In Situ Flow Analysis for MPAS-Ocean Simulations

Hanqi Guo (ANL), Mukund Raj (ANL), Orcun Yildiz (ANL), Wenbin He (OSU), Jiang Zhang (PKU),
Emil Constantinescu (ANL), Han-Wei Shen (OSU), Phil Wolfram (LANL), and Tom Peterka (ANL)

U.S. DEPARTMENT OF ENERGY | Office of Science

CANGA

Argonne NATIONAL LABORATORY

Overview

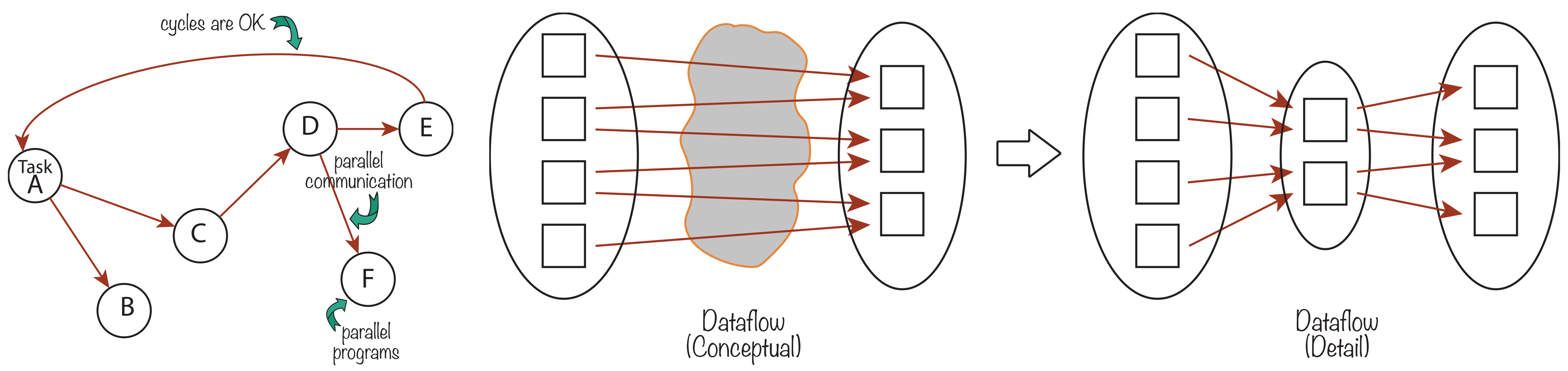
Next generation earth system models that scale to emergent HPC hardware systems will need in situ analysis coupled to climate components. In situ analysis can avail scientists of early and higher fidelity results than postprocessing.

In this work, we are connecting in situ flow analyses to MPAS-Ocean simulations using Decaf [1], a data flow system for the coupling of tasks in a workflow. By doing this, we augment the simulations with a scalable general purpose time-dependent Lagrangian particle tracer that supports both deterministic and stochastic flows. The particle tracer also features the state-of-the-art load balancing algorithms [3, 4] and CPU/GPU task parallelism [2].

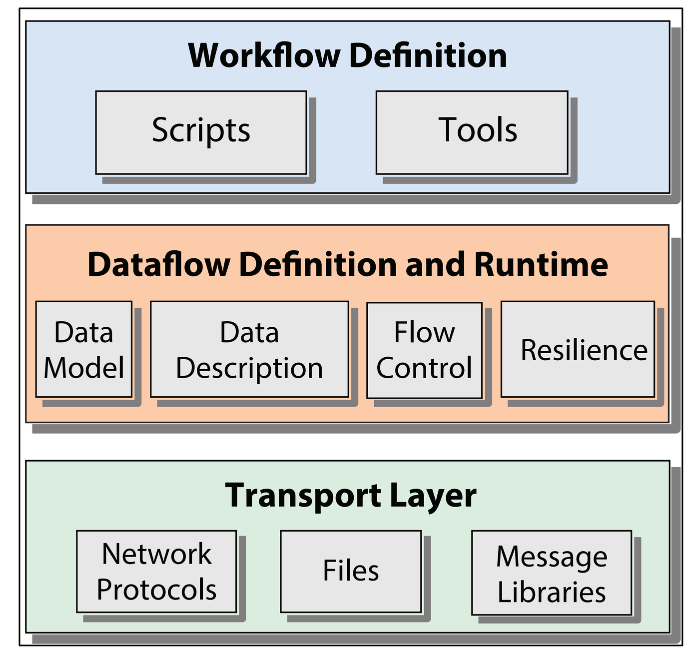
Decaf: Decoupled Dataflow for In Situ Workflows

A **workflow** is a directed graph of tasks and communication between them. Graph nodes are the tasks and graph links are the communication. Both nodes and links are actually parallel processes.

A **dataflow** is the communication over the links in a workflow. Workflows consist of high-level tasks; dataflows consist of communication between MPI process ranks.



The workflow software stack consists of a workflow definition layer, a dataflow definition layer, and a transport layer. The core of Decaf is the dataflow layer. Other workflow definition tools can be used, and transport layers are provided by systems software such as MPI.



Preliminary Particle Tracing Results for MPAS-O

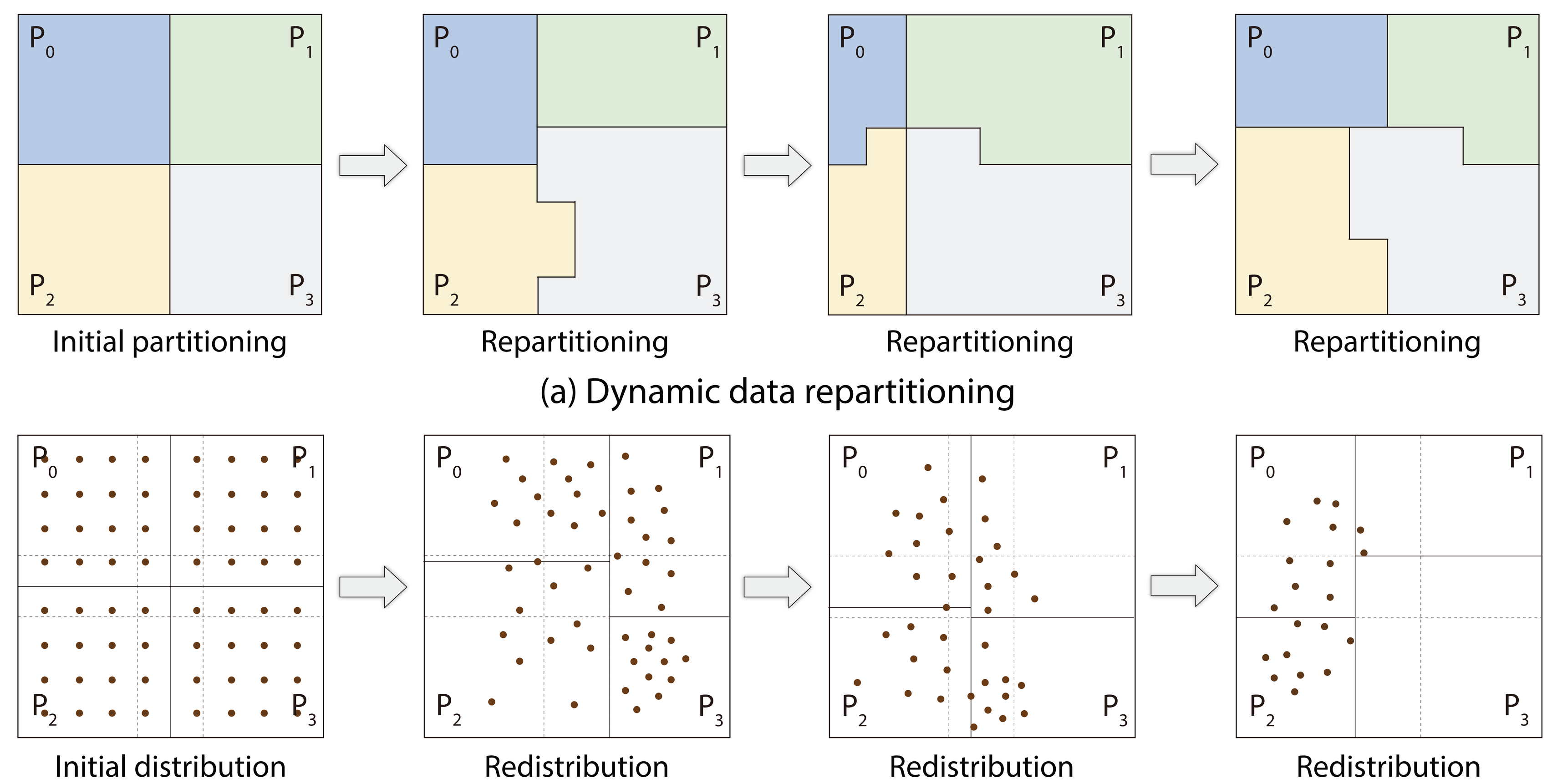
Streamlines and eddies in the SOMA 32 km test case

Dynamic Load Balanced Particle Tracing

We propose two dynamic load-balancing algorithms to trace many particles for flow visualization and analysis. Both methods significantly improve the load balancing and scale up to 8,192 processes in our test.

Dynamic data repartition: we periodically repartition the domain decomposition based on the workload prediction [3]. The prediction is based on the high order data access patterns in the flow.

Dynamic particle redistribution: Each process is assigned with a statically partitioned, axis-aligned data block that partially overlaps with neighboring blocks in other processes. We dynamically determine the k-d tree leaf node that bounds the active particles for load balanced computation.

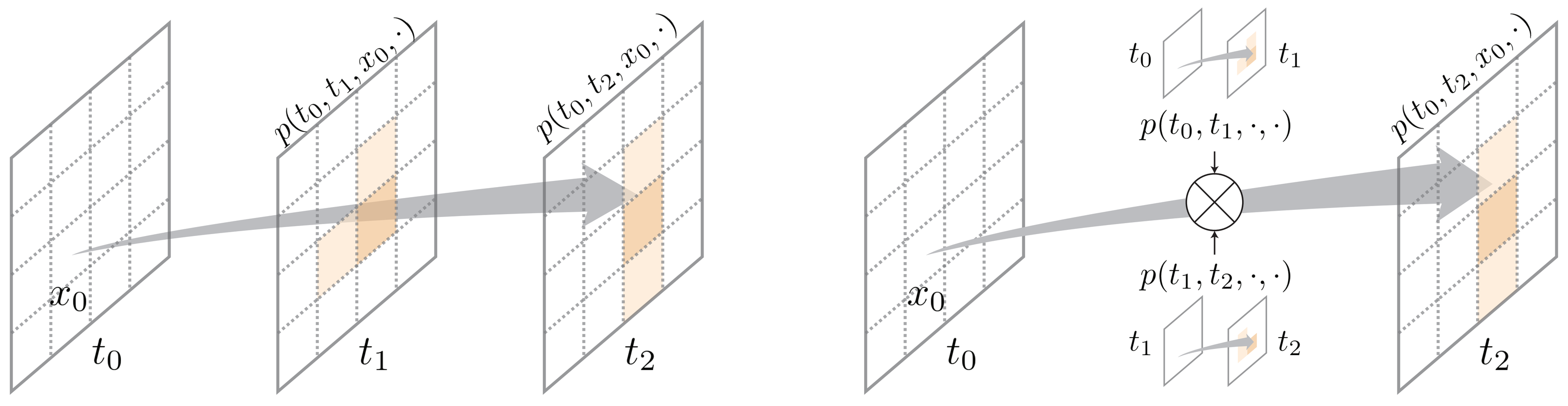


(a) Dynamic data repartitioning

(b) Dynamic particle redistribution based on constrained k-d tree decomposition

Decoupled Stochastic Particle Tracing

Computing flow maps for uncertain flows is extremely expensive because it requires many Monte Carlo runs to trace densely seeded particles in the flow. We decoupled the time dependencies so that we can process short time subintervals independently in parallel. We also derived the theoritical error bound of this approach [2].



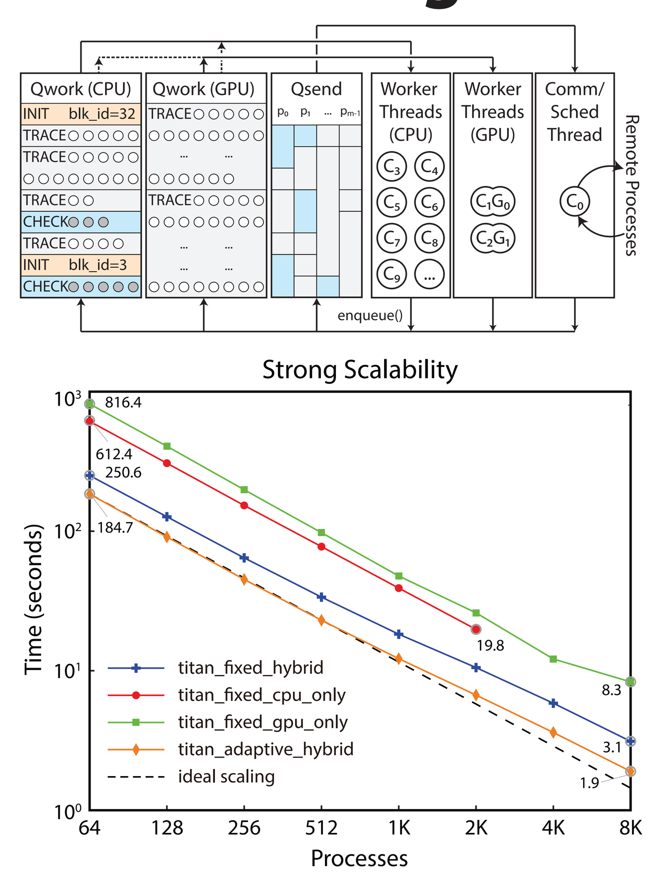
Direct Stochastic Flow Map Estimate

Decoupled Stochastic Flow Map Estimate

CPU/GPU Co-Processing for Particle Tracing

We tailored the task model for parallel particle tracing to achieve high efficiency in MPI/thread environments and to enable CPU/GPU coprocessing. Tasks are scheduled by a thread pool, which overlaps asynchronous and nonblocking communication with computation.

We show the scalability on two supercomputers, Mira (up to 256K Blue Gene/Q cores) and Titan (up to 128K Opteron cores and 8K GPUs), that can trace billions of particles in seconds [2].



Software and Publications

[1] The Decaf project, https://github.com/tpeterka/decaf

[2] H. Guo, W. He, S. Seo, H.-W. Shen, E. Constantinescu, C. Liu, and T. Peterka. “Extreme-scale stochastic particle tracing for uncertain unsteady flow visualization and analysis.” IEEE Transactions on Visualization and Computer Graphics, 2018. (Accepted)

[3] J. Zhang, H. Guo, F. Hong, X. Yuan, and T. Peterka, “Dynamic load balancing based on constrained k-d tree decomposition for parallel particle tracing.” IEEE Transactions on Visualization and Computer Graphics, 24(1):954-963, 2018.

[4] J. Zhang, H. Guo, X. Yuan, and T. Peterka, “Dynamic data repartitioning for load-balanced parallel particle tracing.” In Proceedings of IEEE Pacific Visualization Symposium, pages 86-95, 2018.