ディレクティブベースプログラミング言語 OpenACC の性能評価

星野 哲也[†1]　丸山 直也[†1,†2,†3]　松岡 聡[†1,†3,†4]

1. はじめに

近年における計算環境の変化は著しく，特に演算用にアクセラレータを用いたヘテロジニアス型スーパーコンピュータが台頭している．これらの多くはアクセラレータとして，GPU に代表されるメニーコアアーキテクチャを採用している．GPU は演算性能に対する電力性能が高いことで知られており，電力が大きな制約条件となる大型システムにおけるアクセラレータの利用は，今後より一般的になると考えられる．

しかしここで問題となるのが，従来の CPU 向けに開発されたアプリケーションのアクセラレータ向けの移植である．GPU などのアクセラレータ向けプログラミング言語として，現在では CUDA・OpenCL が主流であるが，これらはアーキテクチャを意識した低レベルな記述を必要とする．この問題を解決するため，元のプログラムに指示文を挿入することでアクセラレータを利用することができる，OpenACC[1] が注目されている．OpenACC を使うことにより，プログラムの生産性・移植性の向上が期待されているが，OpenACC を用いた研究は現在までにあまりなされておらず，どの程度の性能が得られるかは明らかになっていない．本稿では OpenACC の性能を理解するために，マイクロベンチマークと実アプリケーションを OpenACC・CUDA を用いてアクセラレータ向けに移植・最適化を施し，両者の比較を行った．

2. 実験・評価

性能評価実験として，マイクロベンチマークとして行列積，7 点ステンシル，実アプリケーションとして，独立行政法人宇宙航空研究開発機構 JAXA により開発されている UPACS[2] と呼ばれる CFD アプリケーションの OpenACC，CUDA を用いた移植・最適化を行い，TSUBAME2.0 の GPU を用いて実験を行った．**図 1**，**図 2**，**図 3** のグラフはそれぞれ，行列積のパフォーマンス (GFlops)，7 点ステンシルのスループット (GB/s)，UPACS で Navier-Stokes 方程式を解く上での主要フェーズの実行時間を示している．

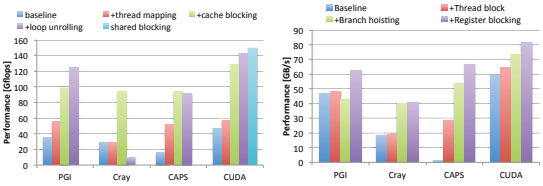

図 1　行列積

図 2　7 点ステンシル

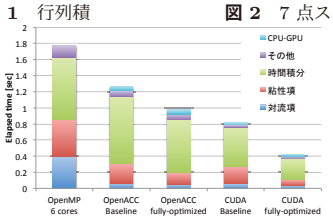

図 3　UPACS

その結果，マイクロベンチマークにおいて PGI コンパイラでは CUDA と比較して 70~80%の性能が得られ，CUDA において有効な最適化は，OpenACC でも同様に重要であることを確認した．同様に PGI コンパイラを用いた UPACS では，最適化を施していない場合には 77%程度の性能が得られたが，両者を最大限まで最適化した場合，40%程度の性能しか得られないことを確認した．これは OpenACC に共有メモリを制御する機能がないためであった．

3. おわりに

現在の OpenACC コンパイラでは，マイクロベンチマークでは CUDA に近い性能が得られることを確認したが，OpenACC の機能の制限により，両者の性能に乖離が生じることを確認した．

参考文献

1) The openacc application programming interface, version 1.0, November 2011.
2) Ryoji Takaki, Kazuomi Yamamoto, Takashi Yamane, Shunji Enomoto, and Junichi Mukai. The development of the upacs cfd environment. In *High Performance Computing*, volume 2858 of *Lecture Notes in Computer Science*, pages 307–319. 2003.

†1 東京工業大学
†2 理化学研究所
†3 独立行政法人科学技術振興機構 CREST
†4 国立情報学研究所