



# PGI C/C++、Fortran编译器的使用

李会民

hmli@ustc.edu.cn

中国科学技术大学 超级计算中心

2015年4月16日



- 1 PGI C/C++ Fortran编译器简介
- 2 PGI编译器编译举例
  - PGI C/C++编译器编译举例
  - PGI Fortran编译器编译举例
- 3 PGI C/C++编译器重要编译选项
  - 一般选项
  - 优化选项
  - 调试选项
  - 预处理选项
  - 链接选项
  - C/C++语言选项
  - Fortran语言选项
  - 平台相关选项
- 4 调试
- 5 联系信息





# PGI C/C++ Fortran编译器简介

PGI C/C++ Fortran编译器是一种针对多种CPU与操作系统的高性能编译器，可用于开发复杂且要进行大量计算的程序。当前安装的版本为：

- 2014 v14.10，安装在 `/opt/pgi/linux86-64/14.10`
- 2010 v10.6，安装在 `/opt/pgi/linux86-64/10.6`

官方手册目录在其doc子目录下

编译命令：

- 编译C、C++源程序的命令分别：`pgcc`、`pgCC`
- 编译Fortran 77源程序的命令：`pgf77`
- 编译Fortran 90的源程序的命令：`pgf90`、`pgf901`、`pgf902`、`pgf90_ex`、`pgf95`和`pgfortran`
- 与NVIDIA CUDA库配合，可以编译Fortran-CUDA程序



# PGI C/C++ Fortran编译器简介

用户要使用，可以采用以下两种方式之一

- 运行 *mpi-selector-menu* 命令设置，需要重新登录才生效
- 在自己的 *~/.bashrc* 之类环境设置文件中添加以下代码设置：

```
PATH=/opt/pgi/linux86-64/14.10/bin:$PATH
MANPATH=$MANPATH:/opt/pgi/linux86-64/10.6/man
export PATH MANPATH
```

需重新登录或者source相应的设置文件才能生效



# PGI C/C++编译器编译举例

- 将C程序yourprog.c编译为可执行文件yourprog:  
*pgcc -o yourprog yourprog.c*
- 将C++程序yourprog.cpp编译为可执行文件yourprog:  
*pgCC -o yourprog yourprog.cpp*
- 将OpenMP指令并行的C程序yourprog-omp.c编译为可执行文件yourprog-omp:  
*pgcc -o yourprog-omp -mp yourprog.c*



- 将Fortran 77程序yourprog.for编译为可执行文件yourprog:  
*pgf77 -o yourprog yourprog.for*
- 将Fortran 90程序yourprog.f90静态编译为可执行文件yourprog:  
*pgf90 -o yourprog -static yourprog.f90*
- 将OpenMP指令并行的Fortran 90程序yourprog-omp.f90编译为可执行文件yourprog-omp:  
*pgf90 -o yourprog-omp -mp yourprog.f90*



- PGI编译器选项非常多，下面仅仅是列出一些本人认为常用的关于编译C程序的*pgcc*命令的重要选项
- 编译C++程序的*pgCC*命令有稍微不同，建议仔细查看PGI相关资料
- 建议仔细查看编译器手册中关于程序优化的部分，多加测试，选择适合自己程序的编译选项以提高性能





# 一般选项 I

- `-#`: 显示编译器、汇编器、链接器的调用信息。
- `-c`: 仅编译成对象文件（.o文件）。
- `-defaultoptions`和`-nodefaultoptions`: 是否使用默认选项，默认为使用。
- `-flags`: 显示所有可用的编译选项。
- `-help[=option]`: 显示帮助信息，*option*可以为`groups`、`asm`、`debug`、`language`、`linker`、`opt`、`other`、`overall`、`phase`、`phase`、`prepro`、`suffix`、`switch`、`target`和`variable`。
- `-Minform=level`: 控制编译时错误信息的显示级别。*level*可以为`fatal`、`file`、`severe`、`warn`、`inform`，默认为`-Minform=warn`。
- `-noswitcherror`: 显示警告信息后，忽略未知命令行参数并继续进行编译。默认显示错误信息并且终止编译。
- `-o file`: 指定生成的文件名。
- `-show`: 显示现有`pgcc`命令的配置信息。





- `-silent`: 不显示警告信息，与`-Minform=severe`等同。
- `-v`: 详细模式，在每个命令执行前显示其命令行。
- `-V`: 显示编译器版本信息。
- `-w`: 编译时不显示任何警告，只显示错误。





# 优化选项 I

- **-fast**: 编译时选择针对目标平台的普通优化选项。  
用 `pgcc -fast -help` 可以查看等价的开关。优化级别至少为O2，参看-O选项。
- **-fastsse**: 对支持SSE和SSE2指令的CPU（如Intel Xeon CPU）编译时选择针对目标平台的优化选项。用 `pgcc -fastsse -help` 可以查看等价的开关，优化级别至少为O2，参看-O选项。
- **-fpic**或**-fPIC**: 编译器生成地址无关代码，以便可用于生成共享对象文件（动态链接库）。
- **-Kpic**或**-KPIC**: 与**-fpic**或**-fPIC**相同，为了与其余编译器兼容。
- **-Minfo [=option[,option,...]]**: 显示有用信息到标准错误输出，选项可为all、autoinline、inline、ipa、loop或opt、mp、time或stat。
- **-Mipa[=option[,option,...]]**和**-Mnoipa**: 启用指定选项的过程间分析优化，默认为**-Mnoipa**。



## 优化选项 II

- `-Mneginfo=option[,option...]`: 使编译器显示为什么特定优化没有实现的信息。选项包括`concur`、`loop`和`all`。
- `-Mnopenmp`: 当使用`-mp`选项时, 忽略OpenMP并行指令。
- `-Mnosgimp`: 当使用`-mp`选项时, 忽略SGI并行指令。
- `-Mpfi`: 生成概要导向工具, 此时将会包含特殊代码收集运行时的统计信息以用于子序列编译。`-Mpfi`必须在链接时也得使用。当程序运行时, 会生成概要导向文件`pgfi.out`。
- `-Mpfo`: 启用概要导向优化, 此时必须在当前目录下有概要文件`pgfi.out`。
- `-Mprof[=option[,option, ...]]`: 设置性能功能概要选项。此选项可使得结果执行生成性能概要, 以便PGPROF性能概要器分析。
- `-mp[=option]`: 打开对源程序中的OpenMP并行指令的支持。
- `-O[level]`: 设置优化级别。level可为0、1、2、3、4, 其中4与3相同。
- `-pg`: 使用gprof风格的基于抽样的概要剖析。



- -g: 包含调试信息。





# 预处理选项 I

- **-C**: 预处理时保留C源文件中的注释。
- **-Dname[=def]**: 预处理时定义宏**name**为**def**。
- **-dD**: 打印源文件中已定义的宏及其值到标准输出。
- **-dI**: 打印预处理中包含的所有文件信息，含文件名和定义时的行号。
- **-dM**: 打印预处理时源文件已定义的宏及其值，含定义时的文件名和行号。
- **-dN**: 与**-dD**类似，但只打印源文件已定义的宏，而不打印宏值。
- **-E**: 预处理每个.c文件，将结果发送给标准输出，但不进行编译、汇编或链接等操作。
- **-Idir**: 指明头文件的搜索路径。
- **-M**: 打印**make**的依赖关系到标准输出。
- **-MD**: 打印**make**的依赖关系到文件**file.d**，其中**file**是编译文件的根名字。



- -MM: 打印make的依赖关系到标准输出，但忽略系统头文件。
- -MMD: 打印make的依赖关系到文件file.d，其中file是编译的文件的根名字，但忽略系统头文件。
- -P: 预处理每个文件，并保留每个file.c文件预处理后的结果到file.i。
- -Uname: 去除预处理中的任何name的初始定义。



# 链接选项

- **-Bdynamic**: 在运行时动态链接所需的库。
- **-Bstatic**: 静态链接所需的库。
- **-Bstatic\_pgi**: 动态链接系统库时静态链接PGI库。
- **-g77libs**: 允许链接GNU *g77*或*gcc*命令生成的库。
- **-lstring**: 指明所需链接的库名。如库为libxyz.a, 则可用-lxyz指定。
- **-Ldir**: 指明库的搜索路径。
- **-m**: 显示链接拓扑。
- **-Mrpath**和**-Mnorpath**: 默认为-rpath, 以给出包含PGI共享对象的路径。用-Mnorpath可以去除此路径。
- **-pgf77libs**: 链接时添加pgf77运行库, 以允许混合编程。
- **-r**: 生成可以重新链接的对象文件。
- **-Rdirectory**: 对共享对象文件总搜索directory目录。
- **-pgf90libs**: 链接时添加pgf90运行库, 以允许混合编程。
- **-shared**: 生成共享对象而不是可执行文件, 必须在编译每个对象文件时使用-fpic选项。
- **-soname**: 生成共享对象时, 用内在的DT\_SONAME代替指定



- **-B**: 源文件中允许C++风格的注释，指的是以//开始到行尾内容为注释。除非指定**-C**选项，否则这些注释被去除。
- **-c8x**或**-c89**: 对C源文件采用C89标准。
- **-c9x**或**-c99**: 对C源文件采用C99标准。







- `-byteswapio`或`-Mbyteswapio`: 对无格式Fortran数据文件在输入输出时从大端 (**big-endian**) 到小端 (**little-endian**) 交换比特, 或者相反。此选项可以用于读写Sun或SGI等系统中的无格式的Fortran数据文件。
- `-i2`: 将INTEGERS变量按照2比特处理。
- `-i4`: 将INTEGERS变量按照4比特处理。
- `-i8`: 将默认的INTEGERS和LOGICAL变量按照4比特处理。
- `-i8storage`: 对INTEGERS和LOGICAL变量分配8比特。
- `-Mallocatable[=95|03]`: 按照Fortran 95或2003标准分配数组。
- `-Mbackslash`和`-Mnbackslash`: 将反斜线(\)当作正常字符 (非转义符) 处理, 默认为`-Mnbackslash`。 `-Mnbackslash`导致标准的C反斜线转义序列在引号包含的字符串中重新解析。 `-Mbackslash`则导致反斜线被认为和其它字符一样。
- `-Mextend`: 设置源代码的行宽为132列。



## Fortran语言选项 II

- `-Mfixed`、`-Mnofree`和`-Mnofreeform`: 强制对源文件按照固定格式进行语法分析, 默认.f或.F文件被认为固定格式。
- `-Mfree`和`-Mfreeform`: 强制对源文件按照自由格式进行语法分析, 默认.f90、.F90、.f95或.F95文件被认为自由格式。
- `-Mi4`和`-Mnoi4`: 将INTEGER看作INTEGER\*4。  
`-Mnoi4`将INTEGER看作INTEGER\*2。
- `-Mnomain`: 当链接时, 不包含调用Fortran主程序的对象文件。
- `-Mr8`和`-Mnor8`: 将REAL看作DOUBLE PRECISION, 将实(REAL)常数看作双精度(DOUBLE PRECISION)常数。默认为否。
- `-Mr8intrinsic` [`=float`]和`-Mnor8intrinsic`: 将CMPLX看作DCMPLX, 将REAL看作DBLE。添加float选项时, 将FLOAT看作DBLE。
- `-Msave`和`-Mnosave`: 是否将所有局部变量添加SAVE声明, 默认为否。



- `-Mupcase`和`-Mnoupcase`: 是否保留名字的大小写。`-Mnoupcase`导致所有名字转换成小写。注意, 如果使用`-Mupcase`, 那么变量名`X`与变量名`x`不同, 并且关键字必须为小写。
- `-Mcray=pointer`: 支持Cray指针扩展。
- `-module directory`: 指定编译时保存生成的模块文件的目录。
- `-r4`: 将`DOUBLE PRECISION`变量看作`REAL`。
- `-r8`: 将`REAL`变量看作`DOUBLE PRECISION`。



# 平台相关选项

- **-Kieee**和**-Knoieee**: 浮点操作是否严格按照IEEE 754标准。使用**-Kieee**时一些优化处理将被禁止, 并且使用更精确的数值库。默认为**-Knoieee**, 将使用更快的但精确性低的方式。
- **-Ktrap=[option,[option]...]**: 控制异常发生时CPU的操作。选项可为- **-Msecond\_underscore**和**-Mnosecond\_underscore**: 是否对已有\_的Fortran函数名添加第二个\_。与**g77**编译命令兼容时使用, 因为**g77**默认符号后添加第二个\_。
- **-mcmmodel=small|medium**: 使内存模型是否限制对象小于2GB(small)或允许数据块大于2GB(medium)。medium时暗含**-Mlarge\_arrays**选项。
- **-tp target**: target可以为nehalem-64等, 默认与编译时的平台一致。



# debug调试

调试命令为pgdbg, 基本语法为:

- `pgdbg [ _pgdbg_options ... _ ] [ _target [ _target_args ... _ ] _ ]`
- `mpirun [ _mpi_options ... _ ] -dbg=pgdbg target [ _target_args ... _ ]`

手册参见: <http://scc.ustc.edu.cn/zlsc/tc4600/pgi/pgdbg14ug.pdf>





- 中国科大超算中心:
  - 办公室: 科大东区新图书馆一楼东侧126室
  - 电话: 0551-63602248
  - 信箱: [sccadmin@ustc.edu.cn](mailto:sccadmin@ustc.edu.cn)
  - 主页: <http://scc.ustc.edu.cn>
- 李会民:
  - 电话: 0551-63600316
  - 信箱: [hml@ustc.edu.cn](mailto:hml@ustc.edu.cn)
  - 主页: <http://hml.ustc.edu.cn>