NVIDIA Grace Software with Generic Linux

Installation Reference

DI-11454-001_01 | December 2023

Document History

DI-11454-001_01

Version	Date	Description of Change
01	December 18, 2023	Initial Release

Table of Contents

Introduction

NVIDIA® Grace systems are capable of running a variety of Linux distributions that support the AArch64 architecture. With the proper kernel support and configurations in place, you can run a Linux distro of your choice and take advantage of the advanced Grace features.

This document provides information about how to install a generic Linux distribution on Grace systems and recommends solutions to issues that might occur.

Attention: Although it is possible to use other Linux distributions, only Ubuntu 22.04, RHEL 9.2, and SLES 15 SP5 have been tested and pre-validated by NVIDIA for Grace platforms.

Related Documentation

Refer to the Grace Release Notes for more information about Grace platforms.

Prerequisites

This section lists the required (or recommended) prerequisites:

Linux Kernel

One of the key components to having a successful experience with a generic Linux distro on the Grace platform is the Linux kernel version. Generally speaking, the closer that the kernel version for your distribution of choice is to the most recently released upstream Linux kernel version, the less likely it is that an issue will be encountered. The majority of Grace enablement is present in Linux kernel v6.2 and earlier and the majority of fixes for issues found post-enablement are present in Linux kernel versions 6.4 and 6.5. NVIDIA recommends selecting Linux distributions that directly (or indirectly via backport) use one of these kernel versions (or later).

Additionally, the configuration of the kernel is also important. These settings dictate the scale of the platform that is supported, the features of the hardware that are exploited, and can also greatly impact the overall performance of the system given. While the upstream arm64 defconfig can be a good starting point for these settings, NVIDIA recommends consulting the *Grace Platform Software Support* document, the *NVIDIA Grace Performance Tuning Guide*, and the *NVIDIA GH200 Benchmarking Guide*. These documents

can be used to help determine whether the kernel for your Linux distro is adequate for supporting the Grace platform and also list required and recommended kernel configuration settings.

NVIDIA Reference Kernel

As an additional resource, NVIDIA recommends consulting the public git repository for the Ubuntu linux-nvidia kernel variant. This has all of the patches required for the Grace platform and can be considered as a reference for the system software stack. It provides an additional stream of kernel patch and configuration information and in some circumstances may be more up-to-date than the *Grace Platform Software Support - Software Patches and Configurations* (NVOnline: **1106744**) guide. Additionally, as a Git repository, it allows for cherry picking specific commits which some users may find useful, those that roll their own kernel for example.

Installing Linux

Most Linux distributions support several installation methods. This section describes how to initiate an installation on the Grace system remotely through facilities hosted on the BMC. **Before you install**, review the platform-dependent workarounds section in this document to determine whether there are any modifications that are required for your environment. That section is also a good place to start troubleshooting should you encounter any issues with getting your Linux distribution running on the Grace platform.

Obtaining a Linux ISO

Obtain the ISO for Arm® image (aarch64) for your Linux distribution and store it on the local disk of the system that will be serving the media. Many Linux distributions offer both a full DVD ISO and a network-based ISO to support different installation strategies and preferences. Where possible, NVIDIA recommends using network-based installations due to the smaller storage footprint of the ISO and because these installations are typically faster. In particular, while the virtual media installation method described in this section is convenient, it sometimes has difficulty serving large ISO images. This can cause issues for some installers, Ubuntu for example, that have been shown to be very picky with respect to latency and timeouts.

Remotely Booting a Linux ISO Image

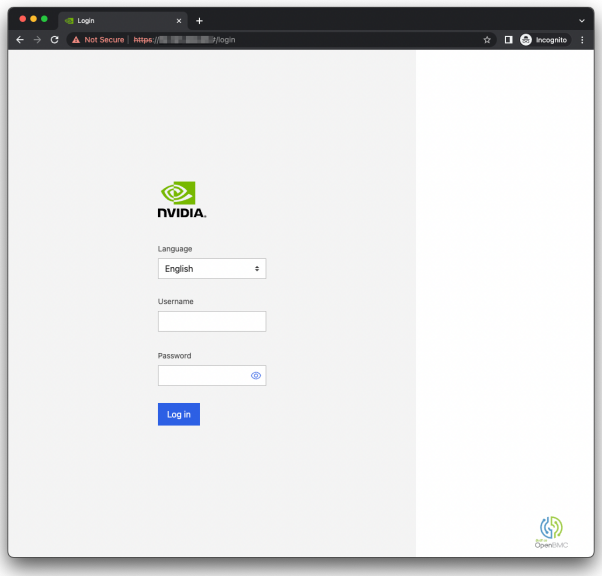
Attention: Here is some important information to know **before** you begin:

- The Grace BMC comes with default login credentials, and NVIDIA recommends that you create a unique user ID and password. Contact the system vendor if you have issues logging into the BMC.
- As a performance consideration, for networks that are distributed across a wide geographical area, such as a corporate VPN, we recommend that you use a browser from a host near the target server.

  This can be accomplished by using a remote application software solution, for example VNC, or by exporting the application X session and tunneling over SSH.
- This sequence is intended for the NVIDIA reference BMC, and your results might vary when using an IBV BMC.

1. Connect to the BMC.
   a. Open a browser in your LAN, navigate to `https://<BMC-IP-address>/`, and log in.

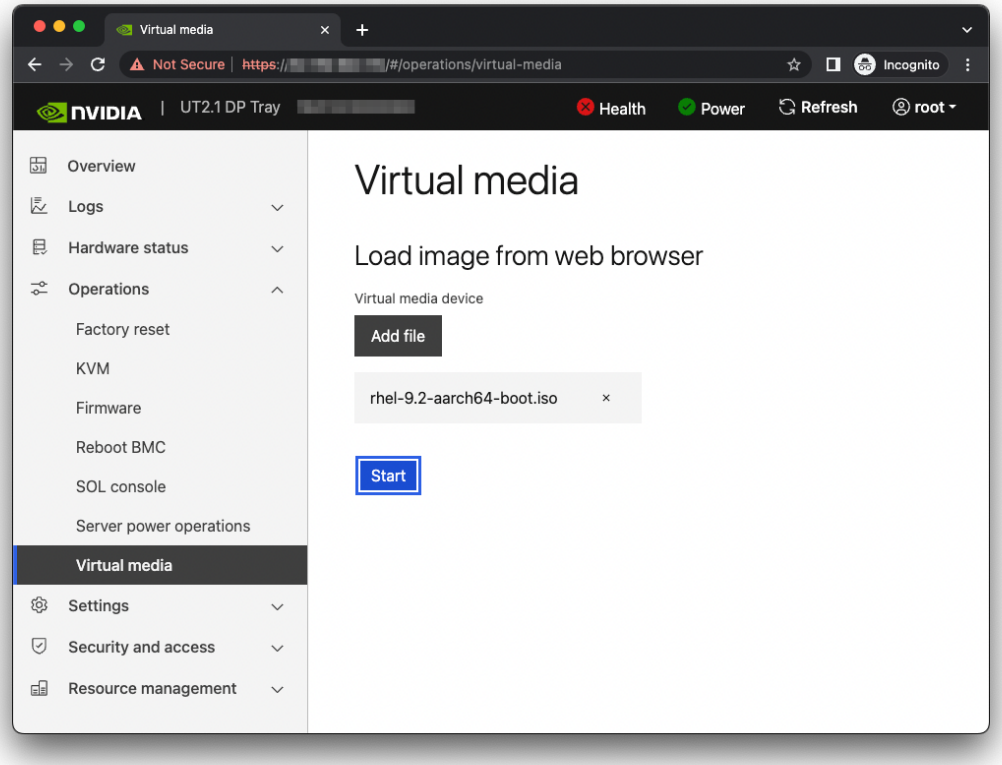
Figure 1. BMC Login



2. Download the ISO image to a location that can be accessed by the browser.
3. Set up the ISO image as virtual media.
   a. From the left hand menu, expand **Operations**.

b. Select **Virtual media**.
c. Click **Add file**, navigate to the downloaded ISO image, and select it.
d. To begin serving the ISO image to the target server, click **Start**.

Figure 2. BMC Virtual Media



4. Boot from the virtual media.
   Typically, the default boot order does not boot the CD-ROM image. You can change this in the BIOS or as a one-time option in the boot menu.
   a. Connect to the console.
      i. From the left hand menu, expand **Operations,** and select the SOL **console**.
      ii. Alternatively, SSH to the BMC from a terminal emulator, log in, and at the prompt, run the `obmc-console-client` command.
   b. To bring up the boot menu, press **Escape** or **F11** at the beginning of the boot process.

Figure 3. Console Splash Screen

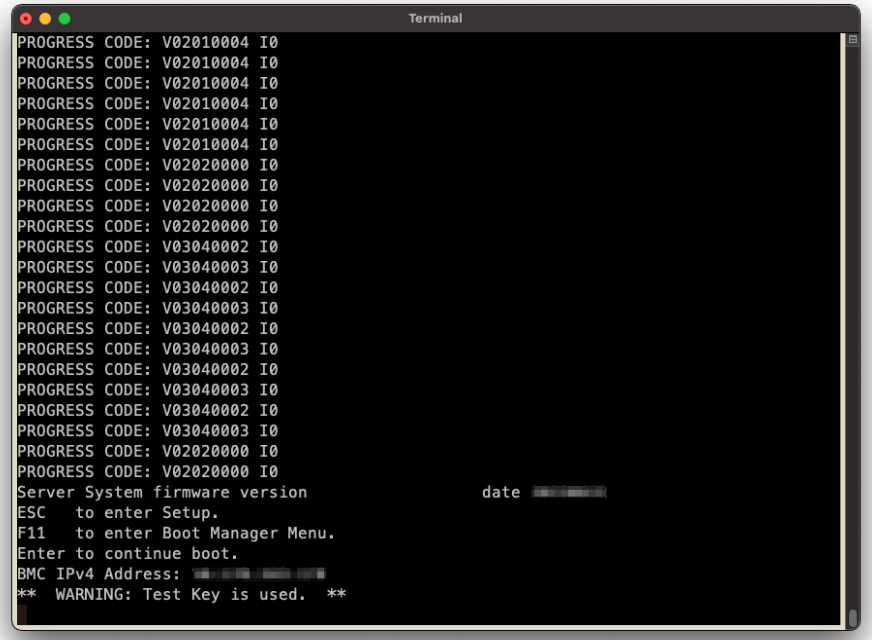
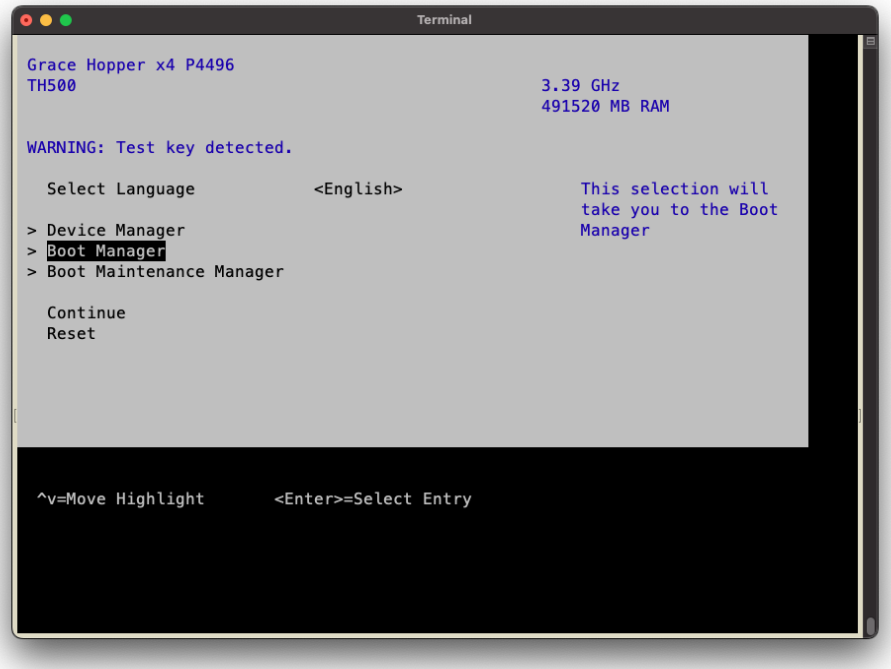
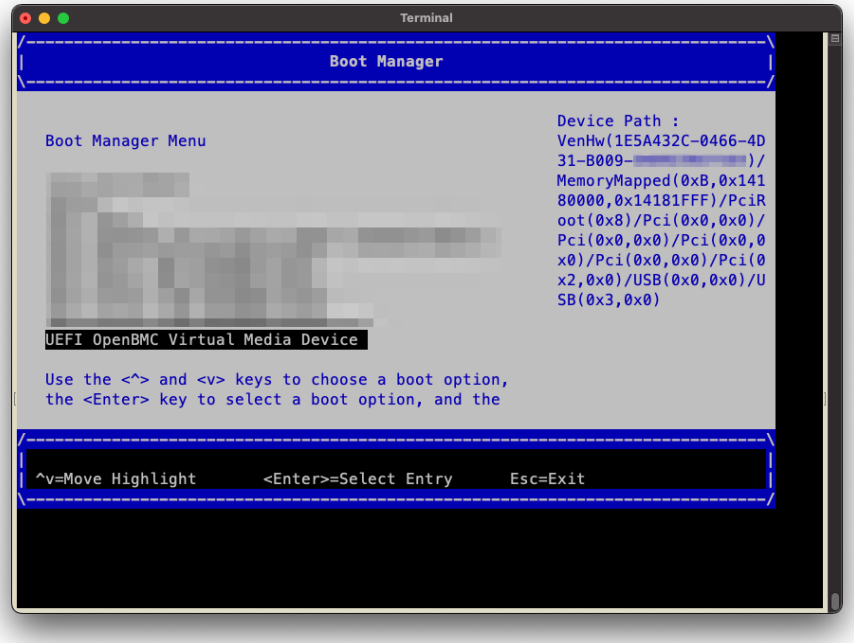


Figure 4. Boot Manager Menu



c. In the boot menu, select **UEFI OpenBMC Virtual Media Device** as the boot device and press **Enter.**
d. Follow the instructions in Installing Linux.

Figure 5. Boot Manager



Installing Linux

Prerequisites: This section assumes you have already booted the Linux ISO image.

- After booting the ISO image, most Linux distributions will display the GRUB menu for the installer and select a default option after a timeout period.

  While the menu entries will vary, there is typically an **Install** entry and a **Rescue** entry. The contents of these entries dictate the kernel to load and which boot options to enable. These can be viewed and/or modified by pressing the **e** key.
- The kernel parameters for workarounds and tunables can be applied during the installation by modifying the **Install** GRUB entry.

Caution: Linux distros will often carry these parameters forward to the installed system, which may or may not be desirable. Advanced installers may also interpret kernel parameters specified at install time and carry them forward in a different manner. For example, a blacklisted module may continue as kernel parameter while other, more advanced installers, may move the blacklist to a /etc/modprobe.d/*.conf script.

- Some implementations of the BMC virtual media interface suffer from slow performance, which can lengthen the installation process and possibly cause some Linux installers to experience timeout failures.

  NVIDIA recommends using a network installation method where possible. This can include a hybrid approach, where the initrd and kernel are loaded from virtual media and then the appropriate parameters applied at the **Install** GRUB entry to perform the installation over the network from the ISO directly or from a repository hosted on the network. For example, Ubuntu supports accessing an ISO using HTTP with the `url=http://path-to-image.iso` parameter and RHEL support accessing an ISO using NFS with the `inst.repo=nfs:[options:]<server>:/<path>` parameter.

Note: The UEFI shell supports the ifconfig and ping network utilities. These can be helpful when troubleshooting network configuration issues.

Appendix A: Known Issues

This section highlights items that are documented elsewhere but are likely to be encountered during the installation process or in your early days with a freshly installed Grace system. This is not an exhaustive list but rather focuses on items that NVIDIA has encountered multiple times in the labs and may be helpful to new users of the Grace platform.

A.1 CPU Frequency Governor

Some distros default the CPU frequency governor to a setting other than `performance`. This may result in overall system performance that does not meet expectations. NVIDIA recommends setting the default frequency governor for the Grace platform to the performance governor, which sets the frequency to the maximum value for that core. For most distros, this can be accomplished with the `cpufreq.default_governor=performance` kernel parameter. See the Section 4.3 of the *NVIDIA Grace Performance Tuning Guide* for further details.

A.2 Network Time Protocol

Some distros do not enable the network time protocol (NTP) by default. This can result in inconsistent time settings on the system that can lead to unintended side effects. For example, SSL certificates may fail verification. NVIDIA recommends that you enable NTP on your distro of choice.

Note: A valid date and time may be required to install your distro on the Grace platform.

A.3 Page Size

Most distros install a kernel that uses the 4K page size by default. While the Grace platform supports both 4K and 64K page sizes, NVIDIA recommends using the 64K page size as it can benefit applications that have a large memory footprint. Many distros support a 64K kernel that can be installed with the package manager for the distro. Refer to *Page Size* in the *NVIDIA Grace Performance Tuning Guide* for more information.

Note: The page size of the current kernel can be displayed with `getconf PAGE_SIZE`.

Appendix B: Platform-Dependent Workarounds

Some Grace platforms require temporary (or permanent) alterations to their configurations to work around known issues, such as hardware errata. These workarounds are described in the following sections by the corresponding Grace platform.

Note: Depending on the kernel support for your distro, at least one of these workarounds might be required to install Linux on the Grace platform.

B.1 All Grace Platforms

AST2600 BMC Workaround

Distros that are based on a Linux kernel earlier than v6.4 and do not carry this `ast` driver patch require a workaround on **all** Grace platforms to avoid undefined behaviors. The absence of this patch can manifest a variety of issues, including kernel hangs and distorted output from the on-board VGA port.

To workaround this issue, NVIDIA recommends blacklisting the `ast` driver. For most distros, this can be accomplished by adding the `modprobe.blacklist=ast` kernel parameter or by creating a file in /etc/modprobe.d that contains a blacklist directive. As a side effect of this

workaround, because the on-board VGA port is inaccessible, a serial console solution (for example, SOL) must be used for console access to the system.

CUDA Application Workaround

CUDA applications on the Grace-Hopper platform require ATS support, and currently, ATS is not enabled on the arm64 platform when IOMMU passthrough is enabled. NVIDIA is working with the Linux kernel community to resolve this issue.

Linux provides the `iommu.passthrough` kernel parameter to configure the DMA to use (or not use) the IOMMU to access the memory for addressing. If your Linux distro of choice sets the IOMMU in passthrough mode by default, for example CONFIG_IOMMU_DEFAULT_PASSTHROUGH=y, it will prevent CUDA applications from running. We recommend that you add the `iommu.passthrough=0` kernel parameter until this issue is resolved.

B.2 Multi-Socket Grace Platforms

NVIDIA Hardware Erratum T241-FABRIC-4 Workaround

Distros that are based on a Linux kernel earlier than v6.4 or do not carry this patch require a workaround on Grace systems with **three-and four-socket configurations**. The absence of this patch can manifest a variety of issues, including kernel hangs, timeouts, and other undefined behaviors. See NVIDIA hardware erratum T241-FABRIC-4 for more information.

To workaround this issue, NVIDIA recommends restricting the system to a one socket configuration. For most distros, this can be accomplished by adding the `nr_cpus=72` kernel parameter to limit the number of cores supported by the Linux kernel to the number of cores found in a Grace processor. As a side effect of this workaround, the server is reduced to one quarter of its total compute capacity.

Caution: Be careful when profiling a system with this temporary workaround.

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.

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 this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform 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 document. NVIDIA accepts no 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 document 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 document. 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 document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND 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. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. 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 Terms of Sale for the product.

Trademarks
NVIDIA, the NVIDIA logo, Grace, 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.

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.

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

Arm
Arm, 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.

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

Copyright
© 2023 NVIDIA Corporation & Affiliates. All rights reserved.

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