PGI Accelerator Compilers

Accelerate!

The PGI 9.0 release includes the PGI Accelerator™ Fortran and C99 compilers supporting x64+NVIDIA Linux systems; PGF95 and PGCC accelerator compilers are supported on all Intel and AMD x64 processor-based systems with CUDA-enabled NVIDIA GPUs.

Overview

Using PGI Accelerator compilers, programmers can accelerate Linux applications on x64+GPU platforms by adding OpenMP-like compiler directives to existing high-level standard-compliant Fortran and C programs and then recompiling with appropriate compiler options.

Sample Fortran matrix multiplication loop, tagged to be compiled for an accelerator.

```fortran
!$acc region
do k = 1,n1
   do i = 1,n3
      c(i,k) = 0.0
      do j = 1,n2
         c(i,k) = c(i,k) + a(i,j) * b(j,k)
      enddo
   enddo
enddo
!$acc end region
```

How It Works

Until now, C and C++ developers targeting GPU accelerators have had to rely on language extensions to their programs. Use of GPUs from Fortran applications has been extremely limited. x64+GPU programmers have been required to program at a detailed level including a need to understand and specify data usage information and manually construct sequences of calls to manage all movement of data between the x64 host and GPU.

The PGI Accelerator compilers automatically analyze whole program structure and data, split portions of the application between the x64 CPU and GPU as specified by user directives, and define and generate an optimized mapping of loops to automatically use the parallel cores, hardware threading capabilities and SIMD vector capabilities of modern GPUs. In addition to directives and pragmas that specify regions of code or functions to be accelerated, the PGI Accelerator compilers support user directives that give the programmer fine-grained control over the mapping of loops, allocation of memory, and optimization for the GPU memory hierarchy. The PGI Accelerator compilers generate unified x64+GPU object files and executables that manage all movement of data to and from the GPU device while leveraging all existing host-side utilities—linker, librarians, makefiles—and require no changes to the existing standard HPC Linux/x64 programming environment.

Additional Resources

- PGI Accelerator Compiler Overview Video (time: 3:20)
- PGI Accelerator Compiler Examples Video (time: 10:00)
- The PGI Accelerator Programming Model on NVIDIA GPUs—Part 1
- The PGI Accelerator Programming Model on NVIDIA GPUs—Part 2 Performance Tuning
- Building Cactus BenchADM with PGI Accelerator Compilers Tutorial
- PGI x64+GPU Fortran & C99 Compilers IEEE webinar presented 16 June 2009 (time: 1:05:00)
- PGI x64+GPU Fortran & C99 Compilers IEEE webinar slides (pdf)
- PGI Fortran & C Accelerator Programming Model white paper (ver. 1.0 Jun. 2009, 188KB)
FAQ

Please also see the PGI Accelerator Programming user forum for additional questions and answers.

- Which programming languages do the PGI Accelerator compilers support?
- On which operating systems do PGI Accelerator compilers run?
- Which accelerators can be targeted by PGI Accelerator compilers?
- What are PGI’s plans for supporting of the new Fermi accelerators coming soon from NVIDIA?
- Do I need to install the CUDA software?
- Does the compiler support IEEE standard floating-point arithmetic?
- Do PGI Accelerator compilers support double-precision?
- Can I call a CUDA kernel function from my PGI-compiled code?
- Does the compiler support two or more GPUs in the same program?
- Is there an effort to open your directives to a standards committee, like OpenMP?
- Can I run my program on a machine that doesn’t have an accelerator on it?
- Do I have to rebuild my application for each different GPU model?
- Can I use function or procedure calls in my GPU code?
- When will you support <my favorite feature> in your compiler?
- Are all the specified directives supported in this release?
- What is an Extended Preview?
- How much does it cost?
- So how can I try it?

Q Which programming languages do the PGI Accelerator compilers support?
A Currently, PGI has added support for GPU accelerators to the PGF95™ Fortran and PGCC® ANSI C99 compilers. While adding support for C++ is technically feasible, we have no current timeline for the availability of this capability. We welcome your feedback.

Q On which operating systems do PGI Accelerator compilers run?
A Linux 64 only is supported in the PGI 9.0 release. PGI is working to add support for this same programming model on Windows and Mac OS X and expects to deliver products on both platforms before the end of 2009.

Q Which accelerators can be targeted by PGI Accelerator compilers?
A PGI Accelerator compilers target all CUDA-enabled NVIDIA GPU accelerators with compute capability 1.0 or higher. PGI is studying the feasibility of supporting other accelerators.

Q What are PGI’s plans for supporting of the new Fermi accelerators coming soon from NVIDIA?
A PGI is continuing to work closely with NVIDIA to ensure that support for the new Fermi architecture based GPUs will be available concurrent with the hardware.

Q Do I need to install the CUDA software?
A Currently, you need to download and install the CUDA software from NVIDIA. In addition, you need to create or edit a file named 'sitenvrc' in the $PGI/9.0/bin/ installation directory. Details are described in section 7.2 of the PGI Server/PGI Workstation Installation Guide.

Q Does the compiler support IEEE standard floating-point arithmetic?
A The GPU accelerators available today support most of the IEEE floating-point standard. However, they do not support all the rounding modes, and some operations, notably square root, exponential, logarithm, and other transcendental functions, may not deliver full precision results. This is a hardware limitation that compilers cannot overcome.

Q Do PGI Accelerator compilers support double-precision?
A Yes, but please be aware that while today's latest GPU accelerators do have support for double-precision, performance can be quite low relative to single-precision.

Q Can I call a CUDA kernel function from my PGI-compiled code?
A PGI is working on the design of a feature to allow you to call kernel functions written in CUDA or PTX or other languages directly from your C or Fortran program. We will announce this feature when it is available.

Q Does the compiler support two or more GPUs in the same program?
As with CUDA, you can use two or more GPUs by using multiple threads, where each thread attaches to a different GPU and runs its kernels on that GPU. The current release does not include support to automatically control two or more GPUs from the same accelerator region.

Q Is there an effort to open your directives to a standards committee, like OpenMP?
A As we gain experience with our directives and programming model, we will be open to exploring a standardization effort.

Q Can I run my program on a machine that doesn't have an accelerator on it?
A Yes. PGI Accelerator compilers can generate PGI Unified Binary™ technology executables that work in the presence or absence of an accelerator.

Q Do I have to rebuild my application for each different GPU model?
A The GPU code generated uses the same technology that is used for graphics applications and games; that is, the program uses a portable intermediate format which is then dynamically translated and re-optimized at run time by the drivers supplied by the vendor for the particular model of GPU in your machine. This preserves your investment by allowing your programs to continue to work even when you upgrade your GPU card, or use your program on a machine with a different model of GPU.

Q Can I use function or procedure calls in my GPU code?
A Current GPUs do not support function calls. The compiler will support function calls only if they can be inlined.

Q When will you support <my favorite feature> in your compiler?
A Some features cannot be supported due to limitations of the hardware. Other features are not being supported because they would not deliver satisfactory performance. Still other features are planned for future implementation. Your feedback can affect our priorities.

Q Are all the specified directives supported in the this release?
A Not all the directives in the PGI Fortran & C Accelerator Programming Model white paper are implemented with PGI release 9.0. The list of directives remaining to be implemented are:

- Accelerator Data Region directive and associated declarative data clauses
- Update directive
- Loop Scheduling directive clauses
  - unroll
  - shortloop
  - cache
- Runtime routines
  - acc_shutdown
  - acc_on_device

Q What is an Extended Preview?
A Targeting x64+NVIDIA Linux systems is enabled in PGI 9.0. It is available now at no charge to PGI Linux licensees with a current subscription as part of the PGI Accelerator Compiler Extended Preview Program. The Extended Preview Program runs through 31-Dec 2009. On that date, the accelerator features will stop functioning (x64 functionality will not be affected). To regain access to the accelerator features after that date will require upgrading to an accelerator-enabled license.

Q How much does it cost?
A New license pricing for the PGI Accelerator compilers will be announced soon. In the meantime, purchasing any new PGI Linux license or renewing your expired PGI Subscription Service will enable you to participate in the Extended Preveiw Program. Please contact PGI Sales for details.

Q So how can I try it?
A To try out the PGI Accelerator compilers during the Extended Preview program, follow these three steps:

a. Download one of PGI 9.0 64-bit Linux software package (PGI Workstation, PGI Server or PGI CDK).
b. Review the PGI Workstation/PGI Server Installation Guide or the PGI CDK Installation Guide and configure your environment.
c. Obtain license keys. Available options include:

1. **You have a current PGI Linux subscription**—you will need to [retrieve the 9.0 update for your permanent license keys](http://www.pgroup.com/resources/accel.htm).

2. **Your PGI Linux subscription has expired**—you can either generate 15 day trial keys as outlined in option 5 below, or you can bring your subscription current and gain access to the Extended Preview accelerator feature through updated permanent license keys. By renewing your subscription, you will also qualify for any license upgrade incentives.

3. **You have a current PGI non-Linux subscription**—you can either generate 15 day trial keys as outlined in option 5 below, or you can exchange or upgrade your license to include Linux support.

4. **Your PGI non-Linux subscription has expired**—you can either generate 15 day trial keys as outlined in option 5 below, or you will first need to bring it current as outlined in option 2 and then exchange or upgrade it as described in option 3.

5. **You don't have a PGI license**—you can [generate 15 day trial license keys](http://www.pgroup.com/resources/accel.htm). The trial keys and all executable files compiled using them will cease operating at the end of the 15 day trial period.

Please contact [PGI Sales](http://www.pgroup.com/resources/accel.htm) for exchange, upgrade or subscription renewal information.