



ACCELERATE YOUR RESULTS WITH GPU COMPUTING

Accelerating applications on hundreds of computing cores in GPUs is easier than ever with a rich ecosystem of libraries, compiler directives, and programming languages for GPU computing. Whether your application runs on a single workstation or thousands of nodes in a cluster, there's a solution that meets your needs.

To get started, simply choose your preferred solution:

- GPU-accelerated libraries,
- Compiler directives, or
- A familiar programming language such as C, C++, or Fortran.

Learn more about getting started with GPU computing in the sections below.



GPU-ACCELERATED LIBRARIES

Take advantage of the massively parallel computing power of the GPU by using the GPU-accelerated versions of your existing libraries. Some examples include:



NVIDIA Math Libraries

A collection of GPU-accelerated libraries—including FFT, BLAS, sparse matrix operations, RNG, performance primitives for image/signal processing, and the Thrust C++ library of high-performance templated algorithms—that all deliver significant speedups when compared to CPU-only libraries. These highly optimized libraries are free of charge in the NVIDIA® CUDA® Toolkit available at www.nvidia.com/getcuda

APPLICATIONS		
Libraries	Compiler Directives	Programming Languages
"Drop-in" Acceleration	Quickly Accelerate Serial Applications	Broadest Applicability and Maximum Performance



EM Photonics CULA

A GPU-accelerated linear algebra (LA) library that dramatically improves the performance of sophisticated mathematics.



MAGMA

A collection of open-source, next-generation linear algebra libraries for heterogeneous GPU-based architectures supporting interfaces to current LA packages and standards (e.g. LAPACK and BLAS).



RogueWave IMSL


A comprehensive set of mathematical and statistical functions for Fortran applications to take advantage of GPU acceleration.

For a more complete list of commercial and open-source libraries, visit developer.nvidia.com/gpu-accelerated-libraries

DIRECTIVE-BASED SOLUTIONS

Directives allow you to quickly add GPU acceleration to the most performance critical sections of your application while maintaining portability. Directive-based solutions for C and Fortran available today include:

Multicore CPU



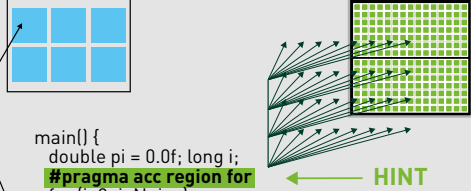
```
main() {
  double pi = 0.0f; long i;
  for (i=0; i<N; i++)
  {
    double t= (double)((i+0.5)/N);
    pi +=4.0/(1.0+t*t);
  }
  printf("pi=%f\n",pi/N);
}
```

Serial code is only able to take advantage of a single CPU core, even on multicore CPUs.

PGI Accelerator™ Compilers

Similar to OpenMP, PGI Accelerator directives can be added to existing C99 or Fortran applications to achieve GPU acceleration on Linux, Mac OS X, and Windows.

Multicore CPU



```
main() {
  double pi = 0.0f; long i;
  #pragma acc region for
  for (i=0; i<N; i++)
  {
    double t= (double)((i+0.5)/N);
    pi +=4.0/(1.0+t*t);
  }
  printf("pi=%f\n",pi/N);
}
```

Add a simple compiler directive to provide a hint for the compiler to automatically parallelize your for loop and execute it on the GPU.

CAPS HMPP Directives

This HMPP compiler integrates powerful data-parallel back ends for CUDA C and OpenCL that dramatically reduces development time. The HMPP runtime ensures application deployment on multi-GPU systems.

BENEFITS

- > **Easy** – Use simple compiler hints
- > **Fast** – Accelerate on GPUs quickly
- > **Portable** – Optimize on CPUs or GPUs

For more information on directive-based programming, visit www.nvidia.com/2xin4weeks

LANGUAGE INTEGRATION WITH C, C++, OR FORTRAN

Gain maximum performance and flexibility for your applications by writing your own GPU code. NVIDIA GPUs support the industry's most robust languages and APIs for GPU computing developers, including:

NVIDIA CUDA C AND CUDA C++

The NVIDIA CUDA Toolkit provides a comprehensive development environment

for C and C++ developers building GPU-accelerated applications. The toolkit includes a compiler for NVIDIA GPUs, math libraries, and tools for debugging and optimizing the performance of your applications. Also included are programming guides, user manuals, API reference, and other resources to help you get started.

PGI CUDA Fortran

A CUDA Fortran Compiler developed by PGI provides Fortran language support for NVIDIA's CUDA-enabled GPUs.

NVIDIA OpenCL

A low-level API for heterogeneous computing that runs on NVIDIA GPUs.

Learn more at: <http://developer.nvidia.com/languages>

DEVELOPER TOOLS

Understanding how your application uses the GPU is crucial for acceleration and optimization. Here are some available GPU debugging and performance analysis tools:

NVIDIA CUDA-GDB

A debugger capable of handling thousands of threads running simultaneously on each GPU in the system.

Allinea DDT

A powerful debugger that debugs hybrid MPI, OpenMP, and CUDA applications on a workstation or GPU cluster.

TotalView

AGUI-based debugger for one or many processes/threads with complete control over program execution, including support for OpenMP, MPI, and GPUs.

NVIDIA Visual Profiler

A cross-platform performance profiling tool that delivers vital feedback for optimizing CUDA C/C++ applications.

TAU Performance System

A profiling and tracing toolkit for performance analysis of hybrid parallel programs written in CUDA C, OpenCL, pyCUDA, or HMPP.

VampirTrace

A performance monitor that provides detailed insight to runtime behavior of accelerators enabling extensive performance analysis and optimization.

NVIDIA Parallel Nsight™ 2.0 for Visual Studio

A fully integrated CPU and GPU development environment for Microsoft Visual Studio 2011.

Visit developer.nvidia.com/cuda-tools-ecosystem

JOB SCHEDULING AND CLUSTER MANAGEMENT SOLUTIONS

Most commercial and open source job schedulers and cluster management solutions support GPUs. For more information on these powerful tools, visit: developer.nvidia.com/cuda-tools-ecosystem

TECHNICAL WEBINARS

Get training designed specifically for developers and participate in Q&A from our GPU computing experts at nvidia.com/gpu-computing-webinars

CONSULTING AND TRAINING SERVICES

Independent consulting and training services are available to support you in accelerating your applications. Learn more at: www.nvidia.com/object/cuda_consultants.html

Learn more about GPU computing at <http://developer.nvidia.com/cuda>

© 2011 NVIDIA, the NVIDIA logo, CUDA, and Parallel Nsight are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated. All rights reserved.

