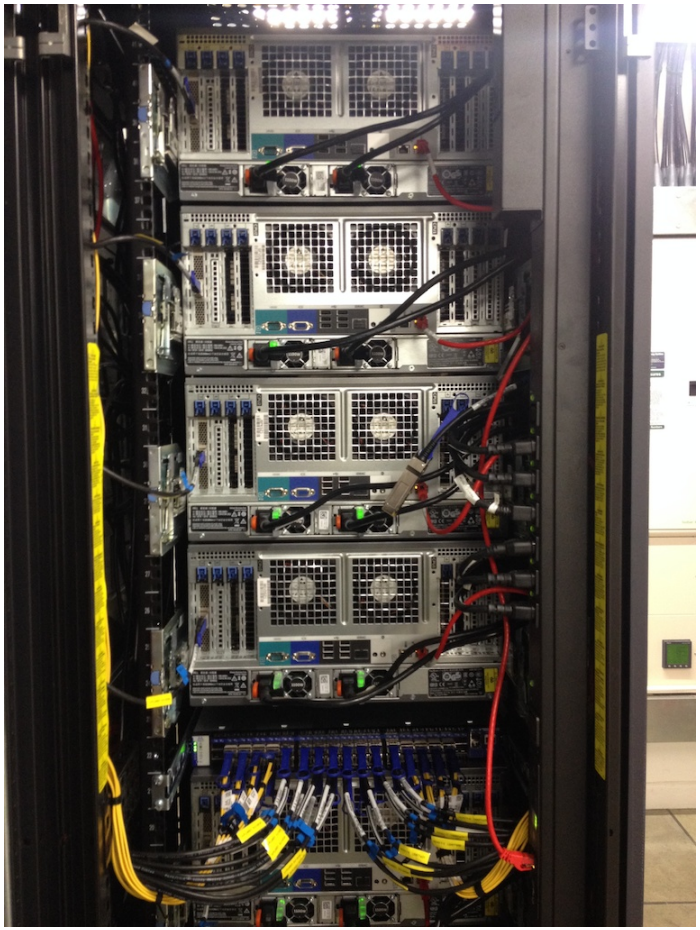
UNIVERSITY OF CAMBRIDGE

GPU Direct Approach for Parallel 3D-FFT in Quantum ESPRESSO

Filippo SPIGA[1,2] <fs395@cam.ac.uk>

[1] High Performance Computing Service, University of Cambridge
[2] Quantum ESPRESSO Foundation

Cambridge GPU system: Wilkes



- 128 nodes Dell T620, dual 6-core Intel Ivy Bridge 2.6 GHz
- dual NVIDIA K20c per node, dual Mellanox Connect-IB FDR
- NVIDIA driver 331.67 (CUDA 6.0), MOFED 2.1-1.0.6, FAST-COPY kernel module

Quantum ESPRESSO

WHAT IS IT...

- QUANTUM ESPRESSO is an integrated suite of computer codes for atomistic simulations based on DFT, pseudo-potentials, and plane waves
- QUANTUM ESPRESSO is free software licensed as GPL. Everybody is free to use it and welcome to contribute to its development
- Huge diffusion in both academia and industries. Growing user community thanks to dissemination actives and international schools.

WHAT IS "SPECIAL" ABOUT IT...

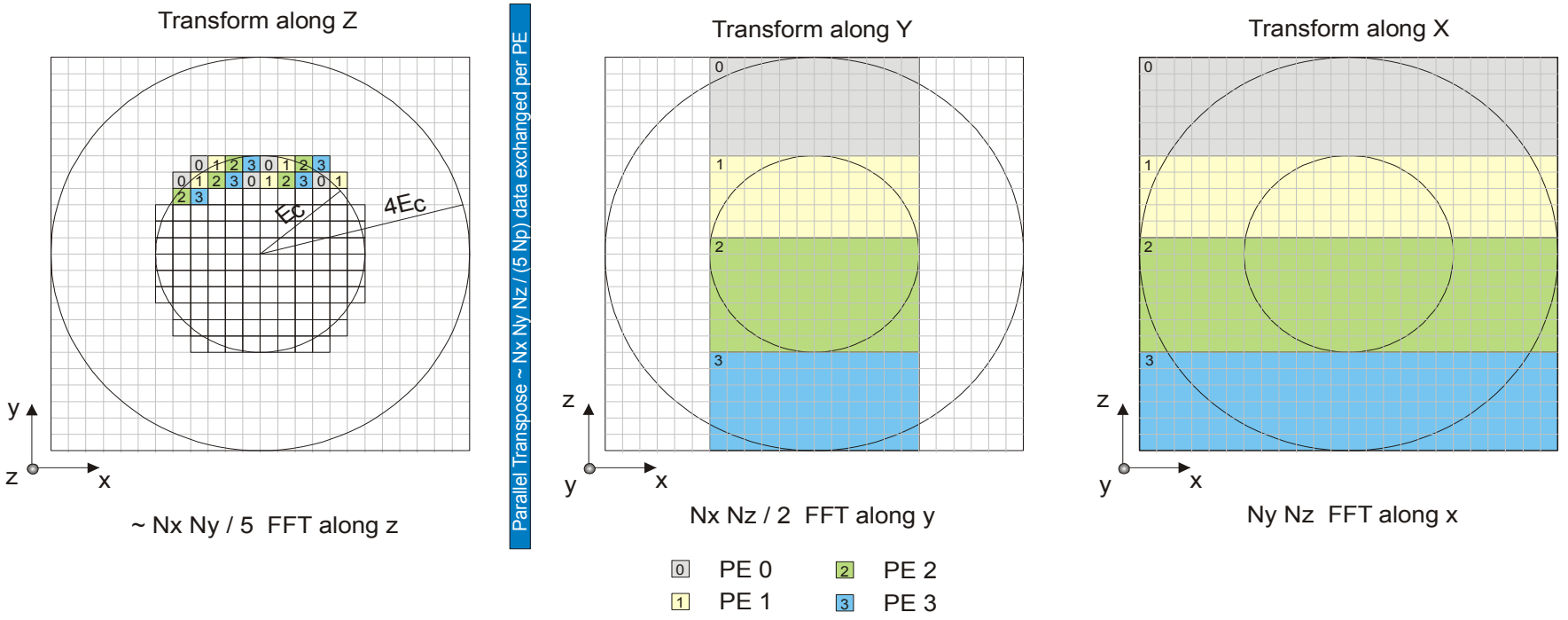
- Not a single "portion" of the code takes the majority of the wall-time
- Code profile varies based on the input case, it can be...
  - 3D-FFT dominant
  - hard solve eigen problem
- Effort for a GPU porting that
  - leverage as much as possible library eco-system (NVIDIA & 3rd parties)
  - avoid massive changes in the data distribution
- Other codes & tools built on top of some shared building-blocks

g-vectors and z-sticks

Commonalities in (lot of) legacy ab-initio plane-wave codes...

- main data distribution around g-vectors
  - #of g-vectors grows based on # atoms, # atomic species and pseudo-potential
- g-vectors have a different "length", the code aims to distribute them homogeneously across all MPI processes involved
- g-vectors that fall "out of the sphere" are not considered, not even computed
- from g-vector it is possible to build z-stick in G space by superposition of multiple of them

3D-FFT peculiarities



H * psi

```
compute/update H * psi:
   compute kinetic and non-local term (in G space)
        complexity : Ni × (N × Ng + Ng × N × Np)
   Loop over (not converged) bands:
        FFT (psi) to R space
             complexity : Ni × Nb × FFT(Nr)
        compute V * psi
             complexity : Ni × Nb × Nr
        FFT (V * psi) back to G space
             complexity : Ni × Nb × FFT(Nr)
        compute Vexx:
             complexity : Ni × Nc × Nq × Nb × (5 × Nr + 2×FFT(Nr))
```

$N = 2 \times N_b$ (where N_b = number of valence bands)
N_g = number of G vectors
N_i = number of Davidson iteration

N_p = number of PP projector
N_r = size of the 3D FFT grid
N_q = number of q-point (may be different from N_k)

Where/Why the (first) GPU porting failed

"Dumb" approach: replace FFT* with CUFFT, leave the (MPI) parallelism as-it-is

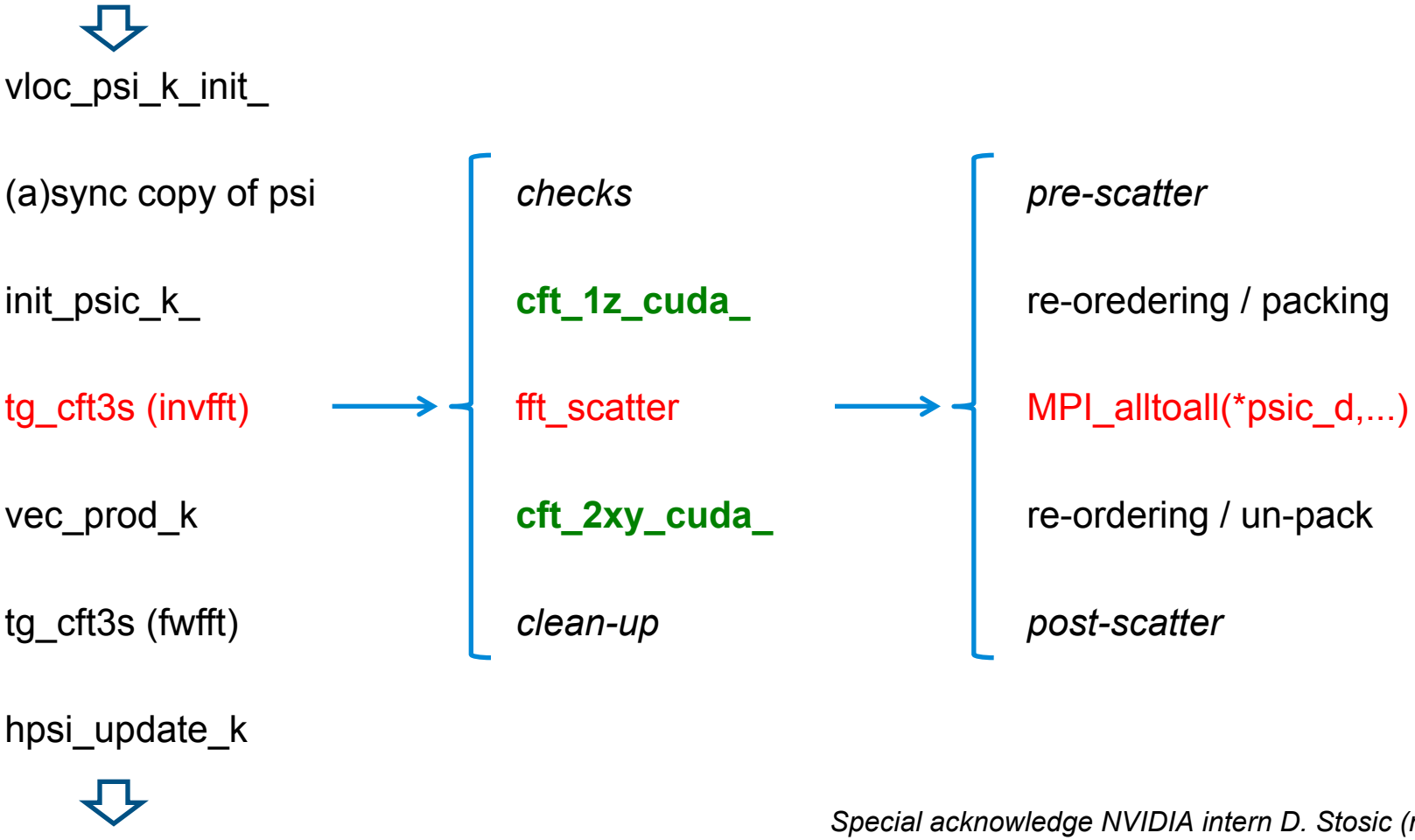
Problems:

- CUFFT is (too) fast for our problem size (~150 double-complex)
- moving data H2D & D&H is too expensive

Workarounds (and drawbacks):

- use task-group to increase local computation...
  - more communication during the "transposition phase" if lot of bands involved
- group batch of (full) 3D-FFT locally and compute...
  - more communication and more memory required
  - we lose GPU memory bandwidth advantage in assembling the FFT grid

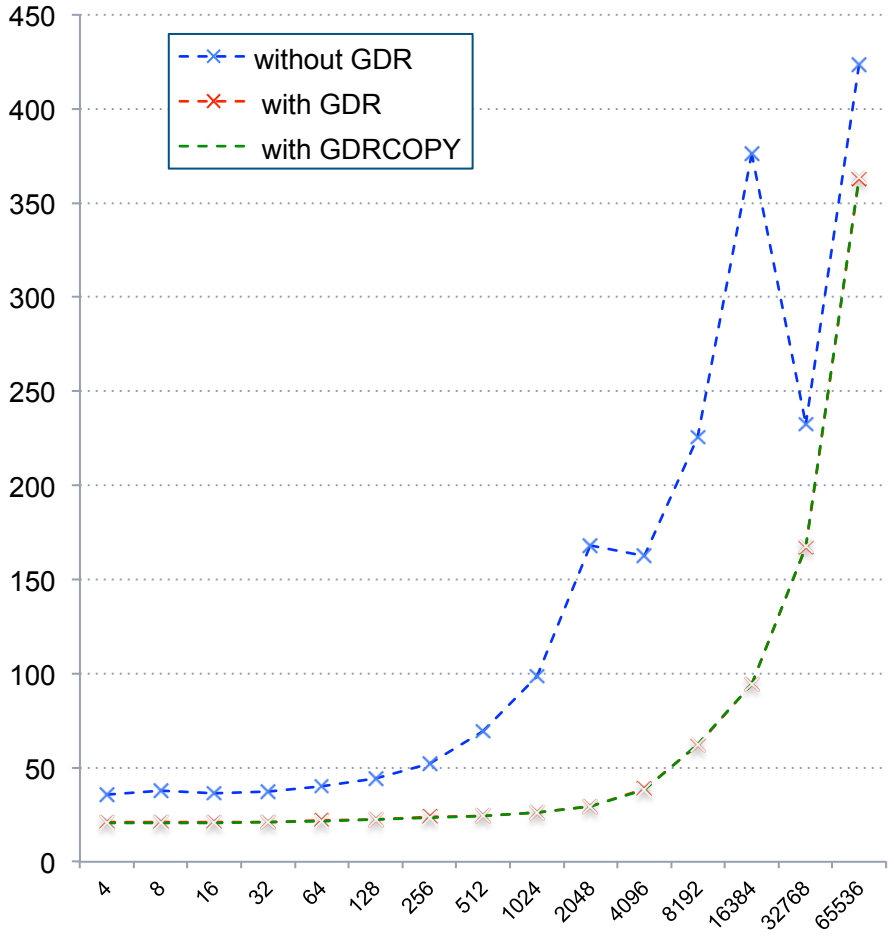
GDR-aware 3D-FFT design



Special acknowledge NVIDIA intern D. Stosic (miniDFT)

Preliminary results (synthetic)

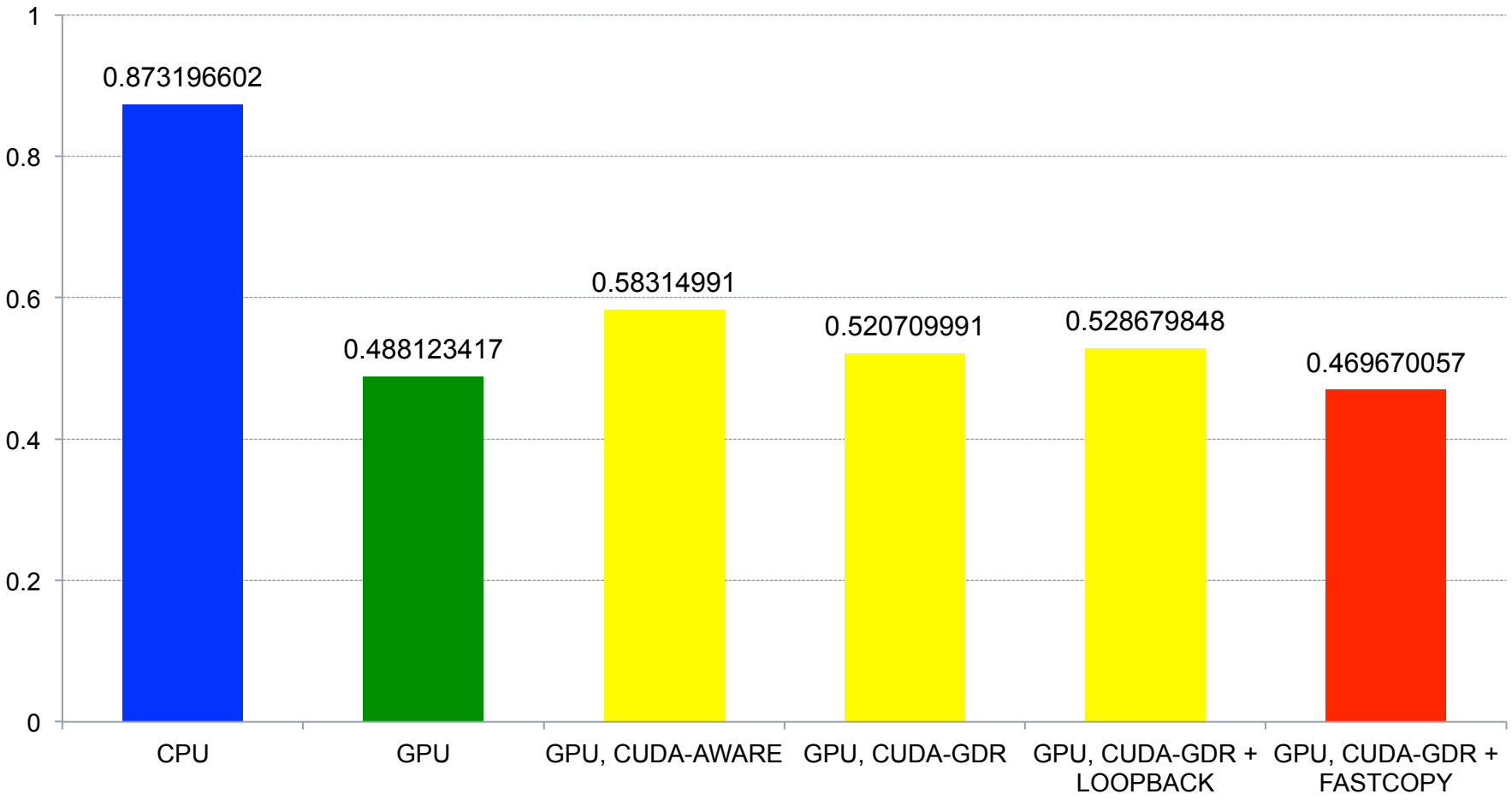
osu_alltoall, mvapich2/2.0-7_GDR/intel-14_cuda-6.0



Size	without GDR	with GDR	with GDRCOPY
4	35.69	21.25	20.72
8	37.66	20.88	20.59
16	36.33	20.99	20.48
32	37.46	21.26	21.22
64	40.23	22.03	21.6
128	44.17	22.29	22.42
256	51.93	23.72	23.4
512	69.09	24.27	24.39
1024	98.74	26.31	26.02
2048	168.03	29.6	29.39
4096	162.68	38.74	38.32
8192	225.64	61.72	62.53
16384	376.26	94.42	94.31
32768	232.3	167.04	167.47
65536	423.54	362.71	363.51

Preliminary results (QE prototype)

Si333, mvapich2-GDR-2.0b-3/intel-12.1_cuda-5.5



Status of the work (in progress)

- Validate design (for the right HW and the right software stack)
- Extend to all cases...
  - "Wave" and "Smooth" grids, gamma and k-point, with/without task-group, ...
- Working on fine-grain tuning (MV2_CUDA*, MV2_GPUDIRECT_*, ...)
- Back-trace special scenarios and handle them
  - Not enough independent 3D-FFTs → group them using task-group
  - Too few independent 3D-FFTs → ignore GPUs
  - non-ideal PCIe topology → let MVAPICH2 team have fun ;-)

THANKS FOR YOUR ATTENTION

«*What I cannot compute, I do not understand.*»
(adapted from Richard P. Feynman)