Accepted Manuscript

Title: A Fault-Tolerant HPC Scheduler Extension for Large and Operational Ensemble Data Assimilation:Application to the Red Sea

Author: Habib Toye Samuel Kortas Peng Zhan Ibrahim Hoteit

PII: S1877-7503(17)31290-5
DOI: https://doi.org/doi:10.1016/j.jocs.2018.04.018
Reference: JOCS 867

To appear in:

Received date: 15-11-2017
Revised date: 7-3-2018
Accepted date: 22-4-2018

We describe the development of a fault-tolerant ensemble data assimilation and forecasting system for the Red Sea.

A scheduler extension has been developed to ease the submission, monitoring and dynamic steering of workflow of ensemble jobs.

The system is validated with numerical experiments assimilating real satellite ocean surface temperature and height data.

Results demonstrate the efficiency of the proposed system and continuous improvement in performances with increased ensembles.

A Fault-Tolerant HPC Scheduler Extension for Large and Operational Ensemble Data Assimilation: Application to the Red Sea

Habib Toye[a], Samuel Kortas[b], Peng Zhan[c], Ibrahim Hoteit[a,c,*]

[a] *Division of Computer, Electrical and Mathematical Sciences & Engineering, King Abdullah University of Science and Technology, Thuwal, Jeddah Zip Code 23955-6900, Saudi Arabia*
[b] *KAUST Supercomputing Laboratory (KSL), King Abdullah University of Science and Technology, Thuwal, Jeddah Zip Code 23955-6900, Saudi Arabia*
[c] *Division of Physical Science & Engineering, King Abdullah University of Science and Technology, Thuwal, Jeddah Zip Code 23955-6900, Saudi Arabia*

Abstract

A fully parallel ensemble data assimilation and forecasting system has been developed for the Red Sea based on the MIT general circulation model (MITgcm) to simulate the Red Sea circulation and the Data Assimilation Research Testbed (DART) ensemble assimilation software. An important limitation of operational ensemble assimilation systems is the risk of ensemble members' collapse. This could happen in those situations when the filter update step imposes large corrections on one, or more, of the forecasted ensemble members that are not fully consistent with the model physics. Increasing the ensemble size is expected to improve the assimilation system performances, but obviously increases the risk of members' collapse. Hardware failure or slow numerical convergence encountered for some members should also occur more frequently. In this context, the manual steering of the whole process appears as a real challenge and makes the implementation of the ensemble assimilation procedure uneasy and extremely time consuming.

This paper presents our efforts to build an efficient and fault-tolerant MITgcm-DART ensemble assimilation system capable of operationally running thousands of members. Built on top of *Decimate*, a scheduler extension developed to ease the submission, monitoring and dynamic steering of workflow of dependent jobs in a fault-tolerant environment, we describe the assimilation system implementation and discuss in detail its coupling strategies. Within *Decimate*, only a few additional lines of Python is needed to define flexible convergence criteria and to implement any necessary actions to the forecast ensemble members, as for instance (i) restarting faulty job in case of job failure, (ii) changing the random seed in case of poor convergence or numerical instability, (iii) adjusting (re-

[*] Corresponding author
Email address: `ibrahim.hoteit@kaust.edu.sa` (Ibrahim Hoteit)

ducing or increasing) the number of parallel forecasts on the fly, (iv) replacing members on the fly to enrich the ensemble with new members, etc.

We demonstrate the efficiency of the system with numerical experiments assimilating real satellites sea surface height and temperature observations in the Red Sea.

Keywords: High Performance Computing, Ensemble Data Assimilation, Bayesian Filtering, Operational Oceanography, Red Sea

1. Introduction

Capabilities in ocean modeling and simulation have witnessed tremendous progress in recent years following the advances in high performance computing (HPC) resources [12], the better understanding of the ocean physics, and the availability of ever increasing amount of in situ and remotely sensed data [14, 10]. To take advantage of all sources of information from models and observations, data assimilation, the process by which observations are incorporated into the model, is becoming more and more popular to improve the model forecasting skills along with quantification of uncertainties in its outputs [8]. Data assimilation is now recognized as a crucial component for the development of an ocean operational system.

The celebrated Kalman filter (KF) computes the best (minimum-variance) estimate of a linear dynamical system given available observations [23], and as such provides a readily efficient algorithm for data assimilation and forecasting [18]. Because of its prohibitive computational requirements when implemented with large scale systems and the nonlinear nature of the ocean dynamics, simplified Kalman filters have been introduced for ocean data assimilation ([36, 40, 18]). One of the most promising Kalman filtering schemes is the ensemble Kalman filter (EnKF), a Monte Carlo approach in which the forecast statistics are estimated from an ensemble of model forecasts [21]. An EnKF assimilation system with a high resolution model and large number of observations is expected to require a large ensemble to provide accurate ocean state estimates [20, 17]. Large ensembles should provide more reliable forecast statistics and a smooth forecast covariances for efficient implementation of the filter update steps with the observations.

Increasing the ensemble size would however not only significantly increase the computational load, but would also weaken the robustness of the system and increase the chances of system failure, and thus the workload of the user. Indeed, in case the system crashes, the user will have to manually identify the issue behind its collapse, reconfigure the system and check for consistency before relaunching the jobs. The system failures may be related to a machine problem or may be the result of a dynamical inconsistency between the statistically updated ensemble members and the forecasting model, both of which are unpredictable. The users need therefore to continuously monitor the system execution progress.

In an operational ocean forecasting system, not only huge amount of data need to be processed in a timely manner [33], but the system should also be fault-tolerant in order to recover from failure and deliver real-time responses. In this study, we address these ensemble data assimilation forecasting challenges with an EnKF data assimilation system that we configured for the Red Sea. The system is complex and brings together different components (program executables, data, computational resources). An ensemble of MIT general circulation model (MITgcm) runs are integrated in parallel to provide the forecast statistics for the Data Assimilation Research Testbed (DART) filter to perform the assimilation update with the observations. To overcome the aforementioned problems, and build an efficient fault-tolerant ensemble system we coupled the existing DART-MITgcm assimilation system [39] to a scheduler extension named *Decimate* [28]. The system in [39] was neither fault-tolerant nor scalable to ensembles of thousands of members, hence the use of *Decimate* to remediate those limitations. *Decimate* automatically generates the submission scripts along with the dependencies between the jobs and runs them in an environment where checking and restarting functions just need to be defined by the user. It simplifies the launching and monitoring processes and allows for automatic reconfiguration in case of system failure. This work describes the development of the different components of the assimilation system, their coupling and the parametrization of *Decimate*. First results from a high resolution ensemble assimilation system for the Red Sea are presented and discussed.

The paper is organized as follows. We first give an overview of ensemble data assimilation concept and the DART-MITgcm Red Sea forecasting system in Section 2. Section 3, briefly describes *Decimate* on top of which the DART-MITgcm assimilation system was implemented. Section 4 presents the results of the assimilation experiments that has been conducted in the Red Sea. Finally, a brief summary and discussion is given in Section 5.

2. Ensemble Data Assimilation and the DART-MITgcm System

2.1. Ensemble Data Assimilation

We follow a Bayesian filtering formulation of the data assimilation problem in which we aim at sequentially computing the probability distribution of the state vector of the system of interest $\mathbf{x}_k$ at time k conditional on the available measurements $\mathbf{y}_{1:k} \equiv \mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k$ up to time k, that is the posterior probability distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ using Bayes' rule [20]. Given an initial distribution $p(\mathbf{x}_0)$, the measurements $\mathbf{y}_{1:k}$, the state space model

$$\begin{aligned} \mathbf{x}_k &= \mathcal{M}_k(\mathbf{x}_{k-1}) + \eta_k && (1) \\ \mathbf{y}_k &= \mathcal{H}_k(\mathbf{x}_k) + \varepsilon_k && (2) \end{aligned}$$

from which one can obtain the transition distribution $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and the likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$, the computation can be performed recursively to incorporate the new observation $\mathbf{y}_k$ into the posterior $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ at time $k-1$ to obtain the posterior $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ at time k. $\mathcal{M}_k$ is the dynamical model for advancing the state vector $\mathbf{x}_{k-1}$ from time $k-1$ to time k, and $\mathcal{H}_k$ is the measurement model (or observation operator) at time k. η_k and ε_k respectively refer to independent Gaussian model and observation errors.

Given $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$, the Chapman-Kolmogorov equation

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1}$$

is used to forecast the state probability distribution at the next time with the dynamical model (1), computing the distribution of $\mathbf{x}_k$ conditional on the observations up to time $k-1$. Bayes' rule is then applied to update the forecast distribution with (the new observation) $\mathbf{y}_k$ to obtain the posterior probability distribution

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}.$$

This forecast-update cycling process is then repeated whenever a new observation is available.

Bayesian filtering finds applications in many fields including signal processing, meteorology, oceanography, hydrology, finance, motion tracking (of fluids, satellites, airplanes, ...), among others.

A special case of the Bayesian filter is the Kalman filter, which is designed for linear systems based on orthogonal projections [23, 34]. Under the assumption of independent Gaussian model η and observation ε noise, the Kalman filter is optimal in the sense that it computes the best linear unbiased estimate (BLUE). Moreover, due to appealing features, namely easy of implementation, markovian property (or memory less feature), and sequential process for incorporating the observations, it is widely used in many fields. Nevertheless in oceanography, where the state dimension could be very large (10^7 or more) and the dynamics are strongly nonlinear, a direct implementation of the Kalman filter is not feasible [18, 20].

To overcome this problem, Evensen [11] introduced the so-called ensemble Kalman filter (EnKF) as a Monte Carlo implementation of the Kalman filter. In the EnKF, a given (analysis) ensemble of state realization $\mathbf{X}^f = [\mathbf{x}^{a,1}, \mathbf{x}^{a,2}, \cdots, \mathbf{x}^{a,N}]$ is advanced with the dynamical model (1) to compute the forecast ensemble, from which the covariance matrix $\mathbf{P}^f$ used to compute the Kalman Gain is approximated by $\mathbf{P}^{f,e} = \frac{1}{N-1}\left(X^{'}X^{'T}\right)$, where $X^{'} = [\mathbf{x}^{f,1} - \overline{\mathbf{x}}, \mathbf{x}^{f,2} - \overline{\mathbf{x}}, \cdots, \mathbf{x}^{f,N} - \overline{\mathbf{x}}]$ is the ensemble of anomalies and $\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{f,i}$ the ensemble mean. Once $\mathbf{y}_k$ becomes available, each member of the forecast ensemble is updated using the Kalman filter update step

$$\mathbf{x}_k^{a,i} = \mathbf{x}_k^{f,i} + \mathbf{K}_k\left(\mathbf{y}_k^o + \varepsilon_k^i - \mathbf{H}_k\mathbf{x}_k^{f,i}\right), \quad i = 1, \cdots, N \tag{3}$$

where $\mathbf{K}_k$ is the Kalman Gain

$$\mathbf{K}_k = \left(\mathbf{H}_k \mathbf{P}_k^{f,e}\right)^T \left[\mathbf{H}_k \left(\mathbf{H}_k \mathbf{P}_k^{f,e}\right)^T + \mathbf{R}_k\right]^{-1}, \tag{4}$$

and ε_k^i is sampled from the distribution of the observation error, assumed $\mathcal{N}(0, \mathbf{R}_k)$ [9]. This directly provides an ensemble to start the next assimilation (forecast-analysis) cycle. Perturbing the observations was however shown to introduce noise in the update when the filter is implemented with small ensembles, which is the case in real ocean applications [19]. For such applications, deterministic EnKFs were introduced to update the ensemble forecast mean and covariance exactly as in the Kalman filter, but in the ensemble space to avoid the explicit computation of $\mathbf{P}_k^f$ [18]. A resampling step is then needed to generate a new ensemble matching the updated state and covariance before forecasting. Many flavors of deterministic EnKFs have been proposed depending on the choice of the resampling procedure (e.g. [18, 3, 2, 19, 38]).

Advancing the ensemble $\mathbf{X}^f$ with the model allows to propagate the state estimation errors, otherwise impossible for very large systems. The integration of the ensemble members could be performed in parallel if enough computational resources are available, which may tremendously accelerate the forecast process.

Using small ensembles is common in realistic ocean applications, but results in a low-rank error covariance matrix, so that an observation could impact locations faraway from its vicinity [16] through spurious correlations in the covariance matrix [21]. Localization is a popular auxiliary technique that is utilized to increase the covariance matrix rank and restrain the correlations to surrounding areas [13, 16, 22, 35, 37]. Inflation might be also applied to avoid inbreeding [2, 15, 18], i.e. all ensemble members collapsing around the mean, due to the omission of various sources of uncertainties.

2.2. DART-MITgcm ensemble assimilation system

DART-MITgcm is an ocean ensemble data assimilation system that combines the MIT general circulation ocean model (MITgcm) for ocean forecasting, the Data Assimilation Research Testbed (DART) package for ensemble updating with the observations, and scripting for the forecast-analysis workflow management.

2.2.1. MITgcm

MITgcm is a numerical model designed for the study of the atmosphere, ocean, and climate. In the ocean enviroment, MITgcm solves the incompressible, Navier-Stokes equations under the Boussinesq approximation. Equations are written in z coordinates and discretized in a staggered "Arakawa C-grid". The pressure field is computed by solving an elliptic equation with Neumann boundary conditions. Finite volume techniques are employed yielding an intuitive discretization and support for the treatment of irregular geometries. Domain decomposition makes the code appropriate for parallel processing on high performance computers by slicing the ocean domain in vertical columns. The model has been developed to perform efficiently on a wide variety of computational platforms. More details might be found in [31, 32].

In this study, MITgcm is configured at a $\sim 4\ km$ (0.04°) spherical polar grid. 50 vertical levels are used with layers of different thickness, ranging from 4 m at the surface to 300 m near the bottom. The domain extends from $10^\circ N$ to $30^\circ N$, and $30^\circ E$ to $50^\circ E$, including the entire Red Sea, the two gulfs in the northern end and the Gulf of Aden that marks the southern limit. The bathymetry is interpolated from the General Bathymetric Chart of the Oceans (GEBCO, accessible at `http://www.gebco.net/data_and_products/gridded_bathymetry_data/`). Monthly data fields including zonal and meridional velocities, temperature, and salinity from the German contribution of the Estimating the Circulation and Climate of the Ocean project (GECCO) are specified as the eastern lateral boundary conditions. The surface forcing is extracted from the 6-hourly reanalysis of the European Centre for Medium-Range Weather Forecasts (ECMWF). These include zonal and meridional wind speed, downward short wave and long wave heat fluxes, air temperature, specific humidity, and precipitation.

A free model run (without assimilation) is first integrated from Jan-1st-1979 with a time step of 200 s for 32 years. In the assimilation experiments, the outputs from 1992 to 2011 are used for constructing the initial ensemble of state realizations that initiates on Jan-1st-2006. The readers are referred to [39] for more details.

2.2.2. DART

DART is an ensemble data assimilation software written in fortran at the National Center for Atmospheric Research (NCAR). It implements parallel scalable algorithms of various ensemble filters, including EnKF, and a deterministic EnKF, the ensemble adjustment Kalman filter (EAKF) [7] suitable for modern suprecomputers, and capable of handling very large simulation models such as the ones encountered in oceanography and meteorology. In the update step, DART assimilates the observations serially i.e. one single observation at a time [19], which allows to efficiently parallelize it. DART is also equiped with sophisticated localization [1, 5] and inflation [4, 6] schemes for efficient EnKFs assimilation with small ensembles.

2.2.3. Assimilation workflow

The assimilation workflow is depicted in Figure 1. Given an initial ensemble of size N, an affordable computation of equations (3) and (4) is performed by DART filter (see the process labeled DART). N MITgcm models are then integrated to complete a forecasting step. Then in the decision block, if the current number of cycles (curr_nb_cycles) did not reach the number of cycles specified by the user (user_nb_cycles), another DART filter computation is carried out, otherwise the workflow terminates.

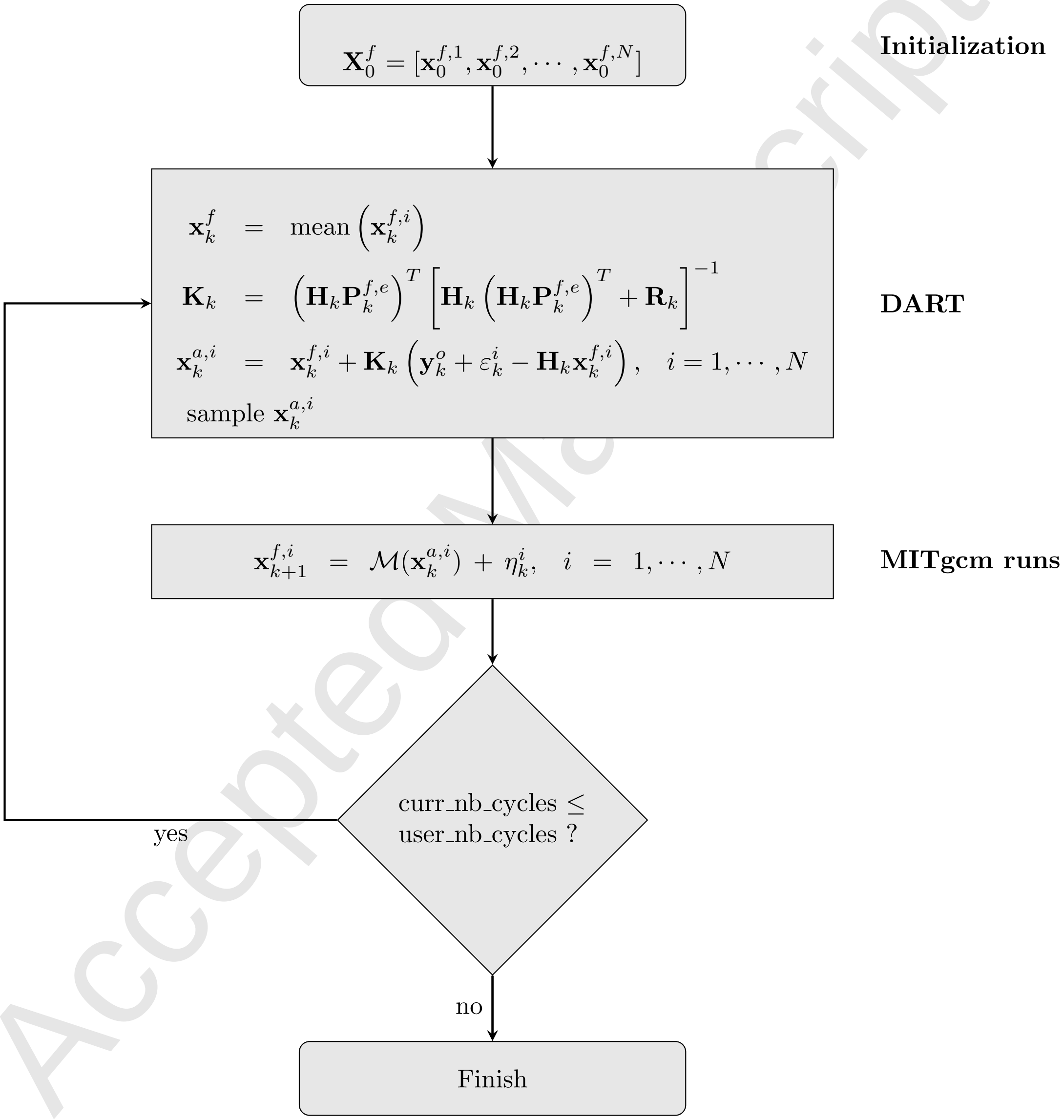
$\mathbf{X}_0^f = [\mathbf{x}_0^{f,1}, \mathbf{x}_0^{f,2}, \cdots, \mathbf{x}_0^{f,N}]$
Initialization
$\mathbf{x}_k^f \; = \; \mathrm{mean}\left(\mathbf{x}_k^{f,i}\right)$
$\mathbf{K}_k \; = \; \left(\mathbf{H}_k\mathbf{P}_k^{f,e}\right)^T \left[\mathbf{H}_k\left(\mathbf{H}_k\mathbf{P}_k^{f,e}\right)^T + \mathbf{R}_k\right]^{-1}$
$\mathbf{x}_k^{a,i} \; = \; \mathbf{x}_k^{f,i} + \mathbf{K}_k\left(\mathbf{y}_k^o + \varepsilon_k^i - \mathbf{H}_k\mathbf{x}_k^{f,i}\right), \quad i = 1, \cdots, N$
sample $\mathbf{x}_k^{a,i}$
DART
$\mathbf{x}_{k+1}^{f,i} \; = \; \mathcal{M}(\mathbf{x}_k^{a,i}) + \eta_k^i, \quad i \; = \; 1, \cdots, N$
MITgcm runs
curr_nb_cycles ≤
user_nb_cycles ?
yes
no
Finish

Figure 1: DART-MITgcm assimilation workflow. curr_nb_cycles is the current number of cycles and user_nb_cycles is the desired number of cycles specified by the user.

2.2.4. Specification and challenges for an operational implementation

To run the assimilation workflow in an operational setting, some practical constraints should be taken into account. Figure 2 gives a graphical picture of those constraints. Once the initialization is done, the filter starts. In case of failure, the filter is restarted up to mr times (mr is a shortcut for maximum number of retries). And if after the $mr+1$ trials the filter still fails, the workflow is aborted and goes in the garbage failure state (not shown in the figure). After a successful filter completion, the N MITgcm instances can begin. The N MITgcm programs run independently and each of them restart up to mr times in case of failure. Any successful MITgcm waits in the barrier state for the remaining MITgcms to complete. If any MITgcm still fails to succeed after the mr retries, the workflow stops and enters the failure state. Upon successful completion of all the MITgcms, the system is in the barrier state. Then another assimilation cycle is launched if the required number of cycles is not reached, if not, the system goes to the end state and the workflow finishes successfully. The required time to compute the solution is of course an important factor for an operational system in order to provide needed information for real-time decisions.

2.2.5. Current implementation, issues and limitations

The workflow is designed to run on supercomputers and therefore launching scripts should be written and submitted through a scheduler. Due to the time wall clock policy of many supercomputer centers limiting the execution time of a single job, the full workflow cannot be submitted within one script and needs to be splitted. For that purpose we generate a submission script for each filter state and MITgcm integration. Generating bash scripts helps in writing the submission scripts since a typical run requires ten thousands of jobs and some submission parameters (e.g. the required number of nodes, the time wall clock, the number of cycles, ...) might vary. Indeed, a manual generation of those scripts is not feasible. SLURM (the current scheduler for submitting jobs on KAUST supercomputer Shaheen) allows dependencies handling by means of the command `--dependency`. Moreover the `--array` command is used to submit the MITgcm jobs in parallel with the same scripts lines for all the jobs of a given cycle, making the code more compact. As many other supercomputer centers, our center imposes a limit on the maximum number of jobs per user Therefore, even though the workflow has been split, it still cannot be submitted if the number of jobs breaks the maximum number of jobs per user limit. Even worse, no assimilation cycle can complete if the ensemble size is greater than the number of jobs limit. Another concern is that the assimilation workflow brings into play huge amount of data so that we need to review and adapt classic data management procedures. The reason is that the stress on the filesystem increases along with the memory usage, which may lead to machine instability and may result in the workflow abortion. In other instances the failure could be related to the divergence of the physical process being modeled. This is because the linear Kalman update, although statistically optimal (among linear estimators), is not constrained to be dynamically consistent with the physics of

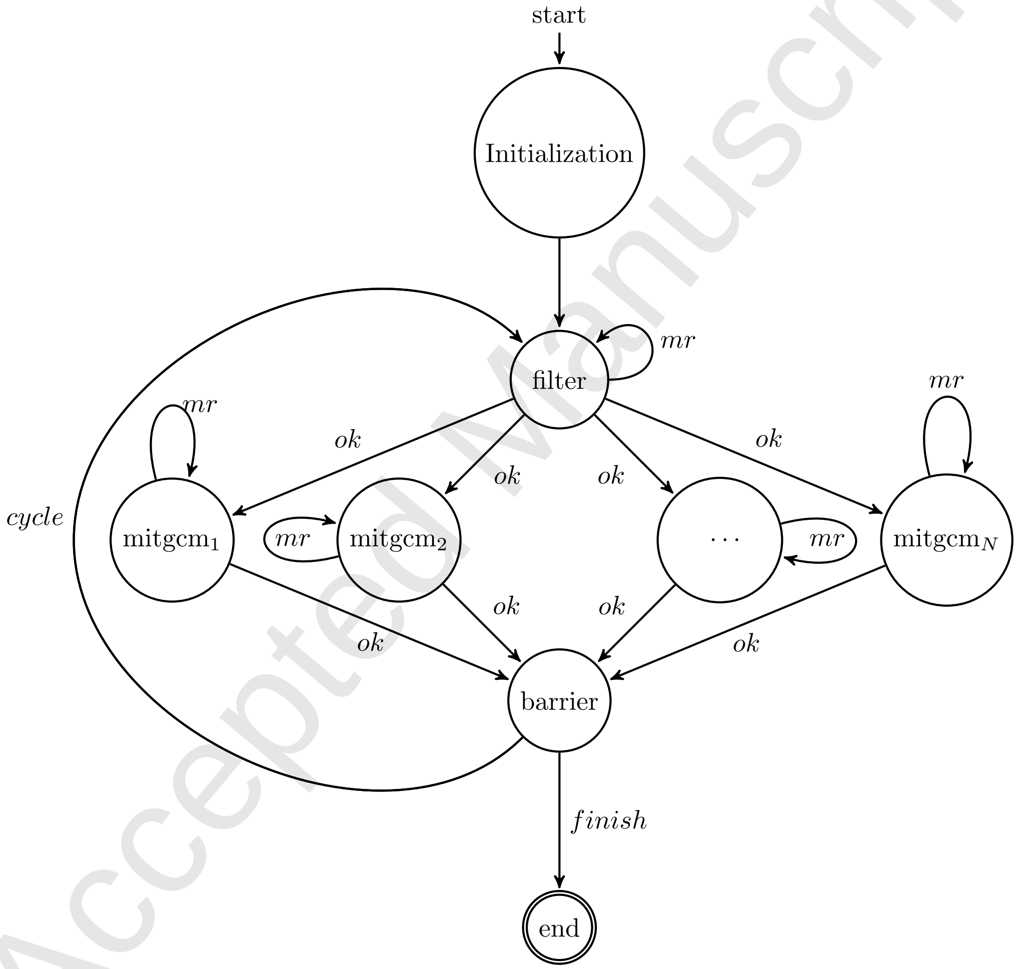

Figure 2: Jobs sequence state machine of DART-MITgcm assimilation workflow. mr stands for maximum number of retries.

the MITgcm (or another ocean circulation model). One may have therefore to often deal with situations during which the MITgcm is not capable of forecasting some of the ensemble members, and the workflow will end. Whatever is the failure cause, the workflow needs to restart. The failures are unpredictable and someone has to check from time to time and relaunch the workflow if necessary. The checking process is exhausting, time consuming and inefficient especially for a real-time operational system. To address and solve the mentioned issues, and also to be more compliant with the specification discussed in Section 2.2.4, the existing system is combined with *Decimate*, a scheduler extension described in the next section.

3. An efficient implementation of DART-MITgcm workflow based on *Decimate*, a fault-tolerant scheduler extension

A launching, monitoring and validating tool named *dart_mitgcm* has been specifically designed for developing a fault-tolerant DART-MITgcm ensemble assimilation framework for the Red Sea. Written in Python 2.7, it inspires from the original shell scripts that can be found in DART or MITgcm documentations and relies on our execution framework *Decimate*.

The purpose of this section is not to present *Decimate* itself but to detail how DART-MITgcm was implemented in this environment. More information about the implementation and use of *Decimate* itself can be found in [28] and a comprehensive documentation is available at [26]. Distributed as an open source software on Github [25] [24], *Decimate* is freely available and can also be easily installed as a python module distributed from the `pypi.python.org` repository [27].

3.1. Decimate: a robust scheduler extension

In a supercomputing environment, simultaneously accommodating needs of users scalability and capacity is challenging. This often leads to the implementation of a scheduling policy limiting the number of jobs per user in the queue in order to reduce waiting times in queue and optimize restitution duration. In order to enable efficient use of the computing resources by users producing large number of jobs, *Decimate* was developed by KAUST Supercomputing laboratory to ease the submission, monitoring and dynamic steering of workflow of dependent jobs. Written in Python 2.7, it extends the SLURM scheduler, transparently adding prologue and epilogue to any user script and submit the right job dependency that automatically add new chunks of work or relaunch a job in case of a hardware, software or numerical convergence failure.

Decimate easily allows a user to:

- Submit any number of jobs regardless of any limitation set in the scheduling policy on the maximum number of jobs authorized per user.
- Manage his set of jobs: all the submitted jobs are seen as a single workflow easing their submission, monitoring, deletion or reconfiguration and

a centralized log file is created capturing all relevant information about the behavior of the workflow. From Python or shell, at any time and from any jobs, the logging levels info, debug, console and mail are available.

- Via a user-written function, check for correctness of the outputs resulting at the end of a given job and if not make the decision either to stop the whole workflow, to resubmit partially the failing components as is, or to modify it dynamically.

3.2. Checking function

Decimate transparently handles the submission, monitoring and resubmission of failed jobs. It is taught what decision has to be made if a part of the workflow failed via a user function written in Python or via a script shell that is passed as a parameter at the initial submission of each MITgcm or DART tasks.

In our case, in a Python function of less than 100 lines, we are checking:

- If output files are produced and contain completeness messages, the task is tagged as COMPLETED and CORRECT. It will not be resubmitted, allowing *Decimate* to go on with the next steps of the workflow.
- If no output file is found, the task is tagged as INCOMPLETE and will be resubmitted by *Decimate* if a maximum number of retries has not been reached. This typically happens in the case of hardware failure or if the job required duration has been under-estimated.
- If an error message related to numerical dynamical convergence issues in the MITgcm forecast step is detected in output or error files, task is tagged as COMPLETE but INCORRECT. In this case, the “faulty” members are replaced. Many replacement strategies can be implemented. We opted for a dictionary based replacement strategy in which the faulty members are replaced by their closest equivalent among the dictionary members, based on different metrics (l1 norm or l2 norm). These tasks will be resubmitted by *Decimate* if a maximum number of retries has not been reached. In the experiments presented hereafter, the dictionary was constructed from the outputs of a long MITgcm run.

4. Experimental setup, application and results

The experimental setup is similar to the one described in [39]. The assimilation experiments are conducted over a 2 months period starting on January-1-2006 and includes 20 assimilation cycles, one update step every three days. The updates were performed based on a deterministic EnKF, the ensemble adjustment Kalman filter (EAKF) [2]. Four experiments are performed (as summarized in Table 1): two experiments with 1000 members to assess *Decimate* efficiency, and two experiments with 100 members as references to evaluate the overall behavior of the assimilation system. This is the first reported 1000-members EnKF run with a high resolution general circulation ocean model. Two of these experiments (with 100 members and 1000 members) use a localization cutoff radius of 0.05 rad (about 300 km), while the remaining two do not apply localization. Moreover, an inflation factor of 1.1 is used in all the experiments.

Table 1: Experiments.

	100 members	1000 members
localization radius = about 300km	experiment 1	experiment 2
no localization	experiment 3	experiment 4

4.1. Decimate assessment

Using *Decimate*, we experienced that handling a workflow involving 1000 members and 20 assimilation cycles was a relatively smooth process. All the experimentation took place on Shaheen from August 21 to September 07 2017 where the average load of the machine was around 90%.

22000 independent successful runs of MITgcm were executed. During the process:

- 617 failed MITgcm runs did not complete due to model failures and were followed by a replacement of members ($\leq 3\%$).
- Roughly 10% of jobs failed because of hardware failures, especially before a maintenance period scheduled on Sept 3, when Shaheen lustre filesystem was highly sollicitated by other users and unstabilities occurred. Our fault-tolerant assimilation system was able to resubmit those jobs.

Table 2: Number of members replaced in each Experiment.

Experiment	*100 members*	*1000 members*
With localization	None	9 at cycle 7, 2 at cycle 10, 583 at cycle 11
Without localization	None	7 at cycle 5, 15 at cycle 10, 1 at cycle 20

Before this new implementation, some attempts to handle similar workflow had been made successfully by our team. But while reaching a complete simulation with 100 members, at least 20% of time was spent in the manual steering of the workflow and the multiple manual correction and resubmission of jobs after sporadic hardware failures or numerical issue. Reducing this overhead to a minimum thanks to the automation of restart and decision making in case of glitch, *Decimate* greatly eased the launching and monitoring process and made the system more trackable even using a much a higher number of members.

4.2. Assimilation performance

One key assumption in the EnKF is the distribution of the forecast error to be Gaussian, based on which the members are updated with the observations using the Kalman linear correction step. The distribution of the forecast error, i.e. the prior distribution, is estimated from the statistics of the forecast ensemble anomalies. We first assess the relevance of this assumption in our setup by analyzing the histogram of sea surface temperature (SST) and sea surface height (SSH) ensembles at three locations in the northern, central and southern Red Sea at assimilation cycle 4 as shown in Figures 3 and 4, respectively. The figures suggest that, for both SST and SSH, the prior distribution with 1000 members is clearly more Gaussian than that with 100 members. A smaller ensemble drastically reduces the computational cost associated with the MITgcm ensemble forecast runs, but seems to provide a more scattered ensemble and less Gaussian Monte Carlo-based approximation of the prior distribution, which may limit the efficiency of the Kalman-based update step of the EnKF.

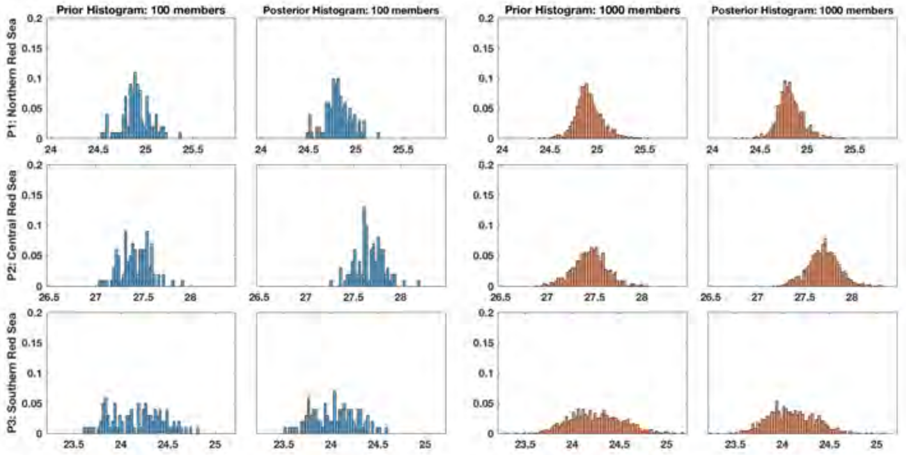

Figure 3: The histograms of forecast (Prior) and analysis (Posterior) in experiments 3 ($N = 100$, blue panels) and 4 ($N = 1000$, red panels) based on SST ensembles at three selected locations in the northern, central and southern basins of the Red Sea, as the 1st, 2nd and 3rd rows, respectively.

It is important to monitor both the state estimatiion forecast and analysis errors to make sure that the filter update is efficient at improving the forecast and that the resulting analysis state is compatible with the MITgcm dynamics. The time-evolution of the root-mean-square errors (RMSEs) between SST/SSH observations and filter forecast/analysis states as they result from the different experiments are plotted in Figure 6. The analysis RMSEs of both SST and SSH are smaller than their forecast counterparts, suggesting the filter's efficiency at

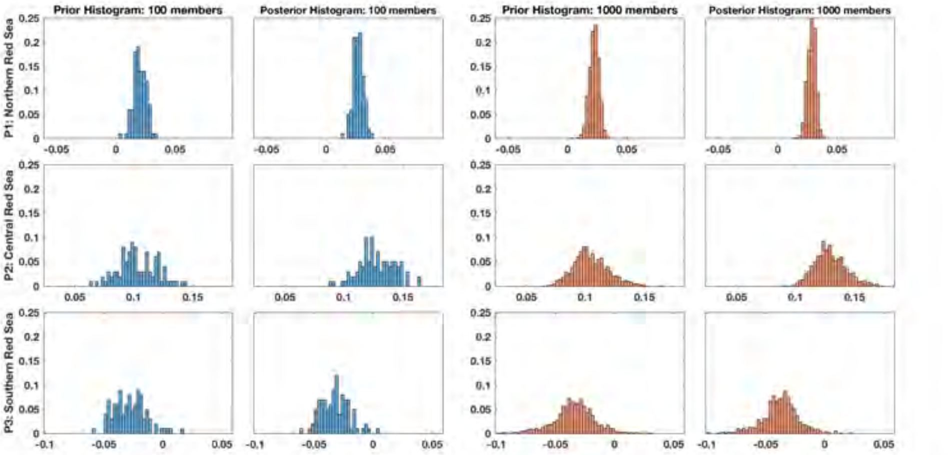

Figure 4: The histograms of forecast (Prior) and analysis (Posterior) in experiments 3 ($N = 100$, blue panels) and 4 ($N = 1000$, red panels) based on SSH ensembles at three selected locations in the northern, central and southern basins of the Red Sea, as the 1st, 2nd and 3rd rows, respectively.

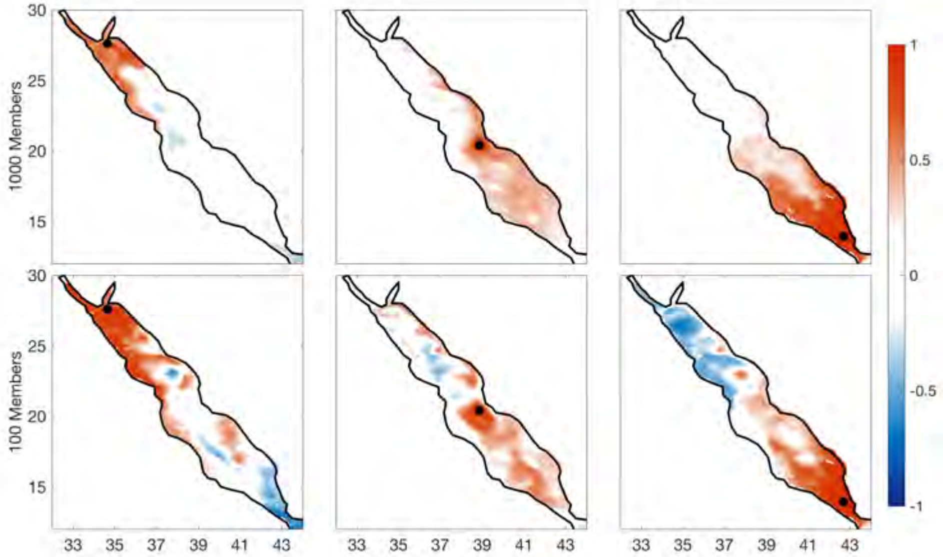

Figure 5: Sampled correlations for SSH as computed from the assimilation runs with 1000 (upper panel) and 100 (lower panel) members at three selected locations in the northern (1st column), central (2nd column), and southern (3rd column) basins of the Red Sea, respectively.

providing reliable estimates. Compared with SST RMSE, the SSH RMSE is more fluctuating since the model-data difference was calculated with different along-track SSH observations whose locations vary from one step to another. A comparison between the blue and red curves plotting the filter RMSEs with 100 and 1000 members, respectively, suggests that the RMSEs of both SST and SSH generally decrease as the ensemble size increases from 100 to 1000. The ensemble spread, an indicator of the filter estimates uncertainties, plotted in Figure 6 (e) and (f) suggest that the ensemble spread is quickly reduced after the first few assimilation cycles before leveling off. The ensemble spread is further better maintained with 1000 members, which should impose more pronounced filter's updates.

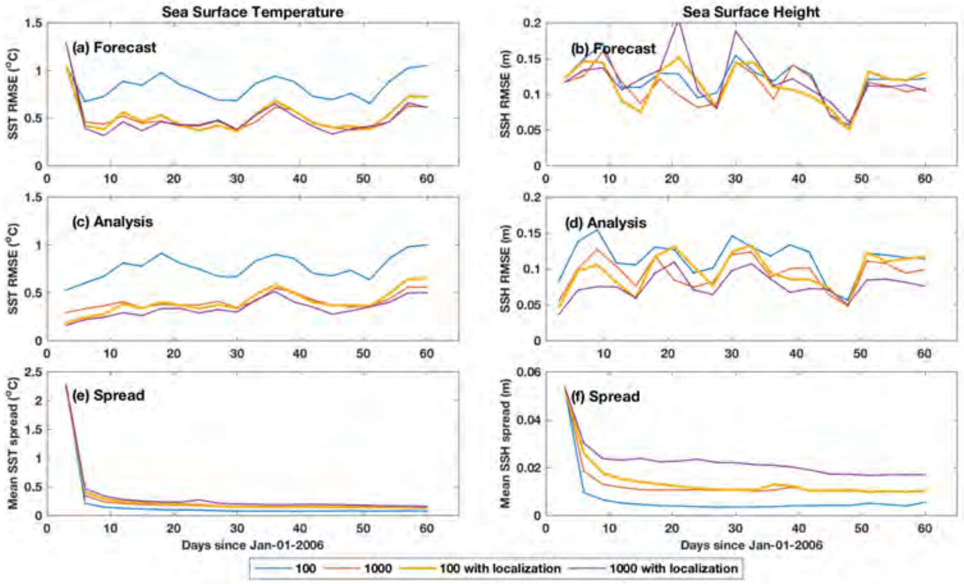

Figure 6: Time evolution of the forecast and analysis SST/SSH RMSE and ensemble spread with ensemble size of 100 (blue), 1000 (red), 100 with localization (yellow) and 1000 with localization (purple).

The error covariance in the EnKF, along the modes of which the filter's update is applied, is estimated via the forecast ensemble anomalies. These only provide $N-1$ directions in phase space, which means that the update step will not be able to exploit more that $N-1$ "information" from the observations. To deal with this low rank problem and to remove eventual spurious long-range correlations, a covariance localization [16] cutoff radius of about 300 km is implemented. Localization is a simple technique that enables efficient implementation of an EnKF with small ensembles. As shown in Figure 6 (b) and (d), the experiment 1 ($N = 1000$ with localization, purple curve) has the smallest analysis, but not forecast, RMSE of SSH. This means that the data might have been over-fitted in this experiment, and suggests testing with larger covariance localization scale. In addition, a close examination of the results shows that localization also helps to maintain the spread of both SST and SSH with proper tuning. Using more members allows to rely less on localization and improves the filter's performance, but at a significant increase in computational cost. Indeed, Figure 5 shows that the correlation range with 100 members is wider than the one with 1000 members, and that the impact at the selected points are more localized with 1000 members.

As an illustration of the system performance, the spatial distribution of the forecast and analysis states and their increment (the difference between the analysis and forecast states) on Jan-12-2006 are compared with remote sensing observations of SST (Fig. 7) and SSH (Fig. 8). Overall, the forecast and analysis fields agree well with the remote sensing data. By exploiting the high-resolution model dynamics, the results provide more details of the mesoscale variability in the basin, which is one of the key features of the Red Sea circulation [41, 42]. The distributions of lower SST in the northern Red Sea in experiment 4 ($N = 1000$, Fig. 7-a) and experiment 1 ($N = 100$ with localization, Fig. 7-c) are closer to observations (Fig. 7-d) compared with experiment 3 ($N = 100$, Fig. 7-b). A better maintained ensemble spread also helps to extract more information from along-track SSH data, as can be seen in the increments fields of SSH in experiment 4 ($N = 1000$, Fig. 8-h) and experiment 1 ($N = 100$ with localization, Fig. 8-j) compared with experiment 3 ($N = 100$, Fig. 8-i). The increment fields of both SST and SSH show that an EnKF with larger ensemble generally leads to smoother analysis states.

5. Conclusions and discussion

Numerical prediction of oceanic conditions is of foremost importance for navigation, offshore operations, fisheries, and many other marine activities. However, ocean models are never perfect and can be subject to many sources of uncertainties. Data assimilation combines a prior knowledge of the ocean state from numerical simulations with the observations to provide best possible ocean state estimates along with their uncertainties. The ensemble Kalman filter (EnKF), a popular Monte Carlo data assimilation scheme, is now widely used by the community.

In realistic oceanic applications, the ensemble size is restricted by the computational cost of integrating the numerical ocean model. Using a small ensemble ($10 - 100$) would limit the filter performance, to fit the data at the update step and to provide reliable spread after the forecast step. Although inflation and localization techniques have been proven efficient at mitigating these problems, physical balances in the model could suffer from arbitrary inflation and localization. With the recent tremendous advances in the developments of HPC resources, an EnKF system with large ensembles ($1000-10000$) becomes feasible and less dependent on these auxiliary techniques.

Nevertheless, increasing the ensemble size introduces new issues and difficulties. Obviously the computational load is the first challenge to deal with. For

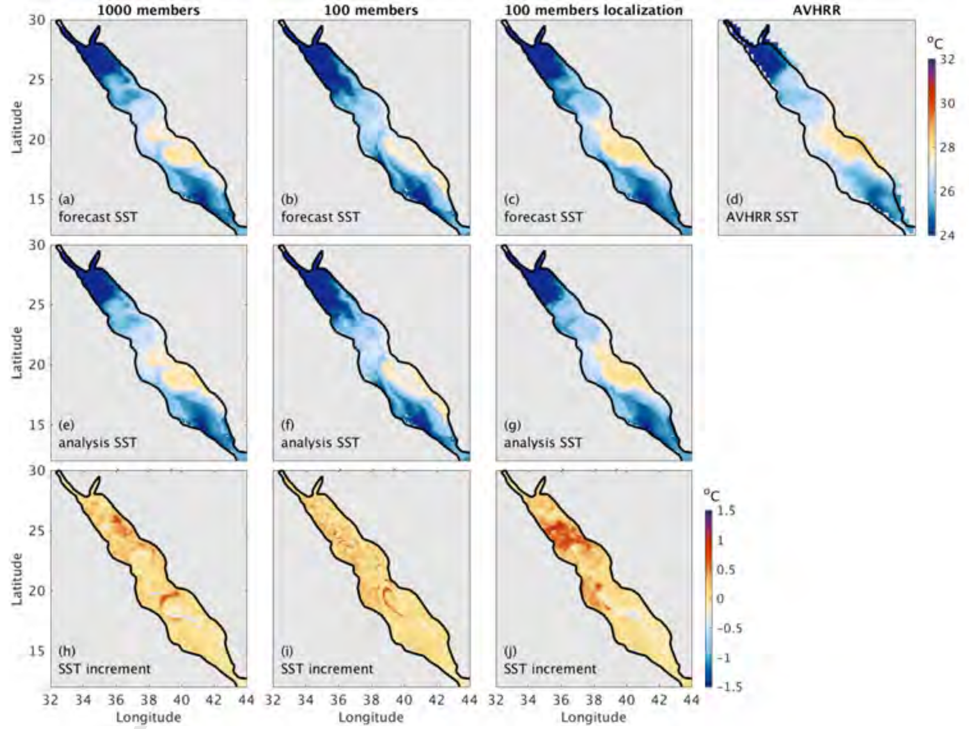

Figure 7: SST forecast/analysis/increment from assimilation experiment with ensemble size of 1000, 100 and 100 with localization compared with gridded AVHRR product on Jan 12, 2006.

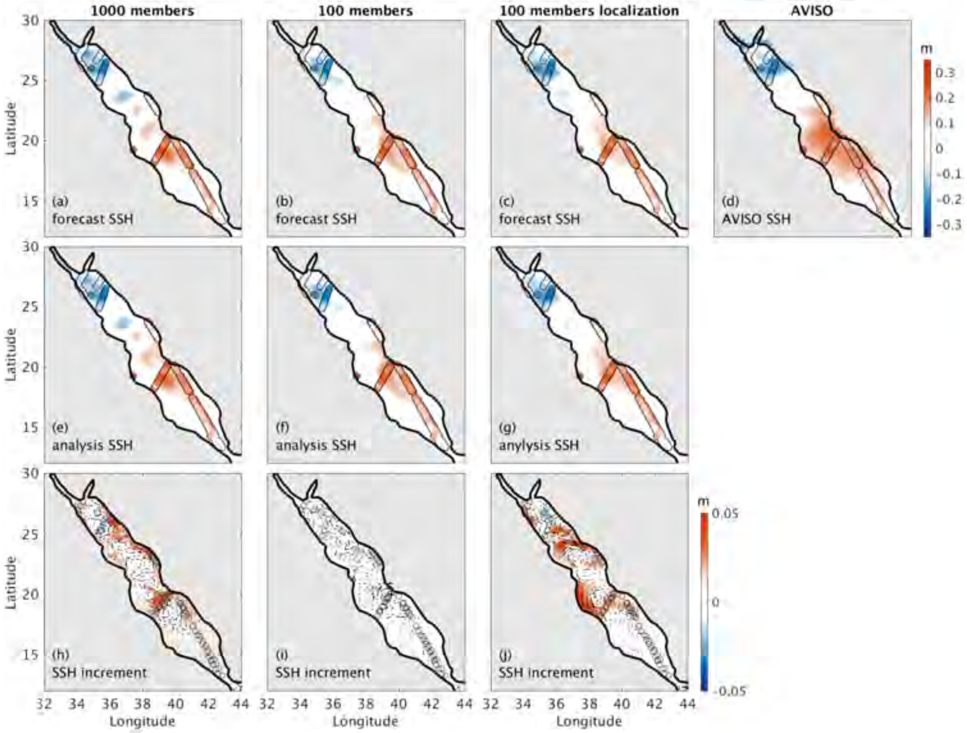

Figure 8: SSH forecast/analysis/increment assimilation experiment with ensemble size of 1000, 100 and 100 with localization compared with gridded AVISO product on Jan 12, 2006, superimposed with the alongtrack altimeter data.

this study, we were able to mitigate this issue with the KAUST world class supercomputer, Shaheen. Another challenge when dealing with a large ensemble is the increasing risk of system failure. The assimilation may indeed impose some increments that are not compatible with the model dynamics, and the system will also be exposed to a higher chance of system failure with large number of heavy jobs running. The interruption of any single member will cause a collapse of the whole ensemble assimilation system.

We developed a fault-tolerant ensemble data assimilation system based on the state-of-the-art MIT general circulation model (MITgcm) for forecasting and the Data Assimilation Research Testbed (DART) for assimilation, and a newly designed scheduler extension, *Decimate*. One key and simple parametrization of *Decimate* consists in describing what to do in case of hardware or numerical failure: here we detailed our choices in this matter. With this parametrization, *Decimate* made it possible for the system to recover from failures, without human intervention. This enabled the implementation of a high resolution ensemble assimilation system for the Red Sea with thousands of ensemble members. This study described the development of the system and its components. From our preliminary experiments with an ensemble with 1000 members we demonstrated significant improvements in the system performances, and as expected less dependence on inflation and localization. We are currently studying the ensemble assimilation system behavior with many more ($5000 - 10000$) members and the early results suggest continuous improved performances and ocean state estimate.

These large-ensemble systems will generate huge amount of data to be stored and analyzed. How to efficiently deal, process and exploit such data is one of the directions of our future research. The possibility of using large ensembles will further open a new paradigm toward exploring the prospect of applying more advanced non-Gaussian filtering schemes for realistic ocean forecasting applications [17, 29, 30].

6. Acknowledgment

The research reported in this manuscript was supported by King Abdullah University of Science and Technology (KAUST) and Saudi ARAMCO, and made use of the resources of the Supercomputing Core Laboratory of KAUST. The data used in this study may be obtained from the authors upon request (ibrahim.hoteit@kaust.edu.sa).

[1] Anderson, J., Hoar, T., Raeder, K., Liu, H., Collins, N., Torn, R., Avellano, A., 2009. The data assimilation research testbed: A community facility. Bulletin of the American Meteorological Society 90 (9), 1283–1296. URL `https://doi.org/10.1175/2009BAMS2618.1`

[2] Anderson, J. L., 2001. An ensemble adjustment kalman filter for data assimilation. Monthly Weather Review 129 (12), 2884–2903.

URL https://doi.org/10.1175/1520-0493(2001)129<2884:AEAKFF>2.0.CO;2

[3] Anderson, J. L., 2003. A local least squares framework for ensemble filtering. Monthly Weather Review 131 (4), 634–642.
URL https://doi.org/10.1175/1520-0493(2003)131<0634:ALLSFF>2.0.CO;2

[4] Anderson, J. L., 2007. An adaptive covariance inflation error correction algorithm for ensemble filters. Tellus A: Dynamic Meteorology and Oceanography 59 (2), 210–224.
URL http://dx.doi.org/10.1111/j.1600-0870.2006.00216.x

[5] Anderson, J. L., 2007. Exploring the need for localization in ensemble data assimilation using a hierarchical ensemble filter. Physica D: Nonlinear Phenomena 230 (1), 99 – 111, data Assimilation.
URL http://www.sciencedirect.com/science/article/pii/S0167278906002168

[6] Anderson, J. L., 2009. Spatially and temporally varying adaptive covariance inflation for ensemble filters. Tellus A 61 (1), 72–83.
URL http://dx.doi.org/10.1111/j.1600-0870.2008.00361.x

[7] Anderson, J. L., Collins, N., 2007. Scalable implementations of ensemble filter algorithms for data assimilation. Journal of Atmospheric and Oceanic Technology 24 (8), 1452–1463.
URL https://doi.org/10.1175/JTECH2049.1

[8] Belyaev, K. P., Tanajura, C. A., O'Brien, J. J., 2001. A data assimilation method used with an ocean circulation model and its application to the tropical atlantic. Applied Mathematical Modelling 25 (8), 655 – 670.
URL http://www.sciencedirect.com/science/article/pii/S0307904X01000038

[9] Burgers, G., van Leeuwen, P., Evensen, G., 1998. Analysis scheme in the ensemble Kalman filter. Monthly weather review 126 (6), 1719–1724.

[10] Edwards, C. A., Moore, A. M., Hoteit, I., Cornuelle, B. D., 2015. Regional ocean data assimilation. Annual Review of Marine Science 7 (1), 21–42, pMID: 25103331.
URL https://doi.org/10.1146/annurev-marine-010814-015821

[11] Evensen, G., 1994. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. Journal of Geophysical Research: Oceans (1978–2012) 99 (C5), 10143–10162.

[12] Evensen, G., 2001. Ocean and Climate Prediction on Parallel Super Computers. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 37–37.
URL https://doi.org/10.1007/3-540-70734-4_6

[13] Fu, W., She, J., Zhuang, S., 2011. Application of an ensemble optimal interpolation in a north/baltic sea model: Assimilating temperature and salinity profiles. Ocean Modelling 40 (3), 227 – 245.
URL http://www.sciencedirect.com/science/article/pii/S1463500311001648

[14] Guo, H.-D., Zhang, L., Zhu, L.-W., 2015. Earth observation big data for climate change research. Advances in Climate Change Research 6 (2), 108 – 117, special issue on advances in Future Earth research.
URL http://www.sciencedirect.com/science/article/pii/S1674927815000519

[15] Hamill, T. M., Whitaker, J. S., 2011. What constrains spread growth in forecasts initialized from ensemble kalman filters? Monthly Weather Review 139 (1), 117–131.
URL https://doi.org/10.1175/2010MWR3246.1

[16] Hamill, T. M., Whitaker, J. S., Snyder, C., 2001. Distance-dependent filtering of background error covariance estimates in an ensemble kalman filter. Monthly Weather Review 129 (11), 2776–2790.
URL https://doi.org/10.1175/1520-0493(2001)129<2776:DDFOBE>2.0.CO;2

[17] Hoteit, I., Luo, X., Pham, D.-T., 2012. Particle kalman filtering: A nonlinear bayesian framework for ensemble kalman filters. Monthly Weather Review 140 (2), 528–542.
URL https://doi.org/10.1175/2011MWR3640.1

[18] Hoteit, I., Pham, D.-T., Blum, J., 2002. A simplified reduced order kalman filtering and application to altimetric data assimilation in tropical pacific. Journal of Marine Systems 36 (1), 101 – 127.
URL http://www.sciencedirect.com/science/article/pii/S092479630200129X

[19] Hoteit, I., Pham, D.-T., Gharamti, M. E., Luo, X., 2015. Mitigating observation perturbation sampling errors in the stochastic enkf. Monthly Weather Review 143 (7), 2918–2936.
URL https://doi.org/10.1175/MWR-D-14-00088.1

[20] Hoteit, I., Pham, D.-T., Triantafyllou, G., Korres, G., 2008. A new approximate solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography. Monthly Weather Review 136 (1), 317–334.
URL https://doi.org/10.1175/2007MWR1927.1

[21] Houtekamer, P. L., Mitchell, H. L., 1998. Data assimilation using an ensemble kalman filter technique. Monthly Weather Review 126 (3), 796–811.
URL https://doi.org/10.1175/1520-0493(1998)126<0796:DAUAEK>2.0.CO;2

[22] Houtekamer, P. L., Mitchell, H. L., 2001. A sequential ensemble kalman filter for atmospheric data assimilation. Monthly Weather Review 129 (1), 123–137.
URL https://doi.org/10.1175/1520-0493(2001)129<0123:ASEKFF>2.0.CO;2

[23] Kalman, R. E., 1960. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering 82(1), 35–45.

[24] Kortas, S., 2017. Decimate beta and development branches. https://github.com/samkos/decimate.

[25] Kortas, S., 2017. Decimate stable branch. https://github.com/KAUST-KSL/decimate.

[26] Kortas, S., 2018. Decimate documentation website. http://decimate.readthedocs.io/.

[27] Kortas, S., 2018. Decimate python package webpage. https://pypi.python.org/pypi/decimate.

[28] Kortas, S., Toye, H., Zhan, P., Hoteit, I., 2018. *Decimate*, a portable and fault-tolerant scheduler extension for handling massive dependent jobs: Application to large scale ocean forecasting. In preparation.

[29] Liu, B., Ait-El-Fquih, B., Hoteit, I., 2016. Efficient kernel-based ensemble gaussian mixture filtering. Monthly Weather Review 144 (2), 781–800.
URL https://doi.org/10.1175/MWR-D-14-00292.1

[30] Liu, B., Gharamti, M., Hoteit, I., 2016. Assessing clustering strategies for gaussian mixture filtering a subsurface contaminant model. Journal of Hydrology 535 (Supplement C), 1 – 21.
URL http://www.sciencedirect.com/science/article/pii/S0022169416000664

[31] Marshall, J., Adcroft, A., Hill, C., Perelman, L., Heisey, C., 1997. A finite-volume, incompressible navier stokes model for studies of the ocean on parallel computers. Journal of Geophysical Research: Oceans 102 (C3), 5753–5766.
URL http://dx.doi.org/10.1029/96JC02775

[32] Marshall, J., Hill, C., Perelman, L., Adcroft, A., 1997. Hydrostatic, quasi-hydrostatic, and nonhydrostatic ocean modeling. Journal of Geophysical Research: Oceans 102 (C3), 5733–5752.

[33] Martin, M., Balmaseda, M., Bertino, L., Brasseur, P., Brassington, G., Cummings, J., Fujii, Y., Lea, D., Lellouche, J.-M., Mogensen, K., Oke, P., Smith, G., Testut, C.-E., Waagb, G., Waters, J., Weaver, A., 2015. Status and future of data assimilation in operational oceanography. Journal of Operational Oceanography 8 (sup1), s28–s48.
URL http://dx.doi.org/10.1080/1755876X.2015.1022055

[34] Meinhold, R. J., Singpurwalla, N. D., 1983. Understanding the kalman filter. The American Statistician 37 (2), 123–127.
URL http://www.jstor.org/stable/2685871

[35] Oke, P. R., Sakov, P., Corney, S. P., Feb 2007. Impacts of localisation in the enkf and enoi: experiments with a small model. Ocean Dynamics 57 (1), 32–45.
URL https://doi.org/10.1007/s10236-006-0088-8

[36] Pham, D. T., Verron, J., Roubaud, M. C., 1998. A singular evolutive extended kalman filter for data assimilation in oceanography. Journal of Marine Systems 16 (3), 323 – 340.
URL http://www.sciencedirect.com/science/article/pii/S0924796397001097

[37] Sakov, P., Bertino, L., Mar 2011. Relation between two common localisation methods for the enkf. Computational Geosciences 15 (2), 225–237.
URL https://doi.org/10.1007/s10596-010-9202-6

[38] Sakov, P., Oke, P. R., 2008. A deterministic formulation of the ensemble kalman filter: an alternative to ensemble square root filters. Tellus A 60 (2), 361–371.
URL http://dx.doi.org/10.1111/j.1600-0870.2007.00299.x

[39] Toye, H., Zhan, P., Gopalakrishnan, G., Kartadikaria, A. R., Huang, H., Knio, O., Hoteit, I., Jul 2017. Ensemble data assimilation in the red sea: sensitivity to ensemble selection and atmospheric forcing. Ocean Dynamics 67 (7), 915–933.
URL https://doi.org/10.1007/s10236-017-1064-1

[40] Verlaan, M., Heemink, A. W., Oct 1997. Tidal flow forecasting using reduced rank square root filters. Stochastic Hydrology and Hydraulics 11 (5), 349–368.
URL https://doi.org/10.1007/BF02427924

[41] Zhan, P., Subramanian, A. C., Yao, F., Hoteit, I., 2014. Eddies in the red sea: A statistical and dynamical study. Journal of Geophysical Research: Oceans 119 (6), 3909–3925.

[42] Zhan, P., Subramanian, A. C., Yao, F., Kartadikaria, A. R., Guo, D., Hoteit, I., 2016. The eddy kinetic energy budget in the red sea. Journal of Geophysical Research: Oceans 121 (7), 4732–4747.

Habib Toye is currently a PhD student at King Abdullah University of Science and Technology (KAUST), in the Computer, Electrical and Mathematical Sciences and Engineering Division (CEMSE). His research focuses on Data Assimilation and its application. He is presently investigating the application of Ensemble Data Assimilation in Oceanography.After receivinghis engineering degree from the EcoleNationaleSupérieured'Informatiqueet de MathématiquesAppliquées de Grenoble (ENSIMAG, Grenoble Institute of Technology) in 2010, he hold a position at INRIA (the French National Institute for computer science and applied mathematics) as a Young graduate engineer for two years.

Dr Samuel Kortas, Computational Scientist at the KAUST Supercomputing Core Laboratory

For 25 years, Samuel Kortas has been working as a research scientist at the Los Alamos National Laboratory, the French Atomic Commission of Energy, Electricité de France, or as a technical leader at Cap Gemini involved in consulting missions for Hewlett-Packard, Alcatel Lucent, ST MicroElectronics or Lexmark. His missions were related to performance tuning in high performance or embedded computing for domain as diverse as ocean modeling, nuclear safety, telecom services, driver programming, or technical survey and realization of proof of concept on strategic market trends. As a benchmark expert, Samuel was involved in the procurement and acceptance of five major supercomputers. Since 2013, as a computational scientist at the KAUST Supercomputing Laboratory, he is simplifying the use of Shaheen for industrial or academic users. He develops tools extending the scheduler capabilities in managing workflows of thousands of dependent jobs making them fault-tolerant to hardware or numerical failures. Samuel received his engineering degree from the EcoleCentrale Paris in 1992 and his PhD in Applied Mathematics from the University of Marseille in 1997.

Peng Zhan is a PhD candidate in the Division of Physical Sciences and Engineering at King Abdullah University of Science and Technology (KAUST). His research interests include physical oceanography, ocean modeling and data assimilation.