# ACCELERATING PP-DISTANCE ALGORITHMS ON FPGA, WITH PARALLELIZATION DIRECTIVES AND DATA TRANSFER OPTIMIZATIONS

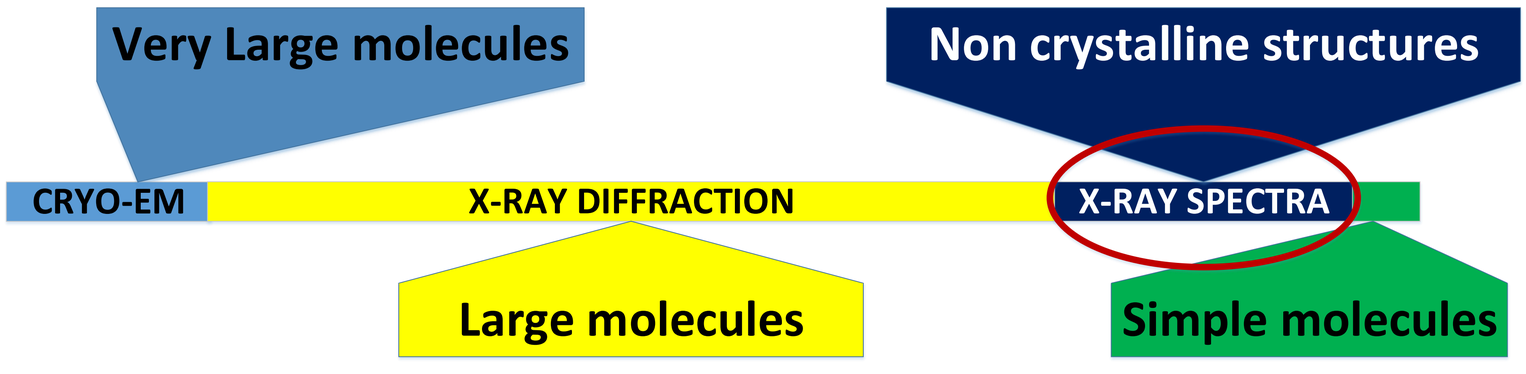
IWOCL / SYCLcon

MUNICH, GERMANY
27-29 APRIL 2020

*César González (BSC / UB / IQAC-CSIC), Simone Balocco (UB) and Ramon Pons (IQAC-CSIC)*

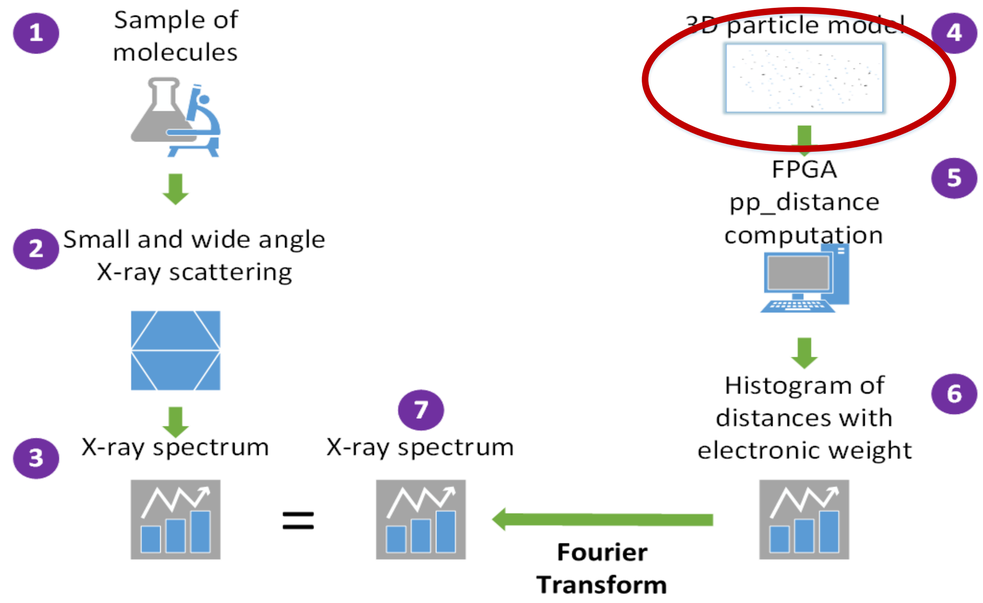
1

**Protein structure** is one of the bases of the biomedicine and nanotechnology bases. Different methods are used to determine the structures.



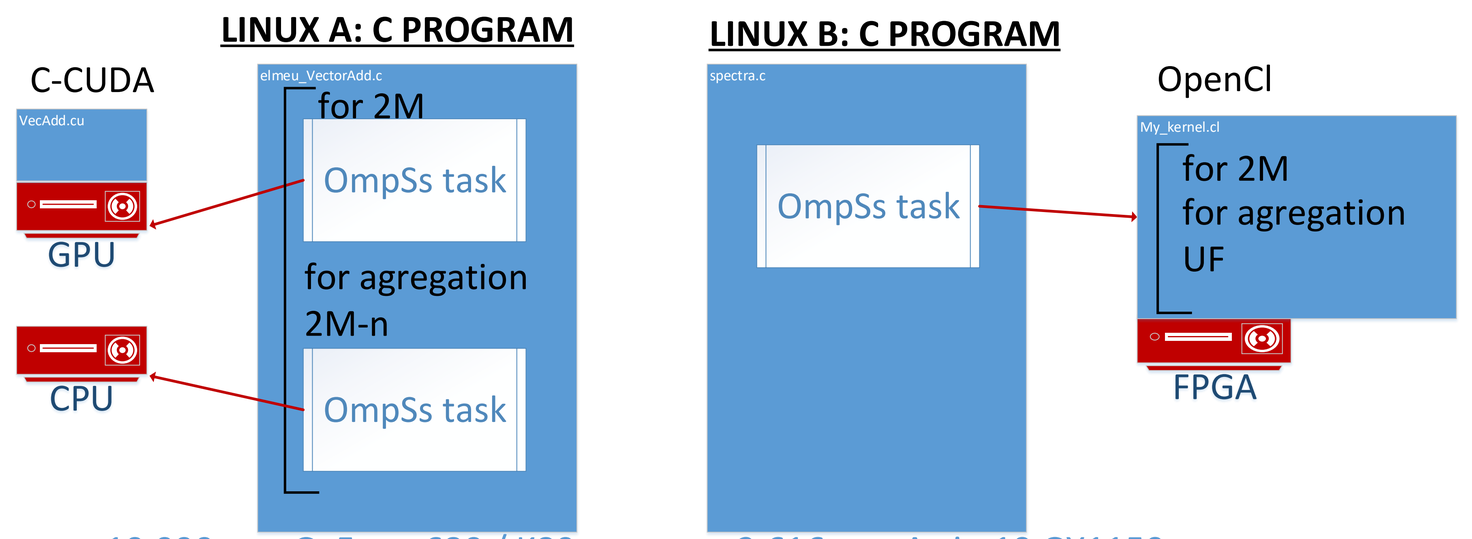

2

The particle-particle distance problem (pp-distance) is used in several applications from **Astrophysics** to **Molecular Dynamics**. We solve the pp-distance problem to calculate the theoretical X-Ray spectra (see right #7) and validate it with the real one (see right #3). Our C program reduce the consuming time using HPC over FPGA and GPU devices (see bellow).



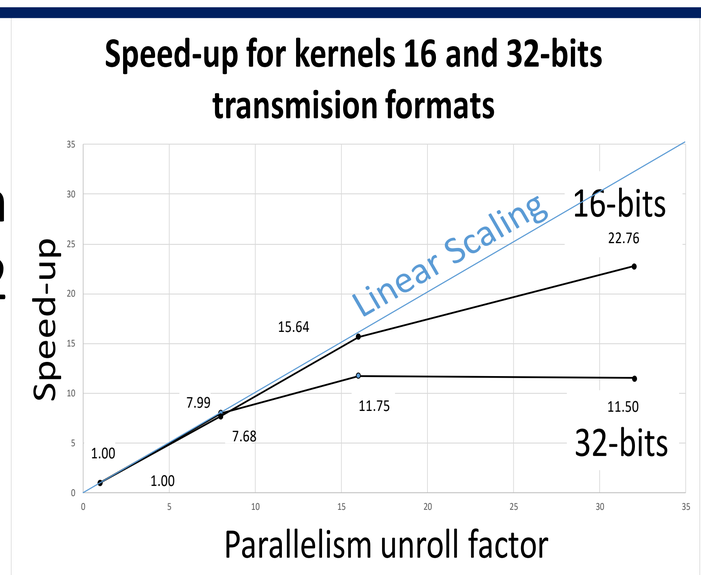

3

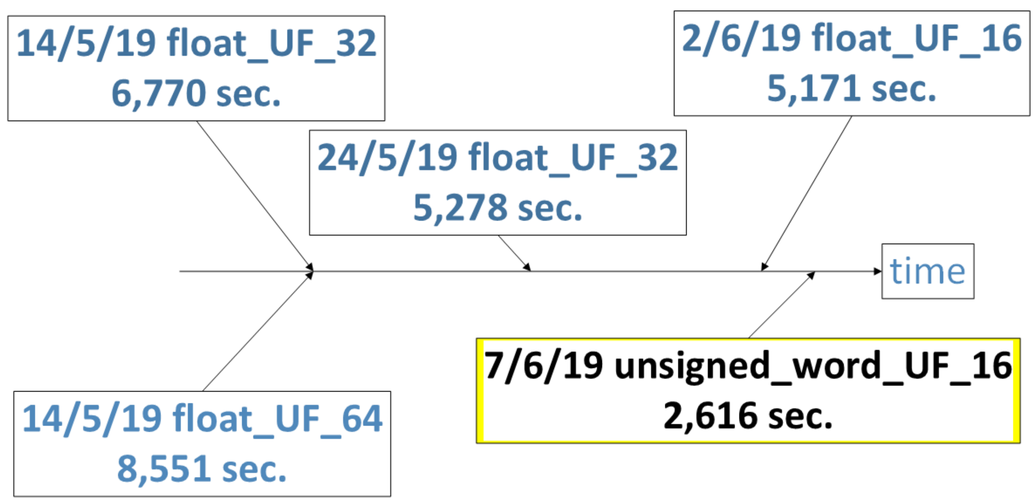
We have to compute all the distances between a set of *N* particles. This is a ***O(N²)* complexity problem**, where *N* is the particles number. FPGA OpenCL Kernel is called by OmpSs.




4

**The benchmarks** have been done computing a model of 2 million particles.

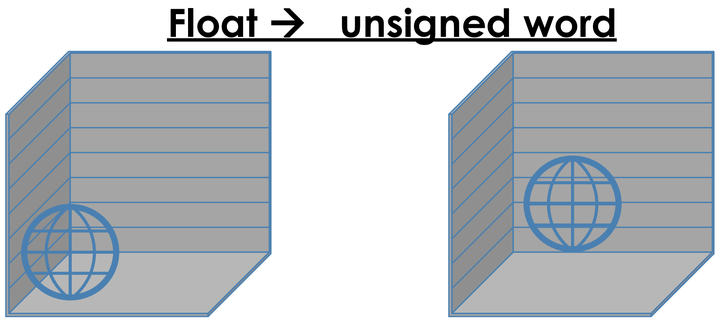



5

The **dynamic unroll**, the kernel code:

```
int my_threshold = ((N-j)/UF) * UF; // particles for the unrolls
for (l=j; l<j+my_threshold; l+=UF) {
   #pragma ivdep
   #pragma unroll UF
   for (m=0; m < UF; m++) {
      int k = l + m;
      int distancia  = (sqrt( (x-h_B1[k]+factor_coor)*(x-h_B1[k]+factor_coor) + (y-h_B2[k]+factor_coor)*(y-h_B2[k]+factor_coor) + (z-h_B3[k]+factor_coor)*(z-h_B3[k]+factor_coor)));
      float densitat = p * (h_B4[k]-factor_pes )  / 1000.0;
      j_B5[distancia][m]+= densitat; // sumem els pesos de les densitats electroniques a la posicio de la seva distancia en el seu vector de paral.lelitzacio
   }
}
// REMAINDER CASES for each “j”
for (k=j+my_threshold; k<N; k++) {
   // printf ("Avancem final %i\n", k);
   int distanciaR  = (sqrt( (x-h_B1[k]+factor_coor)*(x-h_B1[k]+factor_coor) + (y-h_B2[k]+factor_coor)*(y-h_B2[k]+factor_coor) + (z-h_B3[k]+factor_coor)*(z-h_B3[k]+factor_coor)));
   float densitatR = p * (h_B4[k]-factor_pes )  / 1000.0;
   j_B5[distanciaR][0]+= densitatR; // IDEM al vector "0" els remainder cases
}
```




1.- We move to positive coordinates.
2.- We work with the integer coordinates (no decimals).

Barcelona Supercomputing Center
Centro Nacional de Supercomputación
UNIVERSITAT DE BARCELONA
CSIC iQAC
*cesar.gonzalez@bsc.es Programming Models, Barcelona Supercomputing Center (BSC), Spain.*
*Department of Mathematics and Informatics, Universistat de Barcelona (UB), Spain.*
*Institute for Advanced Chemistry of Catalonia, Spanish National Research Council (IQAC-CSIC).*