

Extreme Scalability of DFT-based QM/MM MD Simulations Using MiMiC

Viacheslav Bolnykh,^{*,†,‡,¶} Jógvan Magnus Haugaard Olsen,^{*,§} Simone Meloni,^{||}
Martin P. Bircher,[⊥] Emiliano Ippoliti,[¶] Paolo Carloni,^{*,¶,#,†} and Ursula
Rothlisberger^{*,⊥}

[†]*Department of Physics, RWTH Aachen University, 52056 Aachen, Germany*

[‡]*CaSToRC, The Cyprus Institute, 2121 Aglantzia, Nicosia, Cyprus*

[¶]*Institute for Advanced Simulation (IAS-5) and Institute of Neuroscience and Medicine (INM-9), Forschungszentrum Jülich, 52425 Jülich, Germany*

[§]*Hylleraas Centre for Quantum Molecular Sciences, Department of Chemistry, UiT The Arctic University of Norway, N-9037 Tromsø, Norway*

^{||}*Department of Mechanical and Aerospace Engineering, Sapienza University of Rome, 00184 Rome, Italy*

[⊥]*Laboratory of Computational Chemistry and Biochemistry, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*

[#]*Institute for Neuroscience and Medicine (INM-11), Forschungszentrum Jülich, 52425 Jülich, Germany*

E-mail: v.bolnykh@fz-juelich.de; jogvan.m.olsen@uit.no; p.carloni@fz-juelich.de;
ursula.roethlisberger@epfl.ch

Abstract

We present a highly scalable DFT-based QM/MM implementation developed within MiMiC, a recently introduced multiscale modeling framework that uses a loose-coupling

strategy in conjunction with a multiple-program multiple-data (MPMD) approach. The computation of electrostatic QM/MM interactions is parallelized exploiting both distributed- and shared-memory strategies. Here, we use the efficient CPMD and GRO-MACS programs as QM and MM engines, respectively. The scalability is demonstrated through large-scale benchmark simulations of realistic biomolecular systems employing GGA and hybrid exchange–correlation functionals. We show that the loose-coupling strategy adopted in MiMiC, with its inherent high flexibility, does not carry any significant computational overhead compared to a tight-coupling scheme. Furthermore, we demonstrate that the adopted parallelization strategy enables scaling of up to 13,000 CPU cores with efficiency above 70%, thus making DFT-based QM/MM MD simulations using hybrid functionals at the nanosecond scale accessible.

1 Introduction

The scale spanned by (bio-)chemical processes ranges from the decomposition of small organic molecules due to attosecond laser pulses to the folding of proteins comprising hundreds of thousands of atoms over several micro- to milliseconds. Exploring the dynamics of those processes requires methods of varying precision and computational cost, where usually a trade-off between the former and the latter have to be found in order to describe chemically relevant time scales and system sizes. In particular, while the system sizes that can be afforded with accurate *first-principles* methods are constantly growing, the requirement of sampling a sufficiently long time scale often limits the system size to a few hundreds of atoms in Kohn–Sham density functional theory (KS-DFT) based molecular dynamics (MD) and to less than a few dozen for accurate wavefunction-based methods.

One way, in which this computational bottleneck can be overcome is the use of multiscale methods that bridge size and time scales by combining methods of different accuracy and computational efficiency. The parts of the system that are relevant to a chemical process are often small with respect to the overall system size. Thus, they can be described using

methods of higher accuracy that would not be viable for a description of the entire system. On the other hand, it is usually sufficient for the large remainder of the system to be described by less precise but computationally much more efficient models. Such combination of methods ultimately allows one to cover the time scale that is necessary to properly sample the process under investigation.

MiMiC¹ (multiscale modeling in computational chemistry) is a recently introduced framework for multiscale modeling based on a loose-coupling paradigm. Loose coupling allows for maximum flexibility by allowing the framework to be interfaced with different programs with only minimal modifications; however, such an approach should not compromise efficiency. Alternatively, a tight-coupling strategy is limited to the program it has been designed for, with the immediate advantage of making it easy to minimize the computational overhead of the framework for the very programs it interfaces. This, however, comes at the cost of a loss of flexibility. MiMiC aims at uniting the flexibility of a loose-coupling scheme with the efficiency of a tight-coupling approach by virtue of highly efficient data exchange, thus providing a highly efficient loose-coupling framework.

The MiMiC framework has first been applied in the context of hybrid quantum mechanical/molecular mechanics (QM/MM) simulations, where it does not only interface the QM and MM drivers, but also computes the electrostatic interaction between the QM and MM subsystems.¹ Highly popular for — but not limited to — biochemical applications, the QM/MM approach, allows, for example, a chemically relevant part of a protein (QM) to be simulated using *first principles* methods, while the rest of the system (MM) can be conveniently described at the force-field level with sufficient accuracy. As such, the QM/MM method has been implemented in various program-specific contexts, often in a tight-coupling fashion. Here, we present the parallelization schemes used in MiMiC and demonstrate that the use of the loosely-coupled MiMiC framework allows for more flexibility without compromising computational efficiency with respect to existing, tight-coupling implementations. Using systems with varying sizes of QM and MM regions, we provide benchmarks and per-

formance studies for MiMiC interfacing between the plane wave/pseudopotential CPMD² program as a main MD driver with the classical MD package GROMACS,³ thus enabling both Born–Oppenheimer (BO) and Car–Parrinello (CP) QM/MM MD to be performed. The CPMD program provides a preexisting, tight-coupling QM/MM interface to GROMOS;⁴ its choice as a main driver therefore allows for a direct comparison of the performance of loose- and tight-coupling strategies.

The text is organized as follows: We first provide a summary of the QM/MM methodology adopted in MiMiC,¹ followed by a description of the implementation including aspects related to communication between coupled programs and parallelization of QM/MM interactions. We then provide a description of the test systems used for the benchmarks, which are intended to cover a range of typical system sizes. Following this, we present benchmarks showcasing the parallel performance of the MiMiC-based QM/MM implementation for up to several thousand CPU cores, demonstrating superior scaling performance with respect to the preexisting, tight-coupling QM/MM implementation in the CPMD program. We then go on to show that at a large number of cores, parallel efficiency is mainly exhausted due to serial parts of the main driver in CPMD, thus showing that the loose-coupling scheme adopted in MiMiC does not lead to any significant overhead associated to communication, while providing a much more flexible, system-independent framework.

2 Methodology

The MiMiC framework enables multiscale simulations by coupling external programs, which independently compute contributions belonging to different parts of a molecular system. We refer the reader to our original presentation of MiMiC in Ref. 1 for a more detailed description of the framework itself. Here, we provide some details relevant to the MiMiC-based QM/MM implementation in which CPMD and GROMACS are used to enable QM/MM MD simulations. In this implementation, the CPMD program serves as the main MD driver

and additionally computes the QM contributions while GROMACS provides the MM contributions. The MiMiC framework itself is built around a multiple-program multiple-data (MPMD) approach with loosely coupled programs. This allows the external programs to run simultaneously and independently of each other with periodic data exchange, which relies on network-based solutions rather than file-based data exchange (further details are given below). Moreover, it enables separate allocations of processor pools to each of the programs, thus allowing each program to utilize its own optimal parallelization scheme which improves the parallel efficiency of the resulting solution.

2.1 QM/MM Theory

In the following, we provide an overview of the additive electrostatic-embedding QM/MM scheme presenting only the key equations. For a more detailed description of the adopted QM/MM methodology, we refer the reader to Ref. 1.

The total energy of the system is split into three contributions (as usual for additive QM/MM schemes): the energy of the QM subsystem, the energy of the MM subsystem, and the QM/MM interaction energy. The QM subsystem is described according to plane-wave/pseudopotential-based KS-DFT while the MM subsystem is represented by an MM force field. As mentioned before, the QM and MM contributions are computed by CPMD and GROMACS, respectively. Here, we will focus on the QM/MM part, which includes all interactions between the electrons and nuclei in the QM subsystem and the atoms in the MM subsystem.

The QM–MM interactions can be divided into two parts, namely bonded and nonbonded interactions, where the former only appear when there are covalent bonds crossing the QM–MM boundary. The bonded interactions are described classically using the parameters of the same force field that is used in the MM subsystem. Open valencies in the QM subsystem are saturated by adding boundary atoms that are described by a monovalent pseudopotential and thus contribute a single electron to the QM subsystem. The nonbonded interactions consist

of electrostatic and van der Waals interactions. The latter are described through the MM force field. Since the bonded and van der Waals interactions are purely classical, we delegate these to the external MM engine. The electrostatic interactions, which include interactions between the MM charges (or multipoles) and the QM nuclei and electrons, are computed by MiMiC. The current implementation in MiMiC supports an electrostatic embedding scheme in which the electrostatic potential from the MM charges (or multipoles) is added to the KS-DFT external potential. This involves the electronic density of the QM subsystem, which is mapped on a grid, typically of the order of 10^6 points, and thus requires special attention, in particular with respect to computational cost and parallel scalability. Evaluating the full electrostatic part of the nonbonded energy exactly is not only very computationally expensive but also exhibits poor scalability with the size of the system, which can be of the order of 10^5 atoms or larger, since we have to compute the full integral over the electronic grid for each of the MM charges (see below). To overcome these limitations, we have previously formulated and implemented a generalized version of the electrostatic coupling scheme by Laio et al.,⁴ which was shown to yield the same accuracy as an exact electrostatic treatment but at a substantially reduced cost.¹

Within the generalized electrostatic coupling scheme, the MM atoms are grouped into two domains according to their distance from the QM subsystem. The MM atoms that are within a specified distance from the QM subsystem (the short-range domain) interact directly with the QM electron density and nuclei, whereas the remaining MM atoms (the long-range domain) interact through a multipole expansion of the electrostatic potential from the QM electrons and nuclei. In multi-index notation, as introduced in Ref. 1, the

electrostatic QM–MM interaction energy can thus be written as

$$\begin{aligned}
V_{\text{QM/MM}}^{\text{es}}[\rho^{\text{QM}}](\mathbf{R}^{\text{QM}}, \mathbf{R}^{\text{MM}}) = & \sum_{i=1}^{N_{\text{MM}}^{\text{sr}}} \sum_{|\alpha|=0}^{A_i} \frac{(-1)^{|\alpha|}}{\alpha!} M_i^{[\alpha]} \left(\int T_{\text{mod}}^{[\alpha]}(\mathbf{R}_i^{\text{MM}}, \mathbf{r}) \rho^{\text{QM}}(\mathbf{r}) \, \text{d}\mathbf{r} \right. \\
& \left. + \sum_{j=1}^{N_{\text{QM}}} T_{\text{mod}}^{[\alpha]}(\mathbf{R}_i^{\text{MM}}, \mathbf{R}_j^{\text{QM}}) Z_j^{\text{QM}} \right) \\
& + \sum_{i=1}^{N_{\text{MM}}^{\text{lr}}} \sum_{|\alpha|=0}^{A_i} \sum_{|\beta|=0}^{B_{\text{QM}}} \frac{(-1)^{|\alpha+\beta|}}{\alpha! \beta!} M_i^{[\alpha]} T^{[\alpha+\beta]}(\mathbf{R}_i^{\text{QM}}, \bar{\mathbf{R}}^{\text{QM}}) \\
& \times M_{\text{QM}}^{[\beta]}[\rho^{\text{QM}}](\mathbf{R}^{\text{QM}})
\end{aligned} \tag{1}$$

where $N_{\text{MM}}^{\text{sr}}$ and $N_{\text{MM}}^{\text{lr}}$ are the number of MM atoms in the short- and long-range domains, respectively, \mathbf{R}^{MM} and \mathbf{R}^{QM} are the positions of MM atoms and QM nuclei, respectively, $M_i^{[\alpha]}$ is a component of the multipole associated with i th MM atom, $\rho^{\text{QM}}(\mathbf{r})$ is the QM electronic density, N_{QM} is the number of QM nuclei, and Z_j^{QM} is the core charge of the j th QM nucleus. The long-range part involves the multipoles of the QM subsystem which are defined as

$$M_{\text{QM}}^{[\beta]}[\rho^{\text{QM}}](\mathbf{R}^{\text{QM}}) = \int \rho^{\text{QM}}(\mathbf{r}) (\mathbf{r} - \bar{\mathbf{R}}^{\text{QM}})^{\beta} \, \text{d}\mathbf{r} + \sum_{j=1}^{N_{\text{QM}}} Z_j^{\text{QM}} (\mathbf{R}_j^{\text{QM}} - \bar{\mathbf{R}}^{\text{QM}})^{\beta} \tag{2}$$

where $\bar{\mathbf{R}}^{\text{QM}}$ is the origin of the multipole expansion, for which we use the centroid of the QM subsystem, i.e. $\bar{\mathbf{R}}^{\text{QM}} = \sum_{j=1}^{N_{\text{QM}}} \mathbf{R}_j^{\text{QM}} / N_{\text{QM}}$. Finally, eq 1 also involves the unmodified and modified interaction tensors which are generally defined as $T^{[\alpha]}(\mathbf{R}_a, \mathbf{R}_b) = \partial_{\mathbf{R}_b}^{\alpha} \frac{1}{|\mathbf{R}_b - \mathbf{R}_a|}$ and $T_{\text{mod}}^{[\alpha]}(\mathbf{R}_a, \mathbf{R}_b) = \partial_{\mathbf{R}_b}^{\alpha} \frac{r_{c,a}^4 - |\mathbf{R}_b - \mathbf{R}_a|^4}{r_{c,a}^5 - |\mathbf{R}_b - \mathbf{R}_a|^5}$, respectively, where $\partial_{\mathbf{R}_b}^{\alpha} = \frac{\partial^{|\alpha|}}{\partial R_{b,x}^{\alpha_x} \partial R_{b,y}^{\alpha_y} \partial R_{b,z}^{\alpha_z}}$ and $r_{c,a}$ in the latter is the covalent radius of the a th MM atom. The modified interaction tensor is introduced in order to mitigate divergences at short distances thus avoiding the electron spill-out problem which is pervasive in electrostatic embedding QM/MM due to the lack of Pauli repulsion between the QM and MM subsystems. The modified form of the interaction potential was introduced and validated in Ref. 4.

In order to obtain the electronic density of the QM subsystem, we also need to add the potential induced by MM atoms to the external KS potential. The potential is obtained as the functional derivative of the electrostatic QM–MM interaction energy (eq 1) with respect to the electronic density which yields

$$\begin{aligned}
v_{\text{QM/MM}}^{\text{es,sr}}(\mathbf{r}) = & \sum_{i=1}^{N_{\text{MM}}^{\text{sr}}} \sum_{|\alpha|=0}^{A_i} \frac{(-1)^{|\alpha|}}{\alpha!} M_i^{[\alpha]} T_{\text{mod}}^{[\alpha]}(\mathbf{R}_i^{\text{MM}}, \mathbf{r}) \\
& + \sum_{i=1}^{N_{\text{MM}}^{\text{lr}}} \sum_{|\alpha|=0}^{A_i} \sum_{|\beta|=0}^{B_{\text{QM}}} \frac{(-1)^{|\alpha+\beta|}}{\alpha! \beta!} M_i^{[\alpha]} T^{[\alpha+\beta]}(\mathbf{R}_i^{\text{MM}}, \bar{\mathbf{R}}^{\text{QM}}) (\mathbf{r} - \bar{\mathbf{R}}^{\text{QM}})^\beta
\end{aligned} \tag{3}$$

where the first part is the potential from the MM atoms in the short-range domain and the second part is due to the MM atoms in the long-range domain.

The electrostatic QM–MM interactions also give rise to additional forces acting on the QM nuclei and MM atoms. The force on a QM atom due to the MM atoms in the short-range domain can be expressed as

$$F_{\mathbf{R}_j^{\text{QM}}}^{\text{es,sr}}[\gamma] = - \sum_{i=1}^{N_{\text{MM}}^{\text{sr}}} \sum_{|\alpha|=0}^{A_i} \frac{(-1)^{|\alpha|}}{\alpha!} M_i^{[\alpha]} T_{\text{mod}}^{[\alpha+\gamma]}(\mathbf{R}_i^{\text{MM}}, \mathbf{R}_j^{\text{QM}}) Z_j^{\text{QM}} \tag{4}$$

with γ being either $(1, 0, 0)$, $(0, 1, 0)$, or $(0, 0, 1)$ representing the Cartesian components of the force. The corresponding force acting on an MM atom in the short-range domain is then

$$\begin{aligned}
F_{\mathbf{R}_i^{\text{MM}}}^{\text{es,sr}}[\gamma] = & - \sum_{|\alpha|=0}^{A_i} \frac{(-1)^{|\alpha+\gamma|}}{\alpha!} M_i^{[\alpha]} \left(\int T_{\text{mod}}^{[\alpha+\gamma]}(\mathbf{R}_i^{\text{MM}}, \mathbf{r}) \rho^{\text{QM}}(\mathbf{r}) \, d\mathbf{r} \right. \\
& \left. + \sum_{j=1}^{N_{\text{QM}}} T_{\text{mod}}^{[\alpha+\gamma]}(\mathbf{R}_i^{\text{MM}}, \mathbf{R}_j^{\text{QM}}) Z_j^{\text{QM}} \right)
\end{aligned} \tag{5}$$

The forces acting on QM nuclei due to the MM atoms in the long-range domain are obtained

via

$$\begin{aligned}
F_{\mathbf{R}_j^{\text{QM}}}^{\text{res,lr} [\gamma]} = & - \sum_{i=1}^{N_{\text{MM}}^{\text{lr}}} \sum_{|\alpha|=0}^{A_i} \sum_{|\beta|=1}^{B_{\text{QM}}} \frac{(-1)^{|\alpha+\beta|}}{\alpha!(\beta-\gamma)!} M_i^{[\alpha]} T^{[\alpha+\beta]}(\mathbf{R}_i^{\text{MM}}, \bar{\mathbf{R}}^{\text{QM}}) Z_j^{\text{QM}}(\mathbf{R}_j^{\text{QM}} - \bar{\mathbf{R}}^{\text{QM}})^{\beta-\gamma} \\
& - \frac{1}{N_{\text{QM}}} \sum_{i=1}^{N_{\text{MM}}^{\text{lr}}} \sum_{|\alpha|=0}^{A_i} \left(\sum_{|\beta|=0}^{B_{\text{QM}}} \frac{(-1)^{|\alpha+\beta|}}{\alpha!\beta!} M_i^{[\alpha]} T^{[\alpha+\beta+\gamma]}(\mathbf{R}_i^{\text{MM}}, \bar{\mathbf{R}}^{\text{QM}}) M_{\text{QM}}^{[\beta]}[\rho^{\text{QM}}](\mathbf{R}^{\text{QM}}) \right. \\
& \quad \left. + \sum_{|\beta|=1}^{B_{\text{QM}}} \frac{(-1)^{|\alpha+\beta|}}{\alpha!(\beta-\gamma)!} M_i^{[\alpha]} T^{[\alpha+\beta]}(\mathbf{R}_i^{\text{MM}}, \bar{\mathbf{R}}^{\text{QM}}) M_{\text{QM}}^{[\beta-\gamma]}[\rho^{\text{QM}}](\mathbf{R}^{\text{QM}}) \right)
\end{aligned} \tag{6}$$

while the corresponding force on an MM atom is

$$F_{\mathbf{R}_i^{\text{MM}}}^{\text{res,lr} [\gamma]} = - \sum_{|\alpha|=0}^{A_i} \sum_{|\beta|=0}^{B_{\text{QM}}} \frac{(-1)^{|\alpha+\beta+\gamma|}}{\alpha!\beta!} M_i^{[\alpha]} T^{[\alpha+\beta+\gamma]}(\mathbf{R}_i^{\text{MM}}, \bar{\mathbf{R}}^{\text{QM}}) M_{\text{QM}}^{[\beta]}[\rho^{\text{QM}}](\mathbf{R}^{\text{QM}}) \tag{7}$$

2.2 QM/MM MD Workflow

The workflow of a MiMiC-based QM/MM BO-MD simulation using CPMD and GROMACS is illustrated in Figure 1 (an analogous workflow would apply for CP-MD). Both CPMD and GROMACS are executed and run simultaneously. After initialization, MiMiC collects system data from GROMACS (number of atoms, atom types, coordinates, etc.) and propagates it to CPMD. Once in the MD loop, the first two steps of the velocity Verlet integrator are performed as usual. After the coordinates have been updated, they are distributed to GROMACS through MiMiC using the communication library (CommLib) which is part of the MiMiC framework (see below for further details). Then, both CPMD and GROMACS proceed computing energies and forces simultaneously. In the case of BO-MD, as in the example shown in Figure 1, this includes a self-consistent field (SCF) procedure to solve the KS equations in order to determine the electronic density. This requires also the electrostatic potential from the MM part (eq 3), which is computed by MiMiC and added to the KS-DFT external potential in CPMD. After the SCF, CPMD computes the QM forces and receives

the QM/MM part of the forces (eqs 4–7) from MiMiC. Finally, MiMiC collects the energy and forces from GROMACS and propagates them to CPMD, which then performs the second velocity update and goes back to the beginning of the MD loop.

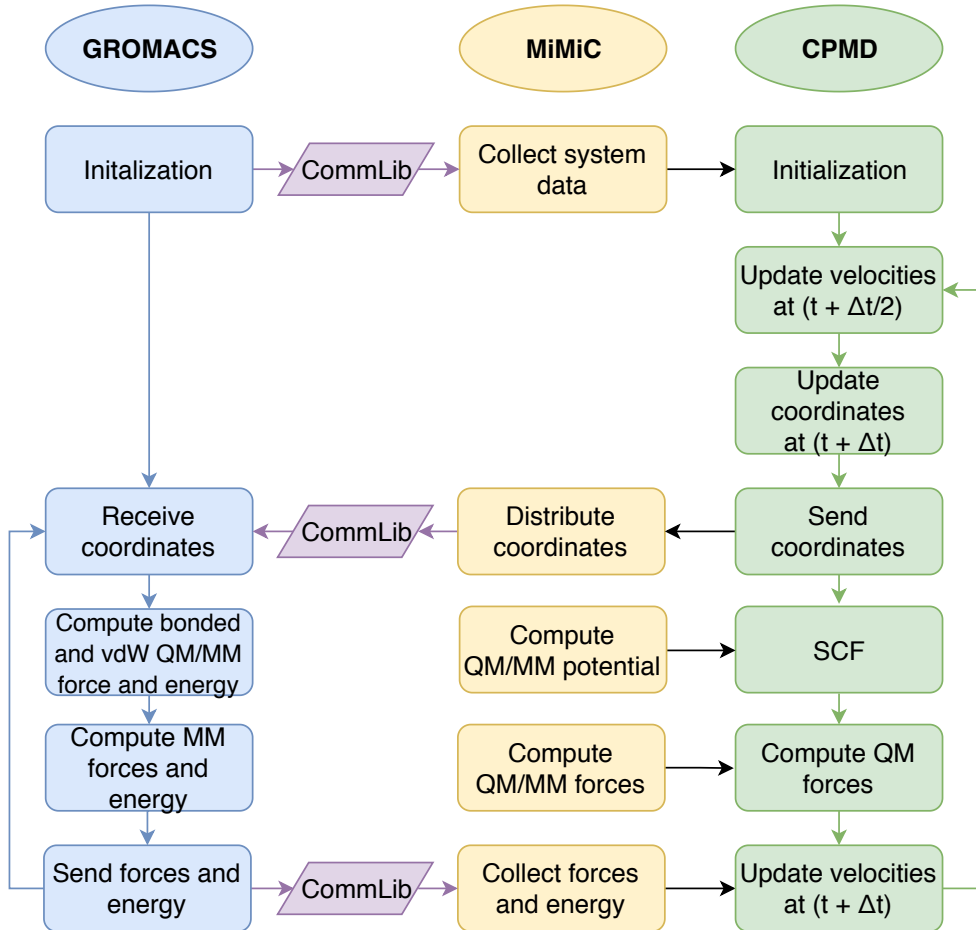


Figure 1: Illustration of the MiMiC-based QM/MM MD workflow using the velocity Verlet algorithm. The workflow shown here is for BO-MD but a CP-MD workflow would be analogous.

In the present work, the equations of motion are integrated by CPMD, including those of the MM atoms. This implies that bond, angle, and other constraints among MM atoms must be enforced by CPMD. The (serial) RATTLE algorithm⁵ implemented in CPMD as the standard constraint solver is suitable for systems of up to a few hundred atoms which is typical of pure QM systems but may become a computational bottleneck for larger systems (hundreds of thousands of atoms are not uncommon in MM). To overcome this problem,

we implemented an improved version of RATTLE proposed by Weinbach and Elber.⁶ This improved version takes advantage of sparse linear algebra to impose the constraints, which greatly reduces the associated computational cost. Even though the method is in principle fully parallel, here we implement a version relying only on OpenMP parallel linear algebra. MPI and hybrid MPI/OpenMP versions of the method will be implemented in the future to further enhance the scalability.

2.3 Communication Library

The loose-coupling model used in the MiMiC framework requires communication of data, e.g. atomic positions and forces, between the coupled programs, which are running independently. To achieve this objective MiMiC provides a communication library (CommLib), which has been designed to minimize the coding effort in the external programs used by the MiMiC framework. In fact, CommLib does not interfere with preexisting communication layers within the coupled programs, e.g. their MPI parallelization scheme does not need to be altered. In practice, one simply invokes the send and/or receive routines in CommLib to exchange data between two programs using an ID associated to a specific domain of the multiscale representation (e.g. QM, MM, etc.) which is created during the CommLib initialization.

To avoid interference with the MPI communication layers, which may be present in the coupled programs, CommLib uses client-server *inter-communication* which is an MPI2 feature. This allows direct exchange of data between processes belonging to different communicators, i.e. the `COMM_WORLD` communicators of the independent programs. Moreover, we can exploit the features of the communication fabric (e.g. Infiniband or Intel Omni-Path architectures) available on the system to reduce the communication overhead. Indeed, our benchmarks show that the overhead introduced by CommLib is negligible compared to the wall time even for large systems (see Sec. 4). The CommLib can be extended to use other communication mechanisms in cases where MPI2 is not available without affecting its use in

the coupled programs.

2.4 Parallelization Strategies

The MiMiC framework relies on a loose-coupling approach that allows the external programs to run concurrently using their own parallelization scheme, which can be based on, e.g., MPI, OpenMP, or GPGPU. Indeed, as detailed above (see subsection 2.2), the pure QM and MM contributions are computed by CPMD and GROMACS, respectively, using their own parallel strategies. Moreover, the bonded and van der Waals QM/MM contributions, which are treated at the MM level, are also computed by GROMACS. On the other hand, the electrostatic QM/MM interactions are directly computed by MiMiC and here we describe their parallelization which is based on a hybrid MPI/OpenMP approach.

Calculations of potential, energy, and force terms, eqs 1-7, requires summation over MM atoms in the short-, $N_{\text{MM}}^{\text{sr}}$, and long-range, $N_{\text{MM}}^{\text{lr}}$, domains and the integration of suitable observables, involving the electronic density, or the evaluation of the electrostatic potential from the MM atoms (eq 3) to be added to the KS-DFT external potential. In a plane wave representation of the KS orbitals, the two latter operations are performed on a regular grid discretizing the space. Our basic parallel strategy consists in distributing planes of the real-space grid among the MPI processes. While the QM/MM implementation presented in this work makes use of a plane wave implementation due to the use of CPMD to compute QM contributions, MiMiC is able to support any other type of basis set that allows for a mapping onto real-space grids. A grid-distribution strategy works well as long as the number of MPI processes does not exceed the number of grid planes. To extend the scalability beyond this limit one can use two complementary approaches, task grouping⁷⁻⁹ and OpenMP fine-level parallelism, or a combination of the two.

In our task group approach, the short- and long-range MM atoms are divided among the groups, and the grid planes are distributed between the MPI processes of each group (see illustration in Fig. 2). Thus, each MPI process receives a subset of short- and long-range MM

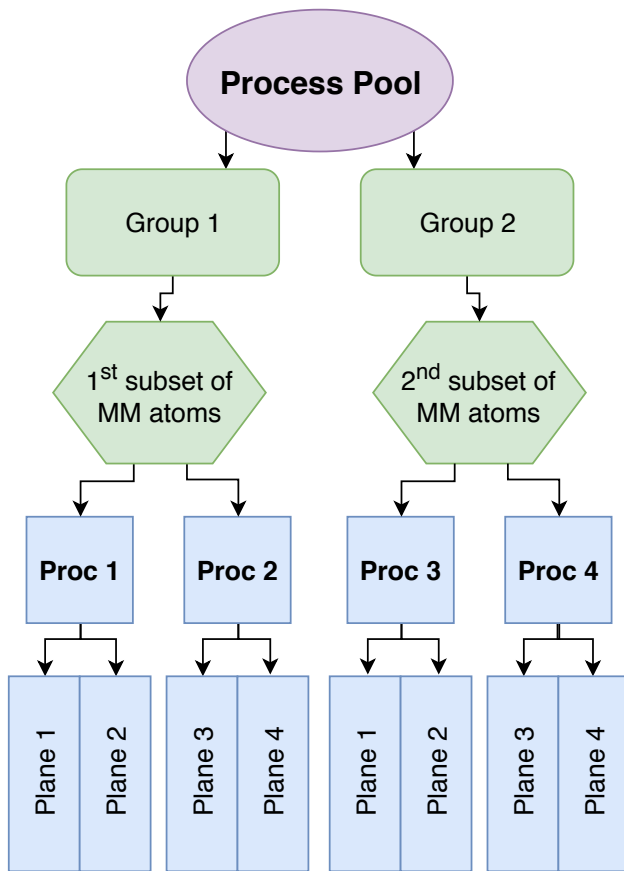


Figure 2: Illustration of the task group parallelization strategy used in the MiMiC-based QM/MM. Here, the MPI processes are divided into groups, each receiving a subset of MM atoms, and each MPI process of a group receiving a subset of grid planes.

atoms and a subset of grid planes, and computes the corresponding quantities of eqs 3–7. The results of the MPI processes are then suitably summed up to obtain the final values.

Within the OpenMP fine-level parallelism, the grid plane(s) assigned to an MPI process are divided into pencils, and each thread of the process is in charge of a pencil. In the long-range electrostatics, the OpenMP threads are also used to parallelize over the MM atoms, as it is done within the task group approach.

It is worth remarking that the parallelization approach described above has equivalents in the parallelization strategy adopted in CPMD (low-level distribution over planes,¹⁰ task grouping (known as *CP groups*),⁹ and threading¹¹), and that the task group and OpenMP approaches can be combined together to obtain the best overall parallel efficiency of a MiMiC-

based QM/MM simulation.

3 Computational details

The performance of our MiMiC-based QM/MM implementation was tested in a number of benchmarks. These report the speedup (i.e. $S_N = \frac{t_1}{t_N}$ where N is the number of compute nodes and t_1 is the timing of one node), strong scaling efficiency ($PE = \frac{S_N}{N} * 100\%$), and absolute timing per MD step. As a rule of thumb, we define a scaling limit as the amount of cores at which the parallel efficiency stays above a 70% threshold. The systems differ by their size and complexity (see below). They are treated using either the BLYP^{12,13} or B3LYP^{14,15} exchange-correlation (XC) functionals. The former is a typical example of a generalized-gradient approximation (GGA), which scales linearly in a plane wave basis, whereas the latter constitutes an example of a hybrid functional, which scales quadratically with respect to the numbers of orbitals in plane waves. The hybrid functional implementation in CPMD can exploit thousands of cores,⁹ thus using B3LYP allows us to examine the scaling limits of MiMiC under highly parallel conditions.

Benchmarks were performed on clusters hosted by the Jülich Supercomputing Centre and RWTH Aachen, namely JURECA,¹⁶ JUWELS,¹⁷ and CLAIX-2018.¹⁸ All three clusters have a fat-tree topology. JURECA’s compute nodes are equipped with two Intel Xeon E5-2680 12-core CPUs while JUWELS has two 24-core Intel Xeon Platinum 8168 CPUs per node. The nodes in CLAIX-2018 have two 24-core Intel Xeon Platinum 8160 CPUs.

The set of benchmark systems consist of two solute-solvent systems and two protein systems. They were chosen to represent a diverse composition and broad range of sizes for the QM and MM systems making it possible to explore the scaling behaviour of several combinations of QM and MM systems. The solute-solvent systems include *n*-butanol solvated in acetone solution (Fig. 3) and N-(6-methoxyquinolyl) acetoethyl ester (MQAE) in a water solution (Fig. 4). The protein systems consist of a zinc-binding GB1 mutant^{19,20} (Fig. 5) and

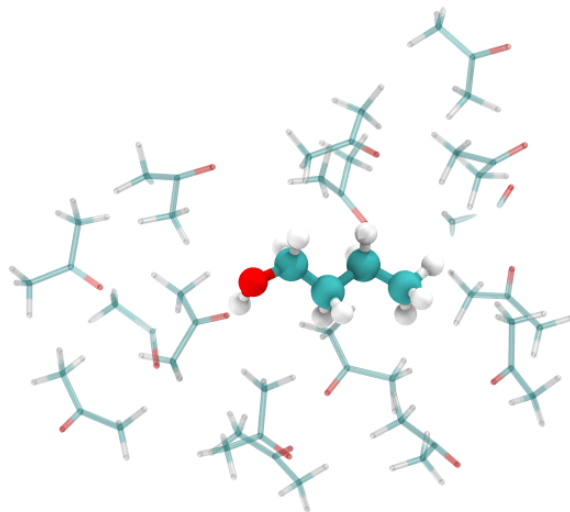


Figure 3: Butanol molecule solvated in acetone solution. Only part of the total system is shown. The QM subsystem is displayed in ball-and-stick representation.

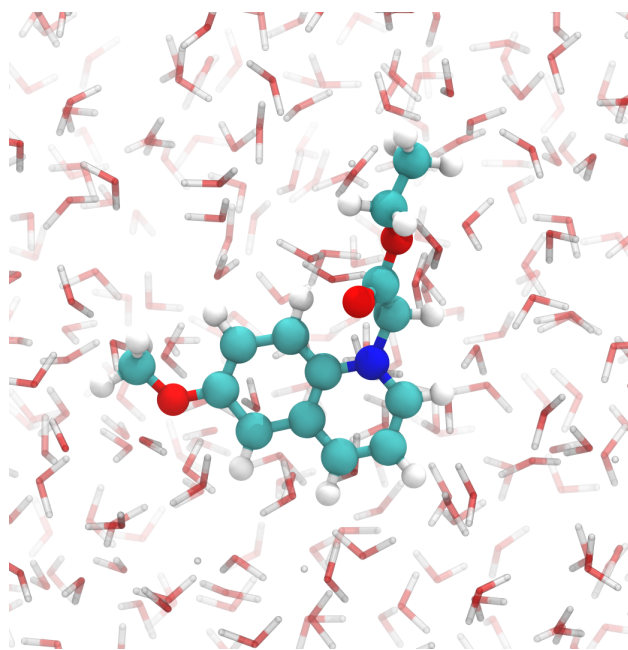


Figure 4: MQAE in water solution. Only part of the total system is shown. The QM subsystem is displayed in ball-and-stick representation.

a chloride ion channel (ClC-ec1) embedded in a lipid bilayer²¹ (Fig. 6). Initial structures for the butanol in acetone and GB1 systems were taken from Ref. 1. Both BO-MD and CP-MD were investigated in the benchmarks. We also compared the performance of the MiMiC-based QM/MM to the existing QM/MM implementation in CPMD.⁴ We used a time step of

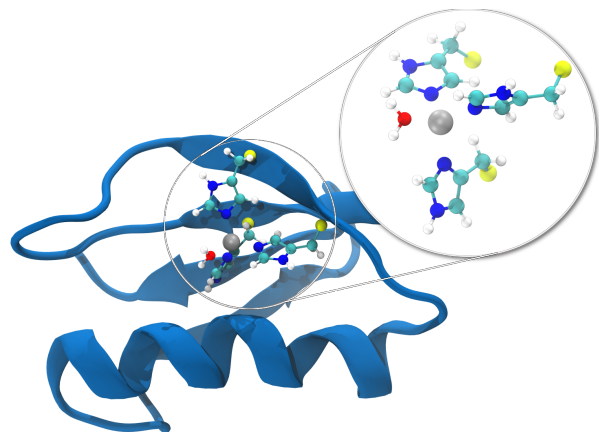


Figure 5: Zinc-binding GB1 mutant. Water solvent is not shown. The QM subsystem is displayed in the inset in ball-and-stick representation. Boundary atoms are highlighted in yellow.

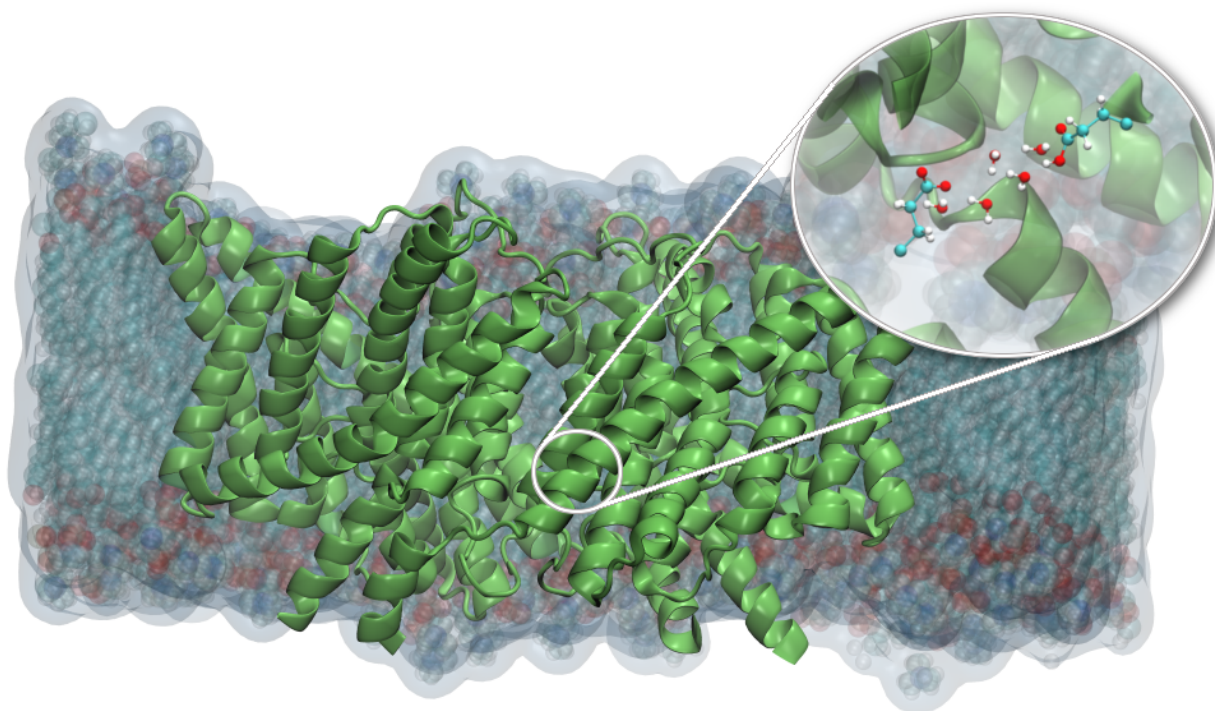


Figure 6: ClC-ec1 ion channel in a lipid bilayer. Water solvent is not shown. The QM subsystem is displayed in the inset in ball-and-stick representation.

20 au (~ 0.48 fs) for BO-MD and 5 au (~ 0.12 fs) for CP-MD. Scaling of the MiMiC-based QM/MM is reported with respect to the amount of cores allocated for CPMD. In these benchmarks it was not necessary to allocate more than one compute node to GROMACS.

The butanol in acetone system consists of one butanol (QM) and 13,948 acetone molecules

(MM) giving a total of 139,495 atoms, which thus represents a system with a *small* QM subsystem and a *large* MM subsystem. The 16 orbitals of butanol were expanded on a real-space FFT mesh of 240^3 points corresponding to a plane wave cutoff of 85 Ry. The number of atoms in the solvated MQAE system is 16,396 of which either 34 (MQAE) or 151 (MQAE + $39 \times \text{H}_2\text{O}$) are in the QM subsystem, representing systems with a *small* and a *large* QM subsystem, respectively, together with a *small* MM subsystem. At a cutoff of 80 Ry, the 47 orbitals of the small QM subsystem are expanded on a mesh of 216^3 points, whereas the 203 orbitals of the large QM subsystem are expanded on a 288^3 mesh. The protein systems considered here also vary in size. The GB1 system contains 25,976 atoms in total, 40 of which were treated at the QM level (*small* QM and *small* MM). Using a cutoff of 85 Ry, the 57 orbitals were mapped on a mesh containing 240^3 points. The ClC-ec1 system consists of 150,925 atoms, where either 19 or 295 atoms were selected to represent a *small* and *large* QM subsystem, respectively, combined with a *large* MM subsystem. The real-space mesh for the small and large QM subsystems consist of 240^3 and $360 \times 300 \times 300$, respectively, corresponding to a 90 Ry cutoff. The amount of orbitals totalled 49 in the small QM subsystem and 383 in the large QM subsystem. Apart from the MQAE and ClC-ec1 setups with large QM regions, which were only treated at the BLYP level, benchmarks for all other systems were performed using the GGA XC functional BLYP as well as the hybrid-GGA XC functional B3LYP. All simulations were performed using Troullier–Martins norm-conserving pseudopotentials²² using the new XC driver by Bircher and Rothlisberger.²³ Note that the setups detailed above do not necessarily correspond to a realistic setup suitable for a production simulation but are prepared in order to demonstrate the performance of systems with different QM and MM subsystem sizes. The timings are given as averages over 5 steps in BO-MD and 10 steps in CP-MD. However, in the case of BO-MD using B3LYP, only 3 time steps were used due to the large amount of time needed when running on a single node.

Based on the low-level, distributed-memory parallelization scheme of CPMD, efficient

scalability can only be expected up to a number of CPU cores that equals the number of FFT planes in the real-space mesh. Higher scalability can be achieved by resorting to shared-memory parallelization; thus the benchmarks using MiMiC were performed using four OpenMP threads per MPI process up to the point where the grid scaling is exhausted. The number of threads per task was chosen because it was found to maximize the performance on the computing clusters used in this work. Attempts to use a number of cores larger than the number of FFT planes, were carried out either using more OpenMP threads per MPI process or by resorting to the high-level, distributed-memory task grouping scheme implemented in CPMD (known as CP groups).⁹ The number of task groups, generally, should be an integer divisor of the number of orbitals in order to achieve an even distribution of computations. Benchmarks for the QM/MM implementation in CPMD are presented based on an MPI-only version of CPMD as the addition of OpenMP threading substantially degraded the performance.

4 Performance Benchmarks

In this section, we present benchmarks of the MiMiC-based QM/MM implementation using CPMD and GROMACS as the QM and MM engines, respectively, for a set of systems with different QM and MM subsystem sizes. We investigate the scalability using the parallelization strategies presented in Section 2.4, namely the low-level grid distributions using MPI, the higher-level task grouping scheme, and the fine-level OpenMP threading. Here it should be noted that when we in the following refer to task groups in CPMD, this corresponds to using the CP group implementation as presented in Ref. 9. We also include a comparison of timings and scaling performance to the existing tight-coupling QM/MM implementation in the CPMD program.

4.1 GB1

We performed a set of benchmarks using the zinc-containing GB1 mutant that were aimed at measuring the scaling performance of both the MiMiC-based QM/MM and the existing QM/MM implementation in CPMD. We measured both the time spent computing the QM/MM interactions only and the total time per time step. The similarities of the QM/MM methodology used in MiMiC and the existing QM/MM implementation enable a direct comparison of performance. Results are presented in Figures 7 and 8, respectively. Based on the grid dimensions (240^3) and the parallelization strategy used in CPMD and MiMiC (see subsection 2.4), we expect to be able to use at least 240 cores efficiently for the existing QM/MM in CPMD (using MPI parallelization only) and 960 for the MiMiC-based QM/MM (using hybrid MPI/OpenMP parallelization).

Figure 7 shows the overall time per time step along with speedups for both CP and BO dynamics. Using two task groups and four OpenMP threads per MPI process, we were able to efficiently use up to 1,920 cores. At this point one BO time step requires only 1.16 second and one CP time step is completed in 0.31 seconds. This can be compared to the existing QM/MM implementation that does not scale beyond 240 CPU cores as expected, which corresponds to 8.50 seconds per BO time step and 2.26 seconds per CP time step. It is also noticeable that in the BO benchmark (Fig. 7b), we observed superlinear scaling at core counts below 1,000. The reason for such a peculiar observation is the scaling performance of the plane wave FFT implementation, which has been reported to scale with more than 100% for certain system sizes.²⁴ This has been attributed to an improved CPU cache utilization as the number of cores increases. For instance, the CPUs in JURECA have 30 MB L3 (level 3) cache shared across all cores in addition to 32 KB and 256 KB of L1 (level 1) and L2 (level 2) caches, respectively, on each core. The size of the FFT mesh for GB1 is estimated to be roughly 100 MB. With two CPUs on a single node, the grid is split between those into 50 MB portions. This means that the portion available for a single CPU does not fit into the cache available on said chip. However, as the number of CPUs increases, the

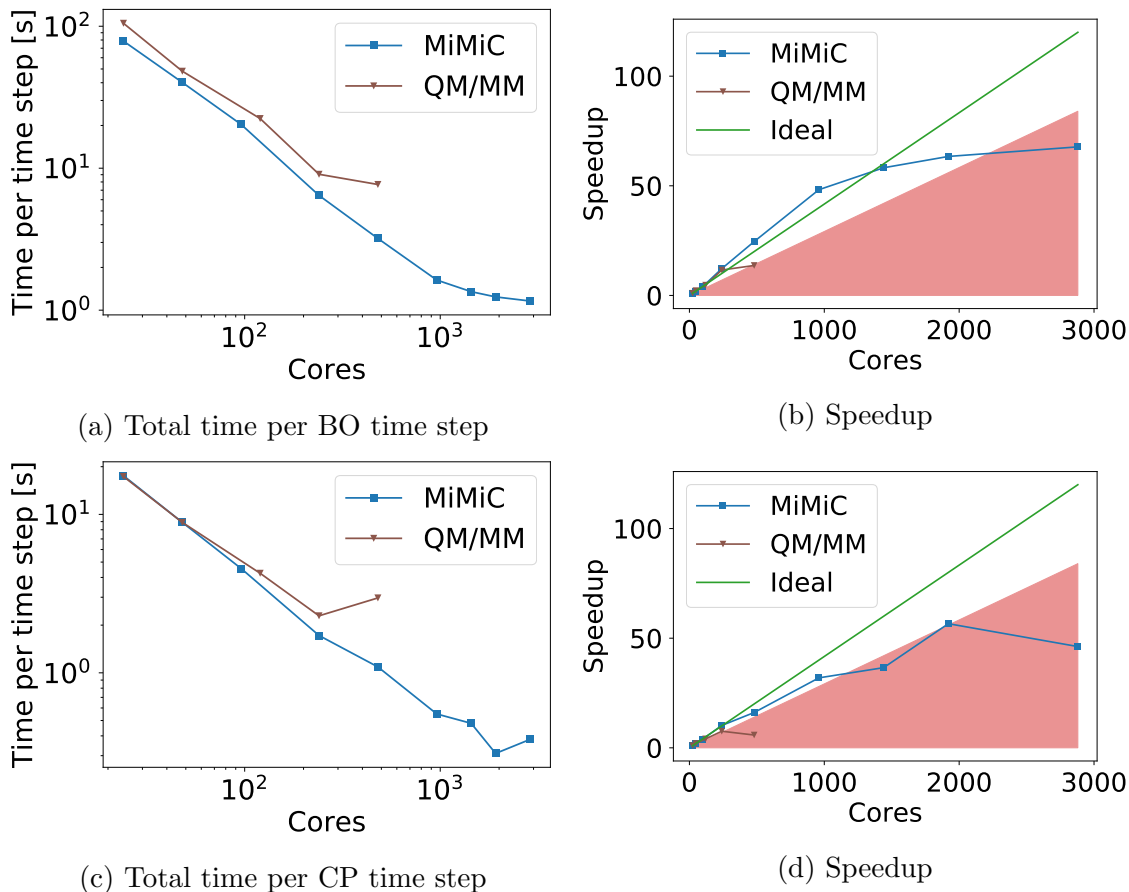


Figure 7: Scaling performance of the GB1 system both within BO-MD and CP-MD using BLYP. Simulations were run on JURECA. QM/MM refers to the existing QM/MM implementation in CPMD. Red area shows the 70% scaling efficiency threshold.

portion sizes per CPU decrease. First, portions will reach a size where they fit into the L3 cache and, once the grid is decomposed even further, the part of the grid treated by a single core will ultimately fit into the L2 cache. The speedup obtained through better cache utilization due to more fine-grained grid distribution can compensate the slowdown due to the communication overhead up to 40 nodes (960 cores). After this point we do not observe the superlinear behavior anymore.

In Figure 8, we present the timings used to compute the QM/MM contributions only. Here it can be seen that the limiting factor in the previous QM/MM implementation is the computation of QM/MM contributions. With MiMiC, on the other hand, a much higher number of cores can be used for the same system due to efficient parallelization. In fact,

we can efficiently utilize at least 2,880 cores, which means that the main factor limiting the scaling in the MiMiC-based QM/MM is in CPMD. The gain in performance achieved by employing MiMiC is especially prominent in BO-MD. This is due to the fact that the existing QM/MM implementation computes QM/MM potentials and forces twice per time step — in the beginning and in the end of the SCF loop. In the MiMiC-based QM/MM, however, the QM/MM potential is computed by MiMiC in the first iteration of the SCF loop and stored by CPMD, and the QM/MM forces are only computed after convergence is reached.

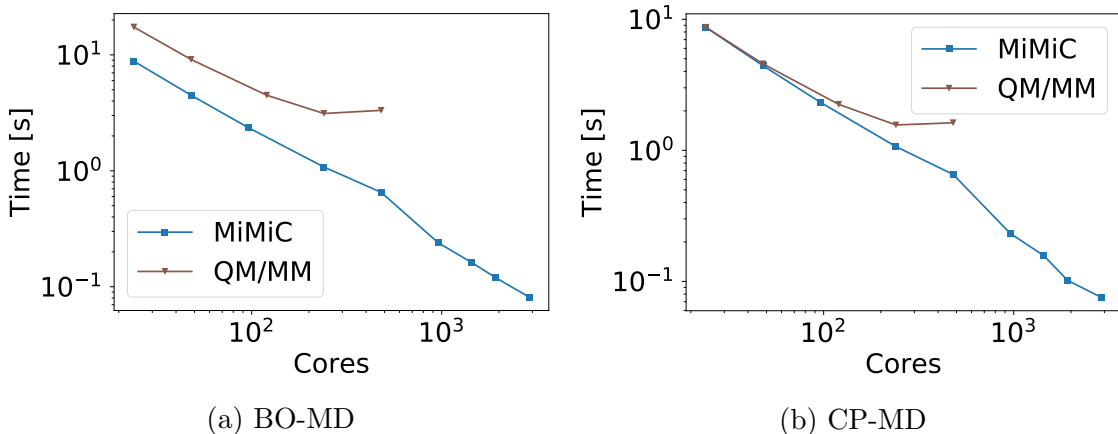


Figure 8: Scaling performance of the GB1 system including only the time spent computing QM/MM contributions. Simulations were run on JURECA. QM/MM refers to the existing QM/MM implementation in CPMD.

4.2 MQAE

The MQAE system is relatively small both with respect to QM and MM sizes. In order to benchmark the performance on systems with small and large QM subsystems combined with a small MM subsystem, we ran two sets of benchmarks: in the first one we included only the solute in the QM part and in the second we also added solvent molecules that were within 3.0 Å of the solute.

Benchmarks using the small QM system with BLYP show that MiMiC retains high scaling efficiency up to 1,296 cores within BO-MD with a total time per time step equal

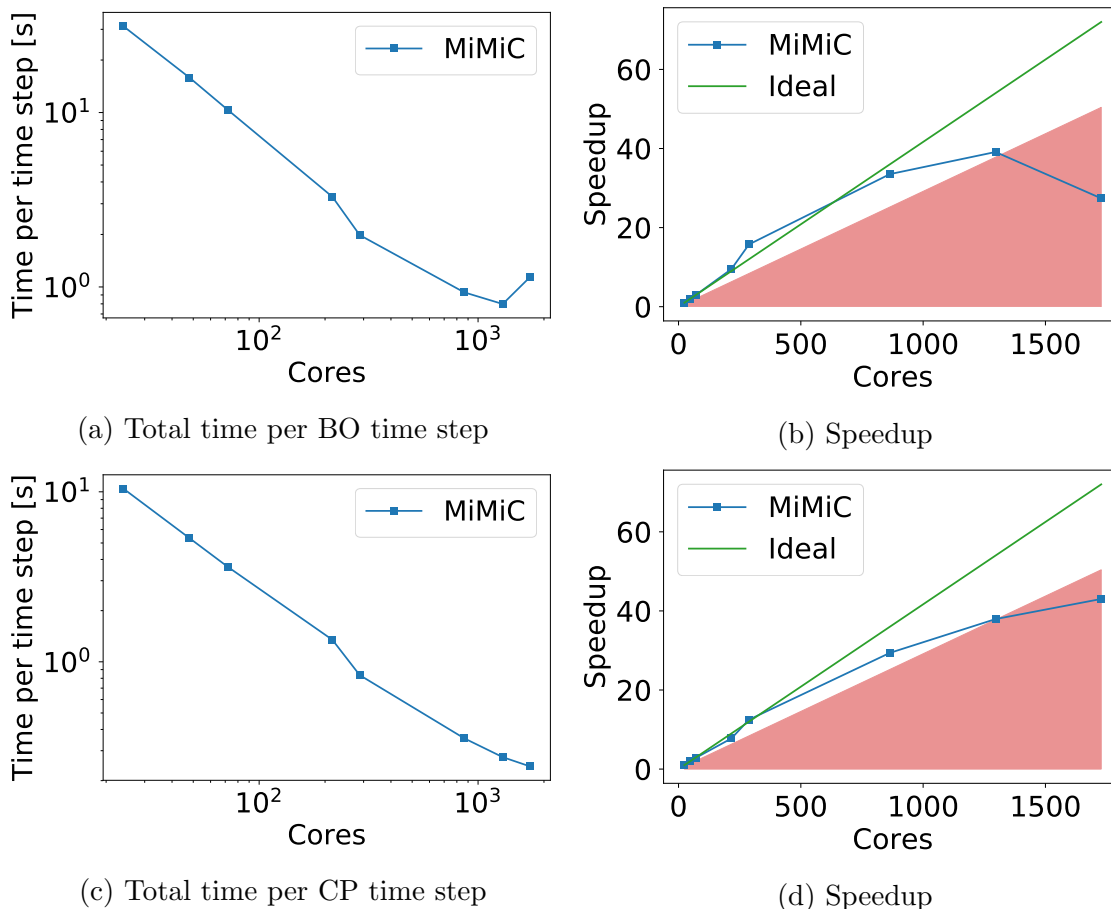


Figure 9: Scaling performance of the MQAE system with the small QM subsystem both within BO-MD and CP-MD using BLYP. Simulations were run on JURECA. Red area shows the 70% scaling efficiency threshold.

to approximately 0.8 seconds (see Fig. 9). With four OpenMP threads per MPI process the scaling via grid decomposition is exhausted at 864 cores. Further scaling of BO-MD is possible by using six threads per process. Using task groups does not extend the scalability any further (last point in Fig. 9b). CP-MD behaves in a similar way allowing use of 1,296 cores with 0.28 seconds per time step. A superlinear behavior is observed near 300 cores, which can be attributed to the phenomena described in subsection 4.1.

The benchmarks based on the B3LYP XC functional show that the use of two task groups extends the scaling to 1,728 cores (see Fig. 10). The scaling is limited by the fact that the number of orbitals in this system is a relatively small prime number (47), which inevitably leads to load imbalance of the exact exchange computation that in turn constitutes the

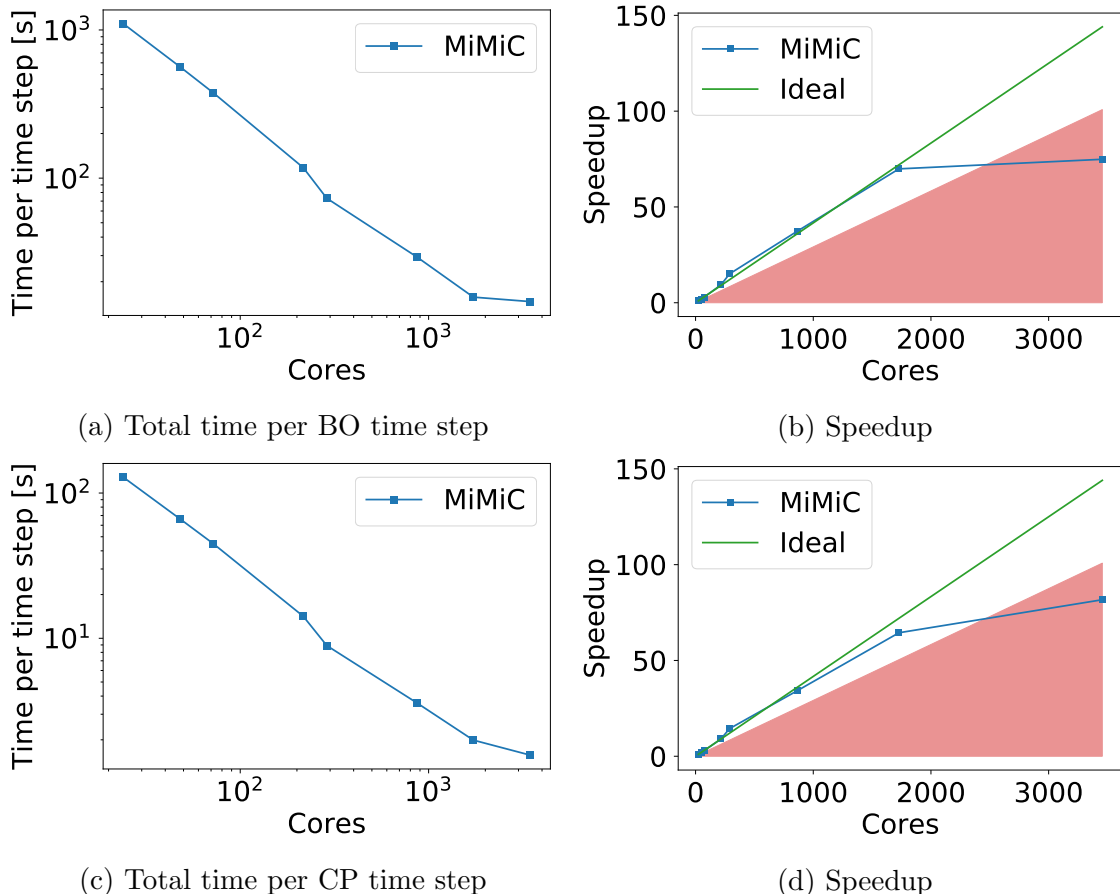


Figure 10: Scaling performance of the MQAE with the small QM subsystem both within BO-MD and CP-MD using B3LYP. Simulations were run on JURECA. Red area shows the 70% scaling efficiency threshold.

bottleneck of the calculation both with respect to execution time and parallelization. Finally, enforcing the orbital orthogonality constraint of the CP-MD on each time step, even though it is parallelized, may further hamper scaling of a pure CP-MD run. Therefore, the maximum achievable performance for B3LYP simulations was 15.88 second per time step of BO-MD and 2.00 seconds per time step for CP-MD.

It can be expected that simulations using the larger QM subsystem will have a scaling limit which is at least 1.3 times higher based on the larger dimension of the grid (288^3). However, as seen in Figure 11, the BLYP-based BO-MD simulation can use over 2,300 cores by employing two task groups (which is roughly 1.8 times more compared to the scaling limit of the smaller system), thus bringing the total time per time step down to about 3.8

seconds. The scaling performance of CP-MD, on the other hand, is as expected, thus allowing efficient use of 1.3 times more cores (1,152) with a total time per time step of 0.81 seconds. This difference between BO-MD and CP-MD can be caused by multiple factors. First, the serial parts of the integrator, including the constraint solver (which is only parallelized using threading), stalls the performance in the sub-second range. Then, the use of task groups adds more communication in CP-MD where it is associated to the computation of the orbital overlap between groups; this substantial communication step is absent in GGA-only BO-MD. These factors taken together explain the superior scaling of the BO-MD simulation with respect to its CP counterpart.

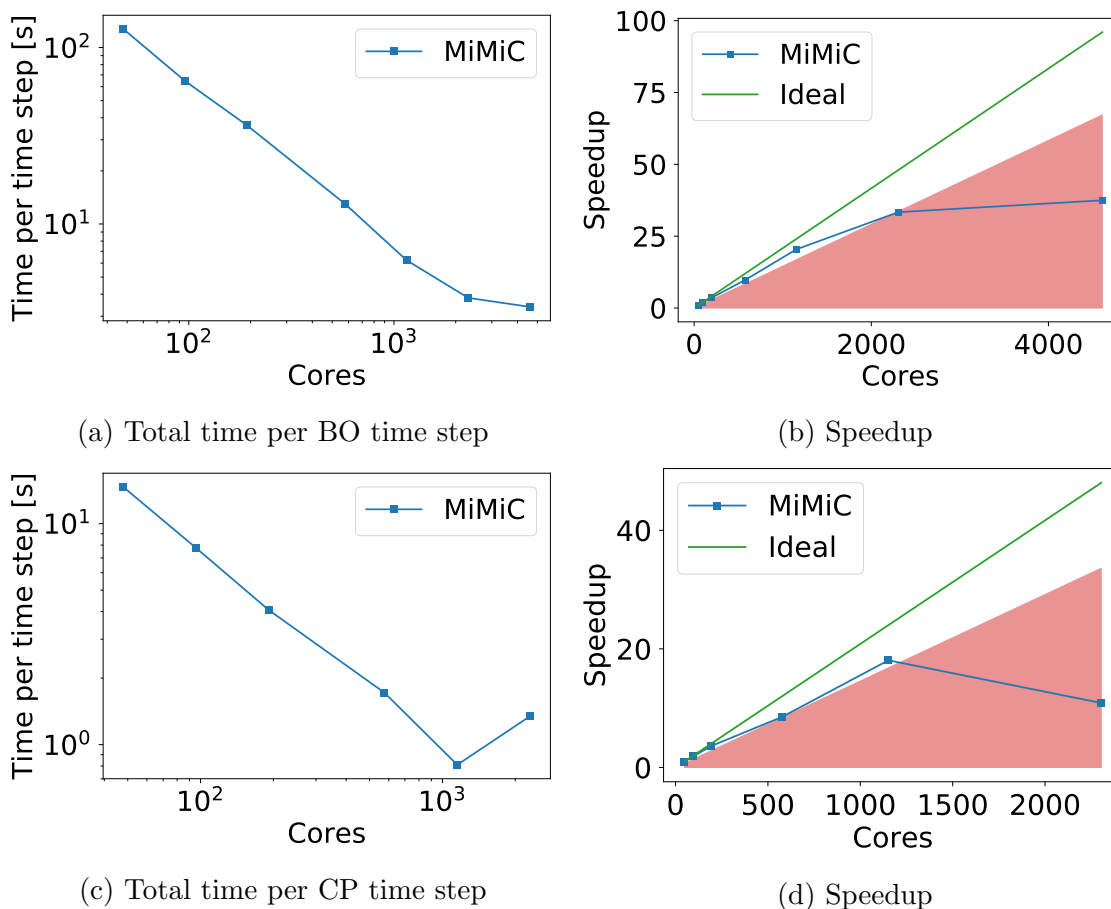


Figure 11: Scaling performance of the MQAE system with the large QM subsystem both within BO-MD and CP-MD using BLYP. Simulations were run on CLAIX-2018. Red area shows the 70% scaling efficiency threshold.

4.3 Butanol

This system has a very small QM subsystem (15 atoms) combined with a large MM subsystem (139,495 atoms). Therefore, it is expected that the QM/MM part plays a major role performance-wise.

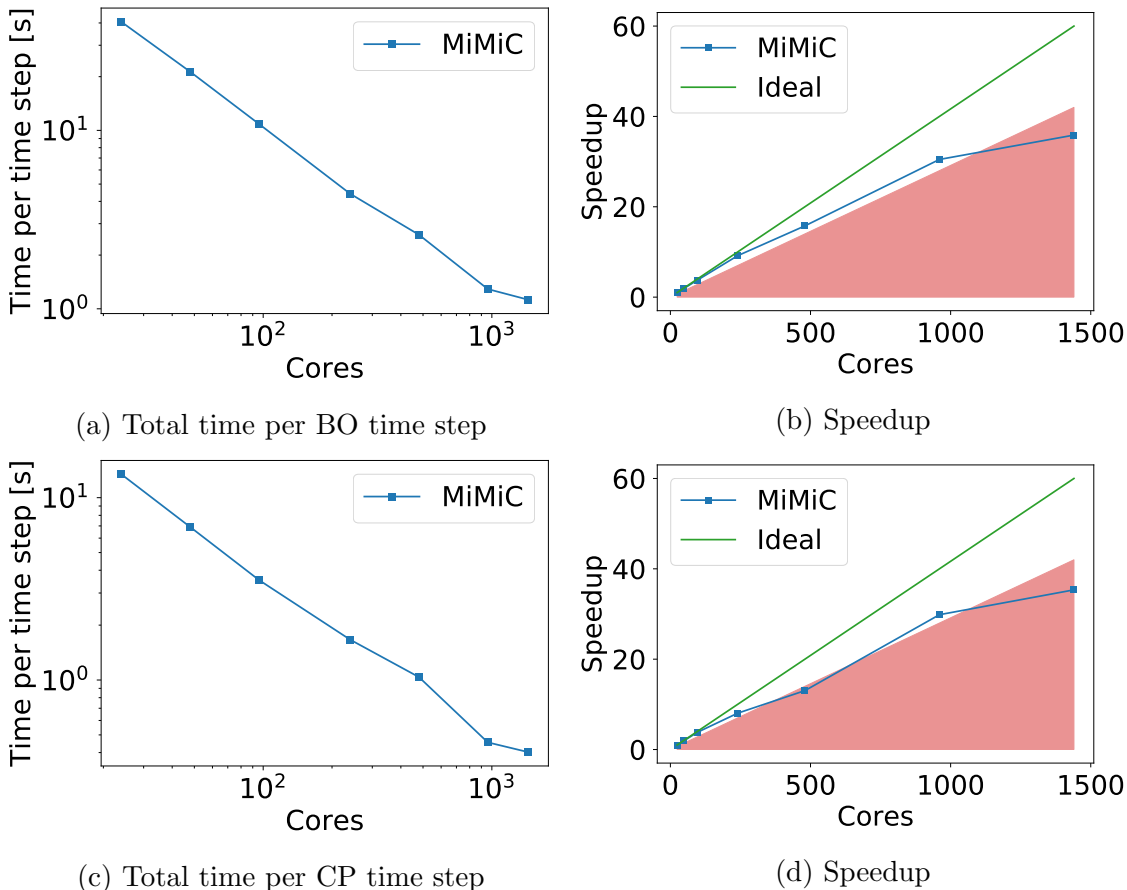


Figure 12: Scaling performance of the butanol system both within BO-MD and CP-MD using BLYP. Simulations were run on JURECA. Red area shows the 70% scaling efficiency threshold.

Results of BLYP-based CP-MD benchmarks are shown in Figure 12c. As expected, QM/MM contributions are especially prominent here, where they take up to 82% of the time on a single node, as shown in Figure 13a. Thanks to the efficient parallelization and excellent scaling of the QM/MM contributions computed by MiMiC, the time per time step can be brought down to approximately 0.45 seconds using 960 cores (see Fig. 12c). Further scaling is stalled due to the small size of the FFT grid (240^3) and the small number of orbitals

(16), which limits the benefits of task groups in CPMD. Moreover, there are issues due to the serial integrator that have been pointed out above and, in addition, the time spent on communication between CPMD and GROMACS is non-negligible at this core count. These bottlenecks appear mostly due to the small size of the QM part compared to the substantial extent of the MM subsystem. The detailed breakdown of the time spent computing each of the contributions at the scaling limit is shown in Figure 13b.

BLYP-based BO-MD exhibits a similar scaling behaviour as the BLYP-based CP-MD and scales up to the same limit of 960 cores (Fig. 12a), pinpointing yet again that the scaling differences in the QM part are less important due to the dominant effect of the QM/MM contributions. Again, the efficient parallelization scheme allows us to substantially reduce the time needed for a BO-MD step (1.29 seconds). Further scaling is mostly hampered by the bottlenecks summarized for CP-MD above.

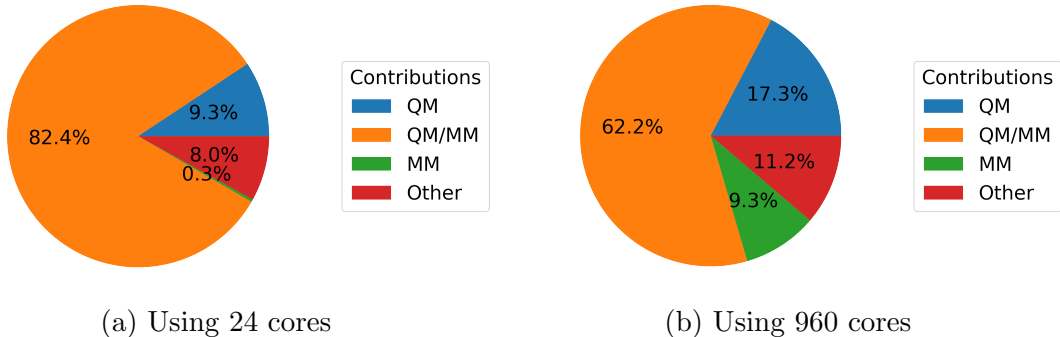


Figure 13: Breakdown of time spent on one time step of CP-MD using BLYP for butanol. Section "Other" includes the integrator with auxiliary functions (e.g. computation of observables)

In the B3LYP-based simulations, the use of two task groups extends the scaling limit of a BO-MD by a factor of two (Fig. 14). This makes it possible to use up to 1,920 cores with an efficiency of 85% and a time per time-step of about 5.3 seconds only. Further addition of task groups, though, causes a decline in performance, typical of a system with a small number of orbitals where the additional communication between groups in the QM part outweighs the gain in performance due to task distribution. Similarly, for the B3LYP-based CP-MD run,

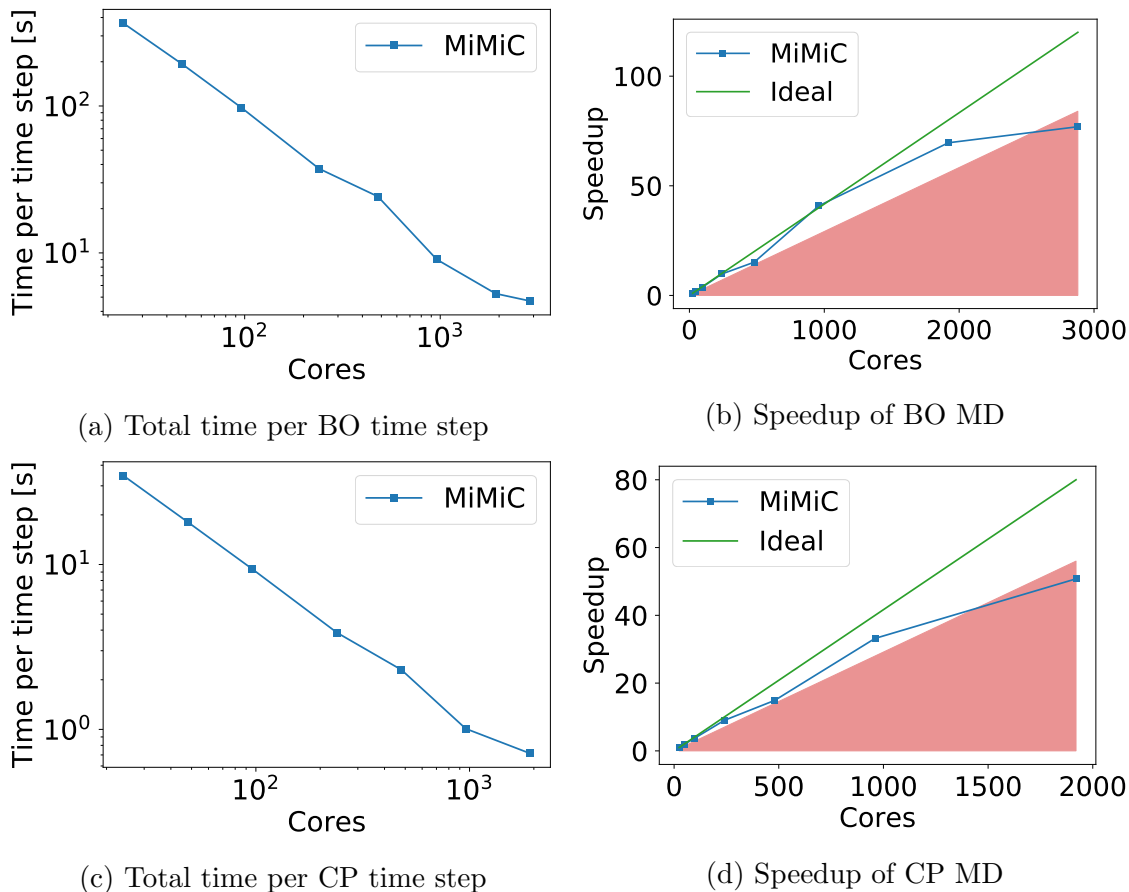


Figure 14: Scaling performance of the butanol system both within BO-MD and CP-MD using B3LYP. Simulations were run on JURECA. Red area shows the 70% scaling efficiency threshold.

the application of task grouping does not improve the scaling limit, which is the same as the one observed for BLYP. Upon investigation, we found that the reasons for the decline in performance are of the same nature as the stalling of the scaling performance of BLYP-based CP-MD outlined above (i.e. the serial integrator and the communication overhead).

4.4 ClC-ec1

The ClC-ec1 system has the largest number of MM atoms considered here (150,925). Therefore, QM/MM interactions are of paramount importance for simulations of this system. In order to examine effects of both small and large QM subsystems combined with a large MM subsystem, we also included a setup with a QM subsystem that consists of the smaller QM

subsystem and all residues that are within 3 Å of it.

The behavior of the BLYP-based simulations with a small QM system is similar to the one of butanol (see Fig. 15). BO-MD scales up to 960 cores with a single time step taking 6.53 seconds. However, there is a drastic decline in the CP-MD performance. For this special case, it is not possible to use more than 480 cores with reasonable efficiency. The reason for this lies in the large size of the MM subsystem and the associated large number of constraints. This, together with the serial velocity- and coordinate-updates in the CPMD integrator, stalls the scaling (see Fig. 16).

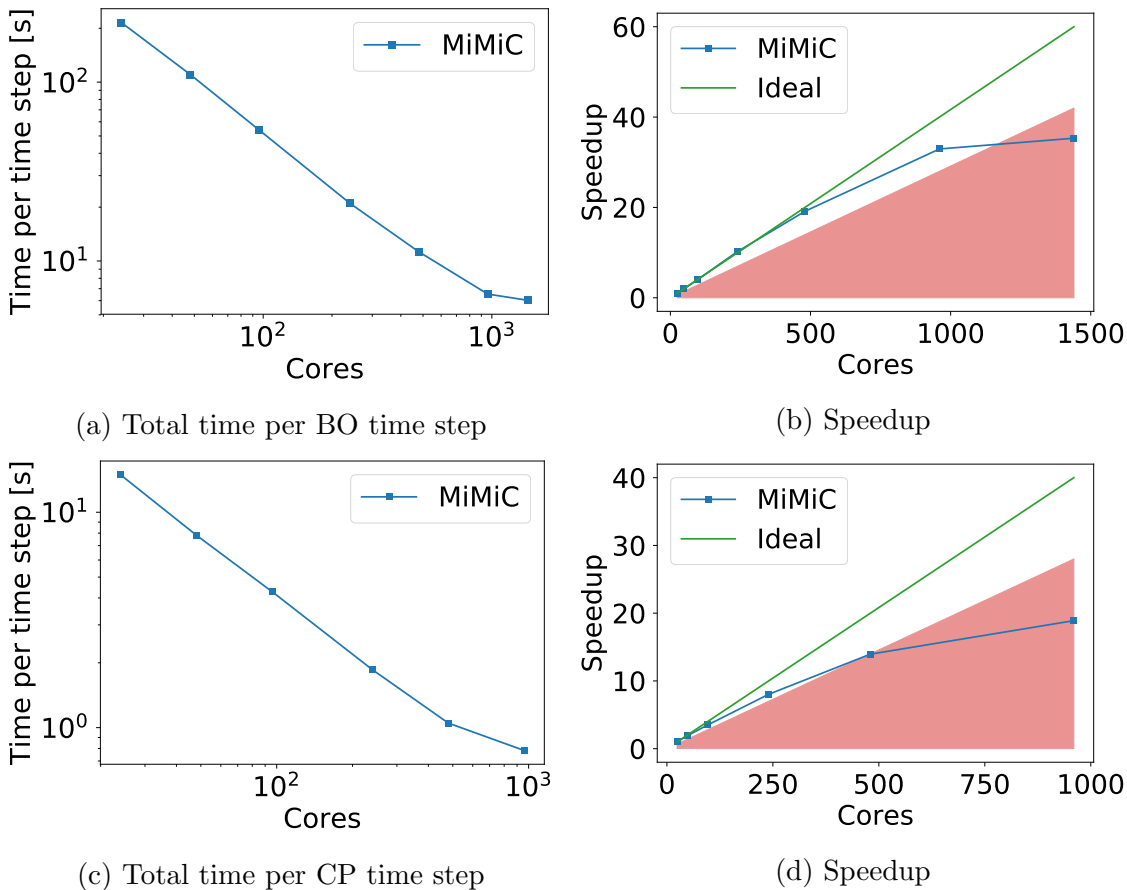


Figure 15: Scaling performance of the CIC-ec1 system with the small QM subsystem both within BO-MD and CP-MD using BLYP. Simulations were run on JURECA. Red area shows the 70% scaling efficiency threshold.

The picture is rather different when the small QM subsystem is treated at the B3LYP level. These benchmarks have been run on the JUWELS supercomputer, due to the signifi-

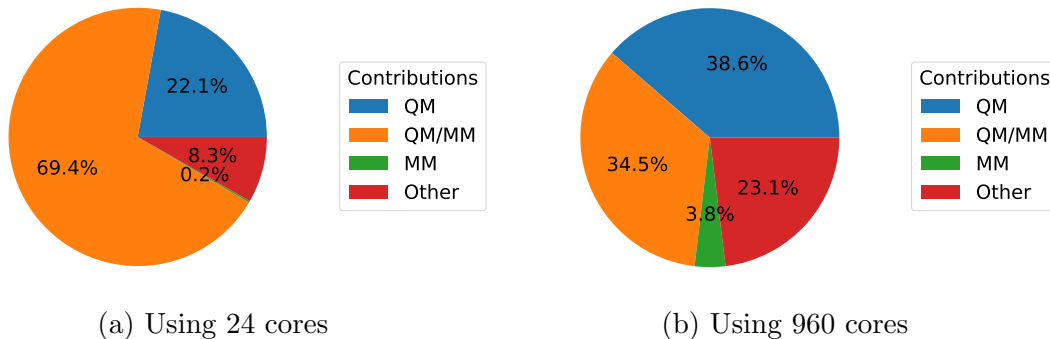


Figure 16: Breakdown of time spent on one time step of CP-MD using BLYP for ClCec1 with the small QM subsystem. Section "Other" includes the integrator with auxiliary functions (e.g. computation of observables)

cant computational resources required. The use of task groups makes it possible to push the scaling limit up to 13,440 cores by using 14 groups, thus reducing the wall time spent per each BO-MD time step to around 13.76 seconds (Fig. 17). In line with the scaling behaviour observed for the other systems, the use of task groups is less beneficial for CP-MD. Even so, it is possible to use two task groups, thus pushing the scaling limit to 1,920 cores. The stalling of the scaling can again be attributed to the inferior scaling of the orthogonalizer used to enforce the orthonormality constraint in CP dynamics at every time step, as well as the number of orbitals (49) where, with its large prime factors, it is not trivial to achieve a reasonable load distribution.

By considering a setup with an enlarged QM subsystem, the performance of the MiMiC-based QM/MM can be benchmarked under conditions where both QM and MM subsystems are large. This system also requires use of a non-cubic QM box. Since the QM mesh has 1.5 times more points along one dimension, an improvement in the scaling limit is expected due to better scaling in the low-level grid parallelization. The results for this setup can be found in Figure 18. Indeed, BLYP-based BO-MD scales excellently, up to 4,320 cores with an efficiency above 70% (rather than 980 cores as for the smaller QM subsystem). The scaling of CP-MD also shows an impressive improvement, extending the scaling limit observed for the small QM subsystem by as much as a factor of 4.5. The cost of force computations,

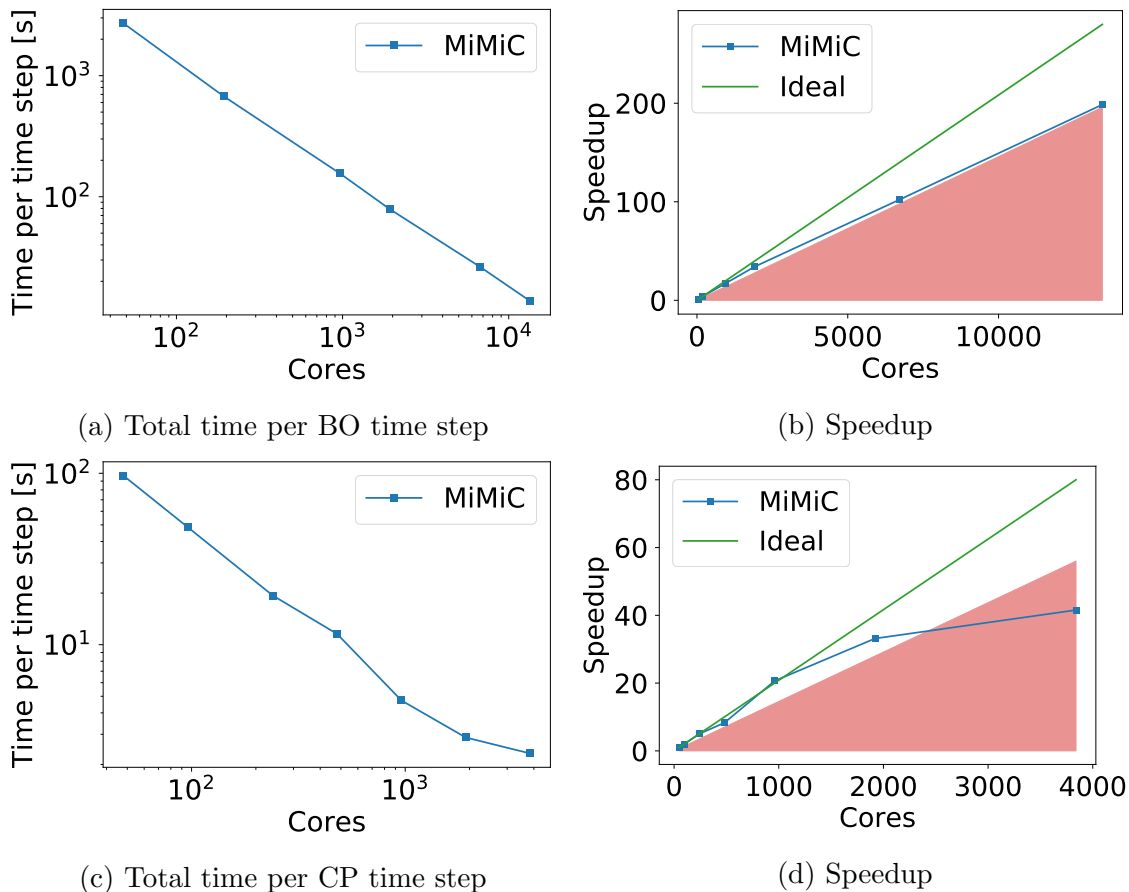


Figure 17: Scaling performance of the CiC-ec1 system with the small QM subsystem both within BO-MD and CP-MD using B3LYP. Simulations were run on JUWELS. Red area shows the 70% scaling efficiency threshold. The first point of the graphs in (a) and (b) is extrapolated because of excessive computational cost of a single time step.

which are highly parallelized, becomes more dominant compared to the orthogonalization of orbitals and the constraint solver, which both on their own have shown inferior scaling with respect to the computation of forces. Additional benefits come from the efficient use of threading. Here it is possible to increase the number of OpenMP threads per MPI process from four to six together with two task groups for BO-MD. In the CP-MD simulations, up to six threads per MPI process were used, but yet again, task groups did not offer any performance improvement. Therefore, the best possible timings obtained for BO-MD and CP-MD with BLYP are 10.43 and 1.42 seconds per time step, respectively.

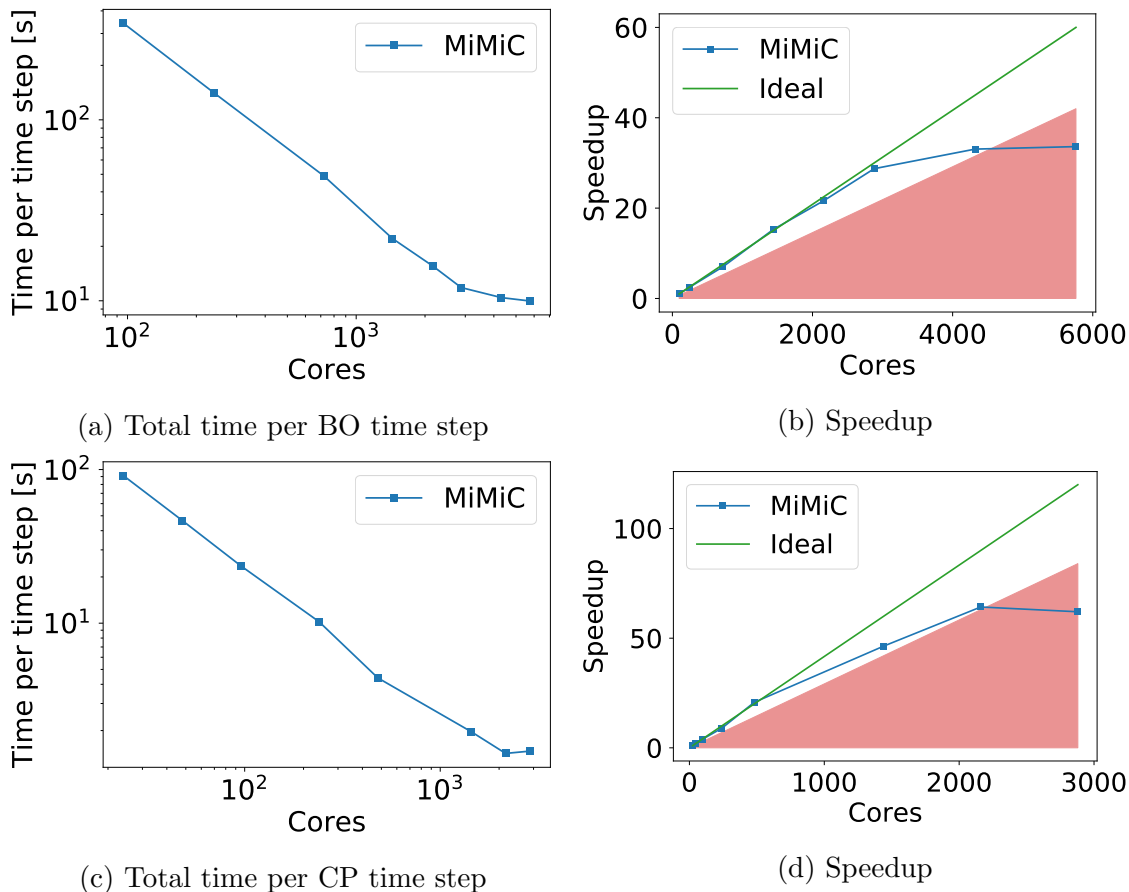


Figure 18: Scaling performance of the CIC-ec1 system with the large QM subsystem both within BO-MD and CP-MD using BLYP. Simulations were run on JURECA. Red area shows the 70% scaling efficiency threshold. The reference for BO-MD is four nodes instead of one due to the limited amount of memory available on a single node.

5 Summary

In this work, we presented the parallelization strategy and performance benchmarks of the MiMiC-based QM/MM implementation¹ using CPMD and GROMACS as the QM and MM engines, respectively. We detail the technical and algorithmic improvements that allowed our implementation to outperform the existing QM/MM interface in CPMD by roughly an order of magnitude (in terms of wall time per MD time step). Exceptional parallel efficiency is achieved through an efficient parallelization of the QM/MM contributions and by exploiting the highly parallel CPMD and GROMACS packages, both of which run concurrently which is enabled by the loose coupling MPMD approach used in MiMiC.

We demonstrate efficient utilization of several thousand cores bringing the average time per time step down to the few-seconds range for BO-MD and sub-second range for CP-MD. A major improvement of the parallel algorithm is achieved via combined use of a MPI-based task group approach and OpenMP threading. This paves the way for QM/MM simulations in the timescale of nanoseconds within the duration of a typical computational project even when using hybrid XC functionals and enables investigations of large and complex biological systems within the plane-wave/pseudopotential DFT-based QM/MM.

Author Information

Author contributions

Conceptualization: J.M.H.O. (equal), S.M. (equal), and U.R. (equal); Software: V.B. (equal), J.M.H.O. (equal), S.M. (supporting), and M.P.B. (supporting); Investigation: V.B. (lead); Resources: P.C. (equal) and E.I. (equal); Visualization: V.B. (lead); Writing – Original Draft: V.B. (lead), J.M.H.O. (supporting), S.M. (supporting), M.P.B. (supporting) and E.I. (supporting); Writing – Review and Editing: V.B. (equal), J.M.H.O. (equal), S.M. (equal), E.I. (equal), M.P.B. (equal), P.C. (equal), and U.R. (equal); Supervision: J.M.H.O. (equal), U.R. (equal), and P.C. (equal); Funding Acquisition: J.M.H.O. (equal), P.C. (equal), and U.R. (equal).

Funding

V.B. and P.C. acknowledge funding received within the European Union’s Horizon 2020 program from project HPC-LEAP (MSCA-ITN-2014-EJD, Grant ID: 642069). J.M.H.O. acknowledges financial support from the Danish Council for Independent Research (DFR) through the Sapere Aude research career program (Grant ID: DFF-1325-00091 and DFF-1323-00744), the Carlsberg Foundation (Grant ID: CF15-0823), and the Research Council of Norway through its Centres of Excellence scheme (Project ID: 262695). E.I. and P.C.

acknowledge funding received within the European Union’s Horizon 2020 program from the BioExcel Center of Excellence (EINFRA-5-2015, Grant ID: 675728). P.C. also acknowledges the funding by the Deutsche Forschungsgemeinschaft via FOR 2518 DynIon project P6. U.R. acknowledges funding from the Swiss National Science Foundation via the NCCR MUST and individual grants. Computational resources were provided by Jülich Supercomputing Center and RWTH Aachen University.

Acknowledgement

The authors thank IBM Zürich Research Laboratory and especially Teodoro Laino, Valery Weber, and Alessandro Curioni for their help with the development of the interface in CPMD. The authors also thank Mark James Abraham, Berk Hess, and Erik Lindahl for valuable advice concerning the GROMACS interface. The authors are grateful to Maria Gabriella Chiarello for providing the preequilibrated structure of ClC-ec1.

References

- (1) Olsen, J. M. H.; Bolnykh, V.; Meloni, S.; Ippoliti, E.; Bircher, M. P.; Carloni, P.; Rothlisberger, U. MiMiC: A Novel Framework for Multiscale Modeling in Computational Chemistry. *J. Chem. Theory Comput.* **2019**, DOI: 10.1021/acs.jctc.9b00093.
- (2) Hutter, J.; Alavi, A.; Deutsch, T.; Bernasconi, M.; Goedecker, S.; Marx, D.; Tuckerman, M.; Parrinello, M. CPMD Copyright IBM Corp 1990-2019, MPI für Festkörperforschung Stuttgart 1997-2012. <http://www.cpmid.org/>.
- (3) Hess, B.; Kutzner, C.; van der Spoel, D.; Lindahl, E. GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. *J. Chem. Theory Comput.* **2008**, *4*, 435–447, DOI: 10.1021/ct700301q.

- (4) Laio, A.; VandeVondele, J.; Rothlisberger, U. A Hamiltonian electrostatic coupling scheme for hybrid Car–Parrinello molecular dynamics simulations. *J. Chem. Phys.* **2002**, *116*, 6941–6947, DOI: 10.1063/1.1462041.
- (5) Andersen, H. C. Rattle: A velocity version of the shake algorithm for molecular dynamics calculations. *J. Comput. Phys.* **1983**, *52*, 24–34, DOI: 10.1016/0021-9991(83)90014-1.
- (6) Weinbach, Y.; Elber, R. Revisiting and parallelizing SHAKE. *J. Comput. Phys.* **2005**, *209*, 193 – 206, DOI: <https://doi.org/10.1016/j.jcp.2005.03.015>.
- (7) Wiggs, J.; Jansson, H. A hybrid decomposition parallel implementation of the Car-Parrinello method. *Comput. Phys. Commun.* **1995**, *87*, 319 – 340, DOI: 10.1016/0010-4655(94)00153-S.
- (8) Hutter, J.; Curioni, A. Car–Parrinello molecular dynamics on massively parallel computers. *ChemPhysChem* **2005**, *6*, 1788–1793, DOI: 10.1002/cphc.200500059.
- (9) Weber, V.; Bekas, C.; Laino, T.; Curioni, A.; Bertsch, A.; Futral, S. Shedding Light on Lithium/Air Batteries Using Millions of Threads on the BG/Q Supercomputer. 2014 IEEE 28th International Parallel and Distributed Processing Symposium. 2014; DOI: 10.1109/ipdps.2014.81.
- (10) Marx, D.; Hutter, J. *Ab initio molecular dynamics: basic theory and advanced methods*; Cambridge University Press, 2009.
- (11) Hutter, J.; Curioni, A. Dual-level parallelism for ab initio molecular dynamics: Reaching teraflop performance with the CPMD code. *Parallel Computing* **2005**, *31*, 1–17, DOI: 10.1016/j.parco.2004.12.004.
- (12) Becke, A. D. Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A* **1988**, *38*, 3098–3100, DOI: 10.1103/physreva.38.3098.

- (13) Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B* **1988**, *37*, 785–789, DOI: 10.1103/physrevb.37.785.
- (14) Kim, K.; Jordan, K. D. Comparison of Density Functional and MP2 Calculations on the Water Monomer and Dimer. *J. Phys. Chem.* **1994**, *98*, 10089–10094, DOI: 10.1021/j100091a024.
- (15) Stephens, P. J.; Devlin, F. J.; Chabalowski, C. F.; Frisch, M. J. Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields. *J. Phys. Chem.* **1994**, *98*, 11623–11627, DOI: 10.1021/j100096a001.
- (16) JURECA Compute cluster. http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JURECA/Configuration/Configuration_node.html, visited on 2019-04-19.
- (17) JUWELS Compute cluster. http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUWELS/Configuration/Configuration_node.html, visited on 2019-04-19.
- (18) CLAI-X-2018 Compute cluster. <https://doc.itc.rwth-aachen.de/display/CC/Hardware+of+the+RWTH+Compute+Cluster>, visited on 2019-04-19.
- (19) Bozkurt, E. Reprogramming the B1 Domain of Streptococcal Protein G (GB1) A Joint Theoretical and Experimental Investigation. **2018**, 214, DOI: 10.5075/epfl-thesis-8029.
- (20) Bozkurt, E.; Perez, M. A. S.; Hovius, R.; Browning, N. J.; Rothlisberger, U. Genetic Algorithm Based Design and Experimental Characterization of a Highly Thermostable Metalloprotein. *J. Am. Chem. Soc.* **2018**, *140*, 4517–4521, DOI: 10.1021/jacs.7b10660.

- (21) Dutzler, R.; Campbell, E. B.; Cadene, M.; Chait, B. T.; MacKinnon, R. X-ray structure of a ClC chloride channel at 3.0 Å reveals the molecular basis of anion selectivity. *Nature* **2002**, *415*, 287–294, DOI: 10.1038/415287a.
- (22) Troullier, N.; Martins, J. L. Efficient pseudopotentials for plane-wave calculations. *Phys. Rev. B* **1991**, *43*, 1993–2006, DOI: 10.1103/physrevb.43.1993.
- (23) Bircher, M. P.; Rothlisberger, U. Plane-Wave Implementation and Performance of -la- Carte Coulomb-Attenuated Exchange-Correlation Functionals for Predicting Optical Excitation Energies in Some Notorious Cases. *J. Chem. Theory Comput.* **2018**, *14*, 3184–3195, DOI: 10.1021/acs.jctc.8b00069.
- (24) Konôpka, M.; Turanský, R.; Doltsinis, N. L.; Marx, D.; Štich, I. *High Performance Computing in Science and Engineering '08*; Springer Berlin Heidelberg, pp 95–108, DOI: 10.1007/978-3-540-88303-6_8.

Graphical TOC Entry

