New information technologies, Computing simulation

Melissa: Large Scale In Transit Sensitivity Analysis Avoiding Intermediate Files

In Proceedings of Super Computing Conference, Denver, Colorado USA, November 2017 (SC'17)

T. Terraz[1], A. Ribés[2], Y. Fournier[2], B. Iooss[2] and B. Raffin[1]

[1]INRIA Grenoble [2]EDF R&D

Context

Multiple simulation runs (sometimes several thousand) are required to compute sound statistics for global **sensitivity analysis**. Current practice consists in running all the necessary instances with different set of input parameters, store the results to disk, often called ensemble data, to later read them back from disk to compute the required statistics. The amount of storage needed may quickly become overwhelming, with the associated long read time that makes statistic computing time consuming. To avoid this pitfall, scientists reduce their study size by running low resolution simulations or down-sampling output data in space and time. Today terascale and tomorrow exascale machines offer compute capabilities that would enable large scale sensitivity studies. But they are unfortunately not feasible due to this storage issue.

MELISSA

We propose a new approach to compute Sobol' indices at large scale by avoiding to store the intermediate results produced by the multiple parallel simulation runs. The key enabler is a new iterative formulation of Sobol' indices. It enables to update Sobol' indices on-the-fly each time new simulation results are available.

To manage the simulation runs as well as the in transit computation of iterative statistics, we developed a full framework, called Melissa, built around a fault tolerant parallel client/server architecture. The parallel server is in charge of updating the Sobol' indices. Parallel simulations, actually groups of simulations in this work, run independently. Once a simulation starts, it connects to the server and forwards the output data available at each timestep (or at a sufficiently resolved sampling in time) for updating statistics. These data can then be discarded. The benefits of our approach are multiple:

- **Storage saving**: zero intermediate files and a memory requirement on the server side in the order of the outputs of one simulation run.
- **Time saving**: simulations run faster when sending data to the server than when writing their results to disk, and our one-pass algorithm does not need to read back some huge amount of data from disk to compute the Sobol' indices.
- **Ubiquitous**: performance and scalability gains enable to compute ubiquitous multidimensional and time varying Sobol' indices, i.e. everywhere in space and time, instead of providing statistics for a limited sample of probes as usually done.
- **Adaptive**: simulation groups can be defined, started or interrupted on-line according to past runs behavior or the statistics already computed.
- **Fault tolerance**: only some lightweight bookkeeping and a few heartbeats are required to detect issues and restart the server or the simulations, with limited intermediate result loss.
- **Elasticity**: simulation groups are independent and connect dynamically to the parallel server when they start. They are submitted as independent jobs to the batch scheduler. Thus, the scheduler can adapt the resources allocated to the application during the execution.

Contact EDF R&D :
Alejandro RIBES, alejandro.ribes@edf.fr

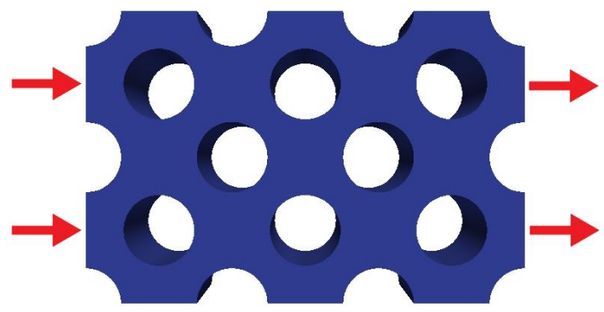

FIGURE 1
Use case: water flows from the left, between the tube bundle, and exits to the right.

Experimental results

We validated our implementation on a fluid mechanics use case simulating a water flow in a tube bundle (fig. 1). The mesh is composed of 9603840 hexahedra. We generated a **multi-run sensitivity study** by simulating the injection of a tracer or dye along the inlet, with 2 independent injection surfaces, each defined by three varying parameters:

1. dye concentration on the upper inlet,
2. dye concentration on the lower inlet,
3. width of the injection on the upper inlet,
4. width of the injection on the lower inlet,
5. duration of the injection on the upper inlet,
6. duration of the injection on the lower inlet.

The experiments presented in this poster ran on Curie, ranked 74th at the top500.org of November 2016. We present the results of a study that took 1h27 wall clock time, 34082 CPU hours for the simulations and 742 CPU hours for the server (2.1% of the total CPU time). At the peak, 55 simulation groups were running simultaneously, on **a total of 28672 cores**. Each Melissa Server process received an average of about 1000 messages per minute during these peaks. Memory usage was about 491GB on the server side, or 15.3GB per node.

During this study, **Melissa Server treated 48 TB of data** coming from the simulations. In a classical study, all these data would be output to the filesystem, and read back to compute statistics.

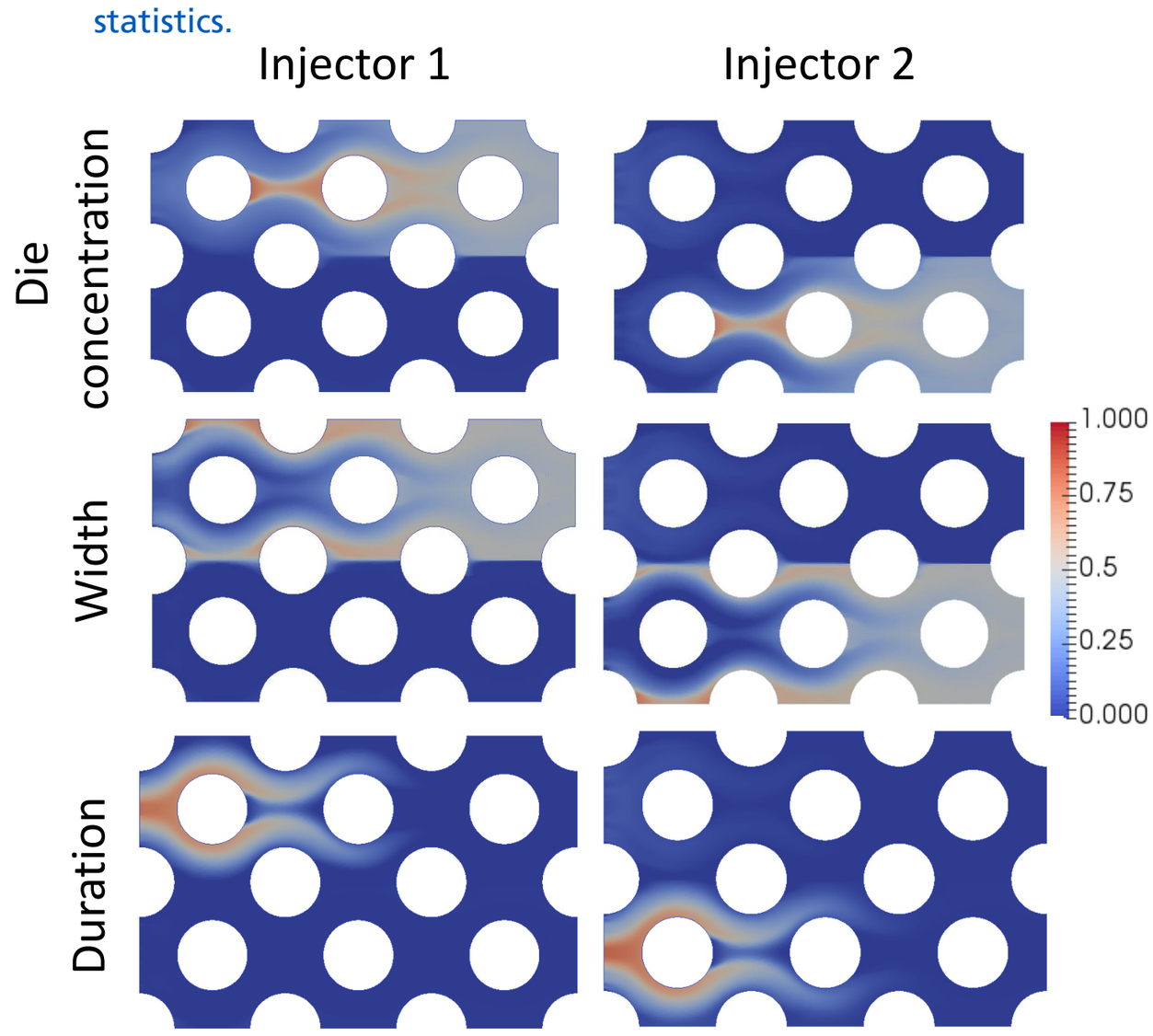

FIGURE 2
First order Sobol' index maps on a slice of the mesh at timestep 80. The left column corresponds to the Sobol' indices for the upper injector while right corresponds to the bottom injector. All maps are scaled between zero (blue) and one (red).