

Large Scale Computation of Quantiles using MELISSA

Alejandro Ribes

EDF Lab Paris-Saclay
91120 Palaiseau, France
alejandro.ribes@edf.fr

Théophile Terraz

Univ. Grenoble Alpes Inria,
CNRS, Grenoble INP, LIG
38000, Grenoble, France
theophile.terraz@inria.fr

Yvan Fournier

EDF Lab Chatou
78401 Chatou, France
yvan.fournier@edf.fr

Bertrand Iooss

EDF Lab Chatou
78401 Chatou, France
bertrand.iooss@edf.fr

Bruno Raffin

Univ. Grenoble Alpes Inria,
CNRS, Grenoble INP, LIG
38000, Grenoble, France
bruno.raffin@inria.fr

ABSTRACT

Quantiles are important order statistics for analysis tasks such as outlier detection or computation of non-parametric confidence intervals. Quantiles being order statistics, the classical approach for their computation requires availability of the full sample before ranking it. This approach is not suitable at exascale. Large ensembles would need to gather a prohibitively large amount of data. We propose an iterative approach based on the stochastic quantile algorithm of Robbins-Monro. We rely on the Melissa framework, a file avoiding, adaptive, fault tolerant and elastic framework in order to compute in transit ubiquitous quantiles. Quantiles are updated on-the-fly as soon as the in transit parallel server receives results from one of the running simulations. We run 80000 fluid dynamics parallel simulations of 6M hexahedra and 100 timespts. They were executed on up to 4800 cores, avoiding 288 TB of file storage. We produce ubiquitous spatio-temporal maps of quantiles and inter-quantile based intervals.

CCS CONCEPTS

Computing methodologies — Massively parallel and high-performance simulations Computing methodologies — Model development and analysis Mathematics of computing — Statistical software

KEYWORDS

Multi-run simulations, ensemble analysis, iterative quantiles

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

WOODSTOCK'18, June, 2018, El Paso, Texas USA

© 2018 Copyright held by the owner/author(s). 978-1-4503-0000-0/18/06...\$15.00

<https://doi.org/10.1145/1234567890>

Context

On-line analytics for large scale numerical simulations has demonstrated its potential to contain the I/O bottleneck, improve simulation and analytics performance, and overall reduce the human time scientists spend in handling large data sets. A key enabler is the availability of one-pass analytics algorithms, i.e. algorithms that can operate on a reduced window of data recently produced by the simulation to avoid the need for massive storage and its associated performance pitfalls. In previous work we proposed the Melissa framework for the on-line data aggregation of high resolution ensemble runs [1]. As soon as each available simulation provides the results produced to a set of staging nodes that process them to update the statistics on a first-come first-served basis thanks to the one-pass algorithm. This in transit processing mode enables to fully avoid storage of intermediate data on disks.

When considering the computation of quantiles, it can be surprising, for the non-initiated, to learn that this computation can be performed iteratively. Indeed, quantiles being order-statistics, the straightforward and classical approach consists on ordering the sample then finding the appropriate quantiles. This strategy necessitates the storage of the full-sample thus making its application in our context impossible. However, stochastic algorithms, such as Robbins-Monro, are devoted to recursive estimation of statistical quantities, and then allow for the iteratively computation of quantiles [2]. We compute ubiquitous multidimensional and time varying quantiles, i.e. everywhere in space and time, instead of providing statistics for a limited sample of probes as usually done.

Fluid Simulation with Code_Saturne

We used Code_Saturne [3], an open-source computational fluid dynamics tool designed to solve the Navier-Stokes equations, with a focus on incompressible or dilatant flows and advanced turbulence modeling. Code_Saturne relies on a finite volume discretization and allows the use of various mesh types, using an unstructured polyhedral cell model, allowing hybrid and non-conforming meshes. The parallelization is based on a

classical domain partitioning using MPI, with an optional second (local) level using OpenMP. In-situ visualization using Catalyst has been integrated into Code_Saturne for several years [4].

MELISSA

Melissa (Modular External Library for In Situ Statistical Analysis, <https://melissa-sa.github.io/>) is an open source framework that relies on a three tier architecture (Fig. 1). The Melissa Server aggregates the simulation results and updates iterative statistics as soon as a new result is available. The Melissa clients are the parallel simulations, providing their outputs to the server. Melissa Launcher interacts with the batch scheduler and server, for creating, launching, and supervising the server and clients.

Results

We validated our implementation of the iterative quantiles computation on a fluid mechanics use case simulating a water flow in a tube bundle. The mesh used for the finite volume simulations is composed of 6002400 hexahedra. We generated a multi-run study by simulating the injection of a tracer or dye along the inlet, with 2 independent injection surfaces, each defined by three varying parameters. This study ran a total of 80 000 simulations for computing ubiquitous variances, covariances, Sobol' indices, and the 5th, 25th, 50th, 75th and 95th percentiles (a type of quantile also called 100-quantile) on the 6M hexahedra and 100 timesteps. All simulations were not used for the quantiles that were computed from 20 000 simulations.

The experiments ran on Occigen, ranked 70th at the top500.org of June 2018. Occigen is composed of two slices, the first one is composed of 2106 nodes of 2 Intel® Haswell 12-Core E5-2690 V3 2.6 GHz, and the second is composed of 1260 nodes of 2 Intel Broadwell 14-Cores E5-2690 V4 2.6GHz, for a total of 85824 cores. These nodes have at least 64 GB memory each, and are connected by an InfiniBand FDR network. Occigen file system relies on Lustre with a bandwidth of 105 GB/s. For this experiment, we used the first slice, and we could only use 200 nodes simultaneously, with a job walltime limited to 24 hours. Each Code_Saturne simulation ran on one node, parallelized on 24 cores. On the server side, Melissa Server must have enough memory to keep all the updated statistics and to queue the inbound messages from the simulations, and must compute the statistics fast enough to consume the data faster than they arrive. Otherwise, Melissa Server inbound messages queue will end up full, eventually blocking the simulations. For this study, we ran Melissa Server on 8 nodes (192 cores).

During this study, Melissa Server processed 288 TB of data coming from the simulations. In a classical study, all these data would be output to the file system, and read back to compute statistics. This would not have been possible on Occigen simply due to the storage limitation per user.

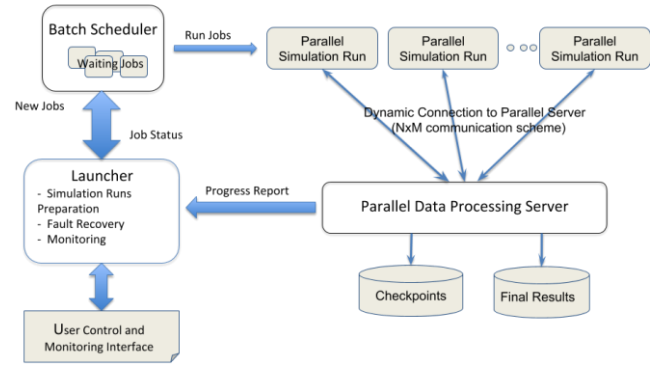


Figure 1: The launcher oversees the execution in tight link with the batch scheduler. The job scheduler regulates the number of simulation jobs to run according to the machine availability, leading to an elastic resource usage. The parallel server, started first, process incoming data as soon as received from the connected simulations. A fault tolerance mechanism automatically restarts the failing simulation runs or a failing parallel server.

Conclusion

Exascale machines will soon become a reality. But so far only very few applications are able to take benefit of the level of parallelization these machines provide. Ensemble runs may require to execute from thousands to millions of a simulation series, making it an extremely compute-intensive process that can fully benefit from those machines. However, the large amount of data generated is a strong I/O bottleneck if this intermediate data are saved to disk.

An alternative consists in processing this data in transit as proposed by the Melissa framework. But this approach requires one-pass data processing algorithms to limit the amount of memory needed. For quantiles, we rely on the stochastic quantile algorithm of Robbins-Monro. We led a massive experiment executing 80000 parallel CFD simulations on a 6million cell mesh, processing on-line 288TB of data that thus did not need to be saved to disk. Because we bypassed the I/O bottleneck, we were able to compute ubiquitous quantiles, i.e. at all cells of a reference mesh and timesteps.

REFERENCES

- [1] T. Terraz, A. Ribes, Y. Fournier, B. Iooss, and B. Raffin. Melissa: Large scale in transit sensitivity analysis avoiding intermediate files. On International Conference for High Performance Computing, Networking, Storage and Analysis (SC'17). Denver, 2017.
- [2] H. Robbins and S. Monro. A stochastic approximation method. The Annals of Mathematical Statistics, 22:400–407, 1951.
- [3] F. Archambeau, N. Méchitoua, and M. Sakiz. Code_saturne: a finite volume code for the computation of turbulent incompressible flows. International Journal on Finite Volumes, 1, 2004.
- [4] B. Lorendeau, Y. Fournier, and A. Ribes. In-situ visualization in fluid mechanics using catalyst: A case study for code saturne. On Large-Scale Data Analysis and Visualization (LDAV), IEEE Symposium, 2013