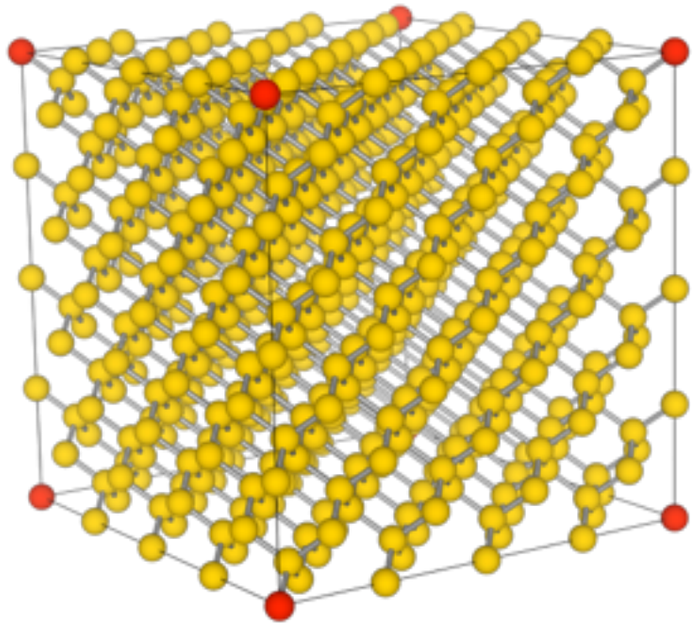
Team SIRIUS

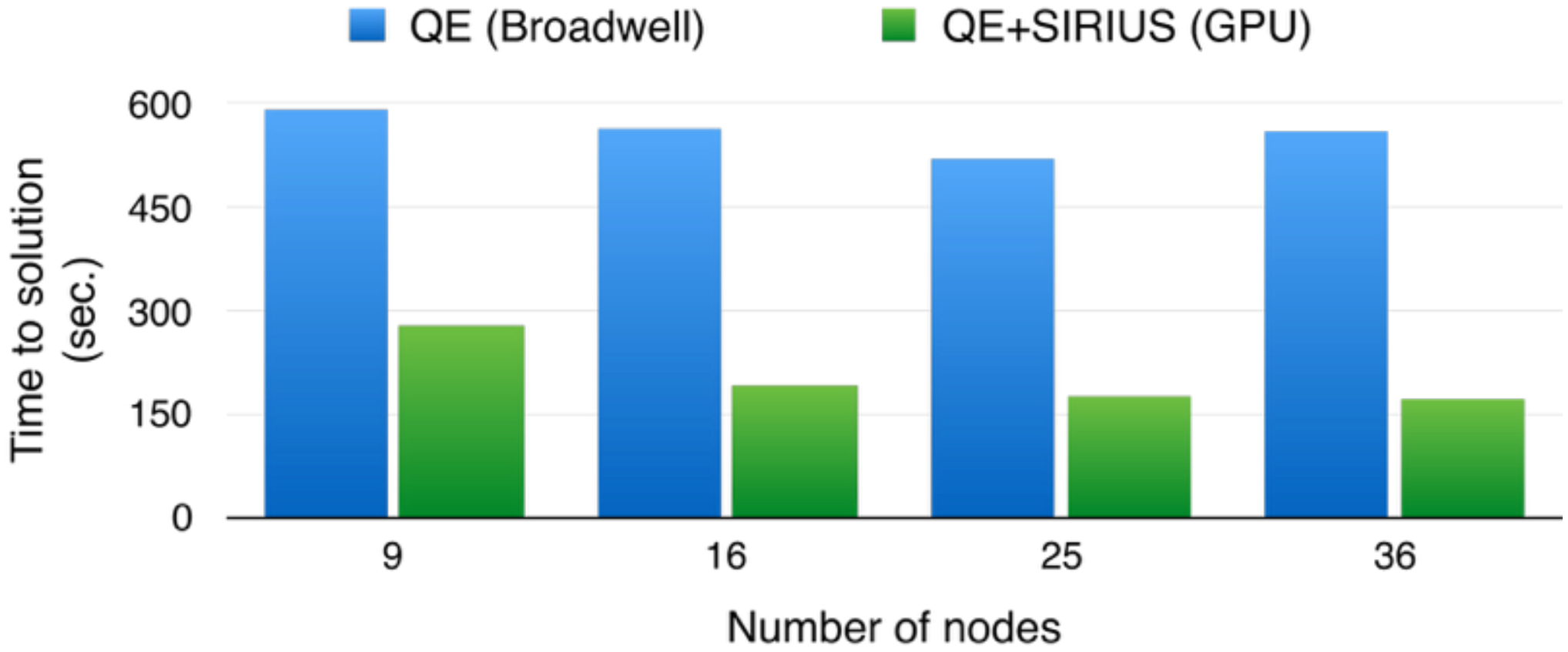
Pietro Bonfà (CINECA)
Marco Borelli (EPFL)
Ilia Sivkov (CSCS)
Mathieu Taillefumie (CSCS)
Anton Kozhevnikov (CSCS)

Problem trying to solve

Ground state of $Si_{511}Ge$ unit cell

- scalability
- performance of GPU backend





Strategy

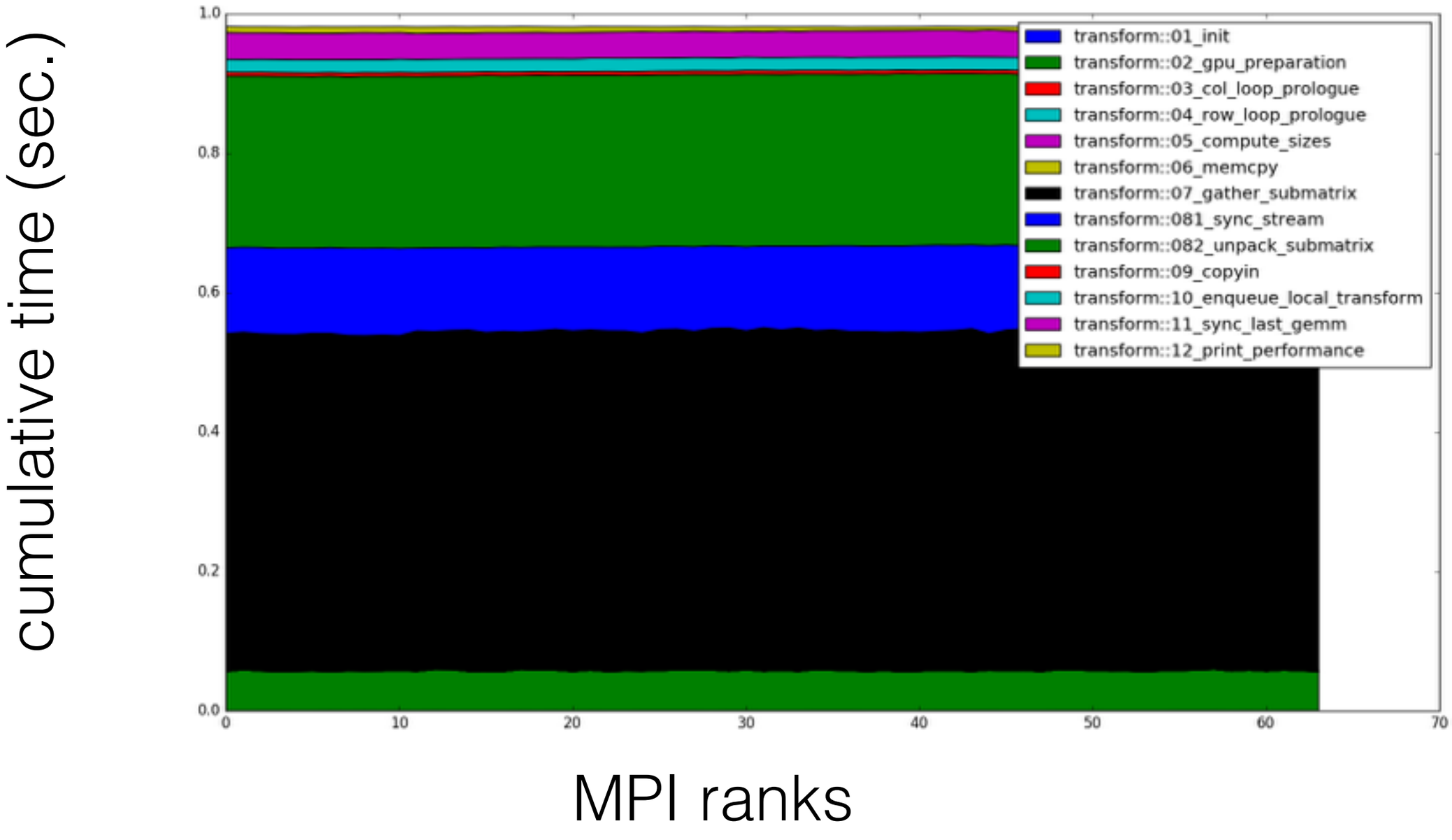
- Profiling
- GPU direct in FFT
- Optimization of two compute-intensive kernels

Profiling

- Perftool-lite based on sampling: works for CPU code
- Perftool-lite based on events: doesn't work! Code can be compiled but the arithmetics is wrong.
- Perftool-lite-gpu: code can be compiled but the code crashes during run
- Score-P: code can not be compiled with all version of the tool. There is interference between the generated code and compilers (gcc or icc)
- nvprof+nvvp: works fine but sometimes is very slow!

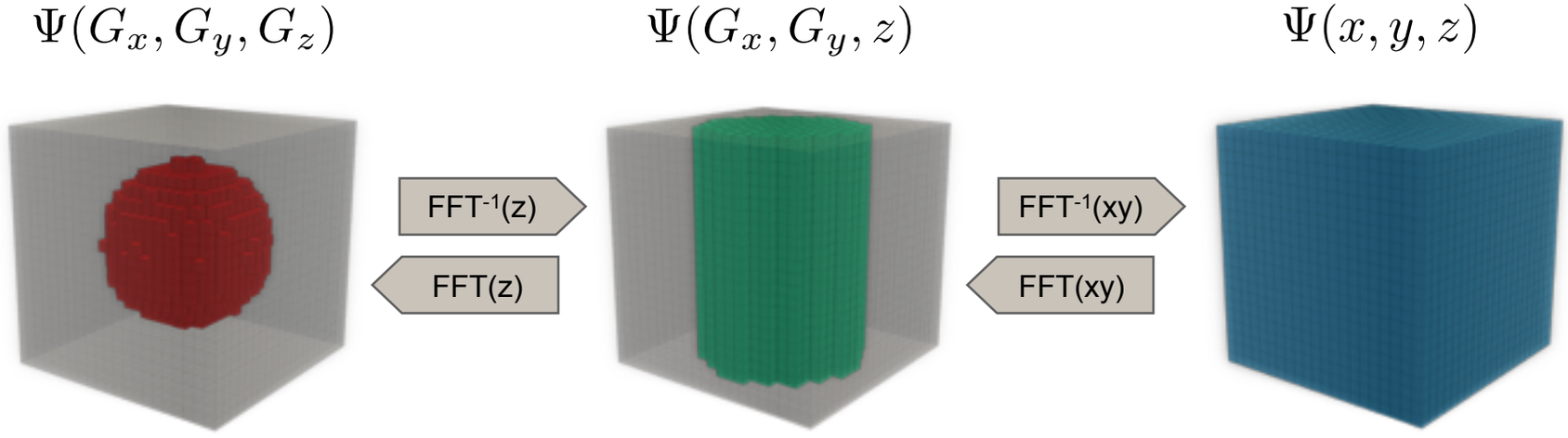
Profiling

Very simple custom time profiling of individual pieces of the code



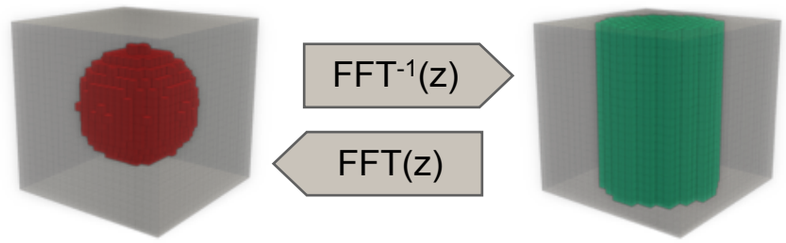
GPU direct in FFT

FFT3D is decomposed in 1D and 2D transforms

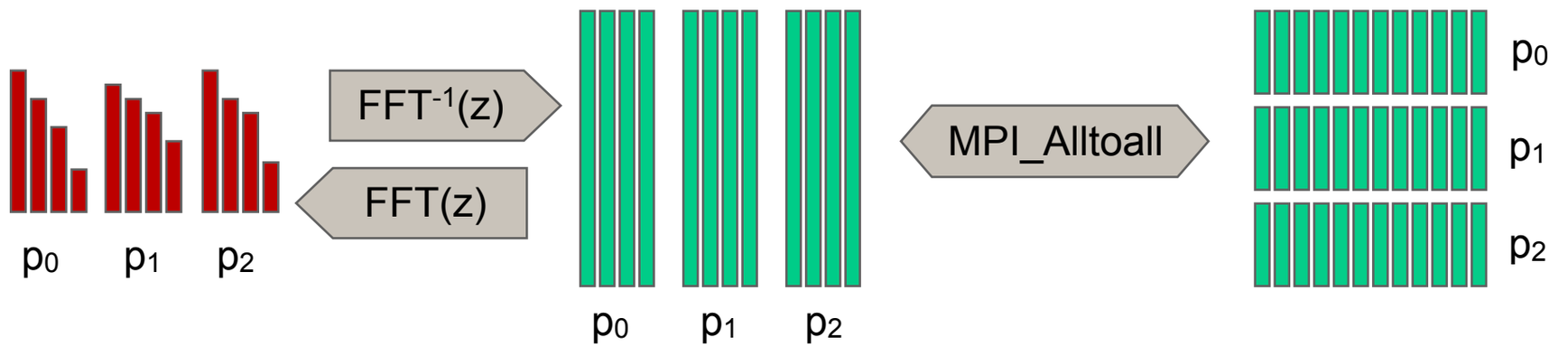


GPU direct in FFT

Parallel transformation of z-sticks



- each rank executes 1D transformations of the local fraction of z-sticks
- z-sticks are swapped between MPI ranks using MPI_Alltoall



{full-z, partial-xy} distribution of a cylinder

{partial-z, full-xy} distribution of a cylinder

GPU direct in FFT

- in/out data is on GPU
- 1D FFT is done with CUDA
- MPI_Alltoall is done with GPU direct

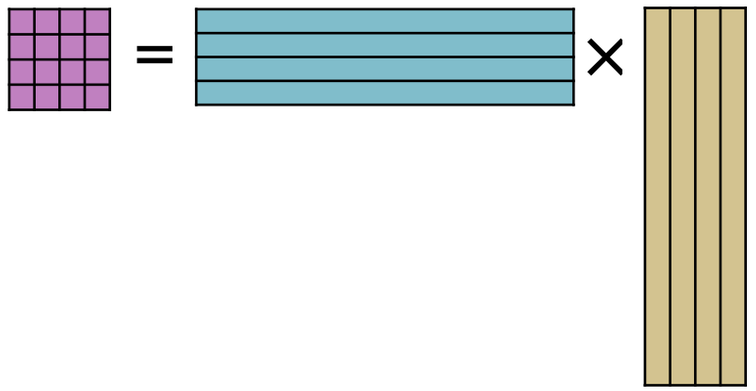
Time for full single FFT

```
CPU run                                  –   1.8   ms
CPU MPI                  + nvprof        –   1.1   ms
CPU MPI     + RDMA       + nvprof        –   21    ms
GPU direct  + RDMA       + nvprof        –   16    ms
GPU direct  + RDMA                       –   1.    ms
GPU direct  + RDMA       + pinning       –   0.75  ms
```

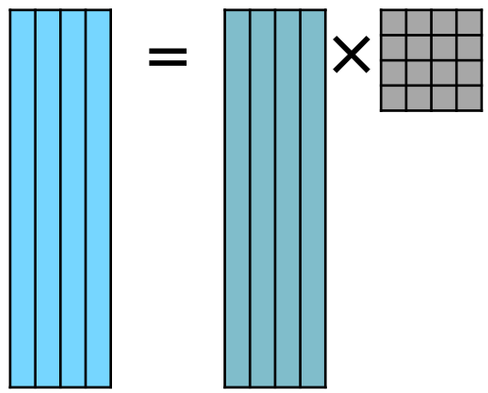
srun -n 8 --ntasks-per-core=2 -c 24 --unbuffered numactl --physcpubind=13-23 ./app

Optimization of compute-intensive kernels

- Subspace Hamiltonian construction

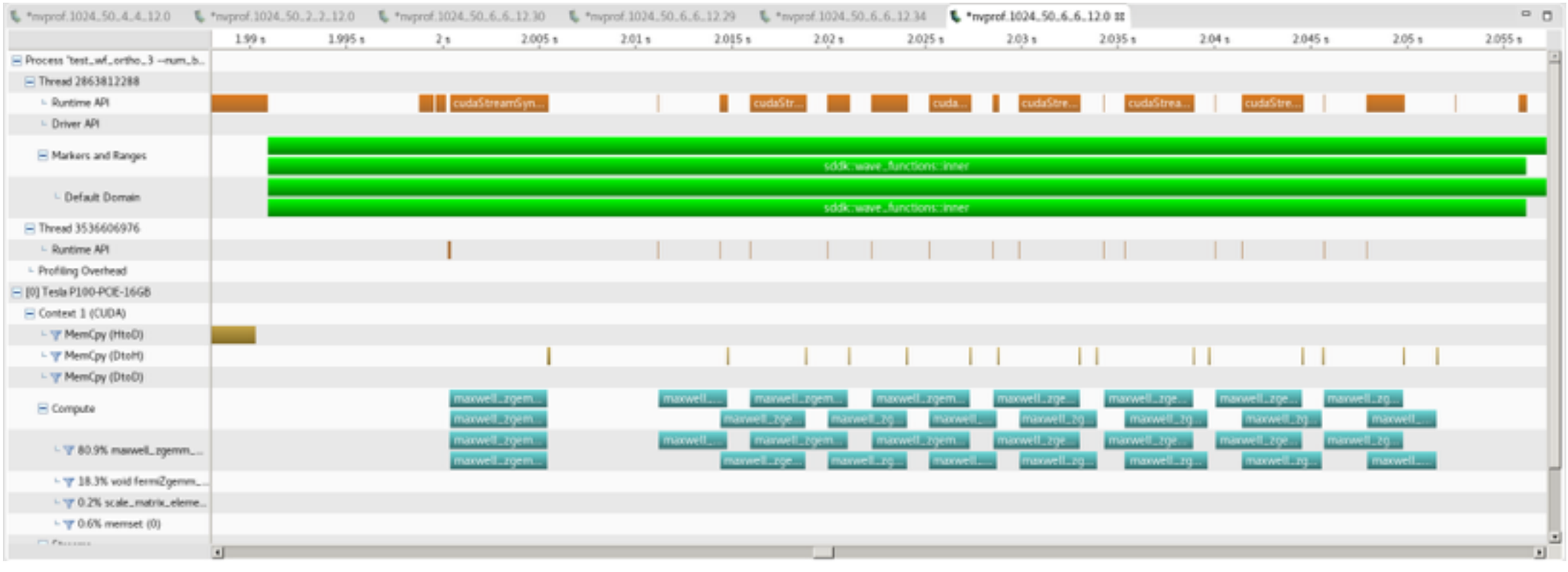


- Wave-functions and residuals update



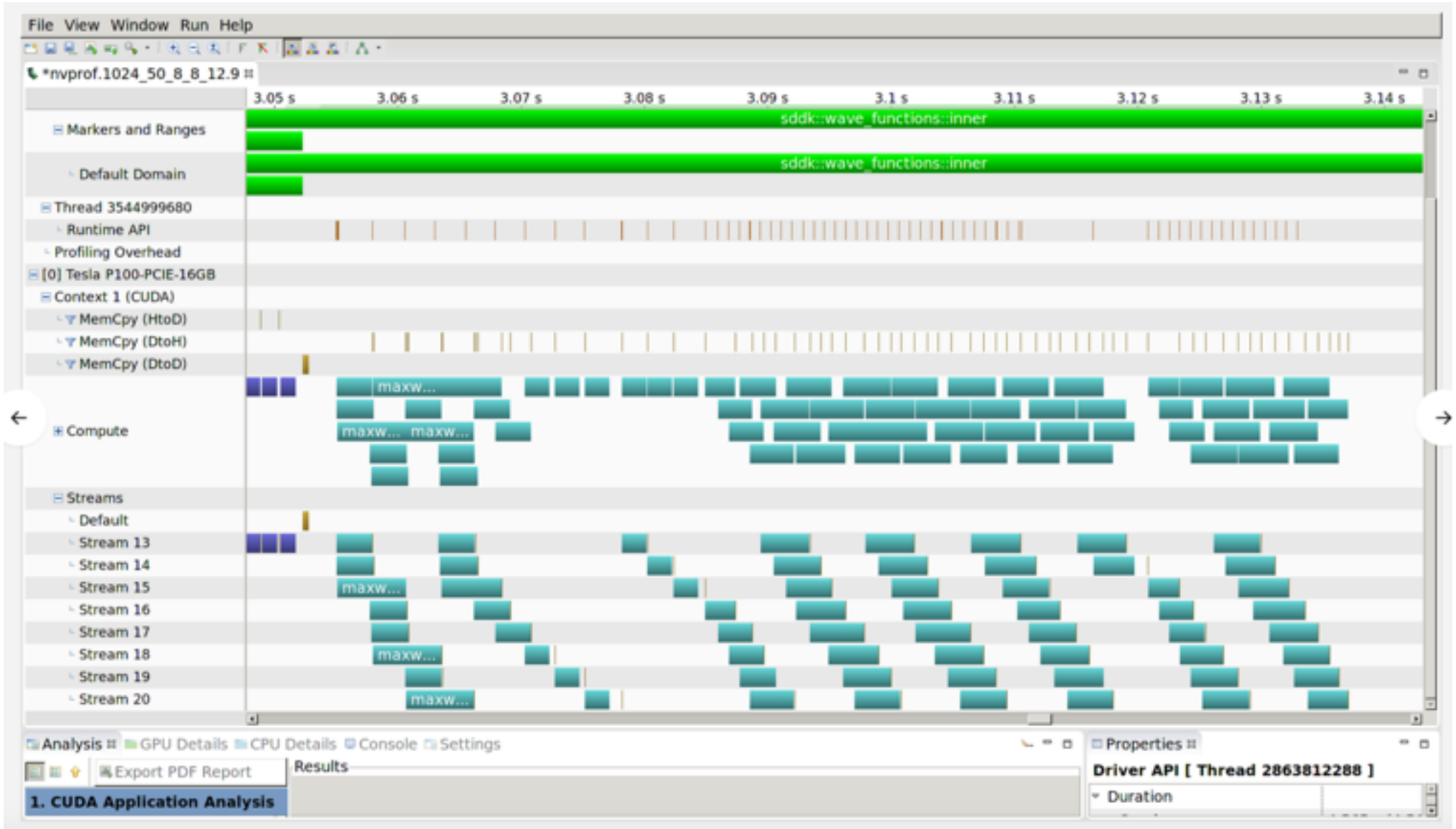
Inner product kernel

Initial implementation
(33 sec. for the total run on 64 nodes)



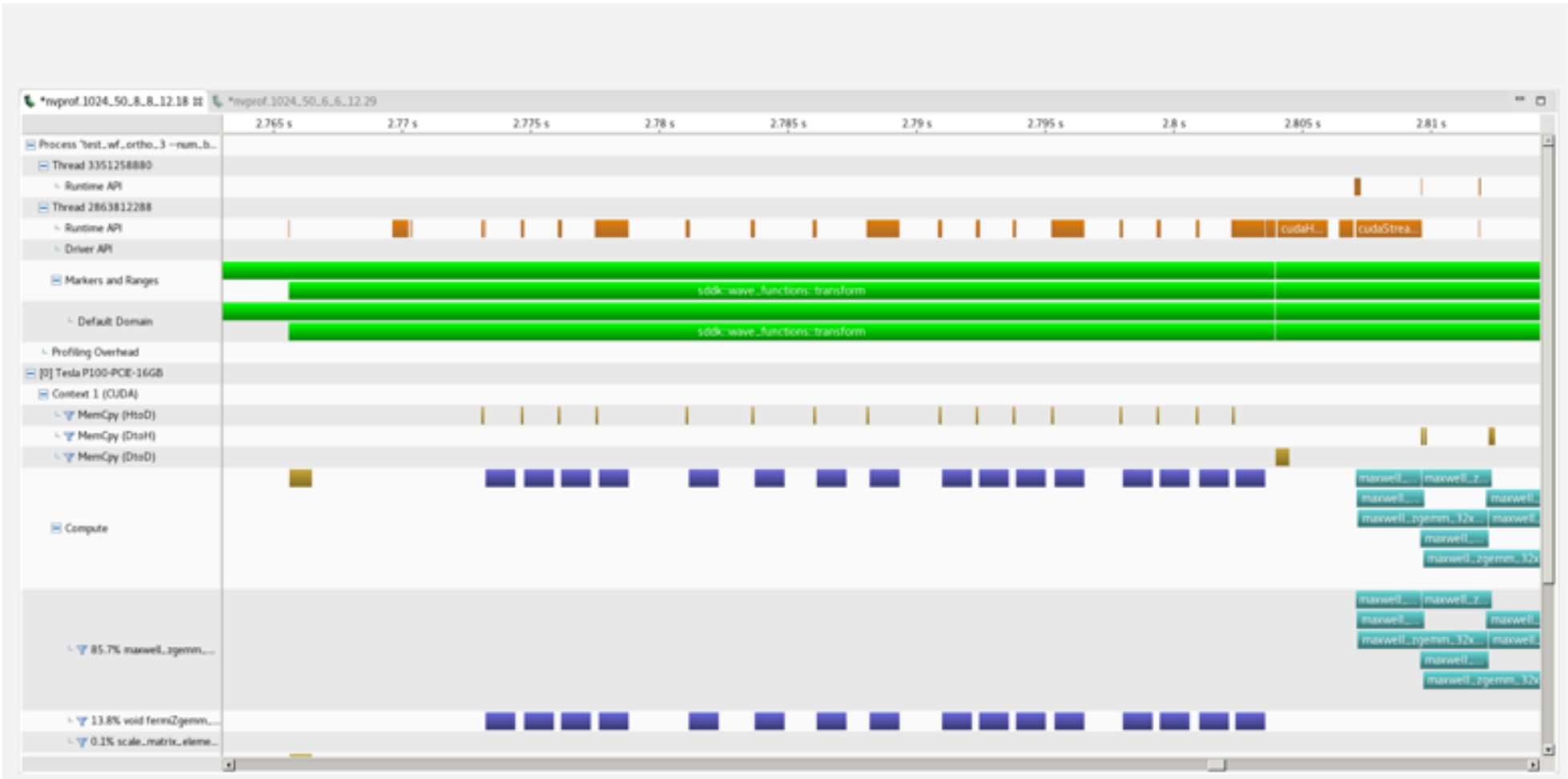
Inner product kernel

Final implementation
(29 sec. for the total run on 64 nodes)



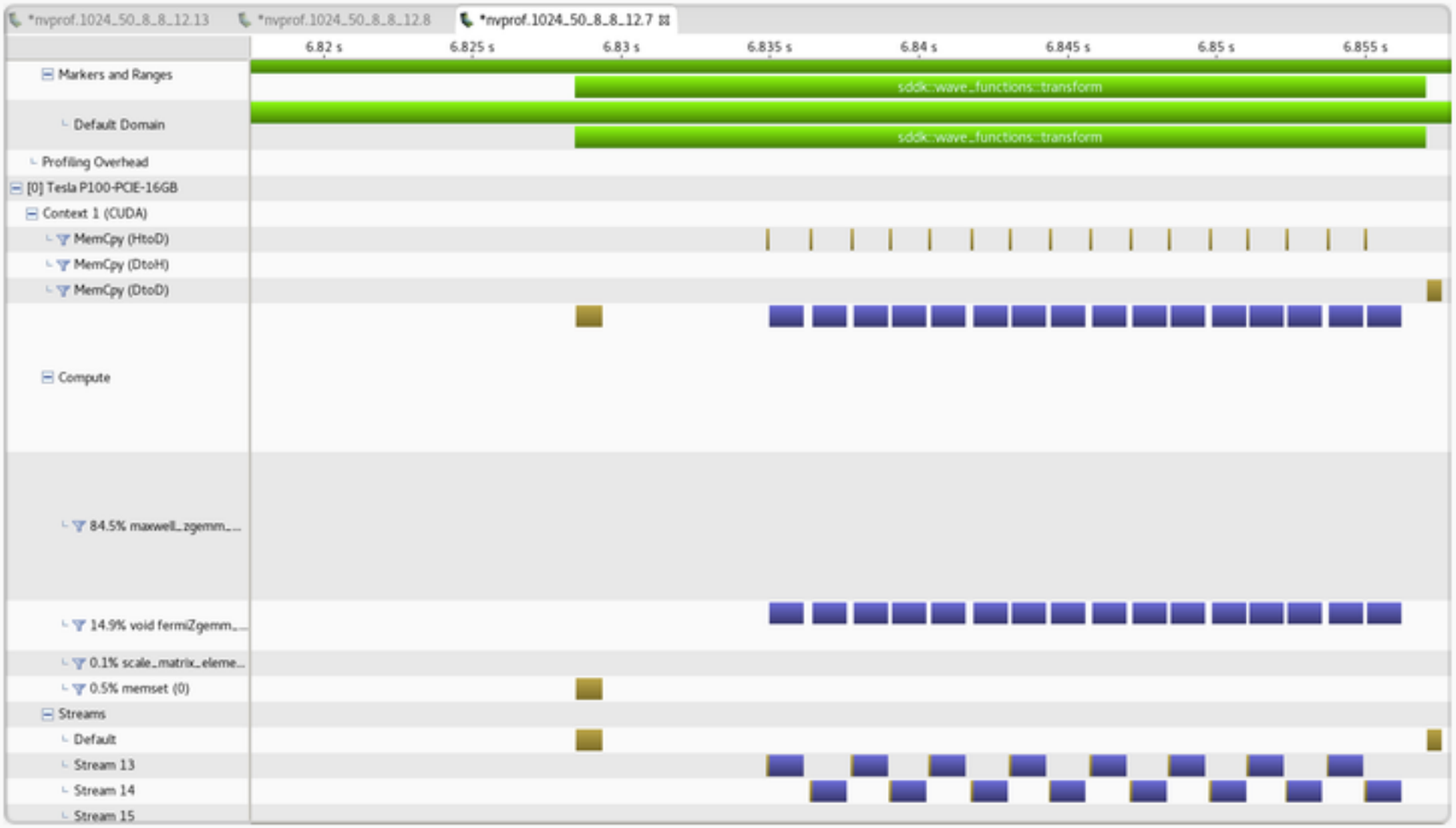
Transformation kernel

Initial implementation
(23 sec. for the total run on 64 nodes)



Transformation kernel

Final implementation
(18 sec. for the total run on 64 nodes)



Was it worth it?

Absolutely yes!

- Good team interaction
- Useful help from NVIDIA
- A lot of homework: merge the hacked versions of the code, cleanup, re-run the benchmarks

Wishlist

- magic profiling tool that can show everything
- *nvvp* application without graphics driver in Linux
- standalone lightweight MPI profiling
- stable and reliable Internet connection