VORtech BV
COMPUTING AND SIMULATION SOFTWARE TECHNOLOGY

Generic parallelization strategies for data assimilation

OpenDA

Deltares
Enabling Delta Life

TU Delft

Nils van Velzen, Martin Verlaan

06-10-11

Outline

- OpenDA
- Coupling of model to DA method
- Parallelism of model and EnKF
- Parallelism for black box models
- Conclusions

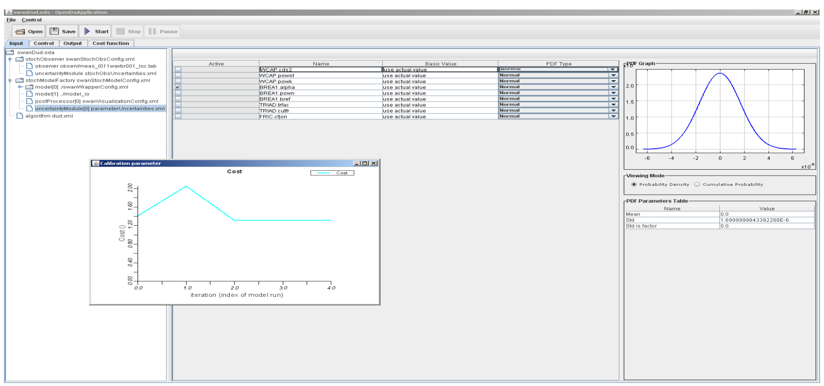
OpenDA

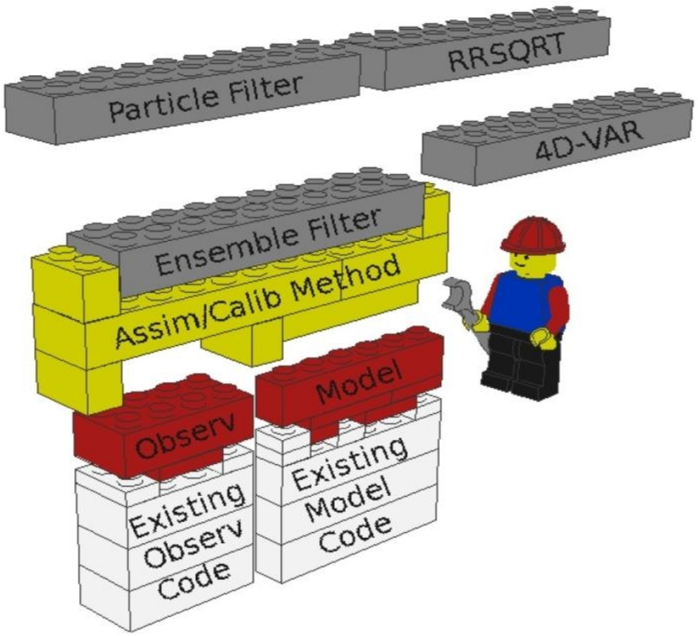
- What is OpenDA?
  - A generic toolbox for data-assimilation
  - Set of interfaces that define interactions between components
  - Library of data-assimilation algorithms
  - Open source

- Why OpenDA?
  - More efficient than development for each application
  - Shared knowledge between applications
  - Development of algorithms with e.g. universities
  - Easier to test

OpenDA

- Object oriented design
  - Classes, software building blocks
  - Interface (set of functions suitable for all models, observations, etc)



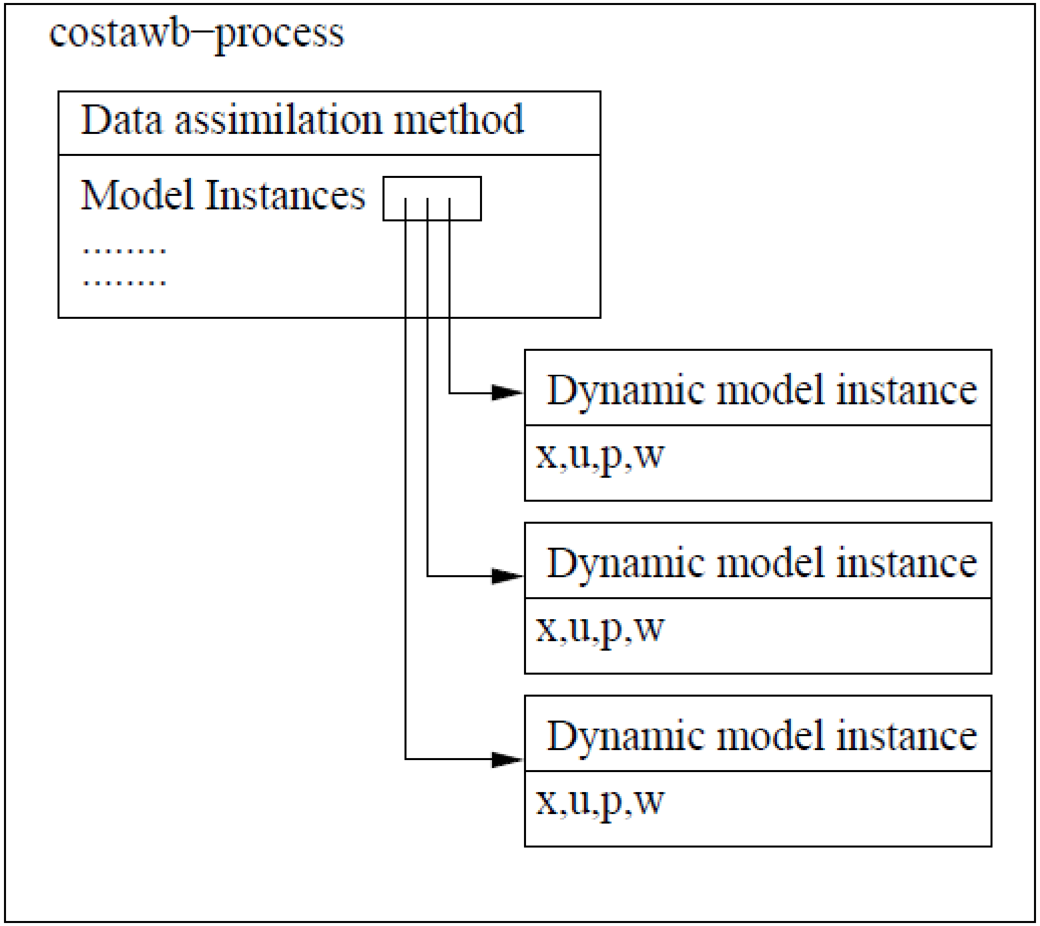


OpenDA

- Formal form of a model
  $$\frac{dx}{dt} = M\left(x(t), u(t), p, w(t)\right)$$
- State of model instance $x(t), u(t), p, w(t)$
- Instance state cannot be directly changed only through the methods like:

  GetState, AxpyState, Compute...
- Algorithm has no knowledge on model internals

OpenDA



costawb−process
Data assimilation method
Model Instances
....
....
Dynamic model instance
x,u,p,w
Dynamic model instance
x,u,p,w
Dynamic model instance
x,u,p,w

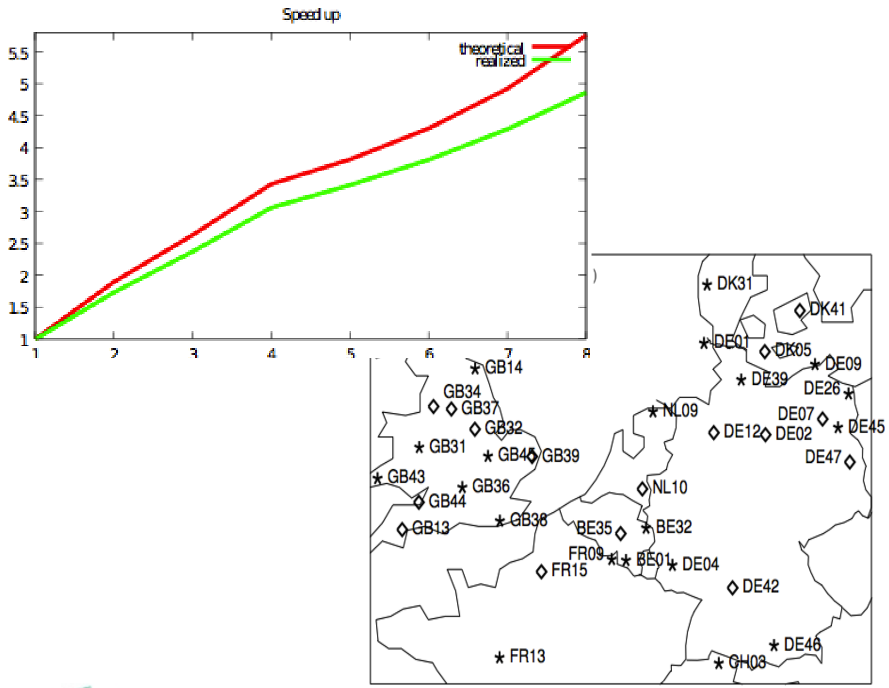
EnKF semi parallel

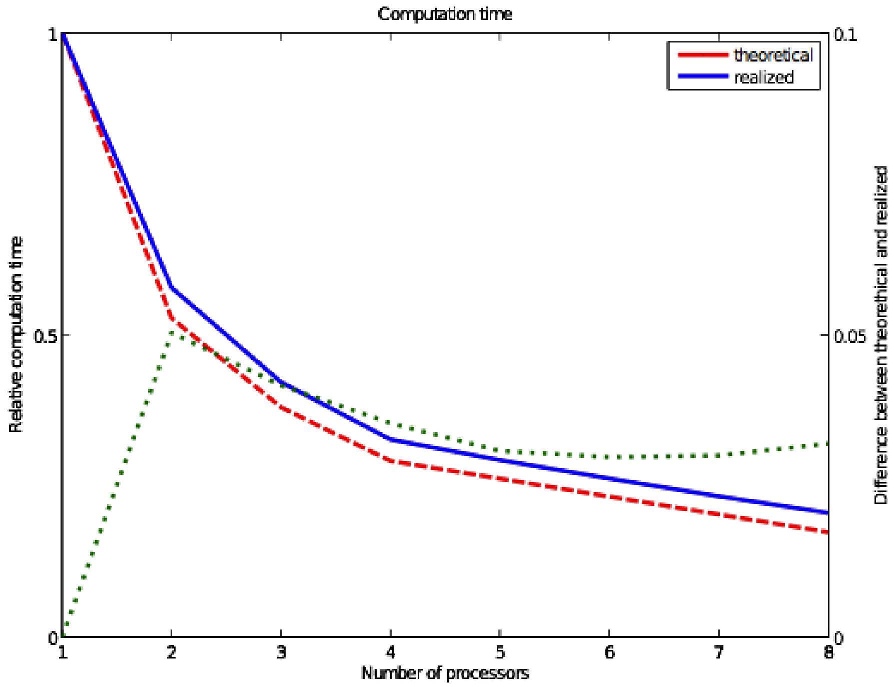
- Only parallelize model steps:
  - Not scalable, often sufficient

Equation	Semi parallel Communication volume
$\xi_i^f(t_k) = M(\xi_i^a(t_{k-1})) + w_i(t_k)$	$C_m + nN$
$x^f(t_k) = \frac{1}{N} \xi_i^f(t_k)$ $E^f(t_k) = [\xi_1^f(t_k) - x^f(t_k), \xi_2^f(t_k) - x^f(t_k), \ldots, \xi_N^f(t_k) - x^f(t_k)]$	0
$\vdots$ $\xi_i^a(t_k) = \xi_i^f(t_k) + K(t_k)[y(t_k) - H(t_k)\xi_i^f(t_k) + v_i(t_k)]$	nN

EnKF semi parallel

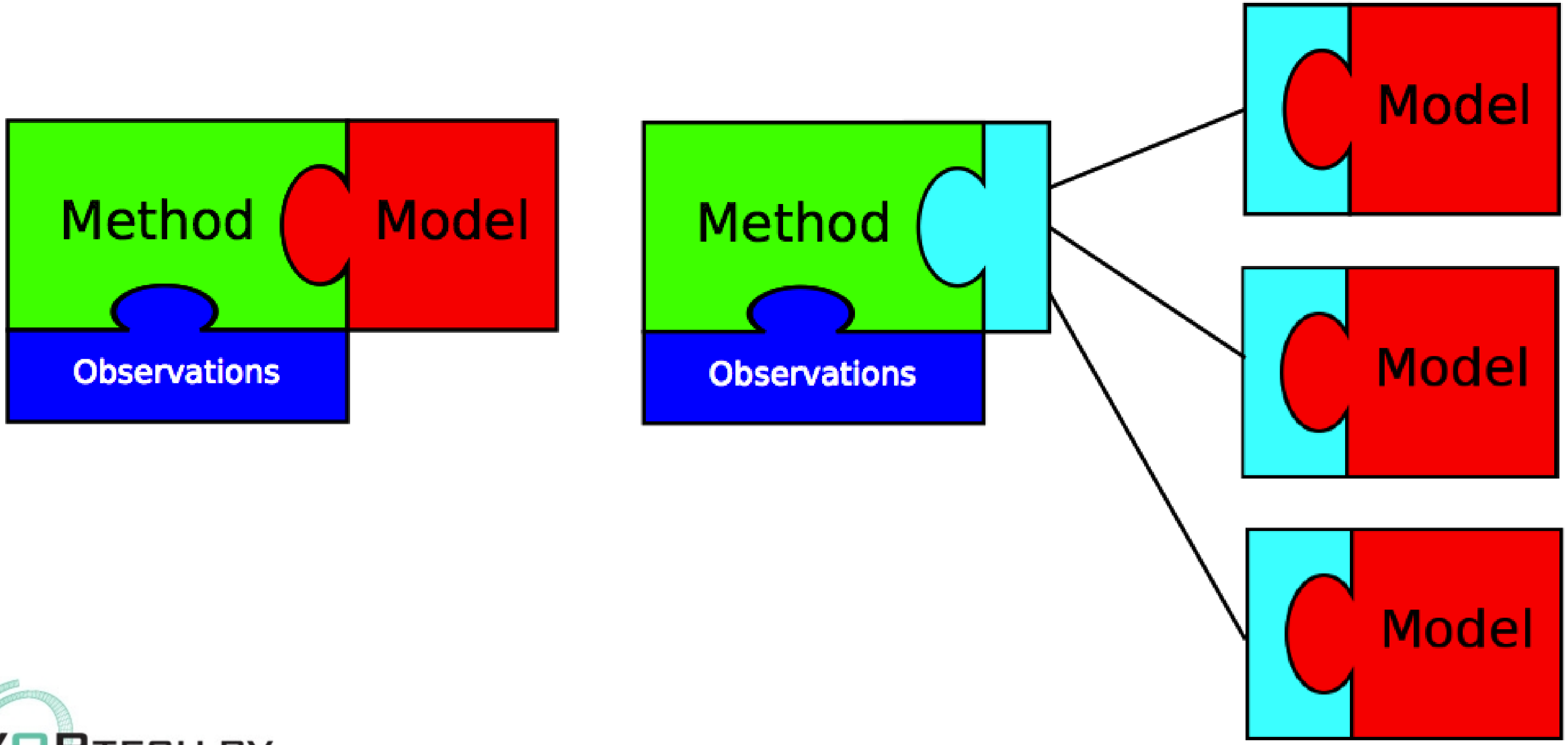
- Lotos-euros air quality model





EnKF semi parallel

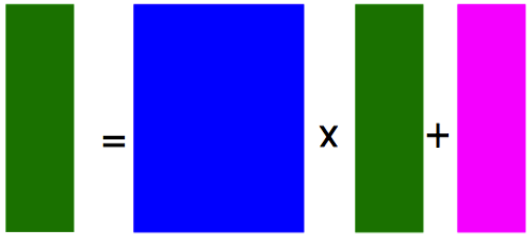
- Generic semi parallel due to OO concepts



EnKF parallel

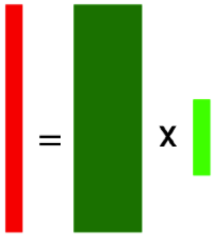
- Stochastic model time steps

$$\xi_i^f(t_k) = M\left(\xi_i^a(t_{k-1})\right) + w_i(t_k)$$



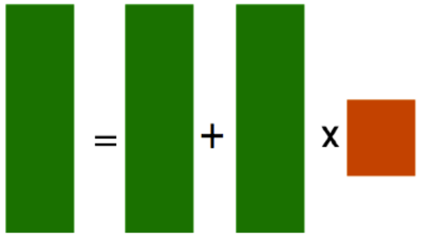
- Mean of Ensemble

$$x^f(t_k) = \frac{1}{N}\xi_i^f(t_k)$$



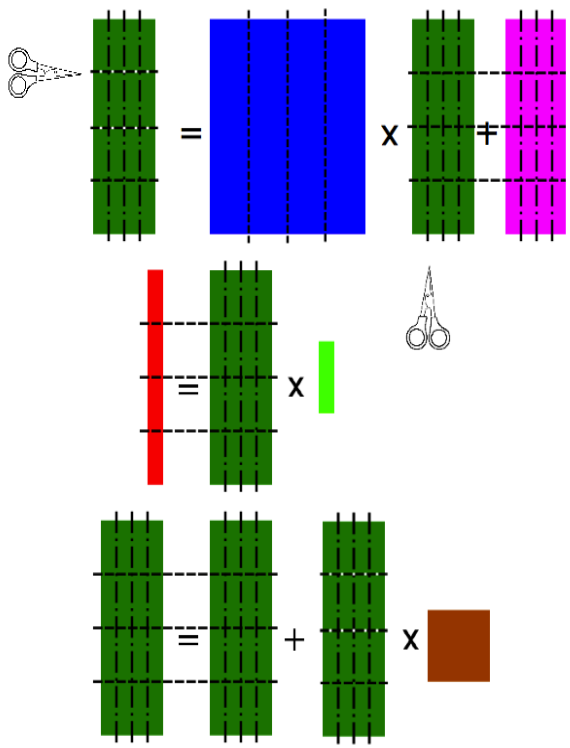
- Update (for all ensemble members)

$$\xi_i^a(t_k) = \xi_i^f(t_k) + K(t_k)\left[y(t_k) - H(t_k)\xi_i^f(t_k) + v_i(t_k)\right]$$



EnKF parallel

Column wise distribution	
Separate	Combined
$\frac{nN}{p}$	0
$n\log_2(p)$	$n\log_2(p)$
$\frac{nN}{p}+nN\log_2(p-1)$	$nN\log_2(p-1)$



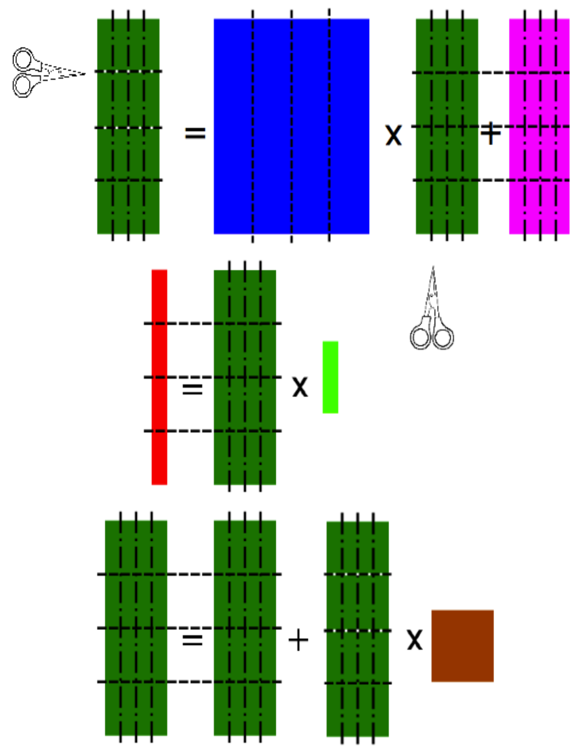
Row wise distribution	
Separate	Combined
$C_m+\frac{nN}{p}$	C_m
0	0
$\frac{nN}{p}$	0

VORTECH BV
COMPUTING AND SIMULATION SOFTWARE TECHNOLOGY

EnKF parallel

OpenDA

Column wise distribution	
Separate	Combined
$\frac{n\,N}{p}$	0
$n\log_2(p)$	$n\log_2(p)$
$\frac{n\,N}{p}+n\,N\log_2(p-1)$	$n\,N\log_2(p-1)$

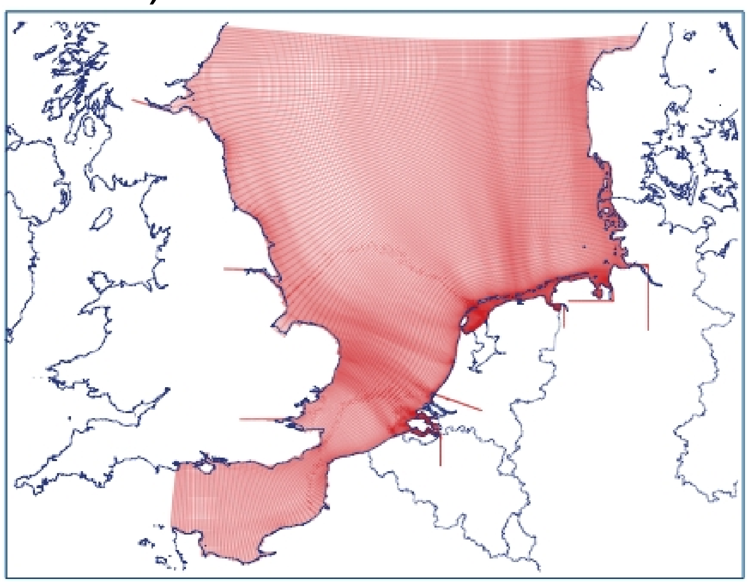


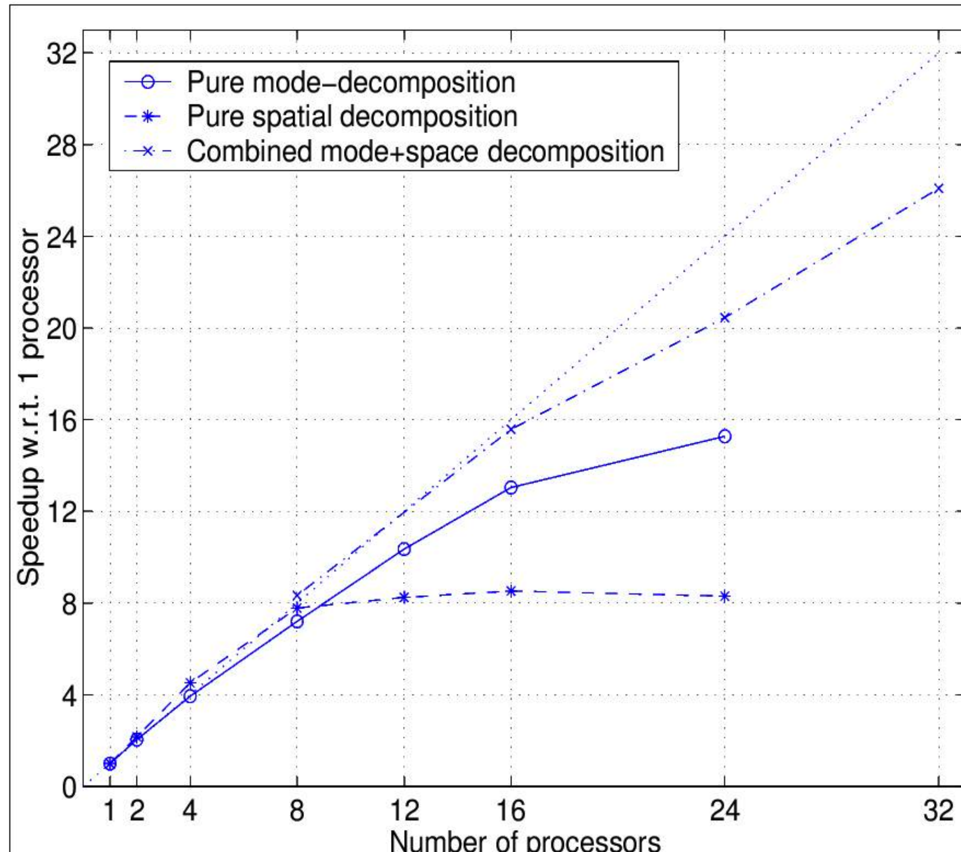
Row wise distribution	
Separate	Combined
$C_m+\frac{n\,N}{p}$	C_m
0	0
$\frac{n\,N}{p}$	0

VORTECH BV
COMPUTING AND SIMULATION SOFTWARE TECHNOLOGY

EnKF parallel

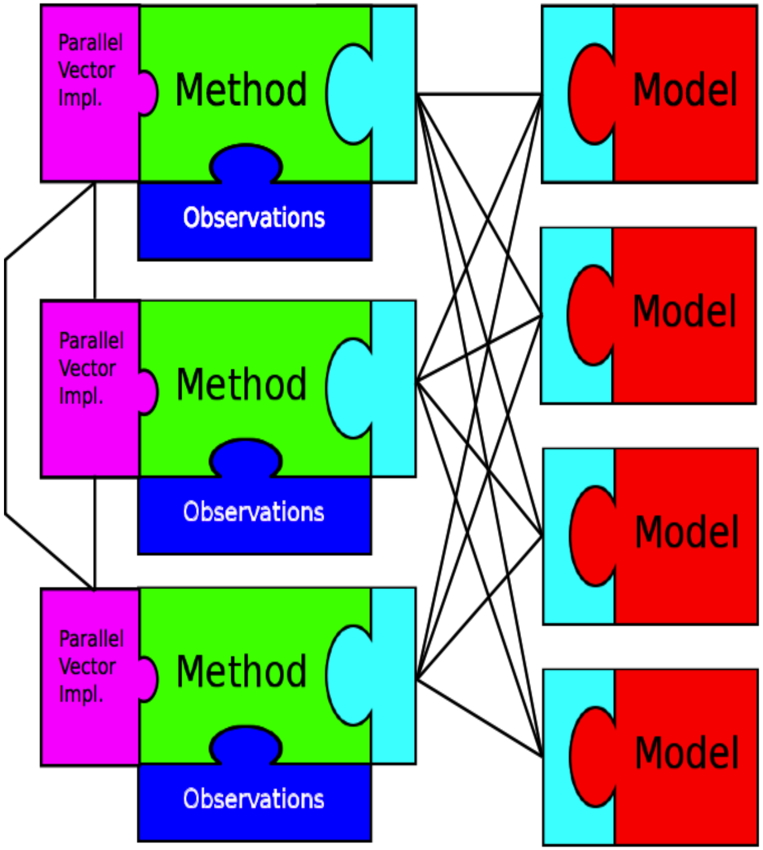
- WAQUA shallow water model
- Comparison of parallelization strategies for RRSQRT (Roest et al.)





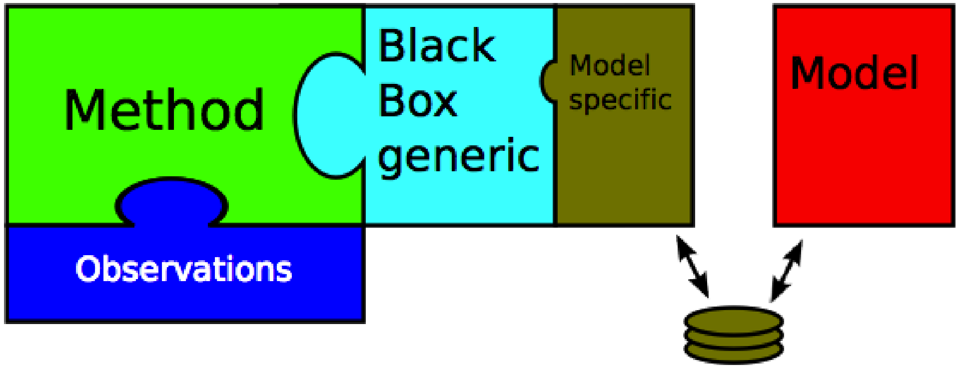
EnKF parallel

- Full parallel using OO concepts:
  - All filters run same code
  - State vectors are distributed (parallel vector)
  - Operations on parallel vectors by a parallel vector implementation
  - NO need to change the model and filter code
  - All complexities hidden in generic support layers/implementations



Parallel computing with black box models

- Black box models
  - No change to model code
  - Data exchange using files
  - Note: disk can be slow/more data written than needed

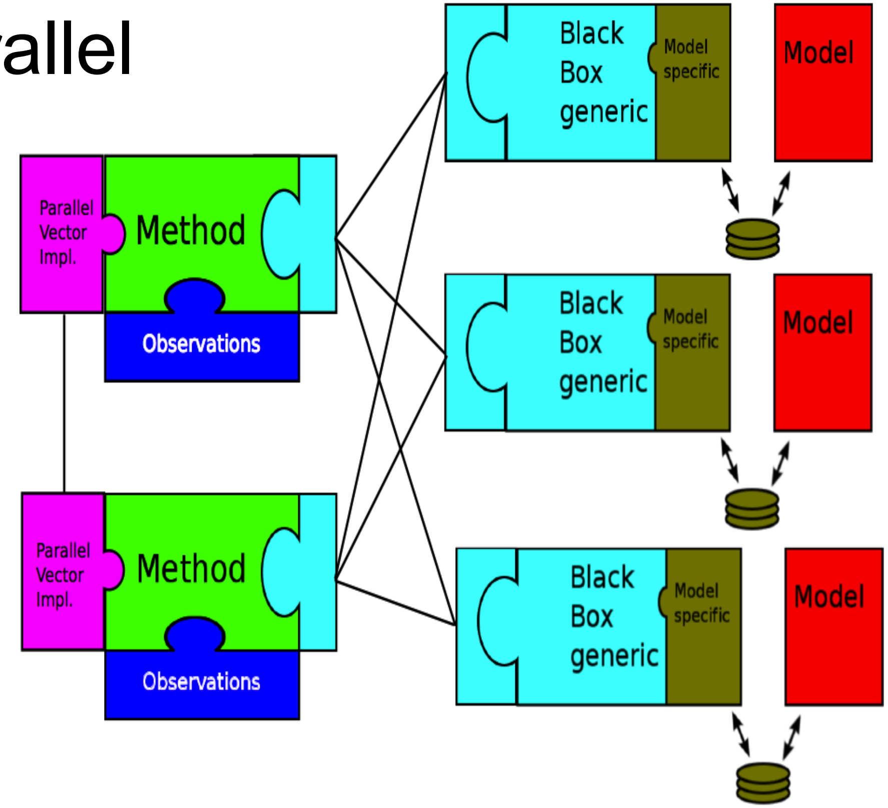


Parallel computing with black box models

- Swan model for wind generated waves
  - Operational model + DA for the north sea
  - Black box model
  - 1 hour, 8 cpu's 10 min
    50% IO
  - EnKF implementation (?)
  - Parallelization of filter

Parallel computing with black box models

- Black box model is normal model
  - Semi parallel+full parallel
  - Note disk speed,
  - Use local disks:
    - Faster
    - No sequential bottleneck



Conclusions

- Generic parallelization stragegies due to object oriented programming concepts.
- Single filter implementation for sequential as parallel computing
- EnKF like algorithms need combination of parallel strategies
- Black box models and IO can be parallelized as well