Controlling Parallel Adaptive Sparse Grid Collocation Simulations with Chiron

Vitor S. Souza[*], Gabriel Guerra[**], Marta Mattoso[*], Fernando A. Rochinha[**] and Alvaro L.G.A. Coutinho[†]

[*]Department of Computer Science
COPPE/Federal University of Rio de Janeiro, Brazil
e-mail: {vitor, marta}cos.ufrj.br - web page: http://www.cos.ufrj.br

[**]Department of Mechanical Engineering
COPPE/Federal University of Rio de Janeiro, Brazil
e-mail: {gguerra, faro}mecanica.coppe.ufrj.br - web page: http://www.mecanica.ufrj.br

[†] High Performance Computing Center and Department of Civil Engineering
COPPE/Federal University of Rio de Janeiro, Brazil
e-mail: alvaro@nacad.ufrj.br - web page: http://www.nacad.ufrj.br

ABSTRACT

Nonintrusive UQ methods, roughly describing, use a (expected small) number of runs of a deterministic computational model, each one having as inputs judiciously chosen points of the stochastic input space. Statistics of the outputs are then estimated from the deterministic computations, generating a large amount of data and thus requiring careful management [1]. Using Chiron, a data-centric scientific workflow engine that executes, in parallel, scientific applications, helps to control and manage these data. Chiron uses a dynamic data-centric approach, where scientific workflow algebra handles the parallel workflow execution efficiently. The algebra also standardizes data consumption and production as algebraic operands, with adherence to W3C provenance data model. Provenance is essential for scientific and engineering experiments and ensures that the experiment can be repeated over different conditions. Chiron provides native support for distributed provenance by storing provenance data during the execution of all samples and making it available for querying at runtime. Thus, it is possible to monitor the status of each input point run and availability of results through runtime provenance dataflow queries. Monitoring some specific attributes, results or checking the elapsed time of a given task may indicate that a failure happened. Such information can be used to refine the task (to prevent it from failing again) and resubmit it. Depending on the gathered results, the user may decide to change (or add) parameters corresponding to an input point or to make other decisions regarding the simulation. Finally, note that since each input corresponds to a parallel job assigned to a number of processors, which is in turn also solved in parallel, Chiron handles the execution of several simultaneous jobs, which configures a two-level overall parallel execution scheme. Uncertainty quantification scenarios with adaptive sparse grid collocation are particularly amenable to be steered and controlled by Chiron [2]. They require the ability of adapting the workflow, at runtime, based on user input and dynamic steering, according to error measures (or input thresholds) given by the user. We evaluate our approach using a novel and real large-scale workflow for uncertainty quantification on a 640-core cluster. The results show impressive execution time savings from 2.5 to 24 days, compared to non-iterative workflow execution.

REFERENCES

[1] G. Guerra, G., F.A Rochinha, R. Elias, D. de Oliveira, E. Ogasawara, J.F. Dias, M. Mattoso, A.L.G.A. Coutinho, "Uncertainty Quantification in Computational Predictive Models for Fluid Dynamics Using Workflow Management Engine", *Int. J. for Uncertainty Quantification*, Vol. **2**, pp. 53–71, (2012).

[2] J. Dias, G. Guerra, F. Rochinhaa, A. L.G.A. Coutinho, P. Valduriez, M. Mattoso, "Data-centric iteration in dynamic workflows", *Future Generation Computer Systems,* DOI:10.1016/j.future.2014.10.021, (2014).