Compas'2023 : Parallélisme / Architecture/ Système
LISTIC/USMB - Annecy France, 4-7 juillet 2023

I/O performance of multiscale finite element simulations on HPC platforms

Louis Peyrondet[1], Francieli Boito[1], Antonio Tadeu A. Gomes[2], Luan Teylo[1]

[1]Univ. Bordeaux, CNRS, Bordeaux INP, INRIA, LaBRI, UMR 5800, F-33400 Talence, France
louis.peyrondet@bordeaux-inp.fr, {francieli.zanon-boito, luan.gouveia-lima}@inria.fr

[2]Laboratório Nacional de Computação Científica (LNCC), Av Getúlio Vargas 333, 25651-075, Petrópolis-RJ, Brazil
atagomes@lncc.br

Résumé

In this work, we present MSLIO, a code to mimic the I/O behavior of numerical simulations that explore domain decomposition strategies, such as multiscale finite element methods. These strategies may render I/O patterns that vary considerably depending on how data associated with each subdomain is read or written to the file system. In this context, such an I/O kernel is useful for HPC research, as it can be executed more easily and more efficiently than the full simulations when researchers are interested in the I/O load only. We validate MSLIO by comparing it to the I/O performance of an actual simulation based on a library that implements the MHM (Multiscale Hybrid Mixed) method, and we then use MSLIO to test some possible improvements to the output routine of this library. Our study was conducted in two systems, which use two widely popular parallel file systems (PFS) : BeeGFS and Lustre. After an analysis of the traces generated by the library, it was determined that the creation of separate sets of output files for each subdomain was causing a significant increase in the metadata load when the number of subdomains was large. This heavy load was overwhelming the metadata server, resulting in poor I/O performance. The impact of optimizations such as buffering small requests and combining all of the output files into a single shared one were studied. These optimizations illustrated the benefits of the MSLIO kernel with its ability to facilitate rapid prototyping, execution, and trace generation for I/O research.

In conclusion, the experiments conducted demonstrate that the buffering optimization did not yield any significant performance improvements. However, transitioning to a single shared file version produced contrasting effects on the different PFS used in the study. The BeeGFS system experienced a significant decrease in performance, while the Lustre system showed improved performance as the number of subdomains increased.

Mots-clés : High-performance computing, parallel I/O, numeric library, I/O kernel, mini-app

---