

GPUプログラムの 開発環境&動作原理

[ご購入はこちら](#)

本橋 弘臣

表1 GPGPU向けソフトウェア開発環境

ラズベリー・パイでもOpenCLが使えるようになった

項 目	Windows		Linux		
	CUDA	DirectX*1	CUDA	OpenCL	OpenGL ES
エヌビディア製GPU	○	○	○	△	—
AMD製GPU	×	○	×	○	—
SoC内蔵GPU	—	—	—	×→○*2	○

*1: DirectX 11に含まれるDirectCompute APIを利用する

*2: clvkの利用によりOpenCL対応が○に変わる(ラズベリー・パイ4/5の場合)

第2部では、ラズベリー・パイ4/5のGPU向けプログラミング環境として、並列処理プログラミングでも一般的に使用されるOpenCLを利用します。

本稿では、GPGPU向けソフトウェア開発環境やラズベリー・パイでOpenCLが使える理由、GPUでプログラムが動作する仕組みなど、実際にプログラムを書く前に知っておきたい基礎知識を解説します。(編集部)

GPGPU向けソフトウェア 開発環境の基礎知識

● 開発環境の種類

表1にGPGPUプログラミングを行う際に利用するソフトウェア開発環境を示します。

▶ CUDAとOpenCLの2つが主流

エヌビディアのGPUを利用する場合は、CUDA(Compute Unified Device Architecture)を利用するのが一般的です。CUDAは、さまざまな便利で高速なライブラリ(行列演算/FFT/機械学習など)が提供されていることと、ユーザが多いためGPU活用のためのさまざまな技術情報が手に入れやすいことがメリットです。

AMDやインテルのGPUを使用する場合は、GPGPU向けのAPIを定めたOpenCL(Open Computing Language)を使用するのが一般的です。

表2 OpenCLの概要

「一度ソースコードを書けば、どのハードウェアでも動作する」の実現を目指す並列コンピューティング向けAPI。クロス・プラットフォームで使える

項 目	説 明
目 的	演算処理を高速化するためのさまざまなアクセラレータを、統一されたAPIでプログラミングすること。Open Computing Languageの略
対象デバイス	GPU/CPU/DSP/FPGA など
コード記述言語	C/ (C++)
特 徴	○ ソースコードのコンパイルをプログラム実行時に行うことで*、同一のプログラムをさまざまなデバイスで動かすことができる(FPGAなど、一部例外あり) × 演算資源の個数や特性が異なるデバイスを統一APIで取り扱うところに若干無理がある(例: GPU向けに最適化したソースコードは、FPGAでは効率良く動作しないなど)

*: コンパイルを実行時に行うという仕様は、プログラム・コードの記述が煩雑になるという点ではデメリットでもある

▶ OpenCLはラズパイでも使える!

ラズベリー・パイに搭載されているようなSoC内蔵GPUの場合は、これまではグラフィックス処理用のOpenGL ESを使用するか、難解でとっつきにくいVulkanを利用するしかありませんでした。今回紹介するclvkを利用することにより、OpenCL対応になります。

● OpenCL…さまざまなデバイスに対応する並列コンピューティング向けAPI

▶ クロス・プラットフォームで使える

プログラミング言語Javaは「Write once, run everywhere(一度プログラムを書けば、どこでも実行できる)」という世界の実現を目標としていました。このJavaのコンセプトのハードウェア版がOpenCLです。

OpenCLは「一度ソースコードを書けば、どのハードウェアでも動作する」の実現を目指しています。表2にOpenCLの概要を示します。OpenCLの開発ターゲット・デバイスは幅広いですが、その中で最も多くOpenCLが活用されているデバイスはおそらく