Compilers

On UD HPC Community Clusters

Anita Schwartz
Client Support & Services

Compilers on Mills and Farber

- The GNU Compiler Collection (GCC)
  https://gcc.gnu.org/

- Intel
  https://software.intel.com/en-us/intel-compilers

- PGI
  http://www.pgroup.com/

PGI® Compilers & Tools

NVIDIA

OpenACC
Directives for Accelerators

What compiler should I use?

Factors to consider:

- Flexibility
- Portability
- Efficiency
- Performance

And what do you want to compile? This answer will likely depend on the language (Fortran, C, C++, ?), architecture (Intel, AMD, GPU,MIC), problem domain, testing and acceptability, serial or parallel, etc.

> *No one compiler is necessarily the best choice for everything*
> *and in some situations only one compiler may be the only choice.*

GCC

- Highly desirable for C and C++, not so much for Fortran
- Free and open source
- Works well on a variety of native and cross platforms - very portable
- Large community with active development
- Extensive development environment including runtime libraries, testsuites, etc
- Good place to start

Intel

- Great fit for Fortran (high level of optimization by default), excellent C and C++
- Not free
- Optimized code to take advantage of core count and vector register width in Intel® Xeon® processors, Intel® Xeon Phi™ coprocessors and compatible processors.
- A toolset that simplifies creating fast, reliable parallel code including new data analytics library.
- Tuned for Intel architecture
- HPC focused

PGI

- Very good Fortran and C, pretty good C++
- Not free
- Unified Binary for Intel and AMD x64 Processors+GPU simplifies cross-platform support by combining into a single executable file, code sequences optimized for multi-core x64 processor families from Intel and AMD and GPU accelerators from NVIDIA.
- Compiler-of-choice among many popular performance-critical applications used in the fields of geophysical modeling, mechanical engineering, computational chemistry, weather forecasting, and high-energy physics.
- HPC focused

How do I use the compilers?

VALET = **V**ALET **A**utomates **L**inux **E**nvironment **T**asks*

- Compiler versions: `vpkg_versions` *compiler*

```
$ vpkg_versions gcc
```

or

```
$ vpkg_versions intel
```

or

```
$ vpkg_versions pgi
```

*similar to `modules` on other HPC clusters

VALET

- Select the default version of a compiler*: `vpkg_require` *compiler* or `vpkg_devrequire` *compiler*

```
$ vpkg_devrequire gcc
```

or

```
$ vpkg_devrequire intel
```

or

```
$ vpkg_devequire pgi
```

Note: Both add selected *compiler* configuration to your environment; `vpkg_devrequire` also includes setup of the LDFLAGS and CPPFLAGS variables with library/header paths for the selected *compiler*.

Compiler commands

Compiler	C	Fortran
GCC	`gcc`	`gfortran`
Intel	`icc`	`ifort`
PGI	`pgcc`	`pgf90`

Help: man*

Compiler	Fortran	C
GCC	`man gfortran`	`man gcc`
Intel	`man ifort`	`man icc`
PGI	`man pgf90`	`man pgcc`

*Only works on the head node

Help: command line option

Compiler	Fortran	C
GCC	`gfortran --help`	`gcc --help`
Intel	`ifort -help`	`icc -help`
PGI	`pgf90 -help`	`pgcc -help`

GCC example

```
$ vpkg_versions gcc

Available versions in package (* = default version):

[/opt/shared/valet/2.0.1/etc/gcc.vpkg_xml]
gcc      GNU Compiler Suite
  4.8    alias to gcc/4.8.3
  4.8.3   Version 4.8.3
  4.9    alias to gcc/4.9.3
  4.9.3   Version 4.9.3
  6      alias to gcc/6.2
  6.2.0   Version 6.2.0
  6.2    alias to gcc/6.2.0
* system  OS-default GCC installation

$ vpkg_devrequire gcc/4.9
Adding package `gcc/4.9.3` to your environment
```

Intel example

```
$ vpkg_versions intel
Available versions in package (* = default version):

[/opt/shared/valet/2.0.1/etc/intel.vpkg_json]
intel                Intel Compiler Suite
  12.1               alias to intel/2011-sp1.13.367
  13.0               alias to intel/2013.1.117
  14.0.3             alias to intel/2013-sp1.3.174
  2011-sp1.13.367  Version 2011 (SP1.13.367) for x86-64
  2013               alias to intel/2013-sp1.3.174
  2013-sp1.3.174   Version 2013 (SP1.3.174) for x86-64
  2013.1.117         Version 2013 (1.117) for x86-64
* 2015               alias to intel/2015.3.187
  2015.0.090         Version 2015 (first release) for x86-64
  2015.3.187         Version 2015 (update 3) for x86-64
  2016               alias to intel/2016.2.062
  2016.1.150         Version 2016 (update 1) for x86-64
  2016.2.062         Version 2016 (update 2) for x86-64

$ vpkg_devrequire intel/2016
Adding package `intel/2016.2.062` to your environment
```

PGI example

```
$ vpkg_versions pgi
Available versions in package (* = default version):

[/opt/shared/valet/2.0.1/etc/pgi.vpkg_json]
pgi   Portland Group Compiler Suite
* 14  alias to pgi/14.10
  14.10  Version 14.10
  14.7   Version 14.7
  15  alias to pgi/15.7
  15.1   Version 15.1
  15.10  Version 15.10 (w/ Accelerator)
  15.4   Version 15.4
  15.7   Version 15.7
  15.9   Version 15.9 (w/ Accelerator)
  16  alias to pgi/16.1
  16.1   Version 16.1 (w/ Accelerator)

$ vpkg_devrequire pgi/16
Adding dependency `gcc/4.9.3` to your environment
WARNING: The Portland compiler suite is not officially supported on Farber.
WARNING: It has been made available by popular request.
Adding package `pgi/16.1` to your environment
```

Optimization and accelerator options

Compiler	Optimization	Phi	GPU	OpenMP*
GCC	`-O3 -ffast-math`			`-fopenmp`
Intel		`-mmic`		`-qopenmp or -openmp`
PGI	`-fast`		`-acc -ta=nvidia`	`-mp`

*Depends on the version of the compiler; `-openmp` for older versions

- Farber is based on Intel (code name “Ivy Bridge”): See Intel 64 and IA-32 Architectures Optimization Reference Manual and Intel Math Kernel Library Cookbook.
- Mills is based on AMD Opteron 6200 Series processors (code name “Interlagos”): Compiler Options Quick Reference Guide

MPI wrapper compiler commands and help

- GCC

```
$ vpkg_devrequire openmpi/1.8.8-gcc-4.9.3
```

Fortran*	**C**
`mpifort --help`	`mpicc --help`
`mpifort --help=optimizers`	`mpicc --help=optimizers`
`man mpifort`	`man mpicc`

*`mpif90` for openmpi versions older than 1.7; `man` only works on the head node

MPI wrapper compiler commands and help

- Intel

```
$ vpkg_devrequire openmpi/1.10.2-intel64-2016
```

Fortran*	**C**
`mpifort -help`	`mpicc -help`
`mpifort -help=opt`	`mpicc -help=opt`
`man mpifort`	`man mpicc`

*`mpif90` for openmpi versions older than 1.7; `man` only works on the head node

MPI wrapper compiler flags

The Open MPI Team **strongly** recommends using the wrapper compilers to compile and link MPI applications.

FAQ: Compiling MPI applications
https://www.open-mpi.org/faq/?category=mpi-apps

If you need to know the options to compile independent of the wrappers use

```
# Show the flags necessary to compile MPI C applications
$ mpicc --showme:compile

# Show the flags necessary to link MPI C applications
$ mpicc --showme:link
```

Hands-on exercises

Laplace examples from XSEDE HPC Workshops documented for Farber

http://docs.hpc.udel.edu/software/tutorial/laplace

- Serial
- OpenMP
- MPI
- OpenACC

Questions and Open Forum