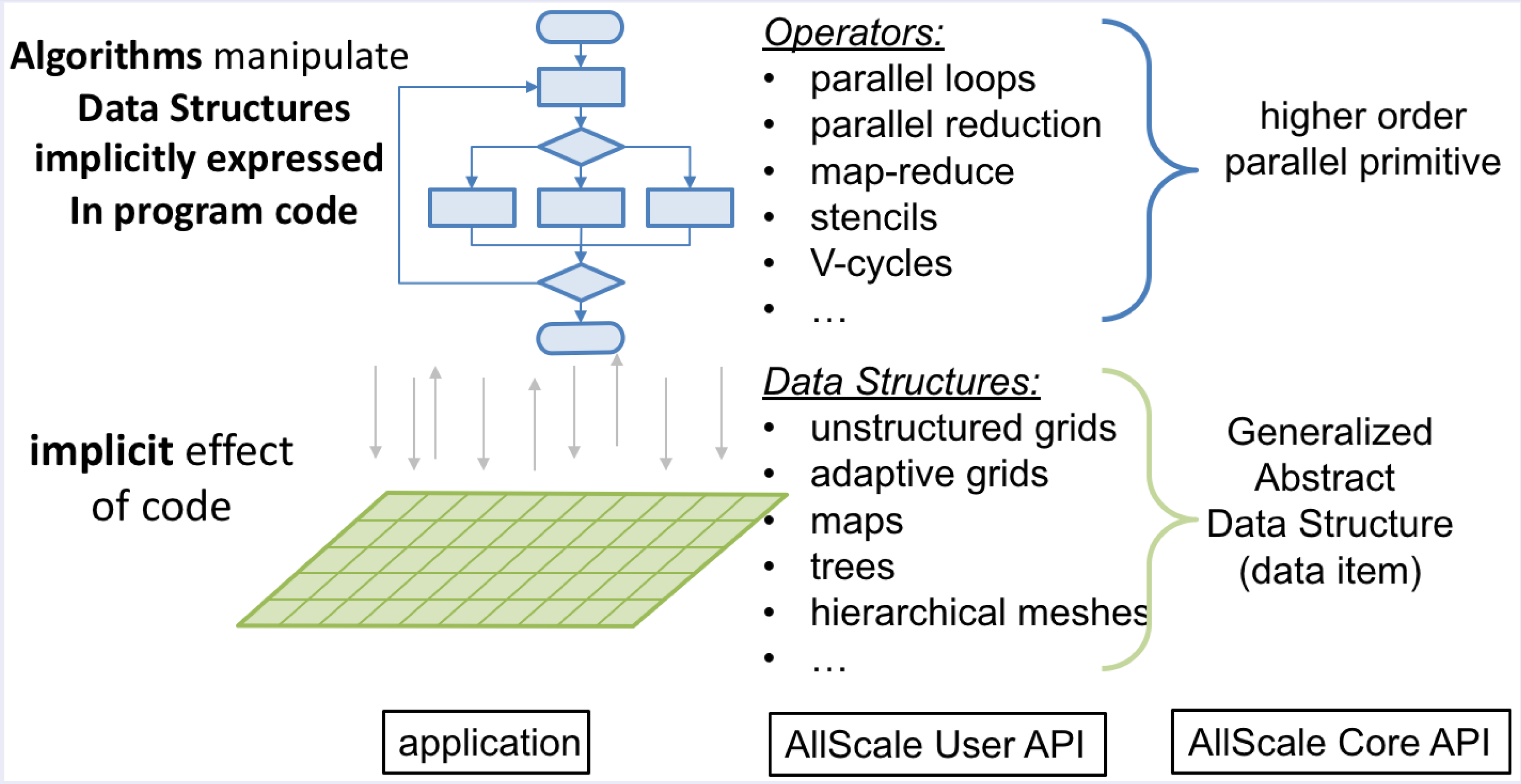
Localised data assimilation framework to simulate advection-diffusion processes within an advanced parallel development environment

Albert Akhriev, Emanuele Ragnoli, Fearghal O'Donncha, IBM Research — Ireland

Introduction

This study presents a framework for localised data assimilation applied to an advection-diffusion based model. The scheme is developed within a novel programming environment aimed at facilitating efficient code development by leveraging advanced "separation of responsibilities" principles. The front-end AllScale API provides the developer with a simple C++ development environment and a suite of parallel constructs that denote tasks to be operated concurrently.

AllScale's Approach



AllScale User API

AllScale User API provides a set of user-friendly constructs for the composition of parallel applications. The list of constructs comprises:

- parallel control flow primitives:
  - parallel loops with support for fine-grained dependency.
    - over numerical ranges (e.g. 1–10).
    - over ranges defined by C++ random access iterators.
  - parallel reductions as an extension to parallel loops.
  - a stencil API utilizing a recursive space-time decomposition schema.
  - an adaptive grid refinement stencil as an extension to the standard stencil.
- data structures:
  - multi-dimensional static and dynamically sized grids.
  - an adaptive refineable grid.
  - an unstructured multi-grid mesh.

Problem Formulation

Here we model oil spill evolution applying the problem to the marine environment and assuming that we can: (1) measure the concentration of contaminant (e.g. dispersion of oil spill) at *sparse* sensor locations; (2) have information on the speed and direction of the current. The data assimilation problem can be formulated as follow: find a reasonably good approximation to the distribution of contaminant in the domain as a function of space and time given only a physical model and sparse observations.

Mathematical Problem

Propagation of contaminant in 2D (marine) environment is modelled by advection-diffusion equation, where density is defined as a function of space and time $u = u(x, y, t)$:

$$\frac{\partial u}{\partial t} = D\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - v_x\frac{\partial u}{\partial x} - v_y\frac{\partial u}{\partial y}, \qquad (1)$$

$$\text{s.t.}\quad u|_{t=0} = \delta(x - x_c, y - y_c), \quad u|_{\partial\Omega} = 0.$$

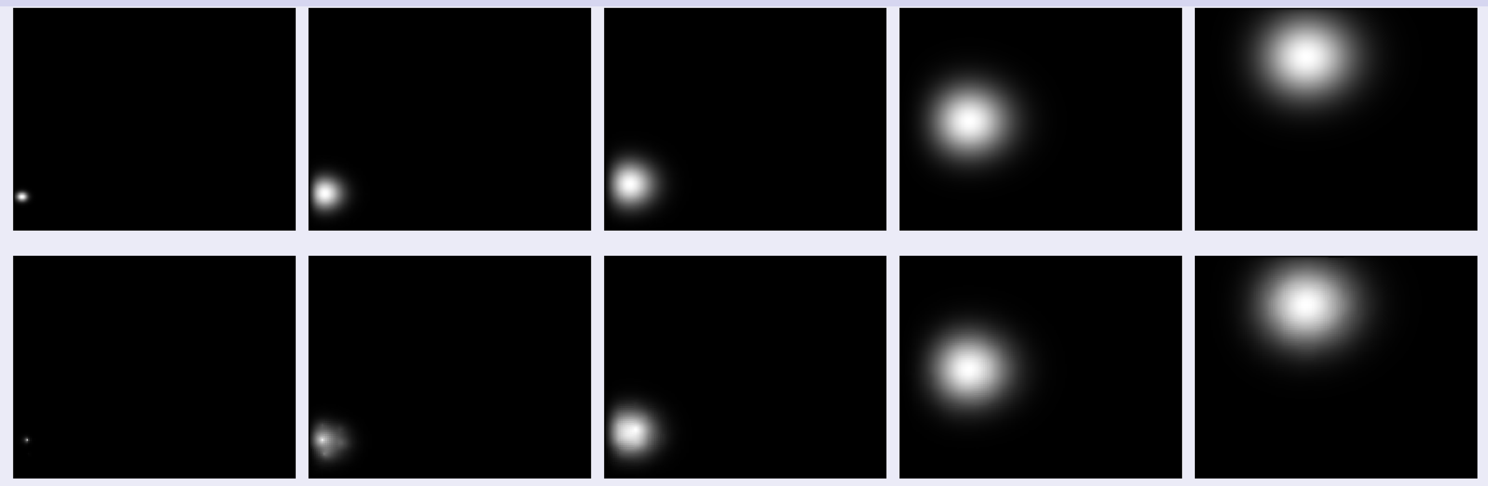
where D is diffusion coefficient, $v_x = v_x(x, y, t)$, $v_y = v_y(x, y, t)$ are the flow velocity components, the initial condition is defined as point source at some location (x_c, y_c), and the boundary condition is of homogeneous Dirichlet type.

Mathematical Details

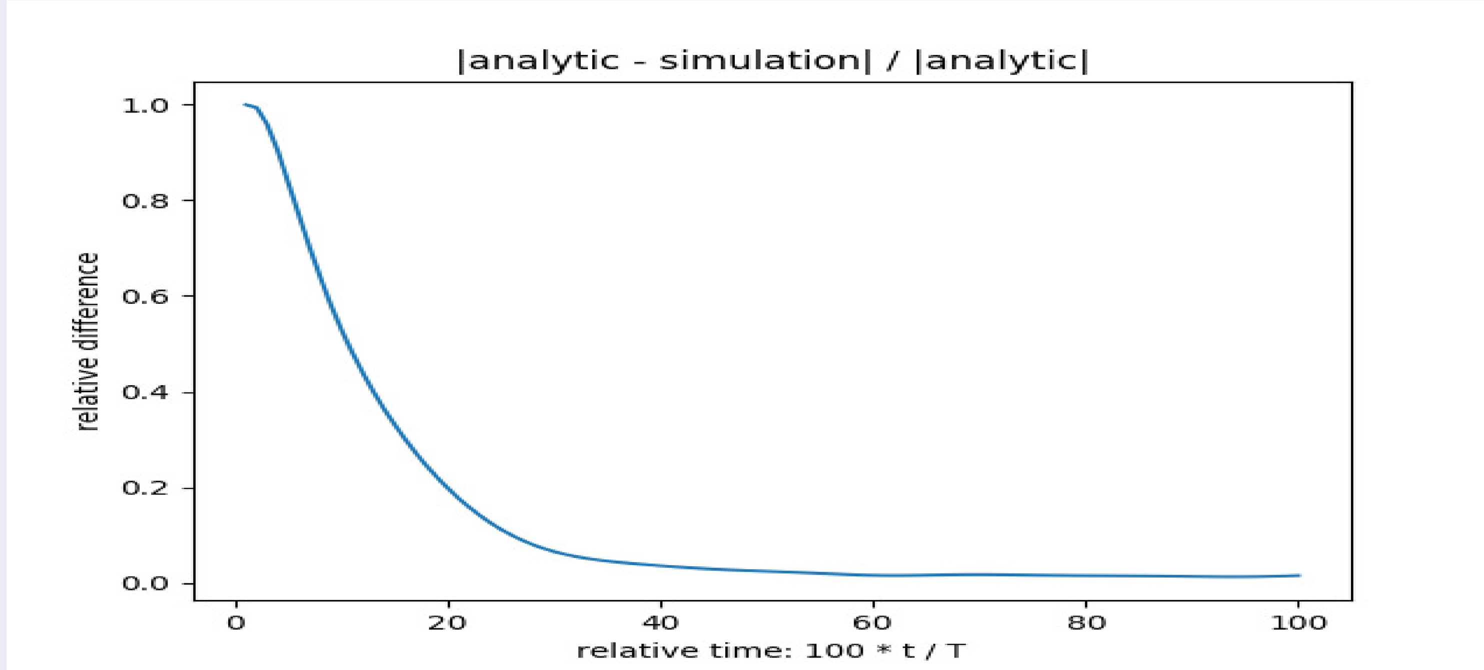
Domain decomposition and Kalman filtering (applied per sub-domain) are among the most important features of proposed method:

- Domain decomposition reduces the computational expense of data-assimilation and facilitates nested recursive parallelism by localising state updates to individual sub-domains.
- Kalman filter implements a mechanism for assimilation sensor data into simulation process. Equation (1) provides *a priory* estimation for Kalman filter, where the density $u(x, y, t)$ inside a sub-domain constitutes a state vector.

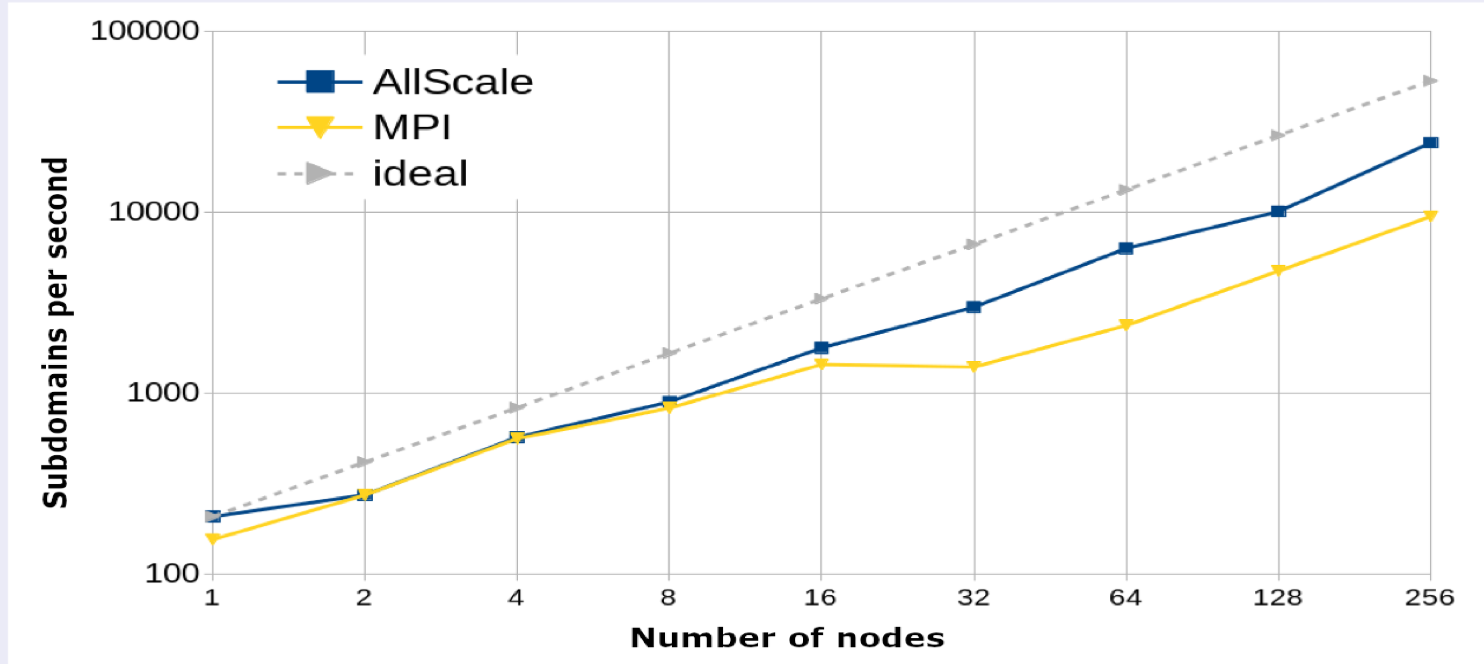
Ground-truth (top) vs. Simulation (bottom)



Modelling Accuracy



Scalability: Intel Xeon E5-2630v4



Scalability: Cray XC40