第2章 構造/関数/抽象化技術/ライブラリ

サンプルを動かしながら理解するCUDAの技術

鈴木 量三朗

GPU上の並列コンピューティング・プラットホームとしてNVIDIAはCUDAを提供しています．使えるGPUはNVIDIA製に限られてしまいますが，GPUの世代やランクによってプログラムを大きく変更する必要がないため，共通のプログラムでアプリケーションを組むことができます．

筆者は開発用のPCとしてWindows 10とLinux（Ubuntu 20.04）を使っています．両OSともにCUDAの開発環境が用意されており，同じソースコードを使用できます．

GPUのハードウェアとしては，Jetson Nano，GeForce GTX 750 Ti/RTX 2080/RTX 3060を使っています．Jetson Nanoは単体で使用し，他はPCと組み合わせて使用します．これらの異なるOS，異なるハードウェアに対して，同じソースコードを利用できます．

本章では，CUDAをインストールし，サンプル・プログラムを動かしながら，CUDAの技術的な背景も紹介します．

実際に使ってみることでGPU開発のプラットホームとして何が用意されていて，どのように使うのかをイメージできるでしょう．

環境の構築はインストールするだけ

● PCの場合

WindowsやUbuntuにインストールする場合は，次のウェブ・ページでCPUのアーキテクチャやOSとディストリビューションのバージョンなどを選択し，使っている環境に該当するインストーラをダウンロードおよびインストールします．

```
https://developer.nvidia.com/
                        cuda-downloads
```

ここではPC（i686，Linux）向けの環境を中心に説明を進めます．筆者は次のインストーラをダウンロードしました．

```
Linux/x86_64/Ubuntu/20.04/deb(local)
```

インストールについては，NVIDIAのデベロッパ・サイトのCUDA-Zone(1)にあるCUDAの公式情報も参考になります．

● Jetson Nanoの場合

Jetson Nanoは，Linuxの環境一式をSDカードに書き込んで使用します．CUDAを使う場合は，Jetson Nano用のウェブ・サイト(2)の指示に従い，イメージをSDカードにインストールします．

まずはCUDAのコンパイラでHello World

● コンパイラの場所

CUDAをLinuxにインストールすると，/usr/local/cuda注1が生成されます．その中の/usr/local/cuda/bin/nvccがCUDA用のコンパイラです．

● コンパイルおよび実行

CコンパイラのNVCC（NVIDIA CUDA Compiler）を使ってみます．次のような簡単なHello Worldプログラムをmain.cというファイル名で作り，コンパイルしてみます．

```
void
main()
{
      printf("Hello World\n");
}
```

次のコマンドでコンパイルおよび実行します．

```
$ /usr/local/cuda/bin/nvcc main.c↵
$ ./a.out↵
Hello World
```

a.outという実行用のバイナリができるので実行してみると，おなじみのHello Worldが無事表示されました．このプログラムはHello Worldを表示するだけなのでGPUは全く使っていません．CUDAで提供されているのが普通のCコンパイラらしいことが分かります．

注1：これはシンボリック・リンクです．実体は今回の場合/usr/local/cuda11．複数のバージョンのCUDAをインストールすることが配慮されています．