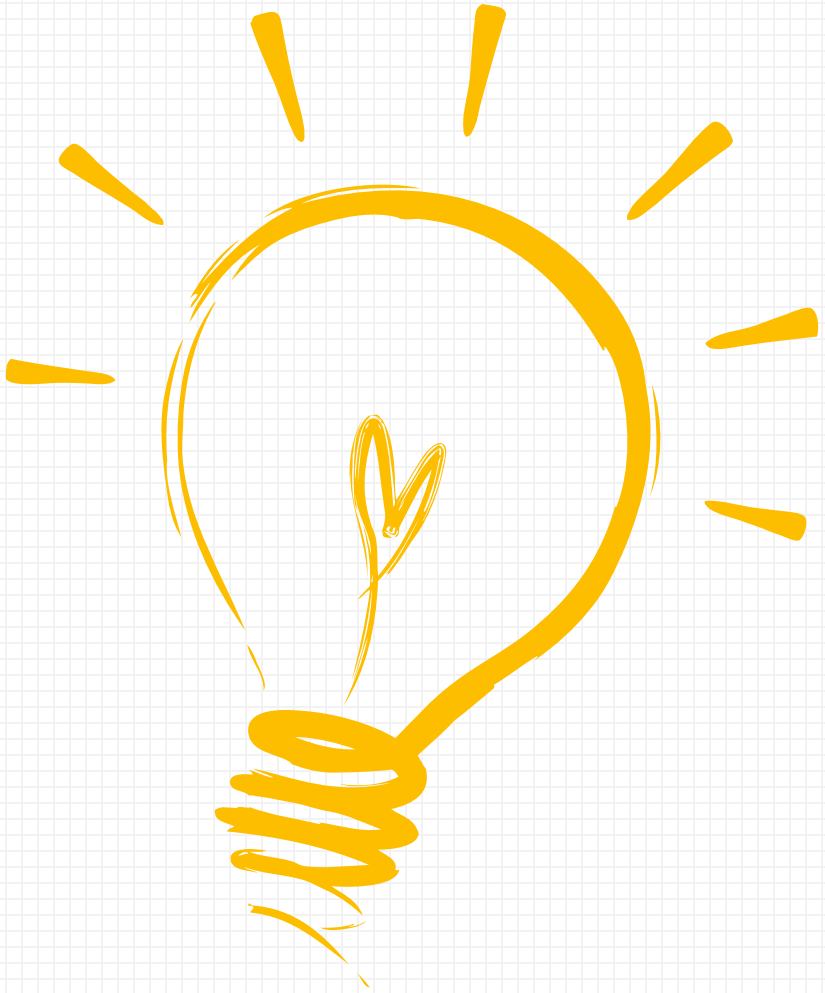
CUDA程序兼容性问题

CUDA并行编程系列课程

主讲：权双

CONTENTS



指定虚拟架构计算能力

★ C/C++源码编译为PTX时，可以指定虚拟架构的计算能力，用来确定代码中能够使用的CUDA功能

★ C/C++源码转化为PTX这一步骤与GPU硬件无关

★ 编译指令（指定虚拟架构计算能力）：

-arch=compute_XY

XY：第一个数字X代表计算能力的主版本号，第二个数字Y代表计算能力的次版本号

★ PTX的指令只能在更高的计算能力的GPU使用

例如:

nvcc helloworld.cu –o helloworld -arch=compute_61

编译出的可执行文件helloworld可以在计算能力>=6.1的GPU上面执行，在计算能力小于6.1的GPU则不能执行。

指定真实架构计算能力

★ PTX指令转化为二进制cubin代码与具体的GPU架构有关

★ 编译指令（指定真实架构计算能力）：

-code=sm_XY

XY：第一个数字X代表计算能力的主版本号，第二个数字Y代表计算能力的次版本号

★ 注意：（1）二进制cubin代码，大版本之间不兼容！！！

（2）指定真实架构计算能力的时候必须指定虚拟架构计算能力！！！

（3）指定的真实架构能力必须大于或等于虚拟架构能力！！！

~~nvcc helloworld.cu –o helloworld -arch=compute_61 -code=sm_60~~

★ 真实架构可以实现低小版本到高小版本的兼容！

指定多个GPU版本编译

★ 使得编译出来的可执行文件可以在多GPU中执行

★ 同时指定多组计算能力：

编译选项　-gencode arch=compute_XY –code=sm_XY

例如：

-gencode=arch=compute_35,code=sm_35	开普勒架构
-gencode=arch=compute_50,code=sm_50	麦克斯韦架构
-gencode=arch=compute_60,code=sm_60	帕斯卡架构
-gencode=arch=compute_70,code=sm_70	伏特架构

★ 编译出的可执行文件包含4个二进制版本，生成的可执行文件称为胖二进制文件（fatbinary）

★ 注意：（1）执行上述指令必须CUDA版本支持7.0计算能力，否则会报错

（2）过多指定计算能力，会增加编译时间和可执行文件的大小

nvcc即时编译

★ 在运行可执行文件时，从保留的PTX代码临时编译出cubin文件

★ 在可执行文件中保留PTX代码，nvcc编译指令指定所保留的PTX代码虚拟架构：

指令： -gencode arch=compute_XY ,code=compute_XY

注意：（1）两个计算能力都是虚拟架构计算能力

（2）两个虚拟架构计算能力必须一致

★ 例如： -gencode=arch=compute_35,code=sm_35

-gencode=arch=compute_50,code=sm_50

-gencode=arch=compute_61,code=sm_61

-gencode=arch=compute_61,code=compute_61

★ 简化： -arch=sm_XY

等价于 -gencode=arch=compute_61,code=sm_61

-gencode=arch=compute_61,code=compute_61

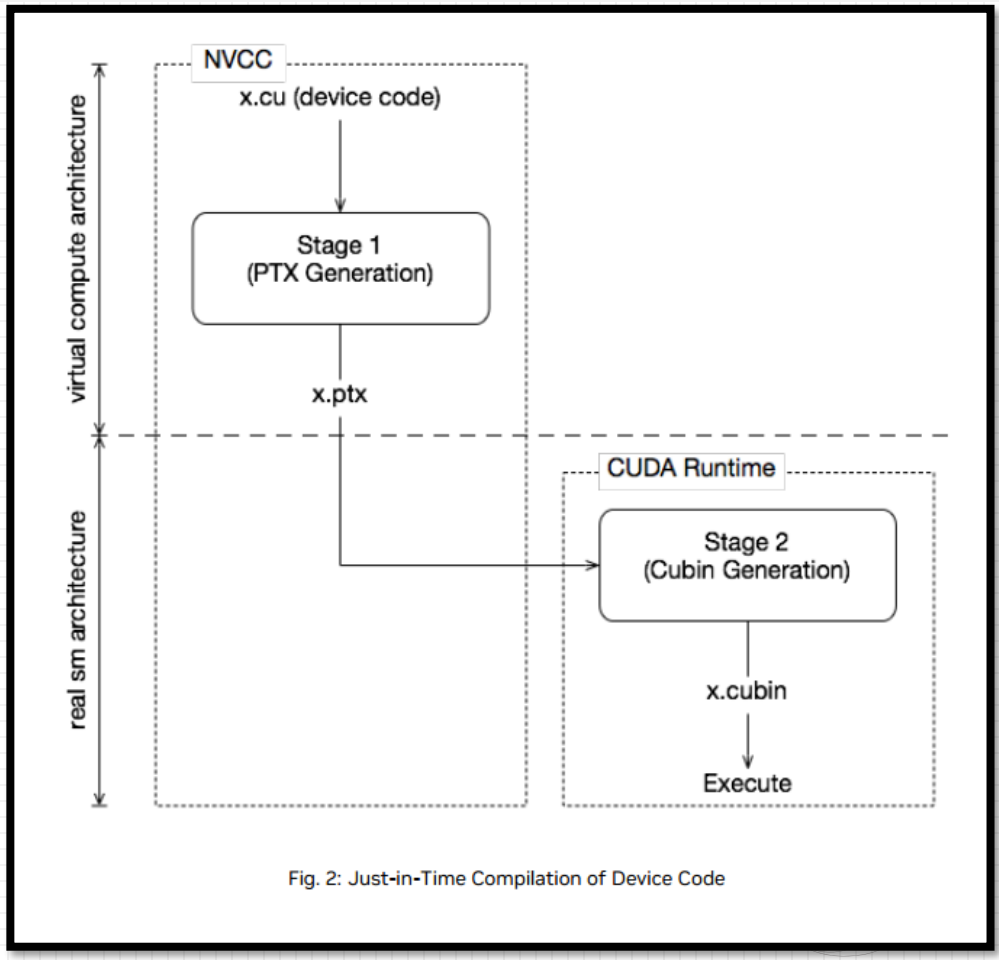


Fig. 2: Just-in-Time Compilation of Device Code

nvcc编译默认计算能力

★ 不同版本CUDA编译器在编译CUDA代码时，都有一个默认计算能力

★ CUDA 6.0及更早版本：　　默认计算能力1.0

CUDA 6.5~~~CUDA 8.0：　　默认计算能力2.0

CUDA 9.0~~~CUDA 10.2：　　默认计算能力3.0

CUDA 11.6：　　默认计算能力5.2

THANKS

谢谢聆听