Take it to the limit

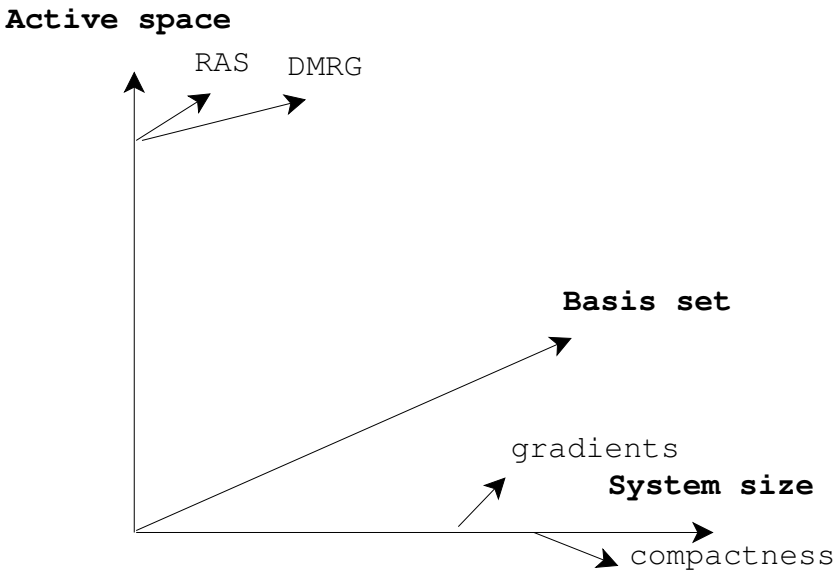
Valera Veryazov
Valera.Veryazov@teokem.lu.se

Department of Theoretical Chemistry
Lund University, Sweden

Overview

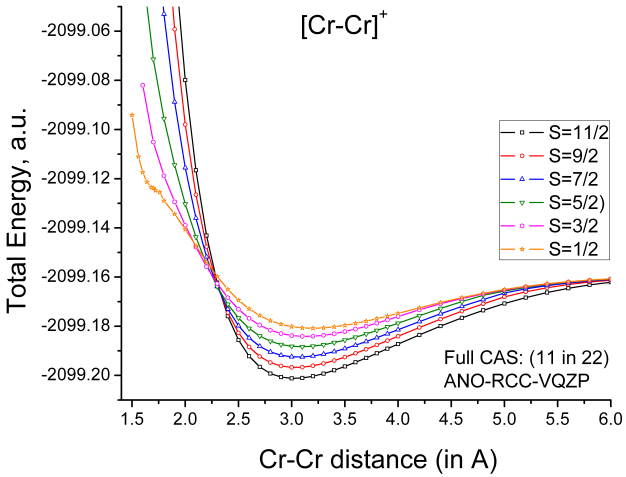
- Some calculations are “big” .. How to run them faster?
- My recent big calculations
- Review of survey about Molcas usage
- New benchmark suite
- Some performance results and recommendations

What is "big"?



Different bottlenecks will be important for different types of calculations.
Let's look at main challenges for large calculations.

- Large basis set: ANO-RCC-VQZP (7s6p4d3f2g1h)
- Large active space: 11 electrons, 22 orbitals (CASSCF)
- CASSCF: Number of CSFs 179,345,082
  1 iteration: 7h, converged solution: more than a week
- CASPT2: 2 days

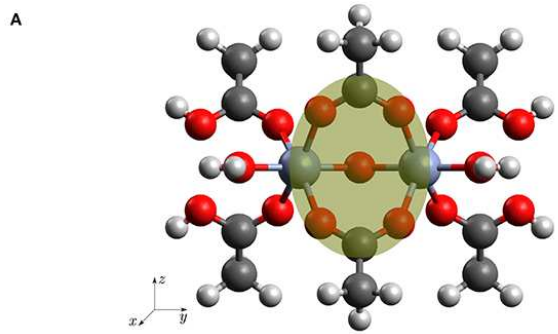


challenges: large memory, long computation time

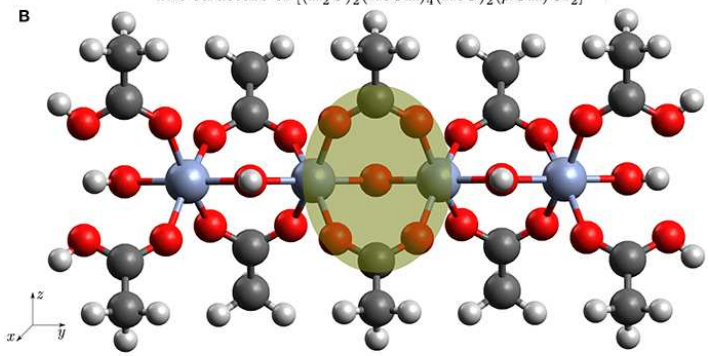
L. Ungur, PÅMalmqist, VV, work in progress

MOF MIL-53(Cr)

- Periodic system, so some claster representation is needed
- 4 (or 2) Cr atoms, so limited CAS
- large amount of atoms



The structure of $[(H_2O)_2(AcOH)_4(AcO)_2(\mu OH)Cr_2]^{3+}$.



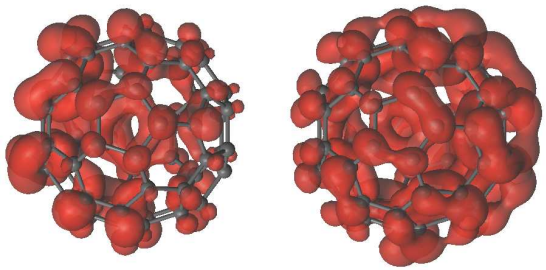
The structure of $[(H_2O)_2(AcOH)_4(AcO)_6(\mu OH)_3Cr_4]^{3+}$.

challenges: large memory, large disk, long computation time

O. Weser, VV, Frontiers in Chemistry 2017

$(C_{60})_2^{-n}$

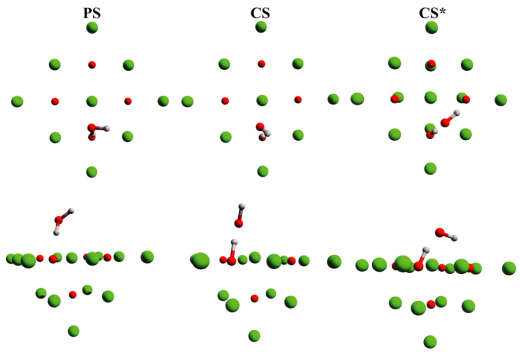
- dimer, as a representation of solid $C_{60}K_n$
- large amount of atoms
- high symmetry, so coarse choice of active space



challenges: large memory, enormous disk, long computation time
F. Naderi, VV, J. Chem Chem Eng 2017; F. Naderi, VV AIP 2017

CaO surface as a catalyst

- cluster representation of surface
- large amount of atoms (compact)
- geometry optimization is impossible by SLAPAF



challenges: large memory, large disk, long computation time

V. Vysotskiy, VV, work in progress

My (biased) view (before I prepared this presentation)

- Optimized BLAS is important (sorry, I will use "MKL" to make it short)
- Parallel MOLCAS should be used with care
- I don't know what is more efficient: use threaded BLAS or parallel Molcas
- Physical memory is essential
- $MOLCAS\_MEM * MOLCAS\_NPROCS$ should be less than RAM

User Survey: technical aspects

56 responses out of about 400.
Installation :

- **45%** - serial with Molcas provided BLAS
- **15%** - serial with optimized BLAS
- **20%** - parallel with Molcas provided BLAS
- **20%** - parallel with optimized BLAS

The majority uses default installation. And this is obviously a bad idea

But what about others???

User Survey: [not] surprising results

- For those who run Molcas in parallel: typical *MOLCAS_NPROCS* = 4 or 8.
  Very few use more than 8.
- Those, who uses *MOLCAS_MEM* $> 8Gb$ usually run MOLCAS in serial
- RAM per workstation in many cases 64Gb or 128Gb
- number of memory per core can be small, but only in a few cases users overloaded *MOLCAS_MEM*

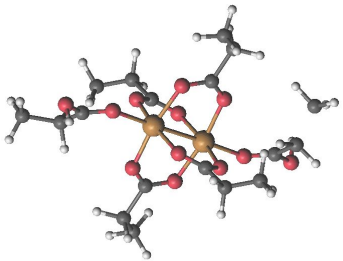
Benchmark suite:2018

- 8 new heavy tests
- Different dimensions of big (slides are coming soon)
- RASSCF and CASPT2 codes
- no benchmarking for SEWARD (it takes a minor fraction)
- Potential candidates for DMRG testing (work in progress)

Testing platforms:

- ODIN - 6 cores Xeon E5-2603, 128Gb RAM, SSD disk
- AURORA - 20 cores Xeon E5-2650, LUNARC node 64Gb RAM, no SSD
- GARM - 24 cores Xeon E5-2650, 32Gb RAM, no SSD

Test 981: large molecule, tiny active space



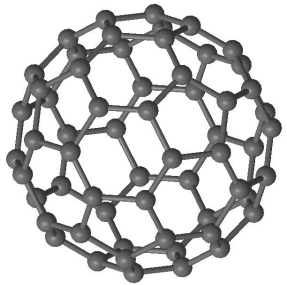
$Cu_2C_{18}O_{12}H_{32}$ - 64 atoms, ANO-RCC-VDZ 388 bf, RICD

Active - 2, CSF - 3
CAS, SS, ITER=35

Memory - 4Gb
Odin: RASSCF= 49', CASPT2=21'

Possible DMRG 4s3d - 22

Test 982: large molecule, small active space



C_{60}^{-2} - 60 atoms, ANO-S-VDZ 540 bf, RICD

Active - 6, CSF - 15
CAS, SS, ITER=22

Memory - 8Gb
Odin: RASSCF= 1h 53', CASPT2=52'

Possible DMRG 8HOMO +3LUMO; 3HOMO+3LUMO

Test 983: tiny molecule, large active space

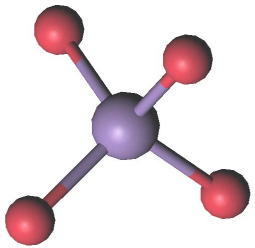
Cr_2 - 2 atoms, ANO-RCC-VQZP* 244 bf, Chol

Active - 12, CSF - 226512
CAS, SS, ITER=64

Memory - 2Gb
Odin: RASSCF= 1h 8’, CASPT2=3’

Possible DMRG 12; +10(22)

Test 984: small molecule, huge active space



MnO_4 - 5 atoms, ANO-RCC-VDZP 90 bf, Exact Integrals

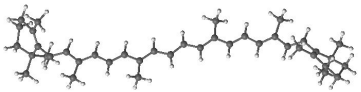
Active - 17, CSF - 2M
CAS, MS, Symmetry, ITER=12

Memory - 2Gb
Odin: RASSCF= 4h 2', CASPT2=28'

Possible DMRG 17; +4dMn=22

Test 985: large molecule, large active space

File:a



$C_{40}H_{56}$ - 96 atoms, ANO-S-VDZP 840 bf, RICD

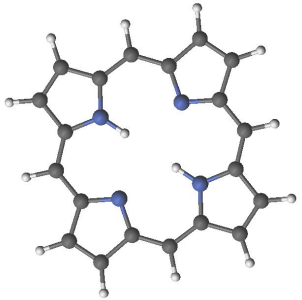
Active - 12, CSF - 113456
CAS, MS, ITER=11

Memory - 8Gb
Odin: RASSCF= 1h 46’, CASPT2=6h 26’

Possible DMRG ??

Test 986: large molecule, huge active space

Files



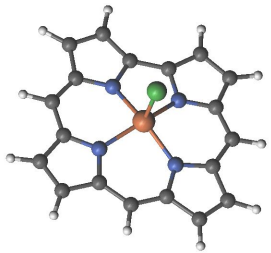
$C_{44}N_{19}H_{15}$ - 38 atoms, ANO-RCC-VDZP 406 bf, RICD

Active - 15, CSF - 2M
CAS, MS, ITER=12

Memory - 4Gb
Odin: RASSCF= 1h 16’, CASPT2=32’

Possible DMRG 15 … till all :)

Test 987



$FeClC_{19}N_4H_{11}$ - 36 atoms, ANO-RCC-VDZP 429 bf, RICD

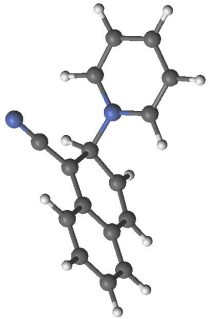
Active - 25, CSF - 1497
RAS, SS, Symmetry, ITER=13

Memory - 4Gb
Odin: RASSCF= 10’, CASPT2=2h 42’

Possible DMRG 25

Test 988: large molecule, huge active space

File a



$C_{16}N_2H_{12}$ - 30 atoms, ANO-S-VDZP 312 bf, Cholesky

Active - 14, CSF - 2M
CAS, MS=4, ITER=10

Memory - 4Gb
Odin: RASSCF= 2h 28’, CASPT2=13h 4’

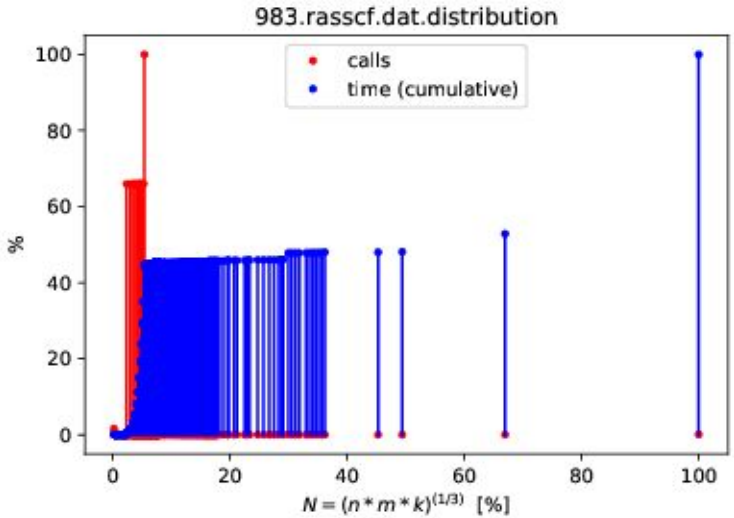
Possible DMRG 14; 23

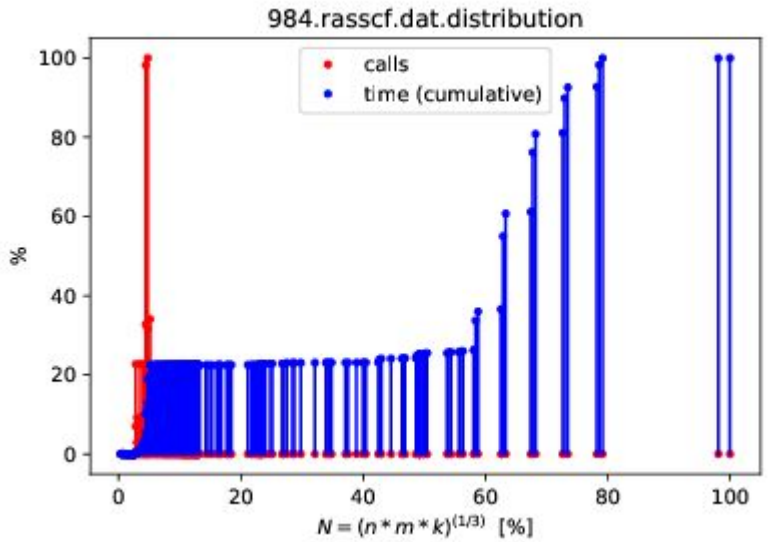
Profiling

1. A bit messy: large tests: long time, large data
2. Callgrind - number of BLAS calls
3. NVIDIA profiling tools for NVBLAS decision
4. RASSCF: Mostly DGEMM
5. CASPT2: DGEMM and eigenvectors solvers
6. Enormous amount of small and middle size calls
7. Presence of huge calls (problematic for GPU due to memory limitations)

Size and time in DGEMM profiling

N of DGEMM calls (test 982): 5 526 982 672, max size 66961





Dilemma:
how to use the machine in the most efficient way?

We have parallelization in MOLCAS and parallelization in BLAS/LAPACK

The latter is done at OMP level, and probably more professional..

And oh yes, 45% - standard installation, but the rest is in equal portions:
serial+MKL, **parallel**, **parallel+MKL**

So, don’t forget to make a pick!

How to read the results. **Theoretical** scaling

Default Serial/NoMKL will be scaled to 64 for simplicity (all tests run more that 1h)

If BLAS is about 50% of the time, and ‘MKL’ is about twice more efficient, so = 75%

	MOLCAS BLAS	MKL
serial	64	48
NPROCS=2	32	
4	16	
8	8	
16	4	

RASSCF

WORST	MOLCAS BLAS	MKL
serial	**64**	**50**
NPROCS=2	**42**	**32**
4	**24**	**18**
8	**18**	**12**
16	**14**	**10**

‘BEST’	MOLCAS BLAS	MKL
serial	**64**	**30**
NPROCS=2	**35**	**16**
4	**18**	**9**
8	**13**	**kaboom**
16	**11**	**kaboom**

CASPT2

WORST	MOLCAS BLAS	MKL
serial	**64**	**42**
NPROCS=2	**47**	**34**
4	**33**	**30**
8	**28**	**31**
16	**46**	

B	MOLCAS BLAS	MKL
serial	**64**	**23**
NPROCS=2	**26**	**15**
4	**17**	**12**
8	**13**	**11**
16	**15**	**12**

Observations/Conclusions

- 1. MKL sequential and MKL threaded vary up to 20% in both directions
- 2. Usage of SSD gives a speed up to 50% (sometimes)
- 3. Several cases fails in RASSCF parallel if MKL is used. [One] bug is localised but it might be more of that kind
- 4. RASSCF scales well up to 16 cores. In parallel usage of MKL allows to improve timing, but just a bit
- 5. CASPT2 scales well up to 4-8. Even in cases where memory is not exhausted
- 6. Perhaps, for huge calculations, a larger scaling is not needed, due to memory requirements
- 7. Use of MKL is fantastic, but only if you have to run Molcas in serial