天数智芯

CUDA 程序迁移指南

版本：V4.0.0-MR

日期：2024.6.4

适用产品：智铠 50| 智铠 100

1 声明

1.1 版权声明

1.2 免责声明

天数智芯可以随时对本文档或本文档中描述的产品进行改进和/或更改。本文档包括与天数智芯产品有关的信息，作为说明典型应用的一种方式, 因此，不一定提供足以进行生产设计的完整信息。对于本文档中内容的准确性或完整性，天数智芯不做任何陈述或保证。

1.3 联系方式

地址：上海闵行区陈行公路 2168 号 3 幢

电话：021-68886607

网址：www.iluvatar.com

Contents

1 声明 **2**
1.1 版权声明 . . . 2
1.2 免责声明 . . . 2
1.3 联系方式 . . . 2

2 CUDA 程序迁移指南 **4**
2.1 修订记录 . . . 4
2.2 迁移概览 . . . 4
2.3 进行程序迁移 . . . 4
2.3.1 1. 直接编译 CUDA 项目源文件 . . . 4
2.3.2 2. 根据提示信息修改项目源文件 . . . 4
2.3.3 3. 进一步程序迁移建议 . . . 5
2.4 高级用法：使用 Makefile 进行迁移 . . . 5
2.5 高级用法：使用 CMake 进行迁移 . . . 6
2.5.1 安装天数智芯适配版 CMake . . . 6
2.5.2 使用 CMake . . . 7
2.6 示例项目 . . . 7
2.7 参考信息：与主流 GPU 通用计算模型的异同 . . . 7
2.8 常见问题 . . . 8

3 商标声明 **11**

2 CUDA 程序迁移指南

2.1 修订记录

- COREX01-MR400-BP01-01：2024/6/4

  文档本次发布与上一次发布 (COREX01-MR400-BP01-00) 相比，有以下更新：

  - 常见问题 新增天数卡计算结果和迁移前结果不同

2.2 迁移概览

天数智算软件栈与主流 GPU 通用计算模型兼容，对于 Linux 内核、thread hierarchy 和 memory hierarchy 有一致理解。天数智算软件栈包括许多主要的 CUDA 等效组件、特性、API 和算法，可助力您实现无痛迁移。天数智算软件栈支持常用的 CUDA API，所以大多数情况下，无需修改 CUDA 代码即能完成迁移。

2.3 进行程序迁移

2.3.1 1. 直接编译 CUDA 项目源文件

为了使 CUDA 程序运行在天数智芯加速卡上，您必须使用天数智算软件栈提供的编译器来重新编译 CUDA 程序。

1) 检查您是否已在使用天数智算软件栈提供的 Clang 编译器。运行结果应显示 `${ILUVATAR_SOFTWARE_ROOT}/bin/clang++`, 此处的 `${ILUVATAR_SOFTWARE_ROOT}` 指的是天数智算软件栈的安装路径。默认安装路径是 `/usr/local/corex/` 。

```
$ which clang++
```

2) 从 `.cu` 源文件构建可执行文件，以编译使用了 C++11 高级特性的 `example.cu` 文件生成 `example` 程序为例：

```
$ clang++ -std=c++11 example.cu -o example -lcudart
```

如需了解编译器详情，请参考《编译器使用指南》。

2.3.2 2. 根据提示信息修改项目源文件

编译过程可能会收到提示信息，这是由于天数智算软件栈与主流 GPU 通用计算模型兼容，但在实现细节上略有区别。具体修改项与您的 CUDA 项目源文件有关，以编译收到的提示信息为准。

可能需要的修改项包括：

- 将代码 device 部分的 double 数据类型替换成 float 数据类型。

  Tip

  修改代码是因为天数智芯加速卡对 double 数据类型仅提供有限支持。

- 修改项目文件夹中所有.cu 文件中对 `max` 和 `min` 的调用：

  - 用 `std::max` 替换 `max`
  - 用 `std::min` 替换 `min`

  Tip

  修改代码是为了绕过 Clang 编译器的已知问题。

- 修改项目文件夹中所有.cu 文件中 `<< <` 和 `<< <`，去掉多余的空格。

  Tip

  修改代码是因为 Clang 编译器不支持 `<< <` 和 `<< <`，只支持无空格的 `<<<` 和 `<<<` 。

- 替换或删除暂不支持的函数库和 API。您可以通过查阅文档了解天数智算软件栈目前支持的函数库和 API，例如，暂不提供对 SIMD Intrinsics 的支持。
- 如果编译过程链接了静态库，适配天数智芯加速卡时，建议修改成链接到动态库。

如需了解支持的 CUDA 特性和 API 列表，见参考信息：与主流 GPU 通用计算模型的异同 和《软件栈 API 参考》。

2.3.3 3. 进一步程序迁移建议

完成 CUDA 程序从其它平台迁移到天数智芯加速卡后，为了针对天数智芯加速卡优化 CUDA 程序性能，您还可以对项目源文件进行以下修改：

- 线程束（warp size）

  天数智算软件栈适用的线程束大小是 64，您可以根据需要，将项目源文件中定义的 warp size 修改为 64。

- 线程块中包含的线程数量

  天数智算软件栈线程块中包含的线程最大值为 4096，您可以根据需要，调整 CUDA 程序中 block 三元组的设置，例如：

```
dim3 blockSize(1,1,4096);
kernel<<<numBlocks, blockSize>>>(N, x, y);
```

2.4 高级用法：使用 Makefile 进行迁移

对于大型 CUDA 程序，您可以选择使用 Makefile 提高编译效率。Make 工具最主要最基本的功能就是通过创建 Makefile 文件来描述源程序之间的相互关系并自动维护编译工作，您创建好 Makefile 后运行 **make** 命令就能为大型项目进行自动化编译。

如您想修改 Makefile 迁移大型 CUDA 程序，可参考以下步骤：

1. 修改 Makefile 中调用的编译器，需要从其它编译器（如 NVCC）指向天数智芯编译器，如您使用默认安装路径，天数智芯编译器的具体路径是 `/usr/local/corex/bin/clang++` 。

   Tip

   由于天数智芯编译器是基于开源 LLVM 项目开发的，因此调用天数智芯编译器的命令是 **clang** 或 **clang++** 。

2. 清除 Make 缓存：在下次编译前，删除 CMakeCache 中新产生的所有文件：CMakeCache.txt，CMakeFiles，Makefile。

   Important

   必须执行这一步，否则会影响下次编译的正确性。

3. 重新编译：使用 **make** 命令执行 Makefile，例如：

   ```
   $ make
   ```

2.5 高级用法：使用 CMake 进行迁移

CMake 可以让程序员通过一个与开发平台无关的 `CMakeLists.txt` 文件来定制整个编译流程，然后再根据目标用户的平台进一步生成所需的 Makefile 和工程文件。

如您既有的 CUDA 程序是用 CMake 和配置文件 `CMakeLists.txt` 编译的，您可使用天数智芯适配版 CMake 迁移这些 CUDA 程序。

天数智芯适配版 CMake 基于官方 CMake V3.25.2，与官方 CMake 的区别是针对 find_packages(CUDA) 以及 enable_language(CUDA) 两种模式做了以下天数智芯适配项：

1. 调用使用天数智芯编译器替代了官方版中的 NVCC 编译器
2. 过滤了不支持的编译选项
3. 禁用了 CUDA_SEPARABLE_COMPILATION 属性

除此之外，天数智芯适配版 CMake 和官方 CMake 功能一致。

Tip

安装完天数智芯适配版 CMake 后，您还可以通过设置环境变量 CMAKE_CUDA_COMPILER_ID 关闭以上对 CMake 的天数智芯适配项：

- export CMAKE_CUDA_COMPILER_ID=”NVIDIA” 或 export CMAKE_CUDA_COMPILER_ID=”Clang”： 等同于官方 CMake V3.25.2
- export CMAKE_CUDA_COMPILER_ID=”ILUVATAR”： 默认值，对应天数智芯适配版 CMake

2.5.1 安装天数智芯适配版 CMake

1. 联系您的应用工程师获取安装包：

   - x86 平台：`cmake-3.25.2-corex.{v.r.m}-linux-x86_64.sh`
   - ARM 平台：`cmake-3.25.2-corex.{v.r.m}-linux-aarch64.sh`

2. 执行以下命令安装天数智芯适配版 CMake，以 x86 平台为例：

```
$ sudo bash cmake-3.25.2-corex.{v.r.m}-linux-x86_64.sh --skip-license --
↪  prefix=${COREX_CMAKE_PATH}
```

安装说明：

- **--skip-license** 是指跳过使用许可展示界面并接受该使用许可，**--prefix** 指定安装路径。
- 安装后可通过执行 **${COREX_CMAKE_PATH}/bin/cmake --version** 检查安装是否成功，版本号应包含 corex 字样。
- 可把 `${COREX_CMAKE_PATH}/bin` 加入 PATH 环境变量，注意如果 PATH 中已经包含了其它版本的 CMake 路径，需正确设置路径优先级。

2.5.2 使用 CMake

使用 CMake 的前提是您要编译的 CUDA 程序已有相应的 CMake 配置文件 `CMakeLists.txt` ，具体请咨询负责该程序建构的工程师。

使用天数智芯适配版 CMake 的参考步骤如下：

1. 通过设置环境变量来指定适配 CUDA 程序的平台，智铠加速卡必须指定

   export CMAKE_CUDA_ARCHITECTURES=ivcore11。若不指定，则默认是 ivcore10，表示天垓加速卡。
2. 执行命令 **cmake ${CMakeLists_PATH}** 从 `CMakeLists.txt` 生成 Makefile，${CMakeLists_PATH} 是 `CMakeLists.txt` 所在的路径。
3. 运行 **make** 命令进行编译。

2.6 示例项目

本次发布的示例提供了 mninstCUDNN 项目，供您进行以下参考：

- 您可按照该项目 README 的步骤说明，用 Makefile 或 CMake 进行 mninstCUDNN 程序的迁移。
- 您可参考该项目 Makefile 或 `CMakeLists.txt` 的内容，进而为您的 CUDA 程序编写 Makefile 或 `CMakeLists.txt` 。

以 root 用户的示例默认安装路径 `/root/corex-samples-{v.r.m}_x86_64` 为例，mninstCUDNN 项目路径是 `/root/corex-samples-{v.r.m}_x86_64/samples/mnist-cudnn` 。

2.7 参考信息：与主流 GPU 通用计算模型的异同

天数智算软件栈与 CUDA 主流版本兼容，支持的主要功能包括但不限于：

- Devices, context and module management (Driver API)
- Most device-side math built-in functions and intrinsic
- Memory management including shared memory manipulation
- Stream and event management

- CUDA-style topology of thread organization (grid and block), and kernel coordinate hierarchy (threadIdx, blockIdx, blockDim, gridDim)
- Kernel launching including using <<< >>> execution configuration syntax
- Barrier synchronization and cross-lane instructions
- Error reporting

从其它平台迁移到天数智芯加速卡时，您需要使用天数智算软件栈进行程序迁移，请注意以下天数智算软件栈和 CUDA Toolkit 的区别：

	天数智算软件栈	CUDA Toolkit
线程束	64	32
线程块中包含的最大线程数量	4096	1024
预定义宏	CUDA_ARCH=ivcore1 1 __ILUVATAR__	__NVCC__ 等

2.8 常见问题

1. error: no matching function for call to 'max'
2. error: __float128 is not supported on this target
3. 天数卡计算结果和迁移前结果不同

问题 1: 遇到报错 `error: no matching function for call to 'max'`

> **问题现象**
>
> ```
> mnist-cudnn/src/layer.cu:588:19: error: no matching function for call to 'max'
> workspace_size = max(workspace_size, temp_size);
> ```
>
> **问题分析**
>
> 这是 Clang 编译器的已知问题。
>
> **解决方法**
>
> 用 `std::max` 替换 `max`, 即把 `workspace_size = max(workspace_size, temp_size)` 修改成 `workspace_size = std::max(workspace_size, temp_size)`。

问题 2: 遇到报错 `error: __float128 is not supported on this target`

> **问题现象**

```
In file included from /usr/lib/gcc/x86_64-linux-gnu/7.5.0/../../../../include/c++/7.5.0/
 ↪  cmath:47:
/usr/lib/gcc/x86_64-linux-gnu/7.5.0/../../../../include/c++/7.5.0/bits/std_abs.h:102:7:
 ↪  error: __float128 is not supported on this target
abs(__float128 __x)
      ^
/usr/lib/gcc/x86_64-linux-gnu/7.5.0/../../../../include/c++/7.5.0/bits/std_abs.h:101:3:
 ↪  error: __float128 is not supported on this target
__float128
```

问题分析

CMake 自动加上了 **-std=gnu++14** 编译选项。

解决方法

如果 **-std=gnu++14** 编译选项对本次编译并非必须，您可以在 CMake 的配置中删除该选项。

例如，mnistCUDNN 项目并不需要设置 **-std=gnu++14** 编译选项，您可以将 `mnist-cudnn/build/CMakeFiles/train.dir/flags.make` 中以下内容：

```
CUDA_FLAGS =  --std=c++14 -O3 -DNDEBUG --cuda-gpu-arch=ivcore11 --cuda-path=/usr/local/corex
 ↪  -O3 -DNDEBUG --cuda-path=/usr/local/corex -std=gnu++14
```

改成：

```
CUDA_FLAGS =  --std=c++14 -O3 -DNDEBUG --cuda-gpu-arch=ivcore11 --cuda-path=/usr/local/corex
 ↪  -O3 -DNDEBUG --cuda-path=/usr/local/corex
```

问题 3：天数卡计算结果和迁移前结果不同

问题原因

代码中出现浮点数据类型的超大常量，而天数编译器和 nvcc 对代码处理略有不同。

超大常量可能会引发 undefined behavior，天数编译器删除了与该超大的常量相关的指令，这种处理和 LLVM Clang 编译器处理一致。

解决方法

如您希望避免因超大常量引发的 undefined behavior，可加上 **-fno-strict-float-cast-overflow** 编译选项，该选项将越界的值截取为取值范围内，从而避免 undefined behavior。可参考以下示例：

```
__global__ void convertLongToFloat(long *result, long *result2) {
    long maxValue = 9223372036854775807ll;
    float res = static_cast<float>(maxValue);
    *result= static_cast<long>(res);
    *result2 = static_cast<long>(9223372000000000000ll);
}
```

天数编译器默认删除 `*result= static_cast<long>(res);`。

如您在编译时加上 **-fno-strict-float-cast-overflow** 选项，则越界的值会被截取为取值范围内，由此也

保留了 *result= static_cast<long>(res);。

3 商标声明

- 天数智芯、天数智芯 logo、Iluvatar CoreX 等商标、标识、组合商标为上海天数智芯半导体有限公司之注册商标或商标，受法律保护。
- 除了天数智芯的注册商标外，本内容中使用的其他产品名称及标志仅用于识别目的，该等名称及标志可能是归属于其各自公司的商标。我们否认对该等名称及标志的所有权利。
- CentOS 标识为 Red Hat 公司的商标。
- Docker 为 Docker 公司在美国和其他国家的商标或注册商标。
- Linux 为 Linus Torvalds 在美国和其它国家的注册商标。
- NVIDIA 和 CUDA 为 NVIDIA 公司在美国和/或其它国家的商标和/或注册商标。
- PyTorch 为 Facebook 公司的商标。
- TensorFlow 为 Google 公司的商标。
- Ubuntu 为 Canonical 公司的注册商标。