

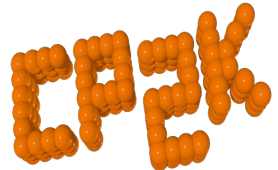
CP2K Developers Meeting

August 18th, 2023 14:00-16:00 CEST



CP2K Developers Meeting

1. Announcements of Meetings
2. Current Development Efforts (all)
3. Current Development Efforts: New Documentation (Ole Schütt)
4. Current Development Efforts: HFX (Robert Schade)
5.
6. Software Support CASUS/HZDR (Frederick Stein)
7.
8. Current Issues when running CP2K (all)
9. Feature Deprecation (all)
10. CP2K Release (all)
11. CP2K-related Events (all)

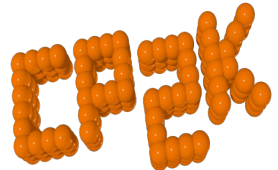


Announcements of meetings

Did Github Discussions work well? How did you learn about the meeting?

Shortcomings:

- everything is public (slide access,...)



Current Development Efforts

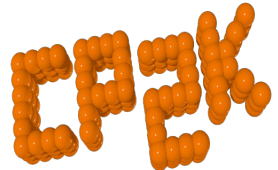
What are you currently working on or planning to work on?

- Jan Wilhelm:
 - improve scaling of low-scaling periodic GW, trade 3-center integrals for lesser memory demands (also other low-scaling approaches could profit from this step)
 - SOC in pseudopotentials now in CP2K
- Augustin Bussy
 - RI-HFX with k-point sampling, now benchmarking for efficiency and accuracy, also works with ADMM
- Dorothea Golze
 - interface to GreenX
 - Help with toolchain part appreciated: Ole
- Bo Thomsen:
 - interface between PIMD and CP2K, working on GPU assignment
<https://ccse.jaea.go.jp/software/PIMD/index.en.html>



Multi-GPU (Ole Schütt)

- Multi-GPU = multiple GPUs per CPU
- Prime example LUMI
- Essentially a new architecture:
 - Data has to remain on GPU.
 - GPU-to-GPU communication is key.
 - More workloads onto GPU (Amdahl's law).
- Bad News: Descent support will take 5+ years at current pace
- Good News: NVIDIA Grace-Hopper returns to 1 GPU per CPU ratio



cuSOLVERMp (Ole Schütt)

- New Eigensolver from Nvidia ([documentation](#)).
- Supports multi-node and multi-GPU.
- Experimental [support](#) shipped with CP2K 2023.2 release.
- Optimization for largish matrices (50k) is ongoing.
- Very slow (and still a bit buggy) for small matrices:
 - unclear how to handle different sized matrices (cusolvermp, scalapack, distribution?)
 - maybe do the same as for ELPA

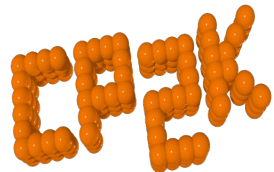


Modernization of Grid Code (Ole Schütt)

- Collocate and integrate already done.
- Let's modernize everything in-between too.
- Goal: Keep grids on the GPU.
- Goal: Support CPUs with 100+ cores.
- Need to apply MPI strategies to OpenMP threads.
- Could leverage new features in FFTW.

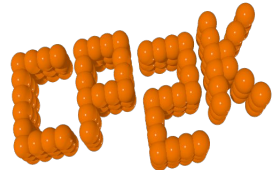
Call to arms!

- Frederick Stein is planning to work on it



Current Development (Thomas Kühne)

- efforts together with Stefanie Gräfe
- use i-pi protocol: forces, calculated with their code
- CP2K for MD propagation
- maybe as an interface to ML-potentials



Current Development Efforts: HFX

1. Restructuring of HFX component:

- a. dynamic load balancing instead of static distribution with support for preferences of MPI ranks for certain quartet classes (currently with master-client system, possibly change to work-stealing)

2. Accelerator implementations

device class	person	primitive cartesian ERIs	primitive spherical ERIs	compression	contracted ERIs	Hamiltonian on device
CPU	libint, libcint, (libintx)	exists		exists	planned to be used	-
CUDA	Marcello Puligheddu, Matt Watkins	done	done		planned	planned
HIP	Lukas Mazur	in progress	in progress	in progress	planned	planned
Intel GPU (OpenMP, OpenCL, SyCL)	anyone interested in trying? maybe SYCLomatic? HIP compiler for Intel					
Intel FPGA	Xin Wu	done	done	done	unclear	unclear
Xilinx FPGA	PC2	in progress		planned	unclear	unclear
Xilinx AI Engines	Johannes Menzel	in progress		planned	unclear	unclear

Current Development Efforts: SOC (Jan Wilhelm)

- benchmarking effort necessary compared to other implementations
-

Software Support CASUS/HZDR (Frederick Stein)

- Since August Professional Support at CASUS/HZDR (Görlitz, Germany)
- Support and Implementation of Features, Events
- Check <https://github.com/users/fstein93/projects/3/views/1> for an overview of possible projects

- Current ideas: Optimization of Grid operations (LibXC, Splines, PW/RS-grids, FFTs, ...)

- Open positions: 1-2 developers (DBCSR/DBM/DBT, k-points/symmetry, ...)

Mail: f.stein@hzdr.de

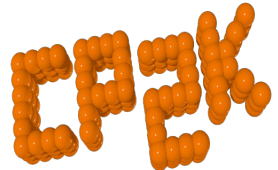
Revamp Documentation (Ole Schütt)

- Move user docs out of the wiki...
- Use git and pull requests instead.
- Use [Markdown and Sphinx](#) for formatting.
- Use [The documentation system](#):
 - **Reference:** Done.
 - **Howtos:** Migrate cp2k.org/howto.
 - **Tutorials:** Migrate cp2k.org/exercises:common.
 - **Explanation:** Write / curate textbook style articles.



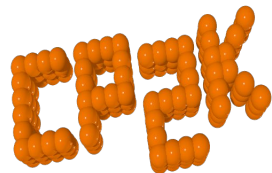
A GUI for CP2K (Ole Schütt)

- Hypothesis: A lot of science is limited by usability.
- Bad News: Usability improvements are not publishable.
- Goal: Make CP2K self-teachable
- Goal: Make CP2K users more productive
- Working title "CP2K-Lab"
- GUI runs in the browser
- Calculations run in the cloud
- Funding: Ideally freemium, realistically charity



Current Issues when Running CP2K

- Matthias Krack: Intel compiler 2021.8.0 and 22.2 and a newer one issue in grid code
 - wait and hope for ifx



Feature Deprecation

Idea:

1. Mark a feature/functionality as to be deprecated. Write large warning message with instruction to notify cp2k-community (e.g. via issue) if feature is still used.
2. Remove feature/functionality some versions later.

Thoughts:

- mention which last version had the deprecated feature
- **keyword importance (expert) or deprecation filtering**

What do you think?

Planned removals:

- old TDDFT code (Hutter)

Suggestions for possible removals:

- maybe old fft code (check wavelet solver and dependencies)
- maybe internal xc-functionals (probably not all, performance considerations, LDA-PADE)
- maybe single-precision on the high-level data structures



CP2K-Release

CP2K-related events:

Ideas:

- Paderborn+CASUS/HZDR:
 - ~Q1/24: Gromacs & CP2K on QM/MM (school with tutorial, 3-4 days in person)
 - ~Q3/24: Post-DFT/HF in CP2K: ADMM, RI, RPA,... (workshop with talks, 1-2 days virtual)