CUDA Features Archive

User's Guide

v11.8 | October 2022

Table of Contents

Chapter 1. CUDA 11.6 Features..... 1

Chapter 1. CUDA 11.6 Features

Compiler

- **VS2022 Support**

  CUDA 11.6 officially supports the latest VS2022 as host compiler. A separate Nsight Visual Studio installer 2022.1.1 must be downloaded from **here.** A future CUDA release will have the Nsight Visual Studio installer with VS2022 support integrated into it.

- **New instructions in public PTX**

  New instructions for bit mask creation—BMSK, and sign extension—SZEXT, are added to the public PTX ISA. You can find documentation for these instructions in the PTX ISA guide: BMSK and SZEXT.

- **Unused Kernel Optimization**

  In CUDA 11.5, unused kernel pruning was introduced with the potential benefits of reducing binary size and improving performance through more efficient optimizations. This was an opt-in feature but in 11.6, this feature is enabled by default. As mentioned in the 11.5 blog, there is an opt-out flag that can be used in case it becomes necessary for debug purposes or for other special situations.

  ```
  $ nvcc -rdc=true user.cu testlib.a -o user -Xnvlink -ignore-host-info
  ```

- **New -arch=native option**

  In addition to the `-arch=all` and `-arch=all-major` options added in CUDA 11.5, NVCC introduced `-arch= native` in CUDA 11.5 update 1. This `-arch=native` option is a convenient way for users to let NVCC determine the right target architecture to compile the CUDA device code to based on the GPU installed on the system. This can be particularly helpful for testing when applications are run on the same system they are compiled in.

- **Generate PTX from nvlink:**

  Using the following command line, device linker, nvlink will produce PTX as an output in addition to CUBIN:

  ```
  nvcc -dlto -dlink -ptx
  ```

  Device linking by nvlink is the final stage in the CUDA compilation process. Applications that have multiple source translation units have to be compiled in separate compilation mode. LTO (introduced in CUDA 11.4) allowed nvlink to perform optimizations at device link time instead of at compile time so that separately compiled applications with several

translation units can be optimized to the same level as whole program compilations with a single translation unit. However, without the option to output PTX, applications that cared about forward compatibility of device code could not benefit from Link Time Optimization or had to constrain the device code to a single source file.

With the option for nvlink that performs LTO to generate the output in PTX, customer applications that require forward compatibility across GPU architectures can span across multiple files and can also take advantage of Link Time Optimization.

- **Bullseye support**

  NVCC compiled source code now works with the code coverage tool Bullseye. The code coverage is only for the CPU or the host functions. Code coverage for device function is not supported through bullseye.

- **INT128 developer tool support**

  In 11.5, CUDA C++ support for 128 bit was added. In 11.6, developer tools support the datatype as well. With the latest version of libcu++, int 128 data datype is supported by math functions.

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Trademarks

NVIDIA and the NVIDIA logo are trademarks or 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

© 2022-2022 NVIDIA Corporation & affiliates. All rights reserved.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
https://www.nvidia.com