ATRIP: A Massively Parallel CCSD(T) implementation for the Exascale Computing Age on CPUs and GPUs

Alejandro Gallo, Andreas Irmler, Andreas Grüneis

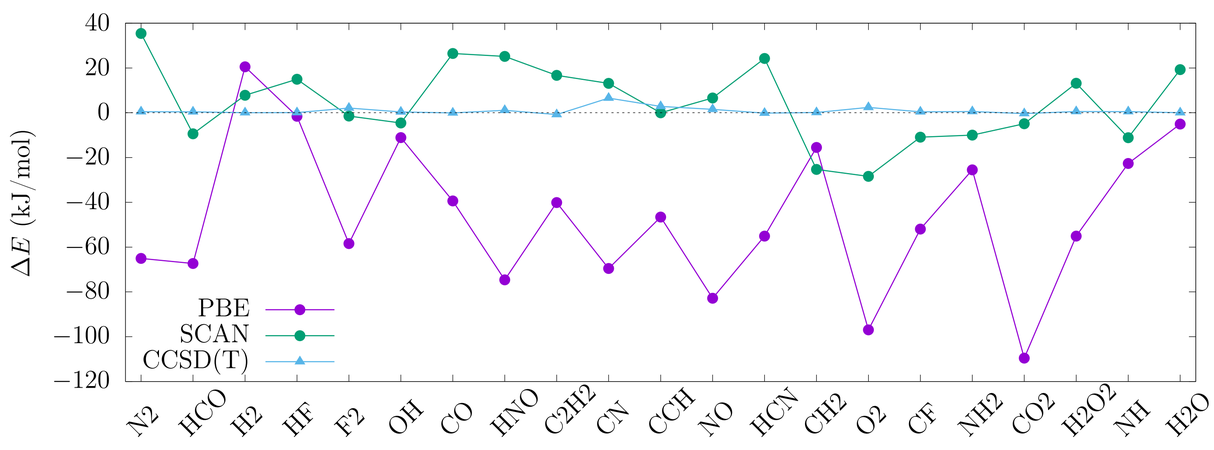
Vienna University of Technology

Abstract

A promising class of approximations widely used in quantum chemistry to solve the many-electron Schrödinger equation is the so-called coupled-cluster (CC) family of methods. These methods achieve for a large class of molecular systems a high accuracy at a comparably moderate cost. Here, we report results of state-of-the-art simulations employing high performance computing resources to produce reliable benchmark results of a molecular absorbtion process on a periodic surface. In this context, we present recent advances implemented in our newly released code coupled-cluster for solids (cc4s), where we apply CC methods to solid state systems employing a plane-wave basis set together with novel basis-set correction schemes. To this end, we employ and contribute to the massively-parallel cyclops tensor framework (ctf). Furthermore, we have developed a high-performance library, ATRIP, which is designed to operate on hundreds of CPU or GPU nodes.

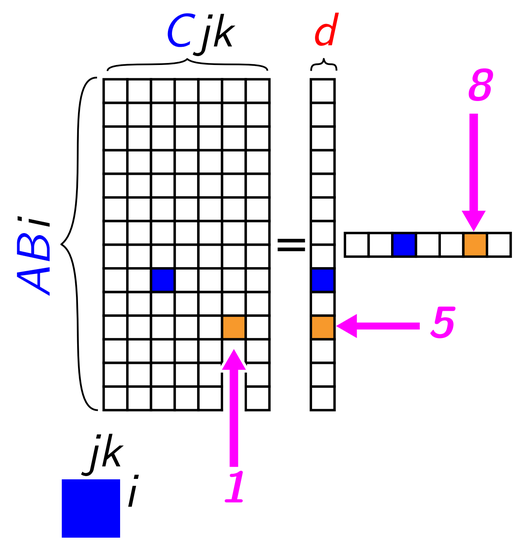
CCSD(T): a hghly accurate electronic structure method

Accuracy of CCSD(T) compared to standard DFT functionals for a small class of molecules. Shown is the error of the atomization energy for 21 molecules, when compared to highly accurate experimantal reference data.

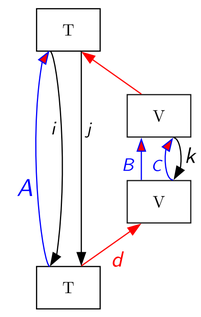


Standard deviation: PBE 56.5 kJ/mol, SCAN 17.5 kJ/mol, CCSD(T) 1.8 kJ/mol [1]

ATRIP: Main algorithm



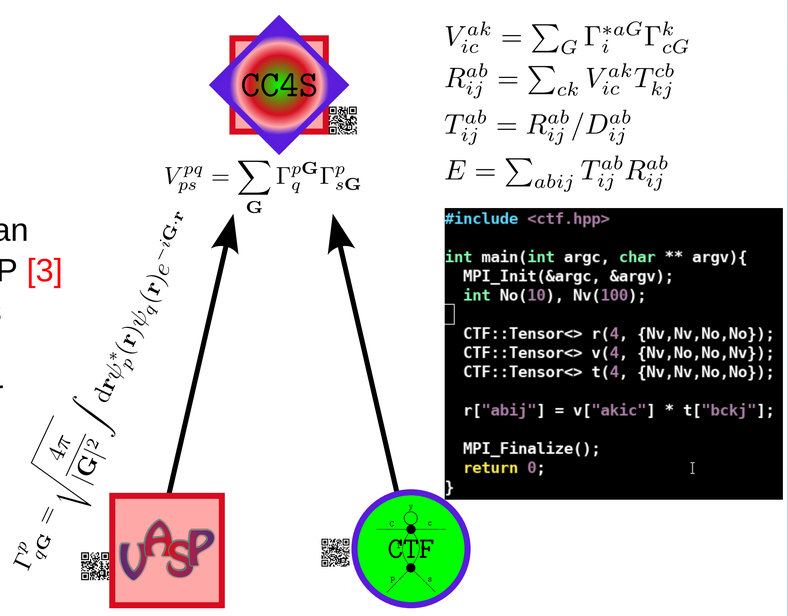

```
for (A,B,C) in {...(A,B,C),(A'B'C'),...} :
    send V(A'B'...) and T(C'...)
    receive V(AB...) and T(C...)
    R(ABi,Cjk) = Σ_d V(ABi,d)T(d,Cjk)
```



[4]

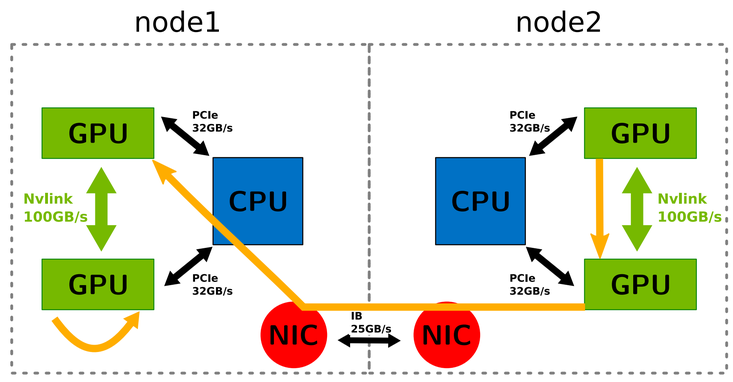
Codes involved

- HF and MP2 calculations performed with VASP
- Reduction of the virtual orbital space using natural orbitals on the level of MP2 [2]
- MP2 CBS estimate employing an efficient implementation in VASP [3]
- wavefunction overlap density is is written to disk
- cc4s solves the coupled-cluster equations
- Tensor contrations are done by using the massively parallel tensor framework CTF

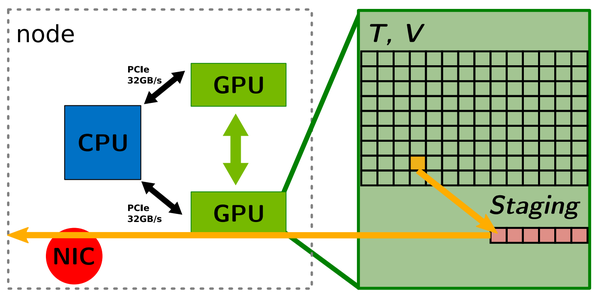



```
#include <ctf.hpp>

int main(int argc, char ** argv){
  MPI_Init(&argc, &argv);
  int No(10), Nv(100);

  CTF::Tensor<> r(4, {Nv,Nv,No,No});
  CTF::Tensor<> v(4, {Nv,No,No,Nv});
  CTF::Tensor<> t(4, {Nv,Nv,No,No});

  r["abij"] = v["akic"] * t["bckj"];

  MPI_Finalize();
  return 0;
}
```

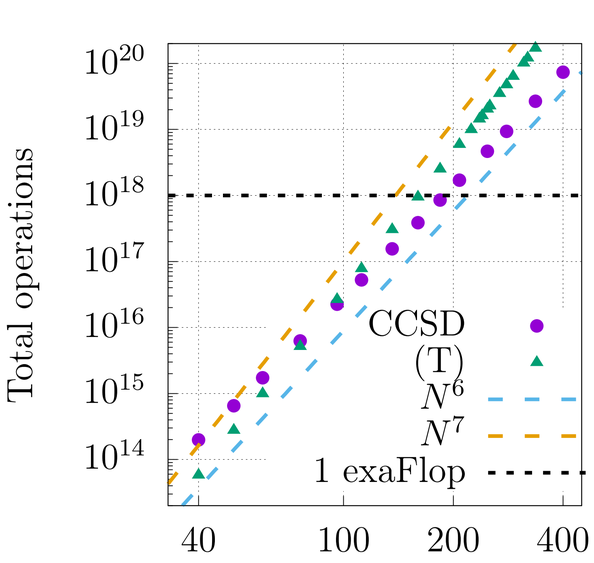
Communication strategies for GPUs

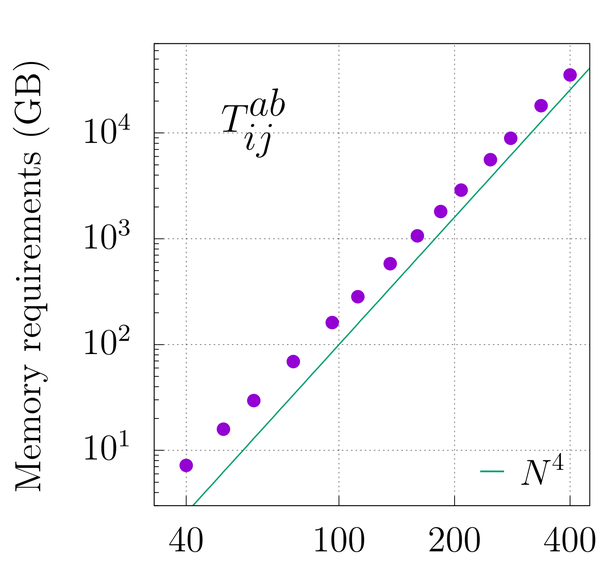


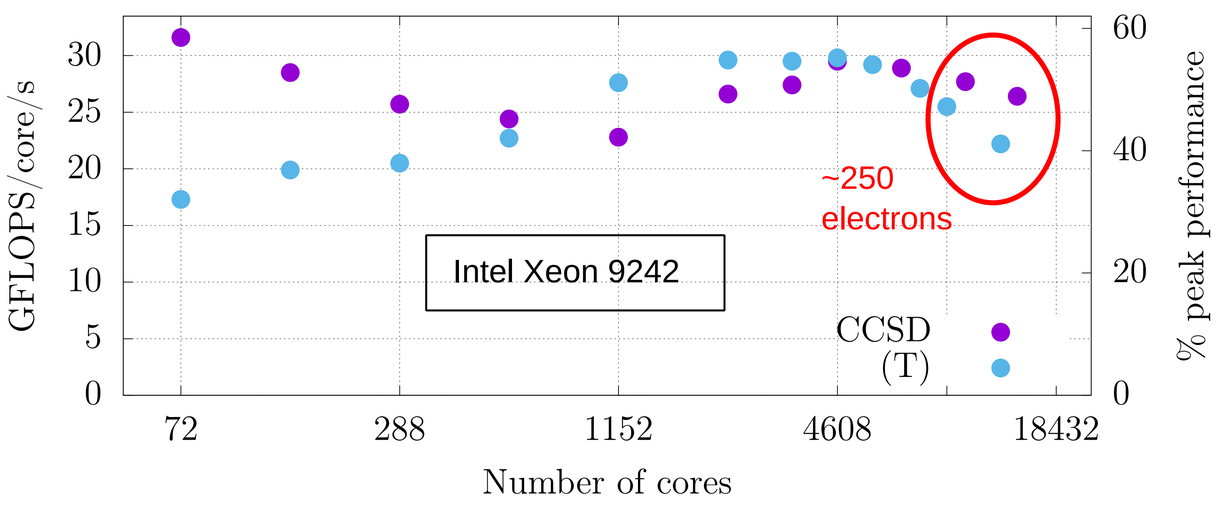
`Atrip` algorithm favors *intra-node* communication.



Computational complexity and performance

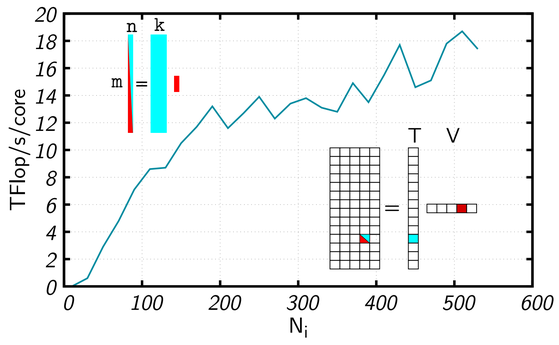






GPU results

- N_i is the number of occupied orbitals (N_{el} / 2)
- Every DGEMM is highly rectangular
- A100 theoretical peak 20 TFlop/s
- It is possible to reach a large chunk of peak with multi-node workloads



ATRIP @A100:
(assistance of A. Brown and S. Mainz)

N_i	Nodes	GPUs	TFlop/s/GPU	% peak
80	2	16	3.7	19
100	6	48	4.24	22
120	12	96	4.82	24
150	25	200	4.63	24

Luna cluster: A100-80GB, one network card per GPU, 8 GPUs per node

Outlook

- Metallic systems with CCSD(cT) [5]
- Block-sparsity to allow calculations with multiple k-points
- Machine-learned force fields from CCSD(T) calculations

References

[1] Tajti, A. *et al.* JCP **121**, 11599 (2004)

[2] Grüneis, A. *et al.* JCTC **7**, 2780 (2011)

[3] Schäfer, T. *et al.* JCP **146**,104101 (2017)

[4] https://github.com/alejandrogallo/atrip

[5] Masios N. *et al.* PRL **131**, 186401 (2023)