

Boosting the Scalability of Car-Parrinello Molecular Dynamics Simulations for Multi- and Manycore Architectures

Tobias Klöffel[†]

Dep. Chemistry and Pharmacy
Friedrich-Alexander University
Erlangen-Nürnberg (FAU)
Erlangen, Germany
tobias.kloeffel@fau.de

Bernd Meyer

Dep. Chemistry and Pharmacy
Friedrich-Alexander University
Erlangen-Nürnberg (FAU)
Erlangen, Germany
bernd.meyer@fau.de

Gerald Mathias[‡]

Leibniz-Supercomputing Centre of
the Bavarian Academy of Science
and Humanities (LRZ)
Garching, Germany
gerald.mathias@lrz.de

ABSTRACT

We present our recent optimizations of the ultra-soft pseudo-potential (USPP) code path of the *ab initio* molecular dynamics program CPMD (www.cpmc.org). Following the internal instrumentation of CPMD, all relevant USPP routines have been revised to fully support hybrid MPI/OpenMP parallelization. For two time-critical routines, namely the multiple distributed 3-*d* FFTs of the electronic states and a key distributed matrix-matrix multiplication, we have implemented hybrid parallel algorithms with overlapping computation and communication. The achievements in performance and scalability are demonstrated on a small reference system of 128 water molecules and further systems of increasing size. Performance evaluation shows gains of up to one order of magnitude and around 50% peak performance for simulation systems readily used in production.

CCS CONCEPTS

• **Applied computing~Chemistry** • *Computing methodologies~Massively parallel algorithms*

KEYWORDS

Ab initio molecular dynamics; hybrid MPI/OpenMP parallelization; batched multiple 3*d* FFT

1 Introduction

The *ab initio* molecular dynamics code CPMD [1] has been running on many top-level HPC systems for the last decades. Besides being a working horse for many quantum chemical applications it has also been an important benchmark code to test new supercomputers [2]. Its underlying method is Kohn-Sham density functional theory

(DFT), which describes the electrons through singly or doubly occupied electronic states. In CPMD these states are expanded in plane wave (PW) basis and propagated via the Car-Parrinello extended Lagrangean technique. Details of the method and the implementations are discussed by Marx and Hutter [3].

CPMD uses pseudo potentials (PP) to describe the chemically inert core electrons, to limit the calculation to the valence electrons and to reduce the size of the PW basis set. So-called ultra-soft PP (USPP) introduce some computational overhead but reduce the size of the PW basis even further. The PW grid is distributed by a 1-*d* domain decomposition to the MPI tasks (see [3] for details). Typical grid sizes do not exceed a few hundred points per dimension, which is an effective limit for the number of MPI tasks that can be efficiently used in a calculation. Thus, for USPP the MPI-only scaling limit is reached very early on modern multi and many core architectures (see blue dashed curve Fig. 1).

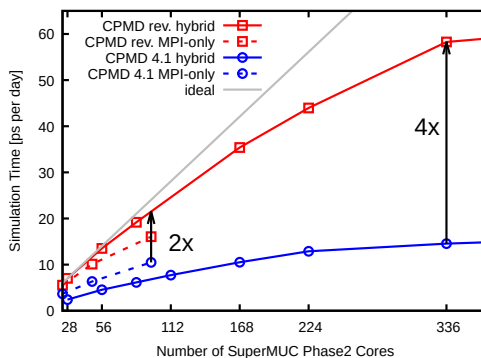


Figure 1: Performance and scalability improvement for the MPI-only and the hybrid mode of parallelization, respectively.

Here we present our revision of the USPP code path in CPMD now fully supporting hybrid MPI+OpenMP parallelization for enhanced scalability. Furthermore, the node-level performance has been boosted by mapping compute-intensive routines to BLAS calls. We exemplify our improvements of the codes on a test system of 128 water molecules, for which performance data of CPMD and other

[†]Main developer

[‡]Presenter

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).

Supercomputing '18, November, 2018, Dallas, Texas USA

© 2018 Copyright held by the owner/author(s).

codes can be readily found for comparison. Although with today's hardware this system can run on a single HPC node, it is still interesting how much one can reduce the time to solution. Actual HPC applications will use larger systems and/or run multiple weakly coupled replicas in parallel for enhanced sampling. All benchmarks are run on SuperMUC (dual Haswell nodes, 2×14 cores at 2.6 GHz) using 4 MPI tasks per node and 7 OpenMP threads per task.

2 Optimization

Working through the list of the most time-consuming routines in CPMD we managed to eliminate many bottlenecks. Some routines were speed up by one or two orders of magnitude through performance libraries, OpenMP parallelization and optimization of memory access patterns, such that these routines became less important or even negligible in the overall timing. Fig. 1 compares the length of MD trajectory generated per day for MPI and hybrid parallelization. The hybrid parallelization of the revised version of the code (red) outperforms the MPI only already at small core counts and reaches a significantly increased maximum performance. For two remaining time-critical routines we, furthermore, revised the parallel communication pattern.

2.1 Batched Multiple 3-d FFT

A major computational bottleneck of the USPP but also of the more frequently used norm-conserving PP is the transformation of the electronic states ($\sim 10^3$) between real and Fourier space. Scaling of the associated multiple 3-d FFTs is hampered by the required all-to-all communication step. We have implemented a batched algorithm, which combines the all-to-all communication for multiple states to increase the communication bandwidth. Comparing the blue and green curves in Fig. 2 shows the significant reduction of the time spent for FFT during each MD step, particularly for large core counts.

Furthermore, we have split off the master thread for the communication while the other six threads compute the actual FFT.

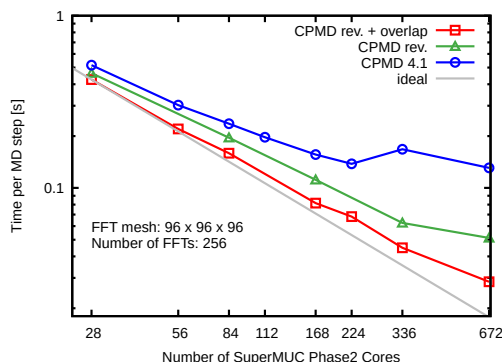


Figure 2: Performance gain by the batched 3-d FFT of the electronic states (hybrid parallelization).

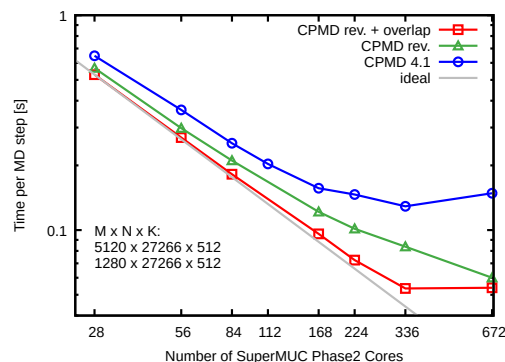


Figure 3: Performance and scalability gain of the distributed matrix-matrix multiplication (hybrid parallelization).

This overlapping of computation and communication further improves the scaling and brings the algorithm closer to ideal scaling (red curve in Fig. 2) and should be even more profitable if more OpenMP threads per MPI task are used.

2.2 Distributed Matrix Multiply

Many of the USPP-specific expressions are computed by distributed matrix-matrix multiplications. Here, a nested loop over chemical species and the atoms of each species is used reflecting the internal data structure of CPMD. Combining loops and thereby increasing the block size of the matrix-matrix multiplication already rendered a significant speedup, as a comparison of the blue and green curves in Fig. 3 indicates. Similar to the 3-d FFT we have, furthermore, overlapped computation and communication and reach almost perfect scaling (red curve), except for the last point, where communication dominates computation in this small example.

3 Outlook

In the poster we show that the scaling and performance is even more favorable for systems containing more electrons like larger water boxes or zinc oxide solids. In a sample application we have used **14 k cores** yielding a **12-fold reduced time to solution** and reaching a sustained performance of 50 % peak.

ACKNOWLEDGMENTS

We thank Georg Hager from the RRZE for fruitful discussions and the KONWHIR software initiative of the state of Bavaria for financial support. Computer time on SuperMUC was granted to T.K. through the "Summer of Simulation 2017" training program by LRZ.

REFERENCES

- [1] CPMD, <http://www.cpmd.org/>, Copyright IBM Corp 1990-2015, Copyright MPI für Festkörperforschung Stuttgart 1997-2001
- [2] 98. First Experiences with *ab initio* Molecular Dynamics on OpenPOWER: The Case of CPMD. Valery Weber, et al; Poster SC16, Saltlake City, Utah, USA
- [3] Dominik Marx and Jürg Hutter. 2009. *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods*. Cambridge University Press, Cambridge