PGI® COMPILERS & TOOLS

RELEASE NOTES FOR X86-64 CPUS

Version 2017

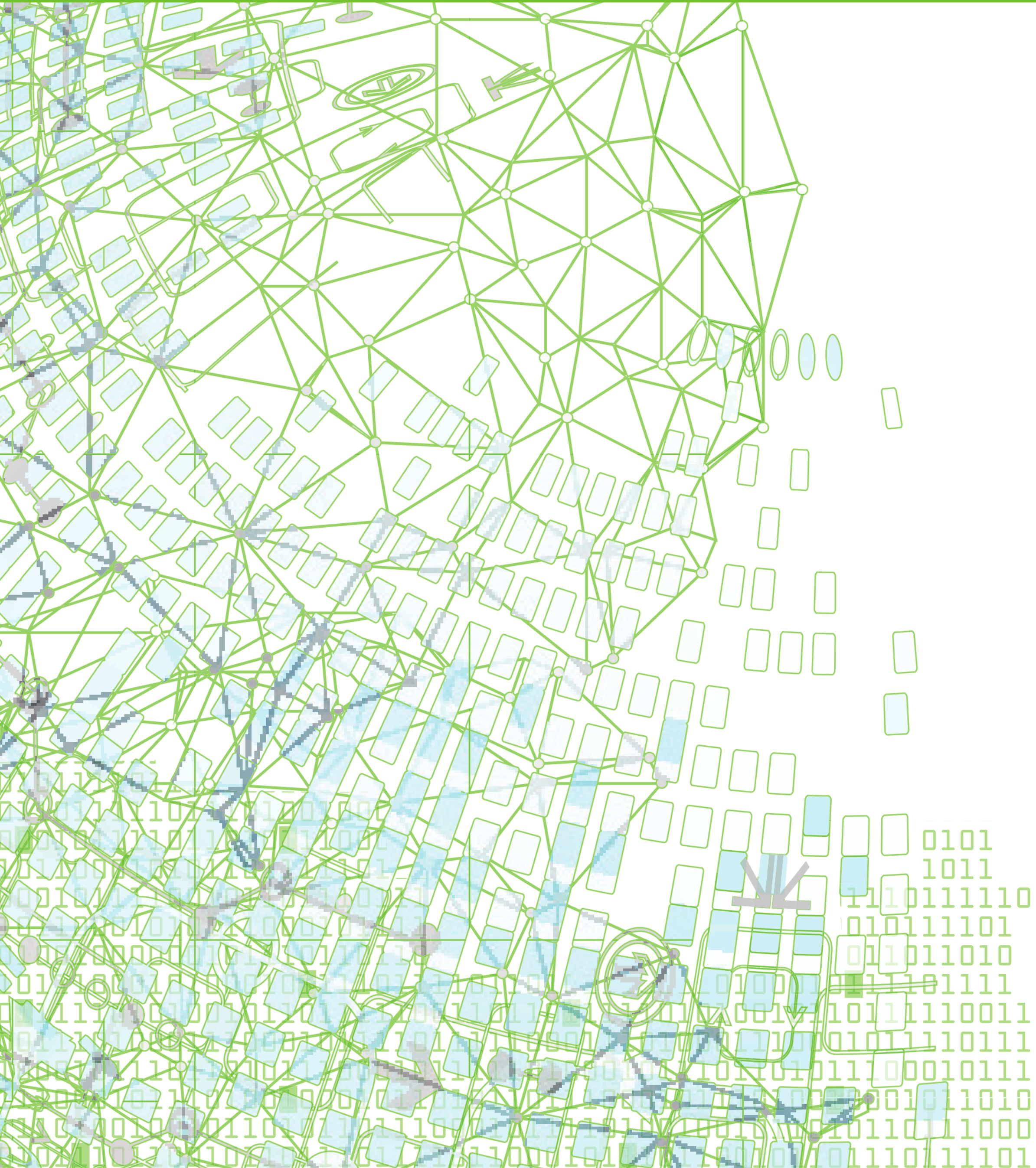

TABLE OF CONTENTS

LIST OF TABLES

Chapter 1. RELEASE OVERVIEW

Welcome to Release 2017 of the PGI Accelerator™ C11, C++14 and Fortran 2003 compilers and development tools for 64-bit x86-compatible processor-based workstations, servers, and clusters running versions of the Linux, Apple macOS and Microsoft Windows operating systems.

This document describes changes between previous releases of the PGI compilers and tools as well as late-breaking information not included in the current version of the PGI Compiler User's Guide.

> PGI Release 2016 version 16.4 and newer includes FlexNet license daemons updated to version 11.13.1.3. This update addresses a FlexNet security vulnerability. These new license daemons also work with older PGI releases. We recommend all users update their license daemons - see the licensing FAQ for more information. This FlexNet update also requires you to update your PGI FlexNet license keys to a new format. Older keys are incompatible.

1.1. Product Overview

All PGI products include exactly the same PGI compilers and tools software. The difference is the manner in which the license keys enable the software.

1.1.1. Licensing Terminology

The PGI compilers and tools are license-managed. Before discussing licensing, it is useful to have common terminology.

- **License** – a legal agreement between NVIDIA and PGI end-users, to which users assent upon installation of any PGI product. The terms of the License are kept up-to-date in documents on http://www.pgroup.com and in the `$PGI/<platform>/<rel_number>` directory of every PGI software installation.
- **License keys** – ASCII text strings that enable use of the PGI software and are intended to enforce the terms of the License. License keys are generated by each PGI end-user on pgroup.com using a unique hostid and are typically stored in a file

called `license.dat` that is accessible to the systems for which the PGI software is licensed.

- **PIN** – Product Identification Number, a unique 6-digit number associated with a license. This PIN is included in your PGI order confirmation. The PIN can also be found in your PGI license file after **`VENDOR_STRING=`**.
- **License PIN code** – A unique 16-digit number associated with each PIN that enables users to "tie" that PIN to their http://www.pgroup.com user account. This code is provided by PIN owners to others whom they wish tied to their PIN(s).

1.1.2. Bundled License Key

Installation may place a temporary license key in a file named `license.dat` in the PGI installation directory if no such file already exists.

If you use a separate license server, for example `LM_LICENSE_FILE=`*port@server.domain.com*, that supports this version, it is recommended that you remove or rename the license key file in the installation directory.

1.1.3. Node-locked and Network Floating Comparison

- With a PGI node-locked single-user license, one user at a time can compile on the one system on which the PGI compilers and tools are installed. The product and license server are on the same local machine.
- PGI network floating products are offered in configurations identical to PGI node-locked products, but include network-floating licenses. This means that one or more users can use the PGI compilers and tools concurrently on any compatible system networked to the license server, that is, the system on which the PGI network floating license keys are installed. There can be multiple installations of the PGI compilers and tools on machines connected to the license server; and the users can use the product concurrently, provided they are issued a license seat by the license server.

1.2. Release Components

Release 2017 includes the following components:

- PGFORTRAN™ native OpenMP and OpenACC Fortran 2003 compiler.
- PGCC® native OpenMP and OpenACC ANSI C11 and K&R C compiler.
- PGC++® native OpenMP and OpenACC ANSI C++14 compiler.
- PGI Profiler® OpenACC, CUDA, OpenMP, and multi-thread graphical profiler.
- PGI Debugger® MPI, OpenMP, and multi-thread graphical debugger.
- Open MPI version 1.10.2 for 64-bit Linux including support for NVIDIA GPUDirect. GPUDirect requires CUDA 7.5 or later. Note that 64-bit linux86-64 MPI messages are limited to < 2 GB size each. As NVIDIA GPUDirect depends on InfiniBand support, Open MPI is also configured to use InfiniBand hardware if it is available on the system. InfiniBand support requires OFED 3.18 or later.
- MPICH libraries, version 3.2, for 64-bit macOS development environments.

- ScaLAPACK 2.0.2 linear algebra math library for distributed-memory systems for use with Open MPI, MPICH or MVAPICH, and the PGI compilers on 64-bit Linux and macOS for Intel 64 or AMD64 CPU-based installations.
- Microsoft HPC Pack 2012 MS-MPI Redistributable Pack (version 4.1) for 64-bit development environments (Windows only).
- BLAS and LAPACK library based on the customized OpenBLAS project source.
- A UNIX-like shell environment for 64-bit Windows platforms.
- FlexNet license utilities.
- Documentation in man page format and online PDFs.

1.2.1. Additional Components for PGI Network Floating Licenses

PGI floating licenses for Linux (formerly the PGI CDK Cluster Development Kit) include additional components available for download from the PGI website, but not contained in the installation package including:

- MVAPICH2 MPI libraries, version 2.0.
- MPICH MPI libraries, version 3.2.

1.2.2. MPI Support

You can use PGI products to develop and debug MPI applications. PGI node-locked licenses support debugging of up to 16 local MPI processes. PGI network floating licenses provide the ability to debug up to 256 remote MPI processes.

1.3. Terms and Definitions

This document contains a number of terms and definitions with which you may or may not be familiar. If you encounter an unfamiliar term in these notes, please refer to the online glossary at http://www.pgroup.com/support/definitions.htm.

These two terms are used throughout the documentation to reflect groups of processors:

Intel 64

A 64-bit Intel Architecture processor with Extended Memory 64-bit Technology extensions designed to be binary compatible with AMD64 processors. This includes Intel Pentium 4, Intel Xeon, Intel Core 2, Intel Core 2 Duo (Penryn), Intel Core (i3, i5, i7), both first generation (Nehalem) and second generation (Sandy Bridge) processors, as well as Ivy Bridge and Haswell processors.

AMD64

A 64-bit processor from AMD™ incorporating features such as additional registers and 64-bit addressing support for improved performance and greatly increased memory range. This term includes the AMD Athlon64™, AMD Opteron™, AMD Turion™, AMD Barcelona, AMD Shanghai, AMD Istanbul, AMD Bulldozer, and AMD Piledriver processors.

1.4. Supported Platforms

There are three platforms supported by the PGI compilers and tools for x86-64 processor-based systems.

> PGI 2017 supports 64-bit operating systems only. Compiling 32-bit applications for execution on either 32-bit or 64-bit operating systems is no longer supported on any platform.

- **64-bit Linux** – supported on 64-bit Linux operating systems running on a 64-bit x86 compatible processor.
- **64-bit macOS** – supported on 64-bit Apple macOS operating systems running on a 64-bit Intel processor-based Macintosh computer.
- **64-bit Windows** – supported on 64-bit Microsoft Windows operating systems running on a 64-bit x86-compatible processor.

1.5. Supported Operating System Updates

This section describes updates and changes to PGI 2017 that are specific to Linux, macOS, and Windows.

1.5.1. Linux

- CentOS 5+, including CentOS 7
- Fedora 6+, including Fedora 24
- OpenSUSE 11+, including OpenSUSE Leap 42.2
- RHEL 5+, including RHEL 7.3
- SLES 11+, including SLES 12 SP 2
- Ubuntu 12.04+, including Ubuntu 16.10

1.5.2. Apple macOS

PGI 2017 for macOS supports most of the features of the version for Linux environments. Except where noted in these release notes or the user manuals, the PGI compilers and tools on macOS function identically to their Linux counterparts.

- The compilers, debugger, and profiler are supported on macOS versions 10.9 (Mavericks) through 10.12 (Sierra).

1.5.3. Microsoft Windows

PGI products for Windows support most of the features of the PGI products for Linux environments. PGI products on all Windows systems include the Microsoft Open Tools but also require that a Microsoft Windows Software Development Kit (SDK) be installed prior to installing the compilers.

PGI 2017 requires the Windows 10 SDK, even on Windows 7, 8 and 8.1.

These Windows operating systems are supported in PGI 2017:

- Windows Server 2008 R2
- Windows 7
- Windows 8
- Windows 8.1
- Windows 10
- Windows Server 2012
- Windows Server 2016

1.6. Precompiled Open-Source Packages

Many open-source software packages have been ported for use with PGI compilers on Linux x86-64.

The following PGI-compiled open-source software packages are included in the PGI Linux x86-64 download package:

- OpenBLAS 0.2.19 – customized BLAS and LAPACK libraries based on the OpenBLAS project source.
- Open MPI 1.10.2 – open-source MPI implementation.
- ScaLAPACK 2.0.2 – a library of high-performance linear algebra routines for parallel distributed memory machines. ScaLAPACK uses Open MPI 1.10.2.

The following list of open-source software packages have been precompiled for execution on Linux x86-64 targets using the PGI compilers and are available to download from the PGI website:

- MPICH 3.2 – open-source MPI implementation.
- MVAPICH2 2.2 – open-source MPI implementation.
- ESMF 7.0.0 for Open MPI 1.10.2 – The Earth System Modeling Framework for building climate, numerical weather prediction, data assimilation, and other Earth science software applications.
- ESMF 7.0.0 for MPICH 3.2
- ESMF 7.0.0 for MVAPICH2 2.2
- NetCDF 4.4.1.1 for GCC 4.x – A set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data, written in C. Included in this package are the following components:
    - NetCDF-C++ 4.3.0 – C++ interfaces to NetCDF libraries.
    - NetCDF-Fortran 4.4.4 – Fortran interfaces to NetCDF libraries.

  - HDF5 1.10.0-patch1 – data model, library, and file format for storing and managing data.
  - CURL 7.46.0 – tool and a library (usable from many languages) for client-side URL transfers.
  - SZIP 2.1 – extended-Rice lossless compression algorithm.
  - ZLIB 1.2.8 – file compression library.
- NetCDF 4.4.1.1 for GCC 5.x – includes all the components listed in NetCDF for GCC 4.x above.
- Parallel NetCDF 1.7.0 for MPICH 3.2
- Parallel NetCDF 1.7.0 for MVAPICH2 2.2
- Parallel NetCDF 1.7.0 for Open MPI 1.10.2

In addition, these software packages have also been ported to PGI on Linux x86-64 but due to licensing restrictions, they are not available in binary format directly from PGI. You can find instructions for building them in the Porting & Tuning Guides section of the PGI website.

- FFTW 2.1.5 – version 2 of the Fast Fourier Transform library, includes MPI bindings built with Open MPI 1.10.2.
- FFTW 3.3.4 – version 3 of the Fast Fourier Transform library, includes MPI bindings built with Open MPI 1.10.2.

For additional information about building these and other packages, please see the Porting & Tuning Guides section of the PGI website.

1.7. Getting Started

By default, the PGI 2017 compilers generate code that is optimized for the type of processor on which compilation is performed, the compilation host. If you are unfamiliar with the PGI compilers and tools, a good option to use by default is the aggregate option -fast.

Aggregate options incorporate a generally optimal set of flags for targets that support SSE capability. These options incorporate optimization options to enable use of vector streaming SIMD instructions . They enable vectorization with SSE instructions, cache alignment, and flushz.

The content of the -fast option is host-dependent.

The following table shows the typical -fast options.

Table 1 Typical -fast Options

Use this option...	To do this...
-O2	Specifies a code optimization level of 2.
-Munroll=c:1	Unrolls loops, executing multiple instances of the original loop during each iteration.

Use this option...	To do this...
-Mnoframe	Indicates to not generate code to set up a stack frame. **Note** With this option, a stack trace does not work.
-Mlre	Indicates loop-carried redundancy elimination.
-Mpre	Indicates partial redundancy elimination

-fast also typically includes the options shown in the following table:

Table 2 Additional -fast Options

Use this option...	To do this...
-Mvect=simd	Generates packed SSE and AVX instructions.
-Mcache_align	Aligns long objects on cache-line boundaries.
-Mflushz	Sets flush-to-zero mode.
-M[no]vect	Controls automatic vector pipelining.

For best performance on processors that support SSE and AVX instructions, use the PGFORTRAN compiler, even for FORTRAN 77 code, and the -fast option.

In addition to -fast, the -Mipa=fast option for interprocedural analysis and optimization can improve performance. You may also be able to obtain further performance improvements by experimenting with the individual -Mpgflag options that are described in the *PGI Compiler Reference Manual*, such as -Mvect, -Munroll, -Minline, -Mconcur, -Mpfi, -Mpfo, and so on. However, increased speeds using these options are typically application and system dependent. It is important to time your application carefully when using these options to ensure no performance degradations occur.

Chapter 2.
NEW AND MODIFIED FEATURES

This chapter provides information about the new or modified features of Release 2017 of the PGI compilers and tools.

Many user-requested fixes and updates are implemented in this release. Refer to http://www.pgroup.com/support/release_tprs.htm for a complete and up-to-date table of technical problem reports fixed in recent releases of PGI compilers and tools. This table contains a summary description of each problem as well as the version in which it was fixed.

2.1. What's New in Release 2017

17.4 Updates and Additions

- PGI Accelerator Compilers
  - Added support for atomic add and atomic subtract in device code for Fortran's single precision complex data types.
  - Added device versions of isfinite, isnan, isinf, and round.
  - Added support for CUDA Fortran warp-vote operations.
  - Changed the impact of `-g` on the optimization level used when compiling device code. For host code, `-g` sets the compiler's optimization level to zero unless another optimization level is specified using a `-O` option. For device code, the same is now true. For details about generating debug information for device code, refer to the PGI Compiler Reference Guide's section on DWARF Debugging Formats.
  - Updated the version of nvlink to 8.0.73; this version of the device linker allows an arbitrary ordering of object files on its command line.
- PGI Tools

  Improved debugging support of C++ programs on macOS versions 10.11 (El Capitan) and 10.12 (Sierra).

17.3 Updates and Additions

- Added debugger support for macOS version 10.12 (Sierra).
- Updated the PGI C/C++ compiler plug-in for Eclipse versions Neon and Mars; the Eclipse plug-in is available for Linux platforms.

17.1 Updates and Additions

- All PGI Compilers
    - Updated how floating point divides are computed to guarantee results will be uniform for scalar and vector operations. This change can cause numerical differences between PGI 17.1 and previous PGI releases. In rare cases, this change can cause an increase in execution time.
    - Improved inlining with the -Minline option. You may see different functions inlined with this release than you observed in prior releases. In some cases, compilation may take longer to complete because of an increase in the number of functions inlined. Some of the -Minline sub-options, which you can use to control inlining, have also changed from previous releases:
        - Added totalsize:n to limit inlining to the total size of n where n is the size of the combined program units on a per file basis.
        - Changed size:n to maxsize:n which allows inlining only of functions smaller than approximately n lines. The compilers silently convert the previous size:n to maxsize:n.
        - Dropped levels:n which limited inlining to n levels of functions. The compilers silently ignore levels:n.
    - Added the -cpp option as an alias for the -Mpreprocess option.
- PGI C++ Compiler
    - Integrated EDG release 4.11 and enabled interoperability with GCC 5.2, 5.3, 5.4, 6.0, 6.1, and 6.2. To take full advantage of the new C++11 features supported by GCC 5.1 and later, use the PGC++ compiler option --c++11.
    - Comprehensive support for C++14; co-installation with GCC version 5.0 or greater is required. Enable C++14 features with the PGC++ compiler option --c++14.
    - Added C++11 support to PGC++ when PGC++ is used as the nvcc host compiler; this support requires a patched post-CUDA-8.0-production version of nvcc. Contact sales@pgroup.com for more information.
    - Improved exception handling support and dropped the following related compiler options: --[no_]exceptions, --need_exception_spec, --sjlj_eh, --[no]zc_eh.
    - Added -M[no]variadic_macros to allow or disallow variadic macros. Variadic macros are allowed by default.
    - Deprecated the compiler options --edg and --gnu.

    - GNU changed the GCC STL header files in version 5.0. These changes may introduce compatibility issues with code compiled with earlier GCC versions. The GCC-supplied workaround to support compatibility with pre-GCC 5.0 compiled objects and libraries is to compile with -D_GLIBCXX_USE_CXX11_ABI=0. For more information, refer to https://gcc.gnu.org/onlinedocs/libstdc++/manual/using_dual_abi.html.
- PGI Accelerator OpenACC Compilers
    - Dropped support for CUDA 7.0.
    - Changed the default version of the CUDA Toolkit used by the compilers from CUDA 7.0 to 7.5. The CUDA Toolkit 8.0 can be used instead of 7.5 by adding the sub-option cuda8.0 to the -ta=tesla or -Mcuda compile- and link-time options.
    - Removed compute capability 2.0 (Fermi) from the compilers' default set of compute capabilities. The cc20 sub-option to the -ta=tesla and -Mcuda options is deprecated.
    - Improved data management within the OpenACC **`cache`** directive.
    - Added support for additional OpenACC 2.5 features:
        - Changed the behavior of the **`exit data`** directive to decrement the dynamic reference count.
        - Added the new optional **`finalize`** clause to set the dynamic reference count to zero.
        - Added the **`if_present`** clause to the **`update`** directive which changes the behavior when data is not present from a runtime error to a no-op.
        - Added new **`init`**, **`shutdown`**, and **`set`** directives.
        - Added new API routines to get and set the default async queue value.
        - Added support for the new definition of **`routine bind`** clause.
        - Updated the value of **`_OPENACC`** to 201510.

    With the exception of nested parallelism, declare link, and adding restrictions to **`cache`** clause variable references to variables within a cached region, OpenACC 2.5 feature support is complete in PGI 2017.
- PGI Tools
    - Updated the version of the JRE included in all installation packages to version 1.8.0_112.
    - Added variable rollover to the PGI debugger's graphical interface.
    - The PGI debugger requires libncurses.so.5. Some newer Linux distributions, such as Fedora 24, do not include this version of this library by default. In this situation, install the ncurses compatibility library to use the debugger.
- Libraries
    - PGI products for Linux ship with Open MPI 1.10.2.

  - Users with PGI subscription support can access separate downloads of MVAPICH 2.2 and MPICH 3.2.
  - PGI-built versions of NetCDF 4.4.1.1 and Parallel NetCDF 1.7.0 are available online at http://www.pgroup.com/resources/tips.htm
  - PGI-built versions of the Earth System Modeling Framework (ESMF) 7.0.0, one per PGI-built MPI distribution, are available online at the address above.
- Other Features and Additions
  - Added support for these new Linux x86-64 operating systems: Fedora 24, RHEL 7.3, Ubuntu 16.10.
  - Added compiler and profiler support for macOS version 10.12 (Sierra); the PGI debugger is not supported on Sierra in PGI 17.1.
  - Added support for Microsoft Windows Server 2016.
- Deprecations and Eliminations
  - PGI 2017 supports 64-bit operating systems only. Compiling 32-bit applications for execution on either 32-bit or 64-bit operating systems is no longer supported on any platform.
  - The PGI 2017 release for macOS no longer supports CUDA Fortran and OpenACC running on GPUs or CUDA-x86 running on CPUs. OpenACC targeting multi-core CPUs is still supported using the -ta sub-options host and multicore. The -acc option now implies -ta=multicore on macOS.
  - Dropped support for the CUDA 7.0 toolkit.
  - Dropped support for OS X 10.7 (Lion) and 10.8 (Mountain Lion).

2.2. New Features in PGI Accelerator OpenACC Compilers

Multicore Support

PGI Accelerator OpenACC compilers support the option -ta=multicore, to set the target accelerator for OpenACC programs to the host multicore. This will compile OpenACC compute regions for parallel execution across the cores of the host X86 processor or processors. The host multicore will be treated as a shared-memory accelerator, so the data clauses (**`copy`**, **`copyin`**, **`copyout`**, **`create`**) will be ignored and no data copies will be executed.

By default, -ta=multicore will generate code that will use all the available cores of the processor. If the compute region specifies a value in the **`num_gangs`** clause, the minimum of the **`num_gangs`** value and the number of available cores will be used. At runtime, the number of cores can be limited by setting the environment variable **`ACC_NUM_CORES`** to a constant integer value. If an OpenACC compute construct appears lexically within an OpenMP parallel construct, the OpenACC compute region will generate sequential code. If an OpenACC compute region appears dynamically within an OpenMP region or another OpenACC compute region, the program may generate many more threads than there are cores, and may produce poor performance.

The ACC_BIND environment variable is set by default with -ta=multicore; **`ACC_BIND`** has similiar behavior to **`MP_BIND`** for OpenMP.

The -ta=multicore option differs from the -ta=host option in that -ta=host generates sequential code for the OpenACC compute regions. In this release, -ta=multicore is not supported in conjunction with -ta=tesla or -ta=host.

Default Compute Capability

The default compute capability list for OpenACC and CUDA Fortran compilation for NVIDIA Tesla targets is cc30, cc35, and cc50. If CUDA 8.0 is specified using the cuda8.0 sub-option, cc60 is added to the default compute capability list. The generation of device code can be time consuming, so you may notice an increase in compile time. You can override the default by specifying one or more compute capabilities using either command-line options or an `rcfile`.

To change the default with a command-line option, provide a comma-separated list of compute capabilities to -ta=tesla: for OpenACC or -Mcuda= for CUDA Fortran.

To change the default with an `rcfile`, set the **`DEFCOMPUTECAP`** value to a blank-separated list of compute capabilities in the siterc file located in your installation's bin directory:

```
set DEFCOMPUTECAP=30 35 50;
```

Alternatively, if you don't have permissions to change the `siterc` file, you can add the **`DEFCOMPUTECAP`** definition to a separate `.mypgirc` file (`mypgi_rc` on Windows) in your home directory.

OpenACC 2.5 Features Not Supported

The PGI 2017 release contains support for all OpenACC 2.5 features except adding restrictions to **`cache`** clause variable refs to variables within a cached region.

OpenACC 2.0 Features Not Supported

The PGI 2017 release contains support for all OpenACC 2.0 features except the following:

- The declare link directive for global data is not implemented.
- Nested parallelism (parallel and kernels constructs within a parallel or kernels region) is not implemented.

Support for Profiler/Trace Tool Interface

PGI compilers support the OpenACC 2.5 version of the OpenACC Profiler/Trace Tools Interface. This is the interface used by the PGI profiler to collect performance measurements of OpenACC programs.

2.3. C++ Compiler

2.3.1. C++ and OpenACC

This release includes support for OpenACC directives for the PGC++ compiler. There are limitations to the data that can appear in data constructs and compute regions:

- Variable-length arrays are not supported in OpenACC data clauses; VLAs are not part of the C++ standard.
- Variables of class type that require constructors and destructors do not behave properly when they appear in data clauses.
- Exceptions are not handled in compute regions.
- Any function call in a compute region must be inlined. This includes implicit functions such as for I/O operators, operators on class type, user-defined operators, STL functions, lambda operators, and so on.

2.3.2. C++ Compatibility

PGI 2017 C++ object code is incompatible with PGI 2016 and prior releases.

All C++ source files and libraries that were built with prior releases must be recompiled to link with PGI 2017 or higher object files.

Optional packages that provide a C++ interface—such as the MPI package included with all PGI products—now require the use of the `pgc++` compiler driver for compilation and linking.

These optional packages include:

- ESMF
- MPICH
- MVAPICH
- NetCDF
- Open MPI
- Parallel NetCDF
- ScaLAPACK

2.4. Runtime Library Routines

PGI 2017 supports runtime library routines associated with the PGI Accelerator compilers. For more information, refer to *Using an Accelerator* in the PGI Compiler User's Guide.

2.5. Library Interfaces

PGI provides access to a number of libraries that export C interfaces by using Fortran modules. These libraries and functions are described in the PGI Compiler User's Guide.

2.6. Environment Modules

On Linux, if you use the Environment Modules package (e.g., the `module load` command), then PGI 2017 includes a script to set up the appropriate module files.

Chapter 3. DISTRIBUTION AND DEPLOYMENT

Once you have successfully built, debugged and tuned your application, you may want to distribute it to users who need to run it on a variety of systems. This section addresses how to effectively distribute applications built using PGI compilers and tools.

3.1. Application Deployment and Redistributables

Programs built with PGI compilers may depend on runtime library files. These library files must be distributed with such programs to enable them to execute on systems where the PGI compilers are not installed. There are PGI redistributable files for Linux and Windows. On Windows, PGI also supplies Microsoft redistributable files.

3.1.1. PGI Redistributables

The PGI 2017 Release includes these directories:

```
$PGI/linux86-64/17.4/REDIST
$PGI/win64/17.4/REDIST
```

These directories contain all of the PGI Linux runtime library shared object files or Windows dynamically linked libraries that can be re-distributed by PGI 2017 licensees under the terms of the PGI End-User License Agreement (EULA). For reference, a text-form copy of the PGI EULA is included in the 17.4 `doc` directory.

3.1.2. Linux Redistributables

The Linux `REDIST` directories contain the PGI runtime library shared objects for all supported targets. This enables users of the PGI compilers to create packages of executables and PGI runtime libraries that will execute successfully on almost any PGI-supported target system, subject to these requirements:

- End-users of the executable have properly initialized their environment.
- Users have set **`LD_LIBRARY_PATH`** to use the relevant version of the PGI shared objects.

3.1.3. Microsoft Redistributables

The PGI products on Windows include Microsoft Open Tools. The Microsoft Open Tools directory contains a subdirectory named `redist`. PGI 2017 licensees may redistribute the files contained in this directory in accordance with the terms of the PGI End-User License Agreement.

Microsoft supplies installation packages, `vcredist_x86.exe` and `vcredist_x64.exe`, containing these runtime files. These files are available in the `redist` directory.

Chapter 4. TROUBLESHOOTING TIPS AND KNOWN LIMITATIONS

This section contains information about known limitations, documentation errors, and corrections. Wherever possible, a work-around is provided.

For up-to-date information about the state of the current release, visit the frequently asked questions (FAQ) section on pgroup.com at http://www.pgroup.com/support/faq.htm.

4.1. Platform-specific Issues

4.1.1. Linux

The following are known issues on Linux:

- Programs that incorporate object files compiled using -mcmodel=medium cannot be statically linked. This is a limitation of the linux86-64 environment, not a limitation of the PGI compilers and tools.

4.1.2. Apple macOS

The following are known issues on Apple macOS:

- The PGI 2017 compilers do not support static linking of binaries. For compatibility with future Apple updates, the compilers only support dynamic linking of binaries.

4.1.3. Microsoft Windows

The following are known issues on Windows:

- For the Cygwin `emacs` editor to function properly, you must set the environment variable **`CYGWIN`** to the value "tty" before invoking the shell in which `emacs` will run. However, this setting is incompatible with the PGBDG command line interface (-text), so you are not able to use **`pgdbg -text`** in shells using this setting.

- On Windows, the version of `vi` included in Cygwin can have problems when the **`SHELL`** variable is defined to something it does not expect. In this case, the following messages appear when `vi` is invoked:

  E79: Cannot expand wildcards Hit ENTER or type command to continue

  To work around this problem, set **`SHELL`** to refer to a shell in the Cygwin `bin` directory, e.g., `/bin/bash`.
- On Windows, runtime libraries built for debugging (e.g., `msvcrtd` and `libcmtd`) are not included with PGI products. When a program is linked with -g, for debugging, the standard non-debug versions of both the PGI runtime libraries and the Microsoft runtime libraries are always used. This limitation does not affect debugging of application code.

4.2. Issues Related to Debugging

The following are known issues in the PGI debugger:

- Debugging of PGI Unified Binaries, that is, programs built with more than one -tp option, is not fully supported. The names of some subprograms are modified in compilation and the debugger does not translate these names back to the names used in the application source code. For detailed information on how to debug a PGI Unified Binary, refer to http://www.pgroup.com/support/tools.htm.
- The debugging of C++ programs is currently limited to macOS versions 10.9 (Mavericks) and 10.10 (Yosemite); there are no limitations on the debugging of C or Fortran programs.
- When debugging on the Windows platform, the Windows operating system times out **`stepi`**/**`nexti`** operations when single stepping over blocked system calls. For more information on the work-around for this issue, refer to the online FAQs at http://www.pgroup.com/support/tools.htm.

4.3. Profiler-related Issues

PGI 2016 introduced the all-new PGI profiler. User feedback will be tracked for reported issues.

4.4. OpenACC Issues

This section includes known limitations in PGI's support for OpenACC directives. PGI plans to support these features in a future release.

ACC routine directive limitations

- Fortran assumed-shape arguments are not yet supported.

Clause Support Limitations

- Not all clauses are supported after the `device_type` clause.

4.5. Corrections

A number of problems are corrected in this release. Refer to www.pgroup.com/support/release_tprs.htm for a complete and up-to-date table of technical problem reports, TPRs, fixed in recent releases of the PGI compilers and tools. This table contains a summary description of each problem as well as the version in which it was fixed.

Chapter 5.
CONTACT INFORMATION

You can contact PGI at:

20400 NW Amberwood Drive Suite 100
Beaverton, OR 97006

Or electronically using any of the following means:

Fax: +1-503-682-2637
Sales: sales@pgroup.com
WWW: http://www.pgroup.com

The PGI User Forum is monitored by members of the PGI engineering and support teams as well as other PGI customers. The forum newsgroups may contain answers to commonly asked questions. Log in to the PGI website to access the forum:

http://www.pgroup.com/userforum/index.php

Many questions and problems can be resolved by following instructions and the information available at our frequently asked questions (FAQ) site:

http://www.pgroup.com/support/faq.htm

Submit technical support requests through the online form at:

https://www.pgroup.com/support/support_request.php

PGI documentation is available at http://www.pgroup.com/resources/docs.htm.

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication of otherwise under any patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all other information previously supplied. NVIDIA Corporation products are not authorized as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

PGI Workstation, PGI Server, PGI Accelerator, PGF95, PGF90, PGFORTRAN, and PGI Unified Binary are trademarks; and PGI, PGHPF, PGF77, PGCC, PGC++, PGI Visual Fortran, PVF, PGI CDK, Cluster Development Kit, PGPROF, PGDBG, and The Portland Group are registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

© 2013–2017 NVIDIA Corporation. All rights reserved.