A multi-threaded tensor algebra library for the ACES III software package

Dmitry I. Lyakh, Beverly A. Sanders, Nakul Jindal, Victor Lotrich, Rodney J. Bartlett
Quantum Theory Project, University of Florida, Gainesville, 32611, USA

The ACES III software package [1] has proven to be one of the most scalable electronic structure codes for massively-parallel execution of *ab initio* (correlated) many-body methods. Scaling up to 60000 and 100000 cores was observed for the CCSD and CCSD(T) approaches, respectively. However, the solely MPI parallelization of the code does not allow to go beyond those numbers due to heterogeneity of modern HPC systems. Indeed, a contemporary HPC machine consists of nodes containing multiple CPU cores sharing the same RAM memory, one or more GPU or MIC accelerators with their own memory banks, etc. The heterogeneity of such an architecture is manifested in having multiple kinds of processing/memory units with their own pros and cons. Consequently, there is a general algorithmic problem of how to exploit such a computer with the maximal efficiency. In many cases, running one MPI process per CPU core is not a good solution because (a) the amount of CPU memory per process is significantly reduced, (b) MPI processes running on the same node do not exploit the advantages of sharing CPU memory, thus increasing the amount of unnecessary communications.

Since the ACES III package is mostly oriented on correlated *ab initio* methodology, the most time consuming parts of each particular electronic structure method (coded using a built-in domain-specific language, called SIAL) are spent in doing tensor algebra operations, predominantly tensor contractions. Each particular global tensor operation is automatically distributed among a certain group of MPI processes by the super-instruction interpreter (SIP), each MPI process performing the same tensor operation on its own arguments, namely, dense tensor blocks generated by segmentation of index ranges. Consequently, to allow running an MPI process on multiple CPU cores, we have implemented a multi-threaded (OpenMP) tensor algebra library which can deal with dense tensor blocks of arbitrary ranks. A particular emphasis was made on a cache-efficient tensor transpose algorithm [2] which is a necessary prerequisite for the tensor contraction kernel based on matrix-matrix multiplication. Our multi-threaded cache-efficient tensor transpose algorithm is a generalization of the cache-efficient matrix transpose algorithm. It was shown to be at least 2-3 times faster than a straightforward scattering algorithm, especially for large tensor blocks of higher ranks. The tensor algebra library has been interfaced with the next generation ACES code (ACES IV) which is a version being developed to target heterogeneous HPC systems (including the support of accelerators, e.g. Nvidia GPU) and more advanced (single- and multireference) electronic structure methods (higher-rank tensor algebra is a necessary prerequisite for that). The factorized equations for those methods (a list of necessary tensor operations to perform) can be generated by the automated module DIAGEN [3] developed by one of the present authors (D.I.L.).

Acknowledgements: AFOSR and DOE are appreciated for their financial support.

1. E. Deumens, V.F. Lotrich, A. Perera, M.J. Ponton, B.A. Sanders, R.J. Bartlett. Software design of ACES III with the super instruction architecture. Wiley Interdisciplinary Reviews – Computational Molecular Science 1, 895.

2. D.I. Lyakh. A multi-threaded cache-efficient tensor transpose algorithm for higher-rank tensor algebra. In preparation (2013).

3. D.I. Lyakh, R.J. Bartlett. Efficient quantum-chemical method designing based on the ACESIII parallel platform. Book of abstracts. 50th Sanibel Symposium, Feb 24 – Mar 2, 2010. St. Simon's Island GA, USA.