天数智芯

编译器使用指南

版本：V4.0.0-MR

日期：2024.6.4

适用产品：智铠 50| 智铠 100

此文档仅供【zhangyj@skysolidiss.com.cn-天固信息安全系统（深圳）有限公司】查阅，请勿分享他人

1 声明

1.1 版权声明

版权所有。未经天数智芯书面许可，不得以任何形式或方式将本文档的任何部分复制，传播，转录或翻译成任何语言。

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

此文档仅供【zhangyj@skysolidiss.com.cn-天固信息安全系统（深圳）有限公司】查阅，请勿分享他人

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 编译器使用指南 **4**
2.1 修订记录 . . . 4
2.2 概览 . . . 4
2.3 快速入门 . . . 4
2.4 示例 . . . 4
2.5 宏 . . . 5
2.6 高级用法 . . . 6
2.6.1 编译阶段 . . . 6
2.6.2 输入和输出文件 . . . 7
2.6.3 编译步骤示例 . . . 7
2.6.3.1 1. 生成 LLVM IR . . . 7
2.6.3.2 2. 将 LLVM IR 编译到 OBJ 文件 . . . 8
2.6.3.3 3. 将 OBJ 文件链接到 `.cubin` . . . 8
2.6.3.4 4. 从 `.cubin` 文件中创建 `.fatbin` 文件 . . . 9
2.6.3.5 5. 编译主机部分的代码并链接到 `.fatbin` . . . 9
2.6.3.6 6. OBJ 文件链接到天数智芯 CUDA 运行环境 . . . 9
2.7 编译选项 . . . 10
2.7.1 **clang** 和 **clang++** 选项 . . . 10
2.7.2 **PTXToLLVM** 选项 . . . 10
2.7.3 **llc** 选项 . . . 11
2.7.4 **lld** 选项 . . . 11
2.7.5 **fatbinary** 选项 . . . 12
2.8 参考 . . . 12

3 商标声明 **13**

此文档仅供【zhangyj@skysolidiss.com.cn-天固信息安全系统（深圳）有限公司】查阅，请勿分享他人

2 编译器使用指南

2.1 修订记录

- COREX01-MR400-UG02-01：2024/6/4
  - 编译选项 新增备注，提示用户 Clang 和 nvcc 的默认选项可能存在少量不同，编译选项请按需调整
- COREX01-MR400-UG03-00：2024/4/3

  本次文档发布内容与 V3.2.0-MR 文档相比 (COREX01-MR320-UG03-00)，有以下更新：
  - 强调选项值 ivcore10 需修改为 **ivcore11**，指代智铠加速卡，避免用户使用默认值

2.2 概览

天数智算软件栈提供的编译器是针对天数智芯硬件架构设计和优化的。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++**。LLVM 项目包括 Clang 子项目，Clang 是 LLVM 原生的 C 语言家族（C, C++, Objective C/C++, CUDA 等）的编译器。

本指南旨在强调天数智芯编译器的用法。关于 Clang 等 LLVM 工具的一般用法，请参考 LLVM 官方文档。

2.3 快速入门

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
   ```

具体的编译实例，请参见示例。

2.4 示例

以下示例是以编译天数智算软件栈提供的 `saxpy.cu` 示例代码为例，构建名为 `saxpy` 的可执行文件。假设使用默认安装路径 `/usr/local/corex/` ，请注意执行示例命令需要您对 `/usr/local/corex/` 有写权限。

1. 切换到示例所在的目录。

```
$ cd /usr/local/corex/examples/cuda
```

2. 运行 **clang++** 命令及相应的必需和可选参数，生成名为 saxpy 的可执行文件。

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

Note

如果您安装了其他版本的 Clang 编译器，请确保您调用的 Clang 编译器路径是 /usr/local/corex/bin/ 。如需要了解更多信息，请参阅快速入门。

示例中的选项详细说明如下:

- **-std=c++11**: 可选项，本示例中 saxpy.cu 实际未使用 C++11 高级特性。如您的代码中使用了某些 C++ 高级特性，则需要使用 **-std**，具体可参考 C++ Support in Clang 。
- **-Wall**: 可选项。启用某些用户认为有问题的构造器的所有警告。这也启用了针对特定语言的警告。
- **-lcudart**: 必选项。确保链接 CUDA 运行库；否则，会出现一些无法解析的符号。

其它相关命令选项:

- **--cuda-gpu-arch=< value >**: 可选项。默认是 **--cuda-gpu-arch=ivcore10**，指代天垓加速卡；使用智铠加速卡，需设置为 **--cuda-gpu-arch=ivcore11**。
- **-o <your output name>**: 可选项。用于设置输出的文件名。默认是 a.out 。

Important

需要您特别注意的是：选项中的默认值 **--cuda-gpu-arch=ivcore10** 指代的是天数智芯的 **天垓加速卡**。当前，您使用的加速卡是天数智芯的 **智铠加速卡**，因此需要将 **--cuda-gpu-arch** 的值修改为 **ivcore11**，即 **--cuda-gpu-arch=ivcore11**。

下文若有类似情况，请一并遵循此处的提示。

编译器默认开启了以下选项:

- **-I/usr/local/corex/include**: 用于在预处理期间设置搜索头文件的目录。
- **--cuda-path=/usr/local/corex**: 指定天数智算软件栈的路径。
- **-L/usr/local/corex/lib64**: 指定库文件路径。

Tip

您可参考 /usr/local/corex/examples/cuda/Makefile 路径下的文件，您可以使用 **make** 完成编译:

```
$ make VERBOSE=1
```

您可以根据实际需求对 Makefile 进行自定义。

2.5 宏

__ILUVATAR__ 当使用天数智芯编译器为天数智芯平台进行设备代码编译时，该宏被定义。你可以使用 __ILUVATAR__ 来确定是针对天数智芯平台进行编译，或是针对其他平台进行编译。

此外，当该宏被定义时，wrap size 设置为 64，否则，wrap size 为 32。

__CUDA_ARCH__ CudaArch GPU 架构参数。您可以使用该宏来确定算力。该宏的值由 **--cuda-gpu-arch=<value>** 选项定义。对于天数智芯编译器，**--cuda-gpu-arch** 的默认值为 **ivcore10**，表示天垓加速卡；使用智铠加速卡，需将该值设置为 **ivcore11**。

2.6 高级用法

2.6.1 编译阶段

编译过程可以分为不同的阶段，您可通过 **clang** (或 **clang++**), **llc**, 或 **lld** 等天数智芯编译器工具执行某个编译阶段。

编译阶段	描述	调用组件	输入文件	输出文件
预处理	编译 C 程序时，编译器在这个阶段进行 #includes 和 #define 宏替换	clang	.c, .cc, .cpp, .cxx	.c, .cc, .cpp, .cxx
生成 LLVM IR	从源文件生成 LLVM IR	clang	.cu	.ll
生成 LLVM IR	从 PTX 生成 LLVM IR	PTXToLLVM	.ptx	.ll
编译到 OBJ 文件	将 LLVM IR 编译到 OBJ 文件	llc	.ll	.o
从 OBJ 文件生成 CUDA 二进制文件	链接 OBJ 文件以生成 CUDA binary 文件	lld	.o	.cubin
生成 fatbin 文件	从 CUDA binary 文件创建 fatbin 文件	fatbinary	.cubin	.fatbin
汇编	编译主机端代码并且将 fatbin 包入其中生成主机端 OBJ 文件	clang	.c, .cc, .cpp, .cxx, .fatbin	.o
创建可执行文件	链接以创建可执行文件	lld	.o	executable (a.out by default)

术语说明：

- LLVM IR （Intermediate Representation）：LLVM 中间表示，连接着编译器前端和编译器后端。
- PTX（Parallel Thread eXecution，PTX）：是编译后的 GPU 代码的一种中间形式，它可以再次编译为原生的 GPU 代码。
- lld：是 LLVM 项目的链接器。
- llc：是 LLVM 项目的静态编译器。
- PTXToLLVM：从 PTX 生成 LLVM IR 的工具，使包含 PTX 的源代码可以经编译在天数智芯平台上运行。

此文档仅供【zhangyj@sky-chips.com.cn-天固信息安全系统（深圳）有限公司】查阅，请勿分享他人

2.6.2 输入和输出文件

天数智算软件栈在输入输出文件类型上兼容主流的 GPU 编程模型。

文件扩展名	描述
.cu	CUDA 源文件, 包含主机代码和设备端的函数
.c	C 源代码
.cc, .cxx, .cpp	C++ 源代码
.ptx	PTX 中间汇编文件
.cubin	单个 GPU 架构的 CUDA 二进制文件
.fatbin	包含多个 PTX 和.cubin 文件的 CUDA 二进制文件
.o	OBJ 文件
.a	库文件
.res	资源文件
.so	共享 OBJ 文件

2.6.3 编译步骤示例

示例 可以扩展到下面的编译步骤，适用于编译更复杂的项目。假设使用默认安装路径 `/usr/local/corex/` ，请注意执行示例命令需要您对 `/usr/local/corex/` 有写权限。

2.6.3.1 1. 生成 LLVM IR

根据输入文件的类型，您可以选择 1-a 或 1-b 来生成 LLVM IR。

2.6.3.1.1 1-a. 为 .cu 文件中的 CUDA 代码部分生成 LLVM IR

假设我们要编译 `saxpy.cu` , 它包含设备内核和主机部分代码。执行以下命令，为 CUDA 代码部分生成一个名为 `saxpy.ll` 的文本 LLVM IR 文件:

```
$ cd /usr/local/corex/examples/cuda
$ clang++ -S -emit-llvm --cuda-gpu-arch=ivcore11 --cuda-device-only -std=c++11 -Wall saxpy.cu -o
 ↪  saxpy.ll
```

选项:

- **-S -emit-llvm**: 必选项。生成一个 LLVM IR 文件。
- **--cuda-gpu-arch=ivcore11**: 必选项。针对智铠加速卡生成文件。

此文档仅供【zhangyj@skysolidiss.com.cn-天固信息安全系统（深圳）有限公司】查阅，请勿分享他人

- **-o <LLVM IR filename>**: 可选项。设置生成的 LLVM IR 文件的名称。
- **--cuda-device-only**: 仅生成 CUDA 代码部分。如果省略此选项，还将生成主机部分。

2.6.3.1.2 1-b. 为 .ptx 文件中的 PTX 生成 LLVM IR

假设设备代码是 PTX 代码 (saxpy.ptx) 而不是 CUDA 代码 (saxpy.cu)。执行以下命令编译 saxpy.ptx 。该命令可以作为步骤 I-a 的替代，以生成所需的 IR 文件 saxpy.ll 。请注意，saxpy.ptx 文件是在另一个目录中提供的，以免混淆常见步骤 1-a。

```
$ cd /usr/local/corex/examples/ptx
$ PTXToLLVM saxpy.ptx saxpy.ll --iluvatarTarget
```

选项:

- **--iluvatarTarget**: 从 PTX 生成 LLVM IR，使包含 PTX 的源代码可以经编译在天数智芯平台上运行。

2.6.3.2 2. 将 LLVM IR 编译到 OBJ 文件

使用 LLVM 静态编译器 **llc** 编译 saxpy.ll 。

```
$ cd /usr/local/corex/examples/cuda
$ llc -march=bi -filetype=obj saxpy.ll -o saxpy.cuda.o
```

选项:

- **-march=bi**: 必选项。针对天数智芯加速卡进行编译。
- **-filetype=obj**: 必选项。生成 OBJ 文件，而不是默认的汇编文件。
- **-o <object filename>**: 可选项。设置输出 OBJ 对象的名称。

2.6.3.3 3. 将 OBJ 文件链接到 .cubin

使用 LLVM 链接器 **lld** 编译 saxpy.cuda.o。

```
$ lld -flavor ld.lld --no-warn-missing-entry --no-undefined saxpy.cuda.o -o saxpy.cubin
```

选项:

- **-flavor ld.lld**: 必选项。
- **--no-warn-missing-entry**: 可选项。当 entry 缺失时不提示警告信息。
- **--no-undefined**: 必选项。即使链接器正在创建共享库，也要报告未解析的符号。
- **-o <cubin filename>**: 可选项。设置输出的.cubin 文件名。

此文档仅供【zhangyj@skysolidiss.com.cn-天固信息安全系统（深圳）有限公司】查阅，请勿分享他人

2.6.3.4 4. 从 .cubin 文件中创建 .fatbin 文件

使用 **fatbinary** 工具编译 saxpy.cubin.

```
$ fatbinary --cuda --64 --image=profile=ivcore11,file=saxpy.cubin --create saxpy.fatbin
```

选项:

- **--cuda**: 指定这是一个 CUDA 编译。
- **--64**: 指定这是一个 64 位的编译。
- **--image=profile=ivcore11,file=saxpy.cubin**: 设置带有配置文件的编译映像列表。语法为 **--image=< keyword >=< string >** ，关键字含义如下：
  - **file** ： 指定 ELF 文件，是 ld 命令生成的 .cubin 文件。
  - **profile** ： 指定架构，支持设置为 **ivcore11**，代表智铠加速卡。
- **--create <fatbin filename>**: 必选项。指定输出的 fat 二进制文件名。

2.6.3.5 5. 编译主机部分的代码并链接到 .fatbin

在前面的步骤中，我们只编译 CUDA 代码部分，接下来我们要编译主机部分。

为了从主机端生成 OBJ 文件并从 .fatbin 收集必要的信息，文件 saxpy.fatbin 和 saxpy.cu (使用其主机部分) 都必须包含在以下命令中。

```
$ clang++ -c -std=c++11 -Wall --cuda-host-only -Xclang -fcuda-include-gpubinary -Xclang
↪  saxpy.fatbin saxpy.cu -o saxpy.o
```

选项:

- **-c -std=c++11**: 使用 C++11 模式。
- **--cuda-host-only**: 忽略代码中的 CUDA 设备部分，只编译主机部分。
- **-Xclang**: 必选项。将 [arg] 值传递给 **clang** 命令。
- **-fcuda-include-gpubinary**: 通知链接器它将把一个 fat 二进制文件合并到输出 OBJ 文件中。
- **-o <host object filename>**: 设置主机 OBJ 文件名。

2.6.3.6 6. OBJ 文件链接到天数智芯 CUDA 运行环境

这是最后一步。在这一步之后，您将获得名为 saxpy 的可执行文件。

```
$ clang++ -lcudart saxpy.o -o saxpy
```

选项:

- **-lcudart**: 链接 libcudart.so 。
- **-o <executable name>**: 设置可执行文件名称。

此文档仅供【zhangsj@sky-solidiss.com.cn-天固信息安全系统（深圳）有限公司】查阅，请勿分享他人

2.7 编译选项

2.7.1 clang 和 clang++ 选项

Clang 编译器是来自 LLVM 项目，为 C 语言家族，如 C、C++、Objective C/C++、CUDA 等提供了编译器前端和基础工具。**clang** 可用于编译 C 程序，**clang++** 用于编译 C++ 程序。

天数智芯编译器支持所有的 Clang 13 选项。有关 Clang 选项的完整列表，请参阅 Clang 编译器选项。

以下是一些常用的选项，供您参考。

Clang 选项	描述
--cuda-compile-host-device	为主机和设备编译 CUDA 代码 (默认)。对非 CUDA 编译没有影响。
--cuda-device-only	只为设备编译 CUDA 代码。
--cuda-gpu-arch=< value >	指定目标 CUDA GPU 架构。为 NVIDIA GPU 指定 **sm_xy** ，或为智铠加速卡指定 **ivcore11**。默认为 **ivcore10**，代表天垓加速卡。
--cuda-host-only	只为主机编译 CUDA 代码。对非 CUDA 的汇编没有影响。
--cuda-noopt-device-debug	启用设备端调试信息生成。禁用 ptxas 优化。
--cuda-path-ignore-env	忽略环境变量以检测 CUDA 安装。
--cuda-path=< value >	指定 CUDA 安装路径。
-cl-single-precision-constant	将双精度浮点常数视为单精度常数。默认已设置，因为天数智芯加速卡对双精度提供有限支持。
-Xcuda-fatbinary < arg >	将 < arg > 参数传递到 fatbinary 调用。
-Xlinker < arg >	将 < arg > 参数传递到链接器。
-x < language >	将后续输入文件视为指定语言的文件。
-fno-strict-aliasing	禁用基于类型的别名分析。

天数智算软件栈兼容 Clang 13 。更多有关 Clang 13 的用法，请参考 Clang 13 文档 。

Note

天数智芯编译器基于 Clang，Clang 和 nvcc 的默认选项可能存在少量不同，所以编译器选项可能需要略作调整，详见《CUDA 程序迁移指南》。

2.7.2 PTXToLLVM 选项

PTXToLLVM 的作用是从 PTX 生成 LLVM IR，使包含 PTX 的源代码可以经编译在天数智芯平台上运行。**PTXToLLVM** 支持以下选项。

此文档仅供【zhangyj@skysolidiss.com.cn-天固信息安全系统（深圳）有限公司】查阅，请勿分享他人

PTXToLLVM 选项	描述
--iluvatarTarget	必选项。将 PTX 转化为 LLVM IR，以便把 PTX 迁移到天数智芯平台。

2.7.3 llc 选项

llc 是 LLVM 项目的静态编译器。它将 LLVM 源输入编译成指定体系结构的汇编语言。天数智算软件栈利用 llc 为.cu 文件中的 CUDA 代码生成 LLVM IR。

天数智芯编译器支持所有 **llc** 选项。有关更多信息，请参阅 LLVM 用户指南中 llc 章节。

以下是一些常用的选项，供您参考。

llc 选项	描述
-filetype=< value >	选择文件的类型。**asm** 发出一个汇编文件 (.s)。**obj** 发出本机 OBJ 文件 (.o) 。**null** 不发送, 供测试性能用。
--iluvatar-spill-sgpr-to-smem	如果子目标支持，如果 Scalar General Purpose Register (SGPR) 已经到达了最大的上限，则对 SGPR 的使用将被溢出到 Shared Memory (SMEM)。
-march=< string >	针对特定架构生成代码。指定针对天数智芯加速卡，您必须设置 **-march=bi** 。

2.7.4 lld 选项

lld 是 LLVM 项目中的链接器，它是系统链接器的替代，并且运行速度较快。天数智算软件栈利用 lld 来链接要生成的 OBJ 文件和.cubin 文件。

天数智芯编译器支持所有的 **lld** 选项。更多信息，请参阅 LLD 文档 并使用 **lld -flavor ld.lld --help** 查看帮助文档。

以下是一些常用的选项，供您参考。

lld 选项	描述
-flavor ld.lld	调用 Unix 链接器。
--no-undefined	报告无法解析的符号，即使链接器正在创建共享库。
--no-warn-missing-entry	不提示 entry 缺失。
--shared	构建共享对象。

此文档仅供【zhangyj@skysolidiss.com.cn-天固信息安全系统（深圳）有限公司】查阅，请勿分享他人

2.7.5 fatbinary 选项

fatbinary 是一个将设备代码组织成"fat 二进制文件"（fat binaries）的工具，这些文件能够保存同一 GPU 源代码的多个编译版本。CUDA fat 二进制文件可能包含多个 PTX 和.cubin 文件。在运行时，当设备功能启动时，CUDA 驱动程序将选择最合适的编译版本。

天数智芯编译器支持所有 **fatbinary** 选项。有关更多信息，请通过 **fatbinary --help** 命令查看帮助内容。

以下是一些常用的选项，供您参考。

fatbinary 选项	描述
--32	说明编译是 32 位的。
--64	说明编译是 64 位的。
--cuda	说明编译的是 CUDA。
--image=< keyword >=< string >	指定带有编译它们的配置文件的映像列表。 可用的 keyword 是: **file**, **profile**。 **file**： 指定 ELF 文件，是 ld 命令生成的 `.cubin` 文件。 **profile**： 指定架构，支持设置为 **ivcore11** ，代表智铠加速卡。
--ptx=< string >	从指定的 PTX 文件添加代码。

更多关于 fat 二进制文件的信息，请参阅 The CUDA Compilation Trajectory 。

2.8 参考

- LLVM 用户指南: Compiling with clang
- Clang 工具链: Tools

此文档仅供【zhangyj@skysolidiss.com.cn-天固信息安全系统（深圳）有限公司】查阅，请勿分享他人

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。

此文档仅供【zhangyj@skysolidiss.com.cn-天固信息安全系统（深圳）有限公司】查阅，请勿分享他人