New FFT data distribution to improve scalability

Stefano de Gironcoli
Scuola Internazionale Superiore di Studi Avanzati
Trieste-Italy

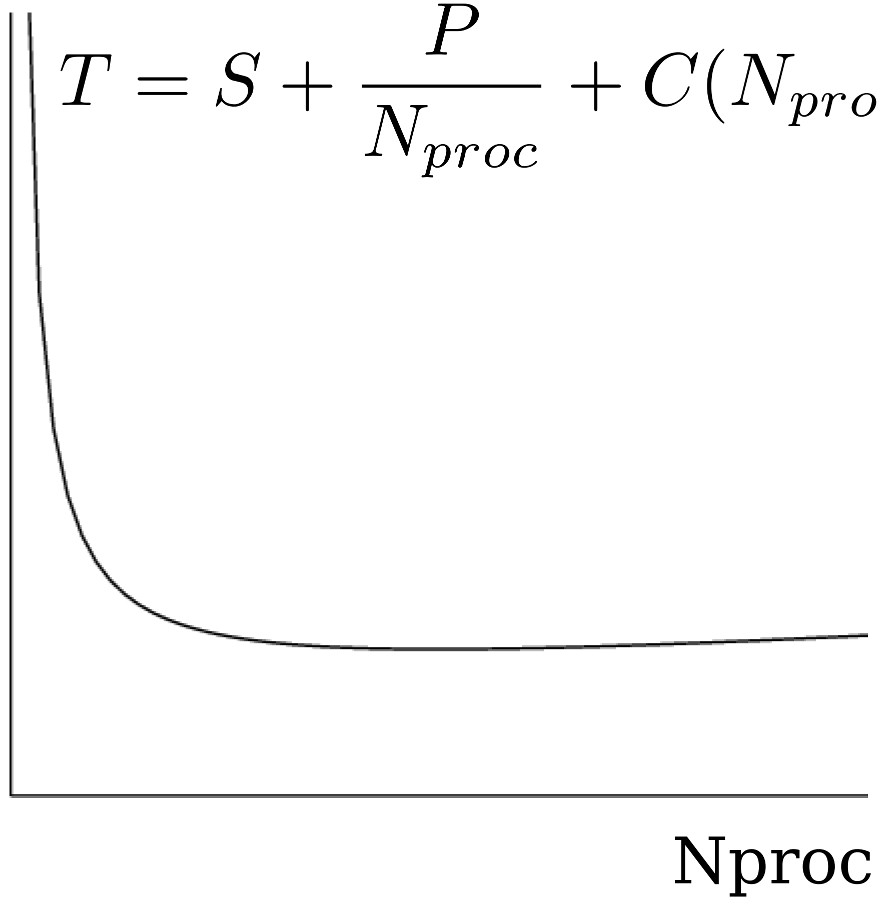
Strong Scaling vs Weak Scaling

Strong Scaling: scaling when system size remains fixed
Weak Scaling: scaling when system size also grows

Strong Scaling is much more difficult to achieve than Weak Scaling.

Amdahl's law

$$T = S + \frac{P}{N_{proc}} + C(N_{proc})$$



No matter how well you parallelize your code on the long run the scalar fraction dominates.

MPI – Message Passing Interface

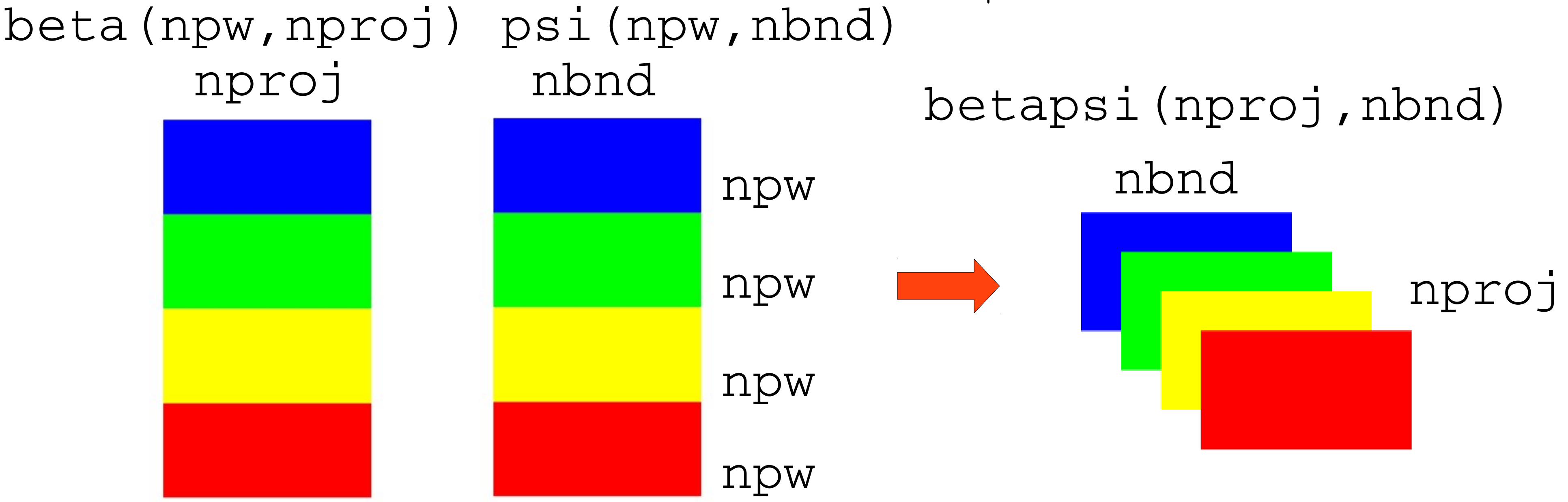
hierarchy of parallelization in PW

```
mpirun -np $N pw.x -nk $NK -nb $NB -nt $NT -nd $ND
                                       < pw.in > pw.out


 -nk (-npool, -npools)    # of k-point pools

 -nb (-nband, -nbgrp, -nband_group) # of band groups

 -nt (-ntg, -ntask_groups) # of FFT task groups
 -nd (-ndiag, -northo,-nproc_diag, -nproc_ortho)
                           # of linear algebra groups

               $N = $NK x $NB x $NT x $Nproc_R&G
```

MPI – Message Passing Interface

a simple example

$$\langle \beta_i | \psi_j \rangle = \sum_{k+G} \beta_i^*(k+G)\psi_j(k+G)$$



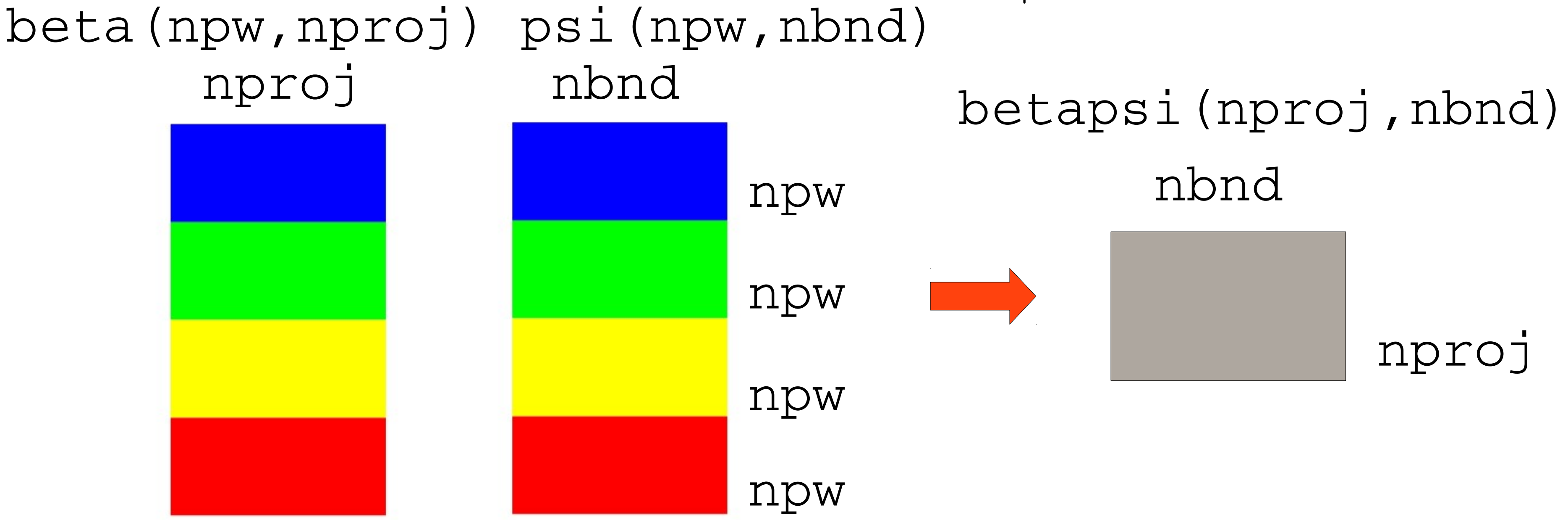
```
CALL ZGEMM( 'C', 'N', nproj, nbnd, npw, (1.0_DP,0.0_DP), &
                  beta, npwx, psi, npwx, (0.0_DP,0.0_DP), &
                  betapsi, nprojx )
```

each processor has a partially summed betapsi

MPI – Message Passing Interface

a simple example

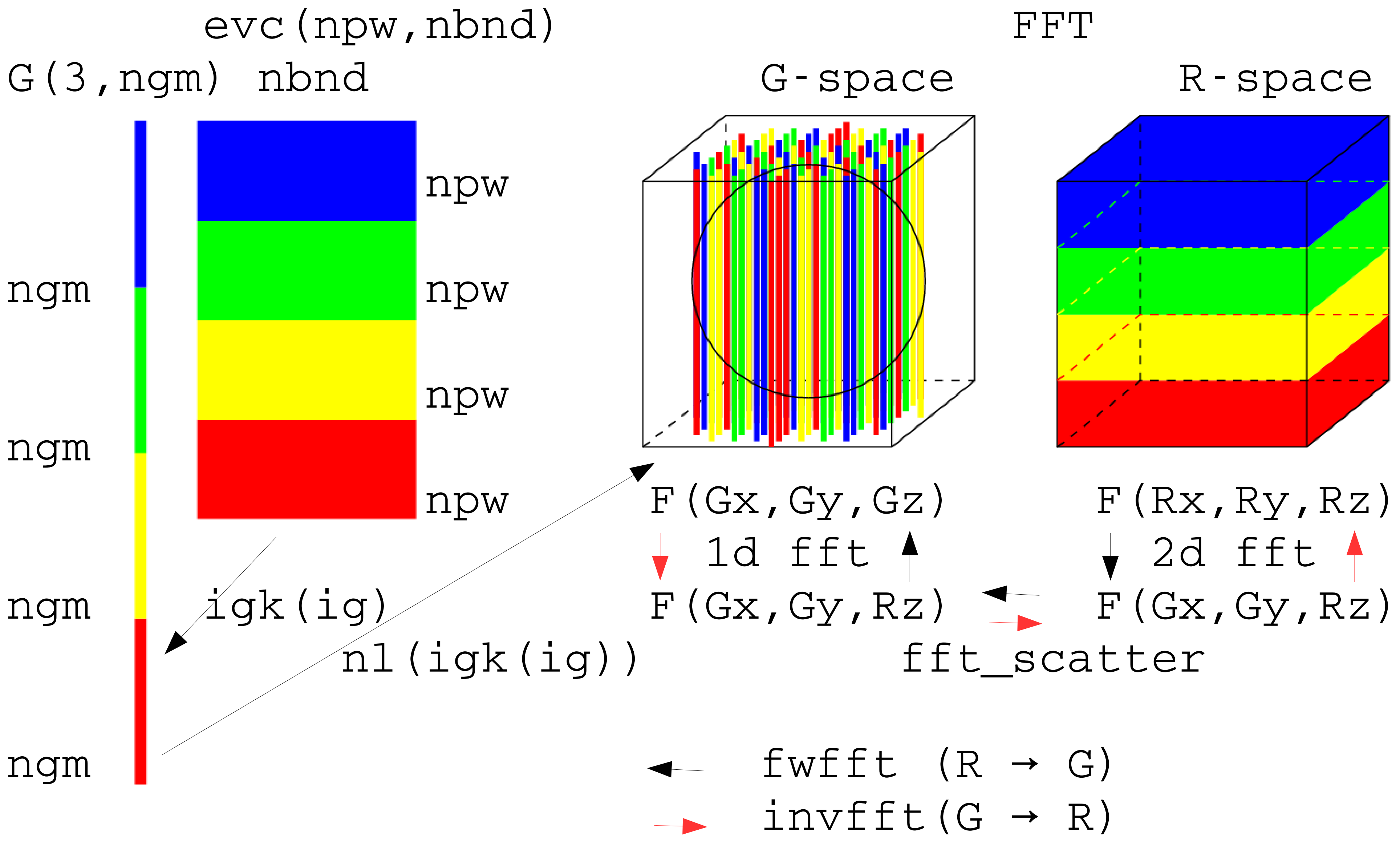
$$\langle \beta_i | \psi_j \rangle = \sum_{k+G} \beta_i^*(k+G)\psi_j(k+G)$$



```
CALL ZGEMM( 'C', 'N', nproj, nbnd, npw, (1.0_DP,0.0_DP), &
                 beta, npwx, psi, npwx, (0.0_DP,0.0_DP), &
                 betapsi, nprojx )
CALL mp_sum( betapsi, intra_bgrp_comm )
```

at the end each processor has the complete betapsi !

R & G parallelization



FFT scalability

-computation time

$$T_{\mathtt{comp}} \propto T_{\mathtt{scalar}} / N_{\mathtt{proc}}$$

-communication time

$$T_{\mathtt{comm}} \propto (N_{\mathtt{proc}} - 1)\,(\alpha + \beta N_{\mathtt{data}})$$

$$N_{\mathtt{data}} = N_{\mathtt{col/proc}} \times N_{\mathtt{plane/proc}}$$

$$N_{\mathtt{col/proc}} \propto \frac{nr1 \times nr2}{N_{\mathtt{proc}}} \qquad N_{\mathtt{plane/proc}} = \frac{nr3}{N_{\mathtt{proc}}}$$

-for large number of processors communication time
becomes important and latency becomes dominant

-the number of processors that can be used efficiently
is limited by nr3 grid dimension !

MPI – Message Passing Interface

hierarchy of parallelization in PW

- **R & G space parallelization**

- **K-point parallelization**

- **Band parallelization**

- **linear algebra parallelization**

- **task group parallelization**
 FFT data are redistributed to perform multiple FFTs at the same time. Needed when number of processors is large compared with FFT dimension (nr3).
 It complicates the code significantly and interferes with band paralellization. Only applies to wfc FFTs, not to rho & potential FFTs.

MPI – Message Passing Interface

hierarchy of parallelization in PW

-**R & G space parallelization**

-**K-point parallelization**

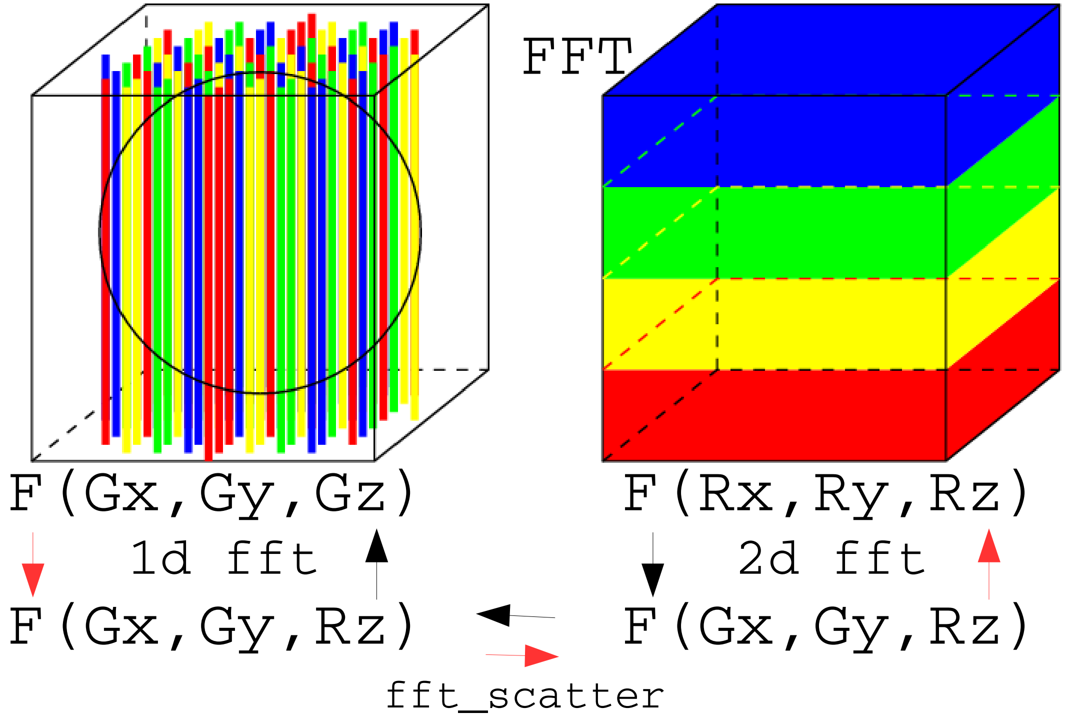
-**Band parallelization**

-**linear algebra parallelization**

-**task group parallelization**
FFT data are redistributed to perform multiple FFTs at the same time. Needed when number of processors is large compared with FFT dimension (nr3).
It complicates the code significantly and interferes with band paralellization. Only applies to wfc FFTs, not to rho & potential FFTs.

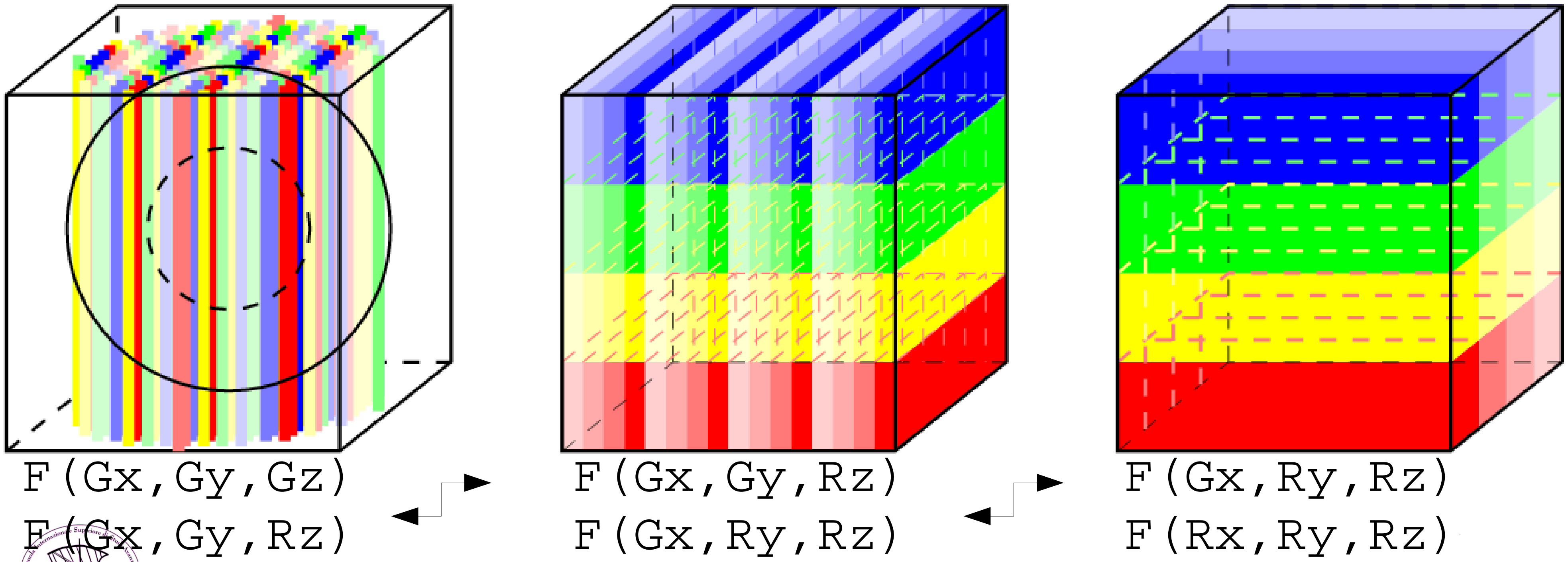
A different R&G distribution might be more useful

Old FFT distribution



New FFT distribution

fft_scatter



Extended FFT scalability $N_{\mathtt{proc}} = N_{\mathtt{P}_1} \times N_{P_2}$

-computation time

$$T_{\mathtt{comp}} \propto T_{\mathtt{scalar}} / N_{\mathtt{proc}}$$

-communication time

$$T_{\mathtt{comm}} \propto (N_{\mathtt{P}_1} - 1)\left(\alpha + \beta N^{(1)}_{\mathtt{data}}\right) + (N_{\mathtt{P}_2} - 1)\left(\alpha + \beta N^{(2)}_{\mathtt{data}}\right)$$

$$N^{(1)}_{\mathtt{data}} \propto \frac{nr1 \times nr2 \times nr3}{N_{\mathtt{proc}} N_{\mathtt{P}_1}}, \quad N^{(2)}_{\mathtt{data}} \propto \frac{nr1 \times nr2 \times nr3}{N_{\mathtt{proc}} N_{\mathtt{P}_2}}$$

-for large number of processors communication time grows more slowly, ideally as $\sqrt{N_{\mathtt{proc}}}$

-the number of processors that can be used efficiently could be extended well beyond nr3 grid dimension !

•task parallelization *vs* new FFT distribution

•The hope was the new FFT distribution could get rid of task group parallelization.

•It can't: when possible TG parallelization makes fewer communications of larger amount of data.

•But the new FFT distribution is possible also with CG diagonalization and applies to densities and potentials as well.

•The idea is then to unify the two parallelization strategies: introduce the distribution of partial planes to be used when task group cannot be used and for densities and potentials.

•A final word on Gamma_only calculations...

•wfcs in real space can be taken *rea*l.

•In order to gain efficiency two wfcs are tranformed togheter. This complicates the code and can only be applied to wfcs (and not with CG)

•If the FFT of a *real* wfc would cost ½ as for *complex* wfc this would not be necessary.

•It can ! One needs to move the 2-wfc trick inside the FFT and apply it to the set of x-row FFTs.