Introduction to High-Performance Computing

Exercise/2

Giorgio Amati
Alessandro Ceci

Corso di dottorato in Ingegneria Aeronautica e Spaziale 2025
g.amati@cineca.it / g.amaticode@gmail.com
alessandro.ceci@uniroma1.it

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
- ✓ Any available Compiler
- ✓ Medium optimization level (`-O2`)

Homework/0

Loop order	Size	Fortran (MFLOPs)
`i,k,j`	512*512	221
`j,k,i`	512*512	6871
`i,k,j`	4096*4096	380
`j,k,i`	4096*4096	4170

- ✓ Compiler used: gcc, rel. 11.4.0
- ✓ Compiler option used: -O2
- ✓ HW used: AMD Ryzen 5 5625U

Homework/1

✓ Unrolling external loop

#unrolling	Size	Fortran	C
0	512*512	4042	-
2	512*512	5726	-
4	512*512	5726	-
0	4096*4096	2091	-
2	4096*4096	3334	-
4	4096*4096	6009	-

✓ Compiler used: **`nvfortran`**
✓ Compiler option used: **`-O2`**
✓ HW used: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz

Homework/2:

✓ Unrolling internal loop

#unrolling	Size	Fortran	C
0	512	4042	-
2	512*512	4908	-
4	512*512	5726	-
0	4096*4096	2091	-
2	4096*4096	2012	-
4	4096*4096	1921	-

✓ Compiler used: **`nvfortran`**
✓ Compiler option used: **`-O2`**
✓ HW used: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz

Homework/3: Fill the table

✓ Cache blocking: try a couple of blocking size…

#blocking	Size	Fortran	C
32	512*512	11453	
64	512*512	9817	-
128	512*512	6871	-
32	4096	6271	
64	4096*4096	5670	-
128	4096*4096	5393	-

✓ Compiler used: **`nvfortran`**

✓ Compiler option used: **`-O2`**

✓ HW used: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz

Homework/4: Fill the table

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

#unrolling	#blocking	Size	Fortran	C
`4x4x1`	`64`	4096*4096	17734	-
`4x4x1`	`128`	4096*4096	23869	-
`4x4x1`	`32`	8192*8192	15605	-
`4x4x1`	`128`	8192*8192	17731	-

✓ Compiler used: **`nvfortran`**
✓ Compiler option used: **`-O2`**
✓ HW used: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz

DIY

✓ in directory BLOCKING_ET_ALL

```
├── compile.fortran.sh
├── mm.0.F90
├── mm.1.F90
├── mm.2.F90
├── mm.3.F90
├── mm.4.F90
└── mod_tools.F90
```

The code….

```
do jj = 1, n, step
do kk = 1, n, step
do ii = 1, n, step
   do j = jj, jj+step-1, 4
   do k = kk, kk+step-1, 4
   do i = ii, ii+step-1
      temp0 = a(i,k+0)
      temp1 = a(i,k+1)
      temp2 = a(i,k+2)
      temp3 = a(i,k+3)
      c(i,j+0)=c(i,j+0)+temp0*b(k+0,j+0)+temp1*b(k+1,j+0)+temp2*b(k+2,j+0)+temp3*b(k+3,j+0)
      c(i,j+1)=c(i,j+1)+temp0*b(k+0,j+1)+temp1*b(k+1,j+1)+temp2*b(k+2,j+1)+temp3*b(k+3,j+1)
      c(i,j+2)=c(i,j+2)+temp0*b(k+0,j+2)+temp1*b(k+1,j+2)+temp2*b(k+2,j+2)+temp3*b(k+3,j+2)
      c(i,j+3)=c(i,j+3)+temp0*b(k+0,j+3)+temp1*b(k+1,j+3)+temp2*b(k+2,j+3)+temp3*b(k+3,j+3)
   enddo
   enddo
   enddo
enddo
enddo
enddo
```