Clang for NVIDIA Grace CPU

Release Notes 16.23.08

Document History

RN_11533-001

Version	Date	Authors	Description of Change
1	August, 2023	NVIDIA	Initial release

Table of Contents

Overview

This package is NVIDIA®'s build of Clang for the NVIDIA Grace™ CPU is based on v16.0.5 of the LLVM™ repository and is the alpha release of the product. As a member of the LLVM community, NVIDIA provides optimized builds of the open-source LLVM Clang compiler to realize impressive performance, stability, and support for NVIDIA Grace CPU.

This package is provided so users of NVIDIA Grace CPU can immediately take advantage of NVIDIA's contributions to LLVM. It is a tested and validated host compiler for NVIDIA® CUDA® on Grace Hopper™.

Release Deliverables

This build of Clang for Grace CPUs is suitable for building C / C++ applications on GNU/Linux systems.

- C compiler driver binary - clang
- C++ compiler driver binary - clang++
- LLVM Linker - lld
- OpenMP Runtime support - libomp

This release supports the use of LLD as the default linker.

OS Distribution Matrix

Architecture	Linux Distributions	CUDA Toolkit
AArch64	Ubuntu 22.04 RHEL 9 CentOS 9 SLES 15-SP4	12.2-Update 2

Dependencies

This package is designed to be executed on GNU/Linux OS platforms and distributions. The dependencies on the system are listed in the table below:

Dependency	Notes
TBB	Install the appropriate package of oneTBB or libtbb that works with the C++ runtime package on your distribution. This is required if your workload incorporates features from stdpar in c++17.
libstdc++	Developer package of libstdc++ for the appropriate Linux distribution you are using. On Ubuntu or Debian this would be the libstdc++-developer package. On RHEL and SLES, this would be the libstdc++-developer package.
libpython	Version 3.11.

Getting Started

Ubuntu:

- Download the debian (.deb) package from:

www.developer.nvidia.com/grace/clang

- Install the package as:

```
apt install clang-grace-toolchain_16.23.08_arm64.deb
```

- The default installation location is:

```
/opt/nvidia/grace/clang
```

- The user can change the install location using `-x` option:

```
dpkg-deb e.g. dpkg-deb -x clang-grace-toolchain_16.23.08_arm64.deb /home/username/install
```

RHEL/SLES:

- Download the RPM (.rpm) package from:

www.developer.nvidia.com/grace/clang

- Use the appropriate packaging tool (yum, dnf or zypper) depending on the distribution to install. For example:

```
if [ -f '/etc/redhat-release ] then;
   PACKAGING_TOOL = dnf
else
   PACKAGING_TOOL = zypper
fi
```

- Install the package as:

```
${PACKAGING_TOOL} install clang-grace-toolchain-16.23.08-1.aarch64.rpm
```

Tarballs

- Unpack the downloaded tar file using the standard tar utility on the system:

```
tar xvzf clang-grace-toolchain-16.23.08.tgz -C <path>
```

Testing the installation

- You can verify if your installation is clean by compiling a simple HelloWorld C++ program.

```
#include <iostream>
int main() {
  std::cout << "Hello World\n";
  return 0;
}
```

- The above code should compile without error with the installed Clang.

For example:

```
<installation_path>/bin/clang++ HelloWorld.cpp
```

should not result in any compiler or linker error.

Known Issues and Limitations

- stdpar with C++17 has not been validated with this release.
- OpenMP target offloading has been disabled in this release.

FAQ

1. How can a user set defaults at installation time to target NVIDIA Grace CPU?

Answer: A user can default to Neoverse-V2 CPU with Clang using the configuration files mentioned in the references (1) section.

For example:

```
$> cat grace.config
-mcpu=neoverse-v2
```

The user can invoke Clang as shown in the example command below to default to generating code for the Grace CPU.

```
$>clang --config grace.config file.cpp
```

Such a config file can be used to extend any further default options one might want to deploy in the installation.

2. How can a developer use a different linker to the one shipped in the package ?

Answer: The Clang shipped in the package defaults to `lld` as the linker. The Clang driver will pick up `lld` from bin/ directory in the installation location. However, the user can override the default linker using `-fuse -ld=<linker>` option.

For example, to use the GNU `ld` located in /usr/bin/, you can use the following command:

```
$> clang -fuse-ld=/usr/bin/ld  file.cpp
```

Report Issues

We would love to hear back from you. Please report issues and provide feedback through grace-compiler-feedback@nvidia.com.

References

1. Clang configuration files:
   https://clang.llvm.org/docs/UsersManual.html#configuration-files
2. Release notes for upstream components:
   - https://releases.llvm.org/16.0.0/tools/clang/docs/ReleaseNotes.html
   - https://releases.llvm.org/16.0.0/tools/lld/docs/ReleaseNotes.html
   - https://releases.llvm.org/16.0.0/docs/ReleaseNotes.html

Notice

The information provided in this specification is believed to be accurate and reliable as of the date provided. However, NVIDIA Corporation (“NVIDIA”) does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information. 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 publication supersedes and replaces all other specifications for the product that may have been previously supplied.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and other changes to this specification, at any time and/or to discontinue any product or service without notice. Customer should obtain the latest relevant specification 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. NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this specification.

NVIDIA products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer’s own risk.

NVIDIA makes no representation or warranty that products based on these specifications will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this specification. NVIDIA does not accept any liability related to any default, damage, costs or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this specification, or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this specification. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA. Reproduction of information in this specification is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

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. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

OpenCL

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

LLVM

LLVM is a trademark of LLVM Foundation.

Arm

Arm, Neoverse, AMBA, and Arm Powered are registered trademarks of Arm Limited. Cortex, MPCore, and Mali are trademarks of Arm Limited. All other brands or product names are the property of their respective holders. "Arm" is used to represent Arm Holdings plc; its operating company Arm Limited; and the regional subsidiaries Arm Inc.; Arm KK; Arm Korea Limited.; Arm Taiwan Limited; Arm France SAS; Arm Consulting (Shanghai) Co. Ltd.; Arm Germany GmbH; Arm Embedded Technologies Pvt. Ltd.; Arm Norway, AS, and Arm Sweden AB.

Blackberry

BLACKBERRY, EMBLEM Design, QNX, AVIAGE, MOMENTICS, NEUTRINO and QNX CAR are the trademarks or registered trademarks of BlackBerry Limited, used under license, and the exclusive rights to such trademarks are expressly reserved.

Trademarks

NVIDIA and the NVIDIA logo are trademarks and/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

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