Tecniche di programmazione in chimica computazionale

Optimisation

Emanuele Coccia

Dipartimento di Scienze Chimiche e Farmaceutiche

Optimization

- Make the program faster and more efficient

Optimization

- Make the program faster and more efficient
- Compiler options to optimize the program

Optimization

- Make the program faster and more efficient
- Compiler options to optimize the program
- User changes of the source code

Optimization

- Make the program faster and more efficient
- Compiler options to optimize the program
- User changes of the source code
- Check the performances (profiling)

Optimization options

- Use optimization options of the compiler: -Ok (k=0,1, 2, 3) (type *man ifort* and search "Specifies the code optimization for applications.")
- Automatic optimization improving code performance
- Example with profiling.f90
- Compile with -O0, -O1, -O2 and -O3 (n=100 and m=1000 as input)
- Run the executable by typing time ./profiling.x

Profiling

- Detailed analysis of the code performance

Profiling

- Detailed analysis of the code performance
- Done using a compiler option:
  1. ifort -pg -o code.x code.f90

Profiling

- Detailed analysis of the code performance
- Done using a compiler option:
  1. ifort -pg -o code.x code.f90
  2. ./code.x

Profiling

- Detailed analysis of the code performance
- Done using a compiler option:
  1. ifort -pg -o code.x code.f90
  2. ./code.x
  3. gprof code.x > profile

Profiling

- Detailed analysis of the code performance
- Done using a compiler option:
  1. ifort -pg -o code.x code.f90
  2. ./code.x
  3. gprof code.x > profile
- Example profiling.f90