Optimisation of computing time

- How does GuineaPig work?
  - bunches are cut into slices which are moved longitudinaly and interact when they are in the same transverse plan

Optimisation of computing time

- for each slice-slice interaction
  - particles are distributed on the grid
  - integration of the field equation
  - particles are moved and photons are generated
  - e-e+ interaction: luminosity, …
  - if asked
    - photons are distributed and moved on the grid
    - if asked, pairs are generated and moved

Optimisation of computing time

- runs with GuineaPig++ : total time consists of the elapsed real time between invocation and termination
- computation time (% of total time)

nx=32; ny=32; nz= 32

	nm= 10,000	nm= 100,000 particles
distribute particles	11.7%	2.3%
fftw	15%	0.5%
distribute photons	6.7%	15%
generate pairs	40%	80%
tracking pairs	22%	0.5%

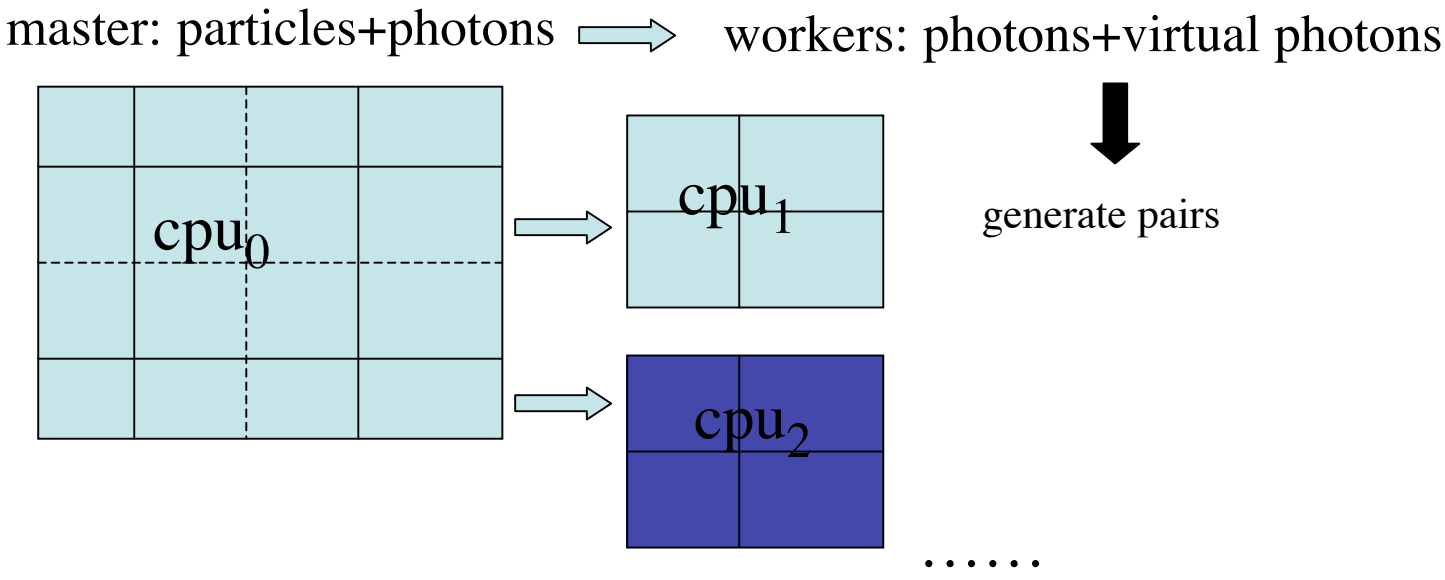
Optimisation of computing time

nx=32; ny=64; nz= 32

	nm= 100,000
distribute particles	4.4%
fftw	1.7%
distribute photons	13.9%
generate pairs	76.82%
tracking pairs	0.86%

Optimisation of computing time

- Distribute particles among processes
  - assign processes to each part of cells



Optimisation of computing time

- with particles & photons received on each process, the pairs are generated

- track the pairs, assuming the field is known on each process

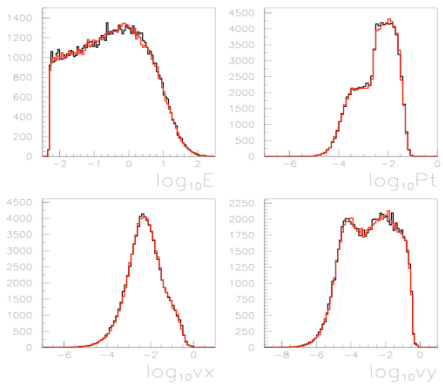
Optimisation of computing time

- Tests for 2 types of decomposed domain:
  - domain is divided into 4 parts [0,nx/2]x[nx/2,nx]U[0,ny/2]x[ny\2,ny] and each part is assigned to a process (type 1)
  - cells are distributed on processes taking into account the number of particles in the cells. A good distribution of the load on each process is expected (type 2)

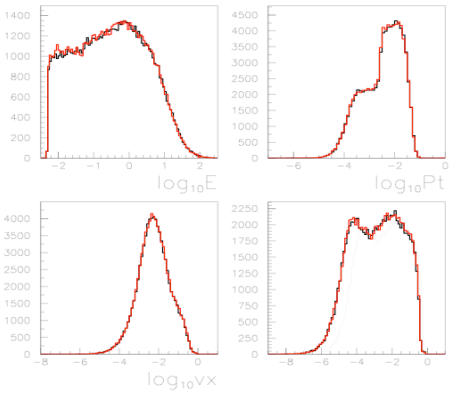
Optimisation of computing time

On the figures are plotted logarithmic distributions of pairs energy,transverse momentum,velocities vx and vy produced with g++ and with the two types of decomposition and using 1 or 4 processes (CPUs)

type1: 1cpu(in black)/4 cpus(in red)

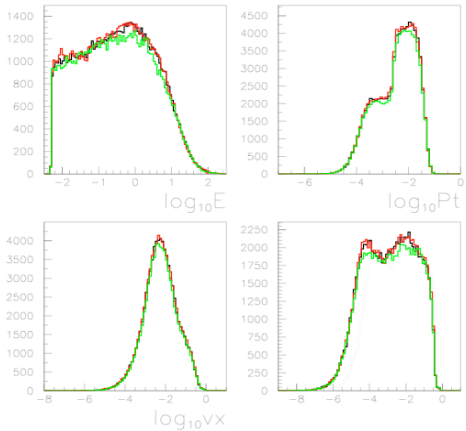


type 1(in black)/type 2 (in red)

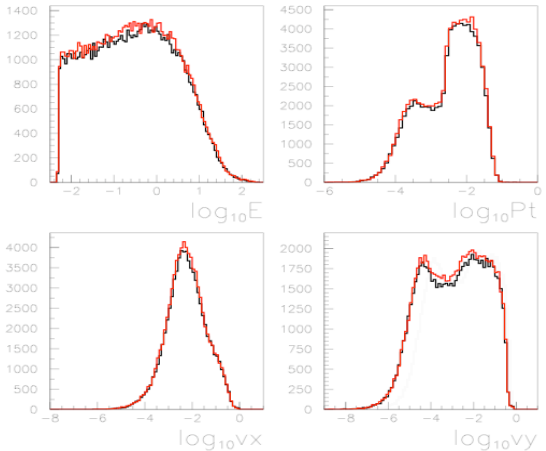


Optimisation of computing time

type 1(in black)/type 2 (in red)
&gp++(in green)



gp++(in red)/type 1(in black)
with 100,000 particles



Optimisation of computing time

- expect good speedups:

  test with 100,000 particles and a grid dimension with nx=32;ny=64;nz=32

  - g++ 1635s
  - type 1 with 4 CPUs = 606s (63%)
  - type 2 with 4 CPUs = 936s (43%)

Optimisation of computing time

- Conclusion
  - transfert of particles are accomplished in MPI
  - focuse on others approaches in the distribution of particles