Parallel performance of MRCPP

Jonas Jusélius,[a)] Stig Rune Jensen[b)] and Luca Frediani[b)]

a) IT Department, University of Tromsø, Norway
b) CTCC, Department of Chemistry, University of Tromsø, Norway

The MultiResolution Chemistry Program Package (MRCPP) is a quantum chemistry code based on multiwavelet functions. As for finite element bases the 3D coordinate space is partitioned in cubic cells, but multiwavelet theory provide the tools to make the representations adaptive, based on a strict and controlled accuracy criterion.

Essential to any computational chemistry code is the ability to solve the Poisson (1) and Kohn-Sham (2) equations:

$$\nabla^2 V(\boldsymbol{r}) = -4\pi\rho(\boldsymbol{r}) \tag{1}$$

$$[\nabla^2 + V_\rho(\boldsymbol{r})]\psi_i(\boldsymbol{r}) = \epsilon_i\psi_i(\boldsymbol{r}), \quad \rho(\boldsymbol{r}) = \sum_{i=0}^{N/2} 2||\psi_i||^2 \tag{2}$$

By rewriting the Poisson and Kohn-Sham equations in their integral form using Greens functions (see equations below), we avoid the use of derivatives, which is not suited for finite element bases.

$$V(\boldsymbol{r}) = \int G(\boldsymbol{r}, \boldsymbol{s})\rho(\boldsymbol{s})d\boldsymbol{s} \tag{3}$$

$$\psi_i(\boldsymbol{r}) = \int H^{(\epsilon_i)}(\boldsymbol{r}, \boldsymbol{s})V(\boldsymbol{s})\psi_i(\boldsymbol{s})d\boldsymbol{s} \tag{4}$$

The formally non-local integral operators decay rapidly to negligible values away from the main diagonal, yielding an effectively banded structure where the band size is only dictated by the requested accuracy. This sparse operator structure gives prospects of linear scaling algorithms, and is crucial for efficient parallel implementations.

A shared memory (OpenMP) parallelization of this code has been achieved, but this strategy has some limitations, as the maximum number of shared memory CPUs in todays clusters is limited. Moreover, the relatively large memory requirements demand the use of a distributed memory (MPI) strategy, where different parts of the function is located on different MPI hosts. This will inevitably lead to some communication overhead since the operator will couple terms across MPI domain boundaries, but its banded structure should limit this communication to close neighbors. This ultimately allows for parallel algorithms with good scaling behavior with respect to the number of CPUs.

We present a hybrid MPI/OpenMP implementation where we combine the fast memory access of OpenMP, with the massively distributed parallelization of MPI.