

Parallel and GPUs programming in PLUMED

Daniele Rapetti

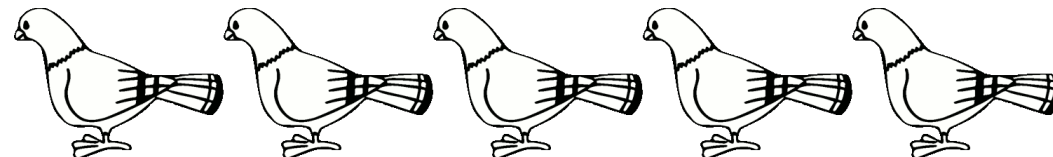
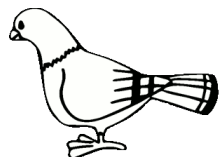
Enhanced sampling methods with PLUMED
2023 @CECAM



Why parallelism

- In the past: you just wait a year, and you would get a 2x performance due to Moore's "law"
- Now you need to write better quality code
- And exploit hardware features other than speed
 - SIMD (Single Instruction Multiple Data) this is advanced, and usually the compiler is better than you in at SIMD
 - Multiple CPUs cores
 - GPUs

Serial vs Parallel



SISSA

4/7/2023

Parallel and GPUs programming in PLUMED@CECAM,
Daniele Rapetti



Different architectures

CPU

- High **serial** performances
- Fast, but few cores
 - speed beats quantity
- **Low latency** (lots of operation in fewer time)

GPU

- High **parallel** performances
- Slow, but 100s to 1000s of cores
 - quantity beats speed
- **High throughput** (higher quantity of data processed)

Different parallelism protocols

openMP (shared memory)

- Calculations are done in different **threads**
- The data is not duplicated
- **Race conditions** will be present, during calculations

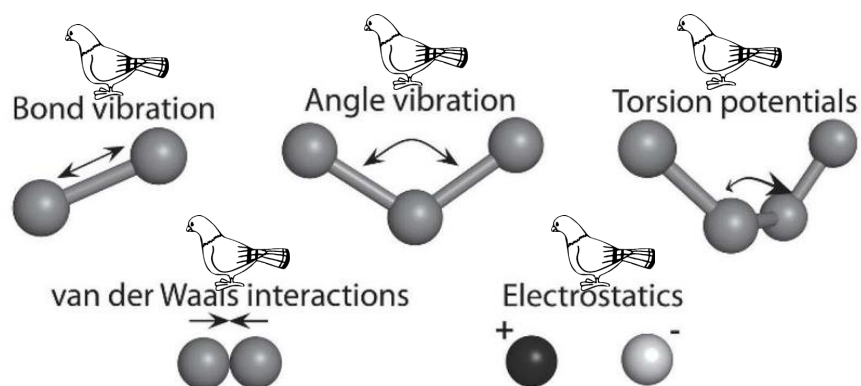
Message Passaging Interface (mpi)

- Calculations are done in different **processes**
- Data is **duplicated**: there is a **communication** layer
- Race conditions may happen, during communication

Concurrency and parallelism

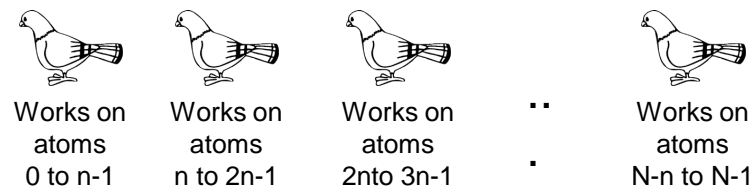
concurrency

- Do different things



parallelism

- Do the same operation on different pieces of data



Parallelism in Plumed

The openmp interface

- PLMD::OpenMP

The MPI interface

- PLMD::Communicator

Better start with some examples

Hands on:

https://github.com/Iximiel/PlumedFlagship_parallelism

- OpenMP:
 - `OpenMP::getNumThreads()`
- MPI:
 - `PLMD::Communicator::Get_size()`
 - `PLMD::Communicator::Get_rank()`
 - `PLMD::Communicator::Sum()`



THANK YOU!!!



4/7/2023

Parallel and GPUs programming in PLUMED@CECAM,
Daniele Rapetti

