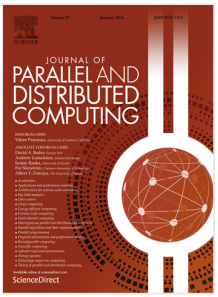
Accepted Manuscript



Cross-scale efficient tensor contractions for coupled cluster computations through multiple programming model backends

Khaled Z. Ibrahim, Evgeny Epifanovsky, Samuel Williams, Anna I. Krylov

PII: S0743-7315(17)30078-3
DOI: http://dx.doi.org/10.1016/j.jpdc.2017.02.010
Reference: YJPDC 3640

To appear in: *J. Parallel Distrib. Comput.*

Received date: 13 September 2016
Revised date: 20 February 2017
Accepted date: 22 February 2017

Please cite this article as: K.Z. Ibrahim, E. Epifanovsky, S. Williams, A.I. Krylov, Cross-scale efficient tensor contractions for coupled cluster computations through multiple programming model backends, *J. Parallel Distrib. Comput.* (2017), http://dx.doi.org/10.1016/j.jpdc.2017.02.010

Cross-scale Efficient Tensor Contractions for Coupled Cluster Computations Through Multiple Programming Model Backends

Khaled Z. Ibrahim[1a], Evgeny Epifanovsky[b], Samuel Williams[a], Anna I. Krylov[c]

[a]Computational Research Division, Lawrence Berkeley National Laboratory
Berkeley, CA, USA
[b]Q-Chem, Inc
6601 Owens Drive, Suite 105, Pleasanton, CA 94588, USA
[c]Department of Chemistry
University of Southern California, Los Angeles, CA, USA
{kzibrahim, swwilliams}@lbl.gov, {epifanov, krylov}@usc.edu

Abstract

Coupled-cluster methods provide highly accurate models of molecular structure through explicit numerical calculation of tensors representing the correlation between electrons. These calculations are dominated by a sequence of tensor contractions, motivating the development of numerical libraries for such operations. While based on matrix-matrix multiplication, these libraries are specialized to exploit symmetries in the molecular structure and in electronic interactions, and thus reduce the size of the tensor representation and the complexity of contractions. The resulting algorithms are irregular and their parallelization has been previously achieved via the use of dynamic scheduling or specialized data decompositions. We introduce our efforts to extend the Libtensor framework to work in the distributed memory environment in a scalable and energy-efficient manner. We achieve up to 240× speedup compared with the optimized shared memory implementation of Libtensor. We attain scalability to hundreds of thousands of compute cores on three distributed-memory architectures, (Cray XC30 and XC40, and IBM Blue Gene/Q), and on a heterogeneous GPU-CPU system (Cray XK7). As the bottlenecks shift from being compute-bound DGEMM's to communication-bound collectives as the size of the molecular system scales, we adopt two radically different parallelization approaches for handling load-imbalance, tasking and bulk synchronous models. Nevertheless, we preserve a unified interface to both programming models to maintain the productivity of computational quantum chemists.

Keywords: Tensor Contraction Engines; Quantum Chemistry; Libtensor; Cyclops. High Performance Computing; Distributed Memory Programming Models; Energy Efficiency

1. Introduction

Tensor contraction routines are important kernels for quantum chemistry computation. The tuning of these kernels is notoriously complex because of inherent computational characteristics such as load imbalance, complex locality, memory requirements, etc. Multiple frameworks provide

[1]Corresponding author: Khaled Ibrahim, kzibrahim@lbl.gov, voice: (510)486-7467, fax: (510)486-6900