Performant portable HPC applications with SYCL and oneMKL interface

Hugh Bird – Staff Software Engineer

UXL Math SIG

31st January 2023

codeplay

Enabling AI & HPC To Be Open, Safe & Accessible To All

Established 2002 in **Edinburgh, Scotland**.

Grown successfully to around 100 employees.

In 2022, we became a **wholly owned subsidiary** of Intel.

Committed to expanding the **open ecosystem** for heterogeneous computing.

Through our involvement in oneAPI and SYCL governance, we help to **maintain and develop** open standards.

Developing at the forefront of **cutting-edge research**.

Currently involved in two research projects - **SYCLOPS** and **AERO**, both funded by the Horizon Europe Project.

Agenda

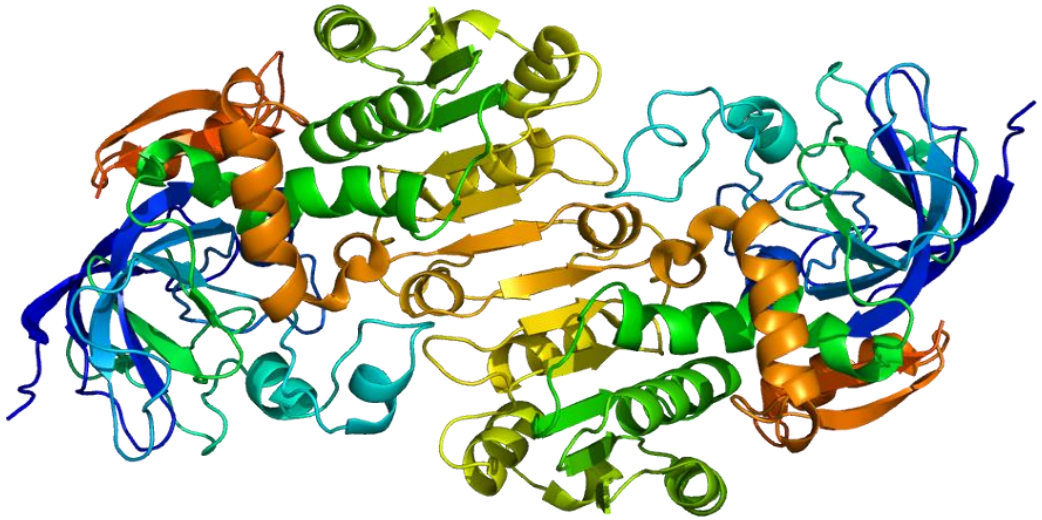
- GROMACS: an example application
- Targeting everything: what could development look like?
- A short introduction to oneMKL
- oneMKL as a backend for GROMACS
- Why should you start developing with SYCL and oneMKL?
- A call to action

What is GROMACS?

Molecular dynamics
- Proteins
- Lipids
- Nucleic acids

Free and open source

Computationally very expensive



ADH5 protein

https://en.wikipedia.org/wiki/Alcohol_dehydrogenase#/media/File:Protein_ADH5_PDB_1m6h.png

GROMACS interactions

Bonded interactions

- Local interaction between particles.
- Can be accelerated with GPU kernels.

Non-bonded interactions

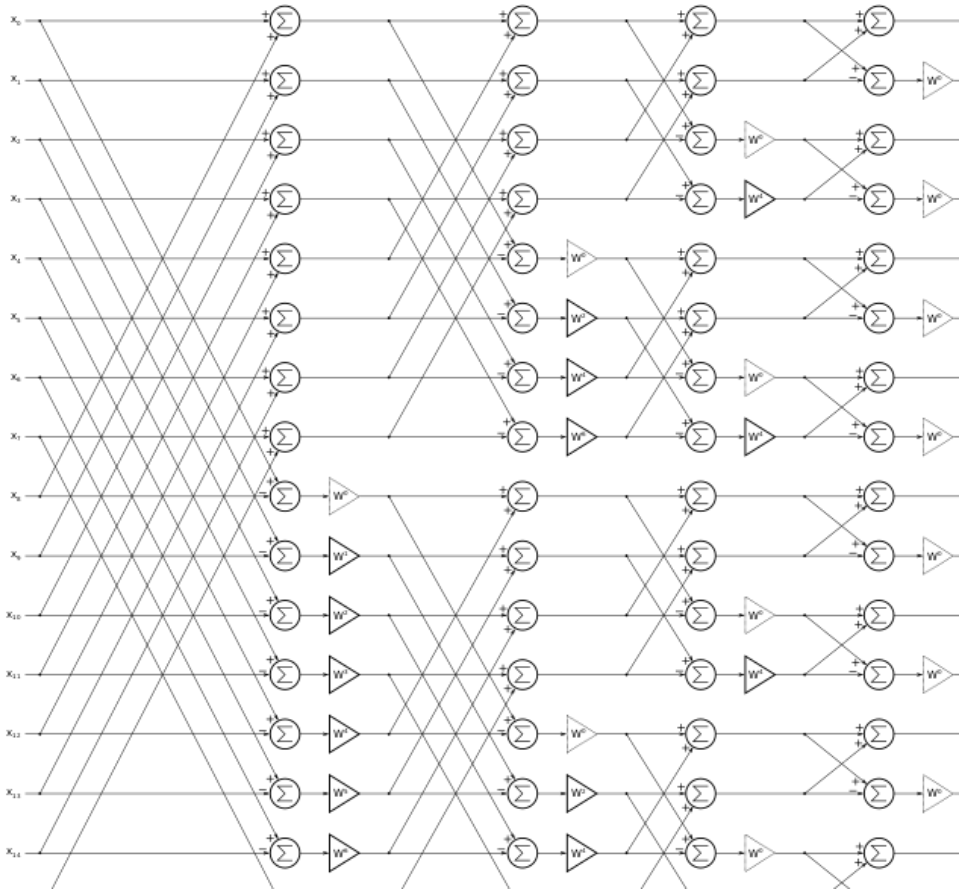
- Interactions can be long-range, with every atom interacting with every other atom
- Convolutions can be FFT accelerated

The Fast Fourier Transform

A fast O(N log N) way of solving the discrete O(N^2) Fourier transform

Can be used for

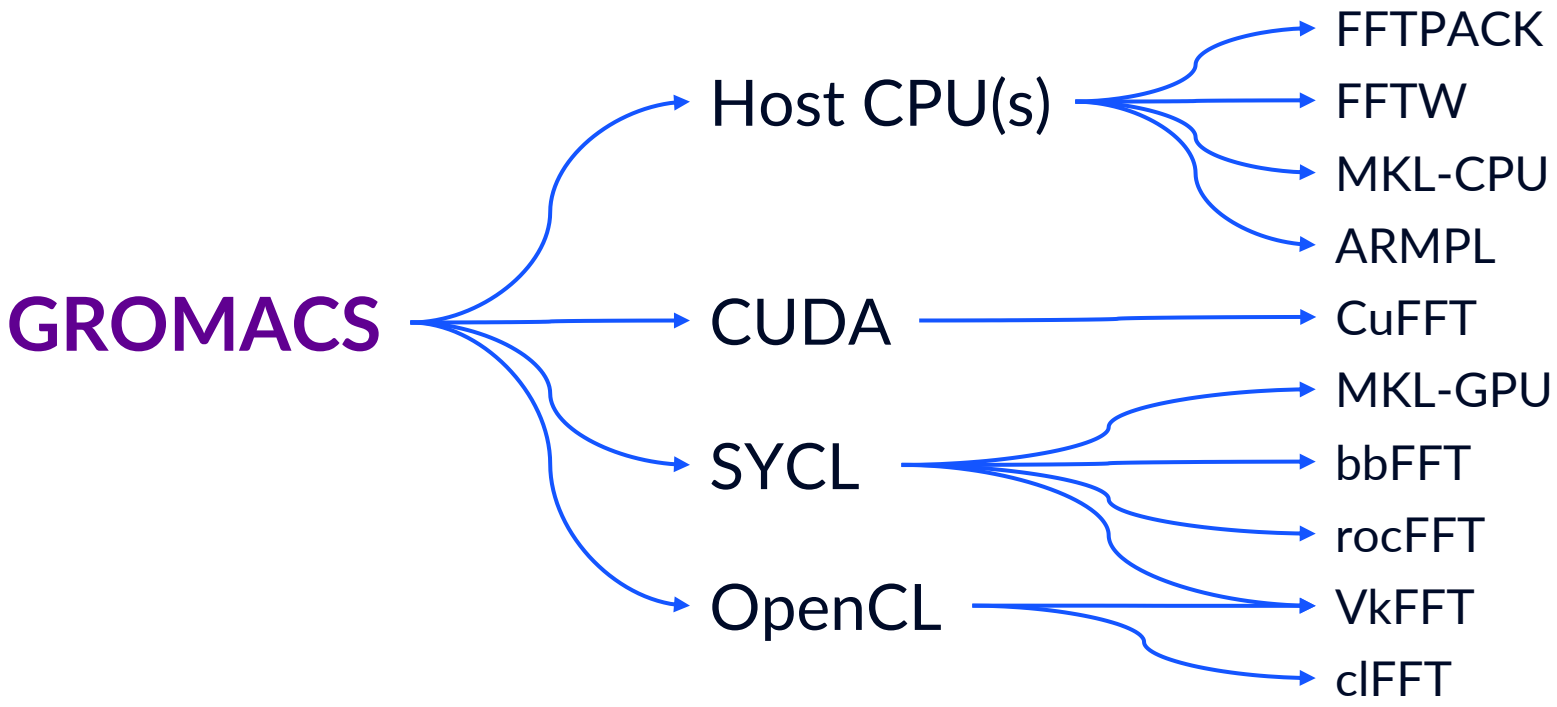
- Signal processing
- Fast multiplication algorithms
- Chebyshev approximations
- Compression
- Fast convolution and cross correlation
  - **Partial differential equations**



Butterfly diagram for a 16-point FFT

https://commons.wikimedia.org/wiki/File:Fft_dif.svg

GROMACS' many DFT backends



… And that's not even every FFT backend!

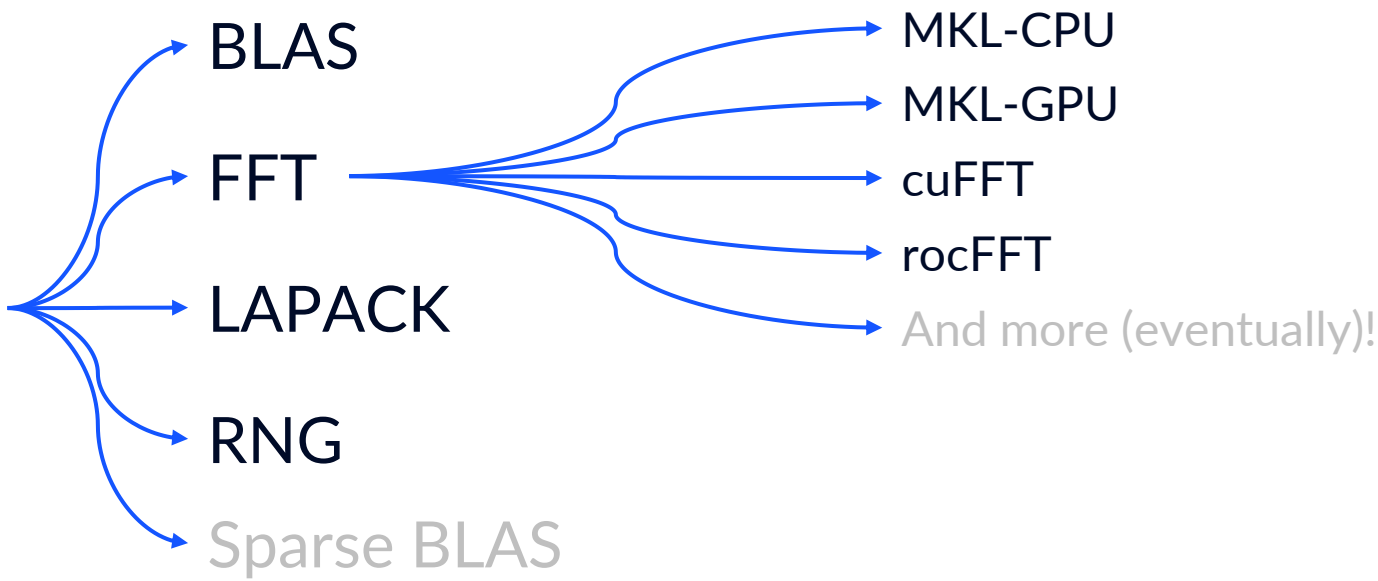
Targeting everything

Runs everywhere with **Intel oneAPI DPC++** and **Codeplay's** plugins for **oneAPI for NVIDIA GPUs** and **oneAPI for AMD GPUs**

And take the vendor's optimized FFT library with you

OneMKL: FFTs

oneMKL Interface Library



The two Math Kernel Libraries

Intel oneAPI Math Kernel Library

- Closed-source Intel proprietary high-performance maths library
- Targets Intel hardware
- CPU and GPU implementations
- I'll call this *MKL-CPU* or *MKL-GPU*

oneAPI Math Kernel Library (oneMKL)

- Open-source interface library
- Targets all hardware
- SYCL interface based on the open oneAPI specification
- I'll call this *oneMKL* or *oneMKL Interface Library*

A lower maintenance alternative

Runs everywhere with **Intel oneAPI DPC++** and **Codeplay's** plugins for **oneAPI for NVIDIA GPUs** and **oneAPI for AMD GPUs**

And take the vendor's optimized FFT library with you

We added oneMKL as an additional backend

- Approximately 500 lines in total
- Copy MKL-GPU implementation:
  - `DFTI_COMPLEX_COMPLEX` to `config_value::COMPLEX_COMPLEX`
- Add backend to CMake configuration

- Tested and working on:
  - NVIDIA A100
  - AMD Radeon Pro W6800
  - Intel ARC A770

We added oneMKL as an additional backend

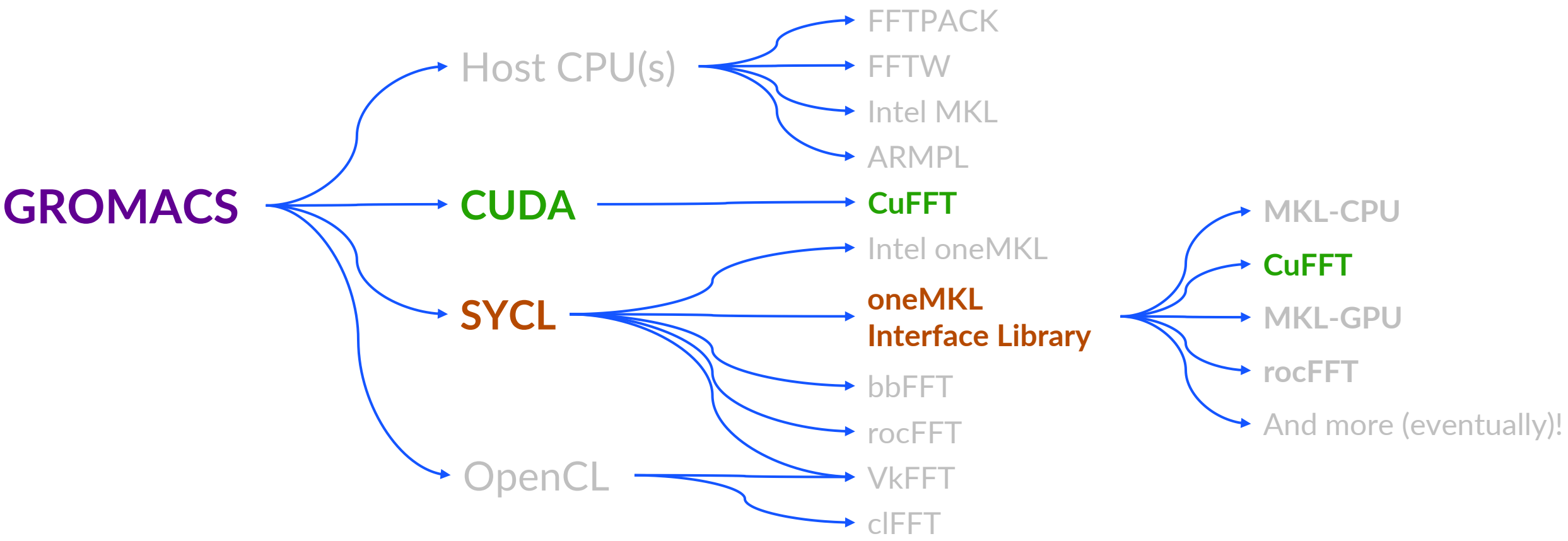
```
// Choose DFT precision, domain and size
using Descriptor = oneapi::mkl::dft::descriptor<oneapi::mkl::dft::precision::SINGLE, oneapi::mkl::dft::domain::REAL>;
Descriptor r2cDescriptor(dftSize) ;

// Set some configurations for how data is formatted
r2cDescriptor.set_value(config_param::INPUT_STRIDES, realGridStrides.data());
r2cDescriptor.set_value(config_param::OUTPUT_STRIDES, complexGridStrides.data());
r2cDescriptor.set_value(config_param::CONJUGATE_EVEN_STORAGE, config_value::COMPLEX_COMPLEX);
r2cDescriptor.set_value(config_param::PLACEMENT, placement);

// Commit the descriptor on a queue – this selects the backend library based on the device associated with the queue.
r2cDescriptor.commit(syclQueue);

// Compute DFTs
oneapi::mkl::dft::compute_forward<Descriptor, float, float>(r2cDescriptor, realGrid, complexGrid);
```

Comparing performance on A100



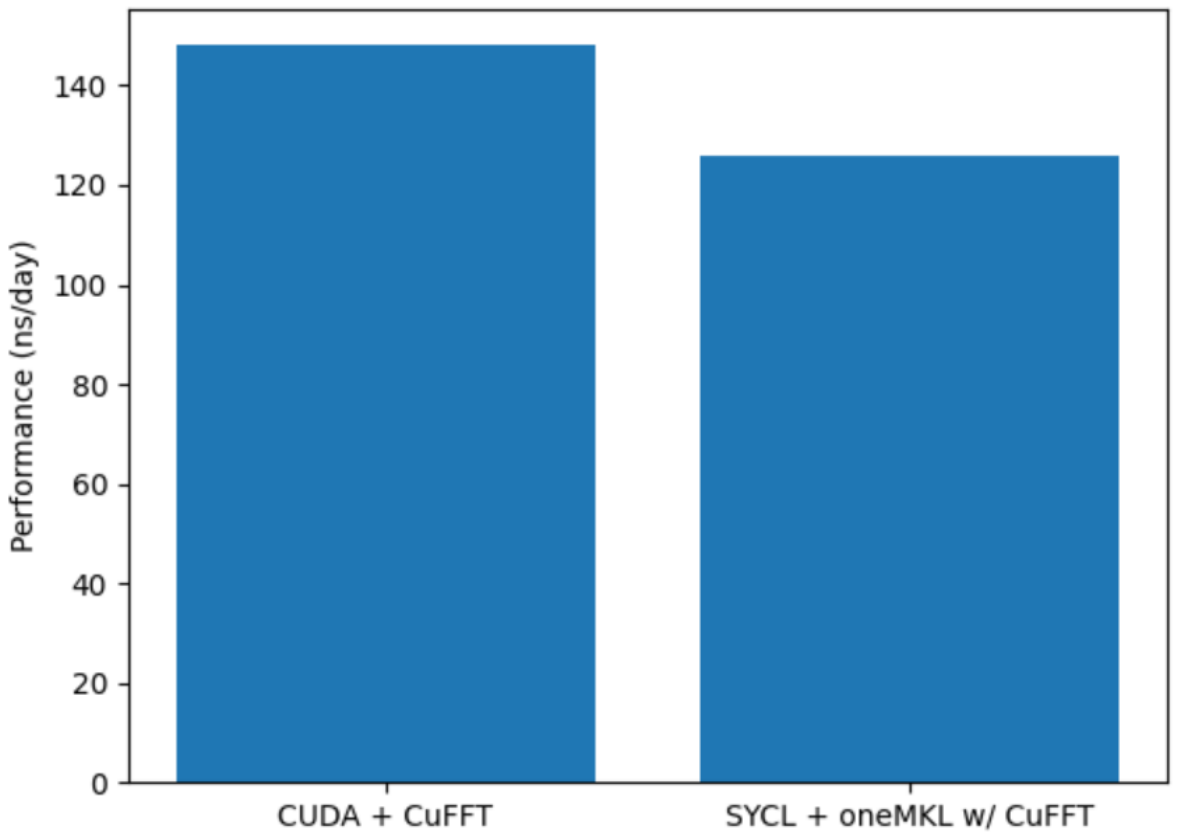
Comparing performance on A100

Common parts:

- Nvidia A100
- The ADH case from GROMACS' benchmark case set, modified to use larger FFTs.

Code paths:

- CUDA implementation with GROMACS CuFFT backend.
- SYCL implementation with oneMKL interface library using CuFFT backend

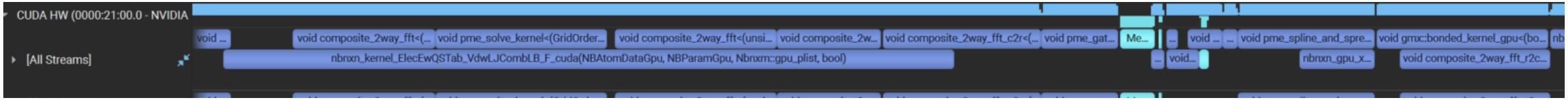


Higher is better

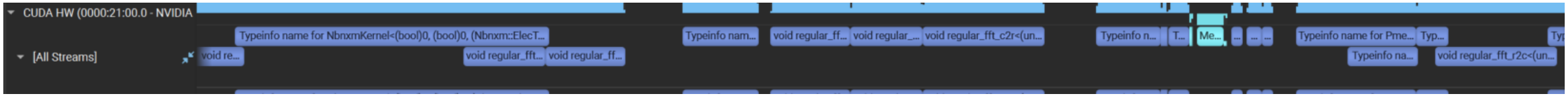
Testing Sep 26 2023 by HJA Bird. System: 2 AMD EPIC 7402 processors with 2 NVIDIA A100 GPUs on Ubuntu 20.04.5 LTS. Build: Modified GROMACS branched from commit df726b872b for oneMKL addition. CUDA version compiled with GROMACS recommended configuration, with GCC 11.3.0 and CUDA 12.2. SYCL version compiled with DPC++ 2023.2.1 with Codeplay oneAPI for NVIDIA GPUs and oneMKL commit 46725c5. Test case: ADH/adh_dodec from https://ftp.gromacs.org/pub/benchmarks/ with fourierspacing set to 0.05. Invocation: gmx mdrun -ntmpi 1 -nb gpu -pme gpu -pmefft gpu -bonded gpu -pin on -v -noconfout -nsteps 10000 -ntomp 24 -s ../topol.tpr -notunepme -resethway -dlb no -update gpu
Your costs and results may vary.

Comparing performance on A100

CUDA + cuFFT



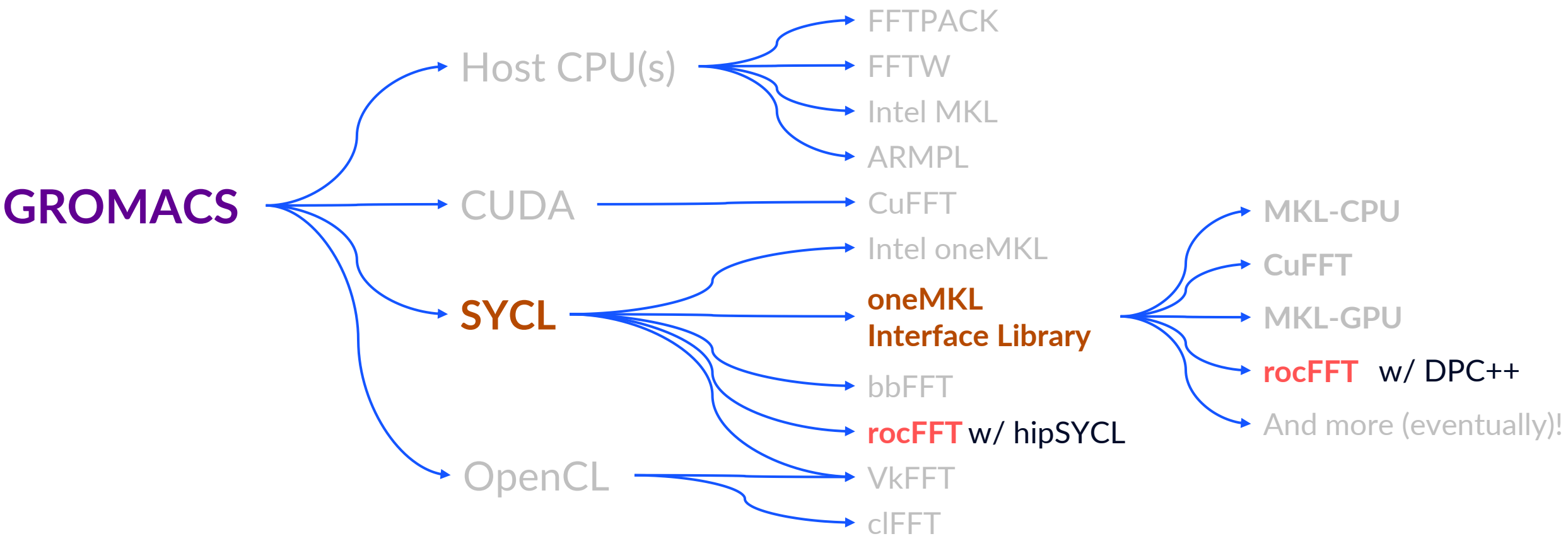
SYCL + oneMKL w/ cuFFT



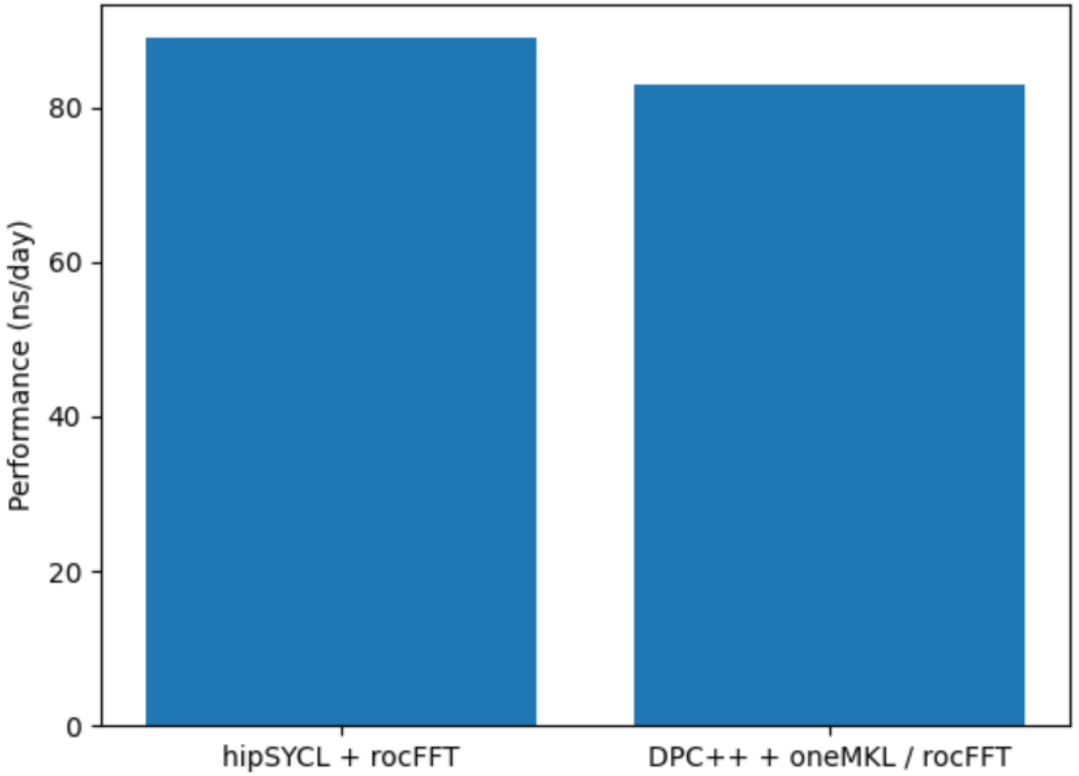
- SYCL + oneMKL w/cuFFT
  - Uses cuFFT kernels
- Performance difference is due to different kernel scheduling between implementations

Testing Sep 26 2023 by HJA Bird. System: 2 AMD EPIC 7402 processors with 2 NVIDIA A100 GPUs on Ubuntu 20.04.5 LTS. Build: Modified GROMACS branched from commit df726b872b for oneMKL addition. CUDA version compiled with GROMACS recommended configuration, with GCC 11.3.0 and CUDA 12.2. SYCL version compiled with DPC++ 2023.2.1 with Codeplay oneAPI for NVIDIA GPUs and oneMKL commit 46725c5. Test case: ADH/adh_dodec from https://ftp.gromacs.org/pub/benchmarks/ with fourierspacing set to 0.05. Invocation: gmx mdrun -ntmpi 1 -nb gpu -pme gpu -pmefft gpu -bonded gpu -pin on -v -noconfout -nsteps 10000 -ntomp 24 -s ../topol.tpr -notunepme -resethway -dlb no -update gpu
Your costs and results may vary.

Comparing performance on W6800



Comparing performance on W6800



Higher is better

Testing Oct 27 2023 by HJA Bird. System: i9-12900k with AMD W6800 on Ubuntu 20.04.3 LTS. Build: Modified GROMACS branched from commit df726b872b for oneMKL addition. hipSYCL version compiled with hipSYCL 0.9.4 and ROCm 5.4.3 with GROMACS recommended configuration and rocFFT 1.0.21. DPC++ version compiled with DPC++ 2023.2.1 with Codeplay oneAPI for AMD GPUs and oneMKL commit 46725c5 and rocFFT 1.0.21. Test case: ADH/adh_dodec from https://ftp.gromacs.org/pub/benchmarks/ with fourierspacing set to 0.05. Invocation: gmx mdrun -ntmpi 1 -nb gpu -pme gpu -pmefft gpu -bonded gpu -pin on -v -noconfout -nsteps 10000 -ntomp 8 -s ../topol.tpr -notunepme -resethway -dlb no -update gpu
Your costs and results may vary.

Start developing with SYCL + oneMKL

- Develop locally with SYCL + oneMKL
  - Test, run and debug on Intel CPU
- Port to GPU on your local machine
  - Test, run and debug on Intel integrated graphics
- Run on whatever HPC resources you can get your hand on!
  - Portable by default
    - Run on Intel, AMD and NVIDIA hardware
  - Performant by default
    - Enabled by oneMKL's vendor optimized backends

A call to action

Use SYCL and oneMKL:

- Less code, fewer APIs, less debugging
- Portable by default
- Performant by default

… And let us know if you have any issues!

https://github.com/oneapi-src/oneMKL

Disclaimers

A wee bit of legal

Performance varies by use, configuration and other factors.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details.

No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

codeplay

Social Media

Don’t forget to follow us for the latest updates!

hugh.bird@codeplay.com


@codeplaysoft


@codeplaysoftware


codeplay-software


codeplay-software