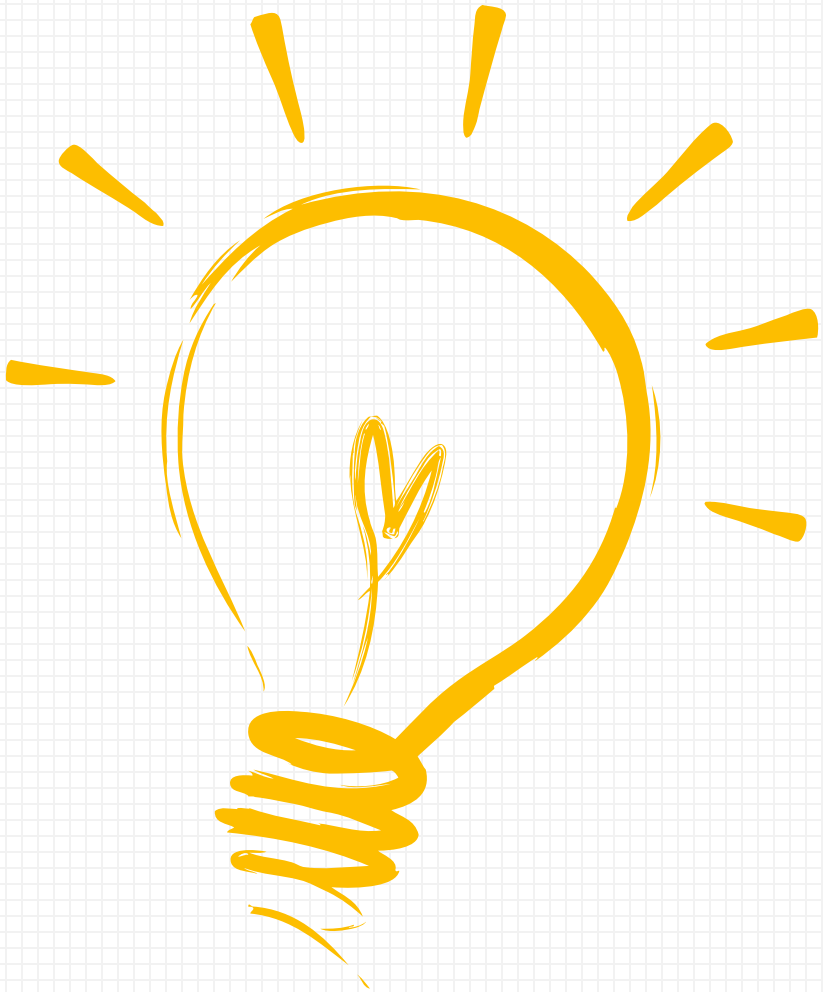
nvcc编译流程与GPU计算能力

CUDA并行编程系列课程

主讲：权双

CONTENTS



nvcc编译流程

★ 1、nvcc分离全部源代码为：（1）主机代码 （2）设备代码

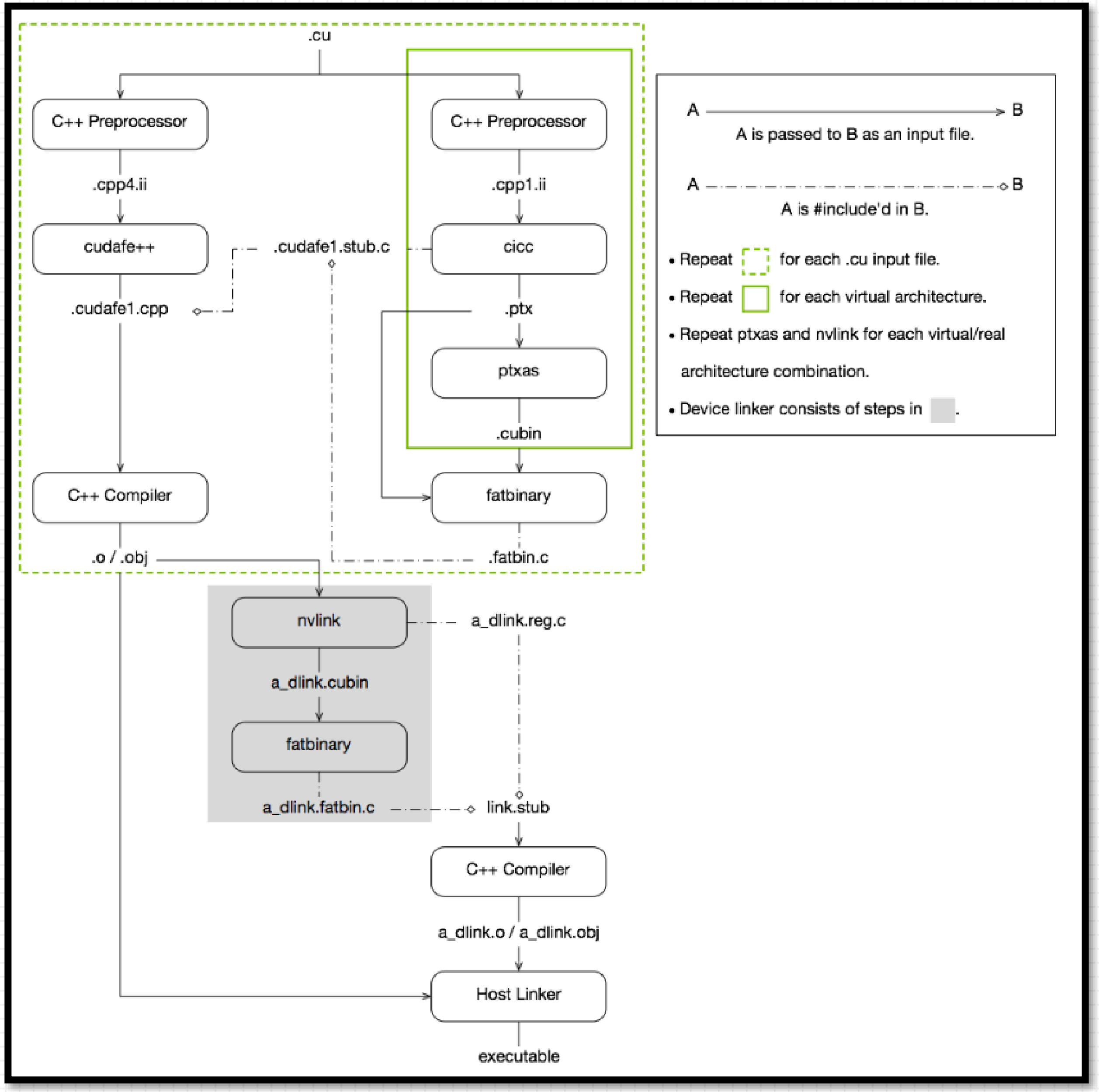
★ 2、主机（Host）代码是C/C++语法，设备（device）代码是C/C++扩展语言编写

★ 3、nvcc先将设备代码编译为PTX（Parallel Thread Execution）伪汇编代码，再将PTX代码编译为二进制的cubin目标代码

★ 4、在将源代码编译为 PTX 代码时，需要用选项-arch=compute_XY指定一个虚拟架构的计算能力，用以确定代码中能够使用的CUDA功能。

★ 5、在将PTX代码编译为cubin代码时，需要用选项-code=sm_ZW指定一个真实架构的计算能力，用以确定可执行文件能够使用的GPU。

nvcc编译流程



★ 具体cuda编译链接流程参考：https://docs.nvidia.com/cuda/cuda-compiler-driver-nvcc/index.html

★ 包含编译流程，编译指令

PTX

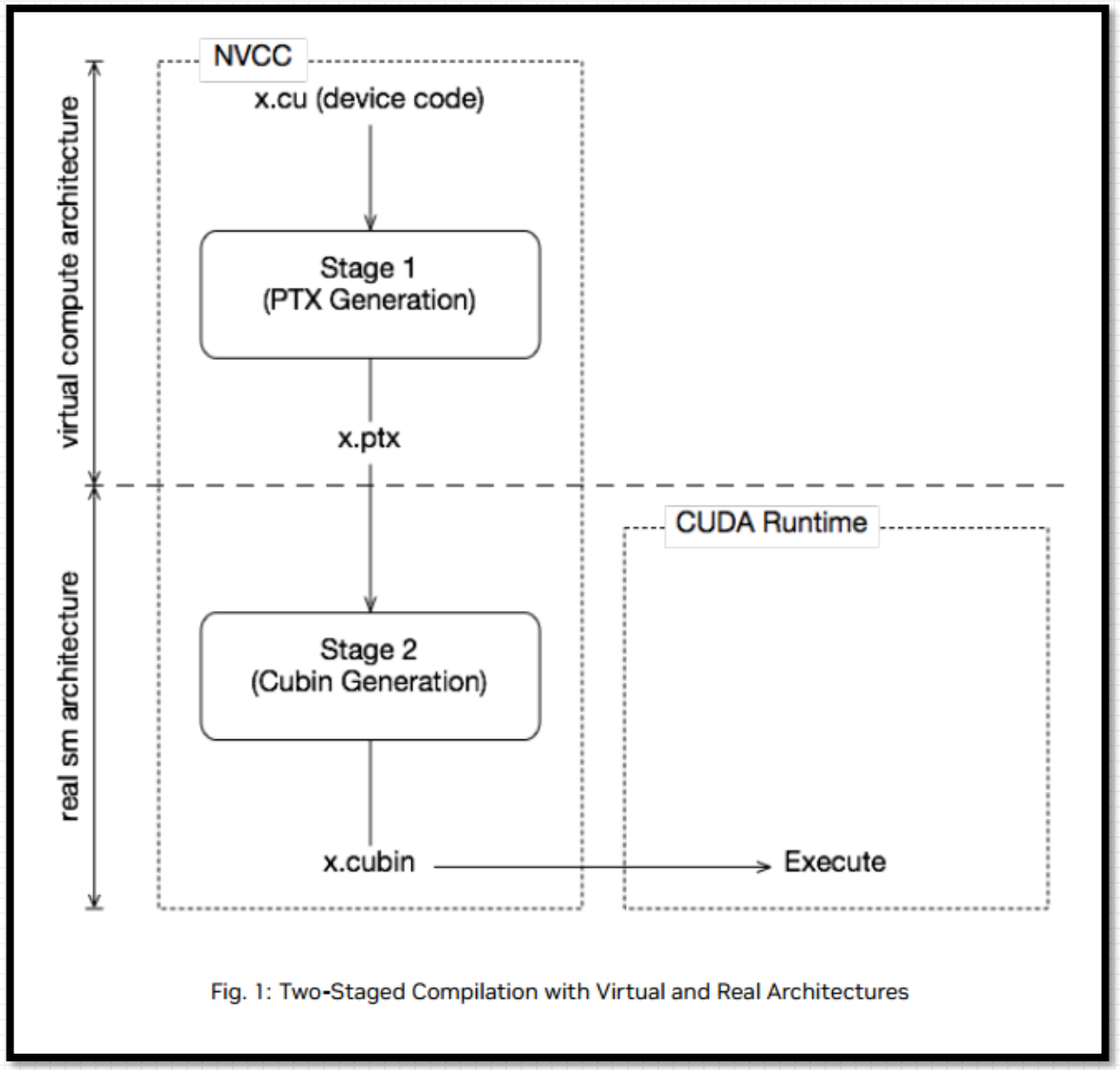


Fig. 1: Two-Staged Compilation with Virtual and Real Architectures

- PTX（Parallel Thread Execution）是CUDA平台为基于GPU的通用计算而定义的虚拟机和指令集
- nvcc编译命令总是使用两个体系结构:一个是虚拟的中间体系结构，另一个是实际的GPU体系结构
- 虚拟架构更像是对应用所需的GPU功能的声明
- 虚拟架构应该尽可能选择低----适配更多实际GPU
  真实架构应该尽可能选择高----充分发挥GPU性能
- PTX 文 档： https://docs.nvidia.com/cuda/parallel-thread-execution/index.html

GPU架构与计算能力

表 1.1: 各个 GPU 主计算能力的架构代号与发布年份。

主计算能力	架构代号	发布年份
X = 1	特斯拉（Tesla）	2006
X = 2	费米（Fermi）	2010
X = 3	开普勒（Kepler）	2012
X = 5	麦克斯韦（Maxwell）	2014
X = 6	帕斯卡（Pascal）	2016
X = 7	伏特（Volta）	2017
X.Y = 7.5	图灵（Turing）	2018

sm_50, sm_52 and sm_53	Maxwell support
sm_60, sm_61, and sm_62	Pascal support
sm_70 and sm_72	Volta support
sm_75	Turing support
sm_80, sm_86 and sm_87	NVIDIA Ampere GPU architecture support
sm_89	Ada support
sm_90, sm_90a	Hopper support

- 1、每款GPU都有用于标识“计算能力”（compute capability）的版本号
- 2、形式X.Y，X标识主版本号，Y表示次版本号

GPU架构与计算能力

★ 并非GPU 的计算能力越高，性能就越高

表 1.3: 若干 GPU 的主要性能指标。

GPU 型号	计算能力	显存容量	显存带宽	浮点数运算峰值
Tesla K40	3.5	12 GB	288 GB/s	1.4 (4.3) TFLOPS
Tesla P100	6.0	16 GB	732 GB/s	4.7 (9.3) TFLOPS
Tesla V100	7.0	32 GB	900 GB/s	7 (14) TFLOPS
GeForce RTX 2070	7.5	8 GB	448 GB/s	0.2 (6.5) TFLOPS
GeForce RTX 2080ti	7.5	11 GB	616 GB/s	0.4 (13) TFLOPS

THANKS

谢谢聆听