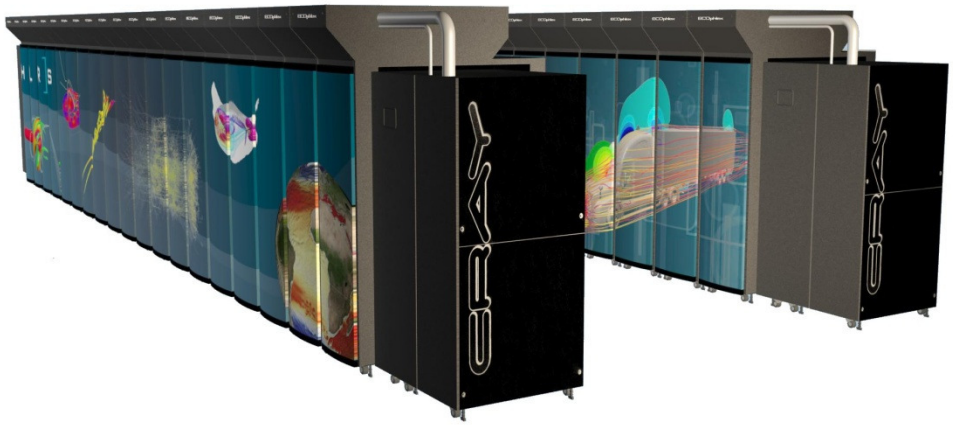
# The Cray Programming Environment-1



Charles Henriet

CRAY THE SUPERCOMPUTER COMPANY

HLRS

# Agenda

- Programming Environment Overview
  - Modules
- Compilers
  - PGI, Cray, GNU(, Intel, Pathscale)

# Cray XE6 programming environment overview

- Several compilers available
  - PGI, Cray, GNU(, Intel, Pathscale)
- Optimized libraries
  - 64 bit AMD Core Math library (ACML)
  - Scilib: Level 1,2,3 of BLAS, LAPACK, FFT, ScaLAPACK, BLACS
  - FFTW, netCDF, HDF5, PETSc
  - MPI-2 message passing library for communication between nodes
  - SHMEM one-sided communication library
  - PGAS : CAF and UPC
- Aprun command to launch jobs; similar to mpirun command
- Torque batch system
- Performance tools: CrayPat, Apprentice2
- Debugger : ddt

# Cray XE6 programming environment is SIMPLE

- Editing and compiling a program

```
$ vi mysrc.f90
$ ftn -o myexe mysrc.f90
```

- Creating a  batch job file

```
#PBS -N myjob
#PBS -lmppwidth=240
aprun -n 240 ./myexe
```

- Running  a job

```
$ qsub myjob.job
```

# The module tool on the Cray XE6

- How can we get appropriate Compiler and Libraries to work with?
- module tool used on XE6 to handle different versions of packages (compiler, tools,...):

  **e.g.: module load compiler1**

  **e.g.: module swap compiler1 compiler2**

  **e.g.: module load perftools**
- taking care of changing of PATH, MANPATH, LM_LICENSE_FILE,.... environment.
- users should not set those environment variable in their shell startup files, makefiles,....
- keep things flexible to other package versions

# Cray XE6 PE: module list

```
hpcander@xe601:~> module list
Currently Loaded Modulefiles:
  1) modules
  2) nodestat/2.2-1.0301.22648.3.4.gem
  3) sdb/1.0-1.0301.22744.3.46.gem
  4) MySQL/5.0.64-1.0301.2899.20.2.gem
  5) lustre-cray_gem_s/1.8.2_2.6.27.48_0.1.1_1.0301.5522.8.1-1.0301.23669.3.42
  6) udreg/1.3-1.0301.2236.3.7.gem
  7) ugni/2.0-1.0301.2365.3.7.gem
  8) gni-headers/2.0-1.0301.2497.4.1.gem
  9) dmapp/2.2-1.0301.2594.5.1.gem
 10) xpmem/0.1-2.0301.22550.3.7.gem
 11) Base-opts/1.0.2-1.0301.21771.3.3.gem
 12) xtpe-network-gemini
 13) pgi/10.9.0
 14) xt-libsci/10.4.8
 15) pmi/1.0-1.0000.7901.22.1.gem
 16) xt-mpt/5.1.1
 17) xt-asyncpe/4.4
 18) PrgEnv-pgi/3.1.37E
 19) moab/5.4.1
 20) torque/2.5.2
 21) ws/1.0
hpcander@xe601:~>
```

# Cray XE6 PE: module avail

```
> module avail cce

cce/7.2.4           cce/7.2.5           cce/7.2.6           cce/7.2.7
cce/7.2.8(default)  cce/7.3.0.145
```

(output from a Cray internal machine)

- module avail : will list all modules available

# Useful module commands

- Basic: load PGI compiler and Magny-Cours specific

  **module  load PrgEnv-pgi**

  **module load xtpe-mc8**

- Change environment (GNU)

  **module  swap  PrgEnv-pgi  PrgEnv-gnu**

- Load Cray environment and use a specific version of the compiler

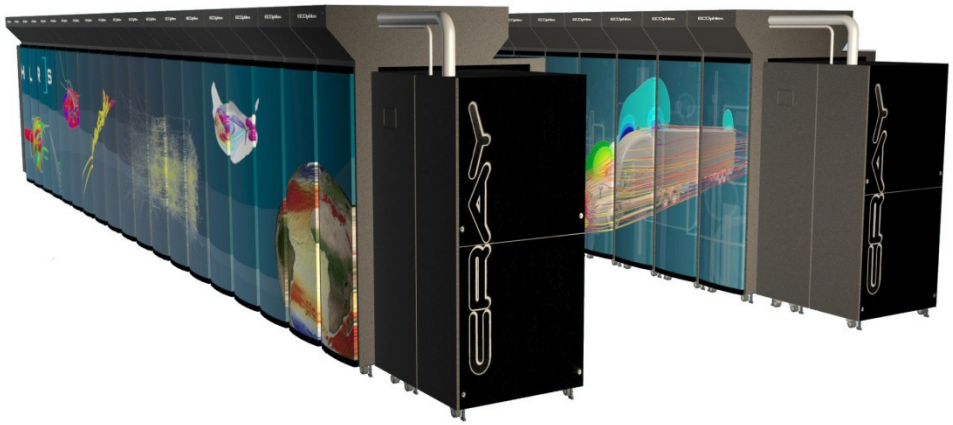
  **module swap PrgEnv-pgi  PrgEnv-cray**

  **module swap cce cce/7.3.0.145**

- Only load the MPICH2 environment

  **module unload xt-mpt**

  **module load xt-mpich2**



# Compilers

# Compiler drivers to create CLE executables

- When the PrgEnv is loaded the compiler drivers are also loaded
  - the compiler drivers also take care of loading appropriate libraries (-lmpich, -lsci, -lacml, -lpapi)

- Available drivers (also for linking of MPI applications):
  - Fortran 90/95 programs: ftn
  - Fortran 77 programs: f77
  - C programs: cc
  - C++ programs: CC

- Cross compiling environment
  - Compiling on a Linux service node
  - Generating an executable for a CLE compute node
  - Do not use standard compler names (pgf90, gcc, mpicc,...) unless you want a Linux executable for the service node

# Getting good compiler options and best libraries

- Load a specific module to produce code optimized for the AMD Magny-Cours processor
  module load xtpe-mc8
- Set important performance compiler flags (arch, ...)
- Loads the best performance libs

# PGI programming environment (PrgEnv-pgi)

- **Overall Options**
  - -Mlist creates a listing file
  - -Minfo info about optimizations performed
  - -Mneginfo why certain optimizations are not performed
- **Preprocessor Options**
  - -Mpreprocess runs the preprocessor on Fortran files
- **Optimisation Options**
  - -fast chooses generally optimal flags for the target platform
  - -Mipa=fast,inline Inter Procedural Analysis
  - -Minline=levels:n number of levels of inlining

# PGI programming environment

- **Language Options**
  - -Mfree process Fortran source using freeform specifications
  - -Mnomain useful for using the ftn driver to link programs with the main program written in C or C++ and one or more subroutines written in Fortran
  - -i8, -r8 treat INTEGER and REAL variables as 64-bit
- **Parallelization Options**
  - -mp recognize OpenMP directives
  - -Mconcur automatic parallelization

man pages: man pgf90, man pgcc, man pgCC

quick ref : pgf90 –help,pgcc –help , pgCC-help

PGI User's Guide (Chapter 2) http://www.pgroup.com/doc/pgiug.pdf

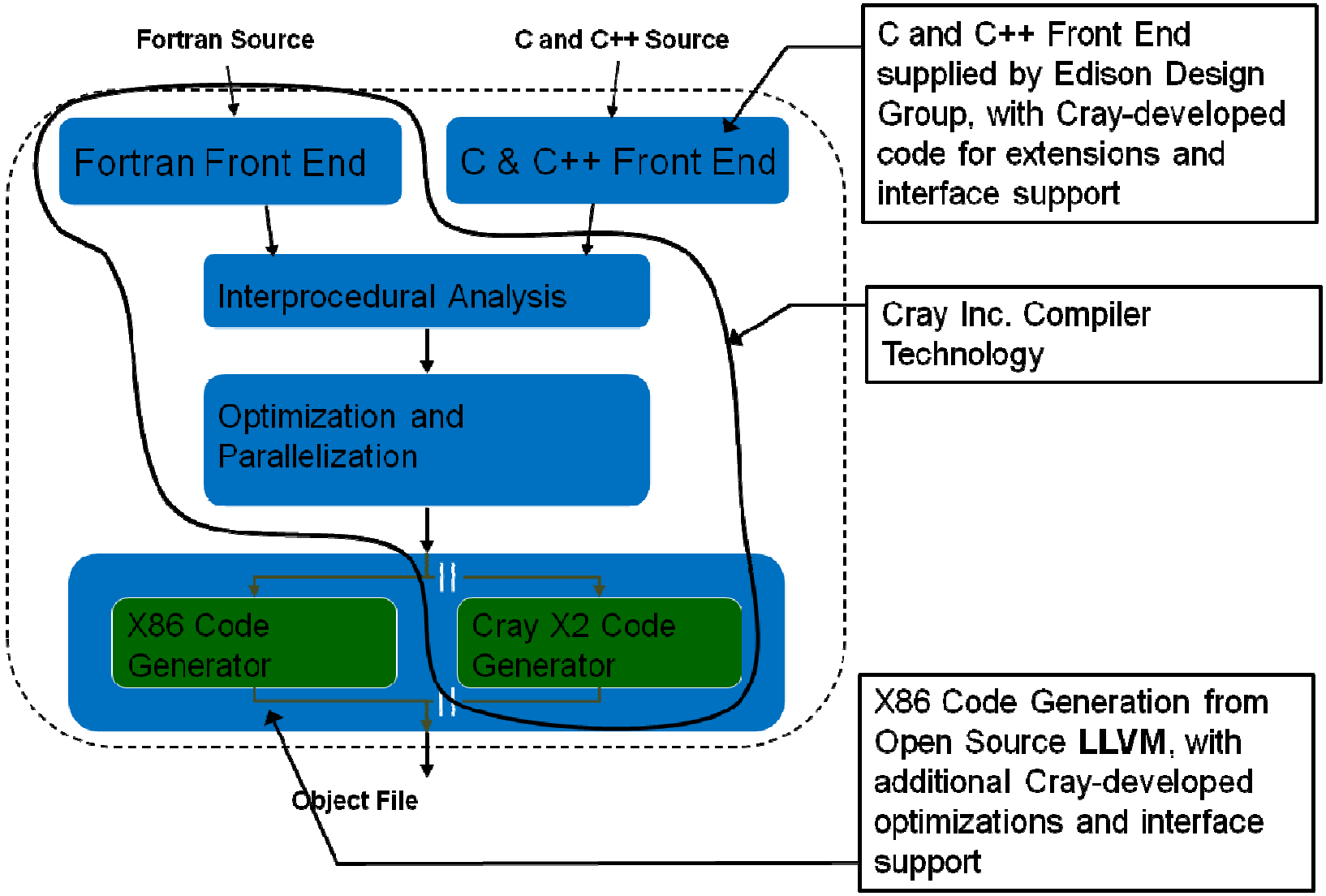
# Performing Endian Conversion on Cray XT Series Systems

- **PGI Compiler**
  - to read big-endian and write little-endian binary files
    - % ftn -Mbyteswapio -c readin_data.f90
      % ftn -c writeout_data.f90
      % ftn readin_data.o writeout_data.o -o *my_code. exe*

# Cray Compiler Environment (CCE): (PrgEnv-cray)

- In 2008 decided to move forward with Cray X86 compiler
  - Vector is back ? (SSE, AVX…)
  - CCE 7.3 is default CCE on HLRS XE6
- Still young but interesting…
  - FORTRAN 2008 Coarray support
  - Strongly supported by Cray

# Technology Sources



# Cray compiler flags

- **Overall Options**
  - -ra creates a listing file with optimization info
  - -rm produces a source listing with loopmark information
- **Preprocessor Options**
  - -eZ runs the preprocessor on Fortran files
  - -F enables macro expansion throughout the source file
- **Optimisation Options**
  - -O2 optimal flags [ enabled by default ]
  - -O3 aggressive optimization
  - -O ipa<n> inlining, n=0-5

# Cray compiler flags

- **Language Options**
  - -f free  process Fortran source using freeform
  - -s real64  treat REAL variables as 64-bit
  - -s integer64  treat INTEGER variables as 64-bit

- **Parallelization Options**
  - -O omp  Recognize OpenMP directives [default ]
  - -O thread<n>  n=0-3, aggressive parallelization, default n=2

=> man crayftn

http://docs.cray.com/cgi-bin/craydoc.cgi?mode=View;id=S-3901-71;idx=books_search;this_sort=;q=3901;type=books;title=Cray%20Fortran%20Reference%20Manual

# Performing Endian Conversion on Cray XT Series Systems

- **Cray Compiling Environment (CCE)**
  - -hbyteswapio: forces byte-swapping of all input and output files for direct and sequential unformatted I/O

# Cray programming environment: assign

- Assigns options for library file open processing

  **assign [assign options] assign_object**

- **Interesting assign options**
  - -R — removes all assign options for assign_object
  - -N <numcon> — specifies foreign numeric conversion

    swap_endian — the endianess of data is swapped during unformatted input and output.
- **assign object** — used to specify the object of assign options
  - f:<filename> — applies to filename
  - u:<unit> — applies to Fortran unit number
  - g:su — applies to all Fortran sequential unform. files

# How to handle byte-swapped files with CCE

- Explicit usage of assign
  - Can control which files are byte-swapped

```
export FILENV=.assign
assign -R
assign -N swap_endian f:aof
aprun a.out
```

- Link the application with –hbyteswapio
  - This is equivalent to set

```
assign -N swap_endian g:su ←all sequential unformatted
assign -N swap_endian g:du <-all direct unformatted
```

- „man assign“ (when PrgEnv-cray loaded)

# Cray Opteron Compiler: Directives

- **Cray compiler supports a full and growing set of directives and pragmas**

!dir$ concurrent

!dir$ ivdep

!dir$ interchange

!dir$ unroll

!dir$ loop_info [max_trips] [cache_na] ... Many more

!dir$ blockable

=> man intro_directives, man intro_pragmas,man loop_info

# Loopmark/Compiler Feedback

- ftn –rm … or cc –hlist=m …
- Compiler can generate an '.lst' with annotated listing of your source code with letter indicating important optimizations

# Example: Cray loopmark messages

```
29.  b-------<      do i3=2,n3-1
30.  b b-----<        do i2=2,n2-1
31.  b b Vr--<          do i1=1,n1
32.  b b Vr              u1(i1) = u(i1,i2-1,i3) +u(i1,i2+1,i3)
33.  b b Vr       >              + u(i1,i2,i3-1) + u(i1,i2,i3+1)
34.  b b Vr              u2(i1) = u(i1,i2-1,i3-1) u(i1,i2+1,i3-1)
35.  b b Vr       >              + u(i1,i2-1,i3+1) + u(i1,i2+1,i3+1)
36.  b b Vr-->          enddo
37.  b b Vr--<          do i1=2,n1-1
38.  b b Vr              r(i1,i2,i3) = v(i1,i2,i3)
39.  b b Vr       >            - a(0) * u(i1,i2,i3)
40.  b b Vr       >          - a(2) * ( u2(i1) + u1(i1-1) + u1(i1+1) )
41.  b b Vr       >                    - a(3) * ( u2(i1-1) + u2(i1+1) )
42.  b b Vr-->          enddo
43.  b b------>       enddo
44.  b------->      enddo
```

*ftn-6289 ftn: VECTOR File = resid.f, Line = 29*

*A loop starting at **line 29 was not vectorized** because a recurrence was found on "U1" between lines 32 and 38.*

*ftn-6049 ftn: SCALAR File = resid.f, Line = 29*

*A loop starting **at line 29 was blocked with block size 4.***

*ftn-6289 ftn: VECTOR File = resid.f, Line = 30*

*A loop starting at line 30 was not vectorized because a recurrence was found on "U1" between lines 32 and 38.*

*ftn-6049 ftn: SCALAR File = resid.f, Line = 30*

*A loop starting at line 30 was blocked with block size 4.*

*ftn-6005 ftn: SCALAR File = resid.f, Line = 31*

*A loop starting at **line 31 was unrolled 4 times.***

*ftn-6204 ftn: VECTOR File = resid.f, Line = 31*

*A loop starting at **line 31 was vectorized.***

*ftn-6005 ftn: SCALAR File = resid.f, Line = 37*

*A loop starting at line 37 was unrolled 4 times.*

*ftn-6204 ftn: VECTOR File = resid.f, Line = 37*

*A loop starting at **line 37 was vectorized.***

# CCE and OpenMP

- OpenMP is **ON** by default
  - -O [no]omp is identical to -h [no]omp option
  - -O omp is default : enable compiler recognition of OpenMPdirectives
  - Using -O noomp
    - is similar to the –O thread0 option : it disables OpenMP
    - but unlike -O thread0 it does not affect autothreading.

**If you do not want to use OpenMP and have OMP directives in the code, make sure to make a run with OpenMP shut off at compile time**

# CCE and Autothreading

- .Autothreading is NOT ON by default;
  - -h[no]autothread: hautothread to turn on
  - -O threadn: Control the compilation and optimization of OpenMp directives, where n is a value from 0 to 3
  - Default is n=2 :OpenMP parallel regions are subjected to some optimizations; that is, some parallel region expansion. Parallel region expansion is an optimization that merges two adjacent parallel regions in a compilation unit into a single parallel region.

# Cray programming environment: explain

- Displays the explanation for an error message: compile, run time
- Compiler error

```
cft90: llvm/lib/VMCore/Instructions.cpp:328: void
   llvm::CallInst::init(llvm::Value*, llvm::Value* const*, unsigned int):
Assertion `(i >= FTy->getNumParams() || FTy->getParamType(i) == Params[i]-
   >getType()) && "Calling a function with a bad signature!"' failed.
ftn-2116 crayftn: INTERNAL

stefan> explain ftn-2116
Internal : "Program" was terminated due to receipt of signal signal.
The process spawned by the command to run the specified program was
terminated by a system signal.  See the signal(2) man page for more
information about this signal.
This message does not indicate a problem with your code, and you may be able
to change your program so that this error does not occur.
Please notify your product support representative with the text of this
message, the source code being compiled, the version of the compiler, and the
Command line options in effect.
```

# Cray programming environment: explain

- I/O runtime error

```
Open IOSTAT=5016

stefan> explain lib-5016
An EOF or EOD has been encountered unexpectedly.
The file terminated unexpectedly, perhaps without the proper end-of-file
record or end-of-data control information.  The file specification may be
incorrect, or the file may be corrupted.
Ensure that the file specification is correct.  If the file is corrupted,
create the data again, if possible.
See the man pages for assign(1) and asgcmd(1).
The error class is UNRECOVERABLE (issued by the run time library).
```

# Pathscale programming env (PrgEnv-pathscale)

- **Overall Options**
  - -LIST:=ON — creates a listing file
  - -LNO:simd_verbose=ON — info about vectorizations
- **Preprocessor Options**
  - -cpp -ftpp — runs the preprocessor on C or FORTRAN files
- **Optimisation Options**
  - -O3 –OPT:Ofast — chooses generally optimal flags
  - -Ofast — aggressive optimization
  - -ipa — Inter Procedural Analysis

# Pathscale programming environment

- **Language Options**
  - -freeform process Fortran source using free form specifications
  - -i8, -r8 treat INTEGER and REAL variables as 64-bit
  - -byteswapio big-endian files in Fortran
- **Parallelization Options**
  - -mp recognize OpenMP directives
  - -apo automatic parallelization

man pages: eko, pathf90, pathcc

# Performing Endian Conversion on Cray XT Series Systems: Pathscale Compiler

- Pathscale uses "-byteswapio“
- a Fortan I/O library that allows defining I/O processing directives for individual files
- Files must be declared by the *assign* command before execution
  - -N be — Declare the file big-endian.
  - u:unit — Identify a file by Fortran unit number.
  - f:filename — Identify a file by specific filename.
  - g:du — Identify all direct-access unformatted files.
  - g:su — Identify all sequential-access unformatted files.
  - g:all — Identify all files.
  - -F f77.mips — For sequential-access files with F77 (big-endian) record headers.

# Performing Endian Conversion on Cray XT Series Systems: Pathscale Compiler

- If file "big_data.dat", is an unformatted, direct-access big-endian file:

- % setenv FILENV $HOMEDIR/.assign_inf
- % assign -N be f: big_data.dat

  - -N be Declare the file big-endian.
  - f:filename Identify a file by specific filename

# Performing Endian Conversion on Cray XT Series Systems- Pathscale Compiler :Batch script

- batch script commands necessary to read or write to a direct-access unformatted big-endian file named "direct_be.dat" and a sequential-access unformatted big-endian file named "sequ_be.dat

  - module swap PrgEnv-pgi PrgEnv-pathscale
  - setenv FILENV $HOMEDIR/.assign_info
  - assign -N be f: direct_be.dat
  - assign -N be -F f77.mips f: sequ_be.dat
  - assign –V (verify the assign declarations exist in the .assign_info file)
  - aprun –n zzz  ./a.out
  - assign -R (remove the assign declarations from the .assign_info file)

# Cray XE compilers

| Feature | PGI | Pathscale | Cray |
|---|---|---|---|
| Listing | -Mlist | -LIST:=ON | -ra |
| Diagnostic | -Minfo -Mneginfo | -LNO:simd_verbose=ON | (produced by -ra) |
| Free format | -Mfree | -freeform | -f free |
| Preprocessing | -Mpreprocess | -cpp -ftpp | -eZ -F |
| Suggested Optimization | -fast | -O3 –OPT:Ofast | (default) |
| Aggressive Optimization | -Mipa=fast,inline | -Ofast | -O3, fp3 |
| Variables size | -r8 –i8 | -r8 -i8 | -s real64 –s integer64 |
| Byte swap | -byteswapio | -byteswapio | -h byteswapio |
| OpenMP recognition | -mp=nonuma | -mp | (default) |
| Automatic parallelization | -Mconcur | -apo | -h autothread |

# Other programming environments

- GNU (PrgEnv-gnu)
  - Suggested options: -O3 –ffast-math –funroll-loops
  - Compiler feedback: -ftree-vectorize -verbose=2
  - OpenMP: -fopenmp
  - Man pages: gcc, gfortran, g++

- Intel (PrgEnv-intel)
  - Suggested options: -O3
  - Aggressive options: -ffast-math -funroll-loops -msse3 -ftree-vectorize
  - OpenMP: -openmp=on
    Careful : An extra control thread is spawn: issues when pinning threads to cores. Try aprun –cc [none|numa_node] instead of –cc cpu
  - Man pages: ifort, icc

# End of PART_1