第5部 ベンダを選ばないAPIで実験

ダウンロード・データあります

第1章 C言語やFortran言語に指示文を付け加えることでGPU化コードにしてくれる

OpenMPとOpenACCを試す

佐藤 三久

OpenACC・OpenMPが求められるようになった理由

● CUDAの登場でGPUは普及した

2000年代後半から急速な高性能化を遂げたGPUですが，普及を後押ししたのは，なんといってもプログラミングができるようになったということです．従来は主な用途はビデオ・ゲームにおいて高速な処理を実現することでした．それに限れば，OpenCLなどの標準的な画像処理のライブラリをゲーム・ベンダが使えればよい状況でした．

本特集でもCUDAに関する記事がありますが，CUDAは本来の名前Compute Unified Device Architectureの通り，GPUのアーキテクチャに合わせた言語モデルであり，GPUに合わせた性能を引き出すプログラミングが可能です．

● 既存のアプリを書き換えて利用するには開発コストが高くなり難しい

GPUのプログラミングは，もともとCPUにおいてループで書いていた部分をカーネル関数として取り出し，それの制御をホストから行うなど，CPUのプログラミングとはだいぶ違います．一から，GPU向けに新たなアプリケーションの開発をするとか，GPUで行う処理が一部のループに限られる場合には問題は少ないかもしれませんが，既存のアプリケーションを書き換えるとなると，なかなか開発コストが高くなり難しいということになります．

それでも，頑張ってCUDAに書き換えて高速化に成功した人たちがいたので，GPUが普及してきたとも言えます．

● CPUでもマルチコアが一般的…並列のプログラムが求められる

CPUのプログラミングにおいても，現在マルチコア・プロセッサが一般的になっています．当初はやはりプログラミング環境が問題になっていました．OSから直接提供されているAPIであるpthreadライブラリを使ってもプログラムができるのですが，これはCUDAで書き換える作業よりもさらに手間がかかるものでした．

リスト1　OpenMPによるループ並列化の例

```
#pragma omp parallel for
for(i = 0; i < N; i++)
  c[i] =a[i]+b[i]
```

● OpenMPはマルチコア向けの並列化されたコードを生成してくれる

そこで提案されたのがOpenMP（Open Multiprocessing）です．OpenMPは新しい言語ではなく，科学技術計算でよく使われるC言語やFortran言語に指示文を付け加えることによって，マルチコア向けの並列化されたコードにしてくれるというものです．

リスト1は，簡単なOpenMPのプログラム例です．配列aとbとを要素ごとに足して，配列cに格納するというループを並列化したものです．C言語では指示文は，`#pragma omp`というように，`pragma`行として書きます．見ての通りですが，元のプログラムの並列化したい部分に指示文を加えればよいので，楽に並列化できます．

このアプローチでGPUプログラミングをできるようにしたのがOpenACC（Open Accelerators）というフレームワークです．実は，2011年にOpenACCが発表された後，後追いで2013年にOpenMPも拡張され，OpenMP 4.0でGPUをプログラミングできる拡張が導入されました（詳しくは第1部第1章を参照）．本稿ではそれぞれを使う場合に，具体的にどのようなコードを書くのかを見ていきます．

APIその1…OpenACC

● OpenMPのプログラムを書き換えてみる

リスト2のプログラムが，先ほどのOpenMPのマルチコア・プログラミングの例をOpenACCで書き換えたものです．OpenACCの指示文は`#pragma acc`で始まります．ループの前に，`#pragma acc kernels`