Introduction to High-Performance Computing

Exercise/2

Giorgio Amati

Corso di dottorato in Ingegneria Aeronautica e Spaziale 2024

g.amati@cineca.it / g.amaticode@gmail.com

Agenda

- ✓ Exploit performance for matrix-matrix Multiplication
  - Fortran/C
- ✓ Always check the results
- ✓ Extract some Performance figure (in MFLOPs)
  - Change order of loops
  - Change size
  - With blocking
  - With unrolling
  - All together!!!!!
- ✓ Any available Compiler
- ✓ Medium optimization level (`-O2`)

How to do

- ✓ Clone the repository
  - `git clone https://github.com/gamati01/HPCLessons.git`

```
.
├── ESER1
│   ├── clean.sh
│   ├── compile.c.sh
│   ├── compile.fortran.sh
│   ├── EXERCISE1.pdf
│   ├── inc_precision.h
│   ├── mm.c
│   ├── mm.F90
│   ├── mod_tools.F90
│   └── README
├── LESSON1
│   ├── HPC-1.pdf
│   └── HPC-1-spoiler.pdf
└── README.md
```

Homework/1: Fill the table

Loop order	Size	Fortran	C
`i,k,j`	512*512		
`j,k,i`	512*512		
`i,k,j`	4096*4096		
`j,k,i`	4096*4096		

- ✓ Compiler used:
- ✓ Compiler option used:
- ✓ HW used:

Homework/2: Fill the table

✓ Unrolling external loop

#unrolling	Size	Fortran	C
2	512*512		
4	512*512		
2	4096*4096		
4	4096*4096		

✓ Compiler used:
✓ Compiler option used:
✓ HW used:

Homework/3: Fill the table

✓ Cache blocking: try a couple of blocking size…

#blocking	Size	Fortran	C
?	512*512		
?	512*512		
?	4096*4096		
?	4096*4096		

✓ Compiler used:
✓ Compiler option used:
✓ HW used:

Homework/4: Fill the table

✓ Put all together (blocking, unrolling,....)

#unrolling	#blocking	Size	Fortran	C
?	?	4096*4096		
?	?	8192*8192		

✓ Compiler used:
✓ Compiler option used:
✓ HW used: