Performance/Scaling of the XChem Code

The XChem code employs the OpenMolcas package to compute the Gaussian integrals needed as input to XChem. The performance of OpenMolcas is highly dependent on the disk media used for storage. By employing the SLURM job manager, one obtains excellent parallelization between nodes. Since there are several parts of the calculation where the jobs can be sent independently, we employ the SLURM manager to initiate several jobs simultaneously. Prior to the use of XChem, it is necessary to perform a quantum chemical calculation using OpenMolcas to obtain the molecular orbitals needed for the target wavefunctions. These are then augmented with a large monocentric basis of Gaussian plus B-spline functions that are needed to describe the continuum.

In this way, the XChem code separates into modules with different performance:

- basis: Prepares all 1 electron integrals. This step is not parallelized.
- orbitals: Creates a set of linearly independent orbitals, where the target and monocentric orbitals are orthonormalized. This step is parallelized using OpenMP.
- integrals: This module is one of the most computationally demanding, but unfortunately it is barely parallelized at the OpenMP level. However, integral calculations can be easily parallelized by sending them to different nodes using SLURM. For a standard calculation of 22 monocentric Gaussian exponents, the module can be parallelized using up to 253 nodes.
- rdm: Calculates the Reduced Density Matrices of the different electronic states and can be efficiently parallelized using the SLURM manager. Moreover, this calculation is independent of the actual integrals and can be done in parallel with the integral evaluation.
- qci: Uses the RDM and the integrals to evaluate all matrix elements. It is parallelized using OpenMP. The use of different modules for RDM and the integrals allows the user to independently compute and use both variables with a minimum of recomputation.
- bsplines: Creates the B-spline basis. It is parallelized using OpenMP and the MKL library.

- ccm: Creates the Close-Coupling Matrix for the different channels and is parallelized using the SLURM job manager.
- boxstates: This is parallelized using OpenMP and the MKL library or using SCALAPACK [1].
- scattering states and dta: Evaluates the scattering and the photoelectron spectrum.

Depending on the system, the most demanding modules can be the integrals (i.e., when large angular momenta are required), rdm (if a big active space is employed) and ccm/boxstates (depending of the number of channels in the continuum). All these codes can be used in a highly parallelized manner by using the SLURM manager. As an example, Fig. 1 shows the performance of the code when several nodes (up to 32) are employed. In this case, the calculation was done for pyrazine, a medium-size molecule containing 10 atoms, maximum angular momentum of 4, a CASSCF calculation of 10 electrons in 8 orbitals with 7 channels. Due to the small number of channels and the modest active space, the time consuming step is integral evaluation. As seen in Fig. 1, the performance is quite good.

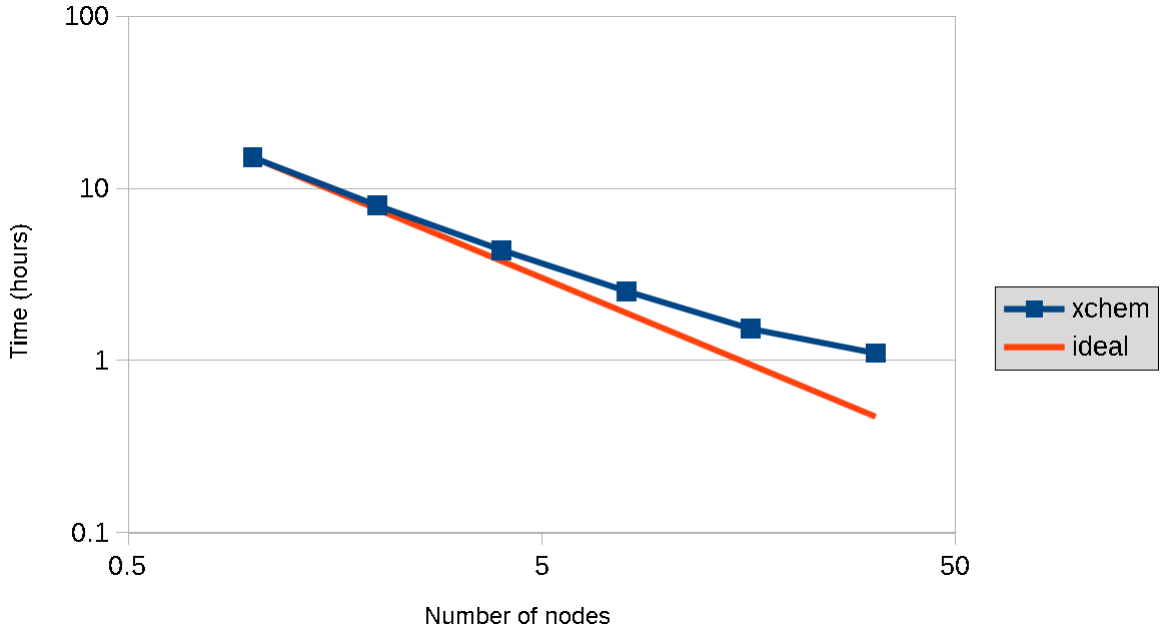

FIG. 1. Performance of the XChem code in pyrazine with different number of nodes