ADF & POP

Alexei Yakovlev <yakovlev@scm.com>
SCM, Amsterdam, The Netherlands

The ADF Modeling Suite

Molecular DFT

Periodic DFT

Approximate DFT (tight-binding)

Semi-empirical NDDO by J.Stewart

Reactive force-field-based molecular dynamics

Thermodynamic properties of liquids and mixtures

SCM
Software for Chemistry & Materials

The ADF Modeling Suite

POP targets

ADF

New HF exchange impl.

1. Performance audit
2. Performance plan

BAND

1. Performance audit
2. Performance plan
3. Proof of concept (in progress)

(not discussed today)

DFTB

Density matrix purification

1. Performance audit
2. Performance plan
3. Proof of concept

Performed by Sally Bridgwater, Nick Dingle, and Jonathan Boyle from NAG

SCM
Software for Chemistry & Materials

POP #1 – ADF

New Hartree-Fock exchange implementation

Basic algorithm:

do atom_pair = 1, N**2

    calculate sub-matrix of $K_{i,j}$ corresponding to the atoms of this pair

end do

SCM
Software for Chemistry & Materials

POP #1 – ADF

New Hartree-Fock exchange implementation

Implementation:

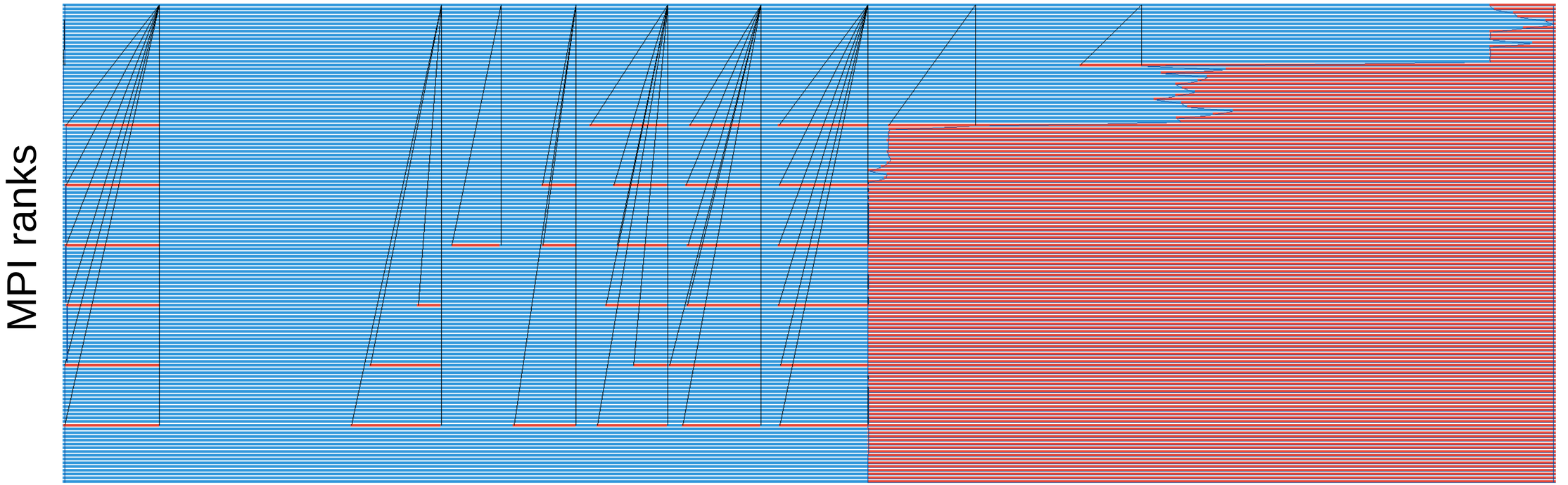
```
class(GlobalIteratorType), pointer :: iterator
iterator => MakeSuitableIterator()
do while (iterator%Next()) !Uses MPI + POSIX shmem to distribute work
      atom_pair = iterator%getIndex()
     calculate sub-matrix of K_{i,j} corresponding to the atoms of this pair
end do
```

SCM
Software for Chemistry & Materials

POP #1 – ADF

New Hartree-Fock exchange implementation

HybridGlobalIterator: MPI between nodes + shmem inside node. 128 processes, 45 atoms



blue – application, red – MPI, total time: 4.24s

SCM
Software for Chemistry & Materials

POP #1 – ADF

New Hartree-Fock exchange implementation

Main results:

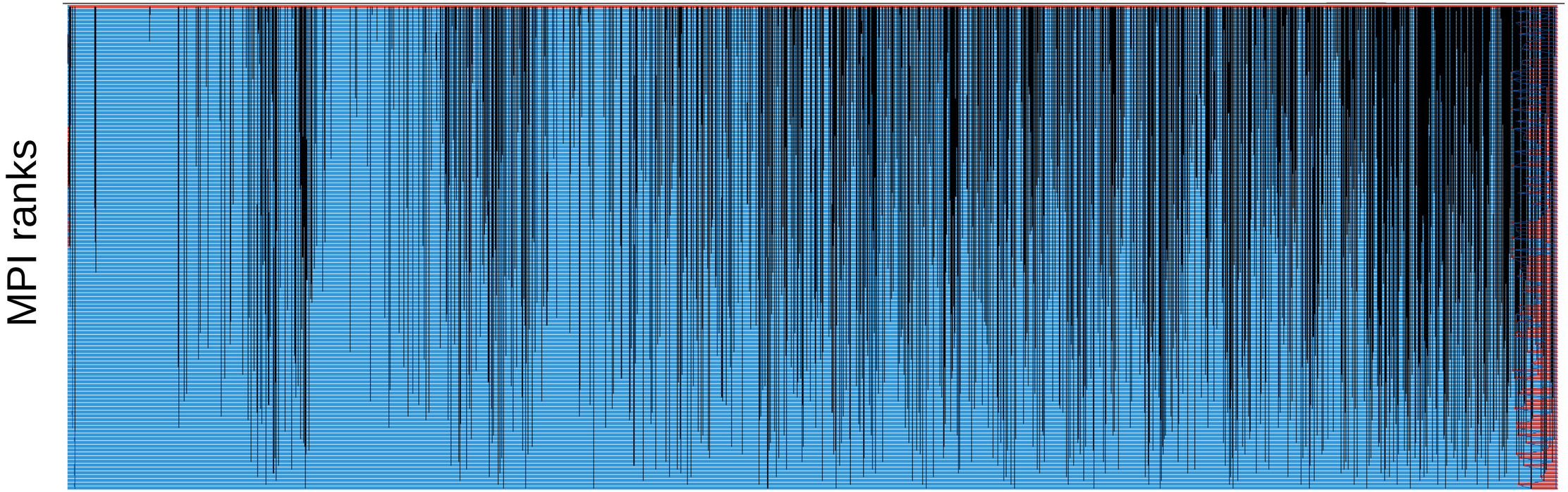
- The Load Balance Efficiency is low, this is due to unequal distribution of work.
- The Computational Scalability is low but this was found to be an artefact of the time cores spend idle waiting to be distributed work.
- The Communication Efficiency is generally good
- **Recommendation**: improve the load balancing algorithm

SCM
Software for Chemistry & Materials

POP #1 – ADF

New Hartree-Fock exchange implementation

DynLoadBalanceType: a sub-class of GlobalIteratorType with a dedicated dispatcher process



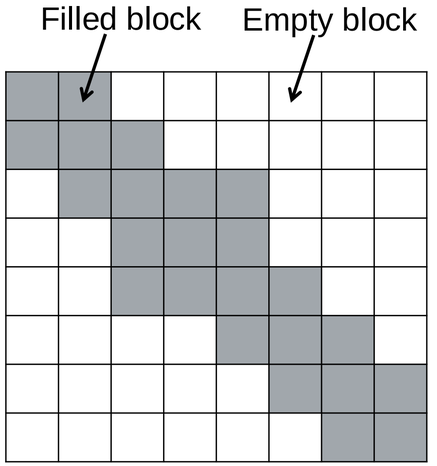
blue – application, red – MPI, total time: 1.992s

SCM
Software for Chemistry & Materials

POP #2 – DFTB

Distributed Block-Sparse Matrix

- $\boldsymbol{FC} = \boldsymbol{SC}\varepsilon$, where $\boldsymbol{F}$ – Fock matrix, $\boldsymbol{S}$ – overlap matrix, $\boldsymbol{C}$ – MO coefficients matrix
- $\boldsymbol{C} \rightarrow \boldsymbol{P}$ the density matrix
- $\boldsymbol{F}$, $\boldsymbol{P}$ and $\boldsymbol{S}$ are sparse for very large systems, $\boldsymbol{C}$ is not
- Block-sparse matrix: dense (up to) 64-by-64 blocks
- Blocks containing zeros only are not allocated
- Blocks are distributed the ScaLAPACK way → easy to convert to/from ScaLAPACK



SCM
Software for Chemistry & Materials

POP #2 – DFTB

Block-Sparse Matrix

Density matrix purification method

- Calculate the density matrix $\boldsymbol{P}$ directly from $\boldsymbol{F}$ and $\boldsymbol{S}$ without computing $\boldsymbol{C}$ by a sequence of matrix-matrix multiplications:

- $\boldsymbol{P_0} = \boldsymbol{S}^{-\frac{1}{2}}\boldsymbol{F}\boldsymbol{S}^{-\frac{1}{2}}$

- $\boldsymbol{P_{i+1}} = \begin{cases} \boldsymbol{P_i}^2, & Tr(\boldsymbol{P_i}) > n \\ \boldsymbol{I} - (\boldsymbol{P_i} - \boldsymbol{I})^2, & Tr(\boldsymbol{P_i}) < n \end{cases}$

SCM
Software for Chemistry & Materials

POP #2 – DFTB

Distributed Block-Sparse Matrix

Block-sparse matrix-matrix multiplication: a variant of the SUMMA algorithm
http://www.netlib.org/lapack/lawnspdf/lawn96.pdf

- Broadcast non-zero blocks in the processor row and column
- Compute local part of the result matrix

SCM
Software for Chemistry & Materials

POP #2 – DFTB

Distributed Block-Sparse Matrix

Main results:

- The DFTB's matrix–matrix multiplication kernel was rewritten to expose the possibility of overlapping communication and computation, but the resulting reduction in runtime was not as large as hoped.
- The overhead of progressing messages in the background during computation outweighed any reduction in runtime from overlapping the two activities.
- Linking the new code with other MPI libraries might lead to a more significant reduction in runtime.

SCM
Software for Chemistry & Materials

Thank you!

Questions?