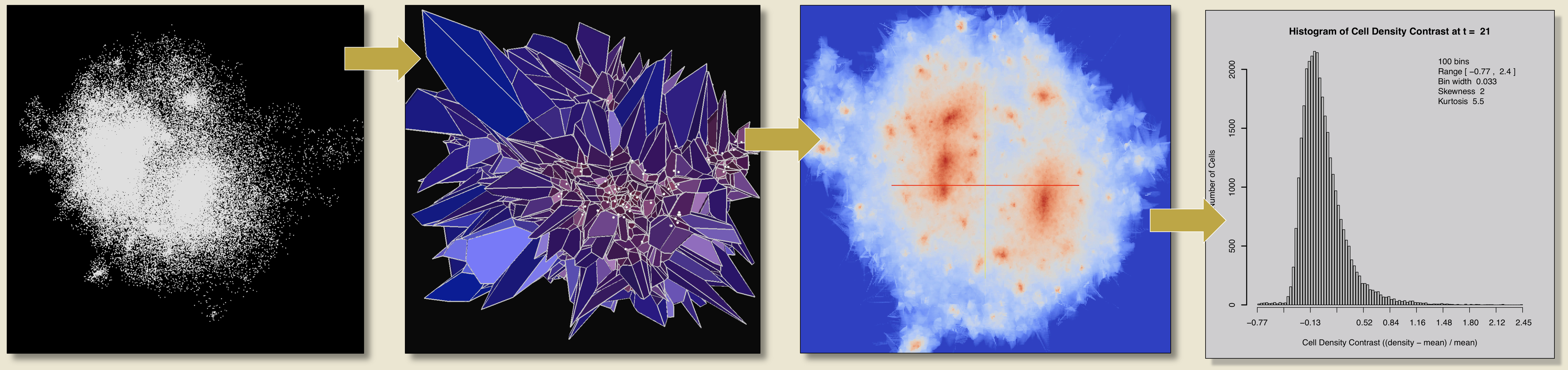
High-Performance Decoupling of Tightly Coupled Data Flows

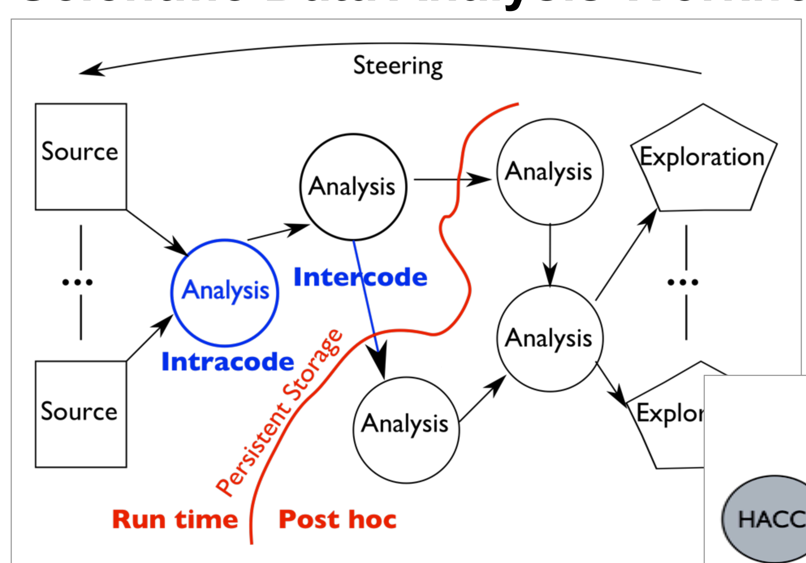


An example workflow in cosmology transforms raw particle positions in an N-body simulation into a Voronoi mesh, which is then used to deposit particle density onto a regular grid. Subsequent density statistics are computed in postprocessing.

Motivation

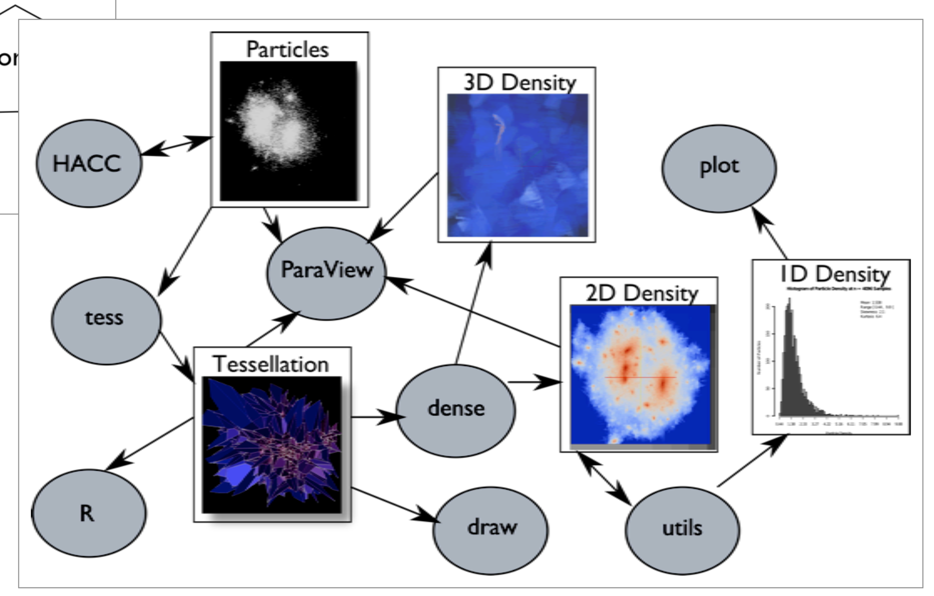
The need to distill enormous amounts of data into useful knowledge is pushing the limits of computational science. Tightly coupled data analysis and data generation--making the analysis interdependent and closely coordinated with the computation--limits the flexibility provided by individual modules. The Decaf project explores a hybrid approach that combines both types of coupling---tight and loose---in effect decoupling tightly coupled applications.

Scientific Data Analysis Workflows



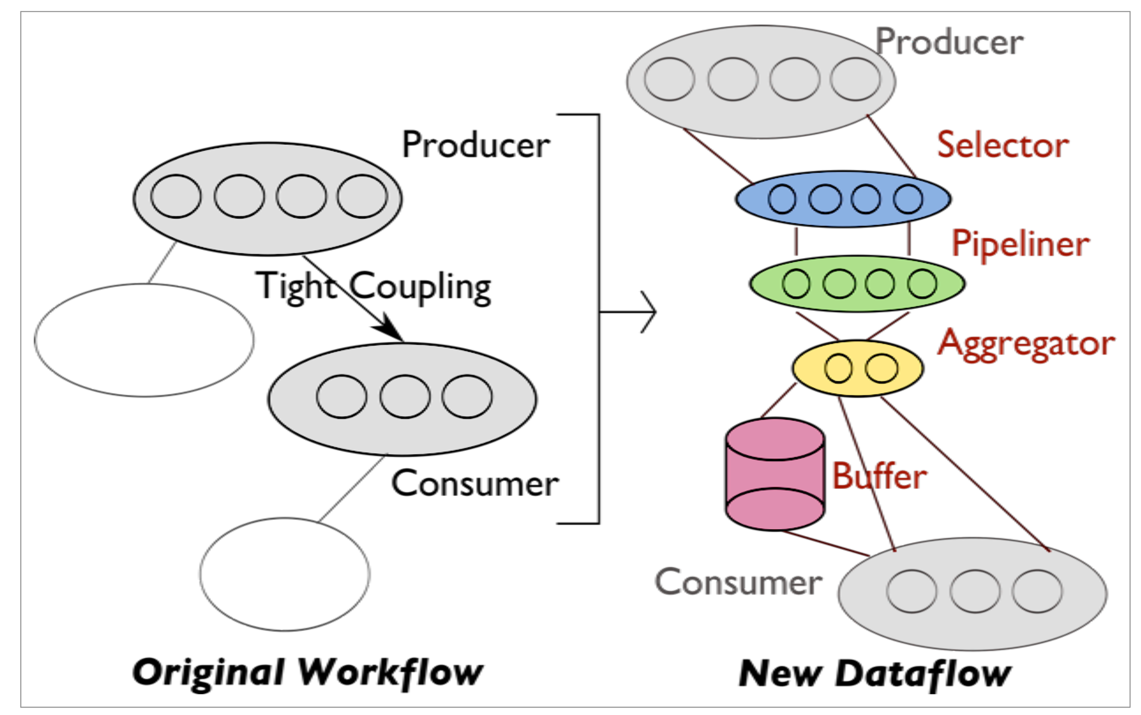
Left: Generic analysis workflow graph, single or ensemble sources and multiple users. Results are written to persistent storage at the red line that partitions the graph into operations done at run time (in situ) and post hoc.

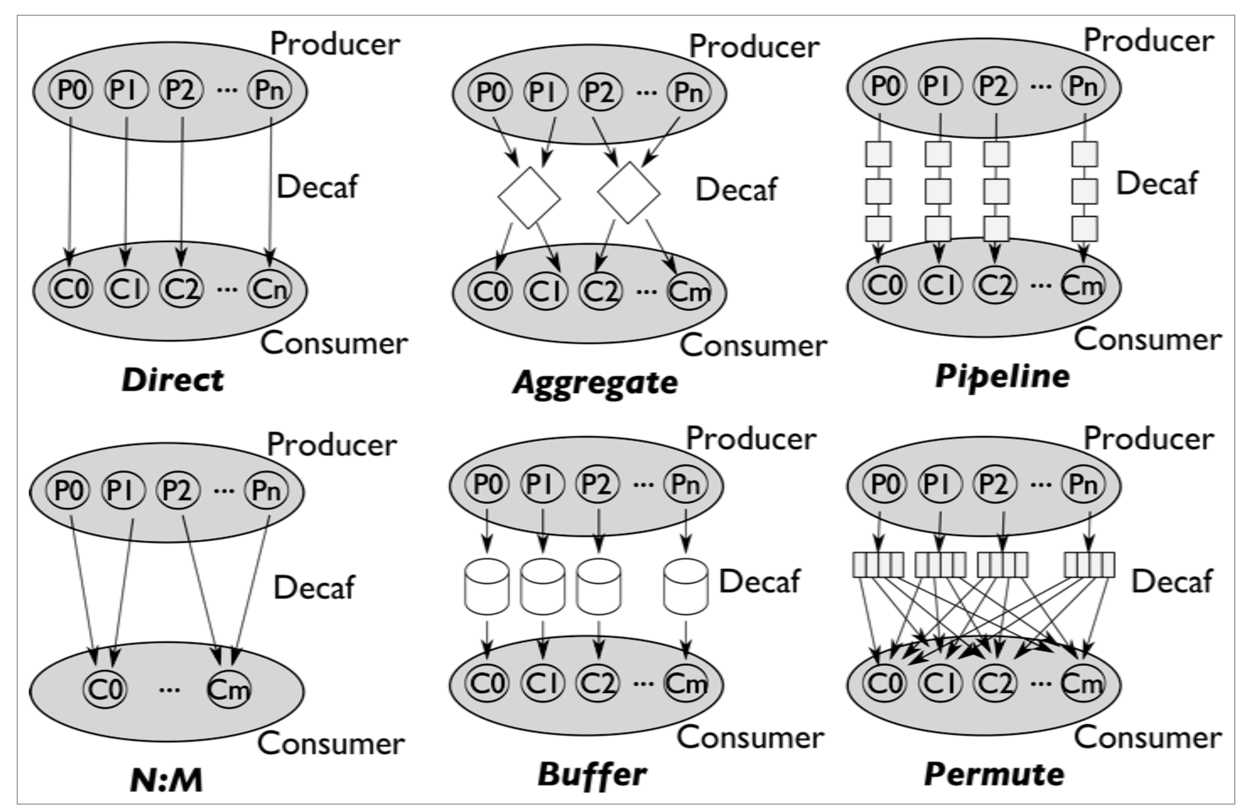
Right: Workflow network for data analysis of particle density from an N-body cosmological simulation is a specific example of the generic graph above.



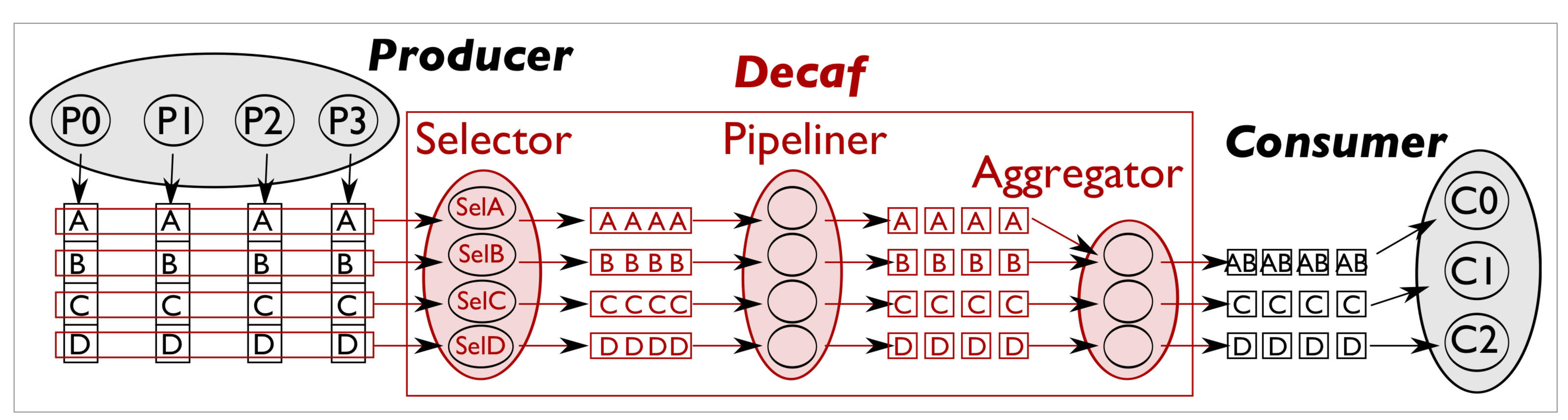
The Decaf Concept

Right: From a workflow to a dataflow, Decaf will automatically expand links with a small set of primitives composed in an optimal way to satisfy user requirements.



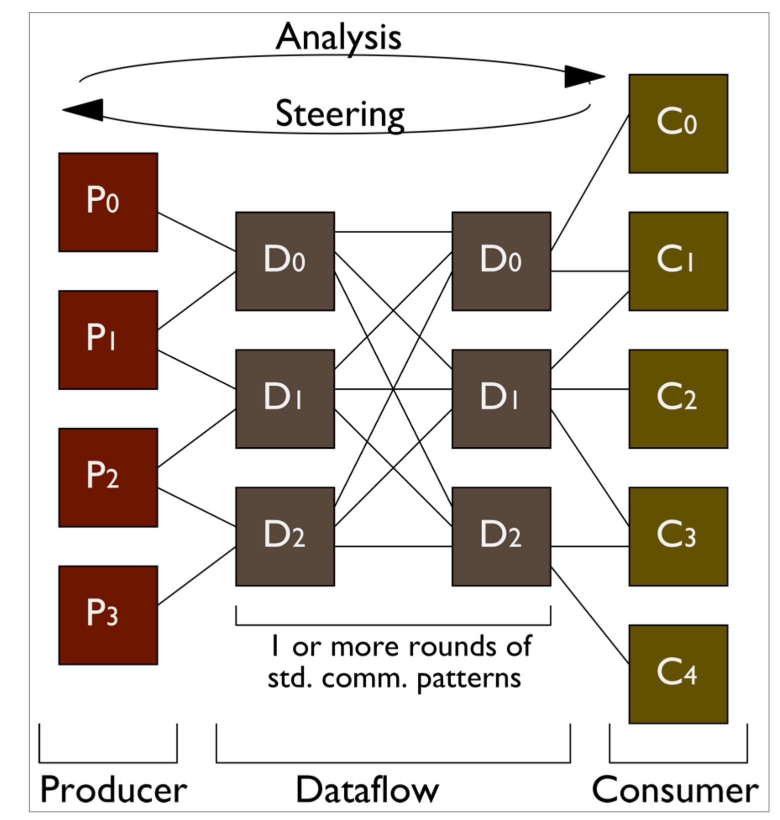


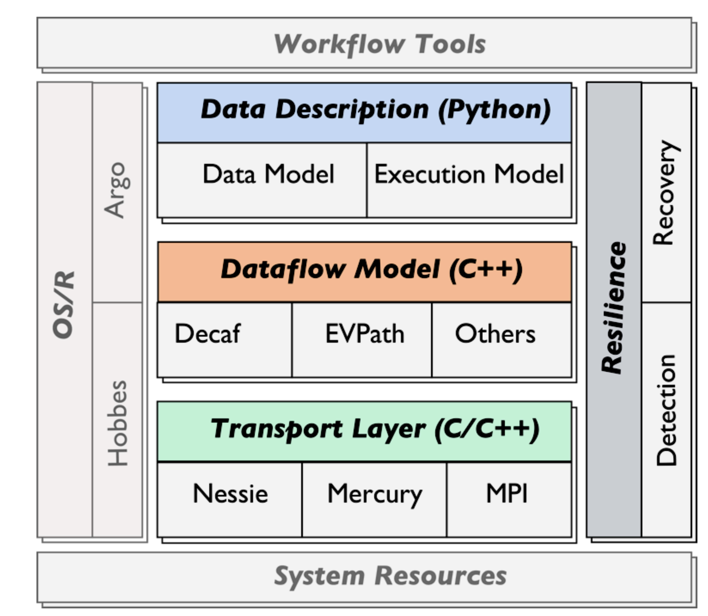
Left: Six major Decaf coupling modes.
Bottom: Coupling example with a data permutation and pipelineable consumer. The producer generates columns of A,B,C,D while the consumer requires rows of A, rows of B, etc. as soon as they are available.



Productivity and Performance

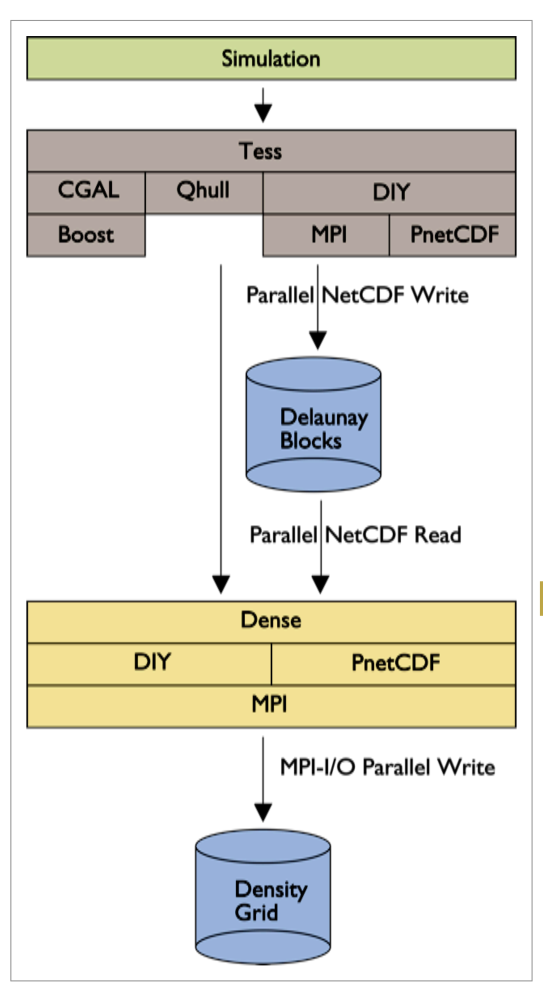
Left: Intermediate dataflow nodes couple producer to consumer. The dataflow can be a simple NOOP or a complete parallel program performing complex data transformations.
Bottom: Decaf software stack.

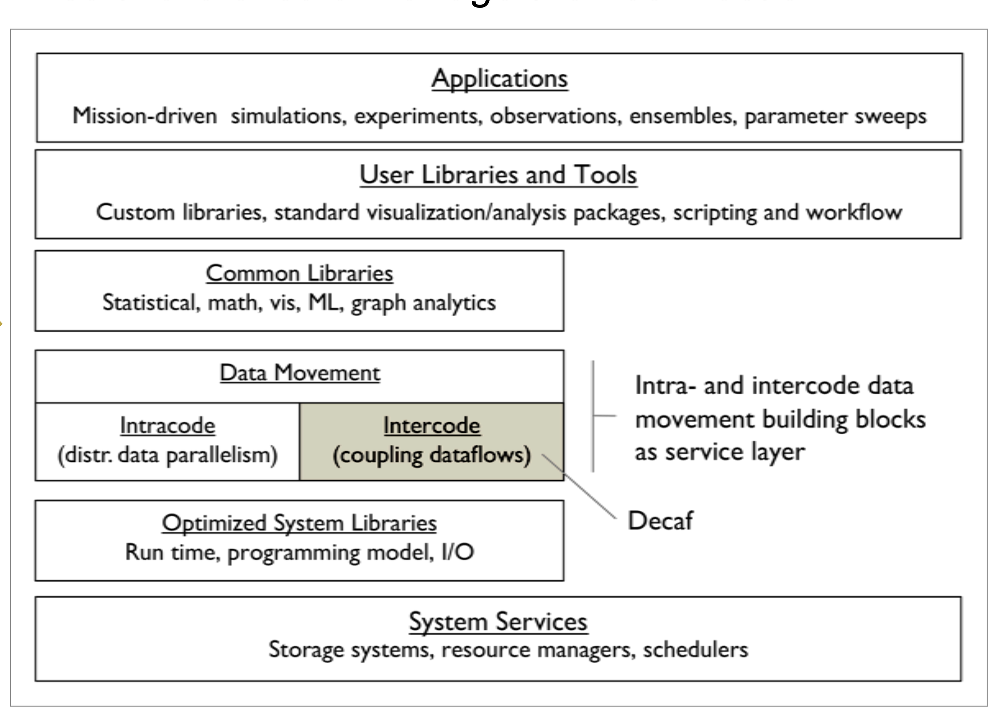




The Need for Middleware

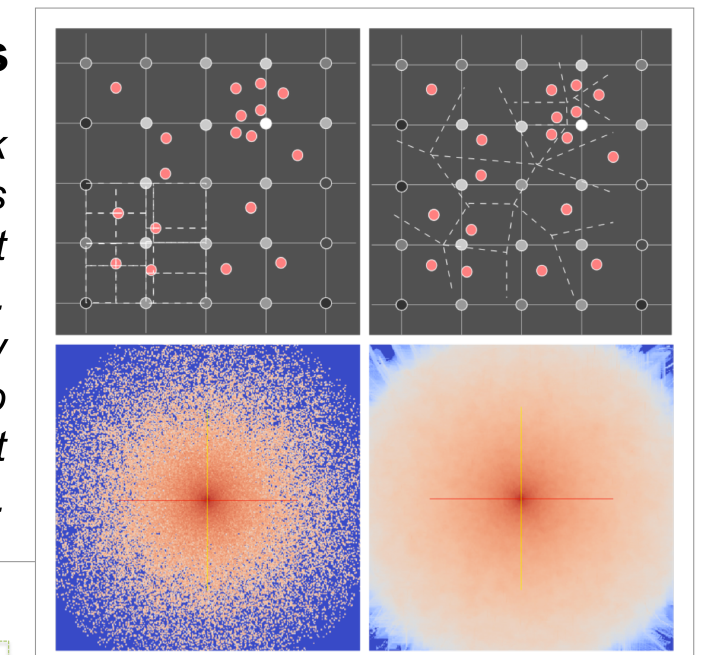
Left: (Today) Tightly-coupling in situ analysis by writing custom main programs for each combination of producer and consumer and tuning the producer to the consumer and vice versa. Bottom: Producers and consumers ought to be written independently, and generic coupling software should manage their connection.

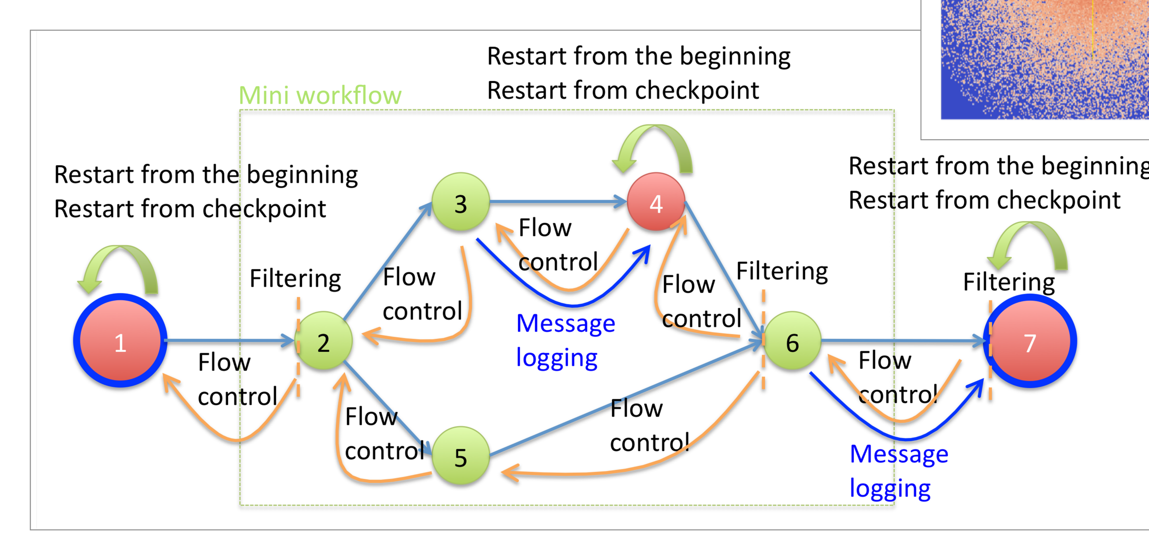




Resilience to Hard and Soft Faults

Right: Silent data corruption in analysis task validated with an auxiliary method, usually less expensive and less accurate, yet able to detect soft errors.
Bottom: Modeling the dataflow and optimally adding replication and roll back mechanisms to recover from hard (fail stop) errors and soft errors detected above.





Tom Peterka, Jay Lofstead, Franck Cappello, Patrick Widener, Florin Isaila, Lokman Rahmani, Hadrien Croubois, Guillaume Aupy