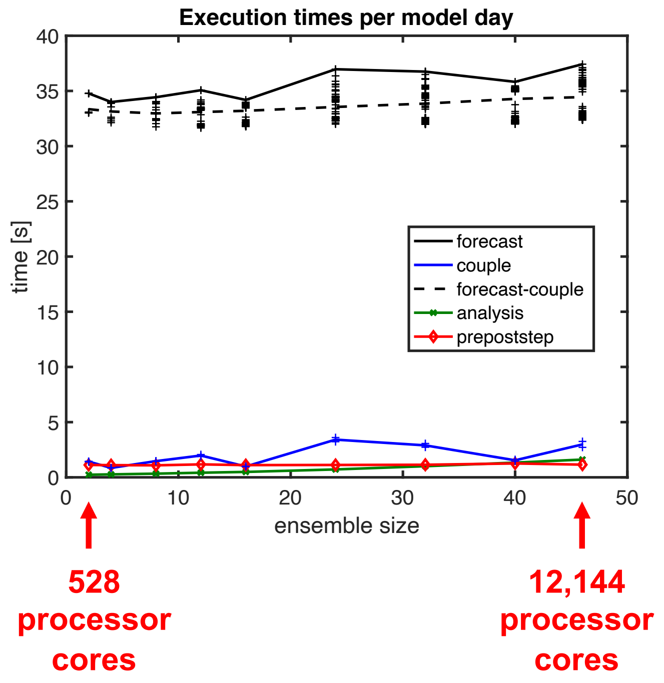
Execution times (weakly-coupled, DA only into ocean)

PDAF Parallel Data Assimilation Framework

MPI-tasks (each model instance)

- ECHAM: 72
- FESOM: 192
- Vary ensemble size

- Increasing integration time with growing ensemble size (11%; more parallel communication; worse placement)
- some variability in integration time over ensemble tasks



Important factors for good performance

- Need optimal distribution of programs over compute nodes/racks (here set up as ocean/atmosphere pairs)
- Avoid conflicts in IO (Best performance when each AWI-CM task runs in separate directory)

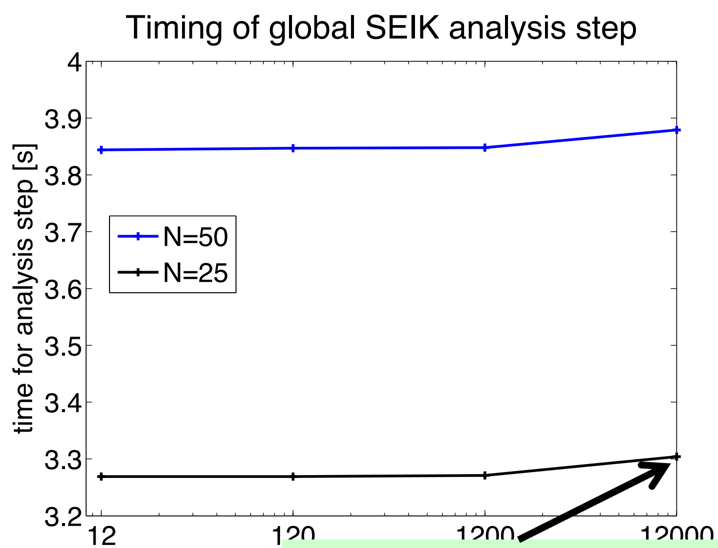
Nerger et al., GMD (2020), doi:10.5194/gmd-13-4305-2020

AWI

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)
- 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

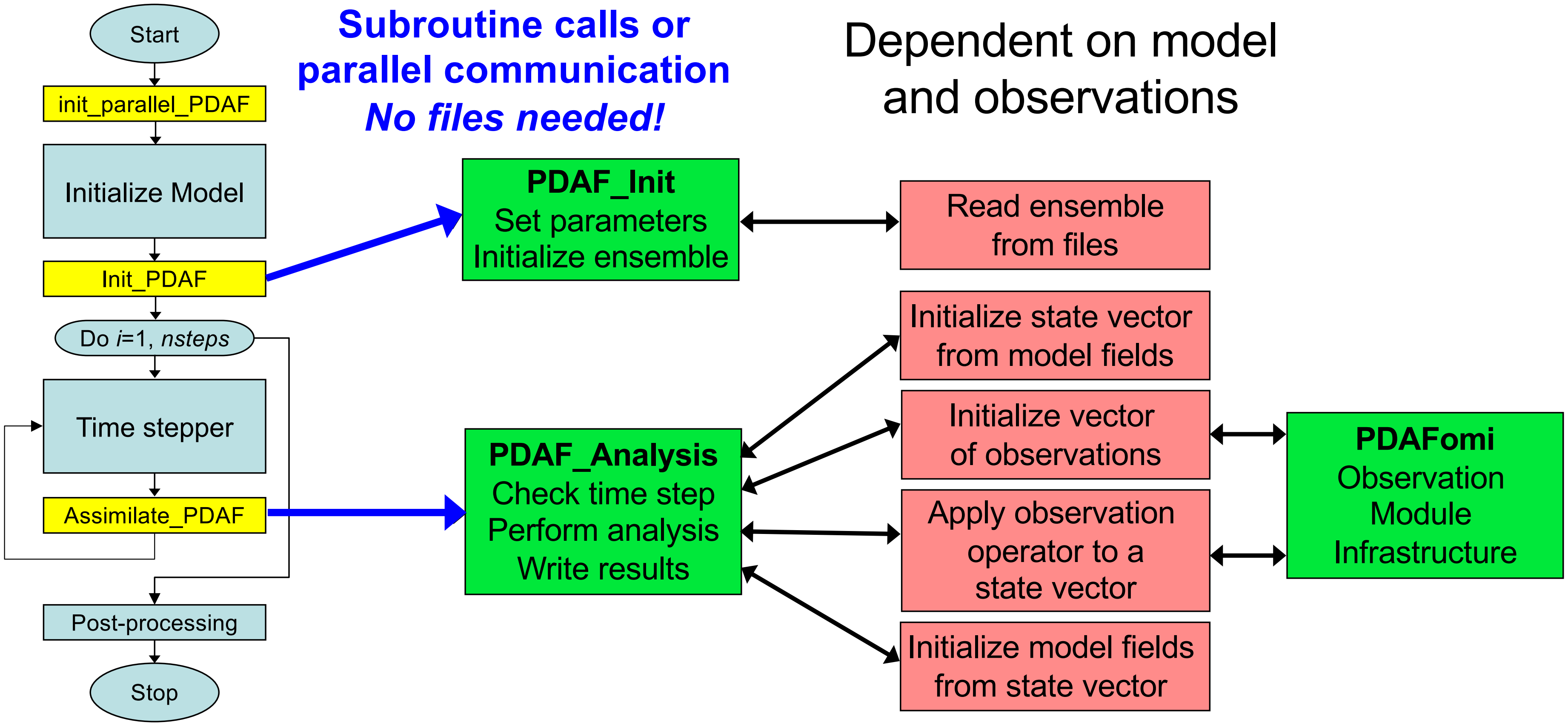


State dimension: 1.2e11

Observation dimension: 2.4e9

Framework solution with generic filter implementation

PDAF Parallel Data Assimilation Framework



Model with assimilation extension

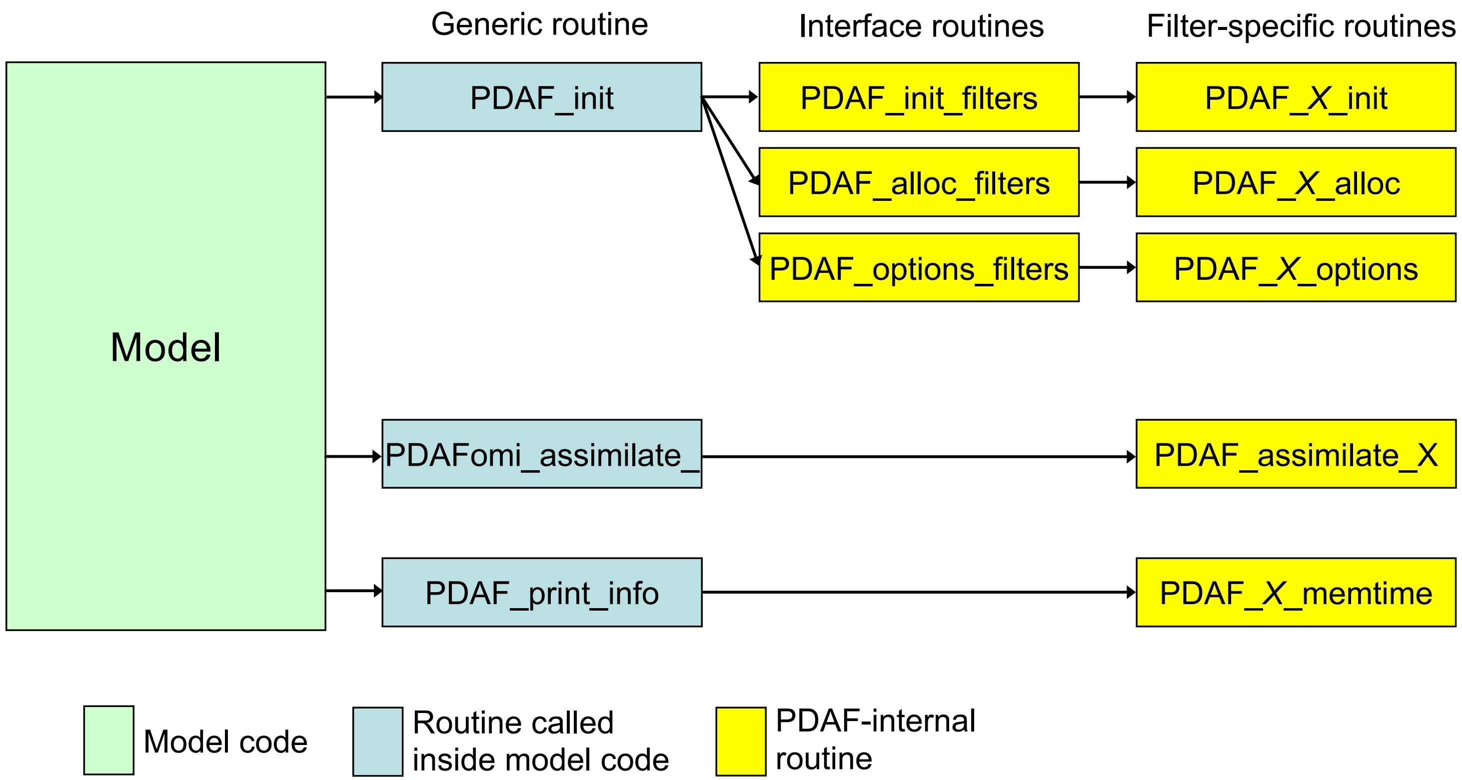
Core-routines of assimilation framework

Case specific call-back routines

Part of PDAF library

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)



Online and Offline Coupling - Efficiency

PDAF Parallel Data Assimilation Framework

Offline-coupling is simple to implement
but can be very ineffcient

Example:
Timing from atmosphere-ocean coupled model (AWI-CM) with daily analysis step:

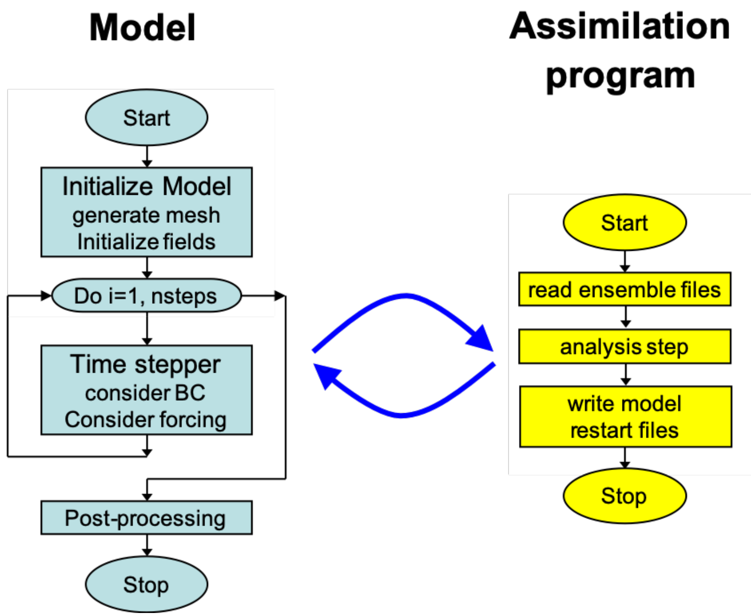
Model startup: 95 s
Integrate 1 day: 33 s
Model postprocessing: 14 s

overhead

Analysis step: 1 s

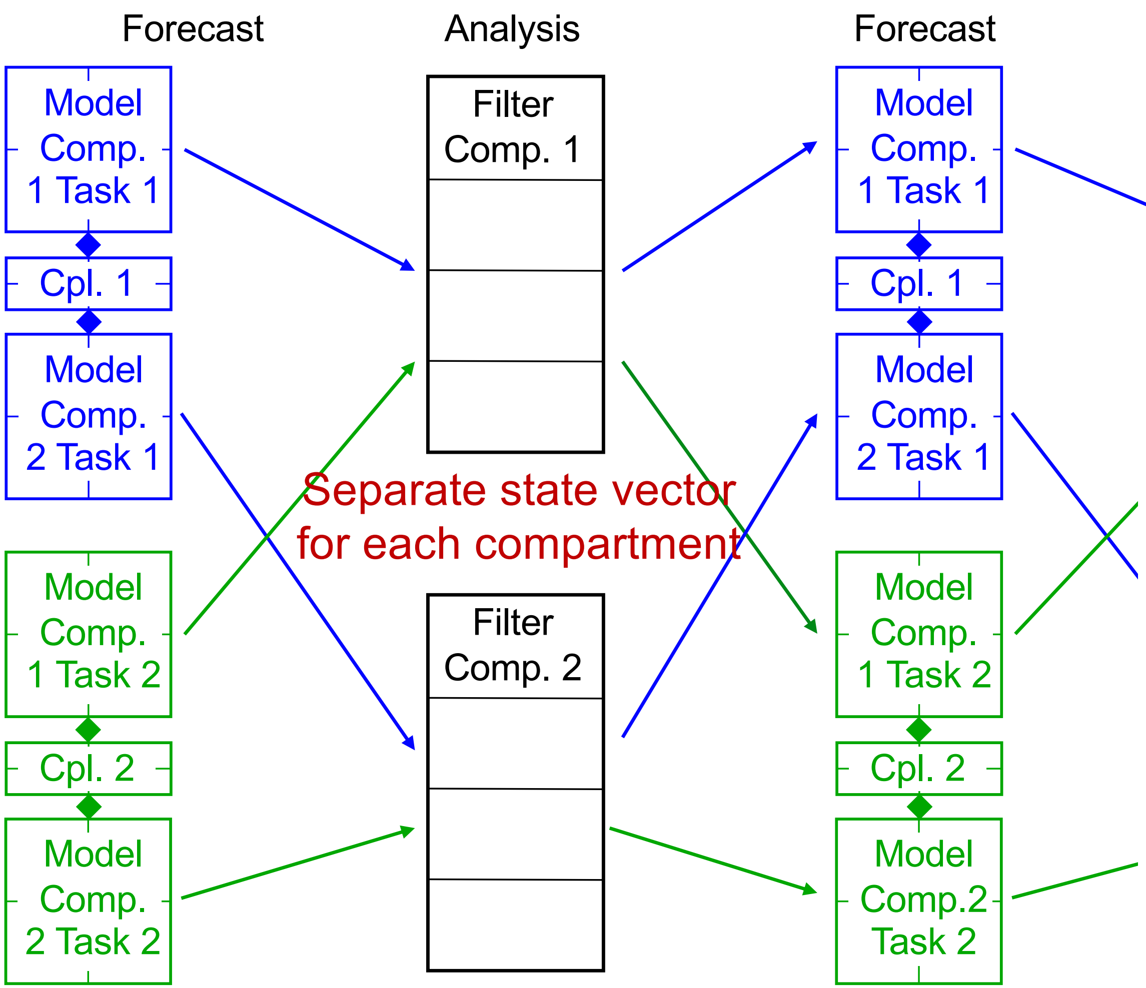
Restarting this model is ~3.5 times more expensive than integrating 1 day

→ avoid this for data assimilation



2 compartment system – weakly coupled DA

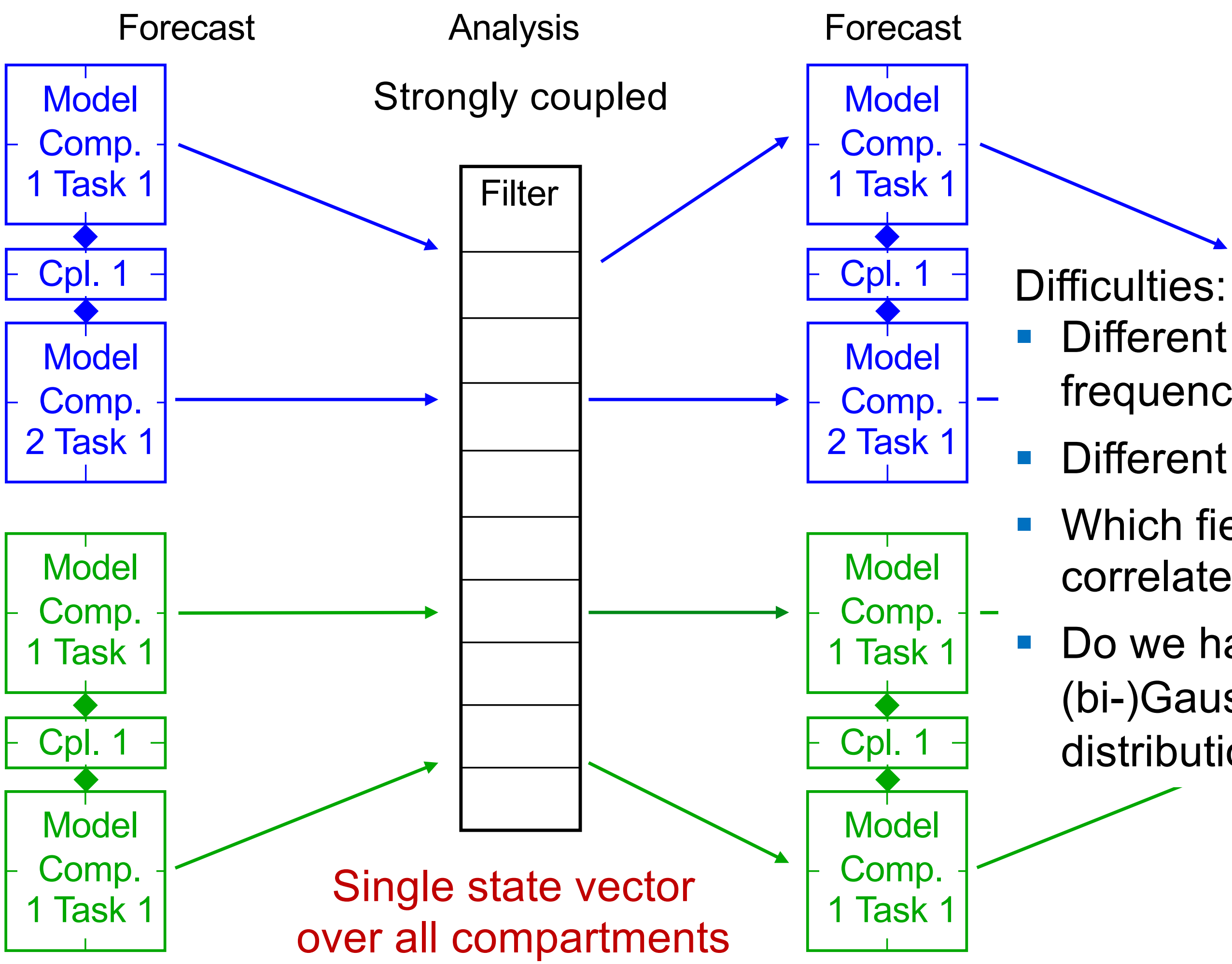
PDAF Parallel Data Assimilation Framework



- Simpler setup than strongly coupled
- Different DA methods possible
- Different timing of DA possible
- But:
  Fields in different compartments can be inconsistent

2 compartment system – strongly coupled DA

PDAF Parallel Data Assimilation Framework



Switch between weakly coupled and strongly coupled assimilation by change in MPI configuration

Difficulties:

- Different assimilation frequency
- Different time scales
- Which fields are correlated?
- Do we have (bi-)Gaussian distributions?