



**16<sup>a</sup> Scuola  
Estiva di  
CALCOLO  
PARALLELO**

## Handbook of Complex Flags

Cristiano Calonaci - [c.calonaci@ Cineca.it](mailto:c.calonaci@ Cineca.it)  
Gruppo Supercalcolo - Dipartimento Sistemi e Tecnologie



1<sup>a</sup> edizione 9 - 20 luglio 2007 2<sup>a</sup> edizione 3 - 14 settembre 2007

Handbook of Complex Flags



**16<sup>a</sup> Scuola  
Estiva di  
CALCOLO  
PARALLELO**

## Pentium IV



## miscellanea

## g77/g95

- -c, -I, -L, -l
- -ffree-form                      free form
- -ffixed-form                    fixed form
- -ffixed-line-length-132    72    132 colonne
- -Dmacro                        definisce una macro per il  
                                      preprocessore (.F, .F90)
- -i8                               integer\*4    integer\*8 (solo g95)
- -r8                               real    double precision (solo g95)
- -pedantic                       warning almost all no standard  
                                      features



## miscellanea

## PGI

- -c, -I, -L, -l
- -module                        search per module files
- -Mextend                        72    132 colonne
- -byteswapio                    big/little endian unformatted I/O
- -Dmacro                        definisce una macro per il preprocessore  
                                      (.F, .F90)
- -g77libs                        g77.o    PGI main
- -pc 32|64|80                    precisione fp
- -i2,-i4,-i8                    cambia integer size
- -r4                               DOUBLE PRECISION    REAL
- -r8                               REAL    DOUBLE PRECISION
- -Mr8intrinsic                   REAL, COMPLEX    Doubles



## miscellanea

### INTEL

- `-c, -I, -L, -l`
- `-free | -fixed` free / fixed source format
- `-extend_source` 72 132 colonne
- `-convert ...` UNFORMAT big\_endian, cray, ibm, ...
- `-cpp` esegue sempre preprocessore
- `-Dmacro` definisce una macro per il preprocessore (`.F, .F90`)
- `-pc32|64|80` precisione fp
- `-integer_size 16|32|64` cambia integer size
- `-real_size 32|64|128` cambia real size



## how to align data

### g95/g77

- `-Malign-double` align variabili double precision



## how to align data

## PGI

- `-Mcache_align` align variabili escluse:  
*structures*, COMMON, ALLOCATABLE,  
*automatic array*
- `-Mdalign` align doubles in *structures* e COMMON
- `-Mlalign` align long long and INTEGER\*8 in  
*structures* e COMMON
- `-Mfunc32` align *functions*



## how to align data

## INTEL

- `-align` align variabili (quali ?)
- `-align commons` align (padding) COMMON 4 byte
- `-align dcommons` align (padding) COMMON 8 byte
- `-align records` align (padding) TYPE, RECORD,  
STRUCTURE
- `-align sequence` align (padding) SEQUENCE



## memory management

## g77/g95

- -fno-automatic      SAVE on all program unit (NO on COMMON blocks)
- -finit-local-zero    local variables inizializzate a 0

## PGI

- -M[no]save      [no]SAVE su tutte le local variables

## INTEL

- -auto\_scalar      scalar INTEGER, REAL, COMPLEX, LOGICAL = AUTOMATIC (on stack segment)
- -auto      all variables – SAVE = AUTOMATIC (on stack segment)
- -save      all variables – AUTOMATIC declared = SAVE (on data segment)



## how to set x87 precision

## g95/g77

- -ffloat-store    precision 80    64 by spilling (+ lento ma standard IEEE754, NON sempre eseguito)
- Settare bits in x87 FPU Control Word (vedi dietro)

## PGI

- -pc 32 | 64 | 80    (some) x87 FP precision

## INTEL

- -pc32 | 64 | 80    (some) x87 FP precision



## how to set x87 precision

```

g95/g77
setpc.c
#include <fpu_control.h>
void setpc_(void){
/* See fpu_control.h for details    Available precision
* _FPU_EXTENDED
* _FPU_DOUBLE      this example
* _FPU_SINGLE      */
fpu_control_t cw = (_FPU_DEFAULT & ~_FPU_EXTENDED) |
_FPU_DOUBLE;
_FPU_SETCW(cw); }

program namemain
dichiarazione variabili
call setpc()

gcc setpc.c -c; g77 namemain.f setpc.o

```



## trapping floating point exceptions

```

g95/g77
- Settare bits in x87 FPU Control Word (vedi dietro)

PGI
- -Ktrap=ovf,unf,denorm,divz,inv,inexact
                        floating exceptions:overflow, underflow,
                        denormalized, divbyzero, invalid, inexact
- -g -O0                trace FPE

INTEL
- -fpe0                 unf   0.0, other   ABORT
- -fpe1                 unf   0.0, other   INF, NaN e go on
- -fpe3                 no trap, migliori performance
- -g -O0 -traceback -fpstkchk          trace FPE

```



## trapping floating point exceptions

g95/g77

```

setfpe.c
#include <fpu_control.h>
void setfpe_(void){
/* See fpu_control.h for details    Available FPEs (_FPU_MASK_    )
* OM: Overflow mask
* UM: Underflow mask
* DM: Denormalized operand mask
* ZM: Zero-divide mask
* IM: Invalid operation mask
* PM: Precision (inexact result) mask*/ /* Only x87, NO SSE/SSE2 */
fpu_control_t cw = _FPU_DEFAULT & ~(_FPU_MASK_OM | _FPU_MASK_ZM |
_FPU_MASK_IM);
_FPU_SETCW(cw);}

program namemain
dichiarazione variabili
call setfpe()

```

gcc setfpe.c -c; g77 namemain.f setfpe.o -g -O0  
**limit coredumpsize unlimited**



## how to debug

g95/g77

- -O0	No ottimizzazioni
- -g	assembler      source
- -g3	+ info di -g
- -ggdb3	+ info x gdb
- -Wimplicit	implicit none
- -Wunused	Warning variabili non usate
- -Wuninitialized	Warning variabili non inizializzate (richiede -O)
- -Wsurprising	Warning costrutti surprising (compiler dependent)
- -W	Extra warning
- -fglobals	IPA diagnostic
- -fbounds-check	check bounds



## how to debug

## PGI

- |                 |                              |
|-----------------|------------------------------|
| - -O0           | No ottimizzazioni            |
| - -g            | assembler      source        |
| - -C            | check bounds                 |
| - -Kieee noieee | IEEE 754 conformance         |
|                 | uso fast math library and op |



## how to debug

## INTEL

- |                   |  |
|-------------------|--|
| - -O0             | No ottimizzazioni                                      |
| - -g              | assembler      source                                  |
| - -traceback      | trace info (altrimenti no info)                        |
| - -fpstkchk       | miglior tracing FPE                                    |
| - -check bounds   | check bounds   |
| - -fltconsistency | limitazione FP opt e ok precisione dichiarata          |
| - -mpl            | = -fltconsistency + "soft"                             |
| - -fp_port        | rounding FP results      user declaration dopo calcoli |
| - -prec_div       | divisioni + precise (NO /      * 1./ )                 |





## how to optimize

## g95/g77

- -O0   1   2   3	ottimizzazioni "generiche"
- -fforce-mem	Ottimizzazione loop
- -fforce-addr	Ottimizzazione loop
- -fstrength-reduce	Ottimizzazione loop
- -frerun-cse-after-loop	Common Subexpression Elimination
- -fexpensive-optimizations	Ottimizzazione misteriosa
- -fdelayed-branch	Ottimizzazione misteriosa
- -fschedule-insns	Ottimizzazione misteriosa
- -fschedule-insns2	Ottimizzazione misteriosa
- -fcaller-saves	Ottimizzazione misteriosa



## how to optimize

## g95/g77

- -funroll-loops	Unroll loop iterativi
- -funroll-all-loops	Unroll loop iterativi + do while
- -fmove-all-movables	Moving loop invariant index comp
- -freduce-all-givs	Moving loop invariant index comp
- -frerun-loop-opt	Run loop optimizer twice
- -funsafe-math-optimizations	Ottimizzazione math op
- -ffinite-math-only	Ottimizzazione math op
- -fno-trapping-math	Ottimizzazione math op
- -finline	Inlining
- -fprefetch-loop-arrays	Prefetching
- VEDI POI man gcc (sez. "Options That Control Optimization")	



## how to optimize

## PGI

- -O0|1|2|3 ottimizzazioni "generiche"
- -Minfo info sulle opt
- -Mneginfo why not opt ?
- -Munroll loop unrolling
- -Mipa InterProcedural Analysis



## how to optimize

## PGI

- -Mvect=..., ... loop opt
  - **cachesize:524288** **dimensione L2**
  - **[no]altcode [no]scalarizzazione loop**
  - **[no]assoc [no]opt associative sulle op**
  - **prefetch** **usa istruzioni PREFETCH se possibile**
  - **sse** **usa SSE, SSE2, PREFETCH se possibile**



## how to optimize

## INTEL

- -O0 | 2 | 3                      ottimizzazioni "generiche"
- -unroll                          loop unrolling
- -scalar\_rep                      scalar replacement in opt loop;  
   -O3
- -vec\_report0 | ... | 5           info vettorizzazione
- -opt\_report\_file *filename*      report opt su *filename*
- -opt\_report\_level min | med | max   livello opt



## how to optimize

## INTEL

- -complex\_limited\_range      opt su operazioni COMPLEX
- -ip                                IPA su singolo file
- -ipo                               IPA su multfile
- -Ob0 | 1 | 2                    inlining opt    -ip
- -assume [no]buffered\_io      buffering I/O
- -prefetch[-]                  abilita prefetch    -O3
- -static                         no shared libraries (.so)



## how to optimize

**SSE / SSE2**

variabili aligned naturally

**g77/g95**

- `-mcpu=?? -march=??` Usa CPU modello ??;  
                                  es. `-mcpu=pentium4 -march=pentium4`
- Pentium IV
- `-mfpmath=387|&sse` Usa come FPU x87 o/e SSE (& =  
                                  sperimentale) *solo per op scalari, no  
                                  vettorizzazione (gcc 3.4)*
- `-mmmx, -msse, -msse2, -m3dnow` Abilita MMX, SSE, SSE2,  
                                  3DNow *solo per op scalari, no vettorizzazione (gcc 3.4)*



## how to optimize

**SSE / SSE2**

variabili aligned naturally

**PGI**

- `-tp ??` Usa CPU modello ??; es. `-tp p7`  
                                  Pentium IV
- `-Mscalarsse` Usa (scalar ?) SSE/SSE2
- `-Mvect=sse` Vettorizza loop by SSE/SSE2



## how to optimize

**SSE / SSE2**

variabili aligned naturally

**INTEL**

- `-tpp?` Usa CPU modello ?;  
es. `-tpp7` Pentium IV
- `-ax?` Vettorizza/ottimizza x CPU modello ? e x86;  
es. `-axN` Pentium IV
- `-x?` Vettorizza/ottimizza x CPU modello ?;  
es. `-xN` Pentium IV



## how to optimize

**SSE / SSE2**

variabili aligned naturally

**PGI** (*`cpgi$g(global)` o `cpgi$1(oop next)`*)

- `ivdep, noeqvchk` NO data/EQUIVALENCE dependencies
- `[no]vector` [no]Vectorization
- `vintr` Vector intrinsics

**INTEL** (*`CDEC$`*)

- `PREFETCH var` Prefetch *var*
- `IVDEP` NO data dependencies
- `[NO]VECTOR ALWAYS` [`[UN]ALIGNED`]  
[NO]Vectorization, dati [un]aligned



## how to optimize

**FTZ /DAZ**

SSE / SSE2

**PGI**– `-Mflushz`

Set SSE in flush-to-zero mode

**INTEL**– `-ftz`

Set SSE in flush-to-zero mode

– `-xP`Set SSE in denormalized-are-zero  
mode

## how to use a debugger

**GNU**

- Compilare e linkare con `-g -O0`
- Run in debug mode
  - `gdb ./executable`
- Analizzare core
  - `limit coredumpsize unlimited`
  - `./executable core`
  - `gdb ./executable ./core`

## how to use a debugger

## GNU

- |                 |                    |                             |
|-----------------|--------------------|-----------------------------|
| - list          | "sourcename":@a,@b | sorgente sourcename         |
|                 |                    | dalla linea @aa @b          |
| - run           |                    | (re)run                     |
| - break         | "sourcename":@a    | stop at sourcename:@a       |
| - delete        |                    | rm breakpoints              |
| - c             |                    | continue from breakpoint    |
| - step num      |                    | esegui num istruzioni, into |
|                 | subroutines        |                             |
| - next num      |                    | esegui num istruzioni, over |
|                 | subroutines        |                             |
| - print varname |                    | mostra valore varname       |
| - where         |                    | traceback                   |
| - help, quit    |                    | help, quit                  |

## how to use a debugger

## PGI

- Compilare e linkare con `-g -O0`
- Run in debug mode
  - `pgdbg -text ./executable`
- Analizzare core
  - Sorry, non funziona ....



## how to use a debugger

## INTEL

- Compilare e linkare con `-g -O0 -traceback`
- Run in debug mode
  - `idb -gdb ./executable` (anche Fortran 90)
  - `gdb ./executable` (sembra compatibile, no F90 ?)
- Analizzare core
  - `setenv decfort_dump_flag TRUE`
  - `limit coredumpsize unlimited`
  - `./executable core`
  - `idb -gdb ./executable ./core` **riga morte**
  - `gdb ./executable ./core` **riga morte**



## how to use a debugger

## INTEL

- comandi = GNU
  - **stop at** **sostituisce break se non -gdb**
- + altri diversi comandi (man idb)





## how to use a profiler

### GNU/PGI/INTEL

- Compilare e linkare con `-ottimiz -pg`
- Run
  - `./executable gmon.out`
  - `gprof ./executable ./gmon.out report a stdout`
  - `report`
    - albero chiamate
    - distribuzione tempo nelle subroutine
- **Attenzione a link librerie (es GNU `-static`) !!!!**



## Power5



## miscellanea

- `-c, -I, -L, -l`
- `-qfixed=132`      72    132 colonne
- `-qfree`      free form
- `-qsuffix=f=X:cpp=Y`    file fortran = .X  
                             file preprocessore = .Y
- `-WF,Dmacro`      definisce una macro per il preprocessore
- `-qintsize=4 | 8`      integer size = 4 | 8 byte
- `-qrealsize=4 | 8`      real size = 4 | 8 byte



## how to address memory size

- `-q32`      Compilazione 32 bit    MAX 2 Gbyte indirizzabili
- `-q64`      Compilazione 64 bit    MAX 2<sup>64</sup> byte indirizzabili
- `-qwarn64`    Alcuni check x eventuali problemi con `-q64`
- `-bmaxdata:0x_num_ _num_` = limite data size in Hex
- `-bmaxstack:0x_num_ _num_` = limite stack size in Hex



## memory management

- `-qsave` local variables = SAVE (data segment) (default xlf)
- `-qnosave` local variables = AUTOMATIC (stack segment)  
(default xlf90)
- `-qinitauto=num` Inizializza a *num* le variabili non  
inizializzate



## how to debug

- `-qflttrap=ov:unf:zero:inv:inex:en` floating  
exceptions: overflow, underflow,  
divbyzero, invalid, inexact
- `-qinitauto=7FBFFFFFF` real non inizializzati = NaN
- `-qinitauto=7FF7FFFF` double precision non  
inizializzati = NaN
- `-C` check bounds
- `-qextchk` check parametri subroutine
- `-g -O0` trace error
- `-qsigtrap=xl__trcedump` abort + core + riga morte

## how to optimize

- `-qmaxmem=-1` Memoria infinita per “calcolo” ottimizzazioni
- `-O2` Opt “sicure” (es. scheduling istruzioni)
- `-O3` Opt “aggressive” (che possono modificare la semantica del codice, `-qstrict` per prevenire queste alterazioni)
- `-qhot` Opt sui loop  
`arraypad` | `noarraypad` auto padding  
`vector` | `novector` op vect vs. op scalar
- `-qcache=auto` Caratteristiche cache per `-qhot`
- `-qarch=pwr5` Genera eseguibile con set istruzioni del Power5
- `-qtune=pwr5` Tuning set istruzioni, scheduling, ... su Power5

## how to optimize

- `-qipa=...` Analisi interprocedurale  
`partition=large` zona grande da  
`analizzare`  
`level=0` | `1` | `2` aggressivita' IPA  
`inline=auto` inlining automatico  
`inline=sub1,sub2:inline=noauto` inlining  
 manuale di `sub1` e `sub2`
- `-qfloat`, `-qalias`, `-qessl`, ... altri flag (see *Xlf 9.0: IBM XL Fortran User Guide*)