Centro Campano per il Monitoraggio e la Modellistica Marina e Atmosferica

Performance assessment of the Incremental Strong Constraint 4DVAR DA Algorithm in ROMS

Arcucci R.[1], L. D'Amore[1], Y. Li[2], A. Moore[3], L. Phillipson[2], R. Toumi[2] and **R. Montella[4,5]**

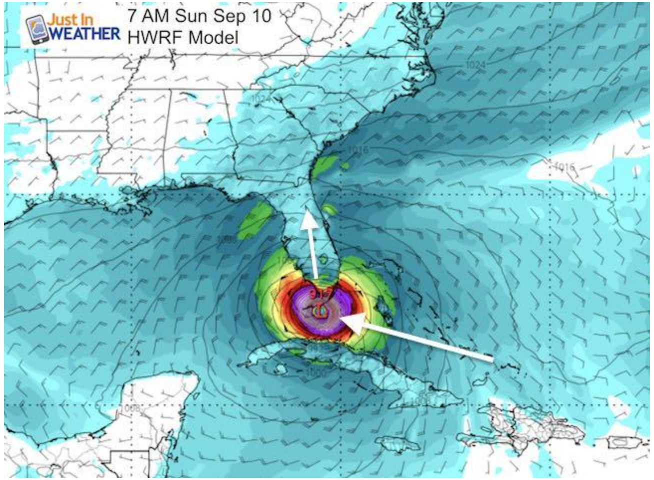
[1]Department of Mathematics and Applications, University of Napoli Federico II
[2]Imperial College of London
[3]University of California in Santa Cruz
[4]Department of Science and Technologies, University of Napoli Parthenope
[5]Campania Region Center for Marine and Atmosphere Monitoring and Modelling

This research has received funding from European Commission under H2020-MSCA-RISE-NASDAC project (grant n. 691184)

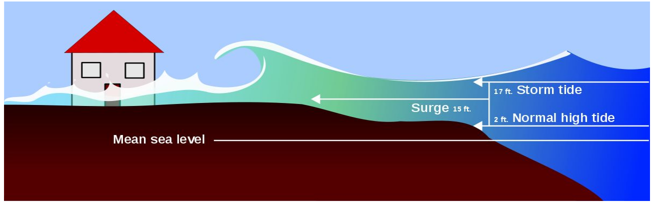
Introduction, contextualization and motivations

- Weather forecast rises dramatically at common people level.
- Modeling atmosphere is supported by many custom and community supported software as Weather Research and Forecast.

- Extreme weather events involve atmosphere and ocean.
- Commonly used ocean models are:
  - Wavewatch III (waves)
  - **Regional Ocean Model System (currents).**



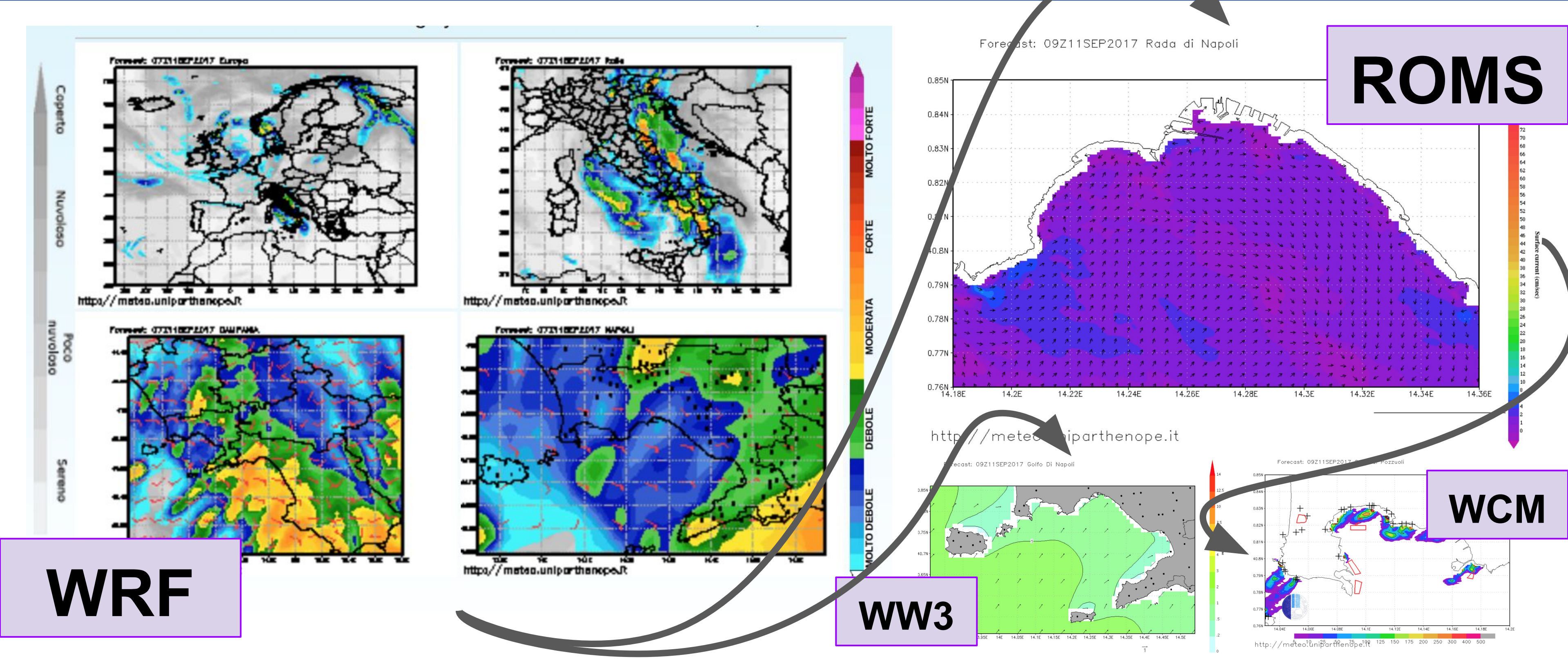
Hurricane Weather Research and Forecast Model



Storm coastal flooding

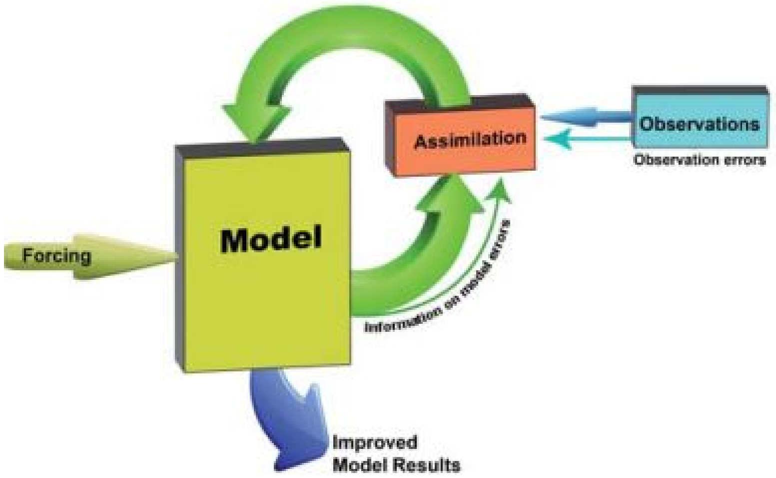
High space and time resolution marine and atmosphere environmental modelling is impressively computing power demanding.

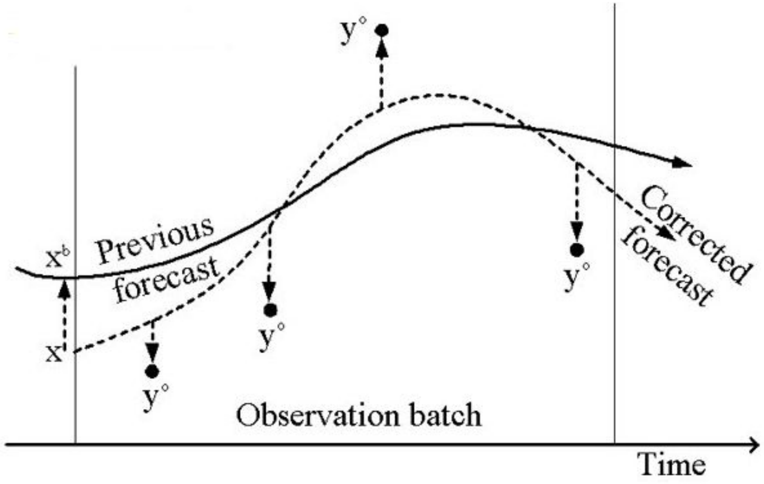
Introduction, contextualization and motivations



Operational application at the Campania Region Centre for Marine and Atmosphere Monitoring and Modelling (http://meteo.uniparthenope.it/forecast)

Introduction, contextualization and motivations





ROMS simulations are initialized by initial and boundary conditions representing the initial state of the ocean in computed domain.

Data Assimilation constraints the model's behaviour with real world sampled data.

This technique is finalized to improve the overall simulations' quality.

Unfortunately is **computing** (and **data**) demanding!

Contribution

- we consider the Incremental Strong constraint 4D Variational (IS4DVAR) algorithm for data assimilation implemented in ROMS

- **the aim is to study its performance in terms of strong scaling scalability on computing architectures such as a cluster of CPUs**.

- we consider realistic test cases with data collected in the U.S. west coast and the California Current System (CCS).

- we assess the performance of the current parallelization strategy of IS4DVAR Algorithm in terms of strong scaling

- we investigate the behavior of the communication overhead in terms of the surface-to-volume ratio.

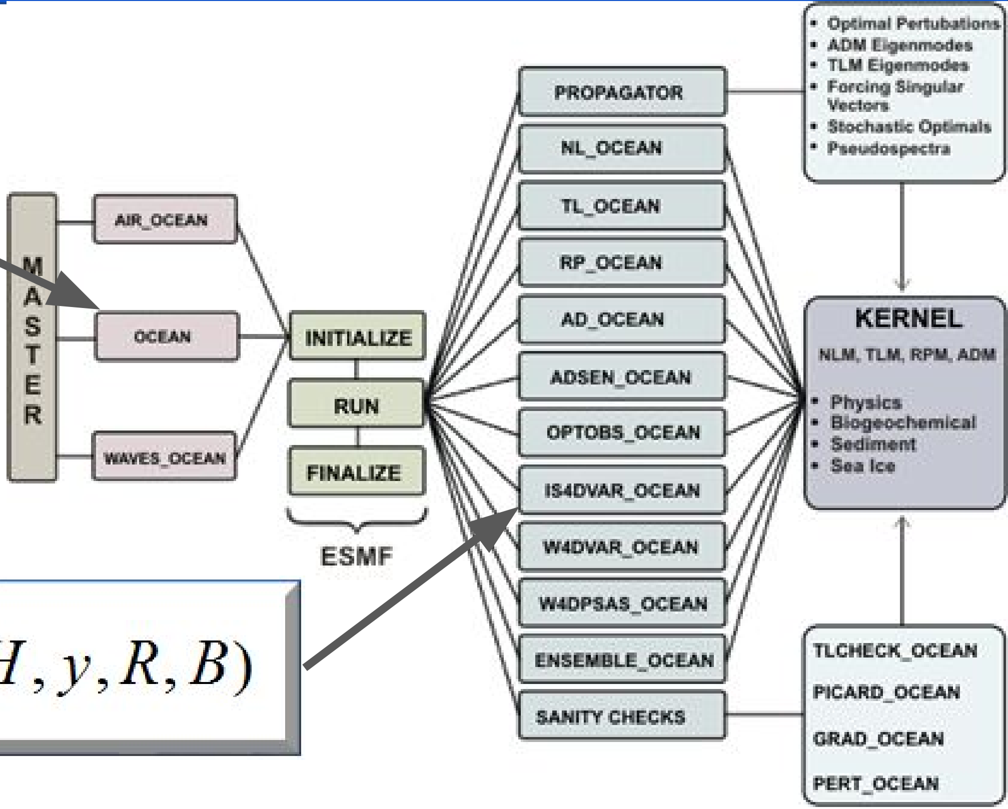
IS4DVAR: Building Blocks

$$x(t_j) = M_j(x(t_0))$$

Where:

- B: background error covariance matrix
- R: observation error covariance matrix
- M(x): nonlinear ROMS state
- H: observation operator
- Y: misfits contribution

$$x_{DA} = \arg\min J(x, x(t_0), M, H, y, R, B)$$

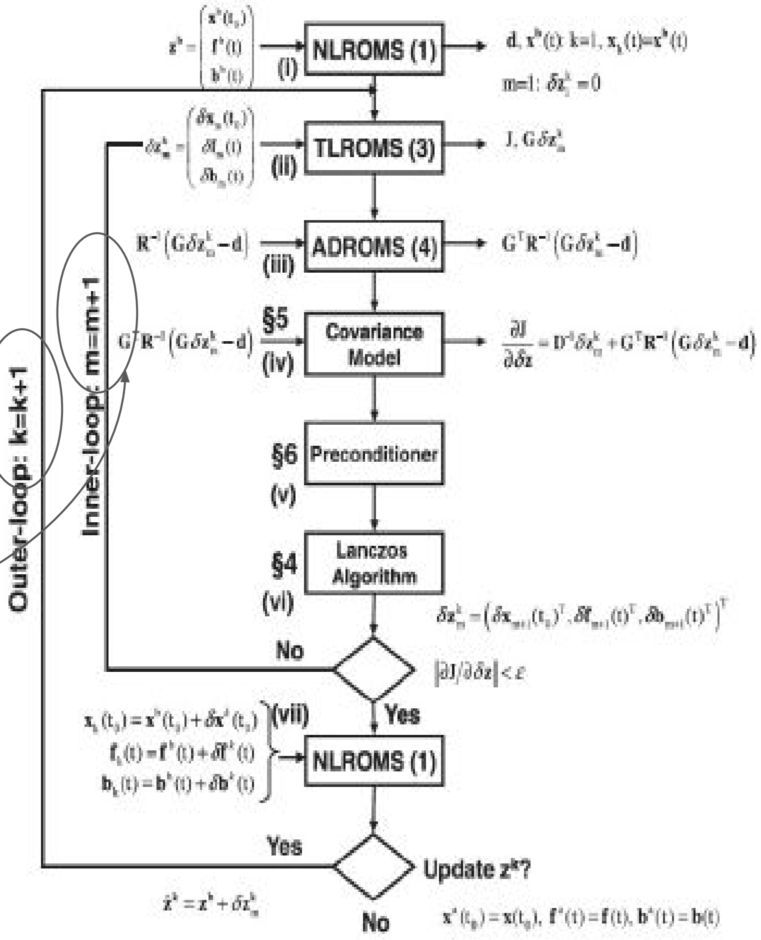


The Incremental Strong Constraint 4DVAR (IS4DVAR) Algorithm is one of Data Assimilation modules of the Regional Ocean Modelling System (ROMS)

IS4DVAR: Algorithm

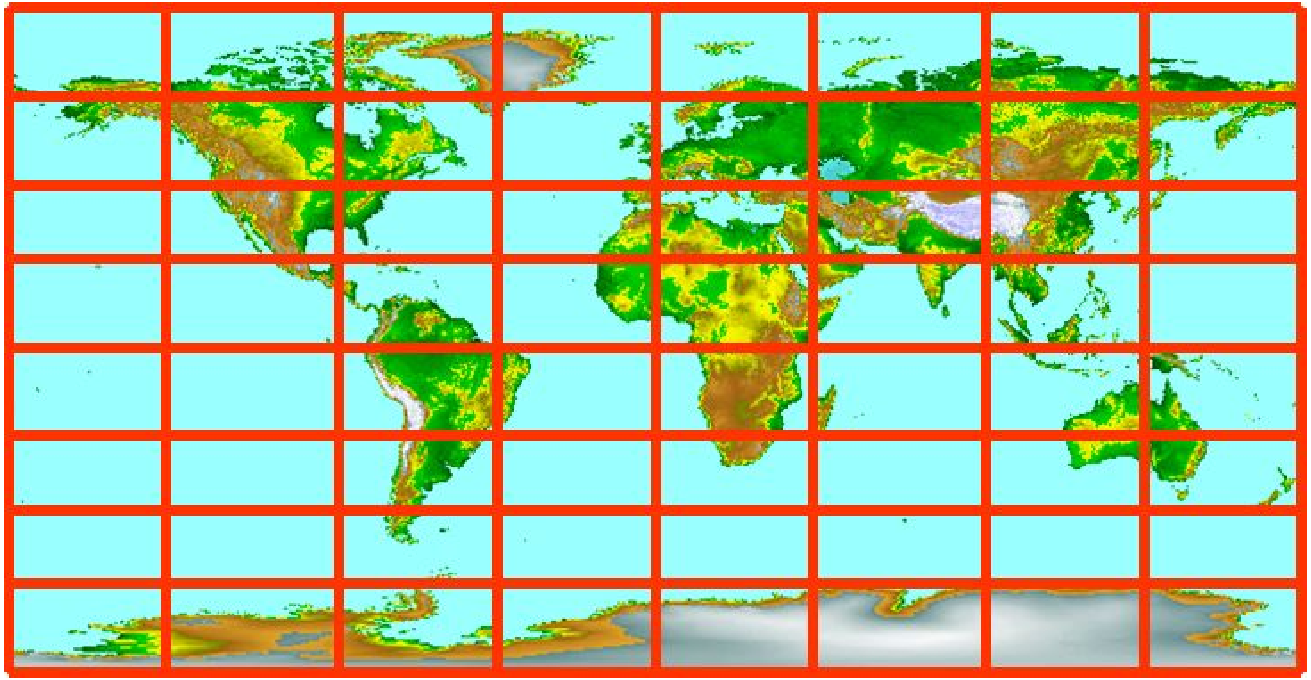
- **NLROMS:** ROMS Non Linear Model
- **TLROMS:** Tangent Linear
  (First Order Taylor Approximation of ROMS)
- **ADROMS:** Adjoint
  (for computing the Adjoint operator of ROMS).

- Parameters (where k << m):
- k: the steps for the linearization
  (First Order Taylor approximation)
- m: and m are and for the minimization algorithms by using Lanczos algorithm



Gauss-Newton: outer k-loop
Lanczos method: inner m-loop

Domain decomposition

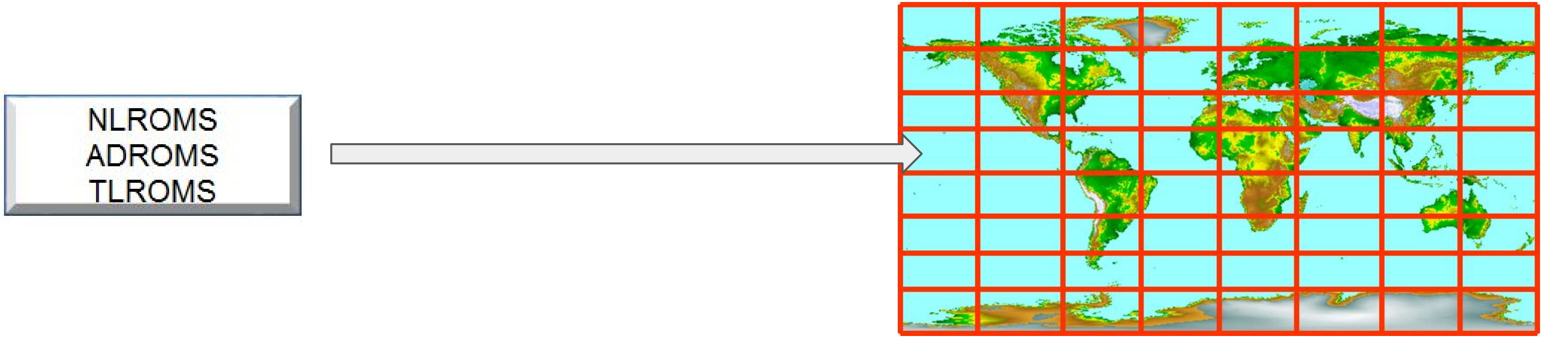


Typical parallel approach in environmental modelling.

ROMS and IS4DVAR use the same parallelization strategy.

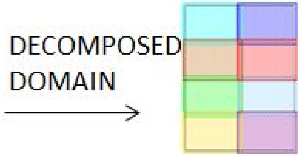
2D Domain Decomposition

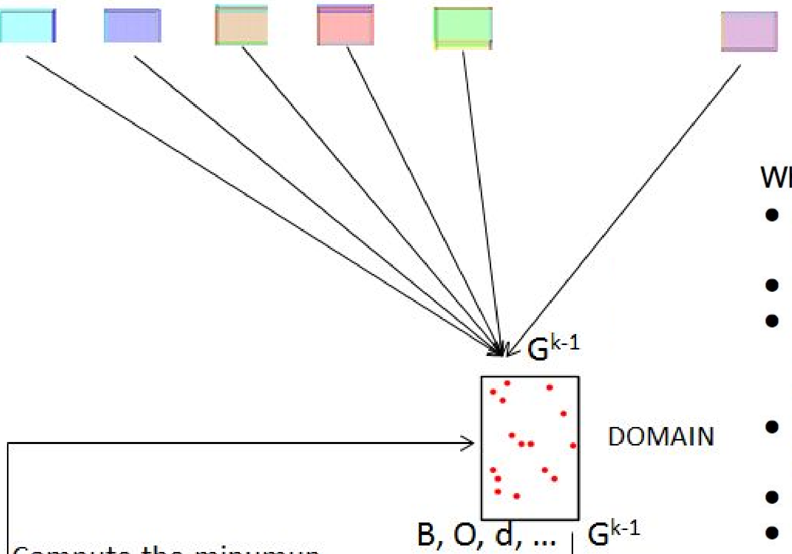
as IS4DVAR is part of the ROMS, the paralellization strategy takes advantage of the 2D domain decomposition implemented in ROMS.



NLROMS, TLROMS, ADROMS are the software modules of IS4DVAR implementing the 2D domain decomposition (across space, only)

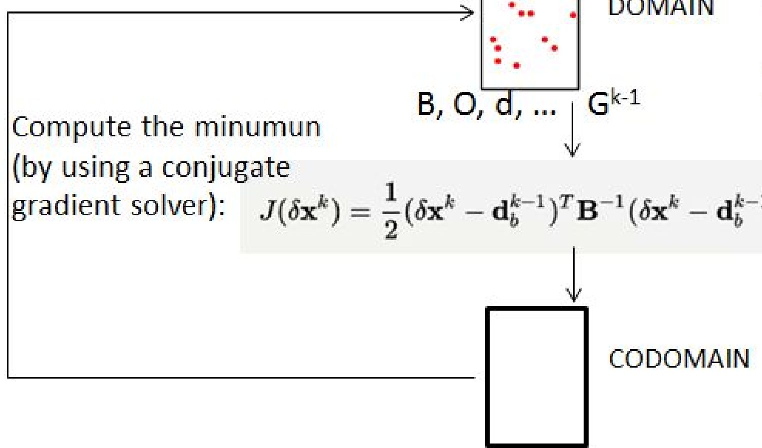
Domain Decomposition





Compute the minumun (by using a conjugate gradient solver):

$$J(\delta\mathbf{x}^k) = \frac{1}{2}(\delta\mathbf{x}^k - \mathbf{d}_b^{k-1})^T\mathbf{B}^{-1}(\delta\mathbf{x}^k - \mathbf{d}_b^{k-1}) + \frac{1}{2}(\mathbf{G}^{k-1}\delta\mathbf{x}^k - \mathbf{d}_o^{k-1})^T\mathbf{O}^{-1}(\mathbf{G}^{k-1}\delta\mathbf{x}^k - \mathbf{d}_o^{k-1})$$



Where

- $\boldsymbol{\delta x}^k$ is the state-vector defined on a domain $\boldsymbol{\Omega}$ (increment for the $\mathbf{k}^{th}$ outer-loop iteration as shown in Figure 1);
- $\mathbf{d_o}^{k-1} = \mathbf{y} - \mathbf{G}(\mathbf{x}^{k-1})$ is the misfit;
- **G(x) ≡ H M(x)** represents nonlinear ROMS state integrated to the observation times by NLROMS (**M(x)**), and interpolated to the observation locations by the observation operator **H**;
- $\mathbf{G}^{k-1}$ denotes the corresponding time integrated and interpolated increment **δx** where the tangent linear model (TLROMS) is linearized about $\mathbf{x}^{k-1}$.
- **B** is the background error covariance matrix
- **O** is the observation error covariance matrix.

1D and 2D Domain Decomposition Strategy

Let the domain Ω be decomposed in Ntile subdomains (also named tiles) with overlap areas, where

$$Ntile = NtileI \times NtileJ$$

If

$$size(\Omega) = N = N_1 \times N_2 \times N_3$$

three dimensional domain

Then in 2D-DD

$$size_{2D-DD}(tile) = \frac{N_1}{NtileI} \times \frac{N_2}{NtileJ} \times N_3$$

While in 1D-DD, it is

$$size_{1D-DD}(tile) = \frac{N_1}{NtileI} \times N_2 \times N_3$$

Experiment setup

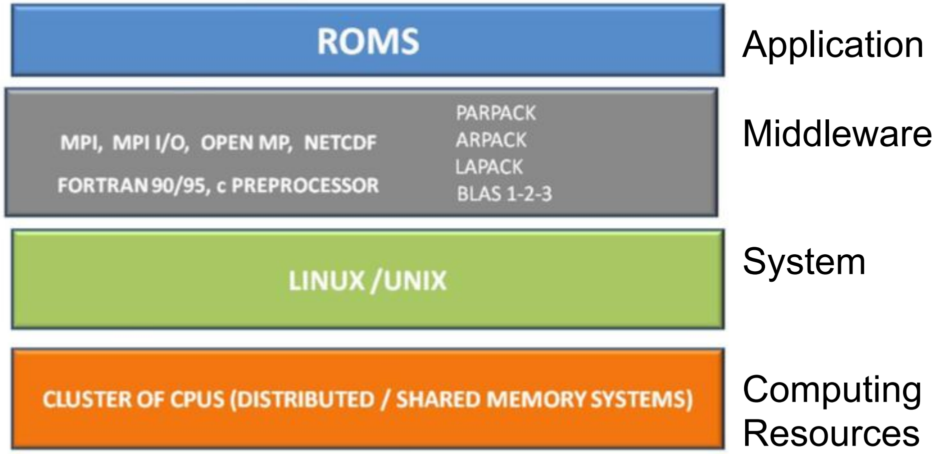
- Model:
  ROMS Version 3.7
  MPICH 3.x

  NetCDF4

  Flags: **MPI**, **IS4DVAR**, NONLINEAR, SOLVE3D, …

  PGI Compiles suite.

- Computational resources:
  HPC Cluster
  22 * Intel Xeon E4550 @ 2.84 GHz
  Infiniband



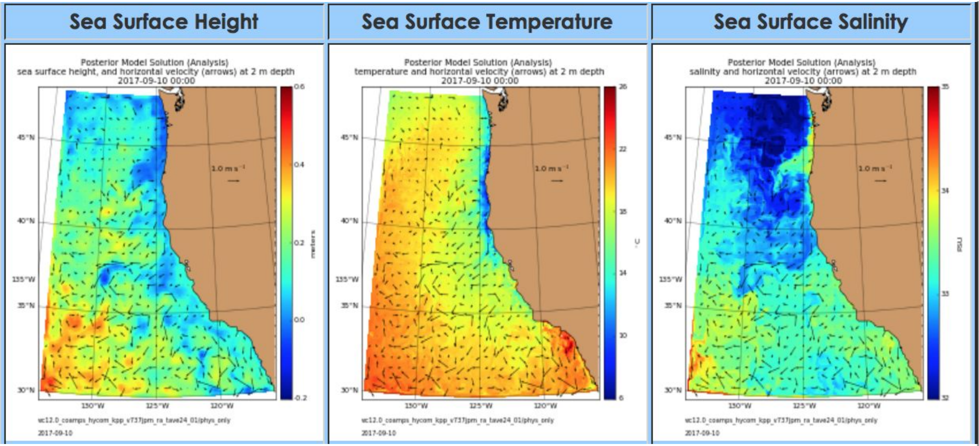
Important:

In this work we use **regular IS4DVAR** implementation in order to analyze and assess performance with the aim of designing and developing the **next** generation of **hybrid parallel** assimilation algorithms.

Test cases: California Current System

TC1: the California Current System (CCS) with 30Km (horizontal) resolution and 30 levels in the vertical direction. The global grid is then:

$$N = 54 \times 53 \times 30 = 8.586 \times 10^4$$



TC1			
p	T_p (secs)	S_p	E_p
1	7088	1	1
2	3859	1.84	0.92
4	2348	3.02	0.76
8	1704	4.16	0.52
16	1770	4.00	0.25

http://oceanmodeling.ucsc.edu/ccsnrt/

Test cases: Caspian Sea

TC2: the Caspian Sea with 8Km resolution and 32 vertical layers. The vertical resolution is set with a minimum depth of 5m. Then, problem dimension in terms of the grid/mesh size consists of:

$$N = 90 \times 154 \times 32 = 4.43520 \times 10^5$$

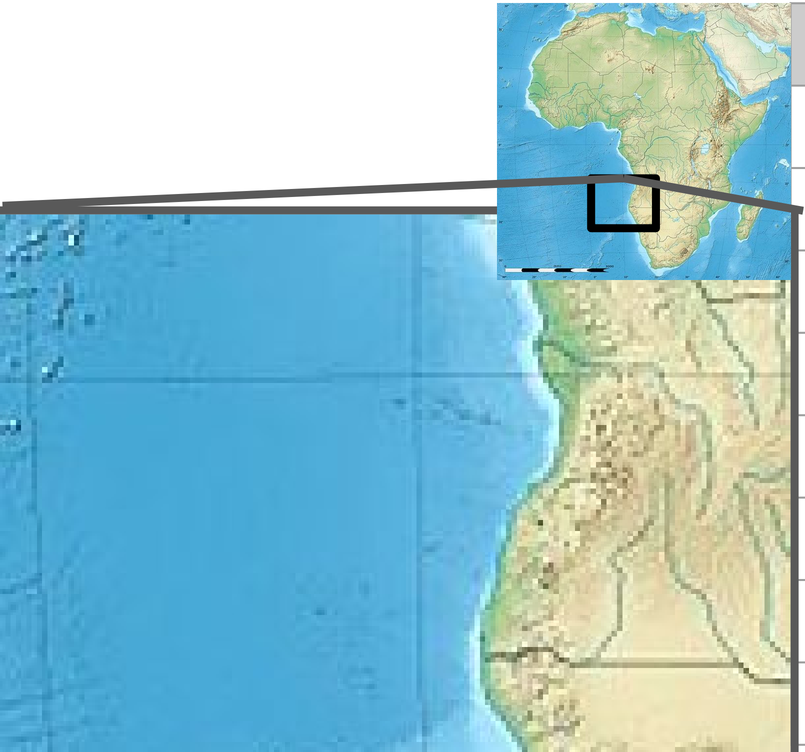
grid points. A set of sensitivity experiments (not shown) suggests that k=1 and m=50. In each of these experiments, only one assimilation cycle (4 day) is conducted.

TC2			
p	T_p (secs)	S_p	E_p
1	42224	1	1
2	24424	1.7	0.8
4	15411	2.7	0.7
8	10501	4.0	0.5
16	9117	4.6	0.3

Test cases: Angola Basin

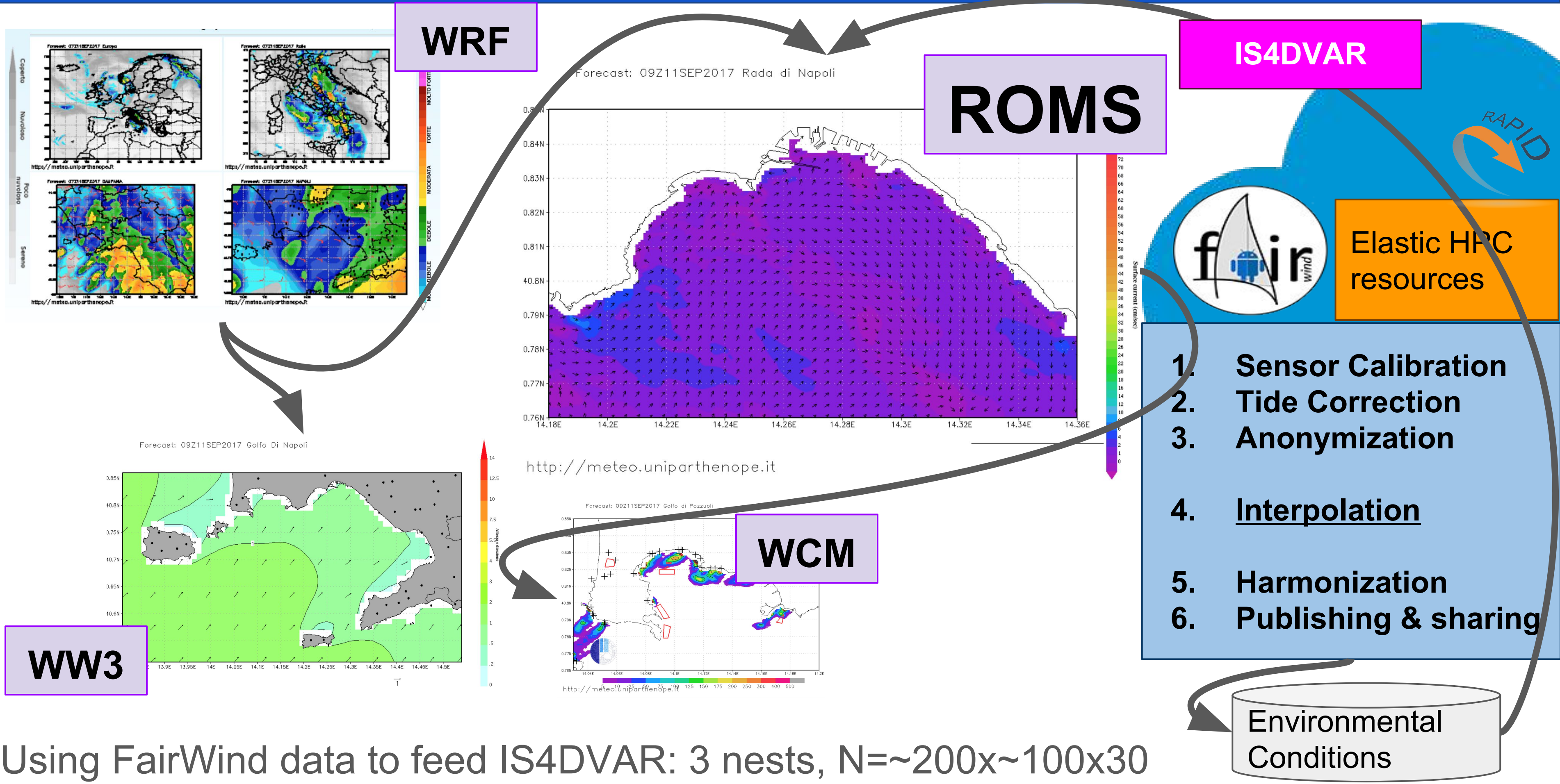
TC3: the Angola Basin with 10Km of resolution and 40 terrain-following vertical levels. The vertical levels are stretched as so to increase resolution near the surface. 2013:

$$N = 200 \text{ x } 100 \text{ x } 40 = 8.0\text{x}10^5$$



TC3			
p	**T_p (secs)**	**S_p**	**E_p**
1	109905	1	1
2	57550	1.91	0.96
4	31648	3.47	0.87
8	18697	5.88	0.74
16	11755	9.35	0.58
32	8022	13.70	0.43
64	5814	18.90	0.30

Operational Scenario



Using FairWind data to feed IS4DVAR: 3 nests, N=~200x~100x30

Conclusions and future directions

We performed some evaluation using the IS4DVAR regular parallel implementation in different use cases in search of performance bottlenecks.

We want to decompose the data assimilation variational function into a subset of problems complaining the regular model domain decomposition approach.

This approach permits hybrid distributed memory, shared memory and GPGPU parallelism.