Private Fock Algorithm

- i,j,k refer to four dimensions over shell indices reflected by the nested loops of the Fock construction.
- Fock matrix replicated across OpenMP threads, other data structures (e.g. Density matrix) shared.
- Fock matrix construction algorithm.
- Master thread of each MPI rank updates the Fock matrix.
- OpenMP parallelization implemented over combined j and k loops.
- No significant difference between static and dynamic Fock load balancing scheduling was observed.
- Significant memory footprint reduction compared to legacy MPI-only code, which depends on MPI/OpenMP computation unit ratio.
- Sharing the Fock matrix is the end goal.

Shared Fock Algorithm

- Fock matrix is shared across threads.
- i,j,k loops are merged into i,j,l loops: MPI parallelization over j. OpenMP parallelization over k (larger MPI iteration space vs. Private Fock).
- ERI contributions are added in a private buffers for i (Fj,Fk,Fil) and j (Fj,Fk,Fil), with Fj updated directly.
- The i buffer is only flushed if the j index is changed after completion of the inner k loop.
- Buffer is 2D array, with outer dimension: data.